



SaaS Fundamentals

A Guide to Modern Cloud Solutions



DAVID HANES
CHAD PATTERSON
PAUL GIRALT
OMAR SANTOS



SaaS Fundamentals

A Guide to Modern Cloud Solutions



ciscopress.com

DAVID HANES
CHAD PATTERSON
PAUL GIRALT
OMAR SANTOS

SaaS Fundamentals: A Guide to Modern Cloud Solutions

David Hanes, CCIE No. 3491

Omar Santos

Paul Giralt, CCIE No. 4793

Chad Patterson

Cisco Press

A NOTE FOR EARLY RELEASE READERS

With Early Release eBooks, you get books in their earliest form—the author’s raw and unedited content as they write—so you can take advantage of these technologies long before the official release of these titles.

If you have comments about how we might improve the content and/or examples in this book, or if you notice missing material within this title, please reach out to Pearson at PearsonITAcademics@pearson.com

SaaS Fundamentals: A Guide to Modern Cloud Solutions

David Hanes, Omar Santos, Paul Giralt, Chad Patterson

Copyright © 2026 Cisco Systems, Inc.

Published by:

Cisco Press

Hoboken, New Jersey

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit <https://www.pearson.com/global-permission-granting.html>.

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

`$PrintCode`

Library of Congress Control Number:

ISBN-13: 978-0-13-533474-4

ISBN-10: 0-13-533474-8

Warning and Disclaimer

This book is designed to provide information about the core technologies and Cisco products that make up Software as a Service (SaaS). Every effort has been made to make this book as complete and as accurate as possible, but no

warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers’ feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Please contact us with concerns about any potential bias at <https://www.pearson.com/report-bias.html>.

Head of IT & Professional Learning, Enterprise Learning and Skills

Julie Phifer

Cisco Alliance Manager

Caroline Antonio

Executive Editor

James Manly

Managing Editor

Sandra Schroeder

Development Editor

Ellie C. Bru

Senior Project Editor

Tonya Simpson

Copy Editor

Chuck Hutchinson

Technical Editor

Trevor Mays

Cover Designer

Chuti Prasertsith

Composition

Indexer

Proofreader

About the Authors

David Hanes, CCIE No. 3491, is a principal engineer in the SaaS Support Engine at Cisco, supporting the Cisco Technical Assistance Center (TAC). Specializing in SaaS, Internet of Things (IoT), and collaboration technologies, he is focused on workflow innovation, solution architecture, and technical solution development. David is the coauthor of two books: *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things* (Cisco Press, 2017) and *Fax, Modem, and Text for IP Telephony* (Cisco Press, 2008).

He has contributed to industry standards bodies such as the IETF and the SIP Forum and holds more than 50 patents, both pending and issued, across the domains of collaboration, IoT, security, AI/ML, wireless, and sustainability. David's technical leadership is recognized globally, and he has been invited to speak at numerous international conferences on collaboration and IoT topics. At Cisco Live, the company's flagship customer event, he has delivered more than 30 presentations and has earned seven Distinguished Speaker awards as well as induction into the Hall of Fame.

Since joining Cisco in 1997, David has served as a TAC engineer and team lead for both the WAN and Multiservice Voice teams, and as an escalation engineer specializing in a wide range of VoIP technologies. He has resolved complex collaboration issues for many of Cisco's largest customers worldwide. Prior to Cisco, David began his networking career as a systems engineer at Sprint, where he worked with frame relay and X.25 protocols. He holds a bachelor of science in electrical engineering from North Carolina State University.

Omar Santos is a distinguished engineer at Cisco focusing on artificial intelligence (AI) security, cybersecurity research, incident response, and

vulnerability disclosure. He is the cochair of the Coalition for Secure AI (CoSAI) and a board member of the OASIS Open standards organization. Omar is also the chair of the OpenEoX and the Common Security Advisory Framework (CSAF) technical committee. His work led the creation of the CSAF ISO standard.

Omar's collaborative efforts extend to numerous organizations, including the Forum of Incident Response and Security Teams (FIRST) and the Industry Consortium for Advancement of Security on the Internet (ICASI). Omar is the cochair of the FIRST PSIRT Special Interest Group (SIG) and was the lead of the DEF CON Red Team Village for several years.

Omar is the author of more than 25 books, 20 video courses, and more than 40 academic research papers. His work in cybersecurity is also recognized through multiple granted patents. Prior to Cisco, Omar served in the United States Marines focusing on the deployment, testing, and maintenance of Command, Control, Communications, Computer, and Intelligence (C4I) systems.

Paul Giralt, CCIE No. 4793, is a distinguished engineer in the Cisco Security and Trust Organization (S&TO) focused on making Cisco's software secure; however, he spent most of his decades-long career at Cisco supporting all aspects of Cisco's collaboration portfolio, from the early days of CallManager 3.0 to modern day cloud-based collaboration, including Webex. He has spent much of his career helping customers accelerate the adoption of Cisco technologies and solutions, working with Cisco's largest collaboration deployments and working closely with engineering on improving product serviceability and cross-architecture product integrations. Paul has spent years designing, troubleshooting, and diagnosing issues on some of the largest and most complex Cisco collaboration deployments. His current focus is on making industry-leading changes to the security posture of Cisco products in the face of evolving threat actor sophistication leading network infrastructure that is resilient to attacks.

Paul holds a degree in computer engineering from the University of Miami, is the author of multiple Cisco Press titles, is an IETF contributor, and holds numerous patents in collaboration and networking technologies.

Chad Patterson is a technical leader in Cisco's Customer Experience SaaS

Engine, with more than 12 years of expertise in cloud collaboration technologies, SaaS security, and full-stack observability. He specializes in supporting and innovating across many Cisco products, including Webex and Telepresence, building solutions that enable customers and support teams to detect and resolve issues faster.

Chad is recognized for his contributions to technical innovation, having developed software tools and workflows that streamline troubleshooting and enhance customer outcomes. He is a prolific author of technical documents, white papers, and training materials, and has been responsible for developing and delivering curriculum to launch new collaboration and IoT support teams.

An active participant in Cisco's technical community, Chad has presented at major industry conferences, including Cisco Live and Cisco Impact, where he leverages his domain knowledge to empower customers and sales teams. He is also dedicated to mentoring colleagues and driving cross-functional innovation within Cisco's Customer Experience organization.

Chad holds a bachelor of science in electrical engineering and a bachelor of science in computer engineering from North Carolina State University.

About Contributing Authors

Kevin D. McCabe is a principal engineer in Cisco's CX Engineering organization and a dedicated expert in data center technology. Since joining Cisco in 2013, Kevin has advanced from TAC engineer to technical leader and now principal engineer, leading the development of critical Intersight features such as Connected TAC (Rapid Problem Resolution) and Proactive RMA. He holds a degree in textile engineering with a concentration in information systems from North Carolina State University and is a two-time recipient of Cisco's Pinnacle Award (Cisco's highest engineering award) for his work on the Automated Problem Detection Engine and the Cisco AI Assistant for Support. Based in Raleigh, North Carolina, Kevin is a chapter author in this volume, a proud father of twin daughters, a devoted Notre Dame football fan, and a passionate bourbon collector.

Justin L. Pierce attended Penn State University, where he majored in security and risk analysis with a focus in information and cyber security. After graduating in 2014, he moved to RTP, North Carolina, and accepted a position at Cisco working as a technical consulting engineer supporting UCS products on the Server Virtualization team. This is where he found his passion for the data center technology space and helping customers solve complex problems. He currently is a technical leader in Cisco's Technical Assistance Center. He lives with his wife, Laci, and their dog and cat.

About the Technical Reviewers

Trevor Mays is a seasoned cybersecurity executive with deep expertise in SaaS customer experience and technical documentation. Trevor was an early employee driving the hyper-growth of Duo Security to its eventual \$2.35B acquisition by Cisco in 2018. When he's not working or writing, Trevor is an avid practitioner of yoga and meditation. He lives in the woods near Ann Arbor, Michigan, with his wife, daughter, and their rescue hound.

Ana Montenegro, CCIE DC No. 68117, is a storage & compute technical leader with more than 10 years of experience in Cisco Data Center Solutions. She holds a master of science in telecommunications and information technology management. Throughout her career, Ana has contributed to public documentation and videos on the configuration and troubleshooting of data center devices, supported TAC teams during multiple escalations, and developed tools to simplify troubleshooting.

Adam Newman is a staff product manager at ThousandEyes, bringing extensive product management experience from leadership roles at ThousandEyes, New Relic, and Akamai. Known for a strong customer focus, he excels at translating customer needs into impactful product solutions and driving innovation across complex, distributed systems.

Mike Hicks is a principal solution analyst at Cisco ThousandEyes and is a recognized expert in network and application performance, with more than 30 years of industry experience supporting large, complex networks and working closely with infrastructure vendors on application profiling and management. He is the author of *Managing Distributed Applications: Troubleshooting in a Heterogeneous Environment* (Prentice Hall, 2000) and *Optimizing Applications on Cisco Networks* (Cisco Press, 2004).

Biswajit Nanda is a senior technical leader at Cisco Systems Inc. and has

more than 22 years of industry experience in multiple roles as software development engineer, DevOps engineer, site reliability engineer, platform engineer, and support engineer, specializing in Java, Python, OpenSource observability, cloud-native technologies and multicloud operations, and container orchestration. He is a certified Kubestronaut and also holds several certifications in AWS, Azure, and CNCF technologies. He has been a guest speaker at KubeCon/Cloud-NativeCon and a cloud-native ambassador. He majored in computer applications at Biju Patnaik University of Technology in India and lives in McKinney, Texas, with his wife and son. He loves long cross-country road trips, exploring national parks, and off-roading with his 4×4.

Vishal Goyal is an engineering graduate in electronics and communications who began his career working on BroadWorks IP telephony infrastructure before transitioning into the Cisco collaboration ecosystem. He has developed deep expertise across Cisco collaboration and played a pivotal role in establishing and anchoring the technical support framework for Cisco's first cloud contact-center platform, Webex Contact Center. Vishal currently leads Webex Contact Center technical operations, focusing on bridging product capabilities with customer expectations and driving continuous improvement across the ecosystem. He remains committed to ongoing learning and to advancing customer and partner experience through thoughtful, scalable solutions.

Dedications

David Hanes:

To my beloved wife, Holly, thank you for your unwavering love, support, and your dedication to our family and the life we have built together. You are a true blessing, and your strength and encouragement make all things possible.

To my wonderful children, Haley, Hannah, and Kyle: You bring me more joy and happiness than you could ever know. Your talents continually amaze me, and your admirable character and pursuit of excellence in all that you do fill me with immeasurable pride.

Omar Santos:

I would like to dedicate this book to my lovely wife, Jeannette, and my two beautiful children, Hannah and Derek, who have inspired and supported me throughout the development of this book. Their inspiration and support have been the greatest gift of this journey and everything that matters most in life.

Paul Giralt:

For Archana, Rohen, and Maya, who make everything worth it.

Chad Patterson:

I would like to dedicate this book to my amazing wife, Emily, whose constant love, encouragement, and support allowed me to complete this work. And to my sons, Cade and Colter, I am beyond blessed to be your father, and you bring so much joy to my life. I love you all beyond measure.

I would also like to dedicate this book to my parents, Eric and Beverly. Thank you for inspiring me even as a child to be creative, to think critically,

and to love to learn.

To God be all the glory and praise!

Acknowledgments

David Hanes: First and foremost, I would like to thank my coauthors on this book, Chad, Paul, and Omar. I am deeply grateful for their dedication, expertise, and the countless hours they invested in bringing this project to fruition. Special thanks to Chad for being the driving force behind this endeavor and for his steadfast consistency throughout the process. His technical acumen and ability to execute at the highest level continue to impress me. I am thankful to Paul for his loyal friendship, unwavering guidance, and continual example of professionalism and technical excellence. Even after more than 20 years of collaboration, he remains a true inspiration. Thanks as well to Omar for generously sharing his insights and wisdom at every stage of this journey. It was a privilege to work alongside him and to learn from his experience as both a consummate expert and accomplished author.

I extend my sincere thanks to Kevin McCabe and Justin Pierce for their significant contributions in writing [Chapter 14](#). Both are remarkably talented engineers and authors, and working with them was a wonderful experience. They deserve special recognition for developing such an outstanding chapter for this book.

A special thanks goes to Trevor Mays for his exceptional work in reviewing this book. His meticulous attention to detail and his technical breadth and depth in SaaS technologies were essential to achieving such a high level of quality. Trevor's countless hours of editing, along with his thoughtful suggestions and corrections, are greatly appreciated.

I would like to express my gratitude to my manager, Regina Moore, for her enduring support and exemplary leadership in advancing SaaS innovation. I deeply appreciate her guidance and encouragement throughout my innovation

projects. Being part of her team and collaborating with some of Cisco's most creative minds are truly a privilege.

Thank you to Neal Alsup for his support of this book and his innovative leadership. Neal's passion for innovation and his clear vision for the future underpin the creative culture in CX Centers that I value so highly.

Thanks to Marty Martinez, for being the "okayest" manager. In all seriousness, I would like to express my sincere appreciation to Marty for introducing me to the world of SaaS, for consistently encouraging me to strive for excellence, and for offering invaluable advice and support throughout my career.

Thank you to Ana Montenegro for generously lending her time and data center expertise to this project, and for reviewing the SaaS architecture sections of this book.

I would like to thank Adam Newman and Mike Hicks for their review of the ThousandEyes content. Their suggestions and edits were incredibly helpful.

Thanks to Rob Barton for his assistance in introducing the SaaS architecture.

I would like to extend a heartfelt thank you to all the engineers and technical leaders who support our SaaS products. These professionals deliver best-in-class support across Cisco solutions such as Webex, AppDynamics, and Umbrella. Their expertise and technical excellence are unsurpassed. I am deeply appreciative of their willingness to answer questions and share their knowledge throughout this project.

Finally, thanks to the production team, Ellie Bru, Nancy Davis, Tonya Simpson, and James Manly. Nancy took this initial idea and turned it into a reality. Ellie and Tonya provided a constant presence, editing and shepherding us through every stage of the authoring and production process. This book would not have happened without their dedication and expertise.

Paul Giralt:

I'd like to thank David Hanes for bringing this team of authors together to write this book. Without his leadership and dedication, it would not have been a reality. Also, thank you to the rest of the author team who have been great to work with on this project.

Thank you to Paul Stojanovski, Esteban Valverde, Luis Ramirez, and Paul Anholt, who have always been gracious to impart their knowledge of the Webex platform with me. I have learned a lot from all of you.

Thank you to Martin Pottie for always spending time to explain the inner workings of Webex Calling and always being willing to share deep technical information with everyone around you.

A huge thank you to Trevor Mays for his detailed review of the chapters and thoughtful comments that helped make the book that much better.

Thanks to the entire Cisco Press team—Ellie, Nancy, James, and Tonya—for keeping us on track and your guidance and understanding throughout the process.

Chad Patterson:

I would like to extend a special thank you to several people who supported me while writing this book.

First, I want to extend my heartfelt thanks to my coauthor, David Hanes, for inviting me to join this project and for guiding me every step of the way. David's mentorship, technical expertise, and unwavering support made this journey both rewarding and enjoyable. I am truly grateful for his partnership and encouragement throughout the writing of this book.

I would also like to thank Biswajit Nanda for his support in writing [Chapter 11](#). His countless reviews, insightful Webex discussions on AppDynamics and Splunk, and guidance on structuring the chapter were invaluable. [Chapter 11](#) would not have been possible without his expertise and dedication. I hope you are as proud of the result as I am; thank you for being an absolute rockstar!

To Amit Goyal, thank you for your valuable support in reviewing [Chapter 11](#) and for generously answering my questions about AppDynamics. Your insights and willingness to help were greatly appreciated.

I would also like to thank Vishal Goyal for his invaluable effort in reviewing [Chapter 7](#) on Webex Contact Center. His attention to detail, insights, and commitment to accuracy ensured the content was both relevant and precise. I am especially grateful for his help in deepening my understanding of the

architecture and for his support throughout this project.

Thank you to Trevor Mays for spending countless hours poring over each chapter as a technical reviewer. His years of experience working with SaaS technologies and attention to detail have made every chapter of this book better.

To my manager, Regina Moore—thank you for believing in me and for your unwavering support of both David and me throughout this journey. Your thoughtful guidance, encouragement, and commitment to our growth have made a lasting impact. I am grateful for your exceptional leadership and for the chance to be part of such an inspiring team.

Thank you to Neal Alsup for his encouragement and support from the very start of this project. When he first learned about the idea of this book, his immediate enthusiasm and approval meant a great deal to me. Neal's clear leadership and commitment to creativity, innovation, and customer focus continue to shape an environment where projects like this are possible.

And finally, heartfelt thanks to the entire production team—Nancy Davis, Ellie Bru, Tonya Simpson, and James Manly. From the start, Nancy saw the vision for this book and played a key role in getting the project off the ground. Ellie has been with us every step of the way, keeping us on track with deadlines, carefully editing each chapter, and providing invaluable support throughout the process. James, your contributions behind the scenes have helped everything come together seamlessly. Thank you all for your dedication and for helping us bring this book to life.

Omar Santos:

Many thanks to the technical reviewers and our incredible Pearson team: Ellie Bru, Nancy Davis, James Manly, and Tonya Simpson. Nancy took a spark of an idea and transformed it into a fully realized project. Ellie was the steady, guiding presence who edited, refined, and shepherded us through every stage of the journey. Their dedication, expertise, and tireless work made this book possible. I am deeply grateful.

Contents at a Glance

Part I: SaaS Fundamentals

- 1: What Is SaaS?
- 2: SaaS Architecture
- 3: Migrating to SaaS
- 4: Security and Privacy for SaaS

Part II: SaaS Solutions

- 5: Collaboration: Webex Meetings and Messaging
- 6: Collaboration: Webex Calling
- 7: Collaboration: Webex Contact Center and Webex Connect
- 8: Identity and Access Management
- 9: Cisco Umbrella and Cisco AI Defense
- 10: Cisco XDR, Splunk, and Cisco Vulnerability Management
- 11: Observability and Monitoring: AppDynamics and Splunk
- 12: Observability and Monitoring: Cisco Thousand Eyes
- 13: Management: Cisco Meraki
- 14: Management: Cisco Intersight

Reader Services

Register your copy at www.ciscopress.com/title/9780135334744 for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account*. Enter the product ISBN 9780135334744 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Contents

Part I: SaaS Fundamentals

Chapter 1. What Is SaaS?

- Cloud Types

- IaaS, PaaS, and SaaS Cloud Computing Models

- Everything as a Service

- Shared Responsibility Model

- The Business Case for SaaS

- Summary

- References

Chapter 2. SaaS Architecture

- Logical Model

- Architectural Model

- Multitenancy

- Summary

- References

Chapter 3. Migrating to SaaS

- Discovery

- Design and Planning

- Implementation

- Value Realization

- Common Migration Challenges

Summary

References

Chapter 4. Security and Privacy for SaaS

SaaS Security Basics

Regulatory Compliance and Certifications

Data Sovereignty

Architectural Considerations for Data Partitioning and Tenant Isolation

Tools and Techniques for Preventing Data Loss in a SaaS Setup

Identity and Access Management (IAM)

Continuous Monitoring and Incident Response

SaaS Security Management

Summary

Part II: SaaS Solutions

Chapter 5. Collaboration: Webex Meetings and Messaging

Product Capabilities

The Webex Platform

Summary

References

Chapter 6. Collaboration: Webex Calling

Product Capabilities

The Webex Calling Platform

Summary

References

Chapter 7. Collaboration: Webex Contact Center and Webex Connect

Cisco Cloud Contact Center Products

Product Capabilities
Webex Workforce Optimization
Webex Contact Center Platform
Summary
References

Chapter 8. Security: Identity and Access Management

Cisco's Zero Trust and Continuous Trust Philosophy
An Introduction to Cisco Duo
The Duo Cloud-Native Platform and On-Premises Components
The Cisco Duo Identity and Access Management (IAM) Platform
Summary
References

Chapter 9. Security: Cisco Umbrella and Cisco AI Defense

From OpenDNS to Cisco Umbrella
Cisco AI Defense: Securing the AI Revolution
Summary
References

Chapter 10. Security: Cisco XDR, Splunk, and Cisco Vulnerability Management

Unifying Telemetry for Accelerated Response Using Cisco XDR
Data-Driven Insights with the Splunk Ecosystem
Cisco Vulnerability Management: Prioritizing Risk with Data Science
The Power of Integration: A Unified Security Strategy
Cisco's AI Assistant
Summary

References

Chapter 11. Observability and Monitoring: AppDynamics and Splunk

The Basics

AppDynamics + Splunk

SaaS Security Practices—AppDynamics and Splunk

Splunk Observability Cloud—Architecture Overview

AppDynamics—Architecture Overview

Core Observability Features

Application Performance Monitoring (APM)

Infrastructure Monitoring

End-User Monitoring

Synthetic Monitoring

Log Observer Connect

Summary

References

Chapter 12. Observability and Monitoring: Cisco ThousandEyes

Architectural Overview

Agent Types

Agent Tests

Path Visualization and Dashboard Snapshots

Internet, WAN, Cloud, and Traffic Insights

Integrations

Summary

References

Chapter 13. Management: Cisco Meraki

Meraki Platform Capabilities

The Cisco Meraki Cloud

Summary

References

Chapter 14. Management: Cisco Intersight

Intersight Overview

Automation and Insights

Architecture

Summary

References

Icons Used in This Book



Cloud



Laptop



Headquarters



Firewall



Wireless Router



Database



Switch



Router

Introduction

The rapid evolution of cloud computing has transformed the way organizations consume, deliver, and manage technology. Among the various paradigms that have emerged, Software as a Service (SaaS) stands out as a key driver of innovation, flexibility, and efficiency for businesses of all sizes. Today, SaaS solutions are powering everything from collaboration and communication to security and observability, enabling organizations to respond swiftly to ever-changing business needs.

We, the authors of this book, have decades of computer networking experience, much of it focused on cloud and SaaS technologies. This experience highlighted the growing demand for a clear, foundational resource with practical guidance on embracing SaaS within the modern enterprise. As more organizations transition away from traditional on-premises systems, IT professionals, architects, and business leaders face new challenges in understanding SaaS architectures, migration strategies, security considerations, and operational best practices. This book aims to bridge that knowledge gap and provide a comprehensive foundation for anyone looking to succeed in the cloud era.

The book is organized into 14 chapters, each focusing on a key aspect of SaaS. Early chapters introduce foundational concepts and migration strategies, while later chapters delve into specific Cisco SaaS offerings—ranging from collaboration solutions like Webex and UCM Cloud, to security platforms such as Cisco Umbrella and XDR, to observability and management tools like AppDynamics, ThousandEyes, Meraki, and Intersight.

The primary goal of this book is to demystify SaaS and provide you with a structured, practical understanding of how SaaS solutions are designed, deployed, secured, and managed. Most importantly, this book helps you

unlock the full potential of SaaS and empowers you to lead successful cloud initiatives in your organization.

Who Should Read This Book?

This book is intended for IT professionals, network and cloud architects, security specialists, and business leaders who are looking to develop a solid understanding of SaaS in general and also its application in modern Cisco enterprises. Whether you are planning a migration, optimizing existing deployments, or evaluating new SaaS offerings, this guide will serve as a valuable resource on your journey.

How This Book Is Organized

Chapter 1, “What Is SaaS?”: This chapter explores the fundamentals of Software as a Service (SaaS) and shows how it plays an integral role in your daily technology experience. You learn about core cloud concepts, compare SaaS with other cloud models such as IaaS and PaaS, examine the shared responsibility model, and understand the key business drivers behind SaaS adoption. By the end of this chapter, you see how SaaS fits into the broader cloud landscape and why it is essential for modern organizations.

Chapter 2, “SaaS Architectures”: This chapter develops your understanding of SaaS architecture by examining its essential components and foundational principles. You learn about the logical and architectural models that guide SaaS design, including infrastructure, application services, databases, presentation, integration, security, and management layers. The chapter also explains multitenancy and compares single tenant and multitenant architectures, equipping you to evaluate, deploy, and integrate SaaS solutions effectively.

Chapter 3, “Migrating to SaaS”: This chapter guides you through the process of migrating from on-premises applications to SaaS solutions, a transition that many organizations now undertake to achieve greater scalability, flexibility, and cost predictability. You examine the four key phases of a successful migration—discovery, design and planning, implementation, and value realization—and learn best practices for each

stage. The chapter also addresses common migration challenges and strategies to mitigate them, ensuring you are equipped to execute a smooth and effective SaaS migration.

Chapter 4, “Security and Privacy for SaaS”: This chapter examines the unique security and privacy challenges of SaaS environments and provides you with practical skills to protect sensitive data and ensure regulatory compliance. You learn how to apply industry best practices, frameworks, and techniques—such as data partitioning, encryption, Zero Trust architectures, and strong identity management—to secure multitenant SaaS platforms. The chapter also covers compliance frameworks, incident response, and the use of tools like SSPM and CASB, empowering you to design and manage robust security controls for modern SaaS solutions.

Chapter 5, “Collaboration: Webex Meetings and Messaging”: This chapter details the messaging and meetings capabilities of Cisco’s Webex platform, delivered as a modern SaaS solution. You explore how Webex enables seamless collaboration through integrated messaging and virtual meetings, supporting productivity in today’s dynamic work environments. The chapter highlights the platform’s key features, flexibility, and ongoing innovation, illustrating how Webex empowers organizations to communicate and collaborate effectively from anywhere.

Chapter 6, “Collaboration: Webex Calling”: This chapter focuses on Cisco’s Webex Calling as a modern, multitenant SaaS solution for enterprise telephony. You explore how Webex Calling enables organizations to deliver scalable, feature-rich calling services from the cloud, eliminating the need for traditional on-premises infrastructure. The chapter details the core capabilities and architectural considerations of Webex Calling and highlights its seamless integration with the overall Webex collaboration platform.

Chapter 7, “Collaboration: Webex Contact Center and Webex Connect”: This chapter examines Cisco’s Webex Contact Center, a cloud-based SaaS solution that enables organizations to manage customer communications across multiple channels, including voice, email, SMS, and chat. You explore the platform’s key capabilities and SaaS architecture, learning how Webex Contact Center supports efficient, scalable, and integrated customer engagement. The chapter highlights how this modern solution empowers businesses to enhance customer experiences while

streamlining operations.

Chapter 8, “Security: Identity and Access Management”: This chapter explores the essential principles and technologies of identity and access management (IAM) in the context of modern cloud and SaaS environments. You examine how Cisco’s IAM portfolio—including Duo, Identity Services Engine (ISE), and Oort—enables secure authentication, authorization, and accounting to protect organizational resources. The chapter highlights Zero Trust concepts, advanced IAM capabilities such as single sign-on and multifactor authentication, and illustrates how these solutions provide unified, scalable security for today’s dynamic enterprises.

Chapter 9, “Security: Cisco Umbrella and Cisco AI Defense”: This chapter presents a unified security strategy for organizations navigating both widespread cloud adoption and the rise of generative AI. You explore how Cisco Umbrella delivers network-centric, cloud-based protection for users and data everywhere, and how Cisco AI Defense addresses the unique security challenges introduced by AI technologies. The chapter highlights Cisco’s approach to providing consistent visibility and control, ensuring robust, scalable defense against evolving threats in both cloud and AI environments.

Chapter 10, “Security: Cisco XDR, Splunk, and Cisco Vulnerability Management”: This chapter demonstrates how Cisco XDR, the Splunk platform, and Cisco Vulnerability Management form a unified, AI-driven security operations strategy for the modern enterprise. You learn how these solutions work together to deliver real-time threat detection and response, deep analytics, and proactive, risk-based vulnerability management across hybrid cloud environments. The chapter highlights the architectural integration and practical workflows that empower security teams to achieve greater visibility, automation, and effectiveness in combating today’s complex cyber threats.

Chapter 11, “Observability and Monitoring: Cisco AppDynamics and Splunk”: This chapter examines how Cisco’s AppDynamics and Splunk solutions empower organizations with full-stack observability across on-premises, cloud, and SaaS environments. You learn how these platforms provide real-time insights into application health, performance, and security, leveraging concepts such as metrics, events, logs, and traces (MELT) and

OpenTelemetry. The chapter highlights key observability features, deployment architectures, and SaaS security practices, demonstrating how AppDynamics and Splunk together enable comprehensive visibility and actionable intelligence across the entire digital ecosystem.

Chapter 12, “Observability and Monitoring: Cisco ThousandEyes”: This chapter explores how Cisco ThousandEyes delivers network intelligence and digital experience monitoring to close the visibility gap across on-premises, cloud, and Internet connections. You learn how ThousandEyes complements solutions like Splunk and AppDynamics by providing end-to-end observability of network performance and traffic in transit. The chapter demonstrates how integrating ThousandEyes enables organizations to achieve comprehensive monitoring of applications, platforms, and the underlying network, ensuring optimal digital experiences.

Chapter 13, “Management: Cisco Meraki”: This chapter overviews how Cisco Meraki delivers simplified, cloud-based network management through its SaaS platform. You learn how Meraki enables organizations to easily configure, monitor, and manage network devices and services across both SMB and enterprise environments. The chapter highlights Meraki’s intuitive dashboard, expanding product portfolio, and integration with Cisco’s broader management ecosystem, demonstrating how cloud-enabled management streamlines network operations and enhances scalability.

Chapter 14, “Management: Cisco Intersight”: This chapter shows how Cisco Intersight delivers scalable, cloud-based management for data center infrastructure through its SaaS platform. You learn how Intersight simplifies operations, reduces costs, and provides unified visibility across compute resources with an intuitive, API-first approach. The chapter highlights Intersight’s ability to manage large-scale environments, support robust integrations, and deliver AI-driven insights, enabling organizations to efficiently operate and scale their modern data centers.

Part I: SaaS Fundamentals

Chapter 1. What Is SaaS?

Software as a Service (SaaS) is everywhere. Stop for a moment and think about your most recent interactions with applications on your phone or computer. More than likely one or more of these interactions was with a SaaS application. Did you access your email through a web browser like Microsoft 365 or Gmail? Did you watch streaming content through a service like Netflix? Maybe you used other applications like Box, Canva, or DocuSign? All of these are examples of SaaS applications. Even if you are not familiar with the terminology of SaaS, you probably use SaaS products and solutions all the time.

Cisco defines SaaS as a delivery and licensing model in which software is accessed on the web via a subscription rather than installed on local computers. This sort of definition is common across the industry and is often expanded further to say that SaaS is a cloud-based delivery model where software is subscribed to over the Internet. What this means is that the provider of the SaaS service has responsibility for the software, its hosting, and its maintenance in exchange for the end user paying a subscription fee of some sort for access and usage.

In this chapter, before diving straight into SaaS, we will first explain foundational cloud topics. Looking at these topics is necessary to understand the place of SaaS in an increasingly cloud-centered technology landscape. Then, SaaS technology itself will gradually become the focus as you will see how it is such a compelling business model that continues to grow rapidly. We will also highlight SaaS pros and cons. More specifically, this chapter is broken down into the following sections:

- **Cloud Types:** Defines the common ways that you can deploy infrastructure for SaaS solutions, including public cloud, private cloud,

hybrid cloud, and multicloud.

- **IaaS, PaaS, and SaaS Cloud Computing Models:** Introduces SaaS and compares it to similar cloud computing models, including IaaS and PaaS.
- **Everything as a Service:** Overviews the Everything as a Service concept and how it relates to SaaS.
- **Shared Responsibility Model:** Looks at the security tasks associated with cloud computing models and how they are delineated.
- **The Business Case for SaaS:** Dives into the business drivers and why so many organizations today continue to migrate to SaaS.

SaaS is a pervasive and critical technology for just about every business today. About 70 percent of the software utilized by companies consists of SaaS applications, but this number is expected to be 85 percent in the near future. SaaS has seen impressive adoption, and it continues to grow. This chapter will help you understand the reasons for this growth as we further define SaaS and align it with cloud computing in general, starting with cloud types in the next section.

Cloud Types

If you look at cloud computing broadly, it is all about securely delivering services over the Internet. Before cloud computing, the majority of these services were handled by on-premises infrastructure. Today, however, these services are hosted in massive data centers and include servers, storage, software, and applications. The scalability of these data centers enable customers of different sizes and business needs to leverage these solutions. Additionally, cloud computing's redundancy and high availability not only ensure a higher degree of data fidelity and uptime but also make it a sensible model for businesses compared to maintaining a full on-premises data center infrastructure.

Cloud computing services can be delivered using various models or types of solutions. These cloud types affect the privacy and security of cloud data and applications along with other aspects such as accessibility and redundancy. You should think about these cloud types as high-level methods for your

business to connect to the cloud and receive cloud services.

Four primary cloud types are commonly seen. The first, the public cloud, is illustrated in [Figure 1-1](#). Just as it sounds, a public cloud is accessible to anyone. It is the most well-known cloud type and the one you are probably most familiar with. Public clouds use technologies like virtualization to share physical resources between many users simultaneously. This means that when companies and organizations use public clouds, like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud to host their applications, they are doing so on a shared infrastructure. Many customers and their applications can potentially reside on the same physical server that is then logically segmented for usage by each client.

Note

Businesses that sell cloud compute services are known as cloud service providers (CSPs). While you will find quite a few CSPs in the market, three are widely considered to be the dominant players: AWS, Microsoft Azure, and Google Cloud.

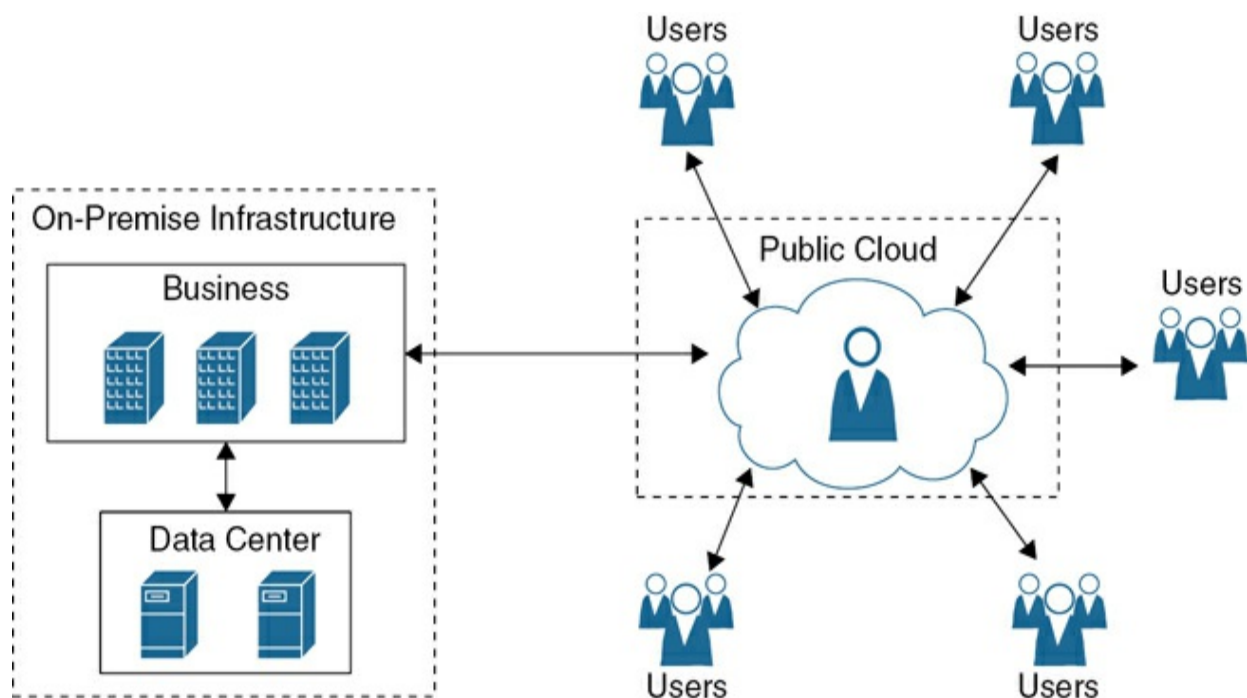


Figure 1-1 Public Cloud

Another important characteristic of most public clouds is that they are

composed of thousands of servers spread out in global data centers. The scale and amount of compute, storage, and other resources is massive and gives the CSPs that run these public clouds a massive amount of flexibility in providing users reliable access.

Figure 1-1 shows a business and its data center to represent on-premises infrastructure. This example could be a corporate headquarters or just be a small business with a local server. The figure shows a connection to a public cloud, along with other users accessing the public cloud. A sample use case for the public cloud could be the hosting of a company's website. With a public cloud, you could securely manage and update the website while your customers can easily access your website from the Internet.

Because of the shared infrastructure on such a large scale, public clouds offer a number of advantages, such as

- **Scalability:** Resources in the public cloud, like compute, memory, storage, and networking components, can be quickly added and removed as needed, often dynamically. You don't have to be concerned about running out of capacity. In this way, businesses can grow quickly and efficiently without having to install and manage a new physical server. For example, if your website hosted on the public cloud suddenly gets a surge in traffic, more resources can be added on demand to handle the surge. When the surge is over, these resources can be removed. This capability is often referred to as elasticity. Additionally, having a multitude of public cloud data centers geographically allows you to choose and deploy services in locations that are closer to end users. This availability results in end users getting better performance even at scale.
- **Cost:** Compared to building out an on-premises data center infrastructure, using cloud computing can save organizations a lot of money. The reason is that you pay only for the service you use in cloud computing. Contrast this approach with an on-premises server and software that you have to purchase, administer, and maintain, even if it is just partially used. You have a large expenditure of capital and resources along with the possibility of a deployment delay whenever you want to scale up hardware in an on-premises infrastructure. With cloud, you can have new resources and services up and running, often in minutes.

- **Reliability:** Providers of cloud computing invest heavily in infrastructure so that they can provide the latest hardware and software to ensure all the necessary patches and upgrades are applied. In addition, various levels of redundancy, disaster recovery, and automatic backup options, combined with a vast network of servers, help ensure that your applications stay up and running in a public cloud.

The second cloud type is private cloud, which replicates the public cloud experience on private infrastructure. This private infrastructure can be located on-premises, in a co-location, or deployed in a public cloud infrastructure. A co-location facility rents out dedicated floor space or an equipment rack where you may host your private cloud.

A private infrastructure hosted in the public cloud is typically referred to as a virtual private cloud (VPC) by much of the industry. One notable exception is Microsoft Azure, where it is termed a virtual network. VPCs provide you a private, logical partition in the public cloud that only you can access. Isolation of your data and application is controlled by security policies that you manage to ensure that your operations are private and not accessible by external threats or unauthorized users.

An important point about VPCs is that because they are virtual, you have the logical appearance of a private cloud, but in reality, you are almost always on shared, physical hardware. This means that your data and applications are on the same physical server as other customers but are logically separated.

For example, AWS is a public cloud, but you can define a VPC in AWS that resembles a private cloud. This VPC is a logically isolated virtual network within the AWS Cloud where users can launch AWS resources. This VPC gives you complete control over your virtual networking environment, including IP address ranges, subnets, and security configurations, making it resemble a traditional network you would operate in your own data center. At the same time, you can leverage many of the benefits of the AWS public cloud, including the elastic scaling of resources. [Figure 1-2](#) shows a private cloud using either a co-location or a VPC.

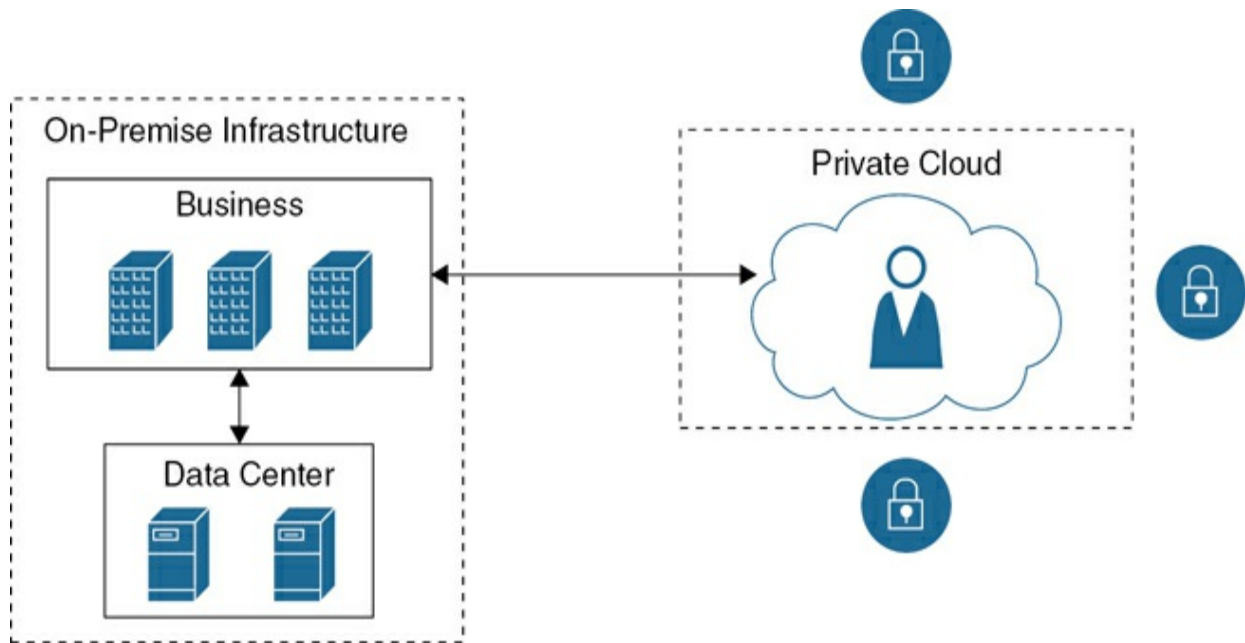


Figure 1-2 Private Cloud

Generally, private clouds are more secure than the other cloud types while offering a higher degree of ownership, customization, and control. This is especially true of the private clouds that you manage in your own infrastructure or in a co-location. At the same time, this increased control and customization come with the added cost of deploying, managing, and securing your private cloud infrastructure. Some of the benefits of private cloud include

- **Full Control and Customization:** Ownership and management of the hardware and software reside with the business, so you can choose whatever is out in the marketplace to meet your needs. Additionally, you can build a custom environment to meet your exact needs and, unless you are on a VPC, are not dependent on what a CSP supports.
- **Visibility:** You have access to all the workloads pertaining to performance, security, and access control because they are running in your environment.
- **Compliance Assurance:** Government bodies mandate certain requirements for certain types of data, like health and financial. You are not dependent on CSP offerings and can ensure that this sort of data is compliant with regulatory standards.

Examples of the type of applications and data that you might use a private cloud for include human resources (HR) data, company financials, or customer/patient data. Information and applications that may require more privacy or need additional security are usually handled by private clouds. You can think of a private cloud as just a logical, secure extension of your internal, on-premises network.

Hybrid cloud, the third cloud type, is depicted in [Figure 1-3](#). Since it is a combination of the public and private cloud types, you get the benefits of both: security and compliancy in the private cloud, plus the flexibility and scalability of a public cloud. Most importantly, you can allow data and applications to move between the two environments to efficiently handle scenarios, such as quickly scaling infrastructure to handle overflow conditions. For this reason, hybrid cloud architectures are common, especially in the enterprise space.

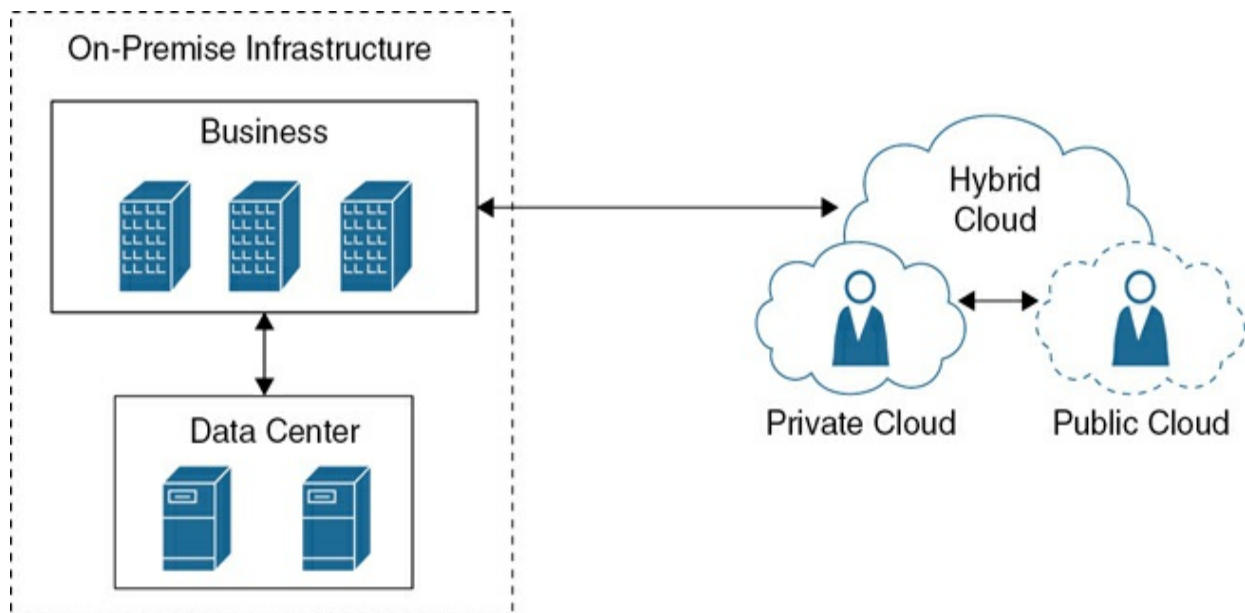


Figure 1-3 Hybrid Cloud

Hybrid cloud environments are customized for each business and their data and applications. You should think about hybrid clouds as a mix of on-premises data centers, private cloud, and public cloud that are distinct entities tied together in a way to form an integrated infrastructure. Proprietary or standardized software coordinates and manages the connections and policies between these entities to stitch together the hybrid cloud architecture.

Organizations control the movement of data and applications between on-premises, private cloud, and public cloud infrastructures based on their needs and requirements. For example, some data and applications must remain in on-premises or private cloud infrastructure for security compliancy, or sovereignty reasons. A healthcare company may be required to keep some of its patient data secured in a private cloud to be compliant with government regulations. At the same time, the company may need appointment scheduling and facility management to be flexible enough to be used or migrated between its private cloud and public cloud. Because every business is different, the management and control of data and applications in a hybrid cloud environment are unique.

The hybrid cloud, with its dual approach, naturally offers many of the same benefits that both public and private clouds provide. Specifically, hybrid cloud advantages include the following:

- **Flexibility:** With the ability to leverage both public and private clouds as well as on-premises infrastructure, you have more options for hosting your data and applications and dealing with various scenarios. One example is that your public cloud can host your web page and customer support portal while your private cloud maintains your sensitive data and applications that require low latency.
- **Scalability:** Infrastructure resources can be scaled up and down in a hybrid cloud automatically and at a low cost. A private cloud alone does not offer this benefit. Because hybrid cloud utilizes a public cloud, you are only paying for the extra compute power or storage space when needed. This public cloud might even host a VPC. For example, during the busy holiday period, a retailer may allocate public cloud resources to its online shopping website to better serve a surge in customer traffic.
- **Ease of Migration and Deployment:** Because of the interconnection hybrid cloud provides between on-premises, private cloud, and public cloud infrastructures, the capability to migrate and deploy workloads and data between these infrastructures is simplified. For example, hybrid cloud allows businesses that want to move over to the cloud in a phased approach to migrate their workloads gradually to the public cloud from their private infrastructures.

The combination of private and public infrastructure poses some challenges in addition to benefits. One of the biggest challenges is the management complexity of these two distinct types of infrastructure. Often the tools and environments are different, so connecting them together for an effective integration can be complicated. Cost can also be a factor because you now must pay for a private cloud infrastructure and a full IT staff to manage it on top of the public cloud access and usage. In addition, while cloud observability software solutions are making this situation better, visibility across private and public clouds can sometimes be challenging.

Multicloud is the last cloud type that you will often come across. As the name implies, it consists of multiple clouds, but in this case the term specifically refers to multiple *public* clouds. Typically for reliability reasons or to procure “best of breed” services offered across different providers, a company can employ a multicloud approach.

In [Figure 1-4](#) public clouds labeled CSP A, CSP B, and CSP C are connected to an on-premises infrastructure in a multicloud arrangement. Although three cloud vendors are shown in this figure, only two providers are technically needed for a multicloud. It is not uncommon to see organizations scale to many cloud vendors.

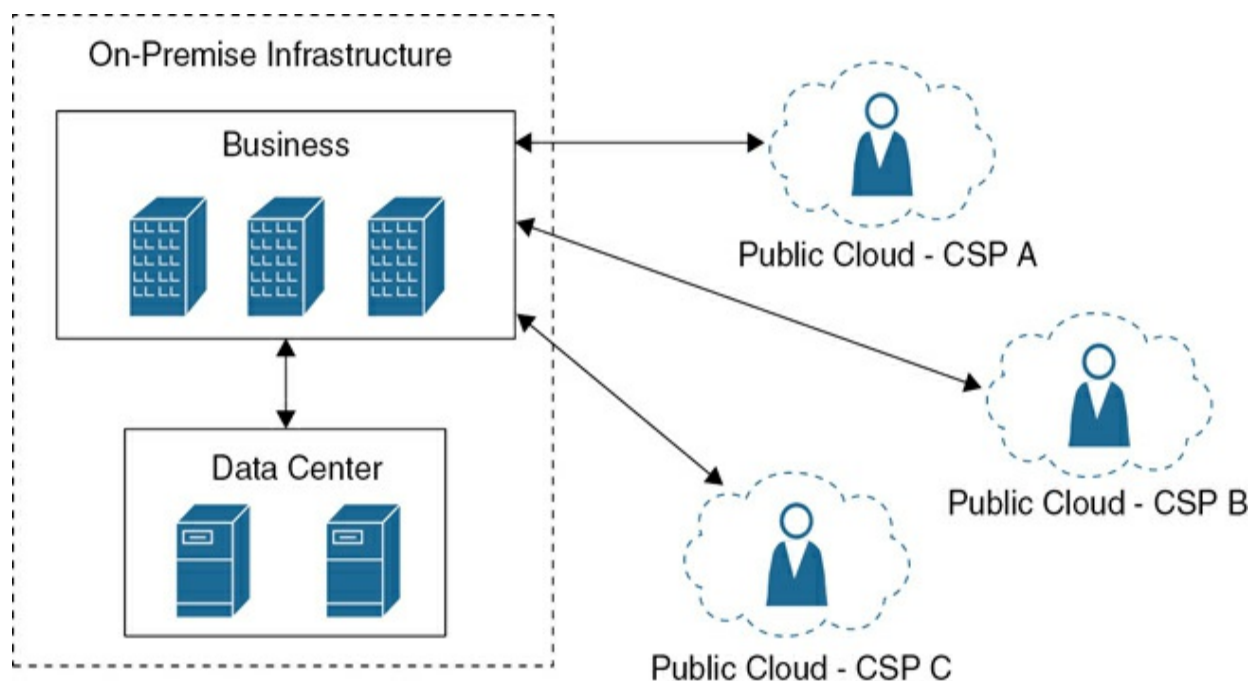


Figure 1-4 Multicloud

With multicloud, you are responsible for configuring and setting up the policies that determine when one vendor's cloud infrastructure should be used versus another. At the same time, most workloads built for a multicloud infrastructure are portable across the different cloud vendors. This portability is accomplished by building these workloads on open-source, cloud-native technologies that are supported by just about all public cloud vendors. In the enterprise, multicloud usually means that these workloads and applications are running on *Platform as a Service (PaaS)* or *Infrastructure as a Service (IaaS)* computing models hosted by different vendors. PaaS and IaaS are both covered in more detail in the next section.

Multicloud offers some unique advantages compared to the other cloud types. These include the following:

- **Optimization:** One of the primary goals of a multicloud strategy is to operate the best computing environment for your workloads. By utilizing multiple cloud vendors, you have the ability to perform this sort of optimization. For example, a cloud vendor may offer better service for a certain geographic area. Additionally, you are no longer locked into one vendor because it is easy to move workloads between vendors for reasons such as better performance or cost.
- **Access to Innovation:** Certain public cloud vendors may provide new and innovative services that can help your business or provide a competitive advantage. These best-of-breed technologies can be adopted as they emerge instead of being limited to the offerings of a single provider. For example, a vendor may specialize in artificial intelligence (AI) and machine learning (ML) and offer superior services in these areas compared to others.
- **Reliability and Resilience:** With multicloud environments providing access to multiple providers, you can easily spread critical applications and data across different vendors. This way, if one provider is not available, important business functions continue to run. The risk associated with data loss and/or downtime related to service outages can be spread out across multiple vendors.

Despite the impactful advantages that multicloud offers to many organizations, managing a multicloud environment is more complicated and

has its own set of challenges. For example, instead of managing workloads on a single provider, you now have to manage and coordinate across many cloud vendors. This same coordination and visibility are needed for security and privacy, performance, and cost comparisons. This complexity increases with the more cloud vendors that you have and can require more business spending for staff with an expertise in running these environments.

Note

Like hybrid cloud, multicloud is also seen in most enterprise businesses. In fact, you will often see these two cloud types joined together into what is referred to as a hybrid multicloud. As you can probably guess, with hybrid multicloud, two or more public clouds are combined with a private cloud environment. Hybrid multiclouds provide all the benefits of the various cloud types, so it is a compelling option. Combined with environment standardization across all infrastructures and the efficient scaling of resources, hybrid multicloud enables rapid development and deployment of workloads at scale with optimized performance and cost.

So, you might be wondering at this point, how does SaaS align with these four cloud types? The answer is that SaaS is almost always hosted on a public cloud infrastructure. Of course, this public cloud could be part of a hybrid and/or multicloud deployment model. This means that private cloud infrastructures are not a part of SaaS. It is probably easiest to think of each SaaS service that you subscribe to as a separate public cloud. Additionally, most organizations have multiple SaaS services. This is why a multicloud or hybrid multicloud deployment makes the most sense when SaaS is part of an infrastructure.

For example, a large enterprise may subscribe to multiple SaaS products, like Cisco Webex, Microsoft 365, and Cisco Umbrella. This enterprise may also have other custom applications that run on a private cloud and a public cloud, like AWS. [Figure 1-5](#) illustrates this type of hybrid multicloud deployment.

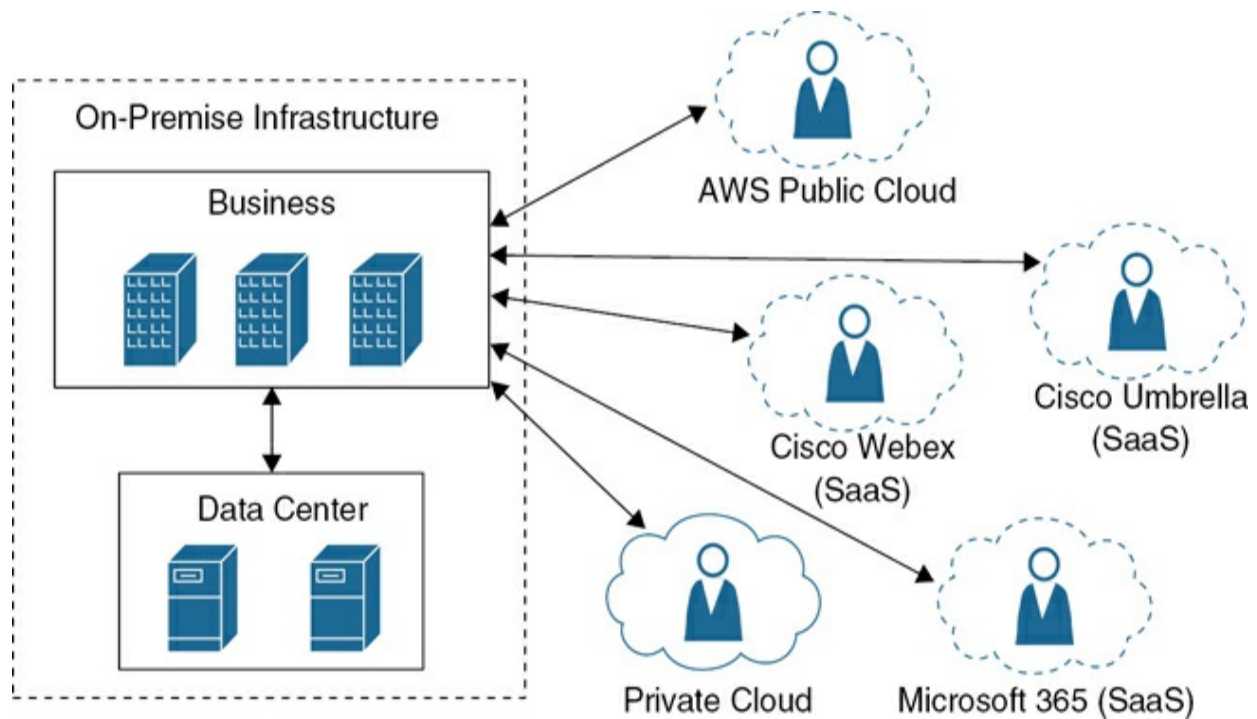


Figure 1-5 Hybrid Multicloud Deployment Example

In this section you learned about the four primary cloud types: public cloud, private cloud, hybrid cloud, and multicloud. Each offers some distinct advantages and disadvantages, which are summarized in [Table 1-1](#).

Table 1-1 Cloud Type Comparison

Cloud Type	Advantages	Disadvantages
Public Cloud	<p><i>Scaling</i> with the addition or removal of resources is efficient, and the <i>cost</i> is usually cheaper than building and managing your own infrastructure. <i>Reliability</i> and uptime are typically high.</p>	<p>While some public clouds and SaaS applications are approved for certain types of confidential data, you need to be careful. <i>Data security, privacy, and sovereignty</i> can be an issue, along with <i>regulatory compliancy</i> for some data types in public clouds. Your data and applications are no longer on infrastructure that you own.</p>
Private Cloud	<p>This type provides <i>full control and customization</i> of the infrastructure hardware and software, along with complete <i>visibility</i> to all the workloads pertaining to performance, security, and access control. You also have the ability to better ensure that <i>compliance</i> and regulatory requirements are met for certain data types, like health and financial.</p>	<p><i>Cost</i> of equipment and personnel to build and manage an on-premises or co-location private infrastructure can be high, and <i>scaling up</i> resources can be difficult. All aspects of security, upgrading, and software updates are your <i>responsibility</i>.</p>
Hybrid Cloud	<p>This type provides <i>flexibility</i> and <i>scalability</i> in leveraging both public and private clouds as well as on-premises data centers to efficiently satisfy business functions and objectives.</p> <p><i>Ease of migrating and deploying</i> workloads and data between these infrastructures is simplified.</p>	<p><i>Management</i> of both private and public infrastructure and the connectivity between them can be complex. <i>Cost</i> for maintaining dual infrastructures can be high, and visibility across them can sometimes be challenging.</p>
Multicloud	<p>This type <i>optimizes</i> the computing environment to provide one that is best for your workloads.</p> <p>It allows for the quick <i>adoption of innovative services</i> and best-of-breed technologies that can help provide a competitive advantage. It also offers the ability to spread critical applications and data across different vendors to minimize risk and increase <i>reliability and resilience</i>.</p>	<p><i>Managing and coordinating workloads, security, performance and cost</i> across multiple cloud vendors is complex. This complexity grows with the more vendors that you have.</p>

We also covered how SaaS aligns to these cloud types in this section. SaaS is almost always a public cloud offering, and you will find it most commonly in multicloud or hybrid multicloud deployments. To further define SaaS, though, we need to take a deeper dive into what makes an application or solution a SaaS one compared to other options. This topic is covered in the next section on cloud computing models.

IaaS, PaaS, and SaaS Cloud Computing Models

Software as a Service, along with Platform as a Service and Infrastructure as a Service, are three common cloud computing models. Understanding these models and comparing them to a traditional on-premises compute model is helpful in further defining SaaS, developing an understanding of it, and clarifying how it differs from other cloud service types.

At this point, you are now familiar with a private cloud and traditional on-premises compute environments with the server architecture usually located in your own data center. This makes you responsible for all aspects of maintaining the hardware and software aspects of that server. You specify the exact CPUs, memory, storage, and so on while also loading and maintaining all the software on the server. You have complete control of the environment, which allows for high customization; the cost, however, is that you are also fully responsible for keeping it operational, secure, and running efficiently. [Figure 1-6](#) shows this on-premises management responsibility juxtaposed with how management of various aspects of the environment can be offloaded to a cloud provider with IaaS, PaaS, and SaaS.

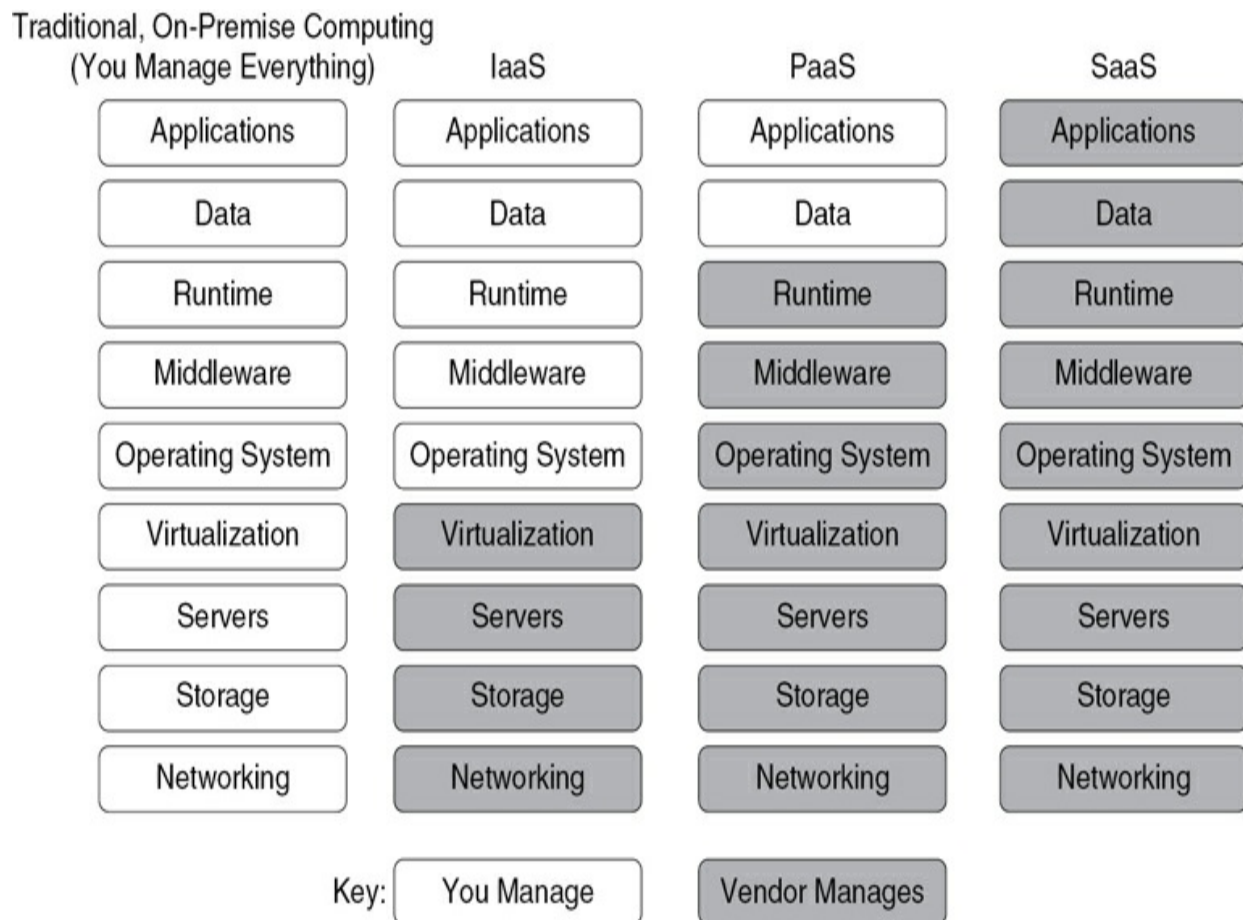


Figure 1-6 Management Comparison Between On-Premises, IaaS, PaaS, and SaaS Compute Environments

In [Figure 1-6](#), the first column on the left details the high-level elements of managing a traditional compute or server environment that is located on-premises. This could also be a private cloud. Similar columns follow for IaaS, PaaS, and SaaS. At the bottom of all the columns, you have hardware components that include networking, storage, and servers. Networking consists of the necessary routers, switches, or any other equipment for connecting the server infrastructure. Storage can include various storage devices such as dedicated storage servers that can be connected directly to servers or connected over a network. Lastly, servers are the hardware devices composed of processors and memory that execute compute functions. A deeper dive into all these hardware components is provided in the “[Infrastructure](#)” section in [Chapter 2](#), “[SaaS Architecture](#).”

Above the hardware components are the software elements found on data

center servers. Just about all servers today have a virtualization layer. This allows a single physical server to be logically segmented into multiple servers. Each of these logical servers can then be provisioned with its own operating system (OS). Microsoft Windows Server or a Linux variant are examples of common OSs that are often used on virtual servers. More details on virtualization can be found in the “[System Software and Tools](#)” section of [Chapter 2](#), where we cover SaaS architecture in much more depth.

Middleware in [Figure 1-6](#) represents software pieces that interconnect applications or components with other applications or resources. Middleware commonly handles database connectivity, authentication, and application programming interface (API) management. APIs are covered in depth in the “[Custom Integrations with APIs, Webhooks, and WebSockets](#)” section in [Chapter 2](#).

Runtime is the environment or system that your application requires for proper code execution. Applications can use many different programming languages to perform business logic and other critical functions for an organization, and each depends on a compatible runtime environment. For example, if your application is coded in Python version 3.12, you will need the Python 3.12 runtime environment on your server to properly support it.

At the top of the columns in [Figure 1-6](#) are Data and Applications. Applications are the software that a business or organization depends on. This software can be anything from an e-commerce platform to inventory management to financial software. Typically, the main reason this software needs to be on a server architecture is that multiple users need access to it at the same time. The Data block in [Figure 1-6](#) is all the information that is required by the application that must be stored and managed. This information could include user files or settings needed by the application.

For traditional, on-premises type deployments, all of the blocks in column 1 of [Figure 1-6](#) can be a lot to manage from a cost and resource perspective. Offloading some or all of this has many advantages, especially for larger companies that often have to manage hundreds or thousands of servers and their applications in this manner.

IaaS begins to take a large part of the responsibility from you and moves that to the cloud provider. For many, hardware and physical infrastructure, along

with virtualization, is one of the most difficult and costly aspects of on-premises server management. With IaaS, this on-premises infrastructure is in the cloud. You no longer have to worry about installing new infrastructure, keeping it up to date, and guaranteeing its uptime. The cloud provider now handles this infrastructure and ensures its stability, reliability, and security.

With IaaS, you still own the software pieces, including the OS and everything up the stack above it. You basically have a server in the cloud that you still need to put all your software on and continue to manage, including applying all software updates and security patches. Control of the hardware and virtualization is handled by the cloud provider, but you still maintain full control of the software stack above the Virtualization layer. In addition to the OS, this software stack includes the databases, applications, functions, and all your organization's data. Therefore, IaaS requires a high level of cloud and technical expertise to manage most deployments.

PaaS hands more management control to the cloud provider but eases even more of the management responsibility for you. In addition to the layers covered by IaaS, PaaS also turns the management of the operating system, middleware, runtime, and even databases over to the CSP. This makes PaaS a great option for when you have some custom software written in a certain language and you just want to port it to the cloud. With PaaS, you can specify all the underlying software for your application, and it is managed and maintained by the cloud provider. This way, you can concentrate only on your application running in the cloud.

SaaS is the last column in [Figure 1-6](#), and you can see how the management of the entire stack now resides with the provider. This means that businesses using SaaS only have to access the service, most often through a web browser, to utilize the software application. For many businesses, SaaS is an ideal choice because it takes away all the infrastructure burden. Unlike PaaS, where you still must manage the software and related data, SaaS customers can just focus on using the software application.

Compare this SaaS experience to the traditional software model where an application is purchased once and downloaded to your laptop or other device. You are then responsible for applying updates and bug fixes, but this task gets increasingly difficult if you are managing multiple applications in this manner for hundreds or thousands of users in an organization. [Table 1-2](#)

provides a quick summary of the pros and cons between IaaS, PaaS, and SaaS.

Table 1-2 IaaS, PaaS, and SaaS Quick Comparison

Computing Model	Pros	Cons	Examples
IaaS	<ul style="list-style-type: none">• Highest level of control and customization over cloud infrastructure• CSP maintains hardware infrastructure• Scalability is on-demand	<ul style="list-style-type: none">• Hands-on configuration and maintenance can be resource intensive• Responsible for your own security and backups	AWS, Microsoft Azure, Google Cloud
PaaS	<ul style="list-style-type: none">• Easy accessibility to a complete development platform• CSP manages development platform and infrastructure• Easy to scale	<ul style="list-style-type: none">• Less control and customization over infrastructure• Development environment may be limited in protocol and application support	AWS Elastic Beanstalk, Microsoft Azure App Service, Google App Engine,
SaaS	<ul style="list-style-type: none">• Easy to start up and use• Full management of the hardware and software by the SaaS provider	<ul style="list-style-type: none">• Least amount of customization and flexibility• No control over the infrastructure and security	Cisco Webex, Dropbox, Microsoft Office 365, and Salesforce

To provide further clarity in understanding the differences between IaaS, PaaS, and SaaS, let's look at a real-world example. Suppose you were tasked with building a website for your organization. You could build the website using on-premises infrastructure or utilizing IaaS, PaaS, or SaaS in the cloud. If you were to build your website on-premises, getting started requires more work typically than a cloud deployment.

First, you need to acquire a physical server, choose and load an OS, get it connected to the Internet securely, and so on. Then, you need to load web

server software such as Apache or Nginx, and any other applications that are needed, like a database. All this needs to be configured correctly before you even start creating and loading content to your website. Some vendors do offer a pre-built server for hosting websites, so that is another option that could make this process easier. On top of all the work required to get your website hardware and software up and running, you must also manage all the ongoing maintenance and fix any issues that may arise. You have a lot of flexibility and customization, from the hardware to the software, with an on-premises deployment, but it is resource and labor intensive to get that deployment up and running and keep it running. [Figure 1-7](#) shows an on-premises web server deployment in the first column, and then you can see it compared with the cloud options.

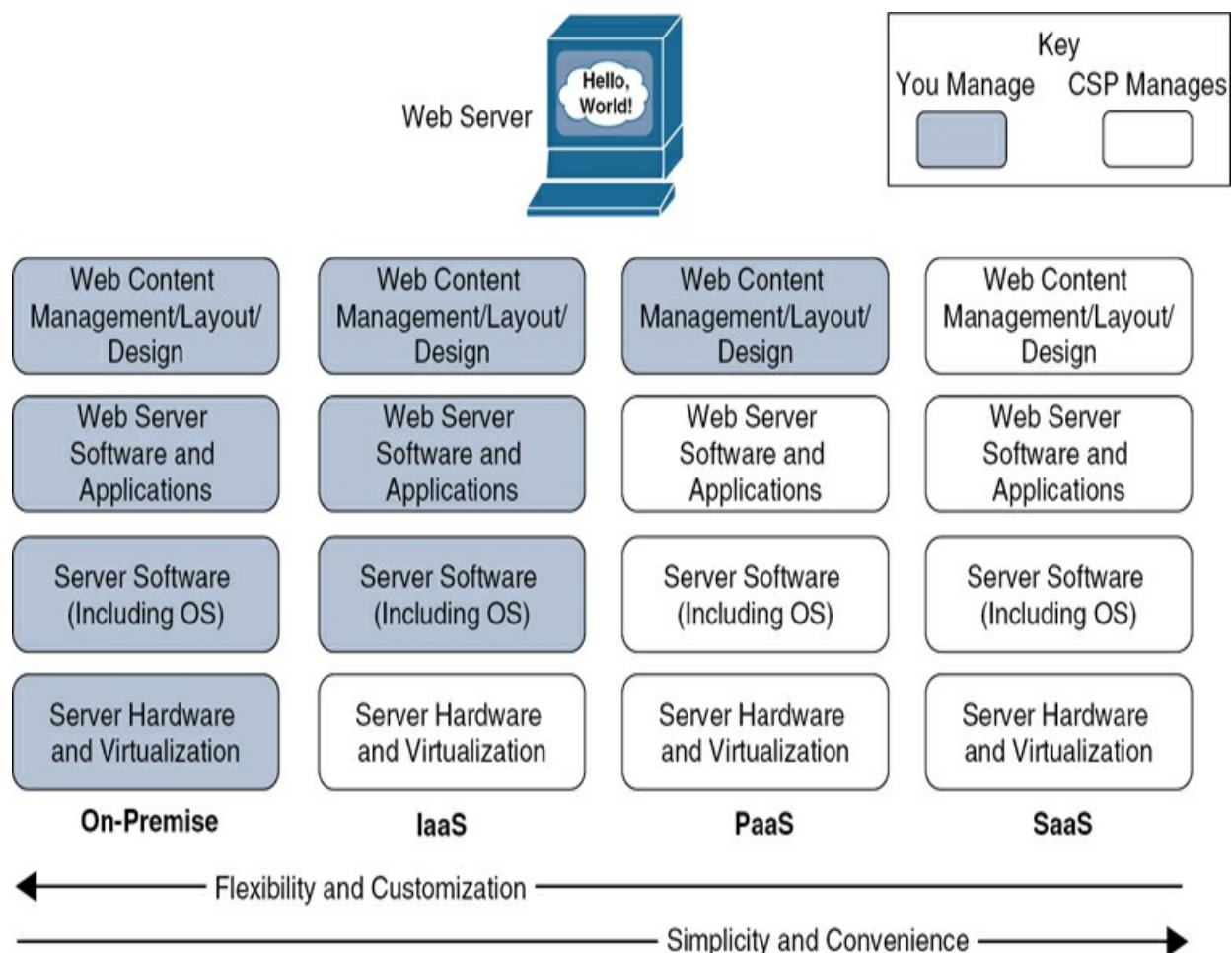


Figure 1-7 Web Server Deployment Comparison Between On-Premises, IaaS, PaaS, and SaaS

With a cloud option like IaaS, the hardware is now owned and managed by the provider. You still have control of most of the software pieces related to your web server, going all the way down to selecting the OS you want to deploy. This means that from a software flexibility and customization perspective, you still have a lot of options, just like with an on-premises deployment. Because you no longer have to worry about the hardware setup, starting on your web server becomes much quicker. You simply log in to a CSP, create a compute instance, select an OS, and then start loading the website software and supporting applications of your choosing.

PaaS builds on top of IaaS. This means that most of the core web server software is now also handled by the CSP as well. You can select a web server environment along with any runtime environments that you need. Then, you just need to load your web content and any web server applications or scripts. The important part of PaaS is that the CSP maintains the underlying web server environment software and runtimes. You do lose some additional flexibility and customization. For example, you may need support for Java for your website, but a particular CSP may not support a runtime environment for that programming language.

With a SaaS deployment, you do not have to worry about the underlying hardware or any of the software for your website. You can simply go to the website of a SaaS provider that specializes in creating websites and build everything. All you need to bring is your content. Most of the time there are many templates and designs for you to choose from, and they usually have some level of customization. Quite often you are also provided with a no-code experience and drag-and-drop placement of web page elements. While you lose deep customization and are restricted to the offerings of that SaaS provider and its roadmap, you gain ease of use and rapid creation. You could technically have a basic website up and running in minutes.

Note

Website as a Service (WaaS) is a term that is often seen when it comes to website development. It is like SaaS for websites but with more services included, such as professional website design and development, and maintenance. WaaS includes all of this in a contract that you typically pay monthly. With this model, the large price of initial website development is spread out, and the complete

day-to-day management of the website, including applying updates and ensuring security, is handled by the provider.

One aspect of this web server example that you should also understand is scalability. As mentioned previously in this chapter, scaling with on-premises infrastructure can be more challenging because often another server or storage platform must be physically installed. However, once you are in the cloud environment, the scaling of infrastructure is much easier and can happen in seconds. This is one reason why so many websites are cloud hosted. If traffic surges, then it can easily be accommodated. At the same time, the resources can be quickly scaled back down when traffic returns to normal levels.

A more lighthearted way to explain the differences between IaaS, PaaS, and SaaS is to relate them to the ways that you can have pizza for a meal. Created by Albert Barron, this popular analogy—and various forms of it that are floating around on the Internet—is quite helpful in showing your responsibility for a pizza meal compared to having a business handle various parts of the preparation and experience. [Figure 1-8](#) shows Pizza as a Service and how you can overlay it with an on-premises deployment, IaaS, PaaS, and SaaS.

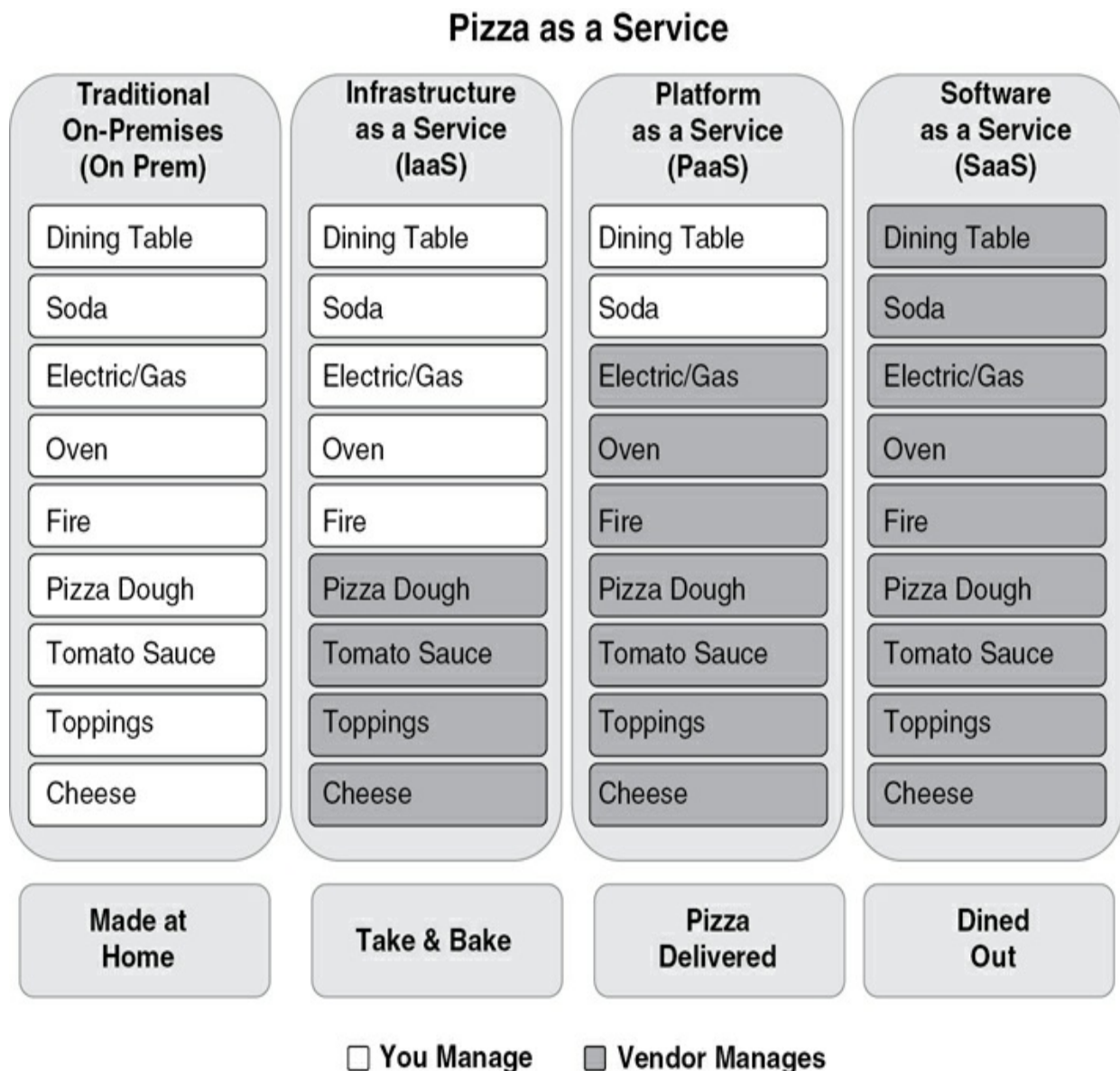


Figure 1-8 Pizza as a Service

If you think about it, this pizza model makes a lot of sense for understanding cloud service types. In the first column, you see that making pizza at home is analogous to an on-premises deployment. This means that you need to have your own kitchen infrastructure, all the ingredients for your pizza, and the supporting pieces, like soda and a table. In addition, you have to provide all the labor and pay for the utilities to cook the pizza. In exchange for all of this, you can have a customized experience. For example, maybe you want to cook your pizza in a fancy brick oven, determine the exact amount of toppings you add to the pizza, or maybe you even want to add an

unconventional topping. You have full control, but the choice comes with more up-front cost and resources, especially if you rarely eat pizza. On the other hand, if you are eating pizza every night, this cost becomes much more reasonable and probably lower than the other options.

With the Take and Bake and Pizza Delivered options, you are trading control and customization but gaining convenience. This is the same as we have discussed for the IaaS and PaaS cloud types. For example, you are limited to the toppings and options provided by the vendor, but at the same time, you no longer must make the pizza. You still have some flexibility in topping selection, when you eat the pizza, and even how it gets cooked with the Take and Bake option.

The last option with Pizza as a Service is Dining Out, which aligns to the SaaS model. Here, you do not worry about any of the infrastructure or pizza preparation. It is all managed by the restaurant, so you are provided a full-service experience. You utilize the kitchen, dining table, ingredients, and labor of the pizza provider, and the provider also handles the cleanup and so on. The trade-off is that you give up control and customization for simplicity, convenience, and up-front costs. A good way to look at SaaS is that it is like selecting your favorite pizza restaurant. You focus on the outcome or service that you need. You then choose the provider that best meets your needs, and that provider takes care of everything else so you can enjoy the end product.

Everything as a Service

If you take a moment and browse around the cloud computing world, you might feel like just about anything and everything is being tagged with “as a service.” You would be right, and the term for this is *XaaS*, which stands for *Everything* or *Anything as a Service*. Typically delivered via cloud computing, XaaS can be defined as a flexible and consumption-based solution for providing tools, products, or technologies that can quickly scale up and down.

The major well-known XaaS models are IaaS, PaaS, and SaaS, which were covered in the previous sections. However, quite a few more exist, and new ones come along regularly. For your general awareness, [Table 1-3](#) provides a quick reference to some of the most common XaaS models. XaaS types that

are bolded are discussed in more detail in other sections of this book. Please see the corresponding reference in the Definition column in [Table 1-3](#) to jump to a more detailed coverage of that XaaS model.

Table 1-3 XaaS Quick Reference

XaaS Type	Definition
AlaaS (AI as a Service)	With the high expense and often limited availability of AI hardware, AlaaS allows you to utilize cloud-based AI resources in a pay-as-you-go model.
BMaaS (Bare Metal as a Service)	A subset of IaaS, BMaaS is a model where you are the sole tenant on a dedicated physical server in the cloud. This service provides you more control and security for your workloads versus the multitenant environment of IaaS. Also known as MaaS or Metal as a Service.
CaaS (Containers as a Service)	An extension of IaaS, CaaS is an offering where the provider manages all the hardware and software infrastructure to allow for the rapid development and deployment of applications using containers.
CCaaS (Contact Center as a Service)	CCaaS is a cloud-based solution that can handle customer communications and route customer inquiries through the proper channels. Also known as hosted contact center, this SaaS-related model can replace an on-premises call center. See Chapter 7, “Collaboration: Webex Contact Center and Webex Connect,” for more details on CCaaS.
CPaaS (Communication Platform as a Service)	CPaaS provides a cloud-based layer for embedding communication capabilities, such as voice, SMS, digital messaging, and video into customer-facing and partner applications using APIs and software developer kits (SDKs). An extension of SaaS, CPaaS is covered in Chapter 7.
DaaS (Data as a Service)	A subset of SaaS, DaaS focuses on making data always available in the cloud no matter the customer’s infrastructure or location.
DBaaS (Database as a Service)	The DBaaS model, which is a subset of SaaS, lets you utilize cloud-based database software on a subscription basis. Databases and how they can tie into DBaaS are covered in the “Relational and Non-Relational Database Types” section in Chapter 2.

FaaS (Function as a Service)	FaaS is a subset of PaaS, and the provider manages the application. You are able to simply load functions, such as snippets of code or scripts, to be executed in the cloud. FaaS is discussed in more depth in the “Microservices and Serverless Architectures” section in Chapter 2.
HaaS (Hardware as a Service)	This model provides cloud-connected hardware, such as an IP phone or video conferencing unit, on a subscription basis. Also known as DaaS (Device as a Service).
IaaS (Infrastructure as a Service)	Covered in detail in the previous section, IaaS provides cloud-based, pay-as-you-go virtualized resources, including compute, storage, and networking services.
IoTaaS (IoT as a Service)	IoTaaS allows for the management, monitoring, and analysis of data from IoT devices in a centralized cloud environment.
NaaS (Network as a Service)	NaaS is a model that enables users to easily operate cloud-based networking infrastructure, like firewalls, load balancers, and hardware-centric VPNs without owning, building, or maintaining it themselves.
PaaS (Platform as a Service)	PaaS is an offering where the provider manages a complete development environment in the cloud for you, and you just need to load in your code or application. A more detailed overview of PaaS was covered in the previous section.
RaaS (Ransomware as a Service)	Even cyber criminals have offerings in the XaaS space. RaaS is a model that enables the criminals that develop ransomware to sell or lease it to other criminals for ransomware attacks.
SaaS (Software as a Service)	The SaaS model involves delivering applications over the Internet, usually through a web browser. You do not have to worry about downloading or maintaining the software locally. Of all the XaaS offerings, SaaS has the biggest market share and is the most well known. The SaaS model was introduced in the previous section but is covered throughout this book.
UCaaS (Unified Communications as a Service)	UCaaS provides a centralized, cloud-based platform for an organization’s communications, including telephone, video conferencing, messaging, and various applications to help improve collaboration. You can find more information on UCaaS in Chapter 6, “Collaboration: Webex Calling.”

Many of the XaaS service types in [Table 1-3](#) are subsets or extensions of SaaS. This means that you will often see a solution simultaneously being referred to as SaaS and another XaaS term. For example, in the Collaboration space, Cisco has several SaaS products. This means that you may see the Cisco Webex Contact Center product called both a SaaS solution and a CCaaS solution. Another example is Cisco's Webex Calling product, which is both SaaS and UCaaS. This way of naming products can be confusing, but just remember that SaaS is a broad term and other XaaS terms are typically more specific to the type of SaaS solution or SaaS technology area. To minimize this sort of confusion, refer to this table as needed to help you better understand the various “as a service” types as you come across them in this book and other places.

Shared Responsibility Model

One of the fundamental aspects of SaaS is that the management of the software and all its underlying and dependent elements is the responsibility of the SaaS provider. This level of management responsibility by the SaaS provider is much greater than other cloud computing models, like PaaS or IaaS, as discussed earlier in the chapter.

When you look at cloud computing models from a security perspective, similar levels of delineation must be made to distinguish between provider and customer responsibility. Based on the cloud computing model, certain security tasks belong to the provider, and others are the customer's. This delineation of security tasks is defined in what is known as the shared responsibility model (SRM). [Figure 1-9](#) illustrates how the CSP takes on more of the security tasks as the computing model transitions from IaaS to PaaS to SaaS.

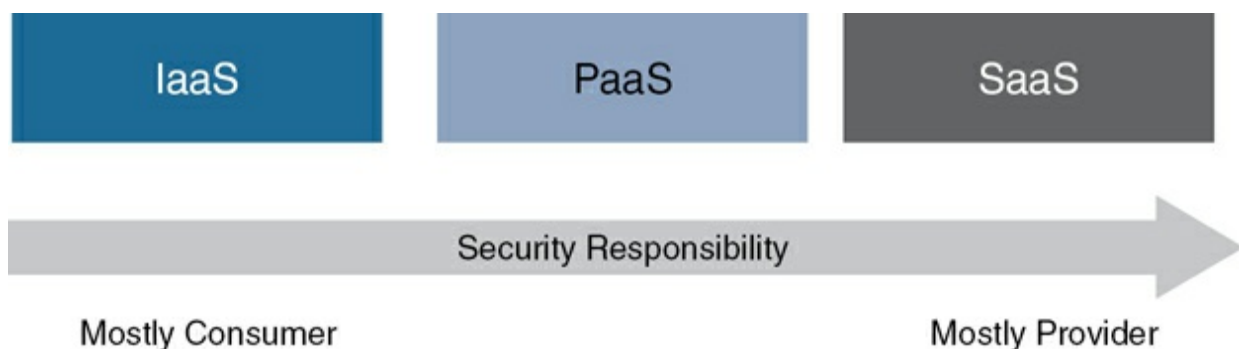


Figure 1-9 Security Responsibility Changes Between IaaS, PaaS, and SaaS

Most of the shared responsibility models that you find are pretty similar and align to [Figure 1-9](#). Many CSPs publish an SRM as well to help explain and clarify the expectation between themselves as the provider and you as the consumer. For example, [Figure 1-10](#) shows the SRM for Google Cloud.

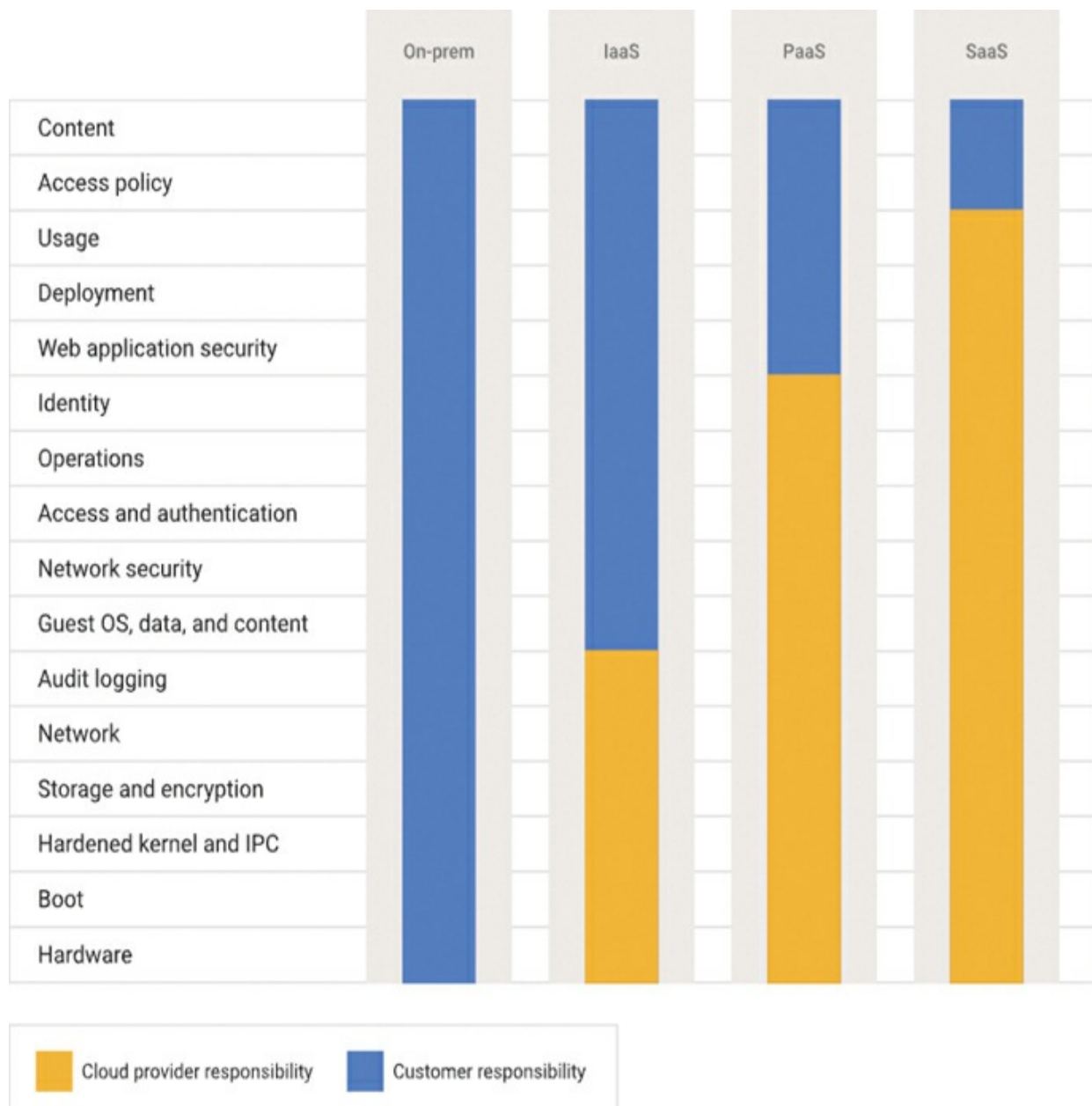


Figure 1-10 Shared Responsibility Model (Google Cloud)

In [Figure 1-10](#), naturally the customer is responsible for all security tasks in

the on-premises scenario in column one. As you move to the other columns on the right, more of the security responsibility shifts to the provider, Google Cloud in this case. By the time you get to the SaaS column on the far right, just a couple of tasks rest with the customer. It is important to note that even though these SaaS security tasks might seem small in comparison to the provider tasks, they are critical. With SaaS, the customer is responsible for the content and data that is utilized by the SaaS solution and is responsible for managing who has access.

Note

Other major CSPs, like AWS and Microsoft, have SRMs for their customers to reference. The SRM for AWS can be found at <https://aws.amazon.com/compliance/shared-responsibility-model/>.

You can view Microsoft Azure's SRM at <https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility>.

The main takeaway when it comes to SRMs is that you as the customer always have an important responsibility when it comes to security. An SRM seeks to further define the security tasks and whether they should be handled by the customer, the provider, or in some cases, both parties. To explore other security topics regarding SaaS, refer to [Chapter 4, “Security and Privacy for SaaS.”](#)

The Business Case for SaaS

Up to this point, we've covered SaaS from a cloud computing perspective and defined it mainly through its relation to cloud types and computing models. In this section, we will look at SaaS from more of a business perspective. Specifically, what are the compelling drivers and use cases for so many organizations embracing and considering SaaS solutions?

SaaS is already one of the most popular software delivery models in the world, and SaaS growth is only expected to continue. According to Statista, “The global Software as a Service market size is projected to grow from \$273.55 billion in 2023 to \$908.21 billion by 2030, at a compounded annual growth rate (CAGR) of 18.7%.” The impressive success of SaaS to date and

its continued growth can be attributed to a number of factors and the fact that there are benefits to both the SaaS customer and the provider. [Table 1-4](#) highlights the drivers behind SaaS's popularity and growth.

Table 1-4 Business Drivers for SaaS

SaaS Driver	Definition
Cost Savings	Up-front costs savings can be significant because you no longer need the infrastructure and resources, like in-house technical staff, to host and manage applications on-premises at scale. Additionally, on-going costs savings add up quickly with the software development, maintenance, and support offloaded to the SaaS provider for a monthly subscription.
Reliability	The SaaS provider is responsible for maintaining the software and ensuring its continuous availability using its own cloud infrastructure. Service-level agreements (SLAs) can define the service continuity that can be expected.
Flexibility and Scalability	As a SaaS customer, you have the flexibility to use more or less of a SaaS service, and you are typically only charged for what you use. Additionally, you can usually add or remove features and capabilities quickly or often discontinue your service at any point if the SaaS solution is not working. From a scalability perspective, the SaaS provider can automatically scale up and down resources with your usage levels.
Fast Deployment and Adoption	In many cases, you can sign for a SaaS product and be up and running in minutes. More complicated integrations with a lot of users can take longer but still be deployed much faster than setting up an application on-premises. Additionally, users are typically quick to adopt and start using the SaaS product because it is web based or can be accessed through an app.
Always Up to Date	SaaS providers keep your software on the latest release, so you have all the important patches and new features automatically. All your software upgrades are simply included as part of the subscription price.

You will notice that we touched on many of the factors outlined in [Table 1-4](#) earlier in this chapter because they are by-products of public cloud or the SaaS computing model in general. Additionally, some SaaS drivers are

simply inherent to cloud architecture. However, when looking at the SaaS factors in [Table 1-4](#) as an aggregate, you should see the distinct advantages and benefits that move businesses to this model.

Note

There is often some confusion around SaaS and the solutions offered by a managed service provider (MSP). While they may seem similar at first glance, they are different. With a SaaS product, you typically work directly with the SaaS vendor to obtain access to its software in the cloud via a monthly subscription fee. That monthly subscription fee includes the licensing and access to the software. With an MSP, a software license is purchased for an organization, and then the MSP is hired to run the software, including its maintenance and upgrades, and maybe even hosting it as well. The MSP services typically cost more, but the close support and customization services of the MSP might be worthwhile, especially for critical software that needs to be customized or tightly integrated with other applications and systems.

A couple more advantages related to the factors of reliability and flexibility are illustrated in [Figure 1-11](#). SaaS vendors or providers almost always build out their data centers across diverse geographic areas. This arrangement leads directly to not only improved reliability or redundancy but also performance. When you're a SaaS customer, connecting to a data center close to your location minimizes network impairments and almost always provides a better experience.

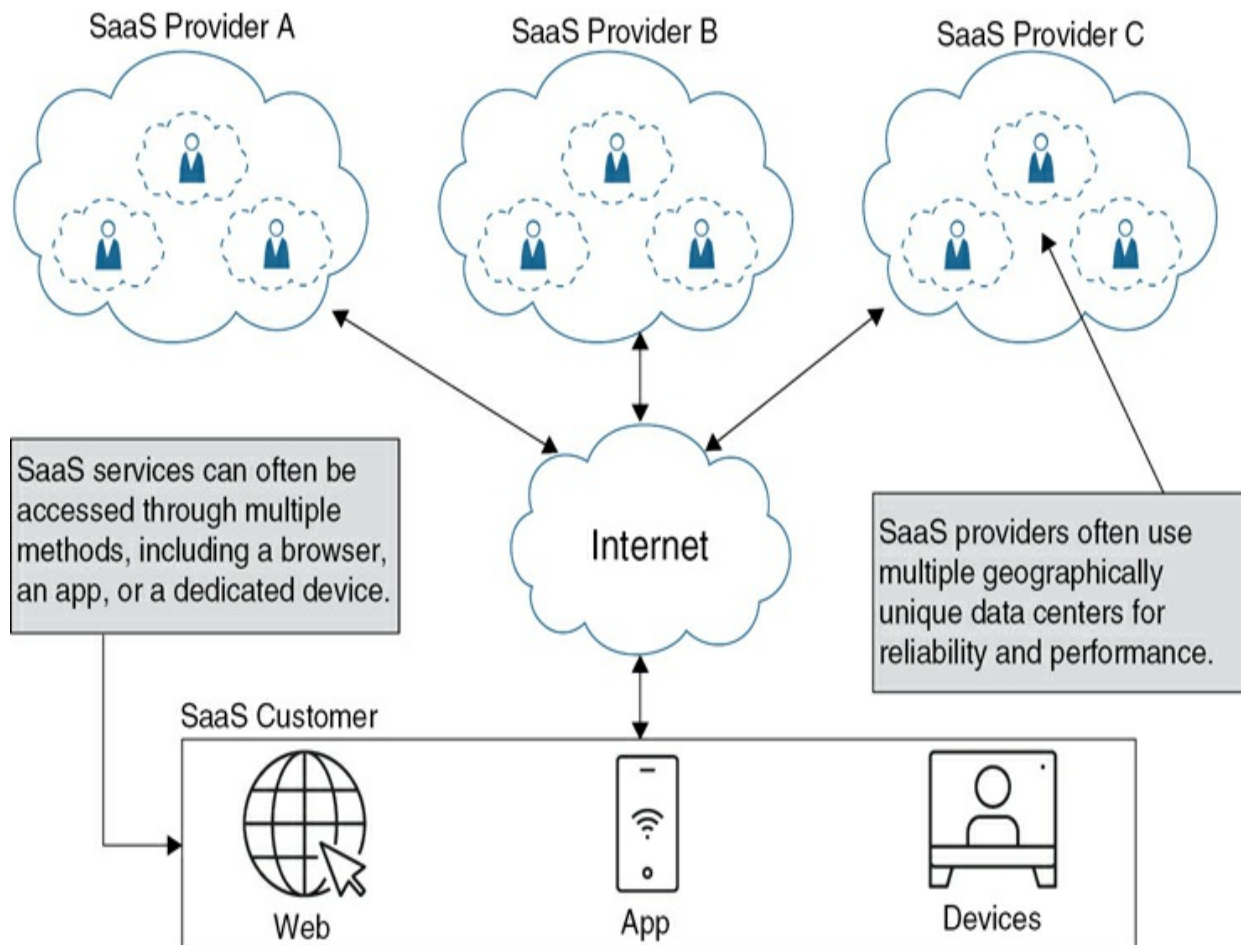


Figure 1-11 SaaS Benefits

[Figure 1-11](#) also highlights some additional flexibility for the SaaS customer in terms of connectivity. While a SaaS consumer typically uses a web browser to connect to a provider, other methods are possible as well. One method is an app, just like what you download from a phone's app store. Although this method does seem to somewhat move away from traditional SaaS because you are now downloading software, often a better user experience is available through the app.

Using dedicated hardware devices for connecting to a SaaS solution is not as common. The best example is probably in the UCaaS space where phones and video devices can be cloud connected to solutions based on a SaaS model. Because they are cloud-connected, these devices are managed and controlled from the cloud, and software updates and features can be pushed down to the devices. Between dedicated hardware devices, web browsers, and downloadable apps, SaaS customers have options for accessing their

service.

It is important to also realize that a SaaS model can also benefit the provider or vendor. For example, costs are cheaper, just like for a SaaS consumer. A SaaS provider leverages multitenancy to use the same hardware and software infrastructure for many customers. Multitenancy is an architecture where multiple tenants or customers utilize a single software instance. For more details on multitenancy, refer to the “[Multitenancy](#)” section in [Chapter 2](#).

Additionally, software upgrades and patches are pushed to customers, so fewer releases must be supported. New features and revenue add-ons can be pushed out quickly to retain customers and attract new ones in often competitive environments. Last of all, the SaaS subscription model provides the vendor a predictable, recurring revenue stream.

At the same time, despite all the benefits and advantages associated with using SaaS solutions, it is not for every business and situation. Challenges exist that must not be overlooked, and in some cases, a non-SaaS solution must be utilized instead. A few of the main barriers or concerns organizations may encounter to using SaaS products include the following:

- **Security:** As a SaaS customer, your data is in the cloud in a multitenant environment that is potentially accessible by anyone on the Internet. While there are ways to mitigate attacks associated with this access, for certain types of data and certain businesses, this is an uncomfortable scenario and poses too much risk.
- **Regulatory and Data Residency Requirements:** Data storage and processing rules vary between countries. Because SaaS solutions often use international data centers, it can be confusing and difficult to ensure that laws and regulations are not broken when using some SaaS solutions with certain types of data.
- **Integration:** Connecting SaaS solutions with other existing applications and systems used by your business is not always straightforward. You have to use the integration tools, processes, and APIs offered by your SaaS provider. You do not have the integration customization options that you do with other cloud computing models like PaaS or IaaS.

Therefore, the business case for SaaS is usually quite compelling, but you

will need to do some work in advance to ensure smooth migration and usage of a SaaS product. Refer to [Chapter 3, “Migrating to SaaS,”](#) for an in-depth look at migrating to SaaS, including best practices and pitfalls to look out for.

Summary

This chapter introduced SaaS and highlighted its main characteristics as a software model. These characteristics include

- **Cloud-based software** means that you do not have to download software to use it. All you need in most cases is a web browser to access the latest version with all the patches and new features installed.
- **Subscription pricing** gives you access to use the software for a fee, usually monthly or yearly. This recurring revenue model is predictable and easy for the provider and the consumer.
- **Accessibility via an Internet connection** allows for access by any customer no matter their location.
- **Scalable resources** in the cloud enables the SaaS service to expand and contract based in usage and demand. This consumption model allows you to pay for just what you use.

In the first two sections, we covered cloud types and cloud computing models so that you can see how SaaS aligned to cloud computing in general. In the cloud types section, we provided an overview of the various cloud types, like public cloud, private cloud, multicloud, hybrid cloud, and hybrid multicloud. SaaS is hosted in public clouds, and enterprises usually access SaaS through hybrid multicloud deployments.

In the cloud computing models section, we compared SaaS to IaaS and PaaS. IaaS allows you the most control and customization, but you own a large part of the software stack in the cloud. PaaS provides a complete development environment with more of the software stack owned by the provider. With SaaS, all of the infrastructure is owned by the provider, and you can simply access the software using a web browser in most cases.

Next, we covered XaaS. This section provided an overview and a handy reference table to some of the common “as-a-service” offerings. Many of the

XaaS offerings are subsets or extensions of SaaS, so you should be prepared for certain solutions being SaaS and another XaaS term.

In the last two sections, we discussed the shared responsibility model and the business case for SaaS. In the shared responsibility model section, you learned how securing SaaS solutions requires both the provider and the consumer to have ownership of certain tasks. Both parties are responsible, and we shared a model from Google Cloud as an example.

In the business case for SaaS section, we looked deeper into the drivers and benefits for moving to SaaS, such as cost savings, reliability, flexibility and scalability, and fast deployment and adoption. SaaS challenges, such as security, regulatory and data residency requirements, and integration, were also highlighted.

Now that you have a good understanding of what SaaS is from a cloud computing and business context, you are able to take on more in-depth topics around SaaS architecture, migration, and security. These topics are covered in the following chapters, which, together with this chapter, make up the first part of this book. Then, in the second part of this book, you will take this foundational knowledge and apply it to Cisco SaaS solutions and how they work.

References

- Redline SaaS Industry Statistics: <https://redline.digital/saas-industry-statistics/>
- Cisco Definition of SaaS, including benefits and challenges: <https://www.cisco.com/c/en/us/products/software/what-is-software-as-a-service-saas.html>
- Creation of IaaS, PaaS, and SaaS figure: <https://dachou.github.io/2018/09/28/cloud-service-models.html>
- Albert Barron—credit for Pizza as a Service analogy: <https://www.linkedin.com/pulse/20140730172610-9679881-pizza-as-a-service/>
- Cisco—What is multicloud?:

<https://www.cisco.com/c/en/us/solutions/cloud/what-is-multicloud.html>

- Splunk—what is multicloud?:
https://www.splunk.com/en_us/blog/learn/multicloud.html
- Cisco definitions of CCaaS, CPaaS, and UCaaS and their differences:
<https://www.webex.com/what-is-ccaas.html>
- Borad coverage of XaaS options:
<https://www.auvik.com/franklyit/blog/aas-as-a-service-list/>
- Microsoft SRM: <https://learn.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility>
- Google's Cloud Architecture Center—Shared responsibilities and shared fate on Google Cloud:
<https://cloud.google.com/architecture/framework/security/shared-responsibility-shared-fate>
- Statista: <https://www.statista.com/statistics/505243/worldwide-software-as-a-service-revenue>

Chapter 2. SaaS Architecture

Imagine that you wake up one morning and decide you want to build a house. You are familiar with houses and may even live in one. So, you know what makes up a house from a high level. For starters, you know you need a floor, walls, and a roof. You will also need some functional systems like plumbing, electrical, and heating and cooling. You come up with a basic supply list.

With this supply list, you go to your local construction supply store and buy the materials. You get concrete for the foundation, wood for walls and floors, and shingles for a roof. Additionally, you get nails and screws to connect it together, wiring and fixtures for the electrical system, piping and faucets for plumbing, and ducting for heating and cooling. A few days later all this material gets delivered and unloaded in your driveway. Where do you begin? How do you assemble all these raw materials into a house? Now you realize something very important is missing!

The missing piece is the architecture. As any builder will tell you, even the simplest construction projects require a plan or architecture to be successful. You begin with architectural plans as represented in [Figure 2-1](#). Architecture in Software as a Service (SaaS) and the cloud is just as important as it is when building a home.

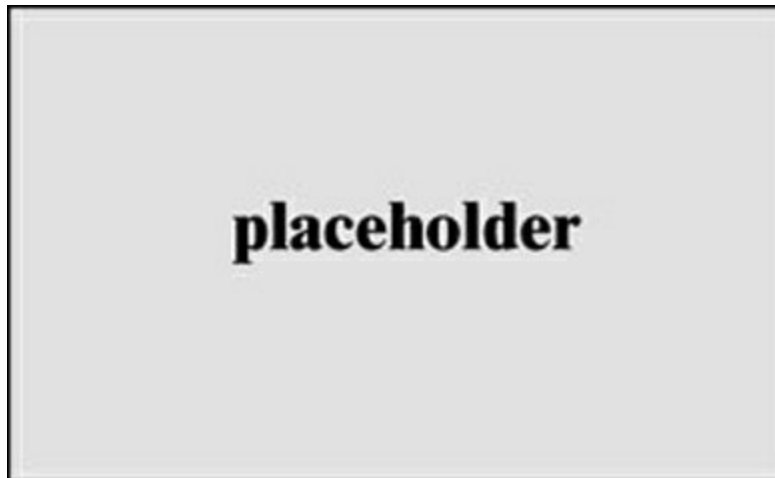


Figure 2-1 Architectural Plans for Building a House

Familiarity with cloud architecture is the key to understanding SaaS from a design standpoint. SaaS architecture serves as the blueprint detailing the layout and components that power a SaaS application. A solid grasp of cloud architecture principles forms the foundation for evaluating, deploying, migrating, and integrating SaaS solutions within your environment. If you already have experience with cloud architecture, much of this chapter will reinforce your existing knowledge, since SaaS is fundamentally a software model delivered through the cloud. For those new to cloud or SaaS architecture, this chapter is essential for building a strong foundation—one that subsequent chapters will expand upon, applying these concepts and terminology to real-world Cisco SaaS solutions in [Part II](#) of this book.

In this chapter, we'll begin with simple architectural constructs and then layer on more advanced concepts in the later sections. Specifically, we'll cover the following topics:

- **Logical Model:** This section introduces SaaS design by building off the SDN Logical Model utilizing logical planes.
- **Architectural Model:** This section presents the SaaS Architectural Model that serves as the core reference for building and understanding SaaS applications and solutions. This model is composed of the following blocks:
 - **Infrastructure** refers to the physical hardware and software resources for compute, storage, and networking that underpins any SaaS

application, along with operating system virtualization technologies.

- **Application Services** forms the engine that powers any SaaS application and includes a microservices or serverless architecture with back-end programming containing the business logic and core functions.
- **Database Services** handles the storage of structured, semi-structured, and unstructured data using a variety of relational and non-relational databases.
- **Presentation Services** includes various access methods, like a web page or app, and front-end technologies to deliver an interface between users and the application.
- **Integration Services** provides the capability to connect with other cloud or onsite applications and services using custom or prebuilt methods.
- **Security and Privacy** implements tools and methods, like cloud security controls; identity access and management; and visibility, monitoring, and logging to secure the SaaS application ensure the privacy of user data.
- **Management and Analytics** allows for efficient infrastructure and configuration management and the ability to monitor and observe the application and its underlying services and resources.
- **Multitenancy:** This section compares the single tenant and multitenant architectures and how they are applied to SaaS applications.

Logical Model

When you're learning about SaaS from an architectural perspective, it is often easier and more approachable to connect SaaS with other networking technologies and frameworks. Software-defined networking (or SDN) has some architectural parallels with SaaS and can be a good starting point. SDN is an architecture designed to make a network more flexible and easier to manage by centralizing management and abstracting the control elements from the data forwarding function in network devices.

To make this abstraction of control elements from the data forwarding function in SDN easier to understand, a logical model is often used to simplify this concept. In the next couple of sections, we will look at the SDN Logical Model and then apply it to SaaS as a way of introducing some of the basic underpinnings of SaaS architecture.

Review of SDN Logical Model

Before the advent of SDN, traditional routers and switches combined the control logic and routing or forwarding functions into a single device. For a router, the control logic would include the routing protocols and the routing table they populate. In a network environment, this control logic is then replicated and individually configured on each router. As you might imagine, the manageability and agility associated with this model can often be much more challenging when scaling out to networks with large numbers of devices.

In essence, you could logically separate traditional router and switch functionality into a control piece and a packet or frame handling piece. These pieces have been termed *planes* and are conceptual groupings where certain processes or functions occur. Specifically, the control part is naturally referred to as the *control plane* and the packet or frame handling functions are called the *data plane*. [Figure 2-2](#) provides a high-level overview of a router logically divided into its control and data planes.

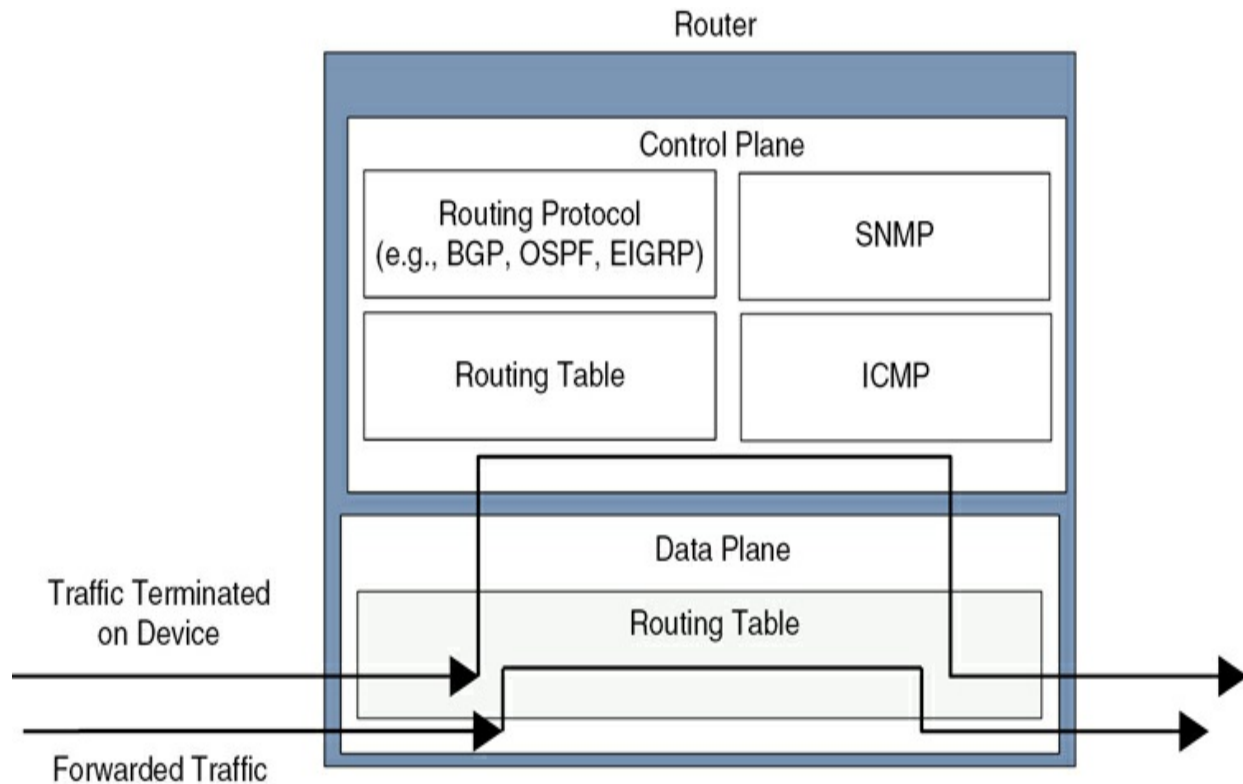


Figure 2-2 Control Plane and Data Plane for a Network Device

In [Figure 2-2](#), you can see that routed traffic that is meant to be forwarded does not leave the data plane. The forwarding function is completely handled in the data plane itself. This approach is optimal from an efficiency and performance perspective because today's network devices handle it mainly in hardware using specialized chips, such as an application-specific integrated circuit (ASIC). Instead of being designed for general-purpose use, an ASIC is a chip that is crafted and specialized for a particular task, like forwarding packets or frames.

Packets or traffic destined for the router itself is passed from the data plane to the control plane. Routing protocol updates, Internet Control Messaging Protocol (ICMP), and Simple Network Management Protocol (SNMP) packets are examples of traffic that is terminated on the router in the control plane.

Routing protocols are a good example of control plane functionality. From keepalive messages to routing updates, routing protocols such as BGP, OSPF, or EIGRP use specific messages to communicate with neighboring routers and determine the optimal paths for traffic. It is important to note that these

messages are unicast or multicast between devices. Therefore, because this message is contained in a packet addressed to the router, it is handled by the control plane. In some cases, a routing protocol message can change the routing table, which in turn changes the forwarding information in the data plane. However, the process and decision-making for changing the forwarding instructions at the data plane level happen in the control plane. To summarize, the control plane creates and maintains the operational logic that populates a routing table and the data plane executes that logic.

Note

In this section, a router is utilized as the example to explain the control plane and data plane concepts, but this information applies to switches as well. Switch forwarding or MAC address tables in the data plane serve inbound and outbound frames. Like the router example, switch control plane logic uses specific protocols like Spanning Tree Protocol (STP) to populate the switch forwarding table.

With SDN, this traditional paradigm of each network device containing both data plane and control plane functions changes. SDN separates the control logic for networking devices, like routers and switches, from the physical devices themselves. So, instead of having each router and switch maintain individually configured policies for forwarding, this intelligence is abstracted to a centralized SDN controller, as illustrated in [Figure 2-3](#). This SDN controller maintains a holistic view of network policies and simply configures each router and switch with the appropriate policies for each.

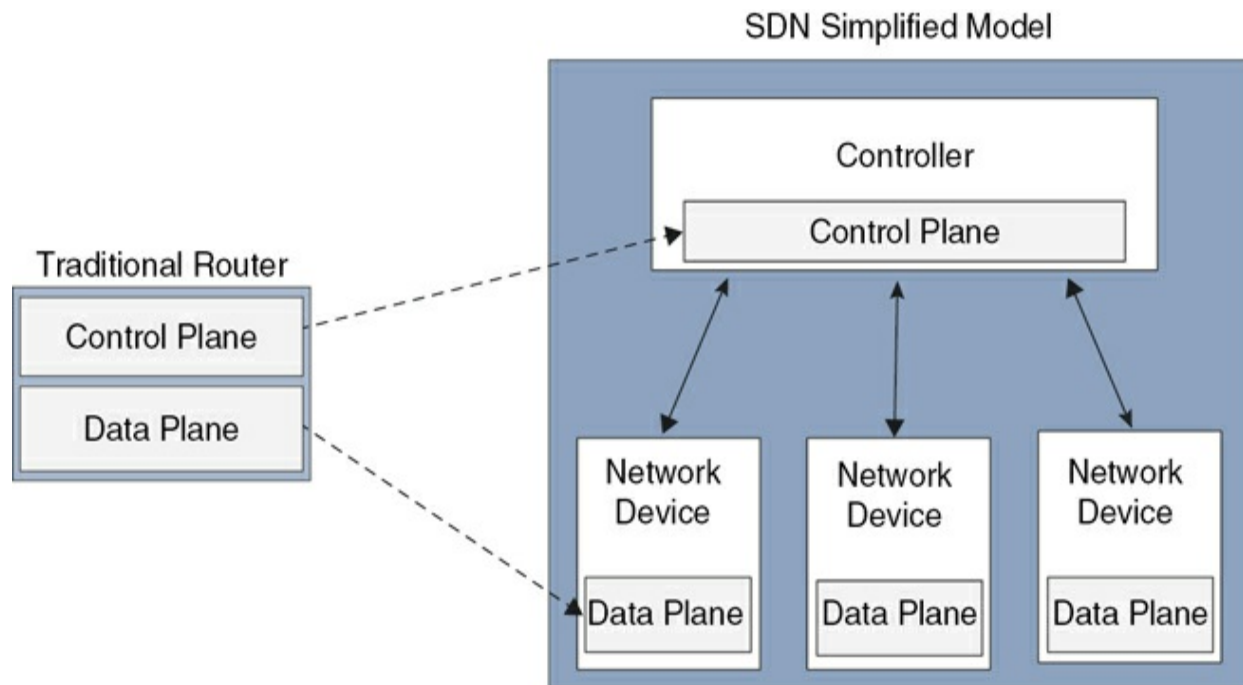


Figure 2-3 Control Plane and Data Plane Separation in SDN

Diving further into the SDN Logical Model, you will discover another plane. This is known as the application plane, and it sits above the control plane. It contains the applications and services that define the behavior and outcomes that need to be achieved by the network. These applications could be security or performance related and have the ability to signal the control plane to adapt and reconfigure the network as necessary.

[Figure 2-4](#) provides an overview of the SDN Logical Model, showing all three planes and their relationship to each other. As previously discussed, the data plane resides at the bottom of this model and consists of network devices that are dedicated to forwarding traffic. The control plane function of these network devices is no longer co-located with the data plane. Instead, the control plane functionality is handled by a centralized device known as the SDN controller.

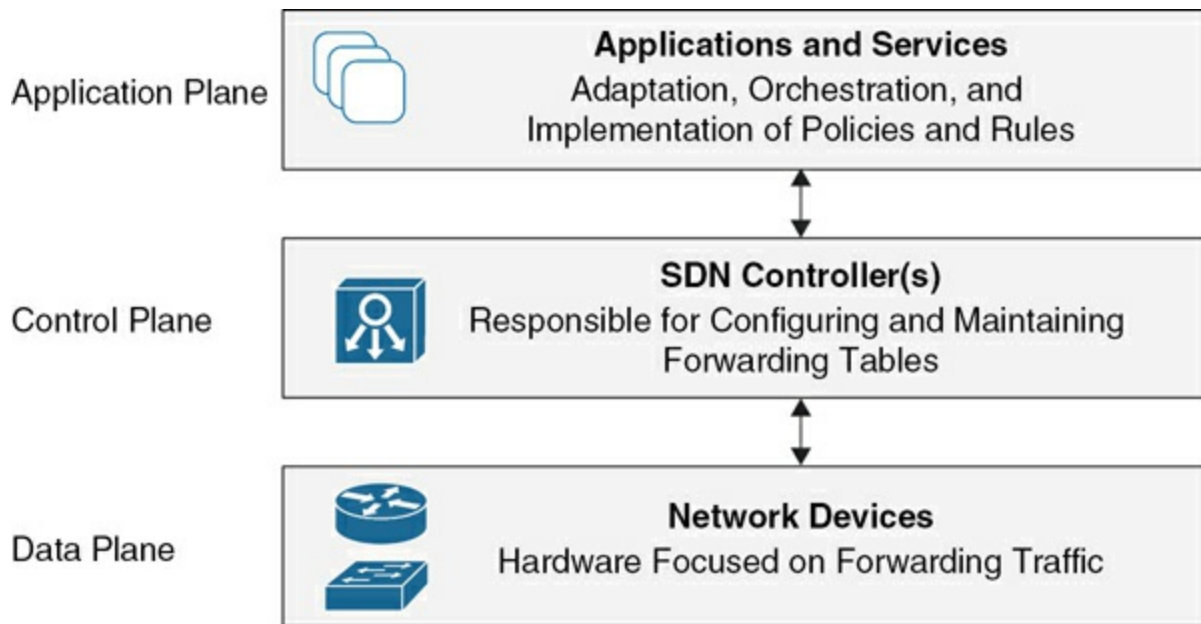


Figure 2-4 SDN Logical Model

The SDN controller is the device that contains all control plane functionality and uses application programming interface (API) connections to communicate with the data plane and application plane. We'll discuss APIs in more detail in the “[Custom Integrations with APIs, Webhooks, and WebSockets](#)” section later in the chapter. For larger numbers of network devices, you may see more than one SDN controller or even a hierarchy of controllers to allow for more flexibility and stability. Other functions of the SDN controller include implementing policies from the application plane while relaying information from the network devices.

To further clarify this SDN model, let's consider the following example. You have a cybersecurity application for your network that resides in the application plane. It detects a threat and determines that all traffic between network subnets 10.1.1.0 and 10.2.2.0 should be blocked except for Secure Shell (SSH) traffic. The cybersecurity application sends this instruction to the SDN controller via API for policy enforcement. The SDN controller translates this instruction into the proper configuration in the control plane. You should note that the cybersecurity application does not need to have any knowledge of network device configuration. The SDN controller has the intelligence to understand application instructions and the best way to implement them.

Once configured in the control plane, the SDN controller messages the network devices impacted by this change via the API to update their forwarding tables with this new information. Now all traffic, except for SSH, between subnets 10.1.1.0 and 10.2.20 is blocked, and this change happened almost instantly. With a traditional router and switch, this example would require control plane configurations being made at each individual device that was involved with this type of traffic.

This example shows the advantages of SDN in modern networks, but from a SaaS architectural discussion, the takeaway should be the relationship and function of the logical planes of this SDN Logical Model. In the next section, we will expand this SDN Logical Model once more to introduce SaaS and how it can be viewed using this architectural construct.

SaaS Control Plane and Application Plane

When applying the SDN Logical Model to SaaS from a foundational level, you may notice that many sources can have a high level of complication with a focus on designing and building your own SaaS application. This outcome is not the intent of this chapter; instead, our goal is to provide a working knowledge of SaaS architecture with an understanding of the core tenants. For this reason, in this section we will leverage an industry logical model for SaaS from Amazon Web Services (AWS). The AWS Logical Model for SaaS provides the most approachable method for understanding the SaaS control plane and application plane.

[Figure 2-5](#) illustrates the SDN Logical Model as it applies to SaaS, focusing only on the application plane and control plane. The data plane, discussed earlier, is not shown because it is not relevant to SaaS, which operates at a higher level and does not involve low-level traffic forwarding. This discussion highlights how SaaS relies primarily on the application and control planes for its functionality.

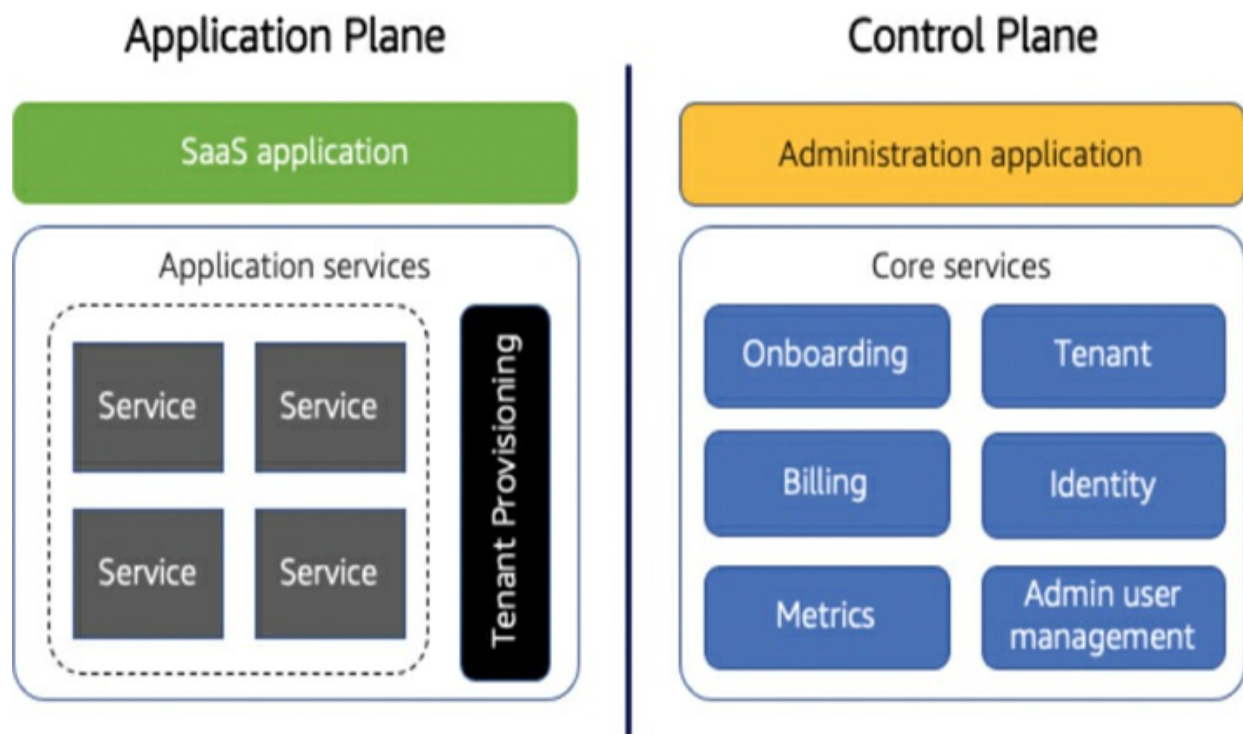


Figure 2-5 SaaS Logical Model for Application Plane and Control Plane

Just like in the SDN model in the previous section, the application plane in SaaS does indeed contain applications and services, but in SaaS, these functions are usually multitenant, virtualized, and built on microservices. We will discuss multitenancy, a core part of most SaaS applications, in detail in a later section, but essentially, *multitenant* means that multiple tenants, or users, have access to the same application or service. This model contrasts with a single tenant where just one user exists for that service or application. Because SaaS is almost always cloud-based, applications and services also tend to be heavy users of virtualized environments where multiple instances of a service can run on the same physical hardware. Microservices are simply the small building blocks of an application that increases agility and flexibility for the architecture. We will also discuss microservices later in this chapter.

Notice in [Figure 2-5](#) that the application plane is divided into two boxes. One is labeled “SaaS application,” and the other is titled “[Application services](#).” The SaaS application is what you, as a user, interact with. It is the piece of the architecture that provides the function and service that the user requires.

The Application Services box in the application plane in [Figure 2-5](#) holds

those functions that support the SaaS application itself. The specific application services utilized vary depending on the SaaS application needs and can range from simple scripts to more advanced integrations. More and more of these services are usually deployed as microservices.

Let's consider the example of Cisco Webex Meetings to better understand the relationship between the SaaS application and application services. The SaaS application portion of Cisco Webex Meetings is the place where you create and attend virtual meetings using various devices. This SaaS application runs in the cloud and is responsible for delivering the SaaS functions that a customer expects from the service. The application services for Cisco Webex Meetings are more obscured from the user and could include metrics and security functions for ensuring the efficient running of the Webex Meetings application. Other common application services include ordering and payments. Note that Webex Meetings is covered in detail in [Chapter 5](#), “[Collaboration: Webex Meetings and Messaging](#).”

The right half of [Figure 2-5](#) shows the control plane and some of its high-level functions. Like the application plane, it is also divided into two sections: “Administration application” and “Core services.” The administration application is accessible by the user for a wide range of tasks related to managing the SaaS application. Going back to the Webex Meetings example, Control Hub would be the administration application. Webex Control Hub is a portal that offers you a holistic view and management of all your Webex services, users, and devices.

The core services of the SaaS control plane provide the capabilities and functions necessary to support and execute the requirements related to the administration application. While the administration application is what the user interfaces with directly, you can think of the core services as the execution functions for the user choices and requests made by the user in the application.

Continuing with the Webex Meetings example, an administrator in Control Hub may upgrade a current trial or free subscription to a paid one. A Webex Meetings core service for this upgrade function would then be notified to take the necessary actions for this subscription change. Another common core service is User Management. In the case of Webex Meetings, the User Management core service executes the requests made in Control Hub for

functions such as the addition or deletion of user accounts for the SaaS solution.

While it is not shown in [Figure 2-5](#), communications between the application plane and the control plane are critical. For example, a subscription change in the control plane that enables additional features must be communicated to the application plane so they can be made available to the user. Most often these communications occur via APIs between the various services and applications. For the sake of simplicity, we have not explicitly illustrated them because they are implementation specific and can vary based on the SaaS solution.

The SDN Logical Model, as applied to SaaS, provides an entry-level introduction to SaaS architecture and breaks down some of the key concepts. Specifically, the concepts of a control plane and application plane illustrate logical divisions for where SaaS services and applications reside. With this understanding and background, you are now prepared to learn about the SaaS Architectural Model in the next section.

Architectural Model

To dive deeper into SaaS architecture, you will need to go beyond the logical model discussed in the previous sections and home in on the SaaS building blocks themselves. As with any architecture, breaking down the system into distinct groupings allows you to better segment these functions and understand the relationships between them. In this section, we will present the SaaS Architectural Model, which is the most pragmatic way to learn and understand SaaS architecture.

The National Institute of Standards and Technology (NIST) Special Publication (SP) 500-292 presents a cloud computing reference architecture. This architecture serves as a standards-based entry point for the foundational SaaS Architectural Model that we will explain in the proceeding sections. This NIST specification is broad and comprehensive in its coverage of the stakeholders and its applicability to cloud architectures that also include Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). After a high-level discussion of this model to provide the proper awareness, we will narrow the focus to a SaaS and cloud provider perspective termed the *SaaS*

Architectural Model. This is the primary model you will see referenced throughout the rest of this book.

Note

You will find a few different cloud computing architectures that can be applied to SaaS. Most are similar and use layers or components to categorize functions. The SaaS Architectural Model explained later in this section is also comparable but utilizes a building block concept and pillars that align to the NIST 500-292 standard to better illustrate SaaS architectural concepts for easier understanding.

NIST Cloud Computing Reference Architecture

In Special Publication 500-292, the NIST details a cloud computing reference architecture. This architecture is vendor neutral and seeks to provide a common lens for evaluating cloud services and solutions by government organizations. However, it also is widely applicable to the broader industry and serves as an introductory standard for learning about cloud architecture in general. [Figure 2-6](#) represents a simplified cloud architecture model based on the NIST Conceptual Reference Model.

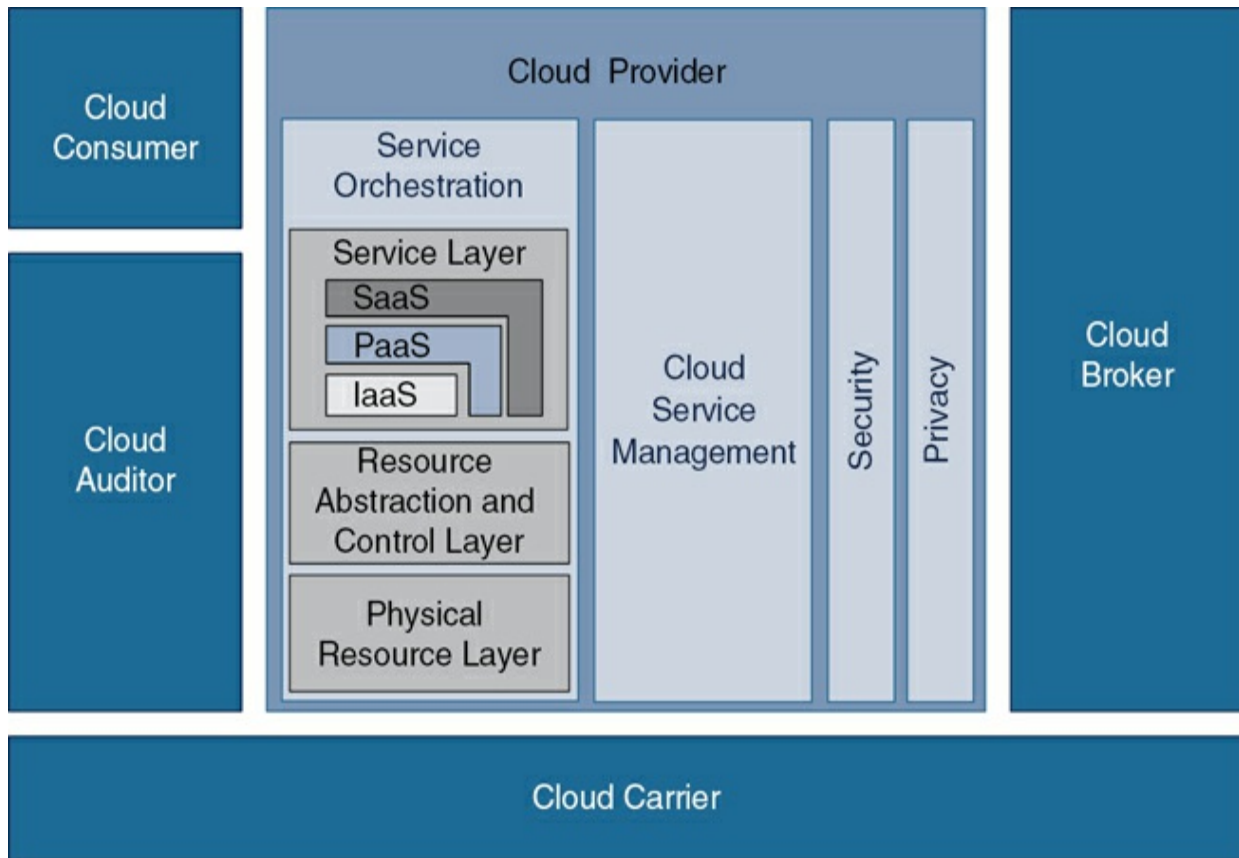


Figure 2-6 High-Level Conceptual Reference Model Based on NIST SP 500-292

In [Figure 2-6](#), NIST organizes the model around the concept of stakeholders, or *actors*. The five actors in this model are the cloud consumer, cloud auditor, cloud provider, cloud carrier, and cloud broker. These actors have different roles and responsibilities that help with the segmentation of cloud functions. [Table 2-1](#) summarizes the functions of the actors shown in [Figure 2-6](#).

Table 2-1 Actors in the NIST Conceptual Reference Model

Actor	Description
Cloud Consumer	Requestor and end user of cloud services from a cloud provider: The cloud consumer selects the services they need, negotiates a service-level agreement (SLA) for the services, and is responsible for paying for the services. A cloud broker may optionally serve as a middleman. Cloud consumers can be an individual or an organization.
Cloud Provider	The supplier of various cloud services to cloud consumers: For SaaS, the cloud provider deploys, maintains, updates, and ensures the availability and features/functions of that software as ordered by the cloud consumer. This responsibility extends to the control and administration of the underlying infrastructure for the SaaS application.
Cloud Carrier	The infrastructure provider for the connection between the cloud consumer and cloud provider: Serving as an intermediary, cloud carriers deliver the network and telecommunications services and products that ensure reliable access to cloud applications and solutions.
Cloud Broker	An entity that manages and provides cloud services to cloud consumers on the behalf of one or more cloud providers: Acting as an intermediary, aggregator, or in an arbitrage capacity, a cloud broker provides cloud consumers simplified management and access to services from cloud providers.
Cloud Auditor	An independent party that provides an opinion on the services provided by a cloud provider: Focused predominantly on security and privacy, a cloud auditor inspects and monitors the controls, policies, and performance of a cloud provider and provides feedback and guidance.

An example is the best way to understand the relationship between the NIST actors described in [Table 2-1](#). Imagine you want to procure a cloud calling SaaS service, like Cisco Webex Calling, for your organization. Webex Calling is a complete enterprise-grade cloud calling and team collaboration solution offered through a flexible subscription model. This SaaS technology will be covered in [Chapter 6](#), “[Collaboration: Webex Calling](#).” In this example, you would be the cloud consumer. As the cloud consumer, you would have a list of requirements for this cloud service. Your requirements

may include local gateway public-switched telephone network (PSTN) connectivity, voice mail, a virtual receptionist, and so on.

With your list of requirements, you then look for a cloud provider to deliver the cloud calling service. In this example, cloud providers are more than likely SaaS businesses that are owners of the cloud calling application. They run their application using various cloud service providers (CSPs) to provide availability and stability. Various cloud providers will probably offer what you need, so you will need to compare services and functionality and most likely consider cost, performance, SLAs, and other factors. We'll discuss CSPs in more detail in the “[Infrastructure](#)” section of this chapter.

Alternatively, you could also order your cloud calling service through a cloud broker. A cloud broker offers a cloud calling service, but it may be a rebranded or private label application from a cloud provider. The cloud broker is a partner to the cloud provider and may layer its own branding along with additional services and even support onto the cloud calling service before offering it to cloud consumers.

Cloud carriers in this scenario are service providers (SPs) you contract with to connect you to the cloud provider or cloud broker that provides the cloud calling service. You negotiate SLAs for performance and availability with the cloud carrier. Optionally, you can purchase other connectivity services, such as a secure and dedicated connection.

Cloud auditors are not typically seen outside of government-related consumption of a cloud service. Remember that this NIST reference model covers usage by government organizations that must engage and look at cloud services from a different perspective than private industry. However, if you were purchasing this cloud calling service for a government agency, then a cloud auditor most likely would be involved. Cloud calling involves voice communications, and a cloud auditor ensures that these communications are properly secured by the cloud provider and that personal information (PI) and personally identifiable information (PII) of government employees are also protected appropriately.

Note

Enterprise SaaS application providers can take on multiple NIST

SP 500-292 actor functions, including cloud provider, cloud broker, and even cloud consumer when dealing with hosting CSPs. This is all abstracted for end users but occasionally comes to the forefront when an outage at a major CSP impacts well-known SaaS applications.

The comprehensiveness and broad coverage of NIST SP 500-292 provide a good entry point into cloud architecture in general. However, to take a deeper look into SaaS specifically, a deeper focus on the cloud provider is necessary. The other actors in the NIST model are critical pieces, but the cloud provider holds the details that are most interesting from a SaaS perspective. Therefore, in the next section, we will narrow the discussion to the NIST cloud provider and how this can be extended and developed into the SaaS Architectural Model.

SaaS Architectural Model

One of the primary actors in NIST SP 500-292 is the cloud provider. The cloud provider holds the architectural components that are used in creating the SaaS Architectural Model. If you recall from [Figure 2-6](#), the cloud provider is composed of the following blocks: Security, Privacy, Service Orchestration, and Cloud Service Management.

The Cloud Provider Security and Privacy blocks in the NIST model refer to important and well-known concepts in cloud architecture and the IT world overall. Security for a cloud provider includes functions and services like authorization, authentication, identity management, monitoring, policy management, and auditing. The Privacy block covers the protection of PI and PII data in the cloud.

The NIST Service Orchestration block covers functions that pertain to the core operation and infrastructure of the cloud application. This coverage includes the physical hardware itself, like servers for compute and storage, along with any resource abstraction layer or software that manages and controls them, like virtual machines (VMs) and hypervisors. Additionally, the interfaces for consumers of a SaaS service are defined and managed in this block.

Last of all, the NIST Cloud Service Management block encompasses all the functions that a cloud provider utilizes for the management, control, and operation of the services being provided to the cloud consumer. Accounting, billing, and provisioning of the cloud service are examples of the functions that you will find in this block.

Now that the high-level cloud provider functional blocks have been defined, they can be extended and evolved to be more SaaS specific. As mentioned previously, the NIST model looks at cloud architecture holistically, covering not only SaaS but also PaaS and IaaS. [Figure 2-7](#) illustrates how the NIST cloud provider functional blocks are tuned to be more SaaS specific.

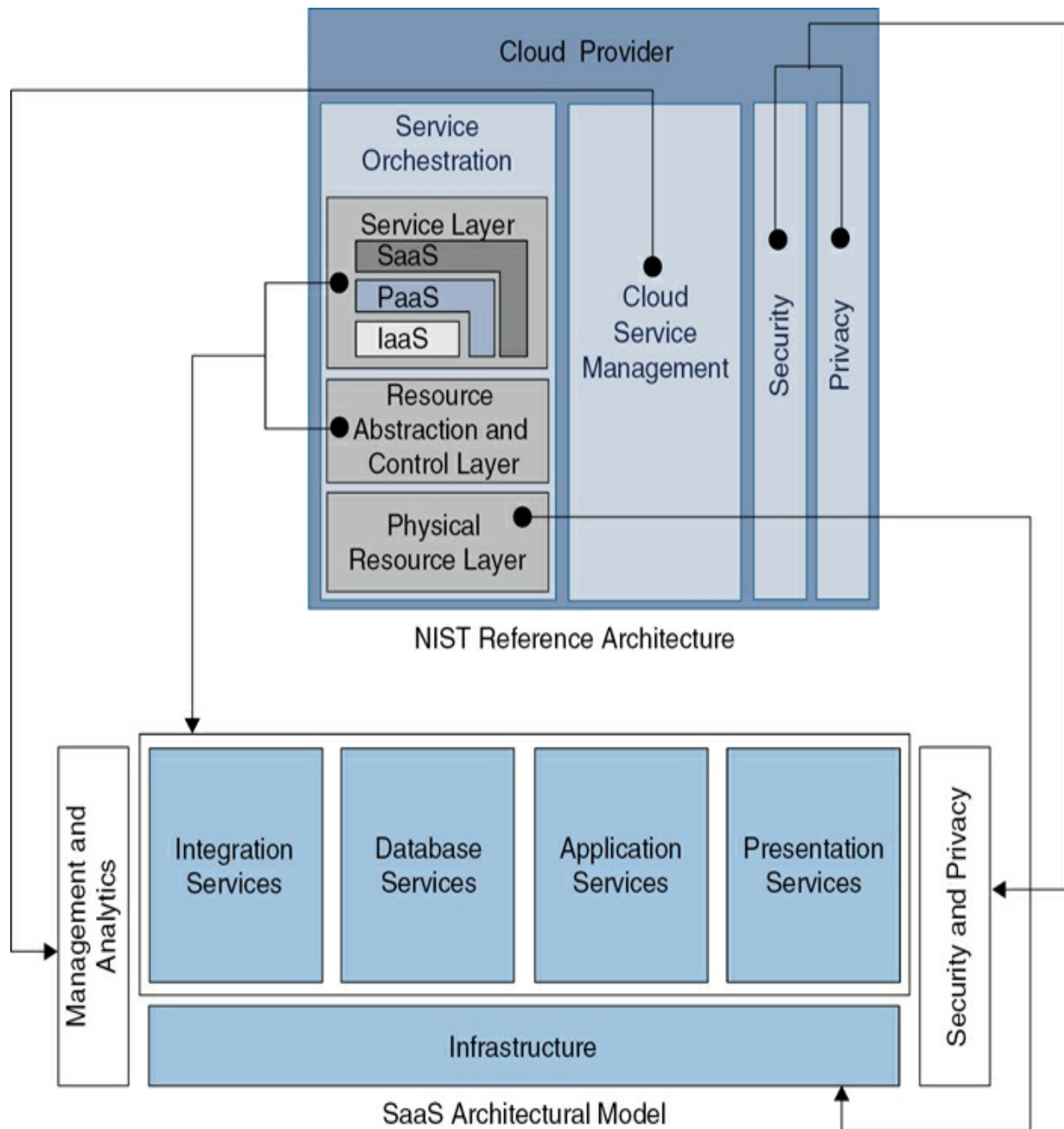


Figure 2-7 Extracting and Redefining SaaS Elements from NIST Cloud Provider Architectural Components

In [Figure 2-7](#), you see the NIST cloud provider functional blocks that were defined in the preceding paragraphs. From these blocks you can further extract the SaaS-specific elements into SaaS functional blocks. Some of these SaaS blocks have new names, like Application Services or Integration Services, while others continue to use a similar naming as the NIST

functional block, like Security and Privacy. Note that the new names are not new cloud architectural elements but simply a regrouping of cloud functions to make them better suited for discussing SaaS architecture.

For example, NIST Cloud Service Management, which handles functions like monitoring and reporting along with provisioning and configuration, map to the Management and Analytics column in the SaaS Architectural Model. Management and Analytics, like Security and Privacy, are applicable to all parts of the model, and this is reflected in both architectures. However, you should find the SaaS Architectural Model more approachable due to its narrower focus and simplified structure.

[Figure 2-8](#) provides a clearer view of the SaaS Architectural Model. As mentioned, the blocks in this model evolved from the NIST Cloud Computing Reference Architecture, as illustrated in [Figure 2-6](#). If you have some familiarity with cloud architecture in general, many of the elements will appear familiar. SaaS architecture is a subset of cloud architecture after all. The main advantage of this model is that it keeps the key architectural principles easy to understand as you develop a base knowledge of the main SaaS services and components. At the same time, as you dive deeper into cloud architecture, you can further expand and build on this model.

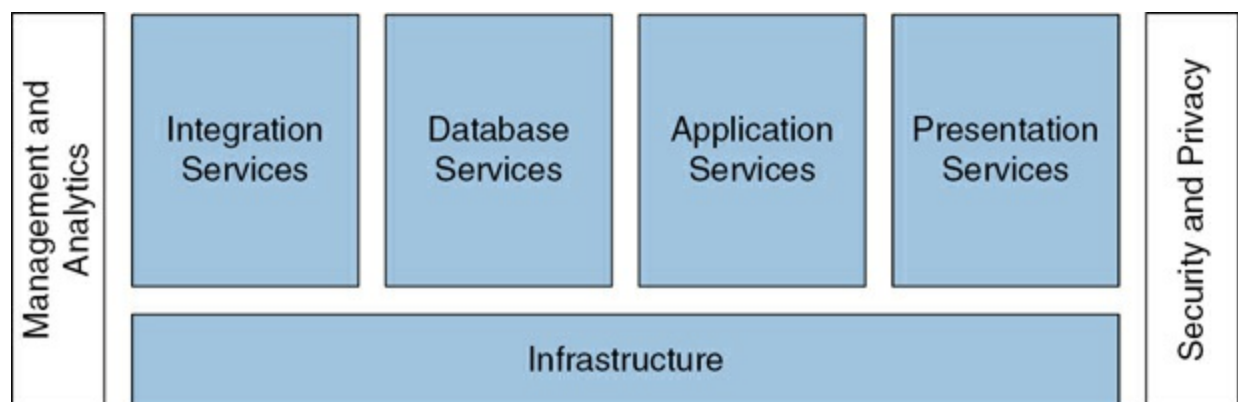


Figure 2-8 SaaS Architectural Model

At the bottom of the architectural model is the infrastructure. Going back to the house analogy at the beginning of this chapter, this is the foundation of the house as well as all the connectivity pieces that support the structure and basic functions, like electrical and plumbing. In the SaaS world, this is almost always handled by the provider or host of the SaaS service. Often referred to

as cloud service providers, these entities are discussed in more detail later.

On top of the infrastructure are four services: Integration, Database, Application, and Presentation. These services are the heart of any SaaS solution. The business logic and core functions reside here and, therefore, are some of the most critical pieces from an architecture perspective.

The pillars on either side of the model are Management and Analytics and Security. The functions contained in these pillars are pertinent for ensuring optimal efficiency and availability of the SaaS application. Both SaaS providers and their customers have high expectations here. A security incident and a lack of visibility and management of a SaaS deployment can be costly.

Table 2-2 summarizes the building blocks of the SaaS Architectural Model in Figure 2-8 and their overall function. You can use this table as a quick reference going forward. For additional details on each of these building blocks, refer to the proceeding sections, where we will discuss each in more detail.

Table 2-2 SaaS Architectural Model Building Blocks

SaaS Architectural Building Block	Function
Infrastructure	Composed of data center servers and networking equipment, Infrastructure handles the underlying compute, storage, and network functions of a SaaS application. It also includes the software running on the servers, including operating systems and virtualization and container management.
Application Services	As the heart of any SaaS solution, Application Services defines the architecture, such as microservices or serverless, along with all the business logic and core functions.
Database Services	Focused on the storage of application and user data, Database Services is responsible for the storage and access of structured, semi-structured, and unstructured data using relational and non-relational databases.
Presentation Services	When users interact with a SaaS application, they do so through Presentation Services. This architectural block defines the access methods, technologies, and protocols for the interface(s) between a SaaS application and its users.
Integration Services	SaaS applications are even more valuable when they can be automated and connected with business workflows and even other SaaS applications. Integration Services provides this connectivity through various prebuilt and custom connections, like APIs and adapters.
Security and Privacy	For effective security and privacy, SaaS application providers and users must work together in what can be a changing landscape of threats and bad actors. Various tools and methods are available for setting up the proper protections, including cloud security access controls, identity access and management, and visibility and monitoring.
Management and Analytics	Being able to accurately see what is happening in a SaaS application and having the ability to manage it at scale are critical. Management and Analytics encompasses these capabilities along with the associated technologies and tools.

Infrastructure

The Infrastructure block in the SaaS Architectural Model is probably the easiest to grasp. It is the part of SaaS that is most often hosted by a cloud

service provider. A CSP is simply a company that hosts scalable computing resources on the Internet for businesses. For example, AWS, Microsoft Azure, and Google Cloud are three of the largest and most well-known CSPs. Support and services for storage, compute, platform, and applications are provided by a CSP.

Instead of a business building out infrastructure for themselves, most SaaS solutions utilize one or more CSPs for this task. The benefits of using a CSP includes easy, scalable access to already-built hardware systems, networking, and software for hosting a SaaS application. You can think of infrastructure as the foundational bedrock with cloud computing at its core. On top of this bedrock reside the SaaS components and services. The infrastructure seamlessly enables and connects these together. The image that should come to your mind when thinking about the Infrastructure block in the SaaS Architectural Model is that of a data center with racks of servers, similar to [Figure 2-9](#).



Figure 2-9 Racks of Servers in a Data Center Are the Infrastructure Hardware of the SaaS Architectural Model

If you peer into the Infrastructure block for SaaS and break it down further, the three primary components are compute, storage, and networking. These are essentially the core elements of any data center and the foundation for every CSP. On top of this data center hardware, you also have system software for these elements, along with other tools. [Figure 2-10](#) summarizes these functions of the Infrastructure block in the SaaS Architectural Model.

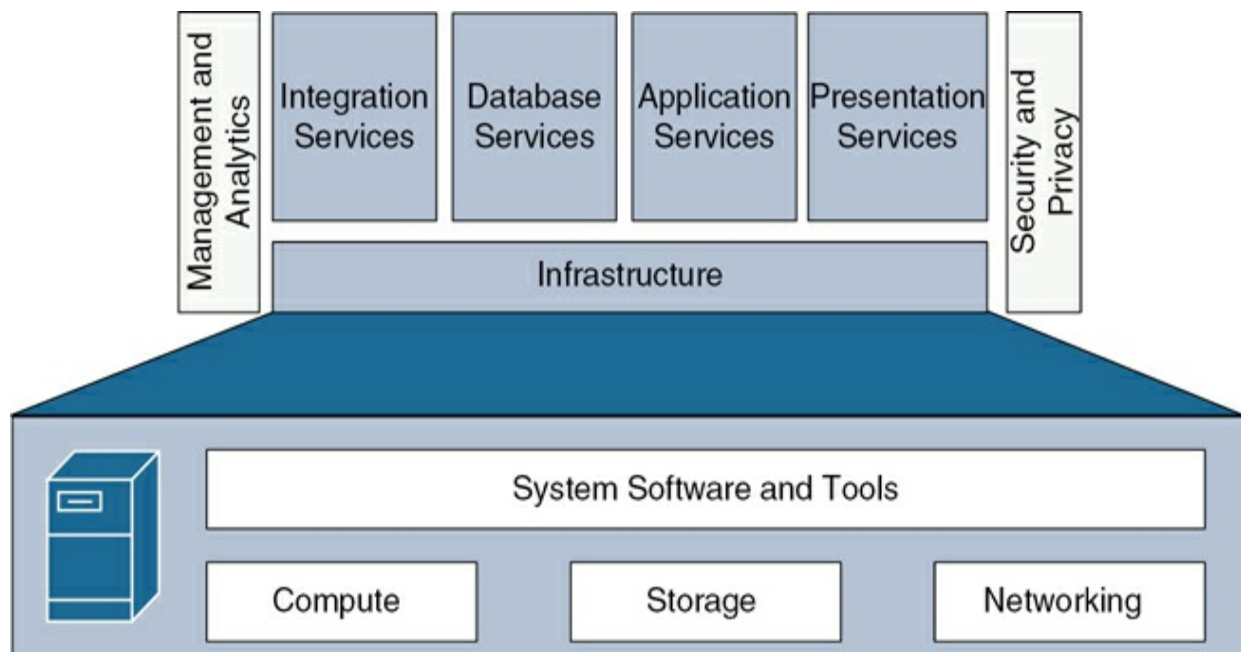


Figure 2-10 Infrastructure Block of the SaaS Architectural Model

Compute

In [Figure 2-10](#), compute, or computational power, is simply the processing capability, which can vary widely. For smaller applications and ones that are less complex, less compute power is needed. Similarly, if there are only a few users versus many users of the application, less compute power is also consumed. A SaaS provider will ensure that the right amount of compute is always available for its application.

In most cases, computational power refers to central processing units (CPUs) performing the calculations required by applications. In a data center, these CPUs are typically just more powerful and advanced versions of what powers your desktop or laptop. These CPUs have dozens of cores, with some having more than a hundred on a single physical chip. These powerful CPU chips are

then packaged so that usually a dozen or more can run in a single physical server. This server is then usually installed on racks with other servers, as shown in [Figure 2-9](#). High-bandwidth backplanes and network connections can join many of these physical CPUs together to share the computational workloads of various SaaS applications.

Note

You should think about the core on a CPU chip as an independent processing unit that can execute instructions and calculations. Early CPUs, if you are old enough to recall the first Intel Pentium P4 processors or anything before that line of chips, utilized just a single core. With a single core, all applications and the operating system had to share that single processing unit. Advances in chip hardware technology since then have allowed for more and more cores to be packaged into a single CPU. Multiple cores allow for parallel processing and a huge jump in computational capability for that same physical CPU chip.

You should be aware that many CSPs offer alternative compute power to CPUs. These alternatives include graphical processing units (GPUs) and quantum processing units (QPUs). GPUs are common in desktop and laptop computers for their ability to provide increased performance with graphics. However, they are also much faster and more efficient when working with artificial intelligence (AI) models. CSPs offer GPU access to improve performance for AI applications. While quantum computing is still growing, QPU resources are available from some CSPs for those that have applications that could benefit from this type of computational resource.

Note

An important part of compute is memory. Generally, the more memory available to processing units, the more efficient these resources can be. At the same time, memory is expensive and raises the cost of compute for a CSP. Therefore, CSPs usually try to balance the increased performance and efficiency with the cost when it comes to determining the amount of memory for servers.

Storage

The next primary component in the SaaS Architectural Model Infrastructure block is storage. Just as you need storage in your PC for applications and files, the cloud also needs this functionality but at a much larger scale. Instead of a single hard drive that you typically find in a PC, a CSP would have many thousands networked together to optimize performance and reliability. Naturally, the drives found in CSP storage arrays in data centers are higher quality than consumer versions and are hot swappable to decrease downtime.

To best understand the power and scale of storage at the Infrastructure level, it is easiest to look at a dedicated storage server. Instead of being full of just CPUs, a storage server is full of storage drives. In a 4 rack unit (RU) server, more than a thousand terabytes or one petabyte of storage can be attained with all slots in the server filled with higher capacity, traditional drives. These traditional drives utilize spinning platters and magnetic encodings for storing data. While traditional hard drives can be used for maximum storage capacity in a server, solid-state drives (SSDs), including Non-Volatile Memory express (NVMe) drives, are preferred for improved performance, but at a higher cost and lower total amount of storage. [Figure 2-11](#) shows a Cisco storage server with its top open to reveal its 56 slots for storage drives. Imagine a CSP data center with many racks of these storage servers for handling SaaS solutions.



Figure 2-11 Cisco UCS S3250 Storage Server Loaded with 56 Hard Drives

To ensure reliability for the data residing on their storage servers, CSPs use various redundancy protocols. The most popular one is Redundant Array of Independent Disks (RAID), which has different levels of redundancy configuration depending on the needs of the CSP. The CSP must balance the level of redundancy that is needed with the storage space required.

Redundancy costs disk storage space, so that lowers the total capacity of a storage server. The storage cost of redundancy is directly related to the chosen RAID redundancy method.

For example, with RAID 1, an exact copy of a disk drive is made. For every disk drive that is used for data storage, another drive is written at the same time to mirror it or maintain an exact copy of it. This drops storage capacity in half but provides the best redundancy. With RAID 5 and RAID 6, parity bits are added to the data being written to enable redundancy. Of course, this

costs disk space as well, but with many drives (e.g., 10 or greater for RAID 5), you can achieve 90 percent disk usage efficiency or greater. In this way, RAID 5 can handle any single disk failure and recover all the data. A failed disk in a RAID 5 array is simply removed, and a new blank drive is inserted. The blank disk is then configured to the same data as the failed drive. RAID 6 utilizes two parity bits, so it can handle two simultaneous disk failures. This gives you more redundancy but less disk usage efficiency compared to RAID 5.

This example is a simplified look at RAID and how storage redundancy can be implemented by a CSP. This approach can be further extended to nested RAID, where you layer different RAID types to extract additional performance or redundancy benefits. The use case and storage offering by the CSP determine the exact method or system used to provide redundancy. In the case of SaaS, the SaaS application provider handles all of this process with the CSP, and it is transparent to you, the user.

Note

Hyperconverged infrastructure (HCI) is a popular technology that combines compute, virtualization, storage, and networking into a single cluster. With HCI, all storage resources are abstracted into a large pool and managed as a single entity. This technology offers a few advantages, including enhanced redundancy as the data is replicated onto different disks on different servers that are part of the cluster. It allows for a disk failure or even a node failure while keeping data availability and performance.

Network

In the previous sections, we discussed compute and storage. But how do you interconnect all these servers with compute and storage resources? These resources must be connected not only within a rack but within the data center itself and often across multiple data centers in different geographic locations. Furthermore, how do users accessing a SaaS application get routed to these resources efficiently? The network component of the Infrastructure block is the answer to these questions.

You can think of the network as the Infrastructure component that

interconnects hardware and software resources in the cloud. Specific examples of network components are switches, routers, load balancers, and firewalls. These are all traditional pieces and technologies that are part of just about any enterprise computer network. So, at a high level, there is obviously an overlap in using these components in an enterprise environment and a data center one. However, just like with compute and storage, the scale and performance necessary in the data center infrastructure dictate that there are some key differences as well, especially in the switching space.

To understand these differences, you should first think about the traffic patterns in a typical enterprise environment and a data center one. In an enterprise network, the major traffic flows are mainly *north-south*, meaning that connecting end users to applications in a data center or to the Internet is the most important objective. Imagine a topology diagram with end users and their locations on the bottom and the core of the network and the Internet at the top. Reading this topology diagram like you would a typical map with north being at the top and south being at the bottom may help you picture the north-south relationship.

Instead of a north-south relationship, a lot of the data center traffic has an *east-west* flow. Compute and storage in the data center and the applications using them are often just communicating with one another. For example, storage and compute can be shared between servers, racks, and if necessary, even other data centers. Or applications are reading and writing from databases or moving files between servers in various locations. Because of this back-and-forth communication at the same level, usually depicted as connections between components on the left (west) and right (east) of a topology diagram, connections must be faster and with less latency.

Switching probably best exemplifies these network differences between enterprise and data center. With switching, vendors typically have an enterprise line of switches and another line of switches optimized for data center. For example, Cisco's Catalyst line of switches is aimed at the enterprise while the Nexus brand is a data center switch.

So, what makes a Nexus data center switch different from a Catalyst switch? There are actually a good number of differences from the operating system to the hardware itself. Some of the main differences include the following:

- **Higher Performance:** Nexus switches generally have a higher throughput and less latency in their switching fabric. This capability provides better support for the high-bandwidth, east-west connections along with the minimal delay required in the data center.
- **Enhanced Programmability:** Nexus switches have an advantage in programmability due to their support for SDN using Application Centric Infrastructure (ACI). ACI is Cisco's solution for capturing higher-level business and user intent in the form of a policy. These policies are translated into network constructs that enable dynamic provisioning, proactive security, and the simplified management of workloads and data connections across single or multiple cloud networks.
- **Data Center Protocol Support:** Data center switches have hardware and software support for protocols that are just found in that environment. For example, Fibre Channel is a fast, data transfer protocol that is primarily used for connecting storage servers to form a storage area network (SAN) and to connect to other servers for data processing. Nexus switches have support for Fibre Channel as well as Fibre Channel over Ethernet (FCoE), which takes fibre channel frames that typically run over a fiber-optic connection and encapsulates them over Ethernet.

Note

You probably noticed that the word *Fibre* in Fibre Channel has the British English spelling of the word *fiber*. This is intentional. When FCoE was standardized, the spelling was changed from *fiber* to *fibre* to retain approximately the same name but no longer tie the protocol exclusively to an optical transport.

Figure 2-12 illustrates a high-level data center topology using a leaf-spine model. This architecture and variations of it are common in modern data centers, and this figure shows the layers of data center switches and how they would interconnect. At the bottom of the figure, you can see the compute and storage servers. Switches are then overlaid as leaf nodes with the connected spine switches serving as the literal backbone. In this figure, ACI on Cisco Nexus switches enables and optimizes this topology.

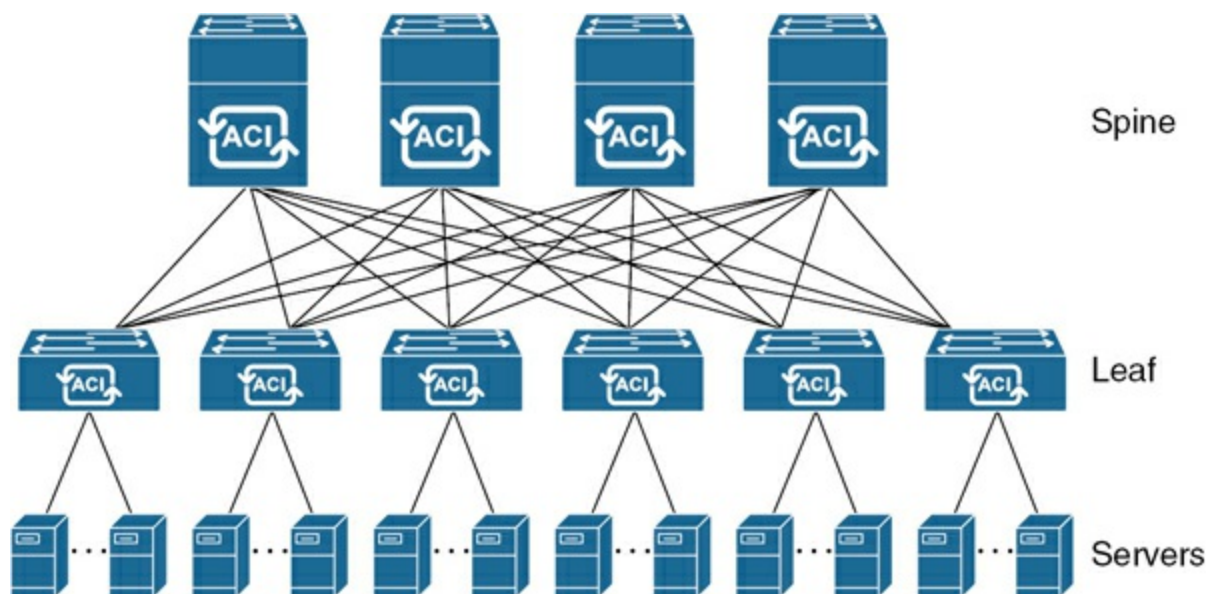


Figure 2-12 Simplified Leaf and Spine Data Center Topology

Another important concept to understand in the networking component of the Infrastructure block is virtualization. Virtualization, which we will cover in more depth from the compute side in the next section, is also applicable to networking devices, such as routers and firewalls. If you think about it, data centers are packed with servers, and often there is not a lot of space for dedicated firewall and router hardware. So, it makes sense to use some of the compute and storage resources already present in large quantities to run virtual firewalls and routers. This is referred to as network function virtualization (NFV).

NFV decouples services like routing, security, and load balancing from the underlying hardware platform that it typically resides on. With NFV, these services or functions run inside virtual machines. This means that they can quickly be deployed, moved, and placed on demand exactly where they are needed in an environment. NFV turns what used to be dedicated hardware functions into software. So, from a functional perspective, NFV is a network component, but because it is software, it also fits into the system software and tools component that is covered in the next section.

System Software and Tools

If you refer to [Figure 2-10](#), you see that the system software and tools component of the Infrastructure block resides above the components of

compute, storage, and network. As we alluded to in the previous section, the main reason for this is virtualization. Virtualization abstracts the compute, storage, and network hardware and is foundational to efficiently building and scaling SaaS applications.

If you look at virtualization at a high level, it logically segments the underlying hardware. It is often referred to as creating a computer within a computer. [Figure 2-13](#) provides an example of this segmentation.

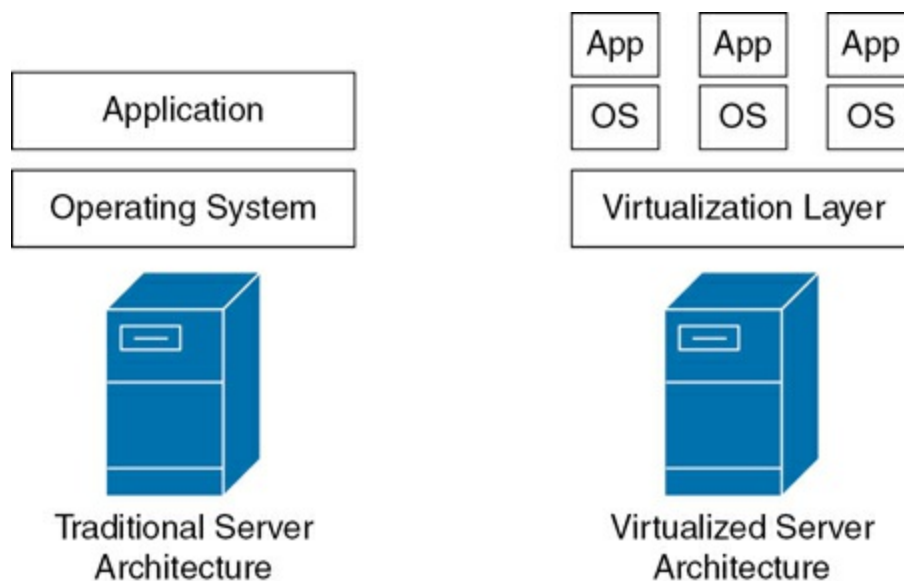


Figure 2-13 Comparison of a Traditional Server and a Virtualized Server

In a traditional server architecture, an application is bound to the hardware that it is running on. With virtualization, this is no longer the case. A virtualization layer, which is simply software known as a hypervisor, allows for the sharing of the hardware resources among multiple operating systems (OSs) that sit above it. Each OS and app run in an emulated computing system called a virtual machine (VM). This OS and its application(s) interfaces interact with the hypervisor instead of directly with the server hardware. For all intents and purposes, the OS and application(s) do not even know that they are not running directly on server hardware.

Note

A number of vendors produce hypervisor software for virtualization commercially. Some of the major players include Microsoft, Red Hat, and VMware.

Virtualization greatly increases the efficiency and scalability for applications running in a data center. For example, picture three traditional servers with an OS and an application, like in [Figure 2-14](#). One server runs an email application, another runs a web application, and the last one is dedicated to a database application. Because of the traditional server architecture, you have a 1:1 relationship between the physical servers and the application.

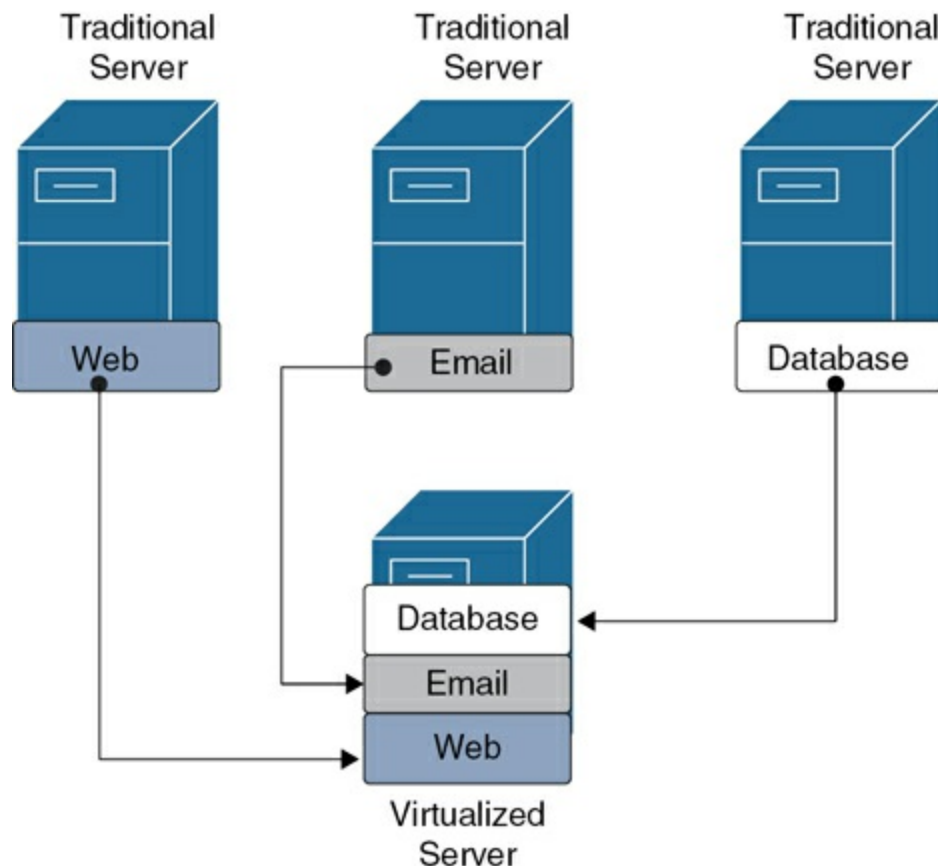


Figure 2-14 Virtualization Enables Scalable and Efficient Data Center Architectures

Utilizing virtualization, multiple applications can now reside on a single physical server. The web, email, and database application can run on a single server in a virtualized environment. Overall, virtualization in a data center allows servers to be used more efficiently, and fewer servers are needed to run the same number of applications.

Virtualization provides other benefits as well. Because the OS and its application are just files, they can easily be moved to other hypervisors on

other servers if a physical server fails. For example, think about multiple compute servers with hundreds of cores all accessing a shared storage server, like the one depicted in [Figure 2-11](#). A shared storage server could hold many VMs, and if one server is having an issue, you can easily load the VM to another server. Additionally, if more performance is needed, you could easily add compute resources to the VM without a restart, unlike a physical server. This portability is also handy for testing and DevOps, where a VM can be copied or cloned for testing in other environments.

Containers build on the concept of virtualization. A container is a lightweight software package containing an application along with all the components, functions, and dependencies it needs to run on a specific OS. Modern applications, especially in the SaaS space, typically have multiple containers that each perform a specific function. Containers are more streamlined and efficient than a VM because they are smaller in file size and don't require a hypervisor. [Figure 2-15](#) provides a high-level comparison of a VM and container architecture.

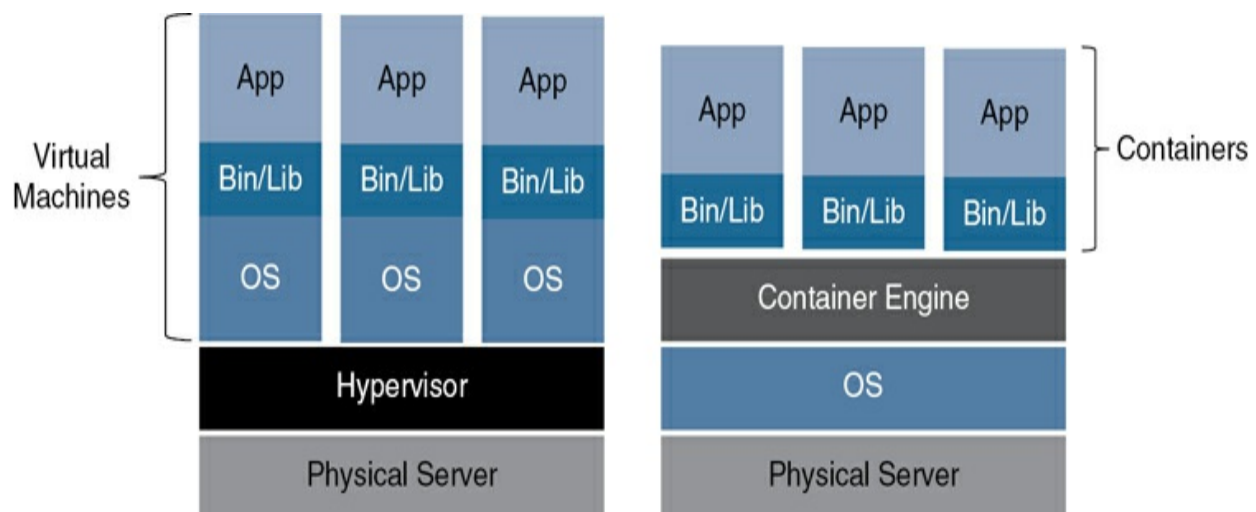


Figure 2-15 Comparison of Virtual Machines and Containers

As illustrated in [Figure 2-15](#), containers utilize a container engine that resides on an OS, such as Linux, directly on the physical server. This container engine plays a similar role to the hypervisor on virtual machines except that it seamlessly joins various containers to a common OS.

Just about every application has various binaries and libraries associated with it. These files and functions are necessary for the application to fully run,

regardless if the application is running on a VM or in a container. Containers ensure that an application or multiple applications stay joined with their supporting binaries and libraries. This allows for a container to be spun up in another environment with the same OS and easily run. All the dependencies necessary for the application to run are stored in the same container.

Note

Just as you can run multiple applications on a VM, you can also run multiple applications in a container. This capability makes sense when the same or similar dependencies are present so that they can be shared and/or the applications in the container are meant to be run concurrently.

As previously mentioned, and shown in [Figure 2-15](#), containers have a common OS layer while an OS is part of each VM. Because of this structure, the container overhead is less compared to a VM running the same application. For example, containers are usually megabytes in size compared to gigabytes for VMs. Additionally, containers can be loaded in seconds because the OS is up and running, whereas a VM must go through the whole process of booting up the OS when it is brought up.

Note

Kubernetes (K8s) is probably the most popular open-source container orchestrator. Commercially, Docker is another container management solution that is commonly utilized.

At this point, you have learned about the Infrastructure block of the SaaS Architectural Model. The Infrastructure block includes compute, storage, networking, and system software and tools and is most often found in a data center environment in the cloud that is managed by a CSP. A SaaS application provider typically works with a CSP to ensure that the correct infrastructure is in place for the SaaS product that is being offered. We gave you a peek at what is happening behind the scenes at the CSP infrastructure. From a SaaS customer perspective, this infrastructure is hidden. This is a huge benefit. After all, one of the main reasons you choose SaaS solutions as a customer is to remove the complication of building and maintaining this hardware and its complexities.

Application Services

The Application Services block is the engine of any SaaS product. All the other blocks of the SaaS Architectural Model depend on it and connect to it in some manner. For example, the Infrastructure block hosts application services by providing the hardware and software structures, like containers. Database services maintain a connection to Applications Services for data access and management purposes, whereas Presentation Services capture user inputs for configuration and provisioning. Essentially, Application Services power the SaaS solution and contain the functionality and features that make that solution viable to customers. Think of this block as the programming or code that provides core functionality, interconnects all the other services, and brings the SaaS application to life. [Figure 2-16](#) provides a more detailed view of the Applications Service block of the SaaS Architectural Model.

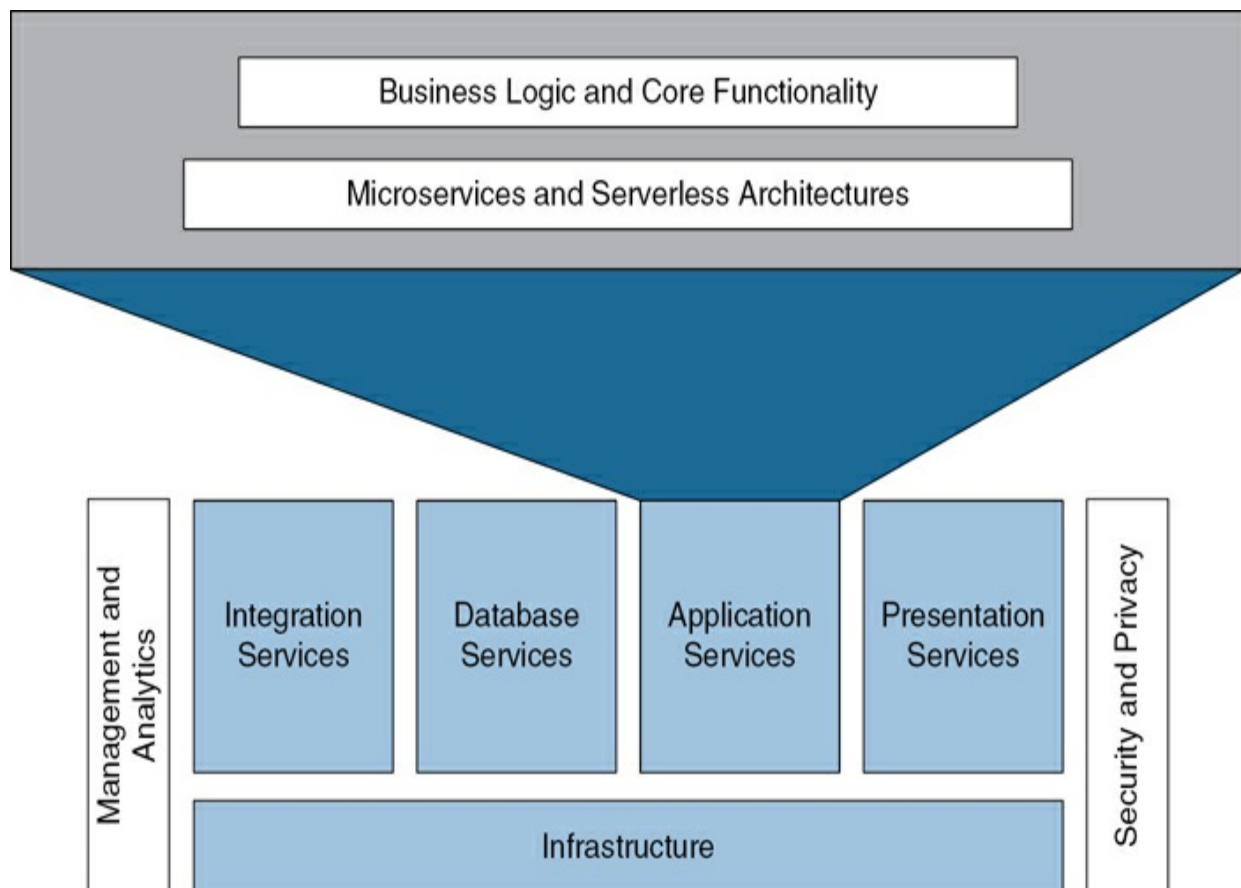


Figure 2-16 Application Services block of the SaaS Architectural Model

If you further subdivided Application Services, you could create two layers.

On the bottom are microservices and serverless architectures and on the top are the business logic and core functionality. We will cover both of these layers in more detail in the proceeding sections, starting with microservices and serverless architectures.

Microservices and Serverless Architectures

Microservices are more architecturally conceptual than the containers (encapsulated applications and dependencies) discussed previously. This is an important distinction. As an architectural concept, microservices can run in either VMs or containers even though most modern microservice architectures opt to run in containers.

Before diving into microservices, you need to understand that cloud application development has been evolving. Traditionally, applications were coded together as a whole with functions and services intertwined. This is often referred to as a monolithic architecture. One of the big disadvantages of this architecture is that this tight integration of services and functions means that a change in one component can have a negative impact to the entire application. Additionally, as you add features to the codebase, the complexity increases greatly. You need more technical resources to maintain an architecture that is becoming less agile. Another disadvantage is that you cannot easily scale an individual portion of an application. You can be left in the situation where you must run another full version of the application on another server or VM just to scale a small part of it.

Applications developed using microservices take a different approach. Microservices segment an application into API-connected functions. In some cases, an event bus utilizing a publish-subscribe mechanism could be used instead of APIs to facilitate communications between microservices. Microservice functions are loosely coupled and can be developed and deployed independently. This is called a microservices architecture. [Figure 2-17](#) shows a basic comparison between a monolithic application architecture and a microservices one.

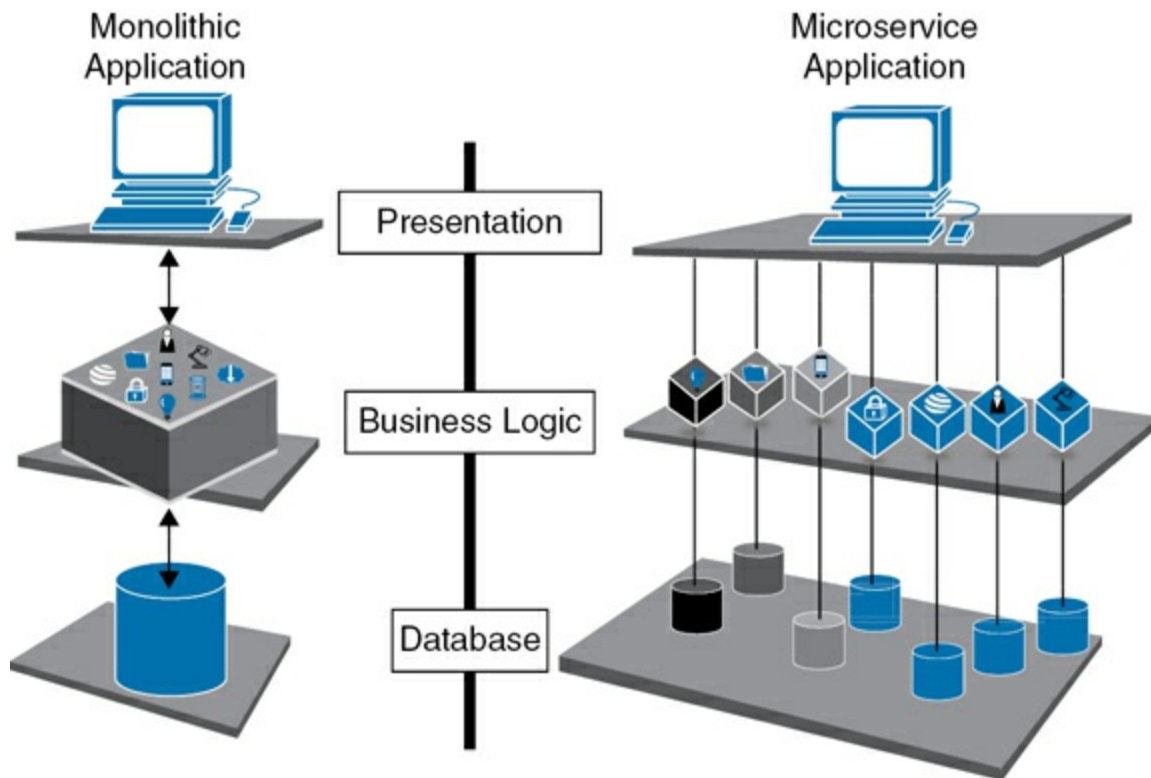


Figure 2-17 Monolithic Architecture Versus a Microservices Architecture

In a microservices architecture, the business logic is partitioned into loosely coupled but independent services. In [Figure 2-17](#), the microservices business logic, represented by the cubes with icons, are connected via APIs for intercommunication. Each microservice also has its own database, but in some cases, a database may be shared between microservices. A common example of microservices architecture is in the online ordering and e-commerce space. A microservice and a corresponding database could be created for the Buy button on a product page, another for calculating shipping, and another one for calculating sales tax.

Note

The term *API* can sometimes be a bit confusing because it can be a general term that refers to a set of defined methods or protocols used for communicating between software components. API can also mean a specific type of API known as a Representational State Transfer (REST) or RESTful API. In the microservices world, this narrower definition of a RESTful API is most common. A RESTful API is often referred to as a request-response connection that is

stateless and uses a client/server architecture compared to an event-based API that utilizes an event bus. We will provide more information on RESTful APIs in the “[Custom Integrations with APIs, Webhooks, and WebSockets](#)” section later in this chapter.

Communication between microservices is not always via request-response APIs, like REST. The other option is an event bus. RESTful APIs are great for one microservice connecting to another microservice to exchange data but are inefficient when you need to have one microservice communicate with multiple microservices at once to share the same piece of data. In this instance, an event bus is a better way. [Figure 2-18](#) provides a simplified overview of an event bus and how it allows a single message about an event to be sent and then received by two other microservices.

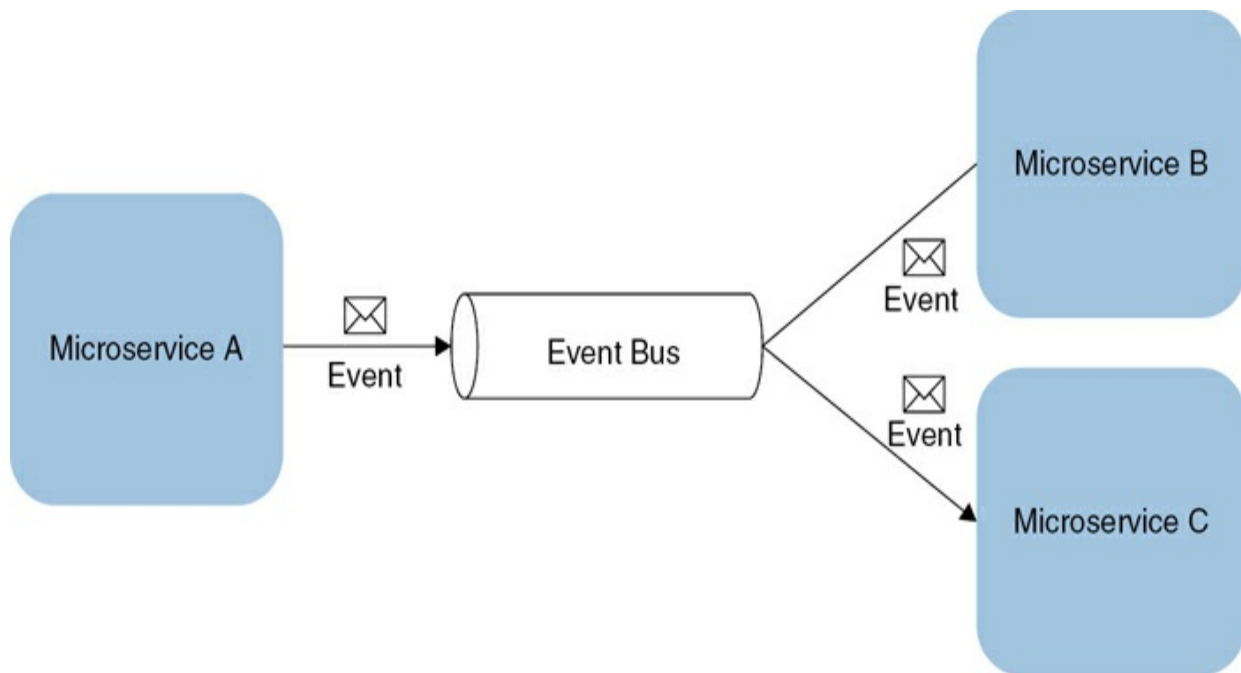


Figure 2-18 Microservice Communication Utilizing an Event Bus

An event bus works by employing a publish-subscribe system. In essence, it is a broker service. Microservices that need to send information can publish data, or events, to the event bus that is tagged with an event type identifier. Other microservices can subscribe to that event type, or topic, and then receive all the corresponding messages. This approach allows for more efficient communications when data needs to flow asynchronously between multiple microservices simultaneously.

Several different protocols can be used as an event bus between microservices. Two of the more well-known ones include Apache Kafka and RabbitMQ. Event buses are also available directly from cloud providers. For example, Amazon utilizes AWS EventBridge, Azure provides Azure Event Hubs and/or Event Grid, and Google Cloud offers Google Cloud Eventarc.

Note

You will hear the term *service-oriented architecture (SOA)* quite often in discussions of monolithic and microservices architecture. SOA was popular before the big move to microservices, and they are similar concepts. In fact, microservices architecture evolved from SOA. Both divide large applications with high complexity into smaller chunks to make working with the code easier. However, there are some key differences. Think about SOA as a broader approach to architecture with it being enterprise-wide, whereas microservices are applied to an application. Additionally, SOA emphasizes service reusability and the sharing of resources among programs, like databases, while microservices are more focused on independence and decoupling.

Be aware that for a typical user, with access to just the presentation layer or user interface (UI) of an application, it is almost impossible to tell whether a monolithic or microservices architecture is being utilized. The differences are in the backend of the application and include the business logic and the database storage.

Large applications needing to scale usually benefit the most from microservices. For example, Uber, the ride-sharing company that connects users with a car ride using a mobile application, is a well-known use case of a SaaS company that transitioned to microservices. Uber started out as a monolithic application and ended up moving to a microservices architecture for scalability and efficiency reasons. [Figure 2-19](#) depicts Uber's evolution from a monolithic architecture to a microservices architecture.

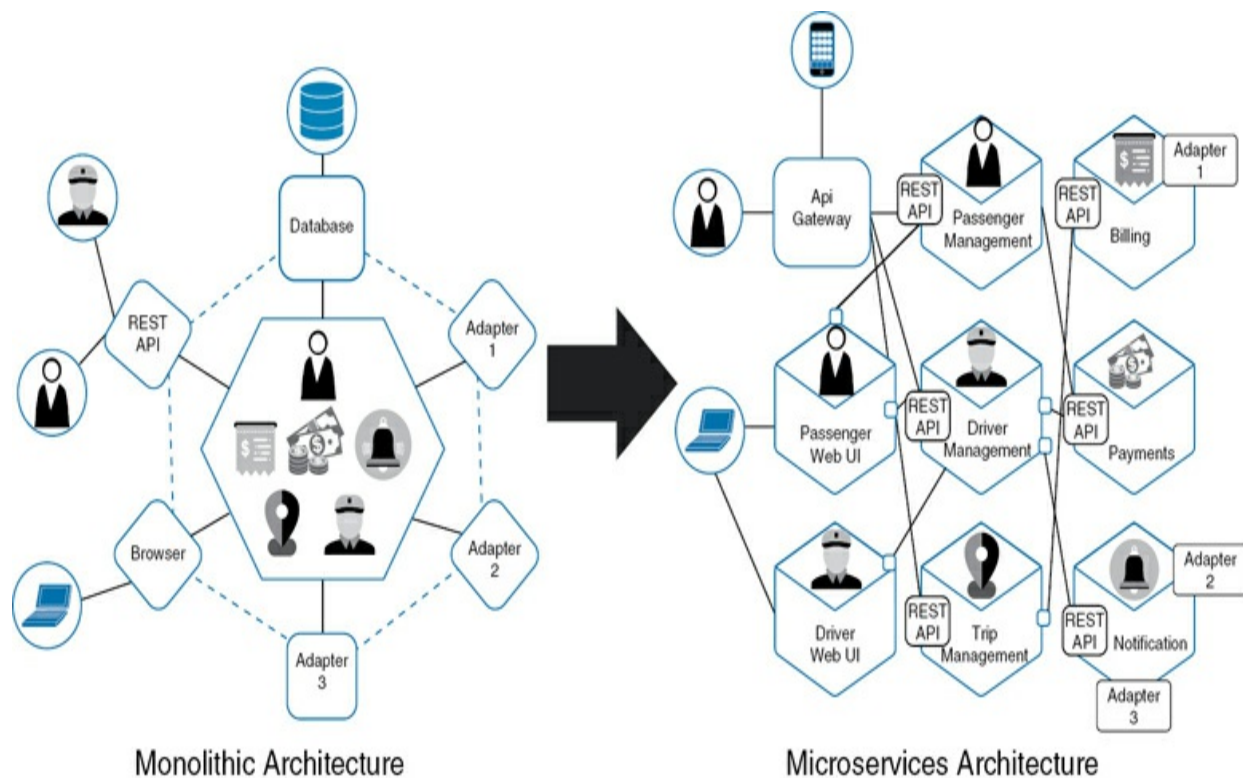


Figure 2-19 Uber's Microservices Architecture

In [Figure 2-19](#), you see that Uber segmented its monolithic architecture into well-defined microservices that align to business functions. Connected with APIs through an API gateway, these independent code segments offer distinct advantages. For example, each microservice is developed and maintained by separate teams that can push code updates and changes without affecting other services. Scaling can also be easily accomplished by expanding microservice resources.

Migrating from a monolithic architecture to a microservices one can be a considerable undertaking for deployed SaaS applications. This migration can take years, but the business value is typically worthwhile. With the proper resources, starting with a microservices architecture makes sense, but SaaS applications often start small where a monolithic approach seems easiest. Sometimes there is just a single developer, and the application seems simple enough. Only later, after rapid growth in features, capabilities, and users, does a microservices architecture become necessary. Today, a microservices architecture has become the preferred approach in software development, and most SaaS applications are being built using microservices from the

beginning instead of the monolithic approach.

As mentioned previously, a microservices architecture is commonly deployed using containers. This is referred to as containerized microservices and is common in SaaS applications. One or more microservices can be assigned to a container. This assignment allows for the fast scaling of microservices on demand and the rapid deployment of fixes and new capabilities. Additionally, containerized microservices can be spread across multiple VMs in a cluster. This capability provides some redundancy and allows the application to continue running even if a VM goes down or is unavailable. As an example, [Figure 2-20](#) shows a basic container cluster where microservices are replicated in containers across multiple VMs. Even though VM 1 is down, the microservices are still available to the application in VM 2 and VM 3.

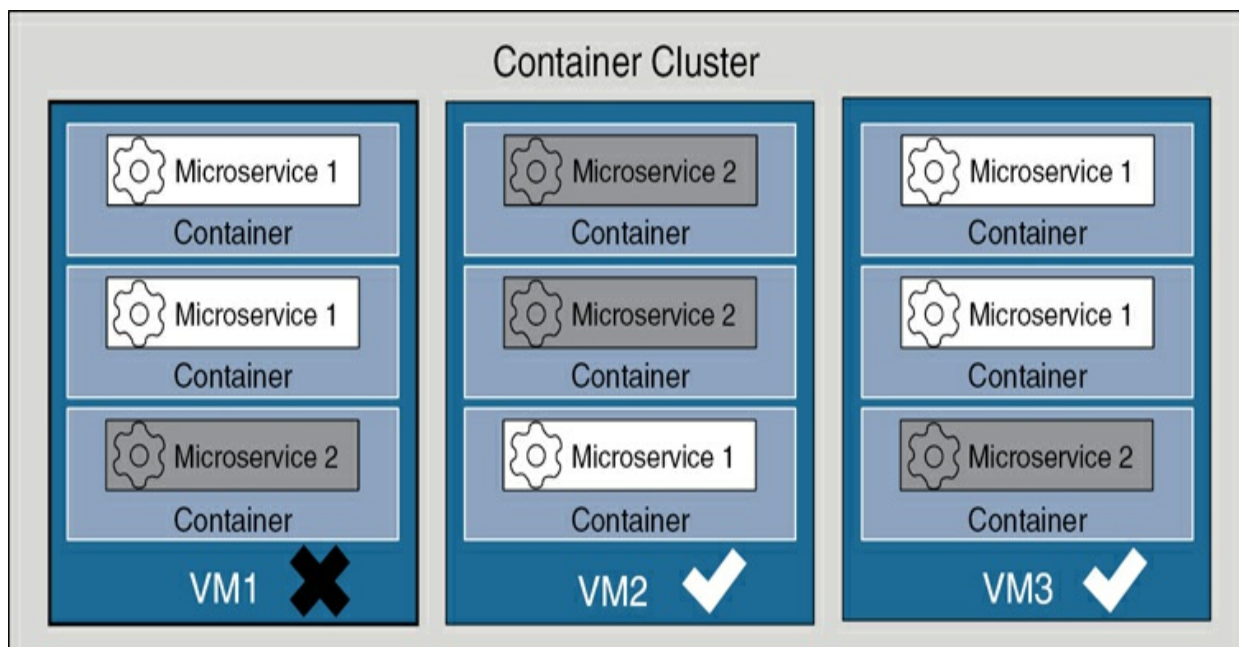


Figure 2-20 High Availability in a Cluster of Containerized Microservices

Serverless architecture is another computing model often found in SaaS applications. Despite the name, a server is still present, but *serverless* alludes to the fact that the server is managed completely by the CSP and is abstracted from the programming function. If you refer to containerized microservices, you have a container and optionally a VM that must run on a server. This approach requires that server resources, like compute, storage, and possibly networking resources, are allocated and maintained to host the containerized

microservices. With a serverless architecture, the server resources are still present but are now completely maintained by the CSP. In this way, functions and services can be built directly on a serverless platform hosted by the CSP, and then these functions can be accessed and used as necessary. With serverless, SaaS application developers can just write code and not have to worry about managing the infrastructure.

The most common form of serverless is Function as a Service (FaaS). FaaS allows you to upload small pieces of code, or functions, to the cloud for execution as needed. You are typically only charged on a per request basis when a function is created and run. As the number of requests increase, more functions can be initiated to handle the increased load. When there are not any requests, the function is terminated, and no charges from the CSP are accrued. SaaS providers constantly seek to minimize their CSP cost, so they must take into account the economics of using FaaS and other forms of serverless functions for various tasks in their overall SaaS application architecture.

An example of a serverless function could be an image modification function for a social media SaaS application. Whenever a user submits a profile picture, this function converts the image to a standard size and image format. If no profile pictures are being submitted, this function is not present. It is created and run only when requested. If this function runs on an image and no other profile pictures are subsequently submitted, this function is ended after a certain amount of time. On the other hand, multiple copies of this function can be created and run concurrently if many users submit profile pictures at once. [Figure 2-21](#) models a sample flow of this event.

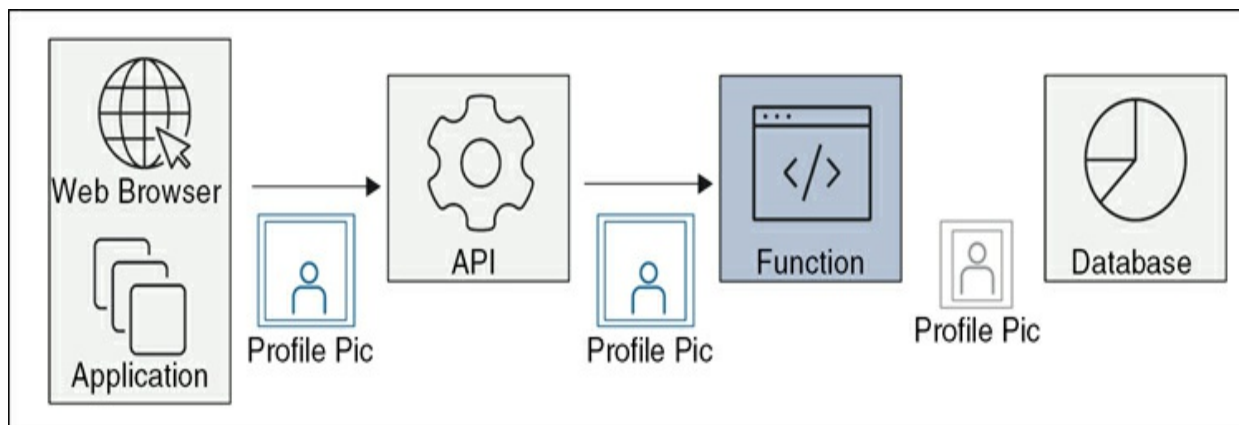


Figure 2-21 A Serverless Function for Image Modification

As you can imagine, dividing up an entire SaaS application to a serverless architecture could be challenging, depending on the application size and other factors. Often, smaller functions, like image adjustments or simple database reads and writes, are pulled out of larger SaaS applications and moved to a serverless architecture. More complicated flows in serverless require the chaining together of multiple serverless functions. For example, a new user fills out a form to access your SaaS service. The creation of a new user in the database based on this form data could be one function. After this first function finishes executing, it could link to another function that sends a welcome email to the user's email address.

Note

Most cloud providers have a serverless offering in the FaaS space. For example, three of the more well-known ones are AWS Lambda, Azure Functions, and Google Cloud Functions. Many functions are already built and available for anyone to use on these platforms. Reusing a shared, prebuilt function can provide a significant savings on development time.

The main advantages of a serverless architecture include cost, because you are only charged when your function is run, and ease of deployment. You do not have to worry about managing the underlying compute, storage, VMs, and containers. Instead, you just focus on developing the functions needed by your SaaS application. The disadvantages include losing control of the underlying infrastructure, security, and cold start delays. You are dependent on the CSP for allocating enough capacity for your functions to run effectively. Also, security could be an issue because the CSP can run your functions concurrently with other customers on the same hardware. Cold start delays of up to a few seconds can occur when functions are dormant and must be created and run. Depending on the infrastructure load level, several seconds could pass before a function finishes executing after a cold start. This delay may not be acceptable for some applications. However, once the function is up and running, the time to execute is minimal.

Both microservices and serverless architectures are common in the SaaS

world. Either one or both together can be effective in hosting a SaaS application. The SaaS application itself is the cornerstone of any SaaS solution and is made up of the business logic and core functionality. In the next section, we'll dive deeper into this topic.

Business Logic and Core Functionality

How does ThousandEyes aggregate data from numerous agents and provide you a real-time picture of network performance and problems for not only your own network but the Internet as well? How does Webex Events bring together thousands of people around the world on an interactive webinar with HD video and broadcast-quality audio? These are examples of the core functionality and business logic of two SaaS applications. Additionally, these capabilities are key differentiating factors that can make a specific SaaS application unique and directly contribute to its value proposition.

When you break it down to its simplest form, the business logic and core functionality are computer code. It is the programming that the SaaS application executes that makes that application distinctive and valuable to its users. This application code resides on servers in the backend. The backend is the place where the processing, storage, and manipulation of data occurs, and the frontend is the visible portion that SaaS application users interact with. Integration Services and Database Services can also be included as part of the backend. The concept of frontend and backend is discussed in more detail in the upcoming “[Presentation Services](#)” section.

Many different programming languages can be used in the back-end coding for the business logic and core functionality in a SaaS application. [Table 2-3](#) highlights some of the more popular ones and their associated development frameworks.

Table 2-3 Common Back-End Programming Languages for SaaS Applications

Programming Language	Common Frameworks	Description
JavaScript (Node.js)	Express.js	JavaScript (or JS) is one of the most popular programming languages. Although it is mostly used in front-end coding, the Node.js environment opens it up for back-end coding. This makes JS somewhat unique in that it can be used for both back-end and front-end coding.
Python	Django Flask	Python is one of the most popular SaaS back-end languages because it is easy to learn, easy to understand, and has a large user base. It also has an extensive number of dictionaries, packages, libraries, and frameworks that can be imported and utilized for just about any programming use case, including machine learning (ML) and data analytics.
Java	Spring Boot	Java already has a well-established enterprise presence due to its performance and portability. This means it has a mature environment that allows for the development of secure and scalable SaaS applications, especially for large-scale SaaS solutions.
Ruby	Ruby on Rails	Ruby, along with its Ruby on Rails framework, is known for being developer-friendly. It is a preferred language when rapid prototyping and development is important.
Go (Golang)	Gin Beego Echo Fiber	Go is one of the fastest-growing back-end languages for SaaS applications because of its balance of simplicity and performance. Developed by Google, it comes with a powerful library and is good for building salable and efficient applications.
PHP	Laravel Symfony	Often criticized for its performance and security, PHP is a stable, well-established language that is still widely used for SaaS applications because of its ease of learning, versatility, and a large and active support community.

You should note that dozens of other programming languages and

frameworks can be used for coding a SaaS backend other than those listed in [Table 2-3](#). Many factors go into the selection of a back-end language, including performance, scalability, and most importantly sometimes, the preference and expertise of the developer or development team. Just about any programming language can be used to code a SaaS application but certain use cases and goals may make one or more language a better choice.

The Application Services block of the SaaS Architectural Model is the brain of any SaaS solution. One or more of the programming languages highlighted in [Table 2-3](#) can be used to code the SaaS application itself, and then this program will reside on one of the architectures discussed here. For example, you might have your SaaS application written in Python and then deployed to a microservices architecture. These microservices might then reside in containers, as shown in [Figure 2-20](#). Various databases are then associated with the microservices to provide the Python code storage capabilities for its data. These databases are contained in the Database Services block of the SaaS Architectural Model and are covered in the next section.

Database Services

SaaS applications are usually heavily reliant on databases to host large amounts of customer data, such as user profiles, configuration settings, system logs, and performance and usage data. CSPs and even third-party vendors offer various database options, but it is important for SaaS application developers to understand the type of data that is being stored and its requirements, such as access and security.

Here, we will discuss the types of data along with various data storage options that SaaS applications typically use. Just like most of the other architectural blocks, the workings of the Database Services are largely hidden from the user. You can look at the information here as providing a behind-the-scenes peek at the decisions and structures that are necessary for efficiently storing SaaS application data. [Figure 2-22](#) provides more details of what is handled by Database Services and the main points of discussion.

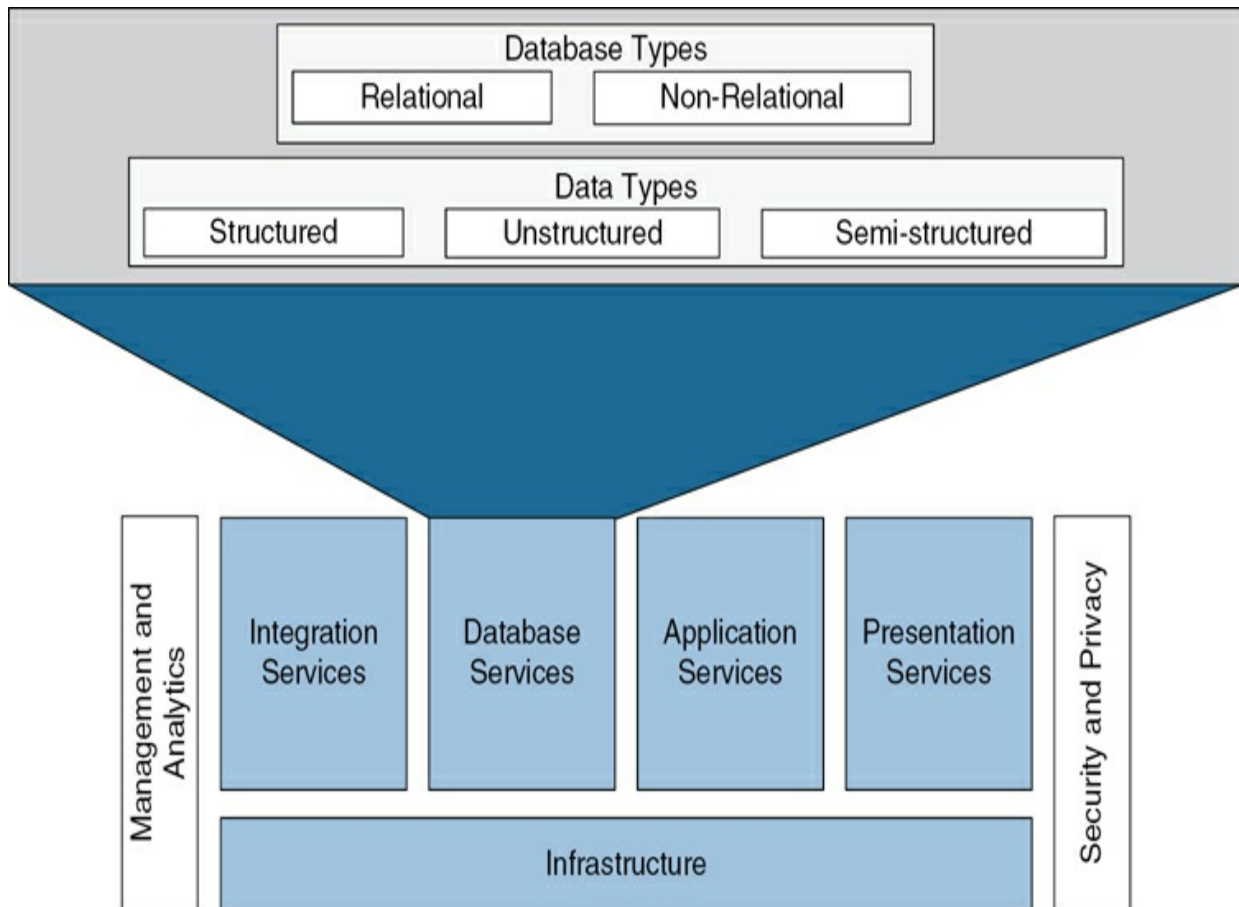


Figure 2-22 Database Services Block of the SaaS Architectural Model

In the following section, we will define structured, unstructured, and semi-structured data types. These are important concepts because databases are built to efficiently handle a certain data type. Later, we also will take a closer look at this topic and dive deeper into the various types of databases commonly seen with SaaS and the data types that they are designed for.

Structured, Unstructured, and Semi-Structured Data Types

The concepts of structured, unstructured, and semi-structured data are key to any discussion about Database Services. The reason is that all data is not the same, so it can't be treated the same if the goal is to store it efficiently. Storing data efficiently allows for its easy access and maintainability. For example, it is a bad experience for users to have a long delay every time they want to access detailed product information from a web page. This delay could occur if the wrong database structure is in place that is not scalable or

optimized for the type of data that is being stored. [Table 2-4](#) provides a quick summary of the common data types.

Table 2-4 Summary of Data Types

Data Type	Definition	Example
Structured	Data that is formatted and organized per a defined model or schema	Spreadsheet, customer information
Unstructured	Data that lacks formatting and a defined way of easily organizing it	Image, video, speech, text
Semi-Structured	Data that blends some well-defined and structured elements with other elements that are unstructured	Email, digital photos with Exchangeable Image File Format (EXIF) metadata

The data type that most people can relate to is structured data. Structured data follows a model or schema that defines how the data is represented and organized. This schema is usually referenced in a tabular form with rows and columns defining the data attributes. This form makes structured data easy to access and understand for humans and machines alike. A good example of structured data is an Excel spreadsheet with clear requirements for each cell based on the columns and rows.

With its clearly defined schema, structured data is commonly used for financial data, inventory control, and reservation systems. In the SaaS world, user profile information is a common example of where structured data is often used. A user profile typically has information such as a user's name, account number, phone number, and email address. A schema can be defined such that each row is a unique customer and each column represents a different piece of customer data.

The lack of any schema or formatting defines unstructured data. Consequently, unstructured data is more cumbersome and not as easy to work with. Without a schema, it can be hard to process and analyze unstructured data with conventional methods and tools. Common examples of unstructured data include an image, an audio/video recording, a social media

post, or a text file, such as a transcript from a Webex meeting. [Figure 2-23](#) illustrates the difference between structured and unstructured data.

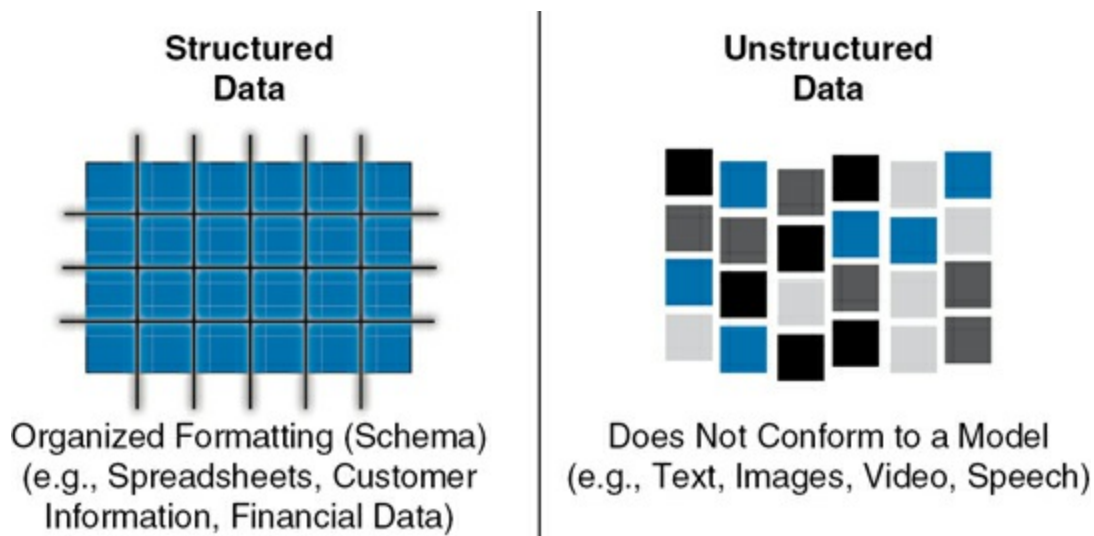


Figure 2-23 Comparison Between Structured and Unstructured Data

Most data being generated today is unstructured data. In fact, projections from multiple analysts estimate that 80 to 90 percent of data is unstructured information. The amount of structured versus unstructured data in SaaS can vary quite a bit. There is a fair amount of structured data in Cisco SaaS applications in the form of user data, configurations, and so on. Unstructured data, however, would be present in a SaaS application, like Webex, in the form of meeting recordings.

Semi-structured data is a hybrid of structured and unstructured, where some of the data may have some predefined formatting, but other parts of it does not. For example, an email has some well-defined, structured header fields, like the To:, From:, and Content-type:. In the From: field, a valid email address and optionally a name define the text structure of that data element. However, in the message body, the text does not follow any sort of model and is completely unstructured. This mixture of data types within an email makes it semi-structured.

Another semi-structured data example is an image with Exchange Image File Format (or EXIF) information. This standardized file format for digital photos attaches structured metadata to a digital image, which is unstructured in terms of the image content. This metadata can include the time and date

the photo was taken, geolocation information, and camera settings.

Note

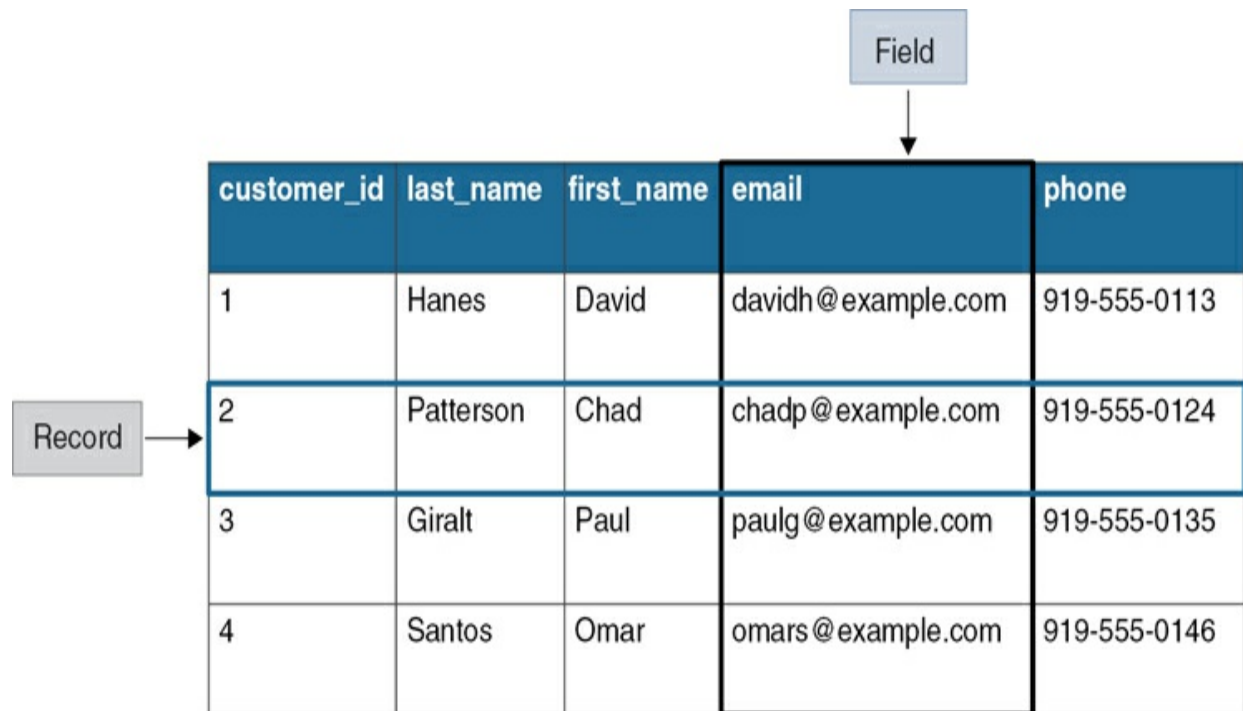
You should be aware that some references do not define semi-structured data as its own data type. Only structured and unstructured data are recognized. Any data that cannot be classified as structured is simply considered unstructured.

SaaS developers have to take into account the types of data that are needed by their application. Knowing these data types helps them define and select the database systems and services that are optimal. CSPs offer multiple database options and integration methods for SaaS applications to leverage. We will discuss some of the most common types in the next section.

Relational and Non-Relational Database Types

There are many different types of databases, but pretty much every one of them falls under one of the following classifications: relational or non-relational. In this section, we will explain these two database types, along with some common database use cases for SaaS applications.

A relational database, as its name suggests, utilizes relationships to connect formatted and defined data across multiple tables. These tables are organized in rows and columns in much the same way as a spreadsheet. Knowing this, you can see the obvious connection between a relational database and the structured data that we discussed previously. Relational databases are a great fit for structured data. In relational databases, though, the rows are referred to as records, and the columns are called fields. [Figure 2-24](#) shows a sample of a relational database table for storing customer contact information.



customer_id	last_name	first_name	email	phone
1	Hanes	David	davidh@example.com	919-555-0113
2	Patterson	Chad	chadp@example.com	919-555-0124
3	Giralt	Paul	paulg@example.com	919-555-0135
4	Santos	Omar	omars@example.com	919-555-0146

Figure 2-24 Relational Database for Storing Customer Information

The power of relational databases occurs as additional tables are built and related or connected to one another. A best practice is that each category of data is given its own table in a relational database. For example, an additional table can be created for customer support tickets. This customer support ticket table could contain data, such as ticket number, product, app version, site, and problem code. A third table could then establish the relationship, as shown in [Figure 2-25](#). This third table could take the customer_id field and associate it with ticket numbers shown by the ticket_id field. Now you can easily look up any customer and all their tickets. This capability gets more powerful with larger data sets and more tables. Continuing with this example, you could add more tables, such as subscriptions, billing information, and customer success–related data. This would lead to more relationships and insights.

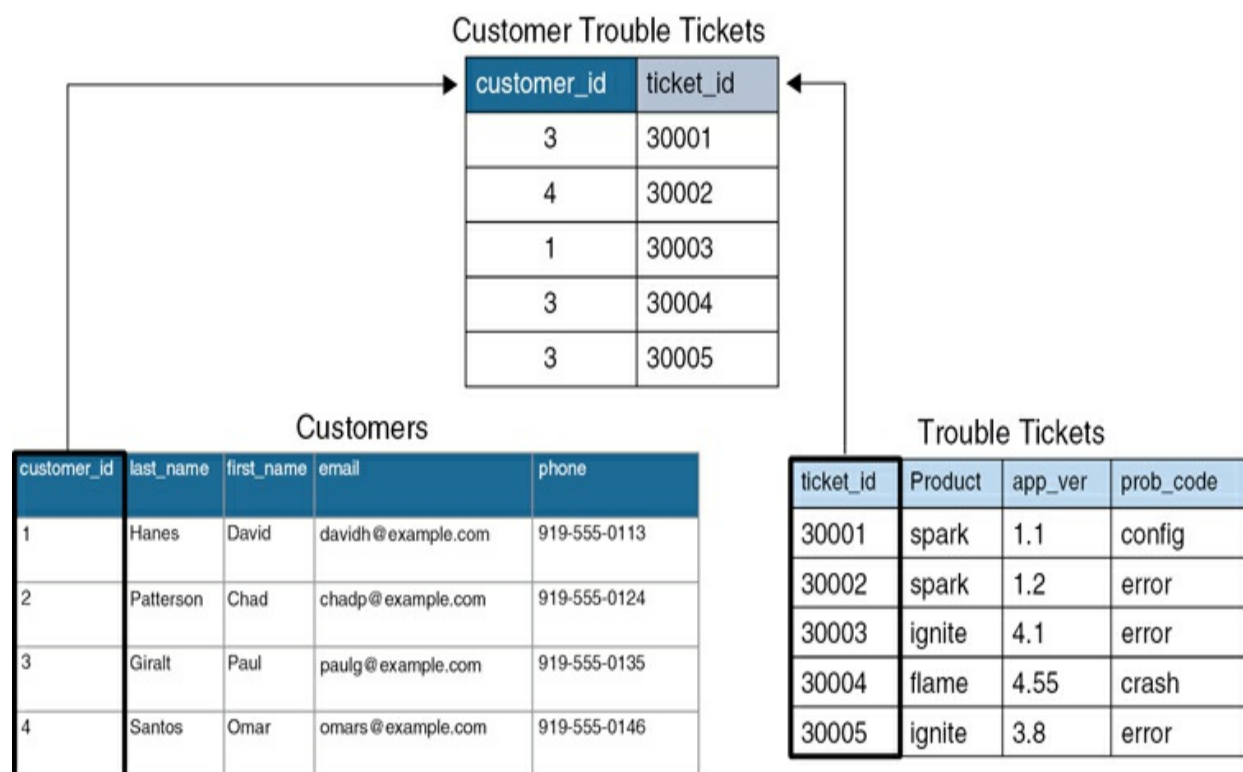


Figure 2-25 Relational Database

Relational databases have a long history dating back to the 1970s when they were first developed along with a standard programming language known as Structured Query Language (SQL). This language allows a database administrator to easily add, delete, and update rows in a table. Because of this history, you will often hear relational databases referred to as *SQL Databases*. Unsurprisingly, non-relational ones are called *NoSQL databases*.

Note

Another term that you will often hear in the context of relational databases is Relational Database Management System (or RDBMS). RDBMS usually refers more specifically to the underlying database software that is being utilized versus the generic concept of a relational database. Common RDBMS software includes Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and IBM DB2.

For data that is unstructured or semi-structured, relational databases are not a good fit. Instead, non-relational database types tend to be better suited. Some

of these non-relational database types include wide-column, key-value, document, graph, and vector databases. [Table 2-5](#) provides an overview of these common non-relational or NoSQL databases.

Table 2-5 Common Non-Relational Databases

Non-Relational Database	Description	Popular Databases
Wide-Column (Column Family) Stores	Contains rows with arbitrary sized columns and uses keys for indexing	Apache Cassandra, HBase, Azure Cosmos DB
Key-value pair	Assigns each data element to a unique key for referencing	Redis, Amazon DynamoDB, Azure Cosmos DB, Memcached
Document	Allows for data storage using hierarchically organized documents	MongoDB, Amazon DynamoDB, Azure Cosmos DB, Databricks, Couchbase
Graph	Built on nodes and edges for identifying entities and complex relationships between them	Neo4j, Amazon Neptune, Azure Cosmos DB, ArangoDB
Vector	Efficiently stores vector embeddings used by ML/AI models	Pinecone, Chroma, Weaviate, Milvus, Qdrant

In addition to structured data almost always being better for relational databases, SaaS application developers must take other factors into account when selecting between a relational or non-relational database. One of these factors is the amount of data that needs to be stored. Relational databases are better for small to medium dataset sizes. Non-relational databases scale to much larger datasets. Another factor is who will access and maintain the database. Relational databases are easier to understand and maintain, whereas non-relational ones usually are more complicated and require training.

Note

Relational databases and non-relational ones, like those highlighted

in [Table 2-5](#), are often provided by CSPs as part of their Database as a Service (DBaaS) offerings. SaaS providers can leverage DBaaS from a CSP to expedite and quickly scale their applications compared to building a database from scratch.

Wide-column non-relational databases are similar to the tabular format of relational databases previously discussed in this section. The difference is that the smallest building block is a two-row column consisting of a name and a value. Columns of related data can then be grouped beside one another to create larger data storage structures in a row formation. For example, a wide-column database could also be used to store customer information. Name-value columns could be created for customer first name, last name, email, phone number, and so on. These personal details can be grouped together alongside other groupings for subscriptions, billing, and so on in what is called a super column family.

Wide-column databases have high availability and scalability. This means they can handle many queries efficiently across a large dataset, including a high number of database writes and minimal latency. Common SaaS use cases for wide-column databases include online gaming, ecommerce, and cloud analytics. [Figure 2-26](#) provides a graphical overview of a wide-column database compared with key-value, graph, document, and vector non-relational databases.

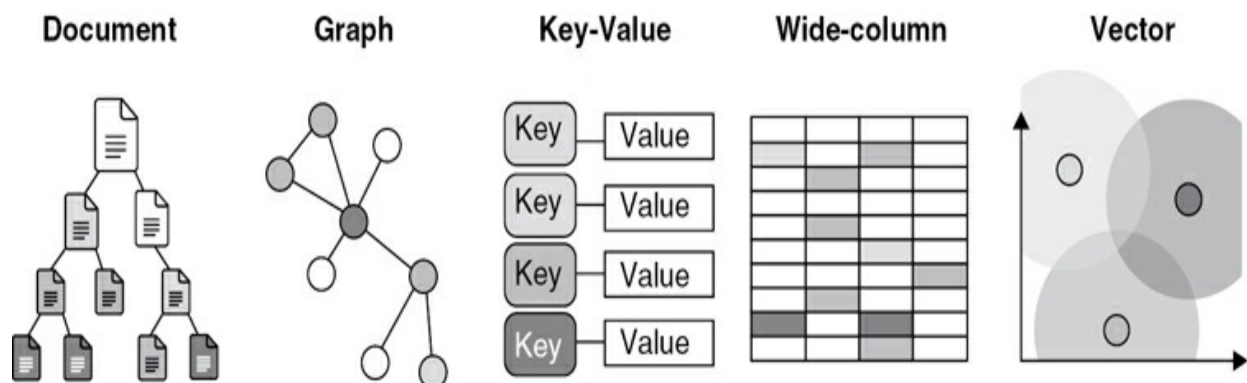


Figure 2-26 High-Level Comparison of Document, Graph, Key-Value, and Wide-Column Non-Relational Databases

A key-value pair database is probably the simplest non-relational database structure. It simply is made up of a numerical key that is associated with a

value. Values can be pretty much anything like numbers, text, or more complex objects. In addition to high availability and scalability, these types of databases are efficient at data storage and retrieval due to their simplicity. Consequently, key-value databases are good for SaaS applications that require larger datasets and need frequent reads and writes. On the other hand, these databases are not great at extracting complex relationships or insights from data.

Just as you would probably guess, document databases store data as a series of documents. A document is simply an object identified by an ID that encodes its data in some sort of standardized form, like JSON. Similar documents can be grouped as a collection. Using customer information as an example again, a single document could contain first name, last name, email, phone, and so on as JSON and be quickly retrieved using the unique document object ID.

Document databases have a flexible schema. This makes them efficient for applications to use and therefore deliver high performance. They are also very scalable. Document databases are often used for functions like product catalogs, user profiles, and content management.

Graph databases are a little different than the other non-relational database types covered in this section. They are composed of nodes and edges between the nodes. A node is simply a piece of data, like a person, place, thing, or any other piece of data. Nodes are then connected by an edge or relationship. The way X (formerly Twitter) users follow one another is probably the easiest example of how a graph database works. Each user is a node, and a “follow” is the relationship link or edge between the nodes, as illustrated in [Figure 2-27](#).

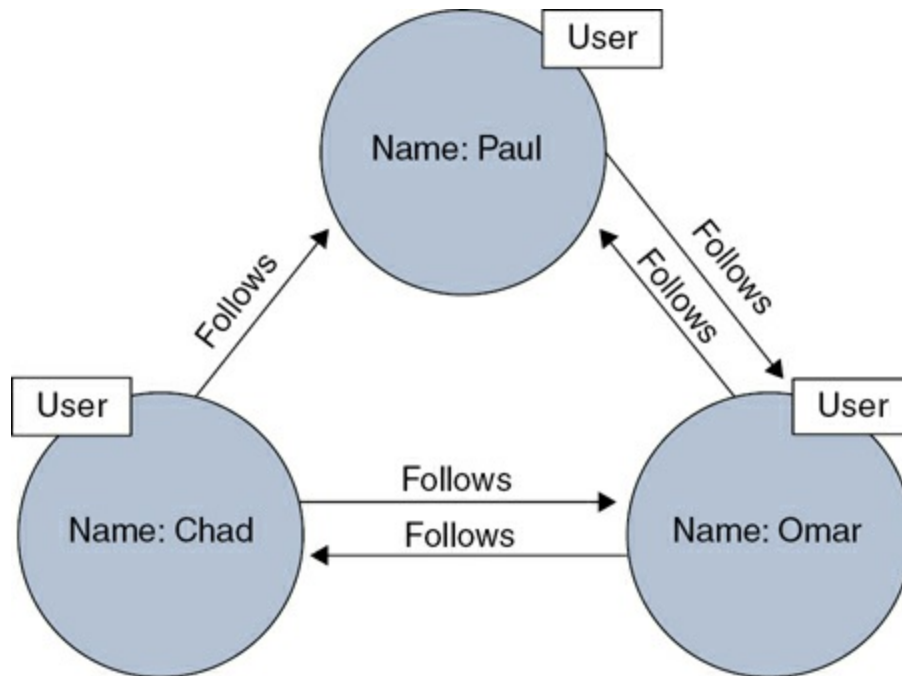


Figure 2-27 Graph Database

As you can see in [Figure 2-27](#), graph databases focus on the relationships between the data. These relationship connections are persistent and easily accessible by applications connected to the database. Therefore, graph databases are great for data sets where an application wants high-performance access to data relationships that are often complex. You can find graph databases used in applications, such as social media and financial fraud detection, but they are not as strong when it comes to transactional data, like billing.

The last non-relational database type in [Table 2-5](#) is vector-based. These databases are built around vector embeddings for a data object. This data object could be text, an image, a video, or audio. An embedding model takes this data object and generates a vector embedding for that object. This vector embedding is simply a series of numerical values aligning to properties and characteristics for that data object. Vector databases can use these embedded vectors that represent data objects and provide correlation and similarities between them. This capability allows for quick search and retrieval of relevant data objects. [Figure 2-28](#) shows a high-level overview of how a vector database works.

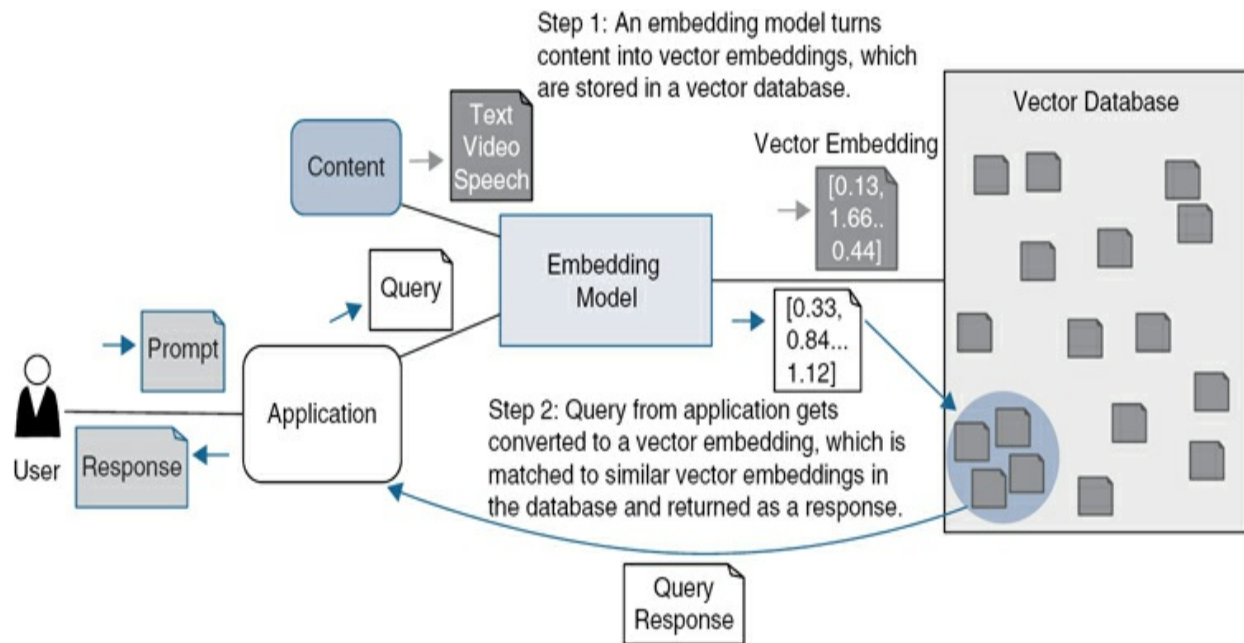


Figure 2-28 Vector Database Overview

For example, a vector database would be efficient for storing data about movies and then helping make recommendations. As a first step, you would use an embedding model to create vector embeddings for a catalog of movies. This model could consist of just a few values or many, depending on the amount of detail you want to capture. You could have a vector for movie genre, starring actors, settings, languages spoken, and so on. High numerical values for action genre, English language, and an urban setting would mean that a movie was an English-speaking action film that mostly takes place in a city. This is a simple example, but you can extrapolate this out to many vectors to better understand the significance of this type of database.

Each movie that you run through the embedding model would end up with numerical scores across these vectors. This would be the vector embedding for that movie, and this information is what is stored in a vector database. Assuming your favorite movie was not already in the database, you could submit it as a query, as shown in Step 2 in [Figure 2-28](#). The embedding model would then create vector embeddings for your favorite movie and search the database and find the most similar movies to your favorite one. It does this by simply comparing the numerical vector embedding values of your favorite movie to the vector embeddings of the other movies in the database to find those that are the closest match.

You find vector databases frequently utilized with AI applications, like large language models (LLMs) and recommendation engines, because of their ability to store and quickly access more complex data similarities and associations. As SaaS application providers integrate more AI into their solutions, vector database usage will continue to increase.

Note

You will sometimes hear the terms *data lake* and *data warehouse* in database-related discussions. The main differences between these two terms revolve around the types of data they are designed for storing and their schemas. Data warehouses have more predefined structure than data lakes. Data ingested by a data warehouse has a schema applied to it, and this makes data warehouses ideal for structured data and sometimes unstructured data. Data lakes do not apply a schema at ingestion, so any sort of raw data type (structured, semi-structured, and unstructured) can be easily stored in a data lake. For analytics involving current and historical data from multiple sources, data lakes and data warehouses can be a good choice, but they are not good for interactive applications. SaaS applications, however, are almost always interactive and need good performance and optimization for operational and transactional workloads. For this reason, SaaS applications utilize databases most of the time.

At this point, we have provided an overview of some of the common databases used in SaaS applications. By no means is this an exhaustive list, and there are other database types. Additionally, you should keep in mind that a SaaS application rarely utilizes a single database for all its storage needs. As depicted in [Figure 2-17](#), each microservice often has its own database. Additionally, different database types are better for certain types of data, and SaaS applications providers take this factor into account when it comes to database selection. Utilizing the right database is critical for optimizing SaaS application performance.

Presentation Services

The Presentation Services block is most likely the easiest part of the SaaS

Architectural Model to grasp and understand. After all, presentation services encompass the user interface for the SaaS application and how you interact, configure, and administer it. The UI is quite impressionable and probably what you visualize first when you hear the name of a SaaS application. For example, if you think about Webex Meetings, you might picture the main meeting screen of the application, where you can see the video feeds or avatars of the participants and maybe even content being shared.

If you are familiar with the concept of front-end software development, this is essentially what comprises Presentation Services. The most common definition of the frontend is the part of the SaaS application that a user sees. This description is true, but the meaning is also a bit broader than this. The frontend includes the programming code, image elements, and the connections to the backend that allow the user to have a good experience as well. One way to look at the frontend, or Presentation Services, itself is to divide it into the access methods and the programming, as depicted in [Figure 2-29](#).

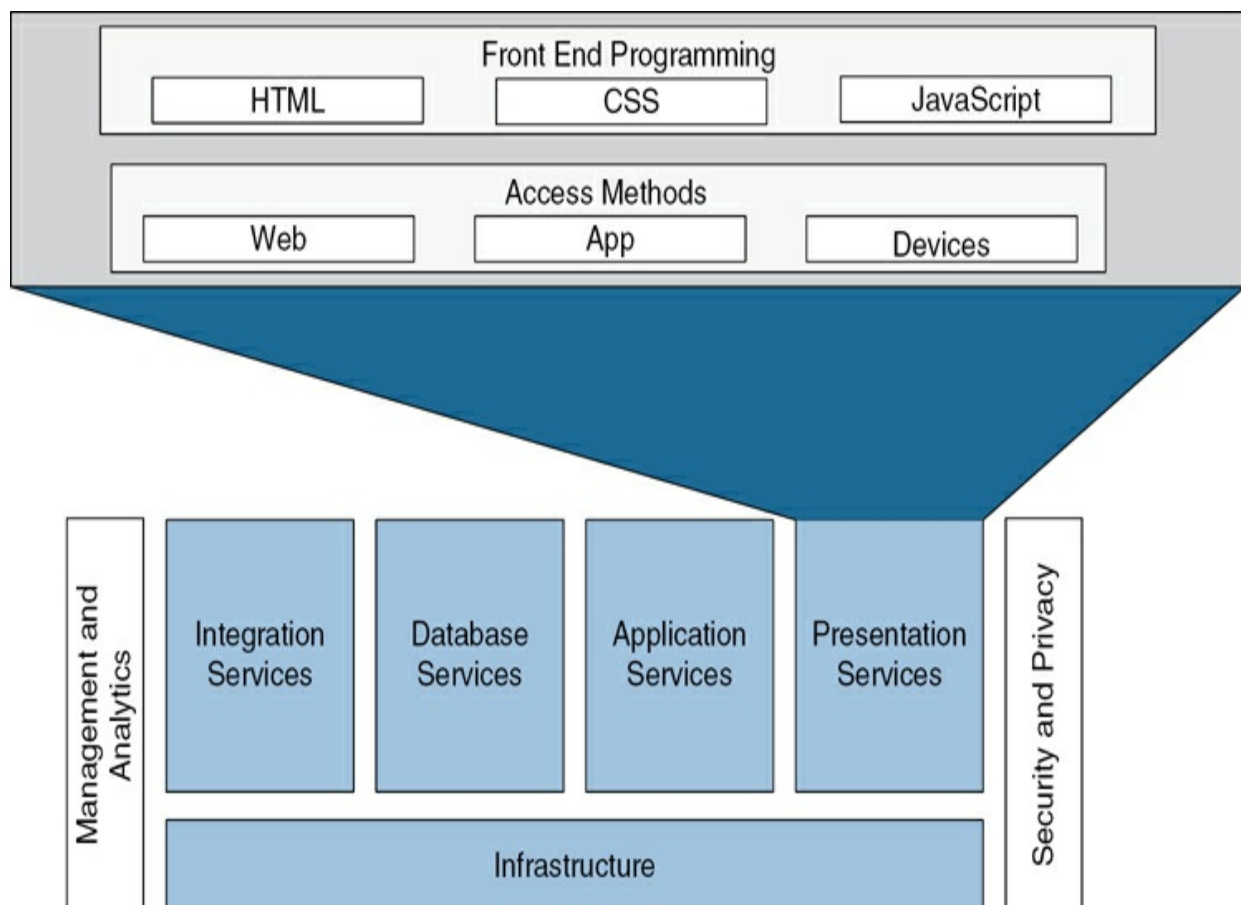


Figure 2-29 Presentation Services Block of the SaaS Architectural Model

The back-end portion is the logic and coding that happen in the Application Services, Database Services, and Integration Services blocks of the SaaS Architectural Model. The frontend accesses these back-end services to facilitate accurate and efficient interactions between the user and the SaaS application. [Figure 2-30](#) shows a basic front-end and back-end architecture for a SaaS application. It also includes the relevant blocks from the SaaS Architectural Model for context.

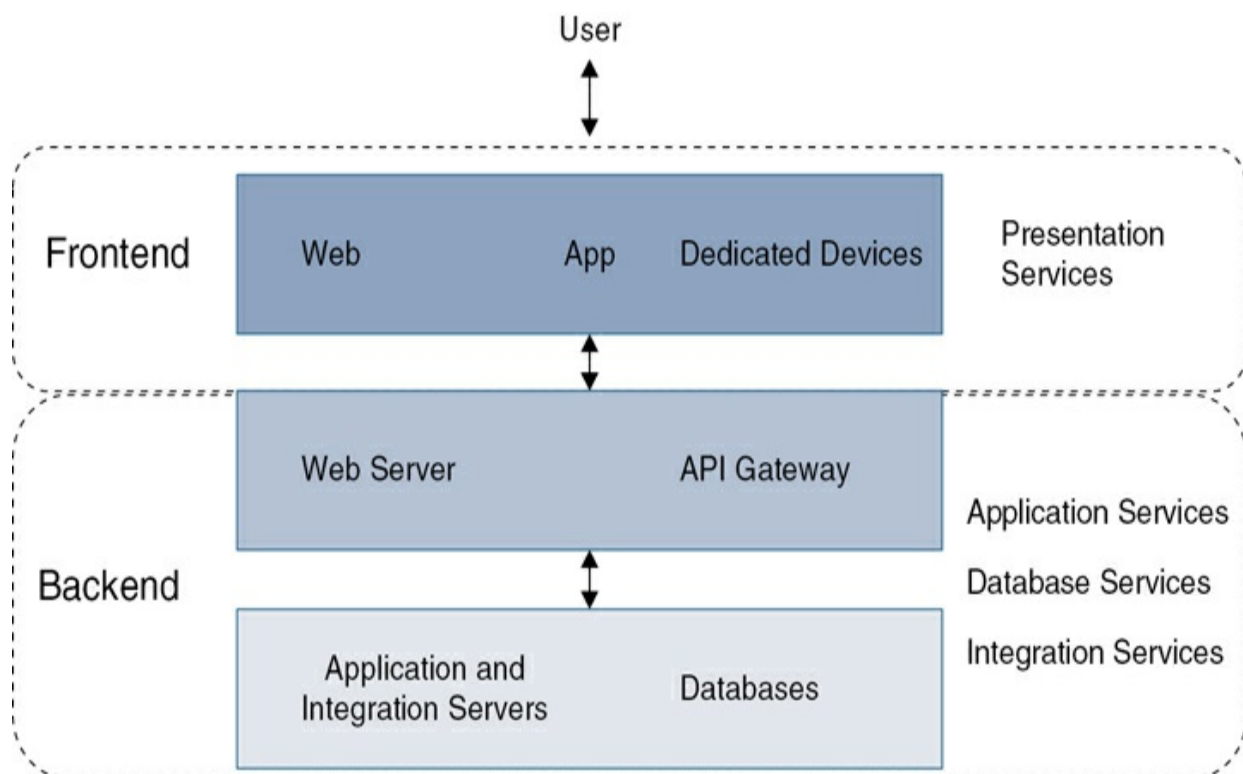


Figure 2-30 Basic Front-End and Back-End Integration for a SaaS Application

You can access a SaaS application using different methods. The most common method is the web. You can open a web browser and enter the URL associated with the SaaS application. This method can be used by any device with a web browser, including laptop and desktop computers, tablets, and smart phones.

Some SaaS solutions have a user application that can be downloaded to computers, tablets, and smart phones. This app is built by the SaaS

application provider and allows for quick and direct access and can be used instead of a web browser. The last access method is the least common and is a dedicated device. A dedicated device is a piece of hardware that runs software from the SaaS application provider.

Webex Meetings is a Cisco SaaS application that best illustrates all these access methods. You can attend a meeting using just a web browser. You can join a meeting using the Webex app on a computer or mobile device. Last of all, you can attend a meeting using a dedicated device like a Webex Desk Pro. Most SaaS applications do not have all these options. At a minimum, the web access method is almost always available because this is the main way that SaaS applications are administered and configured.

Web browsers, apps, and dedicated devices must connect to back-end services to access SaaS application functionality as well as administration. These connections can happen in different ways. The most common method is the web server. On the back-end side, the web server sends web pages and related content for display to the user. The web server communicates with other back-end services using connections, like APIs. For SaaS apps running on smart phones or PCs and dedicated devices, SaaS application providers typically use an API gateway to interconnect their client apps with the necessary back-end resources. This approach allows apps and dedicated devices to efficiently retrieve real-time data and deliver the expected functionality from the SaaS application to the end user.

Note

In most cases, when you access a SaaS application or service using your web browser, the core app functions and the administration occur through the same website. Administrators and other user roles or levels defined by the application are given more options and capabilities than a general user. However, in some cases, higher-level administration and configuration can occur through a different website altogether. For example, Webex Meetings provides general user access at <https://www.webex.com>. However, administration of its sites and organizations occurs through Control Hub (<https://admin.webex.com>).

Note

To assist with the scaling of their application and to ensure good user performance, SaaS application providers use content delivery networks (CDNs). CDNs leverage caching techniques and allow for the serving of content from edge servers close to a user to provide better performance. Dedicated CDN businesses and CSPs offer CDN services.

Taking a deeper look into the web part of Presentation Services because this is common across all Cisco SaaS applications, certain protocols and frameworks are associated with a frontend versus a backend. We covered back-end protocols and frameworks previously in the “[Application Services](#)” section. On the web front-end side, you see mostly HTML, CSS, and JavaScript protocols. These protocols are considered cornerstones of front-end development and the web itself.

Hypertext Markup Language (HTML) is the foundational coding for how web pages are displayed on a browser page by defining the page structure and content. Because HTML does not make use of variables and functions, it is not considered a programming language but a markup language. For example, it can divide pages into sections with various types of content. This content can be easily resized and styled for various looks and customizations. HTML also integrates nicely with CSS and JavaScript to further enhance web pages for users.

Cascading style sheets (CSS) seek to bring more flexibility and efficiency to web pages. This style sheet language accomplishes this goal by defining page elements that are reusable throughout a website and can store these attributes in a shared, external file. Without CSS, every web page must define the same style elements, such as font sizes, colors, spacing, alignment, border sizes, and colors. Defining all these elements makes web pages larger and longer to load while making it difficult to keep styles across all the pages synchronized, especially if changes need to be made. Plus, CSS style rules can be triggered by inputs like screen size and resolution so that content can easily adapt to a device. CSS allows you to easily apply a unique look and even change it across all the pages of a website while ensuring a good browsing experience using any device.

JavaScript (JS) is a client-side scripting language that enables dynamic

content in web pages and applications. *Client side* in this case refers to the end device with the web browser that is talking to the web server. JavaScript is embedded as part of the web page and runs on the client endpoint, not on the server itself in most cases.

Because JavaScript allows you to execute code on the client, it enables web page functions that you have come to expect. In fact, it is estimated that 99 percent of websites utilize client-side JavaScript. One JavaScript function is the ability to bring in new page content without reloading a page. For example, you can launch a chat application for support without leaving the current page. Other functions include browser games or the playback controls associated with streaming media.

JavaScript has continued to evolve and gain momentum as web pages continue to increase their utilization of dynamic content to drive user interaction and engagement. This evolution has driven widespread growth and availability in the various libraries and web frameworks for JS development. You can think of libraries as prebuilt code functions that a developer can call from their code to speed up programming. These libraries provide for a lot of customization, but the cost is often more complex code that is harder to maintain. On the other hand, a framework provides more of a structure or skeleton with tools for rapid development. Frameworks use libraries as well, but the framework determines when a call to a library is needed and may enforce certain coding practices for stability and maintainability. [Table 2-6](#) highlights some of the most common JS libraries and web frameworks for reference. Capabilities and functions can vary somewhat between these libraries and frameworks, and often developer familiarity and preference determine which is chosen for JavaScript development.

Table 2-6 Common JavaScript Libraries and Frameworks

JS Library/Framework	Description
JQuery	JQuery is the most used JS library, and it is free and open source. It is known for being lightweight, easy to work with, and a good choice for simpler web projects. Ensuring compatibility with most browsers, even legacy ones, is another JQuery strength.
React	Backed by Meta, React is another open-source JS library for UI development. It is better for larger and more complicated web projects compared to JQuery, where its component-based architecture is great for lots of dynamic content and high user interaction.
Angular	Led by Google, Angular is a full-fledged and mature framework that has a steep learning curve compared to JS libraries and a lot of other frameworks. However, this open-source software can provide a complete website development experience without additional libraries.
Vue	Vue is an open-source framework that is considered a lightweight alternative to Angular. It has good performance and flexibility along with an easier learning curve.

The Presentation Services block of the SaaS Architectural Model is the most visible portion of a SaaS application. It is what you, as a user, interact with and utilize for administration and configuration. You can think of Presentation Services as the sleek exterior that hides the complications and internal plumbing of the other blocks of the architectural model.

Integration Services

From the Add Webex Meeting button that appears when you are scheduling a meeting using Microsoft Outlook to how ThousandEyes data can trigger an alert in your Splunk On-Call application to your AppDynamics data automatically populating your Splunk dashboard, integrating a SaaS application with other applications is powerful. The block in the SaaS Architectural Model that enables these cross-product features is Integration Services.

Integration Services refers to the capabilities and functions that enable a SaaS

application to integrate, communicate, and easily share data with other products and solutions. These integrations are critical for a seamless integration with existing systems and making SaaS applications more efficient. Although these integrations are often with other SaaS products, this does not necessarily have to be the case. A SaaS application can also connect and integrate with a custom piece of software onsite. Multiple SaaS applications can be integrated with one another to enable complex, automated workflows.

Figure 2-31 shows more details on what is included in the Integration Services block of the SaaS Architectural Model. On the bottom is Custom integrations. Custom integration methods that are often provided by the SaaS application include APIs, webhooks, and WebSockets. With these methods, you can get the exact integration and function that you desire, but you must build it yourself.

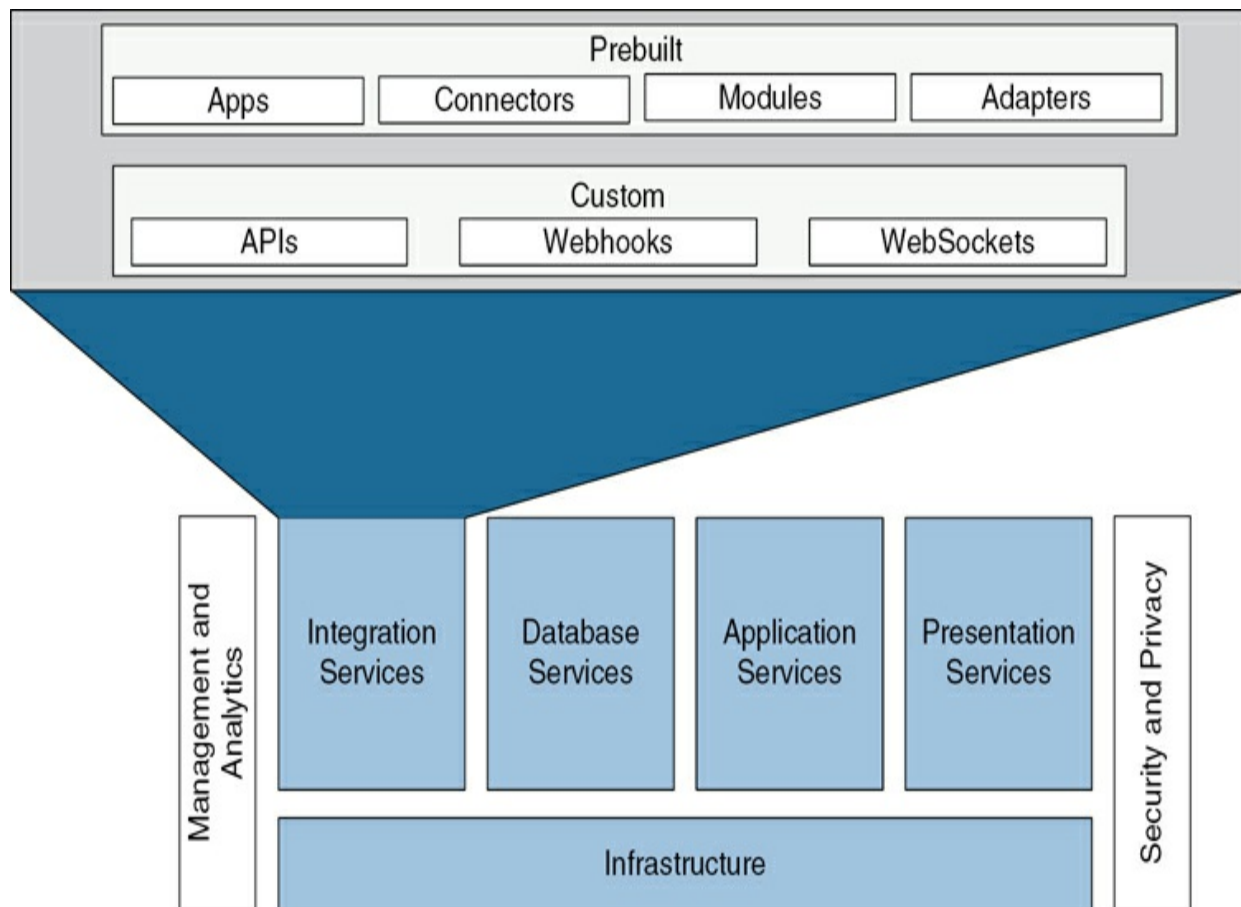


Figure 2-31 Integration Services Block of the SaaS Architectural Model

The upper group of integrations methods in the figure is labeled *prebuilt*. With a prebuilt integration method, the SaaS application or sometimes a third party provides an integration for you to use. These prebuilt integrations can be referred to by a few names, including an application or app, connector, module, or adapter. With one of these prebuilt integrations, you may lose some functionality and flexibility, but integration is much simpler. Whether it is a prebuilt or custom integration method, most SaaS solutions almost always support at least one of these for connecting with other applications.

Custom Integrations with APIs, Webhooks, and WebSockets

Although there are other methods, most custom integrations with SaaS applications occur through APIs, webhooks, and WebSockets. When you're working with these or any custom integrations, coding and development experience is almost always necessary. The benefit, however, is that you can get exactly what you need from an integration perspective. A custom integration may be your only option if a prebuilt integration does not have the capability or function that you need.

As discussed previously, application programming interfaces can be an important part of microservices architectures. They are also important when it comes to connecting to other applications and services externally and are the most well-known custom integration type. An API typically utilizes a client/server architecture, where the client sends an API request to the server that responds with the requested information. The client and server are the “application” in API, and this is simply a piece of software with a specific function. You should think of an API as a communication channel between applications or software functions.

Various types of APIs exist, but the REST API type is considered the most popular and flexible one. REST APIs are characterized by a client/server model using HTTP, the same protocol used for web browser communications. Clients use simple functional commands like **GET** and **PUT** to access data on the server. Additionally, it is important to note that REST is stateless. This means that servers do not save data or “remember” previous requests from a client. Each transaction is independent, and therefore, all the necessary information for processing the request must be included. This design results in good performance and flexibility but can be a

disadvantage in terms of real-time interactions or complex workflows.

An example of a SaaS API integration is pulling data from the Cisco ThousandEyes application and displaying it on a custom dashboard. ThousandEyes has a fairly extensive API catalog that is documented at <https://developer.cisco.com/docs/thousandeyes/>.

Note

If you are not familiar with Cisco ThousandEyes, this SaaS application can provide end-to-end network visibility to problems. It utilizes agents that you can load into your network gear and PC devices or servers, along with public agents on the Internet to gather insights and data. ThousandEyes is covered in [Chapter 12](#), “[Observability and Monitoring: Cisco ThousandEyes](#).”

Python or another coding language can act as the client and send API requests to ThousandEyes, acting as the server, to pull historical data from tests that are monitoring critical network applications or cloud providers. When the Python script receives this data, it can be stored in a database, and then a dashboard can pull from that database and display the data on a web page. [Figure 2-32](#) shows this flow.

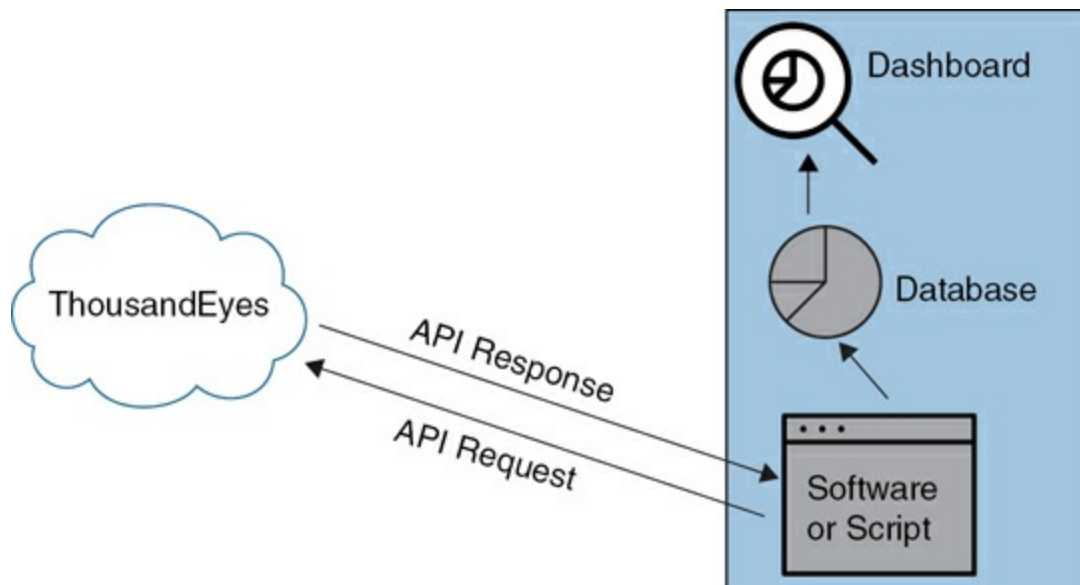


Figure 2-32 Using a ThousandEyes API to Populate a Dashboard

As mentioned previously, software development knowledge will be needed to build out the custom API implementation in [Figure 2-32](#). Because of this, the integration has more complexities up front, but the benefit is that you can choose the exact data you want to pull and how you want to use it.

Webhooks and WebSockets are the two other custom integration types highlighted in [Figure 2-31](#). Although you might sometimes see them grouped together because they allow for the pushing of data, they do have some key differences as well. Let's look at webhooks first. You will find that they are HTTP based, just like the REST API connection method that we just discussed. However, an API depends on a client periodically reaching out and asking for data. A webhook takes the opposite approach and is configured to send data when a predefined event or trigger occurs.

Continuing with the ThousandEyes API example, what happens if you replace the API connection with a webhook? Recall that with an API, a piece of software must send API requests to initiate a data transfer from ThousandEyes. This approach is fine for pulling historical data, but what if you want your software to be notified as soon as ThousandEyes detects a problem in your network? This is where webhooks are a much better fit.

With webhooks, ThousandEyes can be configured so that when a test failure occurs, a notification is triggered to your software. Instead of your software periodically polling an API, a webhook pushes the requested data to your software as soon as it happens. With this sort of use case, your software could then open a trouble ticket and send this to a Splunk dashboard as a real-time update. [Figure 2-33](#) shows this use case.

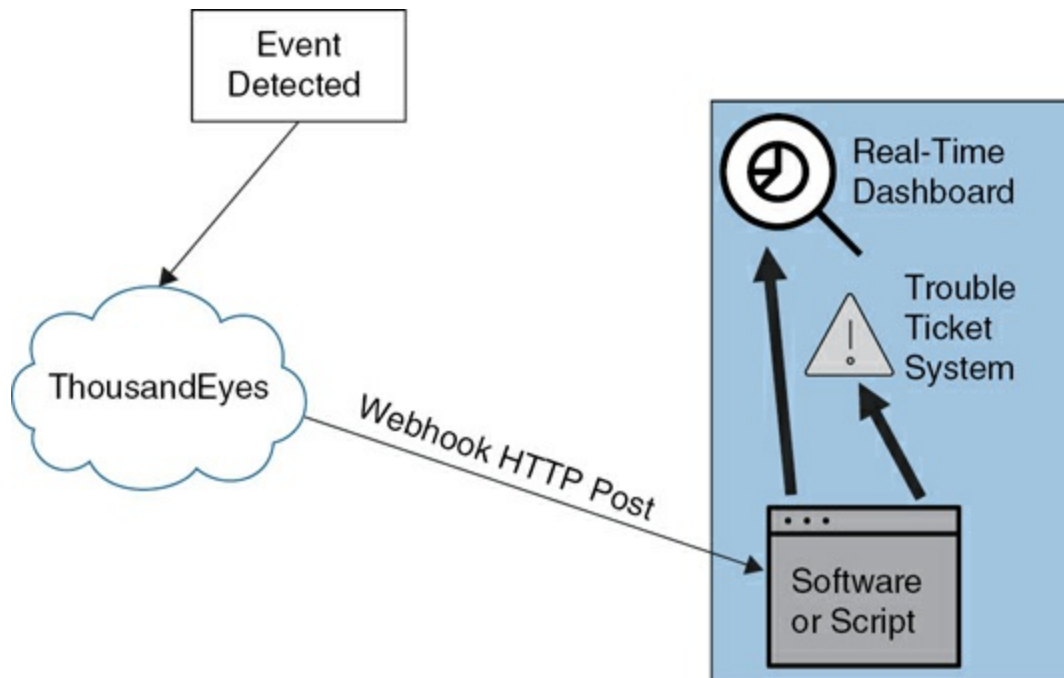


Figure 2-33 Using a ThousandEyes Webhook

A WebSocket also allows for a push notification, like a webhook. However, there are some differences. First, the connection between the two systems with a WebSocket is persistent. A WebSocket session remains established after it has been set up, which is helpful for scalable real-time communications. With a webhook, a session is created only when a message needs to be sent and is then torn down until the next event. Second, the webhook is a one-way communication, whereas a WebSocket is bidirectional once established. Last, WebSockets can be seen as a more secure implementation. With WebSockets, the connection is typically initiated from inside your network out to a resource, whereas a webhook usually involves an outside resource initiating a connection into your network.

Common use cases for WebSockets include real-time applications, like multiplayer games and live video/audio streaming apps. Collaborative document sharing is another use case. Because of its persistent, bidirectional connection, WebSockets are a good choice for any sort of continuous data exchange.

In the Cisco SaaS product space, Cisco Webex is a good example illustrating where a WebSocket integration is offered and is quite useful. Webex allows you to create a chatbot from its developer portal that you can connect to your

own application. The result is that your users can connect to this bot from the Webex app and have a “conversation” with your application.

Prebuilt Integrations with Apps, Connectors, Modules, and Adapters

SaaS application providers usually provide a prebuilt integration to ease the integration and connection of their solutions with other applications. These integrations still utilize various connections like the APIs, webhooks, and WebSockets that we just covered, but the software programming to leverage them has been handled by the SaaS application provider or a third party.

Good examples of prebuilt integration apps can be found at the Cisco Webex App Hub (see [Figure 2-34](#)). The Webex App Hub offers a good selection of downloadable apps that you can install to extend the capabilities and functionality of your Webex application. Some of these apps are built by Cisco, and others are built by other application providers. To learn more, you can access the Webex App Hub at <https://apphub.webex.com>.

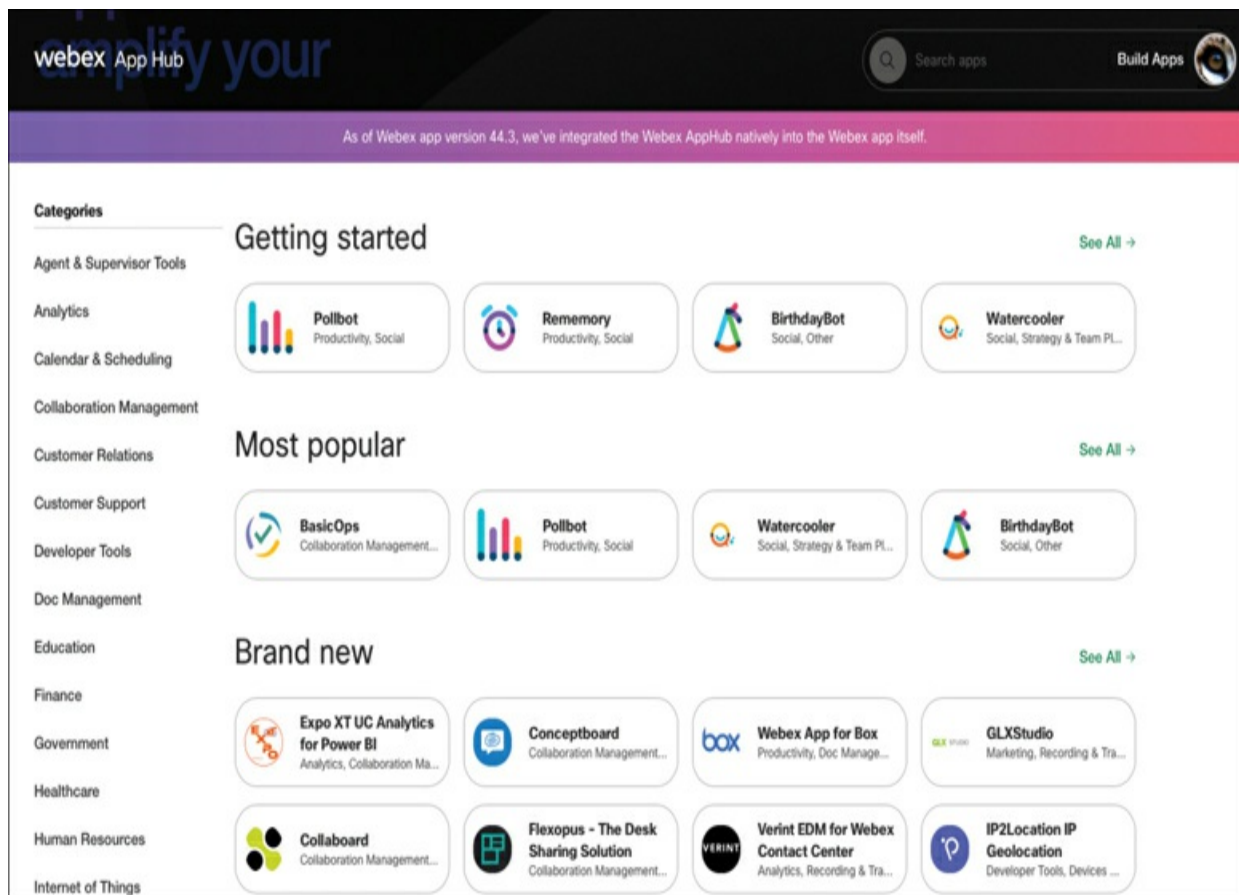


Figure 2-34 Cisco Webex App Hub

Some apps in Webex App Hub introduce more functionality into the product, such as polling, whiteboards, or time management features. These types of apps may require you to download and configure them, or Webex simply adds and enables them for instant usage. Some Webex apps are more focused on integration with other products and may just be referred to as *integrations*. The integration-related apps have third-party dependencies and almost always require that you have an account on the application that you want to integrate with Webex. These integrations are functionally the same as connectors and adapters.

Another example of a prebuilt integration is the Cisco Duo Splunk Connector. This connector, built by Cisco Duo, can be downloaded from Splunkbase, the Splunk equivalent of the Webex App Hub shown in [Figure 2-34](#). If you want to explore and learn more about Splunkbase, you can access it at <https://splunkbase.splunk.com>.

After you have downloaded the Duo Splunk Connector from Splunkbase and installed it in your Splunk deployment, you can then configure it directly from your Splunk app. With this connector configured, you can easily import Duo logs right into your Splunk environment. Once again, you can see the ease of downloading and entering configuration and credentials in a user interface versus coding this same integration entirely on your own.

You probably noticed that the terms *app*, *module*, *connector*, and *adapter* are conceptually the same. They all refer to some sort of prebuilt integration software that must be downloaded or added to an application or, in some cases, is part of the application natively and just needs to be configured. You will often see these terms used interchangeably when looking at various SaaS application integrations.

The main takeaway from this discussion on custom and prebuilt integrations is that you start with prebuilt integrations. They are faster and easier, and if they meet your needs, they are the obvious choice. If a prebuilt integration is not available or an existing one does not meet your needs, a custom integration and the development that goes along with it are your only option.

Security and Privacy

Security and privacy are always top of mind in any sort of networking or application solution. With SaaS applications, they are often even more important because you are trusting that the SaaS application provider is incorporating the proper tools and procedures to protect and secure your data. You are giving a high level of trust to any SaaS application provider you utilize.

Note

The terms *security* and *privacy* are often found together and even used interchangeably at times, but they are different. Security refers to the protection of data in general, including personal data. Privacy focuses on the control of personal data and how that information is used or shared. Another way to look at these concepts is that security protects your data from malicious threats while privacy is concerned with using personal data responsibly.

So, how do you know that your SaaS application is properly secured and privacy is being maintained? To be honest, it can be difficult. As with most other aspects of SaaS, knowing exactly what is happening behind the scenes can be challenging. However, if you know some of the core elements of a secure SaaS solution, you will at least be able to better understand and discuss how your data is being secured. [Figure 2-35](#) breaks down the Security and Privacy block from the SaaS Architectural Model into three categories: cloud security controls, identity management, and visibility and monitoring.

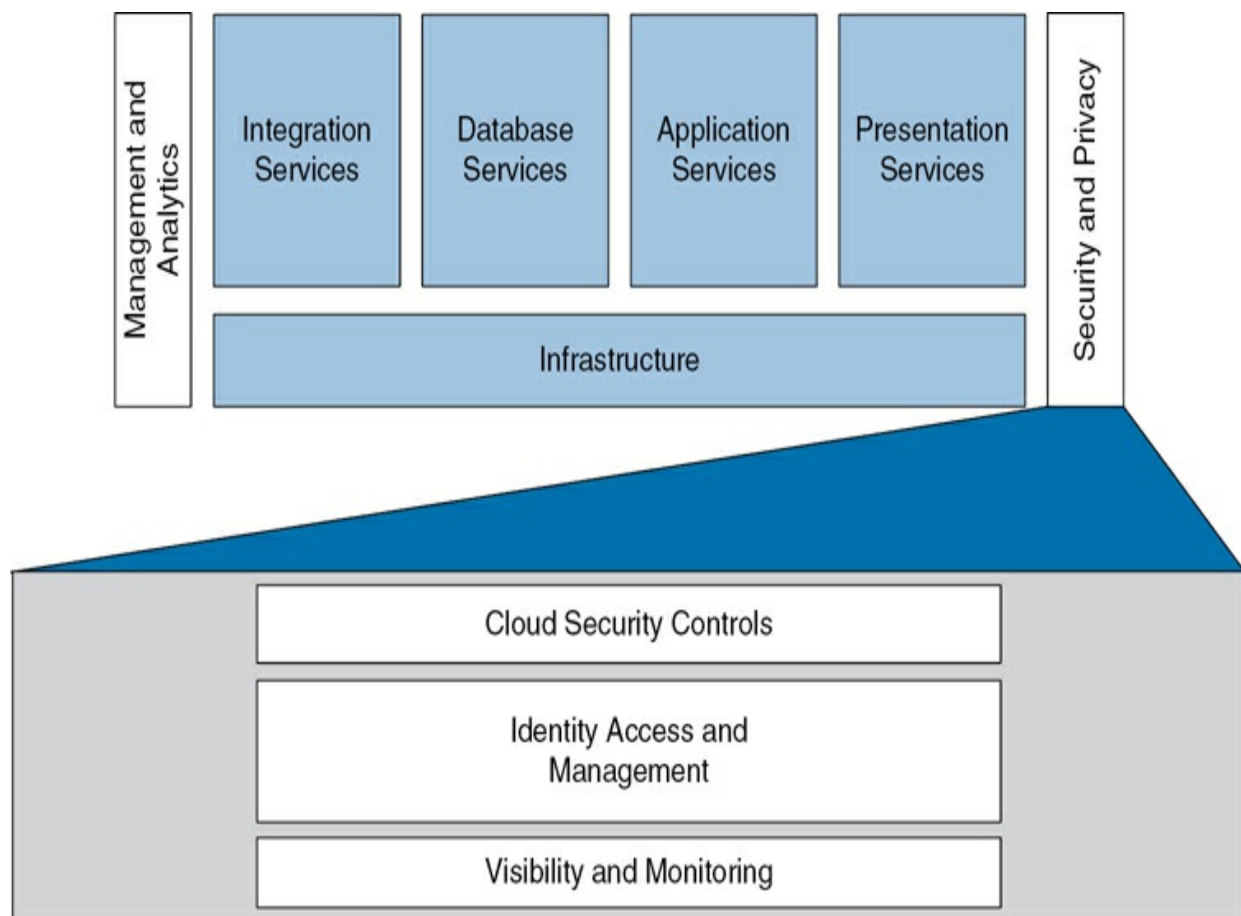


Figure 2-35 Security and Privacy Block of the SaaS Architectural Model

In the following sections, we will cover each of these categories. SaaS security and privacy are a broad area and critical in today's world of continuous threats and bad actors looking to exploit and access cloud data. For these reasons, [Chapter 4, "Security and Privacy for SaaS,"](#) is dedicated to SaaS security and privacy. Therefore, the coverage of the Security and Privacy block in this chapter will be more introductory, and you should

reference [Chapter 4](#) for a more in-depth discussion.

Cloud Security Controls

The first category in [Figure 2-35](#), cloud security controls, defines various best practices, processes, and technologies that organizations can utilize to protect their cloud environment, including cloud services like SaaS applications. With cloud security controls, organizations can better recognize threats, minimize data breaches, and reduce security incidents related to misconfiguration and human error.

To better understand cloud security controls, you will often see them divided into the following types: deterrent, preventive, corrective, and detective. All these types are part of an effective cloud security strategy. [Table 2-7](#) provides an overview of each of these classifications.

Table 2-7 Types and Examples of Cloud Security Controls

Cloud Security Control Type	Description	Example
Deterrent	Warnings that are put in place to make bad actors think twice before attacking.	Conducting criminal background checks on employees.
Preventive	Vulnerabilities that are removed or fixed to make attacks more difficult and prevent them from ever occurring.	Deactivating ports automatically that have no activity or deploying an identity and access management (IAM) system to enforce read-only rights for a user group.
Detective	Identifications, logs, and alerts about security events, risks, and attacks. You can look at these as a second line of defense for issues that get by the preventive controls.	Using intrusion detection software or putting a notification in place that alerts you when a cloud database has been made publicly accessible.
Corrective	Actions and responses that are activated in the event of a security incident to help reduce the damage of an attack and drive remediation.	Automatically disconnecting cloud storage servers when a specific threat is detected, shutting down a process, or rebooting a system.

Cloud security controls in a SaaS environment are mainly handled by the SaaS application provider. This is part of the SaaS service that you are paying for. However, this does not mean that you, as a SaaS application user or administrator, do not have to worry about security and privacy.

While this point may be obvious, it is important: You must ensure that the proper cloud security controls supported by your SaaS application provider are enabled and configured properly. Additionally, you have responsibility for security and privacy for your users outside the cloud and as data transits to and from the SaaS service. This security partnership between you and the SaaS application provider is referred to as a shared responsibility model. These models were covered in the “[Shared Responsibility Model](#)” section in [Chapter 1](#), “[What Is SaaS?](#)” The exact details of these models are typically specific to cloud service providers and even SaaS application providers. So, you are encouraged to check with the appropriate CSP or application provider if you want more information on a specific model.

Identity Access and Management

Identity access and management handles how users access cloud resources and what they are allowed to do with those resources. IAM keeps the bad actors out of a SaaS application and ensures that only authorized users have access. Additionally, it ensures that authorized users are contained and not allowed to do more than what their level of access grants them.

Note

The principle of least privilege (PoLP) in IAM mandates granting only the minimum level of access necessary to perform a specific task. By limiting permissions strictly to what is required—and nothing more—PoLP reduces the risk of unauthorized access or accidental misuse of resources and sensitive data. This approach minimizes the attack surface and limits potential damage from security breaches or vulnerabilities.

If you’re thinking that IAM is a subset of the preventive type of cloud security controls, you would be correct. In fact, when it comes to SaaS security, IAM is one of the most important security controls, and this is why we’ve highlighted it in this separate section. At a high level, IAM can be

broken down into a few core functions. [Table 2-8](#) overviews these functions.

Table 2-8 IAM Functions

IAM Function	Description
Identity Lifecycle Management	Creates, maintains, and monitors the profiles of users on a system. This includes the onboarding and offboarding of users and validating their access rights.
Access Control	Provides the ability to group users and set and enforce granular access policies. This capability allows admins to have more access than another group of users who may only have read-only access.
Authentication and Authorization	Determines that a user is who they say they are and then grants permission once credentials are verified. Technologies like multifactor authentication (MFA) and single sign-on (SSO) are common IAM authentication methods.
Identity Governance	Tracks what users do with their access privileges. This ensures privileges are not abused and helps detect bad actors with unauthorized access.

Two technologies mentioned in [Table 2-8](#) that are key parts of IAM and critical for SaaS security are multifactor authentication and single sign-on. MFA is a process that requires at least two distinct authentication factors for user verification. One common factor is a username and password combination, but another factor could be a one-time password (OTP) that is sent to a mobile number or an authentication application. A great example of this type of authentication is Cisco Duo. With Cisco Duo, an application is associated with a user’s mobile device, and an OTP is sent as part of the MFA process. We will cover Cisco Duo in [Chapter 8](#), “[Security: Identity and Access Management](#).”

SSO is a service that allows a user to utilize one set of credentials for accessing multiple resources or applications. For example, Cisco Webex Meetings supports an SSO integration. This service allows a corporate user to log in to their Webex Meetings application using their corporate login credentials instead of having a separate set of credentials just for Webex Meetings. SSO provides users a more seamless experience without having to keep track of and constantly enter multiple login credentials.

Just about every enterprise-level SaaS application supports IAM to some extent today. More specifically, both MFA and SSO are integral IAM technologies that are becoming more common and, in some cases, expected for enterprise SaaS applications. However, you should be aware that IAM capabilities and functions are usually optional, so you need to make sure you enable and configure it to fully secure your SaaS application.

Visibility and Monitoring

In security circles, you often hear the saying, “You can’t protect what you can’t see.” This simple truth illustrates the importance of visibility and monitoring in the Security and Privacy space. The more visibility and monitoring capabilities that you have, the harder it is for issues to remain unknown or for bad actors to remain undetected.

SaaS application providers have many tools at their disposal for visibility and monitoring. Some of these tools are provided by the CSPs themselves. For example, AWS has Amazon GuardDuty and Amazon CloudWatch. GuardDuty offers a continuous threat detection service that looks for unauthorized or malicious behavior. CloudWatch can watch and collect metrics, monitor and collect log files, set up alarms, and so on. The other major CSPs have their own security-related monitoring tools as well.

Application performance monitoring can also provide visibility in a security context. One of the commercial tools in this area is Cisco AppDynamics, which uses agents to monitor cloud applications and resources. It has capabilities that allow it to not only provide visibility on an application’s performance but also its security health. As you can see, visibility and monitoring in a security context overlap with visibility and monitoring in a management and analytics context. Therefore, we will cover AppDynamics in more detail in the following section, “[Management and Analytics](#),” and also on its own in [Chapter 11, “Observability and Monitoring: Cisco AppDynamics and Splunk.”](#)

An exciting and growing area for security visibility is AI-powered behavior analytics. This technology leverages AI and ML to analyze large data sets to identify unusual patterns that are different from normal patterns and usage. These abnormal patterns can indicate malicious activities from attackers. For

example, a user suddenly starts transferring large amounts of data or downloading large files. This odd behavior could be detected by behavior analytics, especially if this fact is coupled with the fact that their account is being accessed from a new device in a new geographical location. With behavior analytics, suspicious activities can be identified quickly to minimize damage.

In this section, we introduced security and privacy as part of the SaaS Architectural Model. With the increasing number of security threats and bad actors, SaaS security and privacy remain a hot topic. Users of SaaS applications put a large amount of trust in the application providers. At the same time securing SaaS is a shared responsibility between the users and the provider. Cloud security controls, including IAM, are critical to ensuring that users and their data are protected when using a SaaS application. For more in-depth coverage of these topics and many others concerning SaaS security, refer to [Chapter 4](#).

Management and Analytics

A myriad of tools and applications exist for SaaS providers to manage their application and gather analytics for dashboards and visualizations. Some of these tools are available directly from the CSP, whereas others are commercial products or open source. Selecting which product to use depends a lot on the back-end architecture and scale of the SaaS application.

Most of the management and analytics associated with a SaaS application can be covered by the term *cloud operations*, or *CloudOps*. CloudOps is focused on management and operations and includes configuration and deployment, backup and recovery, and cloud monitoring. It is not SaaS specific but typically part of any IT organization that has a cloud infrastructure and applications to manage.

Therefore, most of the tools that a SaaS application provider utilizes for management are often the same as any CSP customer may use to manage their own applications being hosted in the cloud. These tools can be broken down into two main categories: infrastructure and configuration management and monitoring and observability. [Figure 2-36](#) shows how these categories fit into the Management and Analytics block of the SaaS Architectural Model.

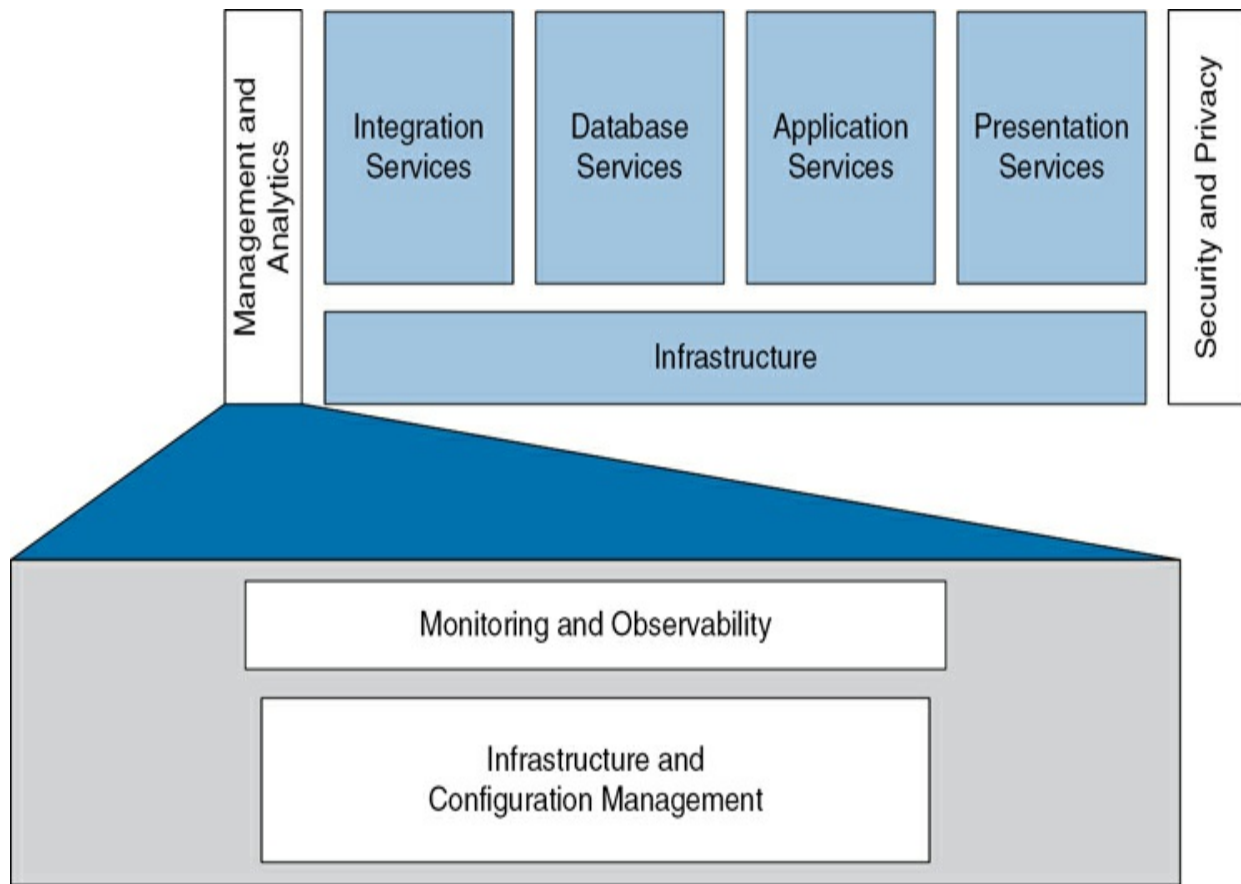


Figure 2-36 Management and Analytics Block of the SaaS Architectural Model

You should be aware that hundreds of tools and pieces of software exist in the Management and Analytics space for cloud applications. The categorization in [Figure 2-36](#) simply seeks to place a loose structure around what is available for ease of understanding. As a way of providing more depth and context, we will provide a couple examples for these categories. This discussion will provide you with a general idea of the functions and capabilities for other tools and software in that category.

In the “[Infrastructure](#)” section earlier in this chapter, we discussed the compute, storage, and networking components that underlie any SaaS application as well as the system software and tools. When you think of the infrastructure and configuration management category in [Figure 2-36](#), you can focus on how to manage and provision the SaaS infrastructure as described in that section.

A few of the tools often associated with infrastructure provisioning, such as Docker and Kubernetes, were introduced earlier in the “[System Software and Tools](#)” section. For some smaller SaaS deployments, a provider may not need much more than these tools. However, as SaaS applications grow larger in scale, more advanced software is used for managing and provisioning in the cloud infrastructure. This software usually integrates directly with tools like Kubernetes and Docker to provide automation and management at scale. You can think about this scenario as automating the provisioning of Kubernetes and Docker on cloud platforms. One example of this type of software is Terraform.

Terraform by HashiCorp uses a declarative configuration language to define a data center infrastructure and its desired “end state.” To reach this end state, Terraform is able to identify the “actual state” of the resources and then determine the changes that are needed to the infrastructure to move it to the end state. Resources, from a Terraform perspective, are just infrastructure objects, like compute or storage, or services, like a security group. You can easily define these resources, their configuration, and their state at large scales across multiple CSPs. [Figure 2-37](#) provides a high-level overview of Terraform.

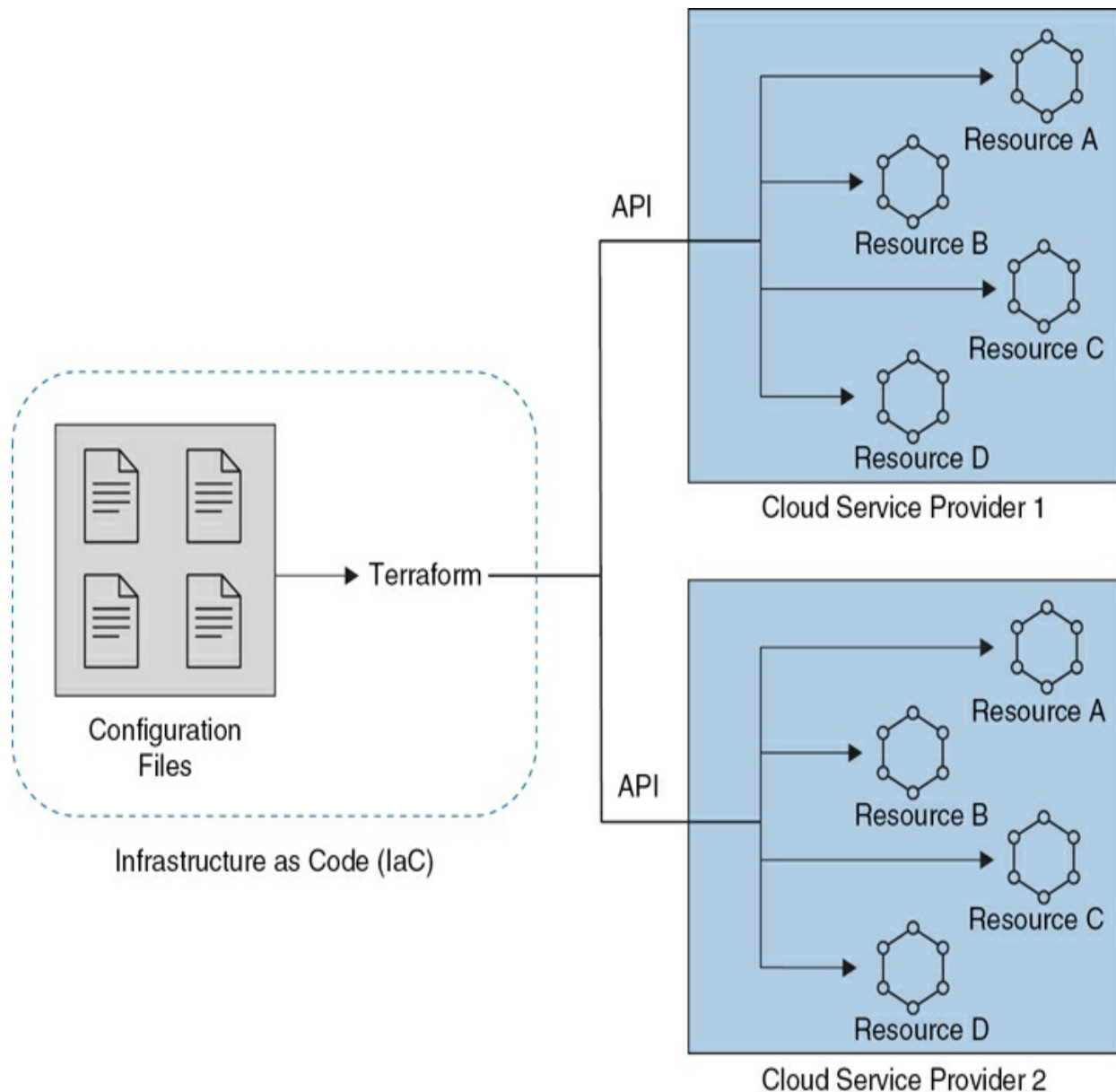


Figure 2-37 Terraform for Infrastructure and Configuration Management

Terraform is one of many infrastructure and configuration management applications that is part of the fast-growing area known as Infrastructure as Code (IaC). IaC refers to the concept of taking configuration files and using them to automate the provisioning and configuring of infrastructure resources. These resources can reside in your own data center or be in a CSP. IaC systems, like Terraform, are compatible with most major CSPs, and this enables them to take a single configuration and apply it to any CSP and its resources.

In addition to Terraform, other powerful IaC tools also exist. Some of them use an imperative or procedural, rather than a declarative, approach for infrastructure and configuration management. With procedural languages, you don't just define the end state. Instead, you specify the steps needed to change your infrastructure and configuration into what you want it to look like. Another way to look at it is that a procedural IaC approach details what to do (that is, the state) to the infrastructure and how to do it, whereas declarative just details what to do. With declarative, the platform determines the "how," or best way, to achieve that state. [Table 2-9](#) provides an overview of Terraform and three more popular IaC platforms that are worth knowing about: Ansible, Puppet, and Chef.

Table 2-9 Common IaC Software

laC Software	Description
Terraform	Terraform leverages a declarative configuration language with a simple syntax called Hashicorp Configuration Language (HCL) to define an “end state” for the infrastructure. This easy-to-use language, combined with the ability to safely and efficiently provision and manage multiple on-premises and cloud data centers, makes Terraform one of the most popular laC tools.
Ansible	Ansible uses a procedural language and playbooks written in YAML to manage infrastructure configuration. Initially, YAML stood for Yet Another Markup Language. However, more recently it has been known by the recursive acronym YAML Ain’t Markup Language to highlight that it is data oriented and not document markup. Ansible is known for its flexibility and being able to work with just about any system. Also, Ansible is agentless and does not require special software to be installed to automate infrastructure hardware.
Puppet	Puppet is one of the oldest laC applications. It is popular with large companies and has a widespread support community. It uses a declarative language known as Domain-Specific Language (DSL) for its configuration files, and one of its common deployment models requires agents to be installed on machines.
Chef	Chef is a well-established laC platform that is procedural. Largely based on Ruby, the Chef Infra Language (CIL) is used to write the configuration code that is then defined by cookbooks and recipes. Known for having a steeper learning curve, Chef does have advanced capabilities and an active community for getting support. Additionally, Chef uses an agent-based architecture where additional software is required on each machine.

Monitoring and observability is the other category in the Managements and Analytics block highlighted in [Figure 2-36](#). Being able to observe and collect data on a SaaS application is critical for ensuring its efficient operation and that customers are having a good experience. Before diving into a deeper discussion and exploring some of the tools in this space, you should be aware of these terms and concepts: *full-stack observability (FSO)*; *OpenTelemetry*; and *metrics, events, logs, and traces (MELT)*.

Today’s IT environment is complex and rapidly changing, especially when it comes to SaaS and the cloud in general. As previously discussed, SaaS

applications are typically built on a microservices framework in the cloud and integrated with numerous other services and applications. In large environments, multiple teams are required for operations and support, and a small issue in one service can have a cascading effect on other services and the user experience. Visibility to an issue and its effect on the entire environment has in the past been siloed and domain specific. Full-stack observability changes this paradigm and correlates telemetry and data across multiple domains and throughout the technology stack. With FSO, you can know the state of every endpoint in a distributed environment. It provides you a real-time, complete overview of the behavior, performance, and health of not only your applications but also the entire underlying infrastructure.

FSO would not be possible without the automatic retrieval of data and measurements, such as metrics, events, logs, and traces from remote sources. The MELT framework focuses on these four fundamental types of telemetry data. Metrics and events are typically smaller bursts of telemetry, with metrics happening more regularly for a parameter-like memory utilization. Events are usually triggered by a potential problem, or a threshold being exceeded. Logs and traces are larger chunks of data, with logs being a record of activities over time and traces focused on capturing an end-to-end flow through components in a system.

As you can imagine, collecting MELT data from different systems using different formats can present a significant challenge. Because of the amount of disparate telemetry data from different manufacturers and its continued growth, the need for standardization has become important. The OpenTelemetry framework is an emerging standard for collecting telemetry that is seeing widespread adoption. By providing a unified standard for the creation and ingestion of telemetry data, OpenTelemetry aims to provide a collection mechanism and format that is vendor agnostic. Before OpenTelemetry, organizations were locked into specific vendor models and systems for telemetry.

Quite a few software platforms can use telemetry to monitor and observe SaaS applications. For comprehensive, centralized monitoring of servers and applications, the Elasticsearch, Logstash, and Kibana (ELK) stack is probably the most well known. A combination of three tools, Elasticsearch stores large amounts of logging and other telemetry for efficient retrieval, search, and

analysis. Logstash collects and normalizes data and telemetry from various sources to be consumed by Elasticsearch. Kibana is the front-end dashboard and visualization tool for the Elasticsearch data.

Cisco has its own SaaS applications in this space that provide some overlapping functionality or can be complementary to ELK, depending on the use case. These solutions include Splunk, AppDynamics, and ThousandEyes, and they are often used for monitoring other SaaS applications.

AppDynamics utilizes agents for monitoring data center systems and software, along with a controller for receiving agent data and sending them instructions. ThousandEyes is focused on network connectivity. It uses agents as well, but they monitor network paths and reachability between points in your network and to locations all over the Internet. [Figure 2-38](#) shows a basic example of the monitoring capabilities for AppDynamics and ThousandEyes and how they can integrate to improve FSO.

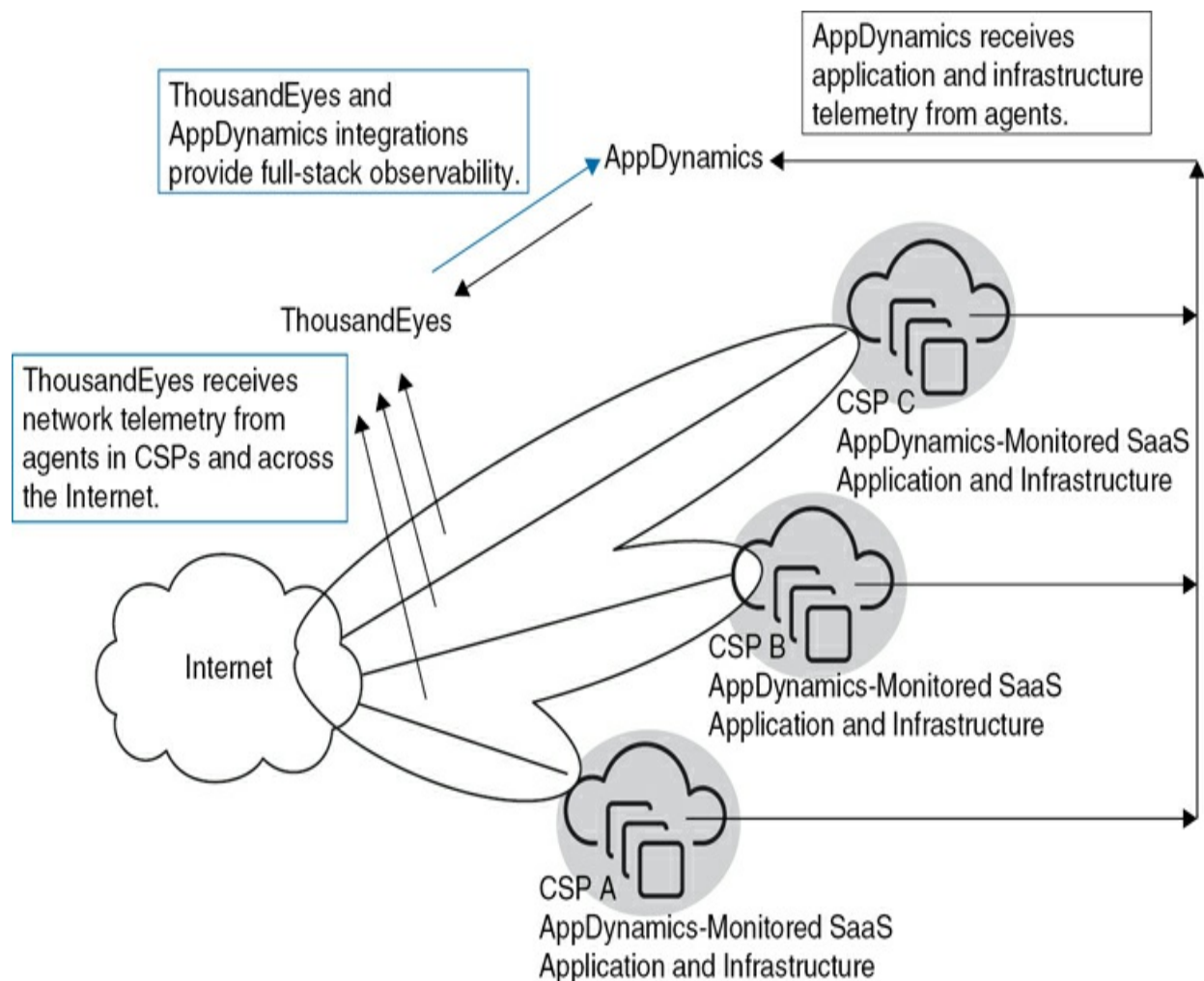


Figure 2-38 AppDynamics and ThousandEyes FSO Integration

In [Figure 2-38](#), a SaaS application provider has its application running on three different CSPs for redundancy and availability reasons, or maybe it is the same CSP but the provider is using different regions or geographical locations. AppDynamics agents are deployed for monitoring the SaaS infrastructure and services at each CSP. ThousandEyes agents are also installed at each CSP so the customer can gather telemetry about connectivity between the SaaS applications running at each CSP and performance of the SaaS application over the Internet in general.

Note

You should be aware that CSPs also have a portfolio of tools that can be used for SaaS management and analytics that are aligned to their offerings. These are typically referred to as CloudOps. SaaS application providers use these tools, services, and features along with others that are not CSP specific (some of which are covered this section) to get the visibility, insights, and automation they need to effectively maintain their application.

As discussed earlier in the “[Integration Services](#)” section, SaaS applications can connect and integrate with other applications. In [Figure 2-38](#), you see that ThousandEyes and AppDynamics can integrate with one another so that their data can also be integrated and displayed together for better view of the SaaS application and its performance.

Being Cisco SaaS solutions, both AppDynamics and ThousandEyes have dedicated chapters in this book. You can read more about AppDynamics along with more in-depth coverage of FSO, MELT, and OpenTelemetry in [Chapter 11](#). If you want to take a closer look at ThousandEyes, refer to [Chapter 12](#).

Note

Cisco Intersight and Cisco Meraki are two SaaS applications that are also in the management space but are a little different. They both are focused on managing pieces of physical hardware from the cloud, unlike AppDynamics and ThousandEyes, which are focused

on telemetry from infrastructure, applications, and the network itself. For more information on Cisco Meraki, visit [Chapter 13](#), “[Management: Cisco Meraki](#),” and Intersight is covered in [Chapter 14](#), “[Management: Cisco Intersight](#).”

The Monitoring and Analytics block of the SaaS Architectural Model is a critical piece of any SaaS architecture, especially for larger applications. It becomes almost impossible to manage the infrastructure and configuration and keep up with the vast amount of telemetry without the proper software and tools. We discussed a few of the options in this section, but you should be aware that many more are available. This is a fast-developing space as SaaS continues to grow and applications need to keep scaling in size.

Multitenancy

Now that you have a basic understanding of SaaS architecture, you are prepared to dive into multitenancy. We touched on multitenancy in [Chapter 1](#); it is one of the attributes of SaaS that makes it a compelling solution for many companies and organizations. At its core, multitenancy is the sharing of resources, and this allows for much better scalability compared to separate resources for every tenant or customer. In this section, we’ll take a deeper dive into multitenant structures in the cloud from a SaaS perspective.

Note

There is often some confusion around the architectural concepts of virtualization and multitenancy. We covered virtualization in the “[System Software and Tools](#)” section earlier in this chapter. If you recall from that discussion, virtualization is the idea that multiple instances of a server can be run on a single physical server. Multitenancy, on the other hand, details the usage or sharing of a single or multiple applications by multiple users.

Single tenant models are easy to understand. One customer is assigned to any given resource. Typical resources include an application instance with a web frontend or a database. This resource is not shared, and the tenant has sole access and use of the resource. [Figure 2-39](#) shows a single tenant model.

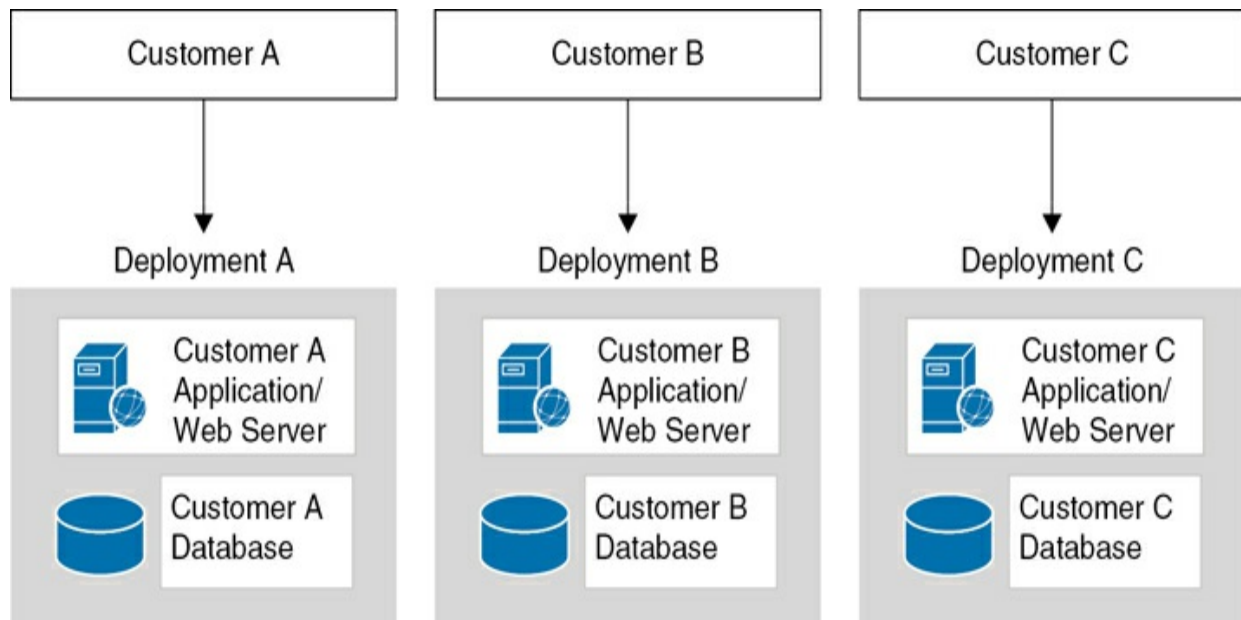


Figure 2-39 SaaS Single Tenant Model

[Figure 2-39](#) shows three tenants or customers: Customer A, Customer B, and Customer C. Each has its own dedicated deployment consisting of its application and web server and database. The customers do not share any resources. Each customer is a single tenant on a resource.

You should be aware that usually in this model, hardware is being shared, even if the software is single tenant. If you think back to the discussion on virtualization, a single physical server can be divided into multiple virtual machines or containers. So, although each VM or container might have an application or database with a single tenant, the underlying hardware is shared with other VMs or containers with other tenants. Some SaaS application providers do offer an option for a dedicated host. With a dedicated host, only your resources are allowed on a dedicated physical server. The SaaS application provider ensures that other customers' workloads or data will not be placed on your host.

With a full multitenant deployment, you have the complete opposite approach to single tenants. With multitenancy, a cloud resource is shared between multiple customers. [Figure 2-40](#) provides an illustration of a multitenant SaaS model.

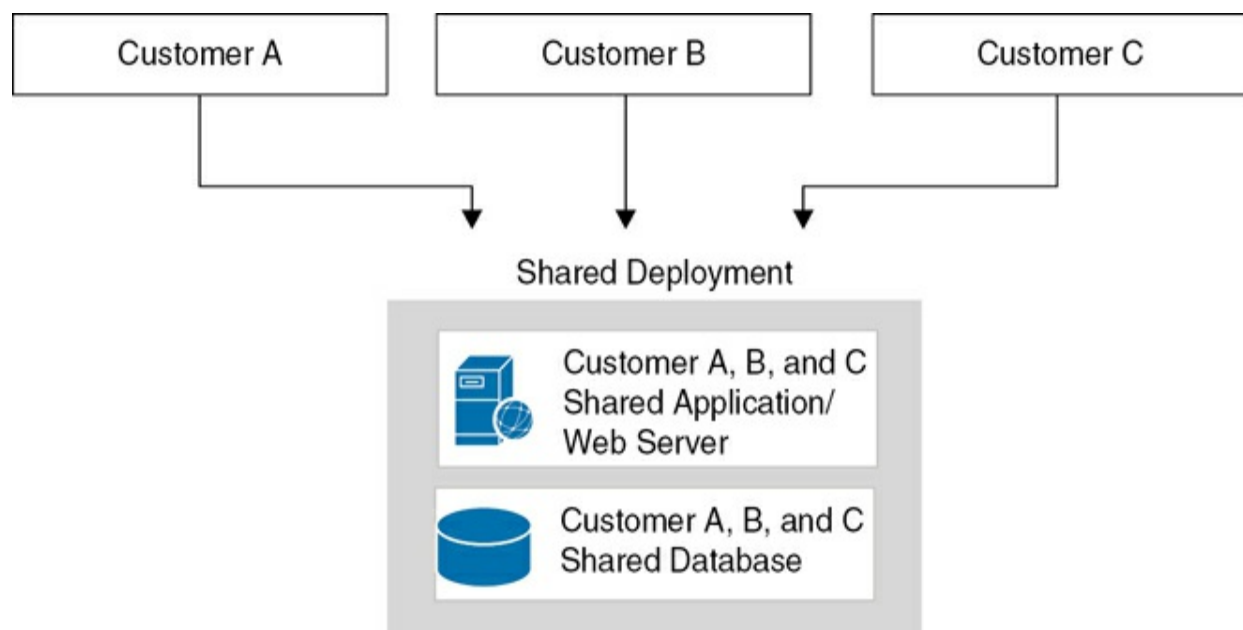


Figure 2-40 SaaS Multitenant Model

In [Figure 2-40](#), there is a single software instance for the application and a single database that Customer A, Customer B, and Customer C all share. With this type of structure, components and resources are shared. This means that both the application and database must be built to handle multiple users in a secure manner that prevents data leakage between tenants. After all, customers now all reside on a single infrastructure, so isolating customers and their data is more of a concern.

Some of the advantages and disadvantages of single tenant and multitenant architectures are highlighted in [Table 2-10](#). This is not an exhaustive list but should give you an idea of the main differences. SaaS application providers take these sorts of factors into account when building their application architecture.

Table 2-10 Comparison of Single Tenant and Multitenant Architectures

Tenancy Model	Advantages	Disadvantages
Single Tenant	<ul style="list-style-type: none"> • There is improved security and a lower risk of accidental data leakage between tenants. • One tenant is unlikely to affect the performance of another. • Customization per tenant is easier. • Changes and updates can be rolled out per tenant to minimize system wide outages. 	<ul style="list-style-type: none"> • It is resource intensive and more costly with multiple infrastructures to maintain. • Scaling up to more users can be challenging because a new instance needs to be created for every new tenant.
Multitenant	<ul style="list-style-type: none"> • A single infrastructure makes maintenance easier. • It is less expensive to operate because the same instance is shared. • It is easier to scale and add more tenants. 	<ul style="list-style-type: none"> • Security breaches can cause more damage and more opportunity for data leakage between tenants. • Changes can be complicated, and issues can affect the entire user base. • Customizing for an individual tenant is harder.

Most SaaS applications are not strictly a single tenant or multitenant architecture. Instead, these architectural constructs are blended to better leverage the advantages of each. For example, extending our simple example of three customers accessing a SaaS application and database, you can see how single tenancy and multitenancy can be utilized together. In [Figure 2-41](#), customers share an application instance but have separate databases.

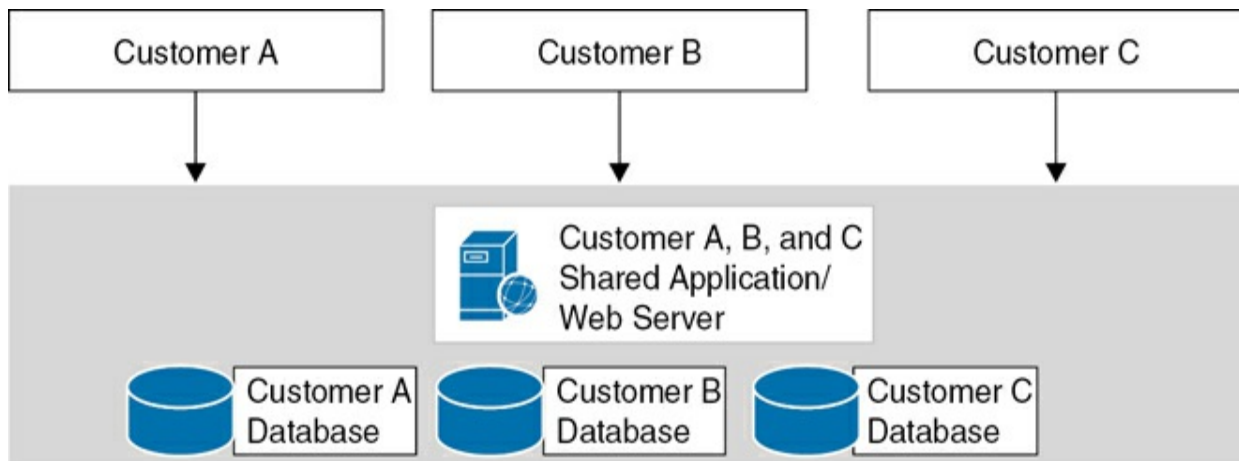


Figure 2-41 SaaS Single and Multitenancy Combined Architecture

A deployment like what is depicted in [Figure 2-41](#) enables the application provider to leverage the management benefits of a multitenant structure for the application but provide more security for user data with single tenant databases. Note that the examples shown in this section are simplified to illustrate the basic concepts of single and multitenancy. As SaaS solutions grow in complexity and the number of users, multiple applications and services are often present, along with numerous databases. Application providers can apply single and multitenancy principles across all these resources, depending on use cases and requirements. In fact, as you get into enterprise-level SaaS applications, how tenancy is applied across all resources can be quite customized and will change as the application scales and evolves.

In this section, we provided an overview of multitenancy architecture and how it compares to single tenancy. Multitenancy is a critical part of SaaS because it allows for the efficient usage and scaling of SaaS applications. At the same time, in some use cases, factors such as security and a requirement for data isolation make a single tenant architecture a better fit. Most SaaS applications today are not fully single tenant or multitenant but instead use both, depending on the use case and application requirements for a resource.

Summary

SaaS architecture is a subset of cloud architecture and, much like building a house, requires an understanding of the foundational elements and thoughtful

planning to ensure success. Additionally, just as you could not learn everything about residential architecture in a single chapter, this chapter serves as a preliminary look at the core elements of SaaS architecture and a behind-the-scenes look at the technologies that make SaaS possible.

We started this chapter by introducing SaaS architecture using two models—a logical one and an architectural one. First, we built a logical model for SaaS by presenting the basic construct of a control plane and application plane that is associated with SDN. We then expanded this construct to show how SaaS can be defined and represented by this context.

We introduced the architectural model next. Based on NIST SP 500-292, this model broke down SaaS architecture into a set of blocks for easier understanding and consumption. We then discussed each block in a separate section. The first block we covered was Infrastructure. The hardware and technologies that are the cornerstone of any cloud deployment were covered, including compute and storage servers and the networking infrastructure that joins all SaaS components. We also discussed virtualization and container technologies.

For the second SaaS Architectural Model block, we examined Application Services. This block is the heart of any SaaS application because it contains the business logic and core functions that make that SaaS application valuable. Additionally, we reviewed the system software and tools that focused on the microservices and serverless architectures that are commonly used by SaaS applications.

Database Services was the third block we discussed, and we delved into how SaaS data is stored. We defined structured and unstructured data types and then covered various database types, like relational, graph, and vector, along with some of the factors that need to be considered when determining the appropriate database to use.

Next, we explored the Presentation Services block. This block is familiar to SaaS users because this is how SaaS applications are accessed utilizing methods, such as the web, apps, or dedicated devices. This section also clarified the differences between the frontend and backend of a SaaS application and introduced the front-end protocols, including HTML, CSS, and JavaScript.

The fifth block we covered was Integration Services; it allows for other apps and services to connect with a SaaS application. Several different methods are available for building these connections. We examined both prebuilt integrations, like connectors and modules, along with custom integrations, such as APIs, webhooks, and WebSockets.

Finally, Security and Privacy and Management and Analytics were the final two blocks we defined in the SaaS Architectural Model. We discussed Security and Privacy areas like cloud security controls, IAM, and visibility and monitoring. We also explored Management and Analytics tools and technologies for infrastructure and configuration management and monitoring and observability.

In the last section in this chapter, we presented multitenancy. Multitenancy is a concept that is critical to SaaS because it allows for the efficient scaling of SaaS applications. We explained single tenant and multitenant models, along with models using a hybrid approach.

This SaaS architecture chapter provided you with a peek at how a SaaS application is planned and built. The architectural details we discussed in this chapter are most often not visible to you, as a user. However, this knowledge is necessary for building a foundational base in SaaS. You will find that the SaaS base level of knowledge from this chapter will be helpful in further understanding the real-world SaaS applications we will discuss in subsequent chapters.

References

- Control plane vs. application plane:
<https://docs.aws.amazon.com/whitepapers/latest/saas-architecture-fundamentals/control-plane-vs.-application-plane.html>
- What is a cloud service provider?: <https://cloud.google.com/learn/what-is-a-cloud-service-provider>
- Odom, Wendell. Introduction to Controller-Based Networking. *CCNA 200-301 Official Cert Guide*, Volume 2. Cisco Press, 2019.
- NIST cloud computing reference architecture, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf, NIST,

2011:

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf>

- What is hyperconverged infrastructure?:
<https://www.cisco.com/c/en/us/solutions/data-center-virtualization/what-is-hyperconverged-infrastructure.html>
- Cisco ACI multi-tier architecture white paper:
<https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-742214.html>
- What is virtualization?:
<https://www.redhat.com/en/topics/virtualization/what-is-virtualization>
- Containerized microservices: <https://learn.microsoft.com/en-us/dotnet/architecture/maui/micro-services>
- Serverless architecture overview, DATADOG Knowledge Center:
<https://www.datadoghq.com/knowledge-center/serverless-architecture/>
- What is structured data?: <https://aws.amazon.com/what-is/structured-data/>
- Tapping the power of unstructured data, Tam Harbert:
<https://mitsloan.mit.edu/ideas-made-to-matter/tapping-power-unstructured-data>
- Graph databases for beginners: A (brief) tour of aggregate stores, Bryce Merkl Sasaki: <https://neo4j.com/blog/aggregate-stores-tour/>
- Getting to know ThousandEyes—Part 2, Flo Pachinger:
<https://blogs.cisco.com/developer/learn1000eyes02>
- ThousandEyes API v7: <https://developer.cisco.com/docs/thousandeyes/>
- Webex app hub: <https://apphub.webex.com>
- AppDynamics:
<https://docs.appdynamics.com/appd/22.x/latest/en/extend-appdynamics/integration-modules/integrate-appdynamics-with-splunk>
- Duo Splunk Connector: <https://duo.com/docs/splunkapp#install-duo->

splunk-connector

- Terraform for beginners, Dave Storey: <https://itnext.io/terraform-for-beginners-dd8701c1ebdd>
- Splunk AppDynamics: <https://www.cisco.com/c/en/us/solutions/collateral/full-stack-observability-aag.html>
- MELT explained: Metrics, events, logs & traces, Austin Chia: https://www.splunk.com/en_us/blog/learn/melt-metrics-events-logs-traces.html
- What is OpenTelemetry? A complete guide, Stephen Watts: <https://www.appdynamics.com/topics/what-is-open-telemetry>
- Tenancy models for a multitenant solution, John Downs: <https://learn.microsoft.com/en-us/azure/architecture/guide/multitenant/considerations/tenancy-models>
- JavaScript: <https://en.wikipedia.org/wiki/JavaScript>
- What is identity and access management (IAM)?, Matthew Kosinski and Amber Forrest: <https://www.ibm.com/topics/identity-access-management>
- What are microservices?: <https://www.redhat.com/en/topics/microservices/what-are-microservices>
- SOA vs Microservices—Difference between them, Elijah Hall: <https://www.guru99.com/microservices-vs-soa.html>
- DB-Engines ranking of vector DBMS: <https://db-engines.com/en/ranking/vector+dbms>

Chapter 3. Migrating to SaaS

Software-as-a-Service (SaaS) applications are not a new concept in the technology industry; they have been around for many years. However, in recent years, the adoption of SaaS within enterprise organizations has grown exponentially. Now, adopting or migrating solutions to SaaS applications is a reality for most, if not all, enterprises. Every year, more companies offer SaaS components within their business models. As the market becomes more saturated with SaaS vendors, finding an application that meets your specific business objectives becomes more complex. Many SaaS applications offer similar or competing functionality, so determining the best solution can prove to be a nuanced and tricky decision.

Companies are also seeing many compelling reasons to start adopting SaaS applications in their businesses and lessening their reliance on on-premises data centers. SaaS allows for more predictable spending, the ability to scale on demand, ease of management, increased feature velocity, and rapid innovation, among other things.

When an enterprise is transitioning from an on-prem to a SaaS application, the level of planning and implementation can vary based on the size of that enterprise. Whether it is migrating from an existing platform or simply adopting a new application, a well-thought-out plan can lead to a higher degree of success. While many SaaS vendors are making steady improvements to support the migration process, allowing for easier onboarding and migration of existing data, there are still opportunities for failure. Successful adoption of a new SaaS product can make a dramatic difference in that application's user experience and willingness to adopt; plus, it can keep the migration costs predictable and reasonable.

Transformation to adopt SaaS takes two primary forms. The first is a

company deciding to implement a new SaaS solution to solve a business or customer problem. The second is a company looking to migrate from an existing on-premises application to a SaaS application. This form may involve rearchitecting or migrating existing solutions to be delivered by a SaaS product. In this chapter, we will focus on the second—migrating from on-premises to a SaaS application.

The migration plan to deploy a SaaS application can be broken down into the four phases shown in [Figure 3-1](#). Each stage of this migration flow builds onto the previous and should ideally be completed in this order. While not every migration is the same, and some may require more emphasis on different phases of this flow, each step provides unique value to aid the migration process.



Figure 3-1 SaaS Migration Flow

In this chapter, we will explore some of the best practices you can follow when migrating to a SaaS application. We will discuss the following topics:

- **Discovery:** Determining the readiness and business value of migrating to a SaaS application
- **Design and Planning:** Defining steps to accomplish the migration and rollback plans
- **Implementation:** Following the plan created to perform the migration
- **Value Realization:** Confirming that the completed project has achieved the desired outcomes
- **Common Migration Challenges:** Examining the common problems that could be encountered during migration and ways to mitigate those problems

Discovery

The discovery phase of migration is the first step that should be taken to evaluate whether there is business value or justification in moving to a SaaS-based application. The discovery process begins by taking a step back to understand how a tool or application is currently being utilized in your organization. This approach allows you to make an informed decision on whether migration to a SaaS application will produce positive results in the form of efficiency gains, cost savings, and so forth. Common rationalization for migrating to a SaaS-based application may vary depending on the use case or business. For some businesses, offloading localized hardware or maintenance costs to a SaaS vendor or product may prove to be financially expedient. Another common rationale is due to software reaching its end of life and pursuing an alternative or a successor solution that improves ease of management or enhances features for your users to leverage.

The reality is that companies are making rapid transitions to SaaS applications to help their teams solve business problems. In [Figure 3-2](#), you can see that from 2021 to 2023, enterprise organizations saw a 49 percent increase in the number of SaaS applications in their portfolio, reaching an average of 473. The question for many businesses is more of *when*, not *if*, they should adopt SaaS solutions.

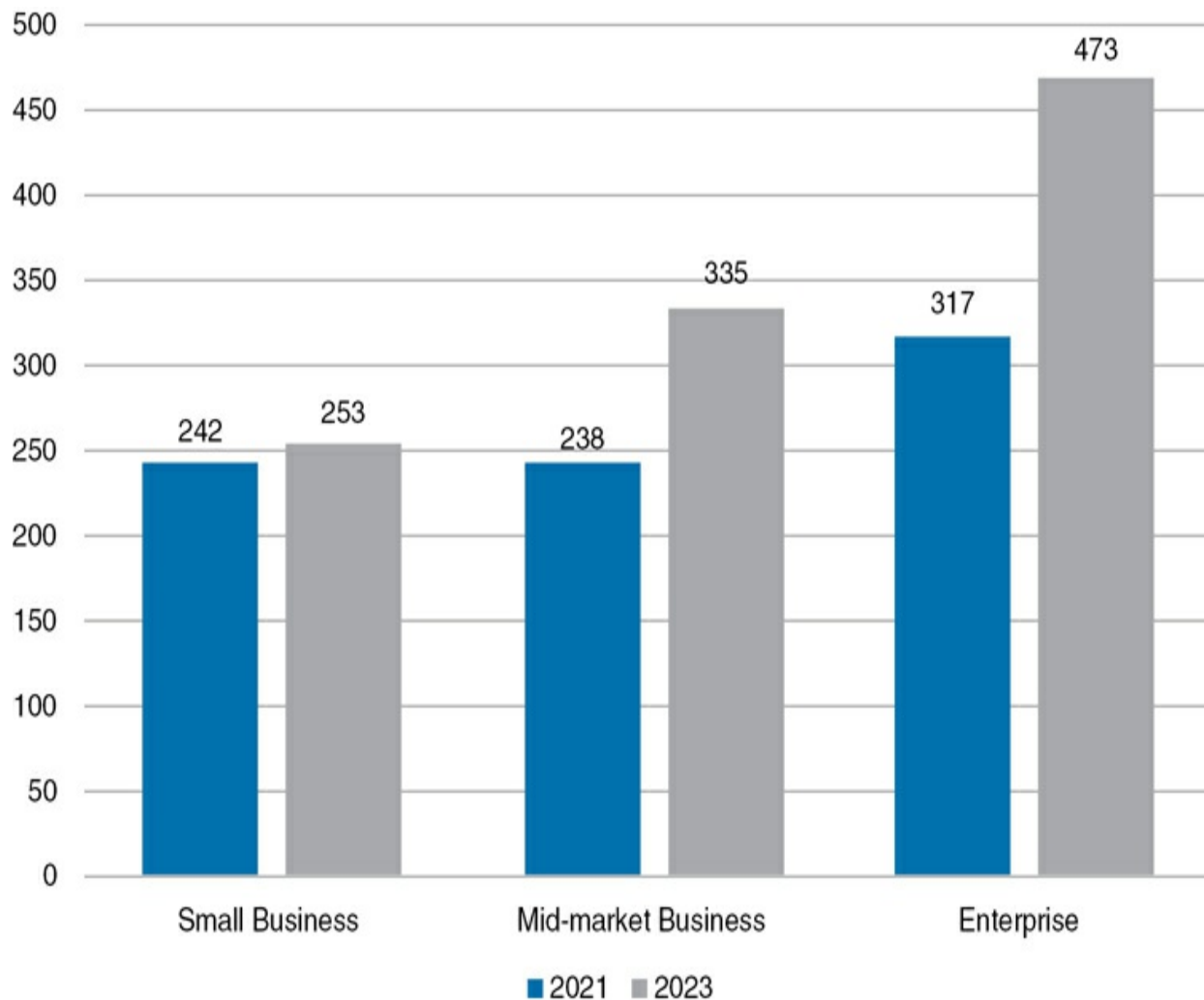


Figure 3-2 Average SaaS Portfolio Size from 2021 to 2023

Once you identify an opportunity to transition to a SaaS application, you should consider several focus areas. Each of these focus areas will help you make an informed decision about the readiness and effort required to migrate. In the following sections, we will explore practical ways to evaluate an application's migration readiness.

Cost of Ownership

When you compare the cost of hosting an application or service on-premises to a SaaS application, one of the key differences is the up-front investment required to deploy a service on-premises. Not only are there likely licensing fees for the application, but infrastructure and hardware components must

also be deployed. This process can be extremely costly, complex, and time-consuming. For a company that spans multiple geographic locations, there is often a need to deploy applications across multiple data centers to improve reachability, latency, and high availability should there be an outage. Additional costs also may be required to scale computational resources as needed to handle increased demand or load on the service.

With a SaaS application, there are often no up-front infrastructure costs. Generally, SaaS services are subscription-based, and the pay structure is scaled to the usage of the application. Many SaaS providers offer a charge per user per month. In this subscription model, your cost to leverage the service is directly related to the number of users you plan to onboard and can scale up to meet growing demand as needed. Additionally, your cost to leverage the service is more predictable.

Suppose you are hosting an application on-premises in a self-managed data center, and a hardware failure causes an outage to your application. In that case, the cost of replacement or repair becomes an unexpected expense related to hosting this application. You also must consider the duration of the downtime having a potential impact on revenue or the cost of having replacement parts or redundant systems on hand.

There may also be hidden, albeit often necessary, costs to migrate to a SaaS application. Migrating to a new service often takes planning, data migration, testing, change management, training, and possibly even downtime for the business. Each of these factors can incur an expense.

[Table 3-1](#) compares a few of the differences between SaaS and on-premises costs. This table can be a helpful guide to understanding how SaaS spending differs from on-premises.

Table 3-1 Comparison of On-Premises vs. SaaS Costs

On-Premises	SaaS
There are initial costs to license and deploy the infrastructure.	There are no up-front costs associated with the infrastructure for hosting.
There are hardware maintenance and support costs for hosting the application.	There are no hardware or hosting fees. This is taken care of by the SaaS provider.
Upgrades can be costly and/or time-consuming, often causing downtime.	You are always on the latest version. Feature velocity is much greater for SaaS, and software upgrades are handled by the vendor. Often, SaaS vendors are contractually obligated to maintain some amount of uptime.
Scaling up is costly, because this effort could require additional hardware to support the added load.	The service scales easily, growing or shrinking with your current needs, without your needing to pay for additional hardware.
Data backup and resiliency may come at a price. Custom solutions are often required for each application.	Most SaaS providers have redundant solutions already in place, along with data back-up strategies for disaster recovery scenarios.
Dedicated employees with training and knowledge are required to maintain and upkeep the service.	Experts support that product or service.

As you migrate to a SaaS application, you should consider the up-front and recurring costs of ownership and maintenance for your on-premises application. This comparison can be leveraged to understand whether the migration will make sense financially.

Partial or Full Migration

When you're migrating to a SaaS application, it is easy to assume that a lift-and-shift approach is being taken where you move entirely from one system or application to another. However, that isn't always the case. In some instances, migrating only a portion of an application to a SaaS solution may make more sense. Consider how you plan to leverage SaaS for your

application. Are you planning to migrate entirely to SaaS in replacement of an on-premises application? Or do you plan to relocate only part of an application or dependency to SaaS?

A partial migration might look like migrating only the database component of an application to the cloud using a SaaS database provider. In this scenario, you would be less concerned about user interactions with your application changing and more focused on database performance, scaling, and redundancy. You would also need to ensure reliable connectivity between your network and the cloud service being used.

A full migration would be to take an entire application and transition the use of that application to a SaaS provider. In this scenario, you must understand how this transition would impact the user experience (front end) and application performance (back end).

In either scenario, the goal is to understand what areas of your business could be impacted by the migration so that you can invest appropriate planning and focus to mitigate risks.

The Reason to Identify Key Features and Functionality

During the discovery phase of a migration, you may be considering different SaaS vendors as potential replacements for your current application.

Gathering business requirements, collecting user stories, and leveraging existing application metrics can help you narrow down your choice of potential SaaS application. If the application you are currently using has a reporting or telemetry capability, you could leverage it to understand what components and features of an existing application are being utilized.

Another option is to gather user stories to understand how they interact with the product, their perceived limitations, or desired functionality. Whatever method you decide to use, the objective is to understand, at a high level, the features and services used by the application.

Consider an application used for task management. A combination of product telemetry from the application and user interviews regarding the task management application can help you determine that this tool is being used

for the features shown in [Table 3-2](#).

Table 3-2 Example Task Management Application Feature List

Feature	Used By	Priority
Capture and organize tasks	All	High
Reminders for task due dates	All	High
Assign tasks to one or more individuals	All	High
Weekly report on upcoming tasks	Project Managers	Medium
Ability to create a task via API	Software Engineer	Low

As you can see, each identified feature was noted along with other potentially important information, such as what groups leverage the feature and what the priority of this feature is. The intended outcome of this exercise is to clearly understand what features are critical to the usage and adoption of an application and what features could be optional. Optional items may not help you achieve feature parity with your existing application, but they may provide additional enhancements that improve return on investment.

This feature list will inform your decision-making when deciding what SaaS vendor to choose for an application migration. The goal is to find a SaaS solution that meets your unique business needs.

Application Configuration

Configurations and features are often related in that each feature will likely have some related configurations. When assessing what features are currently being leveraged by an application, you should also consider how those features are explicitly being used. The goal of looking at the configurations leveraged by an application is to unearth the components of an application that you need to have some level of customization over.

Continuing with the previous task management application example, in [Table](#)

3-2, one of the features is “Reminders for task due dates.” Perhaps a configuration item related to this feature is that the reminders need to be configurable to allow individual users to choose a specific reminder date and how they would like to receive that reminder (email, SMS, directly within the application, and so on). Every organization may leverage an application slightly differently, so it is essential to understand what features are being used and what customizations are used alongside those features.

There are trade-offs between simply replicating a feature from an on-premise solution and finding a SaaS solution that meets your needs in a novel or better way than how it was being done on-premises. By finding a service SaaS solution that replicates your existing application, you may achieve operational efficiency by now having that service managed and hosted for you, but you may be missing out on new and novel solutions that solve problems in a way that provides greater value to your business. It is important to be aware of what customizations are needed versus desired so that you can make the best selection post-discovery. One area within the application configuration domain you will not need to think as much about is application performance and hardware configurations. For example, an application could be deployed in an active/standby configuration so that should one of the application servers become unreachable, the other would be available for failover to ensure maximum uptime. Or you may have done some specific hardware tuning to increase your application’s performance. For SaaS applications, you will not have any control over an application’s hardware or performance tuning. However, most SaaS applications follow best practices for on-demand scaling, performance, and redundancy and come with service-level agreements (SLAs) that detail exactly what level of service you will receive from the service provider.

Application Integrations

Often, enterprises rely upon numerous systems working in combination to allow for smooth day-to-day operations. So, it is very likely that whatever application you want to migrate to a SaaS solution integrates with other systems. An application integration is a configuration that allows independent systems to communicate and share data. [Figure 3-3](#) illustrates examples of integrations with enterprise applications.

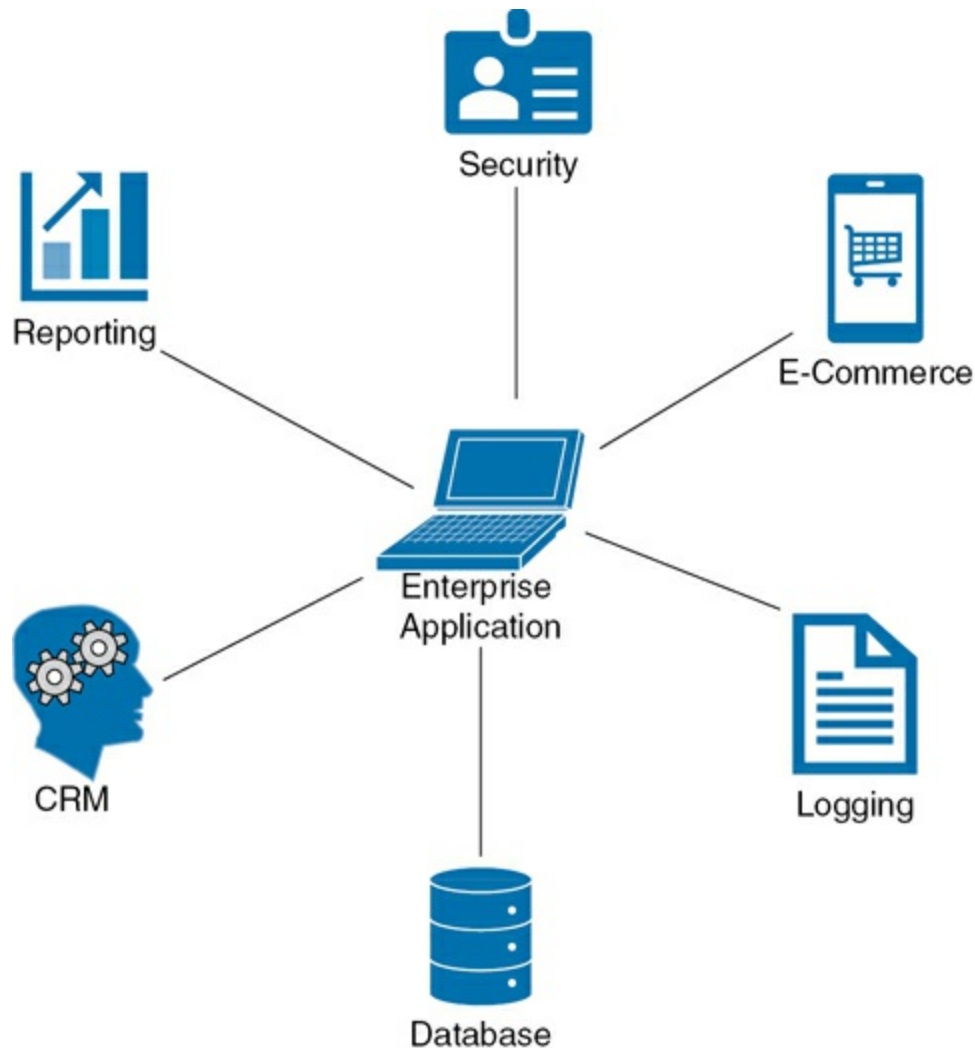


Figure 3-3 Examples of Common Application Integrations

Consider typical applications that you may use daily. Email clients have integrations with videoconferencing software like Cisco Webex to add to scheduled meetings. Web browsers support extensions for modifying the browser or adding additional utilities. Social media apps integrate with your phone's photos and contacts, allowing you to upload media and connect with people you know.

When it comes to integrations, you need to be aware of three main methods for integrating:

- **Native Integrations:** Provide tight connectivity between the source and destination service, typically made in partnership between two vendors

- **Protocol-Based Integrations:** Use common industry protocols such as SAML, LDAP, or SOAP to communicate between services
- **APIs:** Use a web-based protocol that leverages HTTP to communicate between services

Evaluating the integration opportunities for a SaaS application through these methods will help to ensure a successful discovery.

Integrations between applications are pervasive within the technology industry, and understanding what integrations your application is using is vital to completing a successful migration. Whether it is an identity management platform for the authentication and authorization of users, a connection with an e-commerce platform, or sharing data to a telemetry and reporting service, knowing what integrations are in place and how they are being leveraged is crucial.

During the design and planning phase of the migration, knowing what integrations need to be accounted helps ensure that once the migration is complete, you will have a similar integration or capability on the new SaaS application.

Network Requirements

Network engineers have commonly focused on the underlying infrastructure and design of on-premises data centers and connectivity between campuses to deliver a performant network for various workloads. However, with the rise of cloud-hosted solutions, this paradigm has shifted. Now, network engineers must also focus on reliable and secure connectivity between on-premises and cloud-hosted applications where the underlying network for the cloud-hosted applications is not within their control.

The changes required to support a migration to a cloud-hosted application should be considered during the discovery phase. The complexity of this task varies depending on the type of application being migrated and the requirements of the service you are migrating to. While end users are mostly unaware of the complex network that enables their applications to work day to day, they are acutely aware of the performance of an application, which can be directly impacted by a poor network design.

Consider a simple example of a migration in which you are migrating from a to-do list application hosted on-premises to a cloud-based to-do list. During the discovery process, you examine the network requirements for this new application and find that it requires connectivity over TCP port 7000 and TCP port 443, as depicted in [Figure 3-4](#). While TCP 443 is already allowed outbound from the network, TCP 7000 is not. This means that a change in the firewall configuration will be required to allow access to the cloud-based to-do list application.

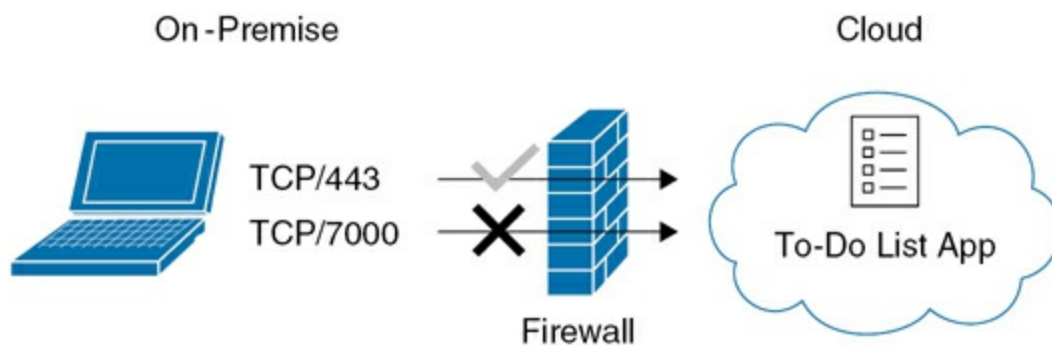


Figure 3-4 Firewall Blocking a Connection to a Cloud Application

During the discovery phase of a migration, when you're looking at the network requirements for an application, the goal is to understand what network changes will be required to allow for this migration to work. This information is an indicator of the level of complexity that is necessary to support the transition. During the design and planning phase of the migration, you can begin to plan for this new network architecture.

Security

Although the shift to SaaS has ushered in a new digital transformation era, it has also brought increased security concerns. As companies shift their confidential data from on-premises data centers to cloud-hosted solutions, they also lose control over the security of that data, trusting third-party vendors to maintain and secure it.

It is increasingly common to hear about data breaches as hackers exploit security vulnerabilities to gain access to information. Not only do data breaches reduce customer trust, but depending on the type of information that is collected, it could even pose a security and privacy risk for individuals.

Data breaches also carry a hefty price tag. According to a report by IBM, the global average cost of a data breach in 2025 was USD 4.4 million. Of the breaches that occurred, 72 percent involved data that was stored in the cloud (public cloud, private cloud, or across multiple environments including a cloud environment). This is why it is increasingly essential for organizations to ensure strict security policies and systems are in place to defend against and detect abnormal behaviors. [Table 3-3](#) covers a few examples of SaaS security audits that help to ensure necessary controls, systems, and configurations are in place to secure the environment.

Table 3-3 Common SaaS Security Audits

Audit	Description
SOC 2	<i>Service Organization Control Type 2</i> is a cybersecurity compliance framework developed by the American Institute of Certified Public Accountants (AICPA). The primary purpose of SOC 2 is to ensure that third-party service providers store and process client data securely.
ISO 27017	This standard provides guidelines for information security controls applicable to the provision and use of cloud services.
ISO 27018	This standard provides a collection of information technology—security techniques—code of practice for the protection of personally identifiable information (PII) in public clouds acting as PII processors.
PCI DSS	The <i>Payment Card Industry Data Security Standard</i> is an attestation provided to customers using Secure Access to help enable compliance with PCI DSS. PCI DSS was developed to encourage and enhance payment card account data security and facilitate the broad adoption of consistent data security measures globally. It provides a baseline of technical and operational requirements to protect payment account data.
HIPAA	The Health Insurance Portability and Accountability Act of 1996 is an attestation that Secure Access helps enable customers to comply with HIPAA requirements. HIPAA required the Secretary of the U.S. Department of Health and Human Services (HHS) to develop regulations protecting the privacy and security of certain health information.

During the discovery phase of a migration, you should explore what security

standards are being leveraged by the vendor you are migrating to, as well as what data will be handled or stored by the vendor. How data is being stored and ingested with the use of AI and large language models is an area of growing interest and concern for cloud applications.

Migration Requirements

If you have ever had the opportunity to watch a space shuttle launch, you will likely have heard the prelaunch checklist that occurs before the launch. Inside the mission control center, as depicted in [Figure 3-5](#), engineers perform a series of checks before the launch and give a verbal “go/no go” as to whether they believe the shuttle is ready for launch. At the end of this series of checks, the flight director then uses this information to decide whether the launch should proceed.



Figure 3-5 Mission Control Center Overseeing a Successful Rocket Launch

This prelaunch checklist is critical to ensure a safe and successful rocket launch into space. The series of checks done before the launch could be likened to the discovery phase of a SaaS migration, where each phase of the discovery process is a check to ensure a successful migration.

By this point in the discovery process, you have thoroughly investigated your business needs, migration considerations, associated costs, migration type, features, configurations, integrations, network considerations, and security. Each of these areas is a tool that you can use to help define a set of requirements for ensuring a successful migration.

The list of migration requirements should specify all of the criteria necessary for a migration to a SaaS application to make financial sense and provide value to your business. It is important to note that this list will change depending on the application or organization completing the migration.

Think of the requirements as your guidepost to help you make the right decisions along the migration journey. You want to ensure that the SaaS application meets your core requirements and will continue to drive equal or greater value to your business than the application it is replacing.

Design and Planning

Imagine yourself as someone managing a data center migration. In the process of moving workloads to the new data center, you fail to properly plan for application and database dependencies for the workloads being migrated. After the migration, you begin to realize that workloads are failing due to critical dependencies not being accounted for. Consequently, you must roll back changes, which costs time and money to your organization.

Additionally, you find that some of the workloads that were migrated to the new data center did not support the new hardware they were running on, causing further outages and delays in the migration. Had the necessary time and thought been put into the migration plan, many of the issues that arose throughout the migration would have been avoided. The same can be true for a migration to a SaaS application. Even for simple workload migrations, having a plan for how the migration will occur can help ensure that steps aren't missed, causing further delays and costs. The second phase of the SaaS migration flow is design and planning. During the discovery phase, careful consideration was given to various areas of the migration to determine the move's viability and to help build a base set of requirements from which a plan could be made. With those requirements, you can begin to plan for network requirements, data migration strategies, and fallback planning, and then form a test plan to confirm that the migration is successful.

Network Requirements

The network design is one of the most critical pieces to a successful SaaS migration. Even with a perfect migration plan, the application will be used

only if you have a network that will deliver reliable and fast connectivity, because most SaaS products require a connection to a cloud-hosted service somewhere on the Internet. Imagine paying for a video streaming service but not being able to use it because you only have a dial-up connection to the Internet, so the video constantly buffers. You must ensure you have the proper network design to handle the services you plan to leverage before you begin using the service.

In the following sections, we will review some common areas that should be considered during the network design and planning step. This list is not exhaustive of all network changes that should be considered, and every network design is different. However, these items are likely to be considered for most SaaS deployments.

Connectivity

It should go without saying, but the migration will succeed only if your on-premises network is designed to allow for connectivity to the new SaaS application. Although this point may sound like a no-brainer, this task can sometimes be much more complicated than it sounds.

It is common for companies to be spread across multiple geographic locations and to support remote workers. Additionally, each location likely has unique network constraints. For example, branch locations may route specific network traffic back to the headquarters office before being routed to the Internet. Or each location may have different bandwidth constraints.

Regardless of the design, you should first understand the network requirements for the application you are migrating to. This information includes IP address ranges, ports, protocols, and bandwidth requirements. Some application payloads require significantly more bandwidth than others. For example, a real-time collaboration application supporting voice and video calls may require more bandwidth than an email application. Careful configuration would then be necessary to ensure that users can properly access the service from each location where your organization plans to use this service.

Connectivity can be tested and even deployed ahead of a migration. Network administrators can leverage many different tools to validate connectivity

across a network. This is often done as a part of testing the deployment ahead of the production deployment.

Quality of Service

Quality of service (QoS) is a configuration that allows for specific network traffic to be prioritized at the expense of other network traffic. Depending on the type of application being migrated, QoS network policies can help ensure that network traffic is treated with the proper priority. Some of the basic features that QoS provides are

- Low latency
- Bandwidth guarantee
- Packet buffering
- Traffic policing

Consider a migration to a SaaS service like Cisco Meraki SD-WAN. Meraki SD-WAN enables administrators to dynamically change the way that traffic is handled and routed on the network to ensure that WAN traffic is properly prioritized. Meraki SD-WAN can leverage QoS to tag traffic for prioritization based on current network demands to ensure optimal WAN connectivity and speed.

Note

It is important to note that QoS will benefit network traffic only on your local area network (LAN) because the Internet has no standardized QoS mechanisms.

QoS implementations vary from vendor to vendor. However, some of the components of Cisco's QoS implementation are

- **Classification:** Classification is the process of distinguishing one type of traffic from another based on access control lists (ACLs), Differentiated Services Code Point (DSCP), class of service (CoS), and other factors.
- **Marking and Mutation:** Marking is used on traffic to convey specific information to a downstream device in the network or to carry

information from one interface in a device to another. When traffic is marked, QoS operations on that traffic can be applied.

- **Shaping and Policing:** Shaping is the process of imposing a maximum rate of traffic while regulating the traffic rate in such a way that downstream devices are not subjected to congestion. Shaping, in the most common form, is used to limit the traffic sent from a physical or logical interface. Policing is used to impose a maximum rate on a traffic class. If the rate is exceeded, a specific action is taken as soon as the event occurs.
- **Queuing:** Queuing is used to prevent traffic congestion. Traffic is sent to specific queues for servicing and scheduling based on bandwidth allocation. Traffic is then scheduled or sent out through the port.
- **Bandwidth:** Bandwidth allocation determines the available capacity for traffic that is subject to QoS policies.
- **Trust:** Trust enables traffic to pass through the device, and the DSCP, precedence, or CoS values coming in from the endpoints are retained in the absence of any explicit policy configuration.

Not all SaaS applications or architectures require a QoS policy to ensure application performance. However, it can be a great tool to leverage if you already have network constraints and want to ensure that specific traffic is prioritized.

Bandwidth

Often, bandwidth can be confused with speed, and although the two are related, they have distinct differences. Bandwidth is a measurement of how much data can be transferred over a given period. Speed is the rate at which data is being transmitted.

Every network application consumes some amount of bandwidth. The amount of bandwidth required, however, will vary based on utilization, configuration, location, and so on. For example, in [Figure 3-6](#), you can see the bandwidth consumption of high-definition video calls using Cisco Webex.

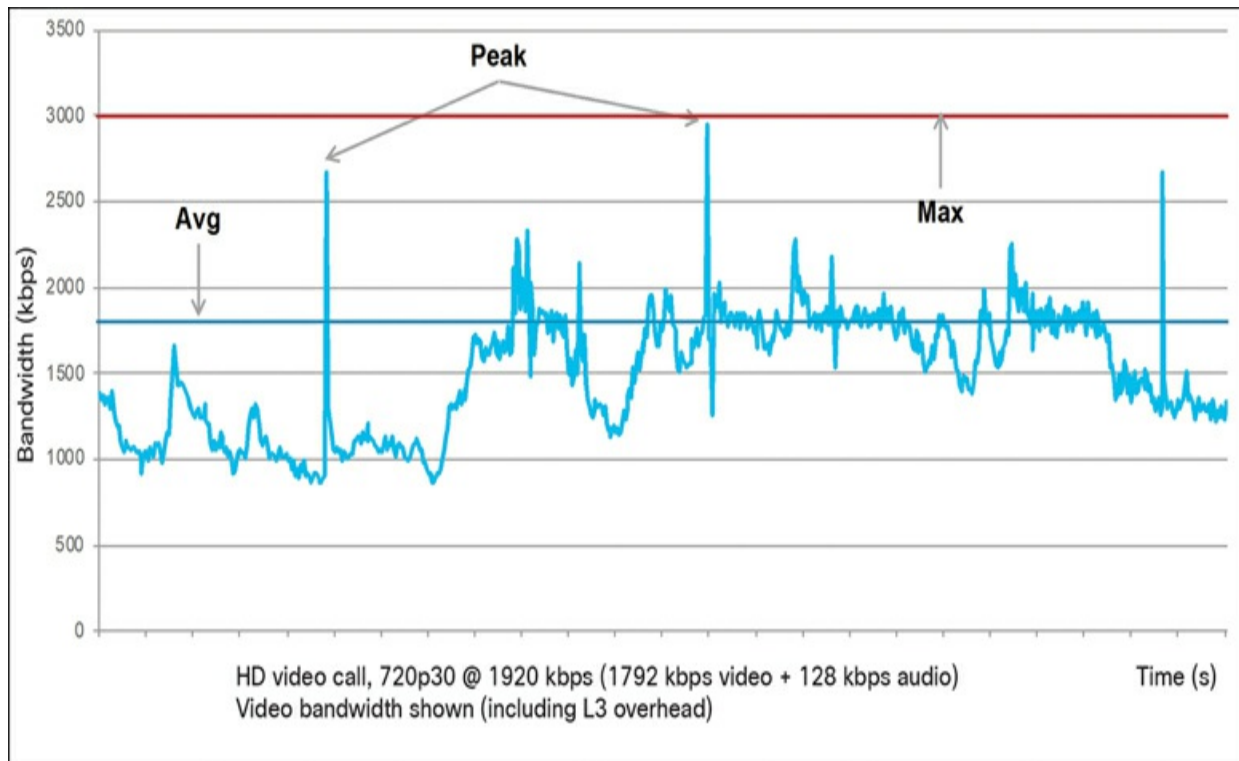


Figure 3-6 Bandwidth Utilization for High-Definition Video Call

Video traffic is bursty, meaning that the traffic is not sent or received at a consistent bit rate. This means that the bandwidth consumption for the video stream will vary throughout the call. Using this example, there are three important values to consider when designing for bandwidth consumption:

- **Average (avg):** The average bandwidth utilized over a period of time
- **Peak (peak):** The peak burst bit rate reached over the same time period
- **Maximum (max):** The maximum bit rate the device will reach

Each of these values represents an important measurement you can use when designing for the bandwidth component of a migration. Using the previous example, ideally, your network should be able to support the maximum bitrate required for a video call, but the average consumption will be slightly lower.

You should ensure that the bandwidth requirements of an application can be met by both the LAN and WAN portions of your network. It is more common that you will find that your edge/WAN bandwidth restrictions are the most limiting and the ones you will need to optimize for the most.

Virtual Private Network (VPN)

A virtual private network is an encrypted connection between two points. Most commonly, it is used to extend your corporate network connection to users across an Internet connection, allowing you to transmit data securely across the Internet. A VPN connection is commonly used for remote workers to allow corporate connectivity from users' homes or while traveling.

There are two types of VPNs:

- **Remote Access:** A remote access VPN allows devices outside a corporate network to connect securely. A device could be a laptop, phone, or tablet. The VPN could also perform security checks to ensure that the device attempting to communicate over a VPN meets a specified security policy.
- **Site-to-Site:** A site-to-site VPN is a secure connection between two locations. For example, a remote office connects to its corporate headquarters online. This connection extends the remote office's connectivity to access devices on the headquarters network.

Why should you consider a VPN when planning for a migration to a SaaS application? Most SaaS applications are entirely cloud-hosted and reachable with simply an Internet connection. However, you may not want users to access that application off your corporate network. This decision could be based on security reasons, where you want to ensure that all traffic to and from the SaaS service is encrypted. You could leverage a VPN policy that would enforce network traffic for specific applications to be routed through the VPN while ignoring other types of traffic.

It is worth noting that security models like Zero Trust Network Access (ZTNA) leverage a VPNless configuration that always requires users to authenticate to the application regardless of what network they are on. This type of security configuration may result in different types of testing that need to be performed.

Data Migration

With the rise of cloud computing and many services and applications shifting

to a SaaS model, migrating data to cloud-hosted services has become increasingly popular. Cloud storage and computing costs have become more competitive over the years. Also, cloud computing brings with it the ability to scale rapidly compared to an on-premises storage solution.

With more applications shifting their workloads to the cloud, this transition introduces a concept called *data gravity*. The idea of data gravity is that as data sets grow, they tend to attract applications and services to them. Why? Because it is often easier to move an application to a larger data source than to migrate this data to be closer to an application. So, as data is being migrated to the cloud over time, that act is also causing other on-premises services to migrate to the cloud.

When you think about data migration in the context of a SaaS migration, there are a few different data migration types to consider:

- **Application Migration:** Moving an entire application, including its data, from one place to another
- **Database Migration:** Moving an application's database to a new location while the application or services leveraging that database remain in place
- **Storage Migration:** Moving an application's physical storage to a cloud-hosted storage provider

With any of these migrations, careful planning must take place to ensure that the data is migrated in a secure and nondestructive way. Often, vendors will have data migration plans in place for migrating data to their service.

Although these services can aid in planning and executing a migration, ultimately, you have the most knowledge about your data, its dependencies, and key stakeholders for that data. Therefore, you must combine your knowledge of this data with the vendor migration recommendations to produce the best plan.

One area that should be given special attention is ensuring all dependencies on that data are accounted for and notified of this migration. It is common for a data set to have many downstream consumers. The recommended approach is to track the stakeholder teams and integrations currently in place leveraging that data to ensure mutual awareness, consider migration factors,

and employ acceptance testing to keep the business operational pre- and post-migration.

Testing

A good migration strategy should always include a methodical testing plan. It allows you to confirm that a migration was successful, but more importantly, it can surface problems you may need to account for in your migration plan. A thorough testing strategy ensures that all business functions of an application are tested, from user experience to performance.

There are multiple ways to include testing in a migration strategy, from fully automated scripts used to test various functions of an application to manual tests performed by a human to test the user experience within an application. Every migration should be tested before being released to production to ensure that the desired performance, connectivity, security, and functionality are achieved. Consider the following test strategies when migrating to a SaaS application.

User Acceptance Testing (UAT)

The most basic form of testing that you could employ is to test your application with end users manually. This could be done by getting a group of power users who are familiar with this system or application and giving them various use cases to test. Their goal is to determine whether the system behaves as intended and desired and find any issues before go-live. While this is a more time-consuming test methodology, user acceptance testing is a great way to ensure an application operates how you expect. It may even catch items that automated testing could not.

Automated Testing

Automated testing has become increasingly popular in software deployments and could also be leveraged in testing a migration. Automated scripts could check anything from network connectivity to data migrations, test application APIs, and simulate user interactions (button presses, web navigation, and so on). So, depending on the type of application you are migrating to, consider using an automated test strategy to help validate the operation of your

application.

One example of an automated test you could deploy for an application migration is using a script to test network connectivity to a new SaaS application from various locations in your network. Ansible is a powerful open-source tool that allows you to write playbooks for the deployment and configuration of infrastructure and applications or even to configure and test network infrastructure. Ansible leverages a controller running the Ansible playbook to control managed nodes remotely in a network, as depicted in [Figure 3-7](#).

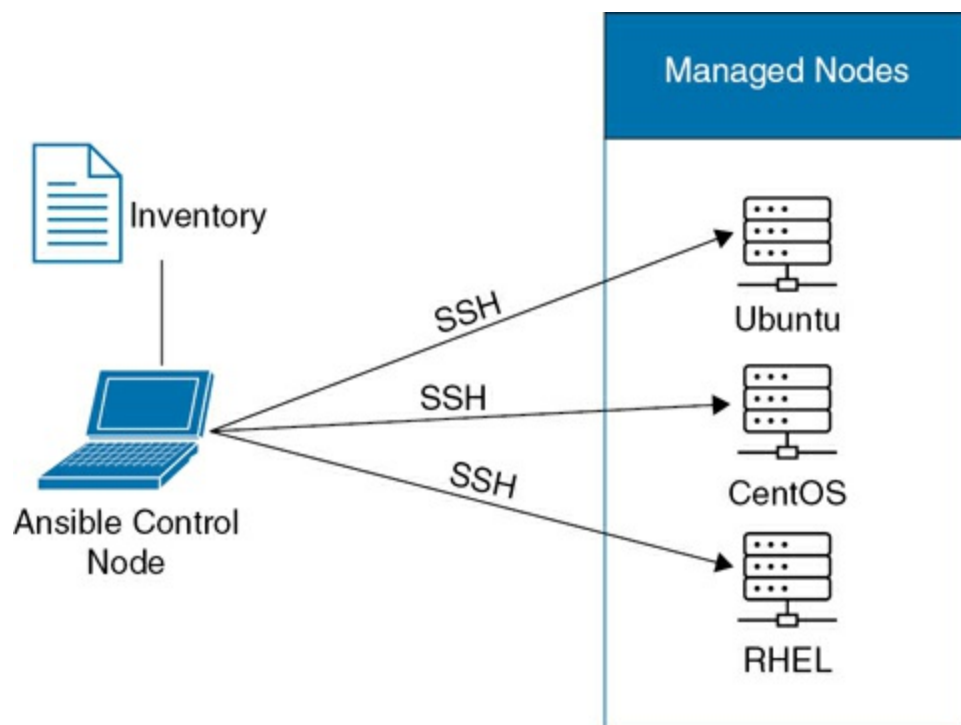


Figure 3-7 Ansible Architecture with Controller and Managed Nodes

The controller node running the Ansible script can be configured to connect to managed nodes and execute commands from them remotely. For a SaaS migration test plan, something like this could be leveraged to connect remotely to systems at various locations and then test connectivity from these systems to the new SaaS application. This method gives you a scalable, consistent, automated testing strategy to validate your network design.

While this is only a single example of an automated test you could leverage for a SaaS migration, it is meant to convey the power an automated testing

solution could have on a SaaS migration. Consider the type of application being migrated to investigate if there are opportunities for automated testing.

Application Metrics

Another great testing strategy is to leverage application metrics to compare the performance of the application you are migrating to with the metrics of the application you are migrating from—or, at the very least, confirming that the metrics of the application being migrated to meet (or, ideally, exceed) your business requirements.

Depending on the application being migrated, the SaaS vendor, and the environment being leveraged, this approach might not be an option. Most SaaS vendors and SaaS architectures do not expose performance metrics of the underlying architecture and compute systems used to host an application but instead rely on specific SLAs to guarantee the performance of their applications. However, some SaaS services have application-specific metrics that could still be leveraged to determine the application's performance.

Note

While SaaS vendors may not expose “under the hood” performance metrics, enterprises can devise network performance metrics against systems and APIs from SaaS products, run heartbeat checks, and leverage reporting and monitoring between their infrastructure and the SaaS vendor. Also, many SaaS vendors include practical information in their documentation about rate limits, infrastructure, high availability, business continuity plans, and so forth.

To see how application metrics could be used in testing to validate a SaaS migration, consider Cisco Webex. Cisco Webex is a SaaS all-in-one collaboration suite that gives users messaging, video meetings, calling, and more. Control Hub, the cloud portal from which the Cisco Webex application can be managed, configured, and monitored, is at the heart of this application.

As a part of a test plan when migrating to this application, you would want to perform test meetings and calls to determine if the audio and video in those meetings were of good quality. To determine this characteristic, you could

leverage the Cisco Control Hub analytics portal to view the media statistics for test calls and meetings to determine if, for example, packet loss, latency, and jitter were all within an acceptable range. Figure 3-8 shows an example of the Webex application metrics that can be seen from Webex Control Hub.

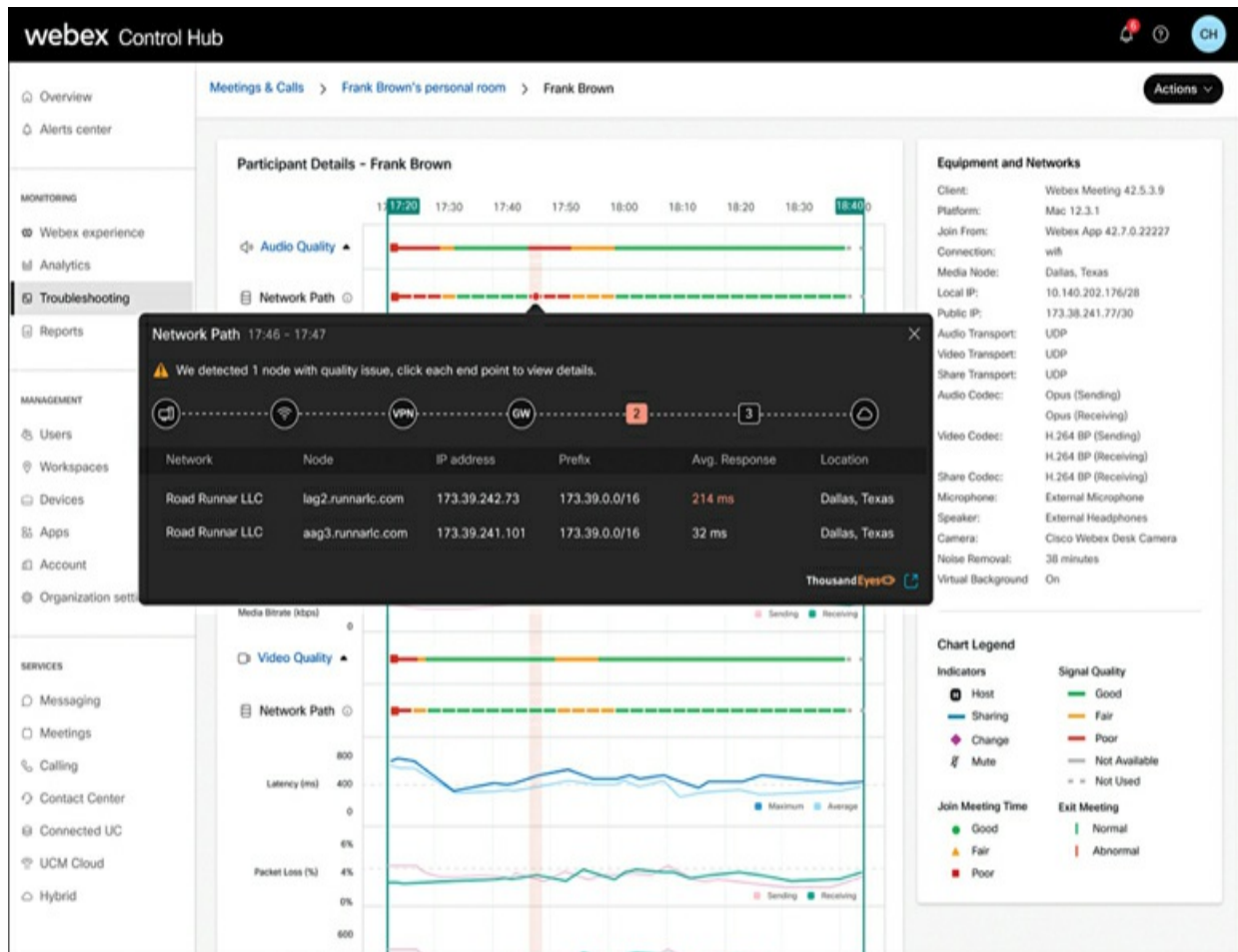


Figure 3-8 Webex Meeting Participant Network Metrics from Webex Control Hub

In this example, an administrator for this organization could leverage these statistics for test calls to determine if the application and underlying network meet expectations.

Security Testing

SaaS solutions often maintain strict security protocols because they are Internet accessible, multi-tenant, and may store highly sensitive data for

customers. But even though most SaaS vendors have a robust set of security standards and practices, it is good to confirm that your specific usage of a SaaS solution is secure.

You should investigate the security protocols used by the SaaS vendor you plan to migrate to. Often, SaaS vendors will detail the security standards they have in place to give their customers confidence in using their platform. The following are a few examples of security testing that you could perform:

- **Data Encryption:** Ensure that any data you transmit to and receive from a SaaS application is encrypted with strong encryption algorithms.
- **API Security:** If you are leveraging an API service for your SaaS application, consider what level of authentication and reporting you have for using this API.
- **User Authentication:** Users will likely be logging in to this service across the Internet. Consequently, you can test that the authentication configuration is strong enough to protect your users' credentials and prevent unwanted logins. When possible, implementing stricter login protocols such as single sign-on (SSO) and multifactor authentication (MFA) can improve your overall security posture as it relates to authentication.

It is also good to check whether the SaaS vendor performs its own application and network-level penetration tests, determine what countermeasures the vendor has in place for network risks, and understand what identity and access management (IAM) controls are in place, just to name a few.

Security best practices are often baked into SaaS applications, and minimal control or insights into these configurations are given to SaaS consumers. However, it is always good to confirm that the security standards you expect to be in place are. [Chapter 4, “Security and Privacy for SaaS,”](#) covers SaaS security practices in depth.

Fallback Planning

The point of planning for a migration is to minimize or, ideally, eliminate the risk of failure. However, not planning for failure is a mistake because only

some things can be controlled, and if life teaches us anything, failures and mistakes will happen. So, what is a fallback plan? A fallback plan is a strategy for rolling back changes made during migration to get a service up and running again.

Fallback planning is commonly leveraged in technology today. You may even do this yourself regularly without even thinking about it. For example, you decided to go on a camping trip for the weekend, and you planned to sleep in a tent. But in case of bad weather, you also found a nearby hotel you could go to if needed. The hotel in this example would be your fallback plan should the storm cause your first plan (sleeping in a tent) to fail.

Alternatively, consider an Internet modem with two Internet service provider (ISP) connections. One is via cable, and the other is an LTE cellular connection. If the primary cable ISP connection were to go down, the LTE cellular connection could be used as the fallback, as depicted in [Figure 3-9](#).

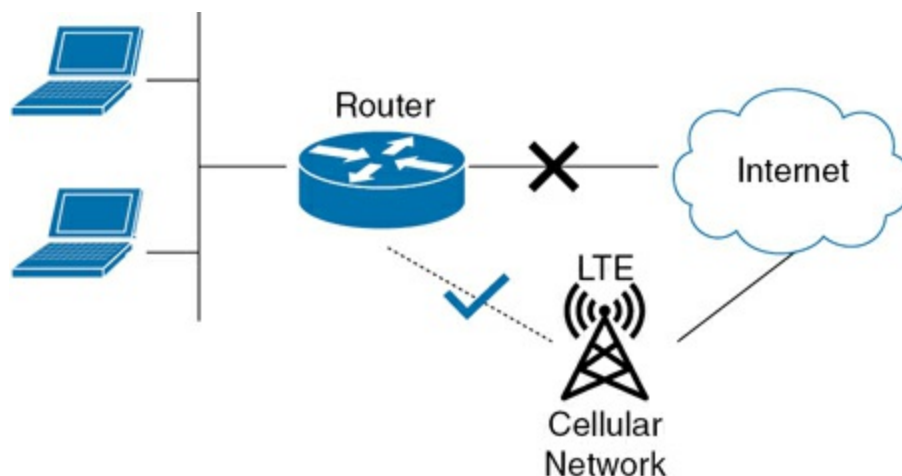


Figure 3-9 Network Topology Supporting Cellular Fallback

A fallback plan was created in both examples just in case the primary plan was to fail. The same should be done as you are making your migration plan. Consider what points in the migration could fail, and plan for how you would fall back should those steps fail for any reason. Ask yourself if you are able to migrate (or roll back) transparently to your users. Ideally, you will never need to use these plans, but having a fallback plan in place could save you time and possibly even money by preventing a business outage for an extended period.

Implementation

By this point in the migration, the discovery work to build requirements for a SaaS migration has been completed, and a comprehensive design and migration plan has been assembled. The next step is to take those plans and execute them.

The implementation phase of a migration could be compared to the construction of a home. For weeks or months, you have been discovering and planning where to build your house, what you want the home to look like, how many bedrooms and bathrooms you need, how large the home should be, and so on. Now, the day has finally come to break ground and begin the construction of the house.

During the implementation phase of migration, apart from following the plan that was formed during the design and planning phase, you have a few areas to consider that could help execute the migration successfully and with minimal disruption. These areas include leveraging a staging environment for testing, implementing a cutover strategy that best fits your needs, and leveraging tools provided by SaaS vendors to make the migration easier.

Staging Environment

Implementing a SaaS migration often requires you to deploy changes within a network to achieve the network requirements laid out during the design and planning phase of the migration. One way to reduce the risk of these changes disrupting service is to have a staging environment for testing the new SaaS application.

In the context of a SaaS migration, a staging environment is an isolated test area that you can use for the preproduction testing of an application. You can see such an example in [Figure 3-10](#). In this staging environment, you can make the necessary network changes required for leveraging a SaaS platform, perform any security testing that you need, and validate the usage and performance of an application in a low-risk or controlled fashion.

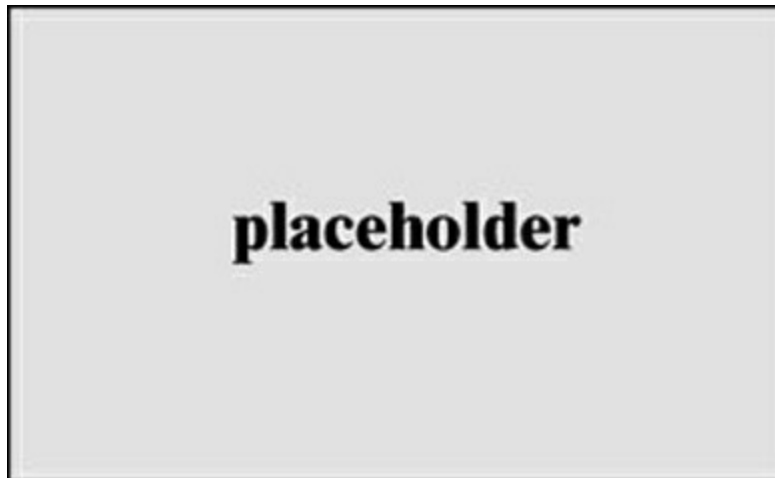


Figure 3-10 Example of a Staging Environment That Mirrors Production

There are two primary benefits to having a staging environment for a SaaS deployment. The first is that it allows you to test an application and associated network and architecture changes required for the application before those changes are made in production. By testing them in the stage environment, you may be able to find bugs or issues you hadn't planned for so that when it comes time to make those changes in production, you are prepared. The second benefit is that this staging environment can be leveraged even after the SaaS application has been deployed and adopted in your organization. By having a staging environment, you can potentially leverage it to test upcoming features for your SaaS application or debug issues that users may be running into in an environment isolated from production so that rapid changes and tests can be performed.

Some SaaS vendors provide staging environments as a part of their subscription, knowing that customers generally need it for successful adoption of the service. However, this is not always the case, and may incur additional costs. Having an isolated test environment means that you will need additional hardware for this environment to run in. This additional hardware will require maintenance and upkeep similar to a production environment, although with less impact if it breaks.

Application Cutover Strategy

Multiple approaches can be taken when it comes time to begin migrating

users to a SaaS application. The approach to moving users to a SaaS application may change depending on the application type, the number of users being migrated, the training required for the new application, and several other factors. There are three main approaches to achieving this strategy.

Hard Cutover

The most straightforward approach to achieving your cutover strategy may be to set a hard cutover date on which all users will be moved from the on-premises application they were using to the new cloud SaaS application. While this approach simplifies the overall migration plan and change management necessary to get users to a new platform, it is the most disruptive and high-risk change strategy. [Figure 3-11](#) depicts an example of a hard cutover where users no longer have access to the on-premises application and have access only to the new SaaS application.

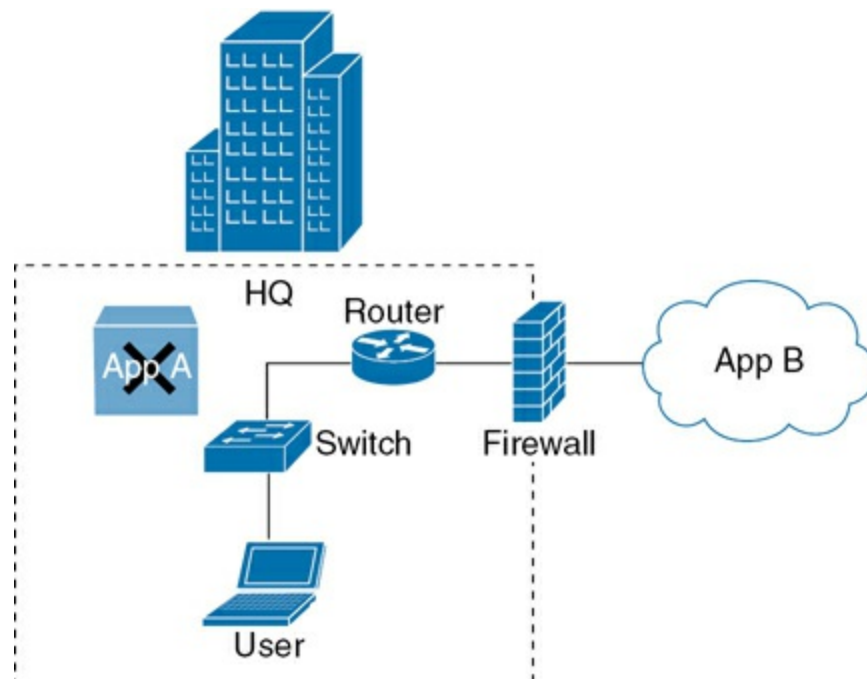


Figure 3-11 Example of a Hard Cutover Strategy Where Users Lose Access to the On-Premises Application (App A) and Are Migrated to the New SaaS application (App B)

In the perfect migration scenario where all users are well trained on the new

platform, testing of the new application has revealed no issues, and you have a high degree of confidence that the configuration and setup of the application are ready for production, a hard cutover strategy may work well.

A hard cutover strategy may also be an easy option for smaller migrations or smaller organizations where the number of migrated users is small.

It is also possible that a hard cutover is the only available option for you because your source application may reach an end-of-life date or end of service, forcing you off by a specific date. However, this strategy has some disadvantages:

- **Service Disruption:** Depending on the business criticality of the application, migration may result in service disruption for your business or internally to users. If users are still getting familiar with how to use the new application, completing tasks may take longer as they attempt to learn it. Or a potential issue with the application may be discovered in production. Users may have to wait for the problem to be resolved before continuing their work.
- **Difficult to Fallback:** In a hard cutover strategy, the assumption is that you will move to the new application without any plan of moving back. The reason may be the data that was migrated to the cloud SaaS application, which can reside only in a single tenant—either on-premises or in the cloud. Leveraging the data in two locations would result in out-of-sync data.

In summary, a hard cutover strategy is one of the most straightforward approaches to migrating users to a new application. However, it comes with inherent risks. Often, this is not a good approach for larger organizations that may require more time to migrate users to avoid service disruptions.

Parallel Applications

Another strategy that could be employed is to run both the new and old applications in parallel. In this scenario, both the source and destination applications you leverage run in production simultaneously, allowing both applications to be used.

This strategy is mainly used when an evaluation is being made between two

applications so that a user can leverage both applications to draw comparisons between them. Additionally, it allows you to collect metrics from both applications as another point of comparison.

Another reason a parallel strategy might be used is to give users a fallback mechanism should the new SaaS application run into any issues early on. After a period of time running both the new and the old application, if things are running smoothly on the new application, the old application can be decommissioned. It is highly recommended in this scenario to keep the parallel application window time boxed to a certain communicated duration to manage user expectations and drive adoption.

Figure 3-12 depicts an example of a coexistence strategy where users can access both the on-premises and SaaS applications.

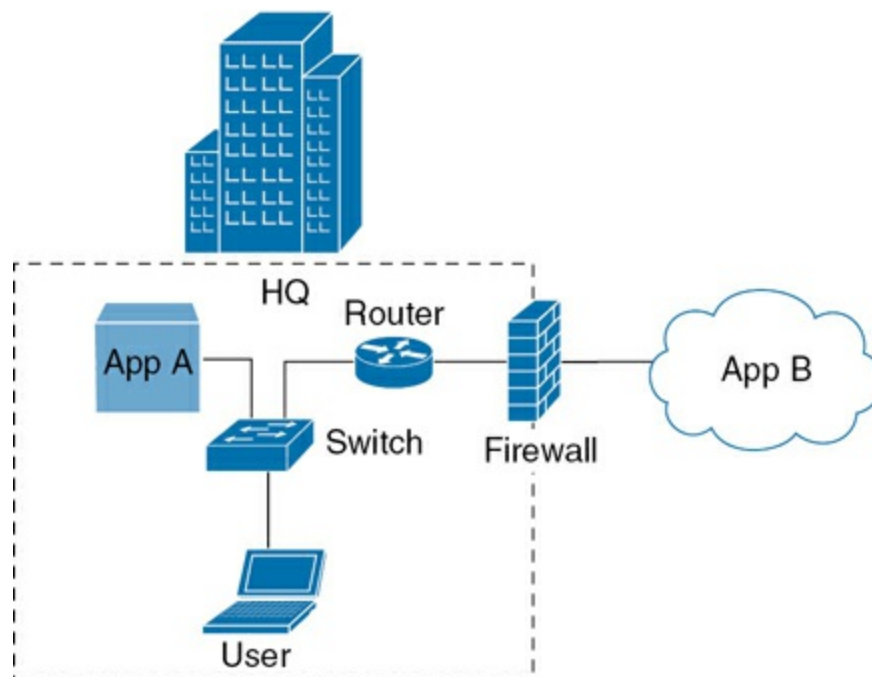


Figure 3-12 Example of a Coexistence Migration Strategy Where Users Can Access the On-Premises Application (App A) and the New SaaS Application (App B).

This strategy is similar to the hard cutover method. However, it does not restrict a user from using either application. Both can be used until a decision is made to limit access to the old application. The following are some disadvantages to this method:

- Both on-premises and SaaS applications must be supported.
- This approach can slow the adoption of the new application.
- Users may be confused on which application to leverage.
- There is a cost to maintain both applications.

In summary, a parallel migration strategy can work well when a comparison needs to be made between two applications simultaneously, or if there is no rush to decommission the old application. However, it can lead to confusion for users who may not know which application to use or can slow the adoption of the new application with users who might want to stay on the old system.

Staged Migration

The final migration strategy is a staged approach where users gradually move to the new application. Moving the users to the new application in stages limits the potential disruption to the users who are being migrated at any given time. This approach is generally preferred due to the high potential of service disruption to end users, and users need to have time to learn and adopt the new application. It also allows you to test the new application in smaller waves to look for any possible issues that were not caught during the deployment and testing of the application.

For example, when rolling out a multifactor authentication solution, a good approach may be to do a staged migration and first deploy more technical teams, who may have likely used MFA for other services, and then follow this up with another user group like Sales, who may require additional training and hand holding after the migration.

[Figure 3-13](#) depicts an example of a staged migration where some users have access to the new SaaS application and others have access only to the on-premises application.

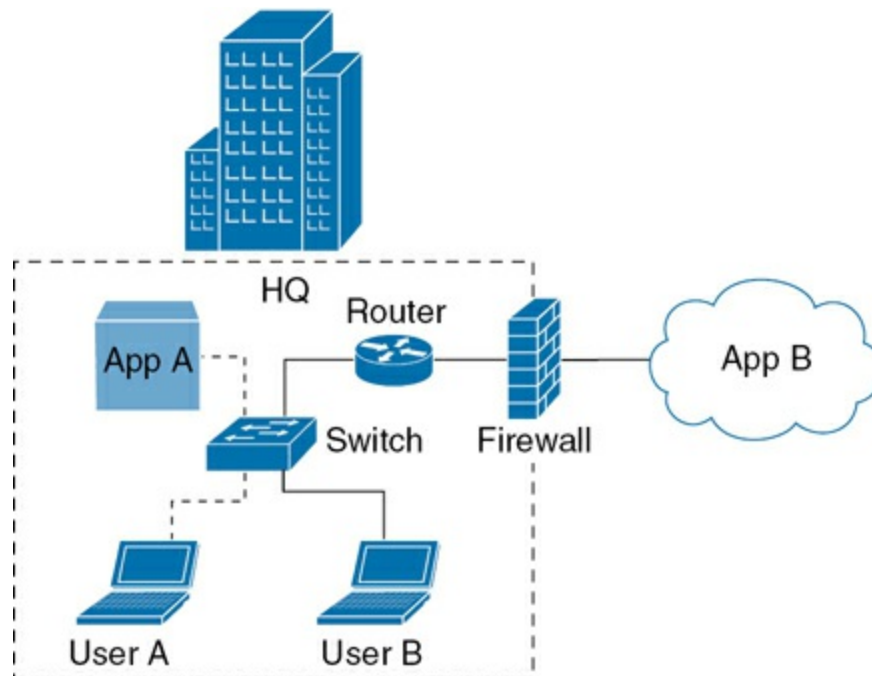


Figure 3-13 Example of a Staged Migration Strategy Where Some Users Have Access Only to the On-Premises Application (App A, and Others Have Access Only to the SaaS Application (App B).

A staged strategy is a common approach for migrating email from an on-premises mail server to a cloud provider. Because mailboxes can be moved individually and relocating an entire organization's mailboxes can be very time-consuming, the migration often cannot be done over a single weekend or maintenance window. As a result, most organizations will migrate mailboxes in batches, giving them a longer runway for completing the migration. This is just one example of a SaaS migration that may leverage a staged migration approach.

The following are some of the potential disadvantages of a staged migration:

- Cost of maintaining two applications
- Support required for two applications
- Longer migration time

In summary, a staged migration strategy is a common approach for larger organizations requiring more time to complete a migration. Completing it in smaller stages minimizes the overall disruption and risk. However, this

approach does extend the migration time and potential cost while maintaining both applications.

SaaS Provider Migration Tools

SaaS applications are purpose-built and are often the evolution of a prior application that may have been distributed via software packages licensed to run on-premises. Because of this, most SaaS vendors have a deep understanding of what market segment their application will attract and who their competitors are. This market knowledge often allows SaaS vendors to develop migration-specific tools to help customers during a migration.

In the implementation phase of a migration, leveraging migration tools provided by a SaaS vendor can often expedite the migration time. These migration tools are sometimes custom-built to aid with migration from specific platforms they know most customers use. There will always be parts of a migration that cannot be automated, but finding even small ways to speed up, automate, or simplify a migration is well worth the effort. SaaS vendors also very commonly have adoption material (documentation, training, and so on) to enable customers to successfully onboard and adopt to the new service. The ability to retain customers is critical to SaaS vendors' success, so the best vendors will always provide content to enable customers to successfully get their users utilizing their platform.

To give you an example of what this could look like, Cisco Webex Calling is an enterprise SaaS solution enabling cloud-based phone systems for businesses. Instead of managing phone system infrastructure on-premises, Cisco Webex hosts that infrastructure in the Webex Cloud. To enable customers to migrate users from on-premises voice systems to the Webex Calling Cloud application, Webex has developed features within the application to bulk import users from the on-premises infrastructure to the cloud, carrying over user details, phone numbers, and locations. The purpose of this type of tool is to make the migration as simple as possible and to avoid mistakes that may happen if this data were manually migrated.

Value Realization

The last phase of a SaaS migration is value realization, which is an ongoing process whereby customers evaluate the return on investment of a given product. After migrating to a new application, it is time to reap the benefits of that application for your business. For a SaaS application, customers evaluate if the cost and service are justified for the value received.

During the design phase of the migration flow, determining the business value for moving to a SaaS application was defined. This definition of business success is used to gauge the effectiveness of a migration. An essential process in the lifecycle of an application to ensure that business objectives are being met is determining and quantifying the business value.

Often, SaaS providers will have go-to-market strategies involving value realization, helping their customers migrate to their platforms and realize the value of those platforms for their business. It is an essential part of SaaS customer success to help businesses see a clear benefit to moving to their platform. Once a business is on the platform, helping them see and feel those benefits is just as important.

Value realization is an iterative process. While the prior phases of a migration had defined start and end points, once you have reached the value realization phase, you will continue in this phase for the duration of the application's usage. One way to think about value realization is as a continuous loop, as shown in [Figure 3-14](#).

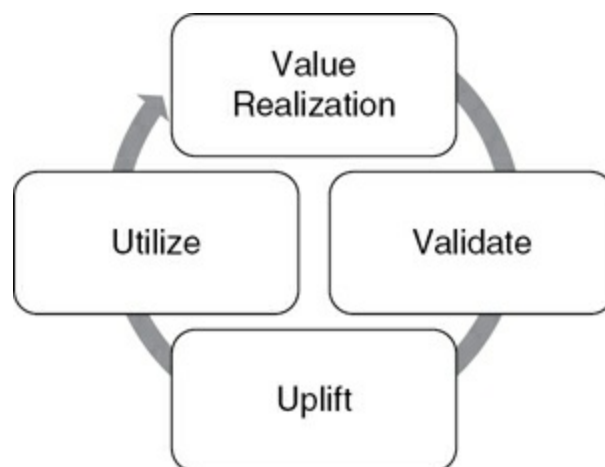


Figure 3-14 Value Realization Lifecycle

Validate

Once the migration has been completed and the value from the new application is realized, it is essential to validate whether the application continues to meet your business needs. Over time, company objectives change, business strategies evolve, and what was a priority for you a year ago may not be essential today. Those shifts in business are why it is crucial to routinely validate whether there is a clear ongoing value to the applications you are currently leveraging.

During the value realization lifecycle's validation phase, you redefine or clarify your criteria for business value and success. Is the way you leverage an application today helping you achieve your objectives and providing a return on investment?

SaaS vendors are often interested in and involved in this step. They want to ensure that customers using their platform remain loyal to their platform. To do that, they must partner with their customers to help them achieve successful business outcomes.

Uplift

After validating your criteria for success for a given application, it is essential to identify any unrealized value gaps in the usage of that application. Unrealized value gaps are where your usage of an application does not align with your success criteria for that application.

For example, perhaps you identify that one of the success criteria for a task management application you recently migrated to is to give you greater visibility into project statuses and estimated completion times. However, after looking at the application's usage, you realize that users are not using the tools and features of that application properly to help you realize these goals. This is a value opportunity whereby users can be trained on the tools and features of the application, and adoption strategies can be put into place to ensure users can leverage them.

Perhaps new features have been released for an application that is not currently being utilized. However, these new features could improve several

key metrics you track. Again, this is a value gap that could be addressed.

Uplift is the phase where opportunities for improving the usage of an application are found to enhance the value of an application.

Utilize

After identifying the value gaps, you can take actions to address those gaps and ensure the proper utilization of the application. The goal of this step is not purely adoption or the number of users who use a feature. Instead, the purpose is to help users realize the value in an application, set of features, process, and so on. Who are the key users who should be leveraging the application? Are they? What are the key features and tools within an application that your business realizes value from? Are they being utilized? The utilize step is creating a roadmap to address the answers to those questions.

When the utilize step is complete, you are back to value realization. Again, in this phase, you quantify the business value of an application. Repeating this cycle helps ensure that the correct measurements are being made and that a consistent return on investment is realized.

Common Migration Challenges

The SaaS migration flow outlined in this chapter is meant to be a repeatable framework that can be leveraged to execute a migration with minimal disruptions and challenges. However, not every migration will be completed without issues. Although there is no silver bullet to avoid problems during a migration, you can take steps to help prevent them.

Ideally, thorough planning and design should be done before the migration to account for possible issues and so that steps are not missed or forgotten. From discovery to implementation, each step in the migration flow laid out in this chapter can be used as a resource to help avoid common mistakes in a SaaS migration.

Another way to help avoid complications during a migration is to make yourself aware of common challenges that are seen during a migration. The

rise in SaaS spending and adoption means that almost every business is moving workloads to the cloud. That means a wealth of information is available to learn about common migration issues, best practices, value realization, and much more.

Let's look at some of the common migration challenges you may face when moving to SaaS.

Network Changes

Modern data centers are complex, especially for larger enterprises. A typical data center comprises computing, storage, and networking resources, all working together to allow businesses to operate efficiently. The networking resources within a data center are often robust, layering switching, routing, and security equipment together. Managing these data centers can be complicated, especially since a company's resource and access needs are constantly evolving.

Approaches like software-defined networking (SDN) have arisen to allow administrators to scale and maintain data center networks, leveraging network policies and automation to speed up deployments and reduce the risk of mistakes.

With the inherent complexities common to data center architectures, network changes required for SaaS deployments can be challenging for organizations. Because SaaS applications are provided through a cloud service, organizations must define network policies to allow for connectivity through their data center fabric, across the network edge, to the Internet. These changes must be handled in a way that does not break existing services and does not cause any security risks for the organization.

SaaS applications have defined network requirements, describing precisely what access is required to leverage their service. Each company must determine how to make those requirements work within its environment while conforming to security best practices.

The following are some common network challenges that are faced for SaaS deployments:

- **Inconsistency:** It is uncommon that a single network change must be made to allow for connectivity to a given host. Often, companies are spread across many different locations, each with its own network policies depending on the requirements of that location. If network changes are not addressed consistently, this deployment can lead to other locations having inconsistent experiences accessing the SaaS application.
- **SaaS Provider Network Changes:** Most SaaS providers are hosted on a public cloud, such as Amazon Web Services (AWS), Google Cloud Provider (GCP), or Microsoft Azure, to name a few. As SaaS providers expand or update their services, it is not uncommon for the network requirements for that service to grow or shrink along with the service. This means that additional IP ranges may be required to access the service. New features to a SaaS service require other network ports to be opened or even making changes to keep up with the latest security enhancements to ensure a greater security posture. Keeping up with these changes can often be challenging if SaaS providers do not communicate them effectively with their customers.
- **Bandwidth and Latency:** Network changes to grant access to a given SaaS application are only one piece of the puzzle. Some SaaS applications require minimal latency or large bandwidth requirements to work correctly. Real-time media applications used for voice and video calling are one example. These applications require low latency and higher bandwidth than typical web traffic. Defining network policies using tools such as QoS may be required to ensure that traffic for these specific applications is prioritized appropriately.

User Adoption

When considering user adoption, you have two main buckets to consider. The first is application adoption. This bucket measures whether users leverage the application they were migrated to. The simplest form of measurement for this could be active users over the past 30 days. What percentage of the application's target users are logging in to use it? The second bucket to consider for adoption is feature usage. This bucket measures whether your users leverage the application in the way you expect them to.

An effective migration strategy will measure both areas because each indicates different problems. If you find that a group of users has never logged in to the new application, the reason could be network connectivity issues where they can't reach the new application and, therefore, cannot log in. Or perhaps it's a communication issue, and they were not informed about the new application or how to log in. In either case, the migration team can use the login metrics to identify these types of issues, explore the cause, and then address the adoption concern. The same is true for feature usage metrics. Using these types of metrics can help you understand how users are using the application. Perhaps the new application has specific tools or features that have extensive business value. You would want to track the usage of this feature to ensure it is adopted at scale. Lack of usage could indicate a need for training, an issue with the application not working correctly, or, again, a communication issue that led to users not knowing what features to use.

Most user adoption issues can be addressed up front with clear communication about the migration to a new application and the expectation for users to use this application. Metrics help you find any communication gaps early in the adoption phase to help ensure that users have a smooth onboarding experience.

Security Concerns

Although SaaS applications can have many benefits for organizations, they also come with many security risks that IT organizations must now contend with. SaaS vendors take on most of the security responsibilities that IT was historically accountable for with on-premises applications. However, it is unwise to assume that every SaaS vendor has the same security standards and practices in place to protect its customers' data. During a migration, careful consideration should be given to ensure that the right level of security is in place to prevent bad actors from accessing your data. Now, let's look closer at some common security risks for SaaS applications.

The Need to Prevent Unauthorized Access

Proper IAM configuration along with using authorization systems like role-based access control (RBAC) can help to ensure that the right personas get

the right level of access. The last thing that you want is for bad actors to somehow get privileged access to a system that they should not have access to, allowing them access to confidential information. Users should always be granted access to the specific data, components, or view required for their job role to reduce the threat surface available to each user persona.

Loss of Data

Leveraging a SaaS application also means less control and visibility into your data. If data loss were to occur, this scenario could result in financial losses or even legal repercussions. This also includes disaster recovery scenarios where a SaaS vendor may have a data center or network outage in a region, causing access to go down. Whereas some SaaS vendors may have a robust disaster recovery process to ensure their service's continuous uptime and no data loss, others may not. Therefore, careful consideration should be given to how a SaaS vendor manages its data and handles outages.

Vulnerability Management

SaaS vendors are responsible for identifying and remediating security vulnerabilities in their software development lifecycle and hosting environment. As a consumer of a SaaS application, you rely on the security practices that are put in place by that vendor to ensure that vulnerabilities are correctly handled. Even a single vulnerability in a SaaS application could give attackers an entry point to your organization's data. You should understand how the SaaS vendor handles the responsible disclosure of vulnerabilities to its systems.

API Security

Most SaaS applications come with APIs to programmatically interact with the application. APIs can often read, write, modify, and delete data they can access. These APIs must have proper security protocols for authorization and authentication and role-based access controls to ensure that APIs can be accessed only by the right people. For example, an end user of an application should have access to an administrative API that would allow you to make configuration changes to an application. This is an example of role-based access control.

It is also important to determine whether the APIs are granular enough to provide the specific details that you need and also whether you have the necessary resources to leverage those APIs. This could be environments to run, store, and test code and a secure key storage system for storing API credentials.

Shadow IT

One of the biggest challenges that IT organizations face when it comes to SaaS applications is dealing with Shadow IT. Shadow IT is the use of an application within a company without IT having any direct knowledge or consent to use the application. Because SaaS applications are so easy to onboard, depending on the number of users who decide to purchase an application, the purchase can be small enough for departments or individuals to buy without needing funding approval.

The risk is that many applications now in use within an enterprise have not been through any sort of standardized security screening or compliance checks to ensure that the use of those applications is safe for both employees and potentially for customer data. Without awareness of the usage of applications, IT cannot put into place security protocols for monitoring this service and employing security measures to safeguard company data.

Organizations need visibility into the usage of cloud applications within their organizations to secure and mitigate risks for their organization. Any business that is embracing SaaS strategies should also have good governance and oversight as to what applications are allowed and accessible on the network. Cisco Umbrella and Cisco Cloudlock are examples of tools that IT organizations can use to improve visibility into Shadow IT applications within their organization to define security policies around those applications. [Chapter 9, “Security: Cisco Umbrella and Cisco AI Defense,”](#) covers the Cisco Umbrella application in more detail.

Another reason that Shadow IT applications pose a challenge for a SaaS migration is inflated costs. Multiple groups within an organization could purchase their own instance of an application, being billed individually. A more consolidated approach could lower spending costs and improve the overall security posture of the organization.

Summary

Migrating to a SaaS application is a common task for most organizations as they look to reduce operational costs, reduce the demand for IT to support applications, and quickly scale their architectural footprint. SaaS is pervasive in the industry and is expected to grow in the coming years.

In this chapter, we covered a four-step migration plan that leads you through discovery, design and planning, implementation, and value realization. Each step provides distinct insights and builds on the prior steps to create a pathway for a successful migration. In the discovery phase, you determine the readiness and value of migrating an application to SaaS. In the design and planning phase, you create a detailed execution plan, which you can then use during the implementation phase to make the migration a reality. In the final step, you explore the idea of value realization: determining whether the migration is beneficial for your business.

We also looked at some common challenges when migrating to a SaaS application and some possible ways to avoid those challenges. The SaaS industry, although mature, is still learning and adapting to the needs of enterprises and adapting to new security standards to protect customers' data. It is up to SaaS consumers to decide on the best SaaS application for them based on the business value that the application can provide and the security and reputation of the vendor selected to secure their data.

This chapter should give you a deeper understanding of the essential steps to migrate to a SaaS application. Adopting a migration framework can help you avoid common migration mistakes, ensure a smooth transition between applications, and improve the adoption of a new application.

References

- SaaS Growth Trends: <https://productiv.com/blog/2023-state-of-saas-series-while-companies-make-progress-cutting-costs-previous-investments-and-growth-of-shadow-apps-like-chatgpt-challenge-efforts-to-manage-saas-spend/>
- Nicole Cowell, SaaS ERP Systems vs. On-Premise—What's the

Difference? March 2022: <https://www.xperience-group.com/news-item/on-premise-vs-saas-erp-systems-whats-the-difference/>

- Global Certifications:
<https://www.cisco.com/site/us/en/products/security/secure-access/compliance.html>
- IBM, Cost of a Data Breach Report 2025:
<https://www.ibm.com/reports/data-breach>
- Johnny Page, 46 SaaS Industry Stats and Insights for 2024 and Beyond:
<https://www.saasacademy.com/blog/saas-statistics>
- Ejona Preçi and Peter H. Gregory, SaaS Security Risk and Challenges, July 2022: <https://www.isaca.org/resources/news-and-trends/industry-news/2022/saas-security-risk-and-challenges>
- Quality of Service Configuration Guide, Cisco IOS XE 17.18.x (Catalyst 9600 Switches):
https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9600/software/18/configuration_guide/qos/b_1718_qos_9600_cg/configuring_qos.html

Chapter 4. Security and Privacy for SaaS

Software-as-a-Service (SaaS) environments hold sensitive data, connect users across geographies, and must meet an ever-evolving landscape of security threats and regulatory requirements. This chapter will equip you with the knowledge and hands-on skills you need to secure SaaS platforms effectively. We will explore what makes SaaS security unique, which types of data are most at risk, and the threat models that every security professional must consider. We also will examine the influence of major compliance frameworks and regulations (including FedRAMP, GDPR, HIPAA, and CCPA) and how they shape provider and customer responsibilities.

We will describe how to apply industry best practices and frameworks such as ISO/IEC 27001 and NIST to strengthen SaaS security posture. We'll work through practical techniques, such as data partitioning, tenant isolation, encryption, and Zero Trust architectures, that are essential to safeguarding multitenant environments.

We'll also focus on identity and access management (IAM) concepts (from role-based access control to multifactor authentication) and explore how SaaS security posture management (SSPM) and cloud access security brokers (CASBs) can enhance visibility and enforcement. Finally, we'll explore how to build incident detection and response workflows, conduct regular security audits, and continuously improve security programs in response to emerging threats.

By the end of this chapter, you will not only understand the security and privacy challenges unique to SaaS environments but also gain the ability to design, implement, and manage robust, scalable security controls that meet

both business and compliance needs.

SaaS Security Basics

The shift from traditional on-premises software and applications to cloud-based SaaS solutions brings forth a new set of security challenges and complexities. Let's explore the foundational aspects of SaaS security, focusing on the protection of data, understanding regulatory impacts, and addressing data sovereignty. Understanding these basics is critical not just for maintaining the integrity and confidentiality of data but also for building trust with users and adhering to compliance and legal obligations.

Data Protection and Privacy Concerns

Data protection is a critical component of SaaS security, including how data is handled, stored, and protected. Different types of data are stored and processed in SaaS environments:

- **Customer Data:** This type of data includes personal and sensitive information about users, such as contact details, financial information, and personal preferences.
- **Operational Data:** This type of data pertains to the operational aspects of SaaS, including but not limited to performance metrics; usage statistics; and authentication, authorization, and accounting (AAA) logs.
- **Metadata:** This is data about other data, which can include configuration settings and user behavior analytics.

As you use cloud services (including SaaS), understanding the potential security threats is crucial for maintaining data integrity and trust.

Common Threats to Data Security in SaaS Platforms

SaaS solutions also expose data to many security risks due to their inherent nature of being accessible over the Internet. Now let's explore the common threats to data security in SaaS platforms.

Data Breaches

Data breaches are perhaps the most notorious threat facing SaaS environments. If you do a Google search on any given day for “data breaches in the cloud,” you will get numerous hits. To find better examples, you can examine the Vocabulary for Event Recording and Incident Sharing (VERIS) Community Database (<https://github.com/vz-risk/VCDB>). This database is a publicly accessible repository of security incidents and breaches. It’s part of an initiative by Verizon and other contributors to improve the understanding and management of cybersecurity risks. The database uses the VERIS framework, which is a structured language designed to provide a consistent, systematic approach to recording and analyzing security incidents. The information gathered and shared through the VERIS Community Database supports the development of industry reports, most notably the Verizon Data Breach Investigations Report (DBIR), which analyzes trends in security breaches and offers insights into how organizations can better protect themselves against future attacks.

There are many root causes for a data breach. The following are some of the most popular:

- **Weak Authentication:** Insufficient authentication processes make it easier for attackers to gain unauthorized access. You will learn about identity and access management, as well as best practices for managing IAM in a SaaS environment later in this chapter.
- **Vulnerabilities in Software:** Unpatched software can allow attackers to exploit vulnerabilities to exfiltrate and steal data.
- **Human Error:** Simple mistakes by employees, such as misconfiguration of privacy settings or mishandling of data, can lead to breaches.

Data Leakage

Let’s expand on that last element in the previous section—human error. Unlike data breaches led by attackers, data leakage typically involves accidental exposure of data due to errors or negligence, which can be just as damaging. Examples include misconfigured cloud storage and insecure APIs.

Incorrectly configured security settings on cloud storage can expose data to the public. SaaS applications often interact through APIs, which, if not secured properly, can be a channel for data leakage.

Case Study: Data Exposure Due to Misconfigured Cloud Storage

A fictitious mid-sized healthcare provider, NCHealthData Inc., provides a cloud-based SaaS platform to healthcare institutions and hospitals to manage patient records more efficiently. Many hospitals chose this platform for its scalability and the promise of enhanced data security and compliance with the Health Insurance Portability and Accountability Act (HIPAA). However, a crucial oversight in the configuration of cloud storage settings led to a significant data exposure incident.

The Incident

Approximately six months after the transition to the cloud, a security researcher discovered concerning data in the dark web. Attackers stole data because of a publicly accessible cloud storage bucket belonging to NCHealthData Inc. The bucket contained sensitive patient information, including names, addresses, medical histories, and treatment details. This data was supposed to be secured and encrypted, accessible only to authorized personnel and secured against external access. [Figure 4-1](#) illustrates the attack.

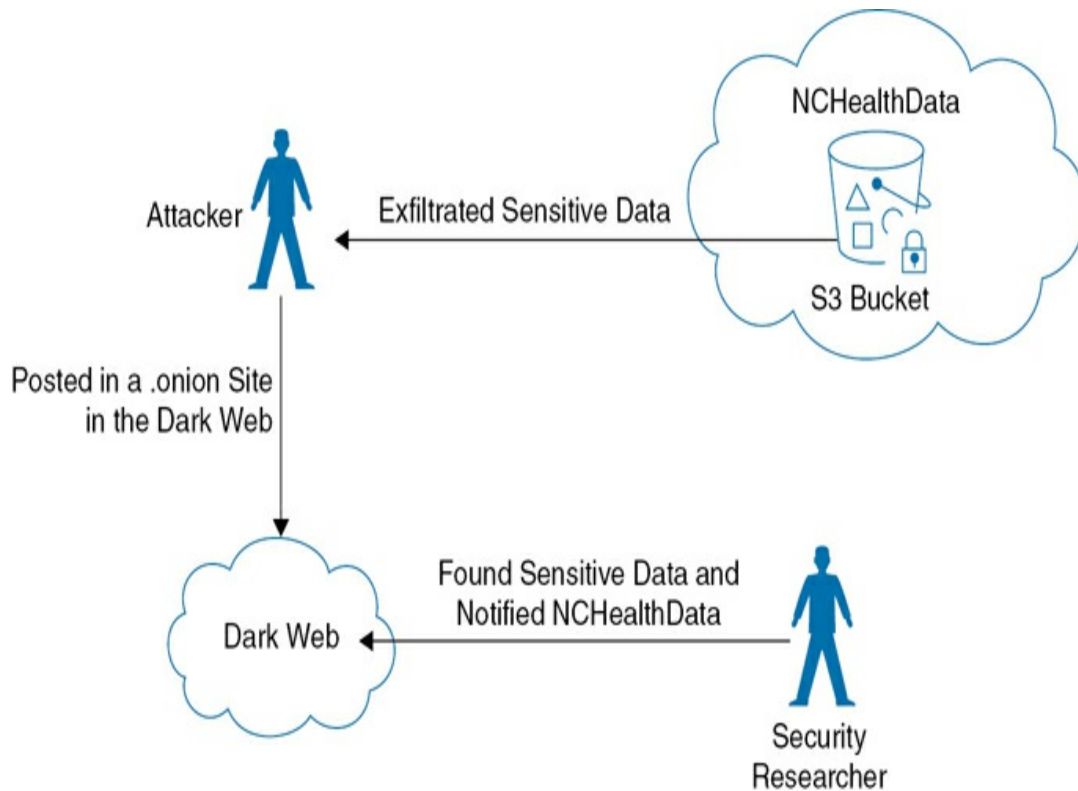


Figure 4-1 NCHealthData Attack

The initial investigation revealed that the data exposure resulted from improper access control, lack of encryption, and insufficient auditing and monitoring. The company was using AWS and an S3 bucket (cloud storage bucket). The cloud storage bucket was set to "public" instead of "private," making all stored data accessible to anyone with the URL. Data stored in the bucket was not encrypted at rest, deepening the severity of the exposure.

The company didn't have adequate mechanisms to monitor the security settings of the cloud storage or alert the IT team of unauthorized access. The IT staff responsible for configuring the cloud storage had inadequate training regarding cloud security best practices. There was a lack of security review and validation procedures for configurations before and after deployment.

Consequences

NCHealthData Inc. faced investigations by healthcare regulators and potential HIPAA violation fines. The exposure severely

damaged the trust relationship with the company's patients and partners. NCHHealthData Inc. incurred significant costs related to legal fees, penalties, and remediation efforts to secure its data and prevent future incidents.

Remedial Actions Taken

The access settings on the cloud storage bucket were corrected, and all data was encrypted at rest. The company used the resources in the AmazonS3 User Guide example bucket policies:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies.html>. Comprehensive training on cloud security best practices was mandated for all IT staff. The company also deployed tools to monitor security settings like AppOmni and alert for misconfigurations. Additionally, NCHHealthData Inc. instituted regular security audits to ensure compliance with security policies and standards.

Lessons Learned

The following are lessons learned from this case study:

- Always verify security settings, especially when deploying new cloud services.
- Implement and maintain monitoring tools to detect and alert on security misconfigurations.
- Ensure that all personnel involved in the deployment and management of IT services are trained in security best practices.

Insider Threats

Insider threats come from people within the organization who have legitimate access to the systems but misuse their privileges, intentionally or unintentionally. There are two major categories of “insiders” contributing to insider threats:

- **Malicious Insiders:** Employees who intentionally steal or sabotage data

- **Accidental Misuse:** Employees who inadvertently mishandle data, often due to lack of proper training or awareness

Case Study: Insider Threat Incident at a SaaS Cloud Provider

GlobalTech SaaS Inc. (a fictitious company) is a leading provider of cloud-based business solutions, offering a wide range of services from data storage to application hosting in many countries around the world. The company faced a significant security breach because of an insider.

The insider threat incident at GlobalTech SaaS Inc. involved a senior software engineer named John Doe. John had been with the company for more than five years and was highly trusted with elevated access privileges due to his role in developing key components of the platform.

John was motivated by personal financial problems and dissatisfaction with the company. He decided to exploit his access to sensitive customer data. He began covertly extracting large datasets that included proprietary business data and personal information of government and financial services clients. His plan was to sell this information on the dark web.

Investigation and User and Entity Behavior Analytics by Splunk

This activity was initially missed by traditional security measures but was eventually detected by a newly implemented user and entity behavior analytics (UEBA) solution by Splunk. Splunk's UEBA flagged unusual data access patterns and large data exports performed during off-hours.

The security team immediately initiated a forensic investigation and found conclusive evidence of unauthorized data extraction linked directly to John's user credentials. His activities were traced back over several weeks, revealing the extent of data theft.

GlobalTech SaaS Inc. promptly moved to contain the breach by securing all potential data leak points and informing affected clients. The incident prompted a companywide overhaul of insider threat detection capabilities and a tightening of access controls.

Response and Outcome

Upon confirmation of the breach, John was suspended from his position, and law enforcement was notified. The company began an extensive review of all operations accessed by John to understand the full scope of the breach. John faced legal charges for violating data protection laws and the company's policies. The case was taken to court where he received a substantial penalty and prison sentence.

Lessons Learned and Preventative Measures

The following are lessons learned from this case study:

- Implement more sophisticated anomaly detection systems like Splunk's UEBA. You will learn more about Splunk in [Chapter 10, "Security: Cisco XDR, Splunk, and Cisco Vulnerability Management."](#)
- Increase monitoring of privileged accounts.
- Introduce more stringent controls and regular audits of accounts with elevated privileges.
- Implement a Zero Trust architecture ensuring that trust levels are continuously verified. You will learn about Zero Trust later in this chapter.
- Conduct regular security training focused on insider threats.
- Create an organizational culture that encourages ethical behavior and reporting of suspicious activities.
- Update and test incident response plans regularly.
- Ensure quick isolation of affected systems and fast legal and

regulatory communication.

Implementing comprehensive insider threat programs that include advanced detection tools, strict access controls, and ongoing employee training is a must for safeguarding sensitive data and maintaining trust in cloud-based services.

Advanced Persistent Threats (APTs) and Nation Sponsored Attackers

Advanced persistent threats are sophisticated, prolonged campaigns where attackers gain access to a network and remain undetected for a long period. They aim to steal data gradually or establish a foothold for future attacks.

APTs often target specific organizations for espionage or disruption. These threat actors use many different methods to breach security, including spearphishing, specialized malware, and exploitation of vulnerabilities.

The MITRE ATT&CK Framework and Examples of APTs

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework is a globally recognized knowledge base used for understanding the tactics, techniques, and procedures (TTPs) employed by threat actors, including APTs. You can access MITRE's ATT&CK at attack.mitre.org. MITRE is a not-for-profit organization that operates research and development centers sponsored by the United States federal government.

The ATT&CK framework catalogs common adversarial behaviors in a structured format, aiding cybersecurity professionals in identifying, understanding, and defending against attacks. The framework divides the adversarial behaviors into matrices organized by different operational environments, such as Enterprise, Mobile, and Cloud.

[Figure 4-2](#) shows a custom export of the MITRE ATT&CK Navigator covering the TTPs of attacks against SaaS implementations.

about

Omar's SaaS TTPs Example
MITRE ATT&CK TTPs for SaaS implementations

platforms

SaaS, Office 365, Google Workspace

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Impact
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation	Abuse Elevation Control Mechanism	Abuse Elevation Control Mechanism	Brute Force	Account Discovery	Internal Spearphishing	Automated Collection	Exfiltration Over Alternative Protocol	Account Access Removal
Phishing	Cloud API	Additional Cloud Credentials	Temporary Elevated Cloud Access	Temporary Elevated Cloud Access	Credential Stuffing	Cloud Account	Software Deployment Tools	Data from Cloud Storage	Exfiltration Over Web Service	Endpoint Denial of Service
Spearphishing Link	Serverless Execution	Additional Cloud Roles	Account Manipulation	Domain or Tenant Policy Modification	Password Cracking	Email Account	Taint Shared Content	Data from Information Repositories	Exfiltration Over Webhook	Application Exhaustion Flood
Spearphishing Voice	Software Deployment Tools	Additional Email Delegate Permissions	Additional Cloud Credentials	Trust Modification	Password Guessing	Cloud Service Dashboard	Use Alternate Authentication Material	Code Repositories	Transfer Data to Cloud Account	Application or System Exploitation
Trusted Relationship		Device Registration	Additional Cloud Roles	Exploitation for Defense Evasion	Password Spraying	Cloud Service Discovery	Application Access Token	Confluence		Service Exhaustion Flood
Valid Accounts		Create Account	Additional Email Delegate Permissions	Hide Artifacts	Forge Web Credentials	Permission Groups Discovery	Web Session Cookie	Sharepoint		Financial Theft
Cloud Accounts		Cloud Account	Device Registration	Email Hiding Rules	SAML Tokens	Cloud Groups		Email Collection		Network Denial of Service
Default Accounts		Event Triggered Execution	Domain or Tenant Policy Modification	Impair Defenses	Web Cookies			Email Forwarding Rule		Direct Network Flood
		Modify Authentication Process	Trust Modification	Disable or Modify Cloud Logs	Modify Authentication Process			Remote Email Collection		Reflection Amplification
		Conditional Access Policies	Event Triggered Execution	Impersonation	Conditional Access Policies					
		Hybrid Identity	Valid Accounts	Indicator Removal	Hybrid Identity					
		Multi-Factor Authentication	Cloud Accounts	Clear Mailbox Data	Multi-Factor Authentication					
		Office Application Startup	Default Accounts	Multi-Factor Authentication Process	Multi-Factor Authentication Request Generation					
		Add-ins		Conditional Access Policies	Steal Application Access Token					
		Office Template Macros		Hybrid Identity	Steal Web Session Cookie					
		Office Test		Multi-Factor Authentication	Unsecured Credentials					
		Outlook Forms		Use Alternate Authentication Material	Chat Messages					
		Outlook Home Page		Application Access Token						
		Outlook Rules		Web Session Cookie						
		Valid Accounts		Valid Accounts						
		Cloud Accounts		Cloud Accounts						
		Default Accounts		Default Accounts						

Figure 4-2 MITRE ATT&CK Covering TTPs Against Cloud Implementations

The data presented in [Figure 4-2](#) is available for further analysis

and use in different formats including JSON, SVG, and spreadsheet form. You can access and download this data from my GitHub repository: <https://github.com/The-Art-of-Hacking/h4cker/tree/master/cloud-resources>.

The MITRE ATT&CK matrix also provides detailed information on numerous APT groups, offering insights into their origins, typical targets, known exploits, and behavioral patterns. You can access the list of threat actor groups (including APTs) at <https://attack.mitre.org/groups>. This repository enables security teams not only in reactive incident response but also in proactive threat hunting and strategic security planning.

Ransomware

Ransomware attacks involve encrypting an organization's data and demanding payment for the decryption key. For SaaS providers, a ransomware attack can cripple service availability and compromise client data. Ransomware can be used for locking away access to data files, databases, or applications; consequently, attackers often threaten to post the captured data publicly unless a ransom is paid.

Case Study: MOVEit Ransomware Attack by the Cllop Group

Let's review a real-life case study. The MOVEit software, a popular SaaS platform used for secure file transfers, fell victim to a ransomware attack orchestrated by the Cllop ransomware group. This example highlights the vulnerability of even well-secured SaaS applications to sophisticated cyber threats. The attackers exploited a critical zero-day SQL injection vulnerability allowing remote code execution. This vulnerability enabled the attackers to perform unauthorized actions on the database, including data exfiltration.

The Cllop group executed commands that compromised the MOVEit Transfer's database, enabling the mass downloading of sensitive data. The breach led to the exfiltration of personal

information from almost 6,800 individuals, including employees and their families. The attack not only resulted in the theft of sensitive data but also exposed MOVEit to potential reputational damage. More than 100 organizations worldwide were affected, including high-profile entities like the U.S. Department of Energy, British Airways, and the BBC. You can see the cascading consequences of security vulnerabilities in widely used SaaS platforms.

The underlying cause of the vulnerability was attributed to configuration drift over time, which often went unnoticed until it was too late. The incident led to a reassessment of security protocols, emphasizing the need for continuous configuration monitoring and regular security audits to identify and rectify potential vulnerabilities promptly.

For more detailed insights and recommendations on SaaS security following the MOVEit incident, you can explore further resources from the Cloud Security Alliance and other cybersecurity research at <https://cloudsecurityalliance.org/blog/2023/11/08/moveit-exploit-ransomware-attack-why-saas-security-is-critical-during-a-cyberattack>.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks

Denial-of-service and distributed denial-of-service attacks aim to disrupt service availability by overwhelming the SaaS platform with a flood of unnecessary traffic. They can result in preventing users from accessing the service.

Yes, it's a common belief that cloud services, such as SaaS platforms, offer improved resilience against DDoS attacks due to their distributed nature and the scalability provided by cloud infrastructure. However, cloud services are still susceptible to these attacks for several reasons. While cloud environments can scale resources, they often operate on a shared infrastructure model. When one tenant of a cloud service is targeted by a DDoS attack, that attack can also impact the performance for other tenants who share the same underlying infrastructure.

Even though cloud providers typically have significant bandwidth, DDoS attacks can still overwhelm the network capacity. If an attack is large enough, it can saturate the network, preventing legitimate user traffic from accessing the SaaS platform. DDoS attacks can also be directed at the application layer, not just the network layer. These attacks are often more sophisticated and target specific application features or functions. Even if the network infrastructure can handle the traffic, the application itself may be unable to process all the requests, leading to service degradation or denial.

There is also the concept of economic denial of sustainability (EDoS). In cloud environments, resources are often paid for based on usage. An attacker can exploit this usage by intentionally increasing the demand on the cloud service, thereby inflating the costs for the service provider or the victim.

Regulatory Compliance and Certifications

Cloud providers (including those that provide SaaS solutions) must adhere to a range of regulatory compliance standards and certifications to ensure they manage and protect their customers' data responsibly. Let's review a few key examples.

General Data Protection Regulation (GDPR)

The General Data Protection Regulation profoundly impacts cloud providers, including SaaS providers, operating within or catering to customers in the European Union. GDPR mandates stringent data protection and privacy measures to safeguard personal data.

For cloud providers, this protection means guaranteeing that all personal data handled meets GDPR's privacy standards, which include consent of data subjects for data processing, anonymization of collected data to protect privacy, and the secure transmission and storage of personal data. Cloud providers must enable data subjects to exercise their rights under the GDPR, such as the right to access, correct, delete, or transfer their personal data. Noncompliance can result in severe penalties, which can go up to 4 percent of the company's annual global turnover or €20 million, whichever is higher, thereby raising the eyebrows of executives on the importance of GDPR

compliance in avoiding financial and reputational damage. You can obtain information about GDPR at <https://gdpr-info.eu>.

Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Data Security Standard applies to cloud providers that handle, process, or store credit card information or any other financial data from their customers. This standard requires providers to maintain a secure environment for cardholder data to prevent credit card fraud and breaches. Compliance involves adhering to a set of controls around data security, which includes encryption, access control, vulnerability management, and regular monitoring and testing of the network.

For SaaS providers, achieving PCI DSS compliance not only boosts consumer trust but also protects against data breaches and the heavy fines associated with noncompliance. Because customer data is often processed and stored on cloud infrastructures, it's critical for these providers to rigorously apply PCI DSS standards to all systems involved in card processing operations, including cloud-based services. You can obtain information about PCI DSS at <https://www.pcisecuritystandards.org>.

International Organization for Standardization (ISO) Standards and Certification

The International Organization for Standardization has developed several standards that specifically address many aspects of cloud security. These standards are part of a broader framework aimed at guaranteeing the secure and effective management of information technology environments. The following are some of the key ISO standards related to cloud security:

- **ISO/IEC 27001:** This standard is the cornerstone of cloud security and information security management systems (ISMS). It provides a systematic approach to managing sensitive company information so that it remains secure. It includes people, processes, and IT systems by applying a risk management process. For cloud and SaaS providers, ISO

27001 certification helps in demonstrating a clear commitment to information security at all levels of the organization. Compliance with this standard involves a systematic examination of the organization's information security risks, including threats, vulnerabilities, and impacts. The standard helps providers develop coherent and comprehensive suites of information security controls and other forms of risk management to address risks deemed unacceptable.

- **ISO/IEC 27017:** This standard is built on the foundation of ISO/IEC 27001 and provides additional security controls for cloud services. It offers implementation guidance for both cloud service providers and users of cloud services.
- **ISO/IEC 27018:** This standard is a code of practice for protecting personal data in the cloud. It addresses public cloud computing and acts as a privacy safeguard, focusing specifically on personal data protection in cloud environments.
- **ISO/IEC 27036:** This standard deals with information security for supplier relationships, including the provisions of cloud services. It provides guidance for managing the relationship between a customer and a provider.
- **ISO/IEC 17788:** This standard provides an overview of cloud computing along with a set of terms and definitions for cloud computing services.

These ISO standards provide a framework to ensure a consistent and secure cloud service environment.

Europe's Cyber Resilience Act (CRA)

Europe's Cyber Resilience Act is designed to bolster the overall security and resilience of digital products, including software and services provided in the cloud. You can obtain the latest information about the CRA at <https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act>. For cloud and SaaS providers, the CRA mandates stricter security requirements throughout the lifecycle of their services, from development to decommissioning.

Cloud providers (including SaaS) must ensure that their services are resilient

against disruptions, including cyber attacks, and must be able to restore operations swiftly in case of an incident. Compliance involves regular vulnerability assessments, the application of security patches, and timely disclosure of any cybersecurity incidents. The CRA aims to ensure that providers not only secure their infrastructure but also remain aware and responsive to emerging threats and vulnerabilities.

Federal Risk and Authorization Management Program (FedRAMP)

The Federal Risk and Authorization Management Program is a U.S. governmentwide program that standardizes security assessment, authorization, and continuous monitoring for cloud products and services used by federal agencies. Its goal is to ensure that cloud services meet a consistent, rigorous baseline of security requirements, reducing duplication of effort across agencies and accelerating the adoption of secure cloud technologies.

For cloud providers (including SaaS providers), FedRAMP compliance is mandatory to serve federal agencies. The process involves a comprehensive review of a provider's security controls and operational processes against the FedRAMP baseline, followed by continuous monitoring to ensure those controls remain effective over time. Achieving FedRAMP authorization not only opens the door to federal customers but also demonstrates a strong commitment to industry-leading security practices, which can be a market differentiator even for nongovernment clients.

A critical step in the FedRAMP process is securing a federal agency sponsor. This agency acts as the advocate for the cloud service provider (CSP) and partners with the provider throughout the authorization process. The sponsor is involved in

- **Initiation and Requirements Definition:** Confirming that the CSP's solution meets the agency's mission needs
- **Participation in the Authorization Process:** Coordinating with the CSP, the Third-Party Assessment Organization (3PAO), and the FedRAMP Program Management Office (PMO) throughout the security

assessment

- **Providing Feedback:** Helping prioritize risk mitigation efforts and approving remediation plans for any findings

Without an agency sponsor, CSPs typically cannot proceed through the full FedRAMP Joint Authorization Board (JAB) process unless they pursue the JAB Provisional ATO (P-ATO) route, which is more selective and reserved for services with high demand across multiple agencies.

At the conclusion of the assessment, the sponsoring agency (or JAB, if pursuing a P-ATO) issues an authority to operate (ATO). The ATO is a formal declaration that the risk posture of the cloud service is acceptable for use by the agency. It is one of the most important milestones in FedRAMP compliance because it signals that the CSP has successfully implemented the required security controls and risk mitigations.

An ATO is not permanent. It is valid as long as the CSP maintains compliance through continuous monitoring and reporting. If security posture drifts outside of acceptable limits, the ATO can be revoked or suspended until issues are resolved.

FedRAMP places strong emphasis on continuous monitoring (ConMon) to ensure that security remains robust after authorization. CSPs must submit monthly and annual security deliverables, including vulnerability scans, Plan of Action & Milestones (POA&M) updates, and incident reports.

When the CSP makes any material changes to the system—such as new features, major architectural shifts, or changes to boundary definitions—it must submit a significant change request (SCR). SCRs are reviewed by the sponsoring agency or JAB to determine whether the changes require a partial or full reassessment before they can be deployed into production. This approach ensures that the system remains within the authorized security baseline even as it evolves.

[Figure 4-3](#) outlines the steps for achieving FedRAMP readiness. The process is divided into five main steps, each detailing the specific actions required by different parties (CSP, 3PAO, and FedRAMP PMO).

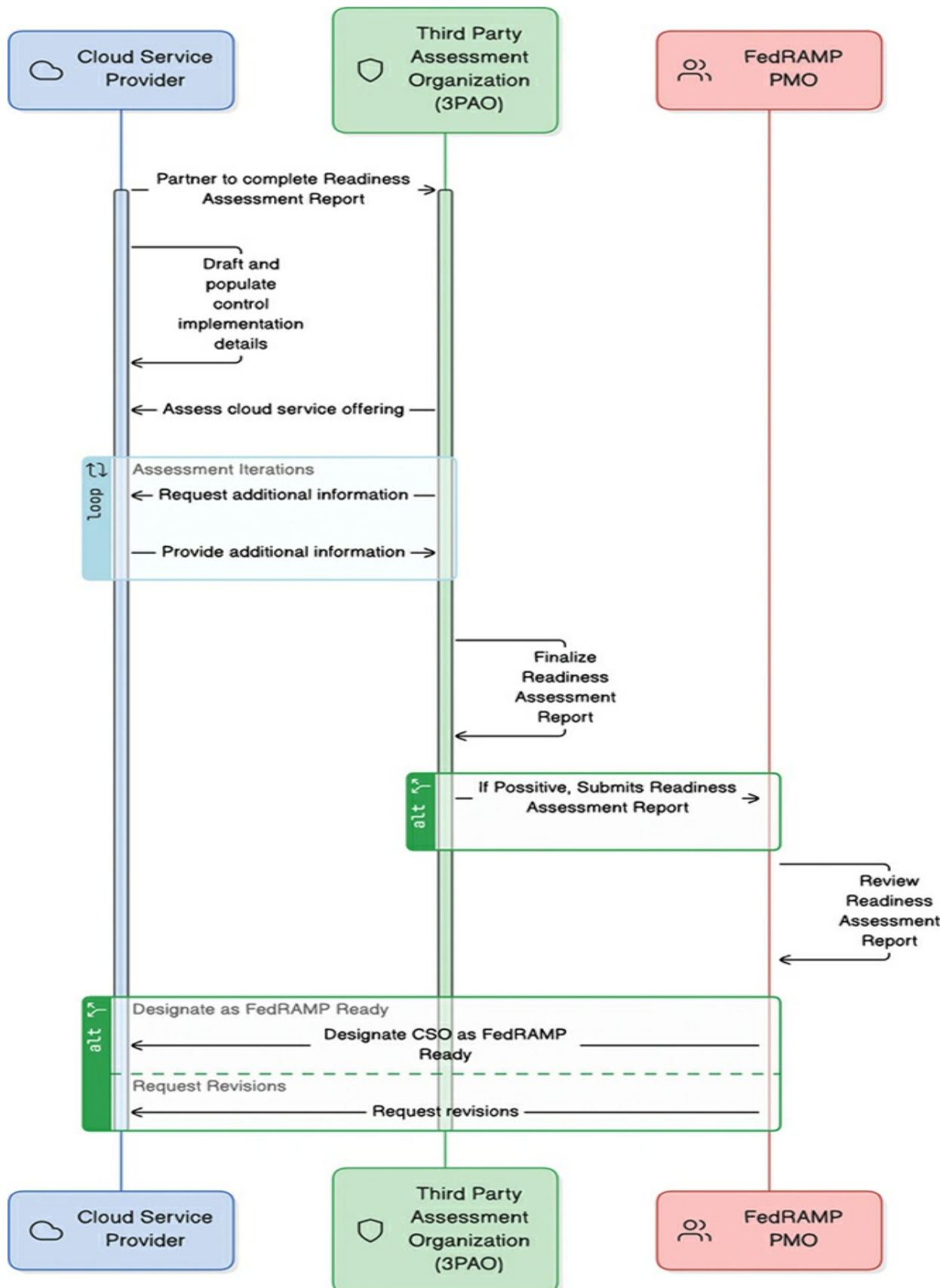


Figure 4-3 Achieving FedRAMP Readiness

The following are the steps to achieving FedRAMP readiness:

- Step 1.** The CSP partners with a 3PAO to complete a readiness assessment report at the Moderate or High impact level.
- Step 2.** The CSP drafts and populates control implementation details in the readiness assessment report.
- Step 3.** The 3PAO assesses the CSP's cloud service offering (CSO) according to the readiness assessment report.
- Step 4.** The 3PAO finalizes the readiness assessment report based on assessment results. If the 3PAO has a positive risk recommendation, the readiness assessment report may be submitted to the FedRAMP Program Management Office for review and acceptance.
- Step 5.** The FedRAMP PMO reviews the readiness assessment report and, if appropriate, designates the CSO as FedRAMP Ready.

Each step involves specific actions by the CSP, 3PAO, and FedRAMP PMO to ensure that the cloud service meets the necessary security requirements and can be designated as FedRAMP Ready.

Tip

The FedRAMP Marketplace offers a searchable and sortable database of CSOs that have received FedRAMP designation, a list of federal agencies utilizing FedRAMP-authorized CSOs, and FedRAMP-recognized 3PAOs qualified to conduct FedRAMP assessments. The FedRAMP PMO maintains the FedRAMP Marketplace. You can find additional information about the FedRAMP Marketplace at <https://www.fedramp.gov/about-marketplace>.

System and Organization Controls (SOC 1, 2, and 3)

System and Organization Controls reports are developed by the American Institute of Certified Public Accountants (AICPA). SOC reports are designed to help service organizations build trust and confidence with their clients by providing transparency and assurance about their internal controls.

SOC 1 reports are focused on the controls at a service organization that are relevant to financial reporting. They are designed to provide assurance to the organization's clients and auditors that the service provider's internal controls over financial reporting (ICFR) are effective.

The following are the types of SOC 1 reports:

- **Type I:** This report describes the service organization's system and evaluates the design of controls at a specific point in time.
- **Type II:** This report goes further by evaluating both the design and operating effectiveness of controls over a specified period, usually six months.

SOC 1 reports are particularly relevant for organizations that provide services impacting their clients' financial statements, such as payroll processors, data centers, and SaaS providers dealing with financial transactions.

SOC 2 reports address a broader range of controls than SOC 1, focusing on the systems and processes that relate to security, availability, processing integrity, confidentiality, and privacy. These reports provide insights into the effectiveness of a service organization's controls in these areas.

The following are the types of SOC 2 reports:

- **Type I:** This report evaluates the design of controls related to the trust service criteria at a specific point in time.
- **Type II:** This report assesses the operational effectiveness of these controls over a specified time period.

SOC 2 reports are essential for technology and cloud computing companies, data centers, and other service providers that manage customer data, ensuring that their clients can trust their systems to protect data and maintain privacy.

SOC 3 reports are like SOC 2 but are designed for a general audience. They provide a high-level overview of the service organization's controls related to

security, availability, processing integrity, confidentiality, and privacy without going into the detailed information found in SOC 2 reports.

SOC 3 reports are suitable for organizations that want to publicly demonstrate their commitment to trust and transparency without disclosing sensitive information. They are often used for marketing and public relations purposes, providing reassurance to potential customers and stakeholders.

Let's go over some of the differences between SOC 1, 2, and 3. The following are the focus area of SOC 1, 2, and 3:

- **SOC 1:** Internal controls over financial reporting
- **SOC 2:** Controls related to security, availability, processing integrity, confidentiality, and privacy
- **SOC 3:** High-level summary of SOC 2 controls for a general audience

Additionally, SOC 1 and SOC 2 require detailed reports intended for specific stakeholders, such as clients and their auditors. SOC 3 requires a general summary intended for broad distribution without disclosing detailed control descriptions or results.

SOC reports are designed to help service organizations, including cloud and SaaS providers, build trust and confidence in their service delivery processes and controls through a report by an independent certified public accountant. SOC 2 has become one of the most recognized trust signals for SaaS providers. Customers often fixate on SOC 2 when evaluating cloud vendors because it provides independent, third-party assurance that the provider has implemented rigorous controls around security, availability, processing integrity, confidentiality, and privacy. While SOC 2 compliance is not a legal requirement like FedRAMP or GDPR, achieving SOC 2 Type II certification significantly increases customer confidence and can be a deciding factor in vendor selection. For SaaS companies, an SOC 2 report is not just a compliance checkbox; it is a powerful market differentiator that demonstrates a mature security posture and commitment to protecting customer data.

The Cloud Security Alliance (CSA) Security, Trust, and Assurance Registry (STAR)

The Cloud Security Alliance developed the Security, Trust, and Assurance Registry. STAR is a comprehensive and publicly accessible registry designed to provide transparency and assurance regarding the security practices of CSPs. You can access STAR at <https://cloudsecurityalliance.org/star/registry>. The CSA STAR program is a security assurance program for cloud services, which includes key principles of transparency, rigorous auditing, and continuous improvement. It offers multiple levels of assurance based on the provider's requirements for security, compliance, and transparency.

The following are details about each of the levels of the CSA STAR program:

1. **Self-Assessment:** The initial level of STAR involves self-assessment, where CSPs document their compliance with CSA's Cloud Controls Matrix (CCM) and Consensus Assessments Initiative Questionnaire (CAIQ). This level allows CSPs to showcase their security measures and controls voluntarily. It helps them demonstrate their commitment to best practices and provides a starting point for clients to evaluate each provider's security posture. Self-assessments are publicly accessible, promoting transparency and trust among clients and stakeholders.
2. **Third-Party Audit:** The second level of STAR involves an independent third-party audit based on the CCM. This audit can be conducted in alignment with various standards such as ISO/IEC 27001. This level provides a higher degree of assurance by having an external party validate the CSP's security controls. It enhances credibility and demonstrates a serious commitment to security. Upon successful completion, CSPs receive a STAR Certification, indicating their adherence to stringent security standards.
3. **Continuous Monitoring:** The highest level of STAR focuses on continuous monitoring, where CSPs provide real-time updates on their security posture and control effectiveness. Continuous monitoring ensures that security measures are maintained and improved over time, providing ongoing assurance to customers. This level emphasizes dynamic and proactive security management, reflecting the ever-changing threat landscape.

By participating in the STAR program, CSPs can provide clear and

accessible information about their security practices, helping clients make informed decisions. The STAR certification signifies that a CSP adheres to industry best practices and standards, fostering trust among clients, partners, and stakeholders. CSPs with STAR certification can differentiate themselves in the marketplace by demonstrating their commitment to security and compliance. The CSA STAR program plays a crucial role in advancing cloud security by providing a structured framework for CSPs to evaluate and improve their security practices. It aligns with global standards and best practices, offering a universally recognized mark of security assurance.

Data Sovereignty

Data sovereignty is a critical concept in today's global digital landscape, where data is subject to the laws and governance structures of the country in which it is collected, stored, or processed. This principle has important implications for cloud services, particularly SaaS, because it affects how data is managed, secured, and accessed across different jurisdictions.

Note

Data residency and legal jurisdiction can have far-reaching implications. Some countries have mutual legal assistance treaties (MLATs) or similar agreements that allow data stored within their borders to be accessed and used in criminal investigations, even if the data belongs to foreign entities. This means that data hosted in a particular country could potentially be subject to subpoenas, search warrants, or other legal processes from that country's government or its treaty partners.

SaaS providers and customers must evaluate where data is physically stored, the legal frameworks governing that jurisdiction, and the potential for cross-border data access when choosing hosting regions. Understanding these extradition and compliance obligations is essential for managing risk, particularly in regulated industries or where sensitive data is involved.

Many countries have enacted laws that require certain types of data, such as personal, financial, or health information, to be stored within their borders. SaaS providers must ensure that their data centers comply with these local

data residency requirements, which may involve setting up multiple data centers in different regions.

SaaS providers must navigate a complex landscape of local regulations that dictate how data can be collected, stored, and processed. Noncompliance can result in severe penalties, legal actions, and damage to the provider's reputation. As a result, providers require a robust compliance framework and constant monitoring of regulatory changes.

Figure 4-4 illustrates how a SaaS application handles data in compliance with data sovereignty requirements by segregating data processing and storage based on geographic regions.

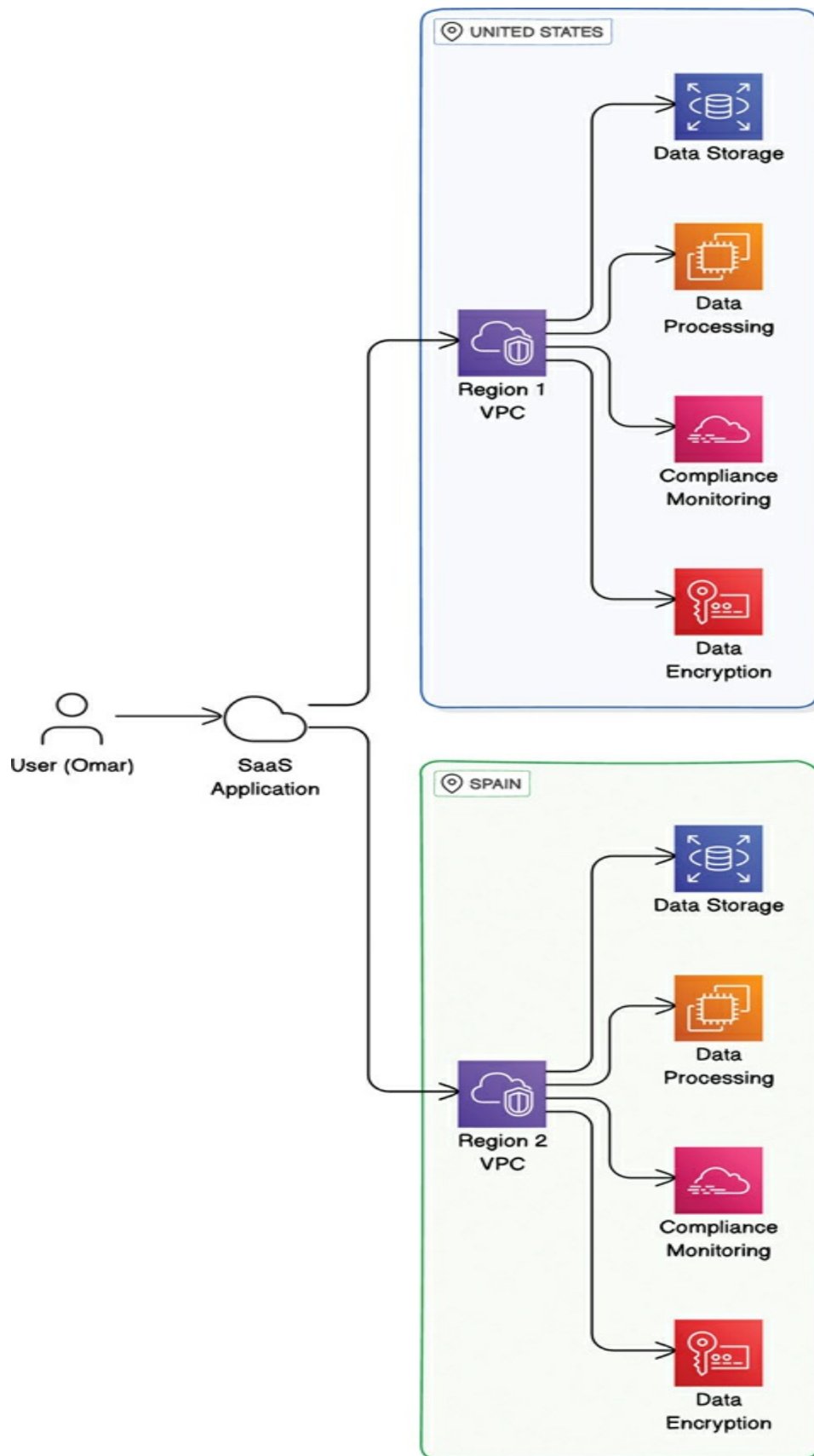


Figure 4-4 Examining Data Sovereignty

In [Figure 4-4](#), the user (Omar) interacts with the SaaS application, which provides services requiring data storage, processing, compliance monitoring, and encryption. The application is responsible for managing Omar's data according to the data sovereignty laws applicable to his location—the United States (Region 1). Omar's data is stored in a data center within the United States. Data processing activities are conducted within the U.S.-based data center. The system ensures that data processing and storage comply with U.S. regulations. Data is encrypted to ensure security and compliance with legal requirements.

Similar to Region 1, data for users in Spain is stored within Spain to comply with local data residency requirements. All data processing activities are performed within the Spanish data center. The system ensures compliance with Spain's data protection laws and regulations.

In this example, the SaaS provider maintains separate virtual private clouds (VPCs) in different geographic regions (the United States and Spain) to comply with local data sovereignty laws. This approach ensures that data remains within the legal jurisdiction where it was collected, addressing regulations such as the GDPR in Europe or CCPA in the United States.

By storing and processing data locally, the SaaS provider ensures compliance with data residency requirements, which mandate that certain types of data must not leave the country of origin. This mandate helps avoid legal issues and potential penalties for noncompliance. [Figure 4-4](#) highlights the importance of data encryption and compliance monitoring. Users like Omar can trust that their data is handled in accordance with local laws, enhancing their confidence in the SaaS provider. This trust is crucial for SaaS providers to build and maintain strong relationships with their clients.

Note

Different countries have different laws regarding government access to data. For example, some countries may require SaaS providers to grant access to data for national security or law enforcement purposes. SaaS providers must be transparent about such requirements and ensure that clients are aware of how their

data might be accessed. To comply with data sovereignty requirements, SaaS providers may need to establish data centers in multiple countries. This geographical expansion involves significant infrastructure investment and increased operational costs. Providers must weigh these costs against the benefits of accessing new markets and complying with local laws.

Providing localized services tailored to the regulatory environment of each country can increase operational complexity. SaaS providers need to adapt their services to meet specific legal and regulatory requirements, which can involve customizing features, language support, and data handling procedures. SaaS providers that effectively manage data sovereignty issues can gain a competitive edge. Clients are more likely to choose providers that offer robust compliance and data protection solutions, which can differentiate them in a crowded market.

Architectural Considerations for Data Partitioning and Tenant Isolation

Data partitioning and tenant isolation are critical for ensuring that each customer's data (referred to as a tenant in multitenant architectures) is securely isolated from other tenants. These concepts help in achieving better security, performance, and compliance.

For example, data partitioning could involve dividing a data service into distinct, independent parts or databases, which helps in managing and accessing data efficiently. This treatment is crucial for multitenant SaaS applications, where each tenant's data needs to be kept separate and secure.

[Figure 4-5](#) provides an example of data partitioning and tenant isolation.

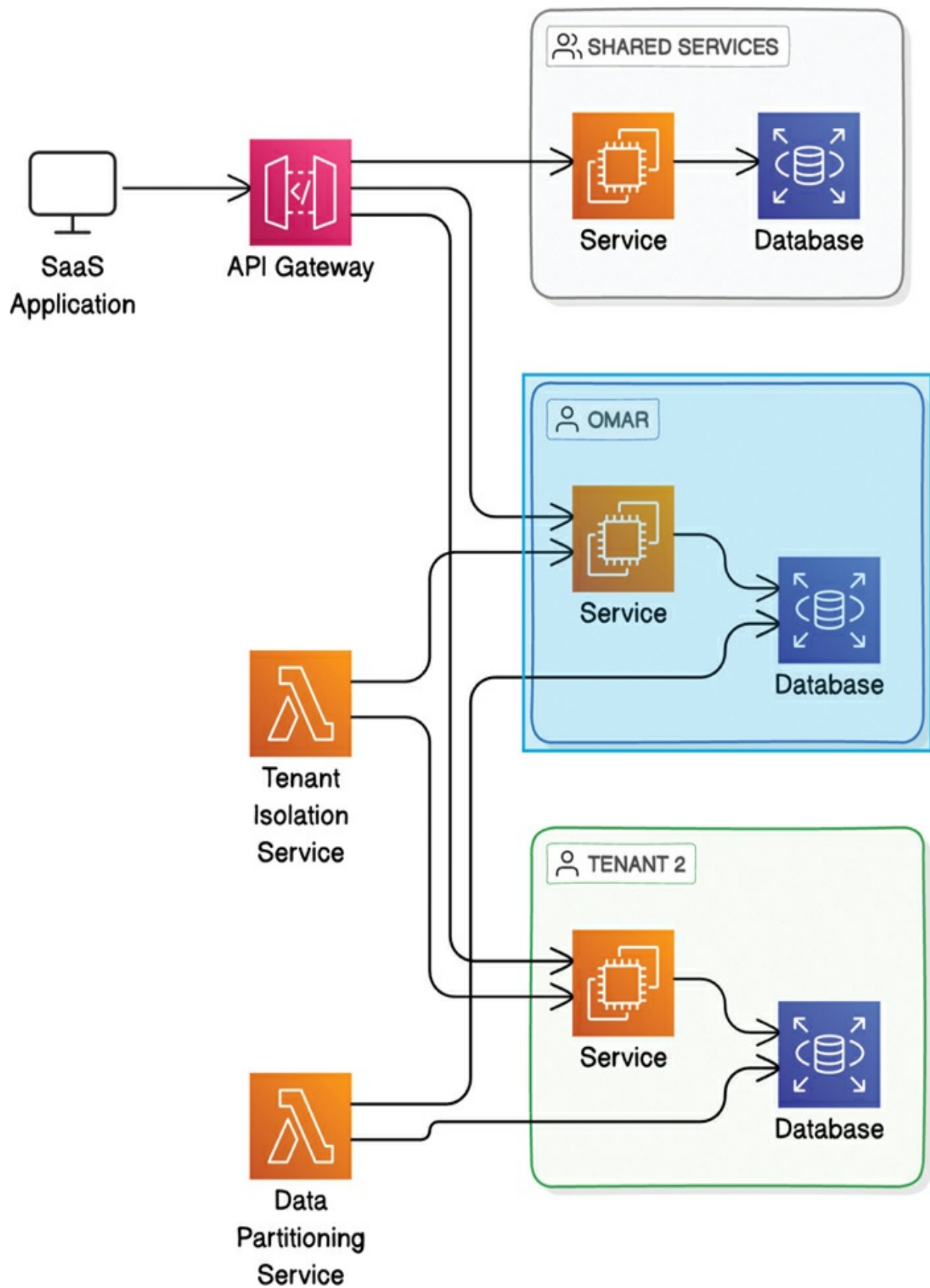


Figure 4-5 Examining Data Partitioning and Tenant Isolation

Figure 4-5 illustrates how a SaaS application handles data partitioning and tenant isolation, specifically showing how different tenants' data and services are managed. In this context, Omar represents Tenant 1.

The “SaaS Application” is the main application that users interact with. It routes requests through an “API Gateway”. The API Gateway acts as an entry point for all incoming requests to the SaaS application. It directs the requests to the appropriate services and handles API call management.

The “Shared Services” section contains common services and databases used by all tenants for shared functionality. This setup helps optimize resource usage and simplifies maintenance for features used across all tenants. The “Tenant Isolation Service” ensures that each tenant's data and services are isolated from one another. It routes requests to the correct tenant-specific services and databases, maintaining strict separation to prevent data leakage and unauthorized access.

The “Data Partitioning Service” handles the logical or physical partitioning of data. It ensures that data for each tenant is correctly segmented, whether through separate databases or distinct sections within a shared database. This implementation handles the business logic and processes specific to Omar (Tenant 1). It stores data exclusively for Omar, ensuring data isolation and security.

Note

When a user (Omar or another tenant) interacts with the SaaS application, that tenant's requests are sent to the API Gateway. The API Gateway routes the requests to the appropriate shared or tenant-specific services based on the request context (e.g., which tenant the user belongs to). For services common to all tenants, the API Gateway routes requests to shared services, which interact with the shared database.

In logical partitioning, data from multiple tenants is stored in the same database, but each tenant's data is logically separated using unique identifiers (e.g., tenant ID). One of the advantages is to reduce cost. Logical partitioning

requires fewer resources because a single database is shared among multiple tenants. However, it requires stringent access controls to ensure that tenants can access only their data. A disadvantage could be potential performance issues if the database is not properly optimized.

In physical partitioning, each tenant's data is stored in a separate database or separate tables within a shared database. The main advantage is strong isolation because data is physically separated. Additionally, this approach provides reduced contention for resources, leading to better performance. However, it requires more resources and can be more expensive to manage. Managing multiple databases or tables could also be more complex.

Tools and Techniques for Preventing Data Loss in a SaaS Setup

Data loss prevention (DLP) is a critical strategy for safeguarding sensitive information and preventing data breaches. Let's explore some of the types of tools and techniques for preventing data loss in a SaaS setup and the integration of DLP solutions with existing systems.

Encryption

Of course, data encryption is one of the best ways to protect data. At-rest encryption ensures that data stored within the SaaS application is encrypted using strong encryption algorithms, protecting it from unauthorized access.

In-transit encryption encrypts data as it moves between the SaaS application and the user's device, typically using protocols like Transport Layer Security (TLS). [Figure 4-6](#) illustrates encryption at rest and in transit.

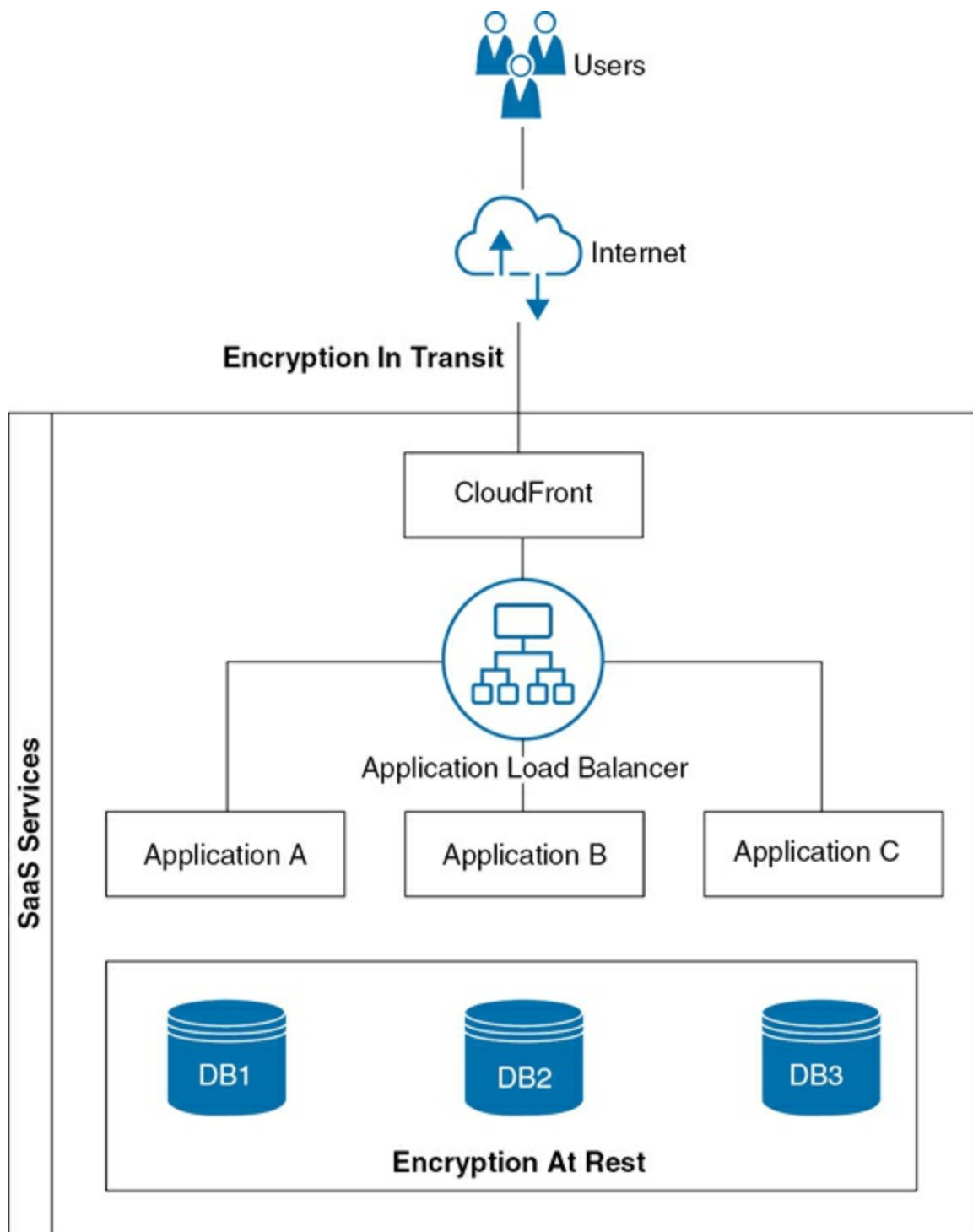


Figure 4-6 Encrypting Data at Rest and in Transit

Encryption in process, also known as encryption of data in use, ensures that data remains encrypted while being processed by applications or services. Traditional encryption schemes typically focus on data at rest (stored data) or

data in transit (data being transmitted). There are a few techniques used for encryption in process:

- **Secure Computation over Encrypted Data (SCED):** This technique allows specific computations to be performed directly on encrypted data using specialized algorithms. It often leverages homomorphic encryption and other advanced cryptographic techniques. The Paillier cryptosystem is a partially homomorphic encryption scheme that supports addition operations on encrypted data. The paper titled “Secure Computation over Encrypted Databases” (available at <https://arxiv.org/pdf/2308.02878>) provides a detailed overview of SCED.
- **Homomorphic Encryption:** This technique allows computations to be performed on encrypted data without decrypting it first. The results of the computation are also in encrypted form and can be decrypted only by the owner of the decryption key. Use cases of homomorphic encryption include secure data analytics and privacy-preserving computations in cloud environments and AI workloads. Microsoft Simple Encrypted Arithmetic Library (SEAL) is an open-source homomorphic encryption library.
- **Trusted Execution Environments (TEEs):** These secure areas within a processor provide an isolated environment for sensitive computations. Data and code within a TEE are protected from outside access and tampering. An example of TTE is the Intel Software Guard Extensions (SGX), which provides a TEE for Intel processors, allowing applications to execute code and process data in a secure enclave. An open-source implementation example is Project Oak at <https://github.com/project-oak/oak>.
- **Secure Multi-Party Computation (SMPC):** This technique enables multiple parties to jointly compute a function over their inputs while keeping those inputs private. Each party’s input is secret-shared, and computations are performed on these shares. The Fairplay system and MP-SPDZ framework provide tools for implementing SMPC protocols.
- **Functional Encryption:** This technique allows users to compute specific functions on encrypted data without revealing the data itself. Only specific functions’ results can be decrypted, depending on the

decryption key's capabilities. CryptDB is a system that uses functional encryption to enable SQL queries on encrypted databases.

- **Confidential Computing:** This technique involves the use of hardware-based TEEs to process sensitive data while keeping it protected from unauthorized access. This approach often combines TEEs with encryption techniques to secure data in use. Google Cloud Confidential Computing provides VMs and services that use AMD Secure Encrypted Virtualization (SEV) technology to encrypt data in use.

Note

The article titled “An Overview of Searchable Encryption, Homomorphic Encryption, and Multiparty Computation in AI Implementations” (available at <https://becomingahacker.org/8cb593e4a441>) provides an overview of some of the techniques listed above.

Post-quantum cryptographic algorithms are becoming increasingly important for SaaS due to several key factors. Quantum computers have the potential to break widely used cryptographic algorithms, such as RSA and ECC, by leveraging quantum algorithms like Shor's algorithm. This capability introduces a significant threat to the security of data protected by these traditional algorithms.

SaaS providers often handle sensitive data that needs to be secured over long periods. As quantum computers become more powerful, data encrypted with current algorithms could be at risk in the future. Post-quantum algorithms aim to provide security assurances even in the era of quantum computing.

Regulatory bodies and industry standards are beginning to recognize the potential threat of quantum computing. Adopting post-quantum cryptographic algorithms can help SaaS providers stay ahead of compliance requirements and avoid potential future penalties.

Note

Implementing post-quantum cryptography can enhance customer trust by demonstrating a proactive approach to securing their data against future threats. This effort is especially important for SaaS

providers dealing with highly sensitive information.

Transitioning to post-quantum cryptographic algorithms helps future-proof SaaS services. By adopting these algorithms early, SaaS providers can ensure continuous protection and reduce the need for urgent, large-scale cryptographic transitions when quantum computers become more prevalent.

Examples of post-quantum algorithms include

- **Lattice-Based Cryptography:** Algorithms like NTRU and NewHope are considered strong candidates for post-quantum security.
- **Code-Based Cryptography:** McEliece cryptosystem is another candidate, leveraging error-correcting codes.
- **Hash-Based Cryptography:** Algorithms such as Merkle tree signatures (e.g., XMSS) provide quantum-resistant digital signatures.
- **Multivariate Quadratic Equations:** Algorithms like Rainbow and HFEv- are based on solving systems of multivariate quadratic equations.

The NIST Post-Quantum Cryptography Project

The National Institute of Standards and Technology (NIST) launched the Post-Quantum Cryptography Project to evaluate, standardize, and promote cryptographic solutions capable of withstanding quantum attacks. This project involves a rigorous, multiyear process of soliciting, analyzing, and vetting candidate algorithms from the global cryptographic community. The key objectives of the NIST Post-Quantum Cryptography Project include algorithm selection, collaborating with international researchers, and developing new standards. You can access the NIST Post-Quantum Cryptography Project details at

<https://csrc.nist.gov/projects/post-quantum-cryptography>.

Data Masking and Tokenization

Data masking is the process of hiding sensitive data by replacing it with fictitious data that retains the same structure, ensuring that unauthorized users

cannot access the actual data.

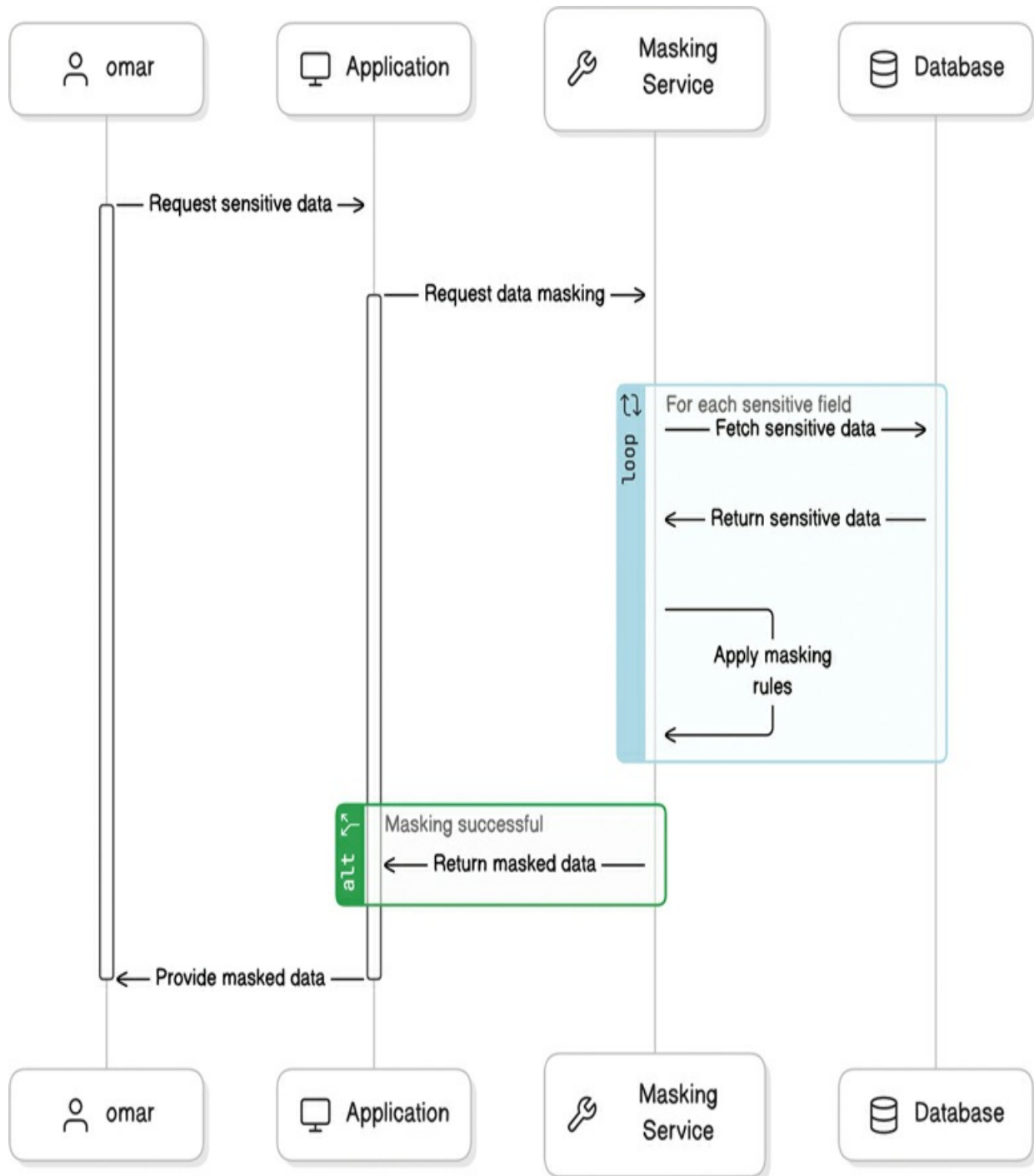


Figure 4-7 Data Masking Process

Figure 4-7 illustrates the process of data masking. The entities involved in this process are the user (Omar), the application, the masking service, and the

database.

1. Omar requests sensitive data through the Application.
2. The Application forwards a request to the Masking Service to mask the sensitive data before providing it to Omar.
3. The Masking Service processes the request in a loop for each sensitive field.
4. The Masking Service retrieves the sensitive data from the Database.
5. The sensitive data is sent back to the Masking Service.
6. The Masking Service applies specific masking rules to the sensitive data to create a masked version of the data.
7. After all sensitive fields have been masked successfully, the Masking Service returns the masked data to the Application.
8. The Application then provides the masked data to Omar.

Tokenization replaces sensitive data elements with nonsensitive equivalents (tokens) that can be mapped back to the original data, reducing the risk of exposure. [Figure 4-8](#) shows an example of the data tokenization process.

Tokenization Process

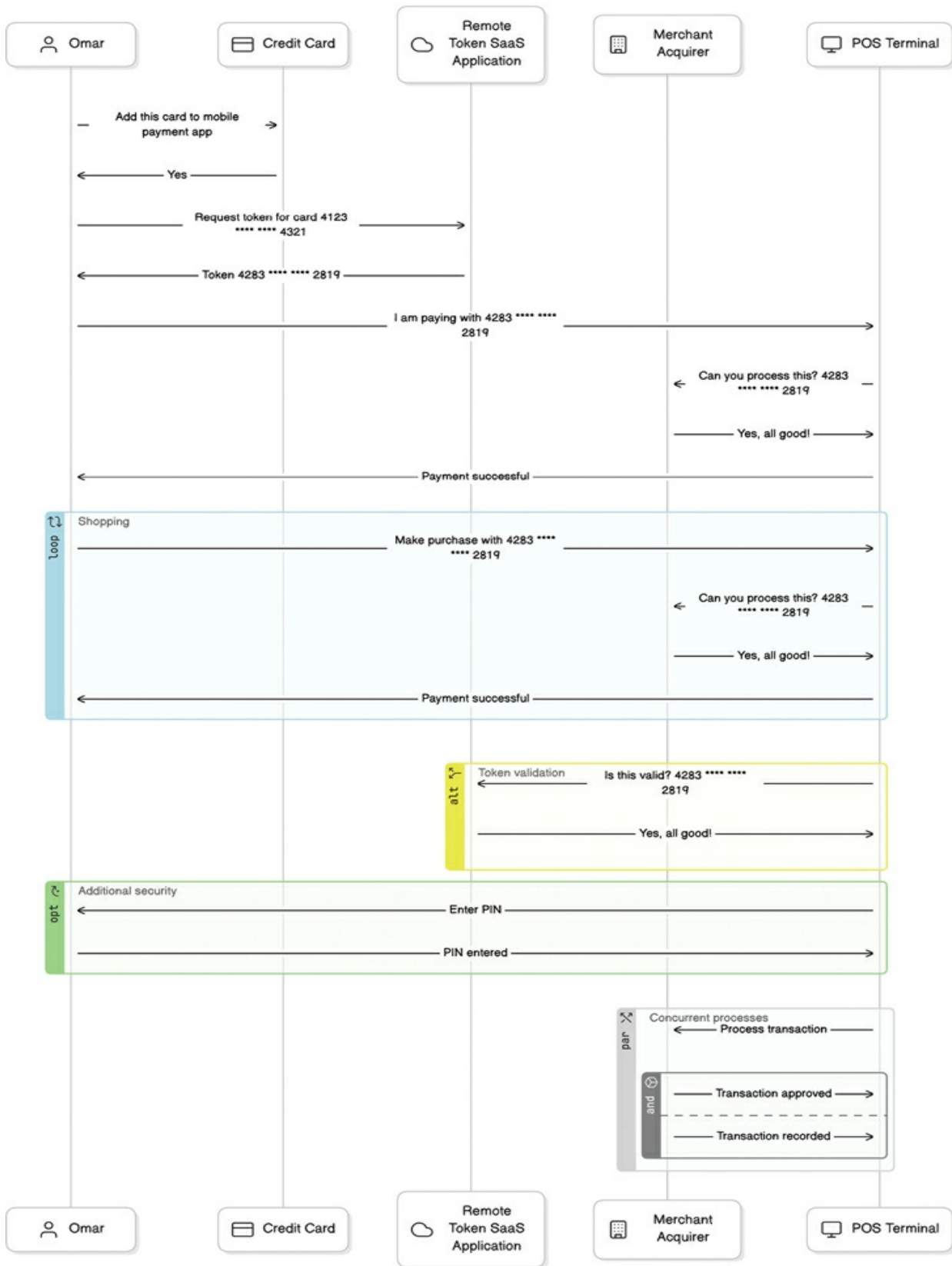


Figure 4-8 Data Tokenization Process

Figure 4-8 illustrates the tokenization process for credit card transactions using a mobile payment app. The process involves several entities: the user (Omar), the credit card, the remote token SaaS application, the merchant acquirer, and the POS terminal.

1. Omar adds his credit card (4123 **** * 4321) to a mobile payment app.
2. The app sends a request for a token to the Remote Token SaaS Application.
3. The Remote Token SaaS Application receives the request and generates a token (4283 **** * 2819) corresponding to Omar's credit card.
4. The token is sent back to the mobile payment app, which Omar can now use for transactions.
5. When making a payment, Omar uses the token (4283 **** * 2819) instead of his actual credit card number.
6. The token is sent to the POS Terminal for the transaction.
7. The POS Terminal sends the token to the Merchant Acquirer to process the payment.
8. The Merchant Acquirer verifies the token with the Remote Token SaaS Application to ensure its validity.
9. Upon successful validation, the Merchant Acquirer authorizes the transaction and informs the POS Terminal.
10. The payment is marked as successful, and the transaction is recorded.
11. An additional validation step might be involved where the token's validity is checked with the Remote Token SaaS Application before proceeding with the transaction. Once validated, the transaction continues as usual.
12. For added security, Omar may be required to enter a PIN during the transaction. Once the PIN is entered and verified, the transaction proceeds.

Data Classification and Tagging

Data classification and tagging in SaaS environments are vital for ensuring data security, regulatory compliance, efficient data management, risk management, operational efficiency, enhanced collaboration, and cost management. These processes help organizations maintain control over their data, protect sensitive information, and meet various legal and regulatory requirements.

Data classification is the process that categorizes data based on its sensitivity and criticality, allowing organizations to apply appropriate DLP policies and controls. Tagging is the process of using metadata to label data with specific classifications, facilitating easier management and protection of sensitive information. [Figure 4-9](#) demonstrates the data classification and tagging process.

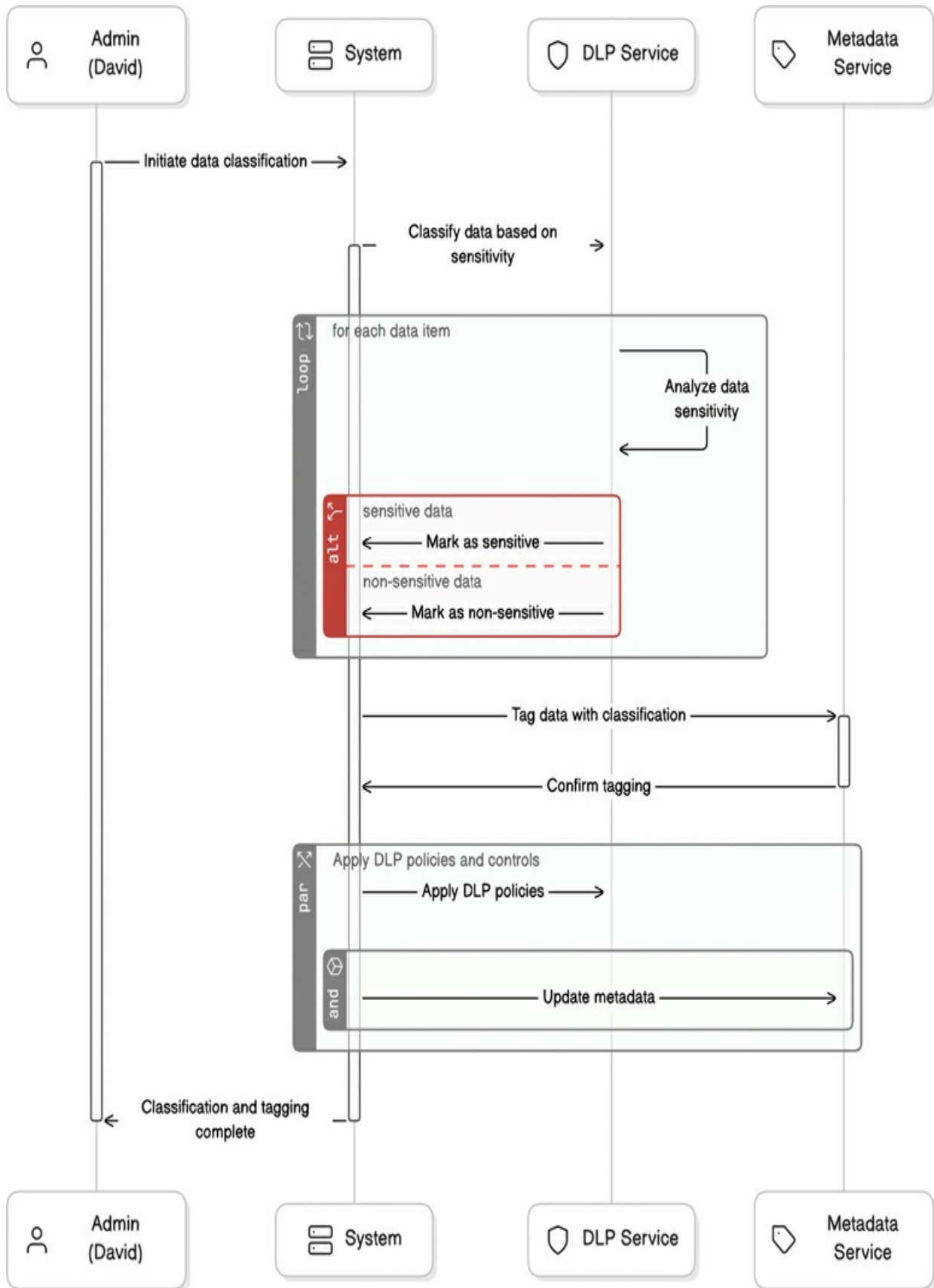


Figure 4-9 Data Classification and Tagging Process

Figure 4-9 shows a detailed flow of how data is classified and tagged in a SaaS environment. It highlights the roles of different services and how they interact to ensure data is correctly classified and protected. The process involves analyzing data sensitivity, tagging data, applying DLP policies, and updating metadata to ensure proper data management and security.

By classifying data based on its sensitivity and importance, organizations can implement appropriate access controls. Only authorized users can access sensitive data, reducing the risk of data breaches. Different levels of data may require different encryption standards. Data classification ensures that highly sensitive data is encrypted with stronger algorithms.

Many regulations, such as GDPR, HIPAA, and CCPA, require organizations to protect specific types of data. Data classification helps in identifying and applying necessary safeguards to comply with these regulations. Classification and tagging facilitate easier data audits and assessments, ensuring that compliance requirements are continuously met.

Data classification helps in managing the data lifecycle, ensuring that data is stored, archived, or deleted according to its classification. This process optimizes storage costs and improves data management efficiency. Tagging data makes it easier to locate and retrieve specific data, improving efficiency in data handling and operations.

By understanding the sensitivity and value of different types of data, organizations can perform better risk assessments and implement targeted security measures to mitigate potential risks. In case of a data breach or security incident, knowing the classification of the affected data helps in assessing the impact and responding appropriately.

Data tagging enables the automation of workflows and processes based on data sensitivity and type. This process leads to more efficient and consistent handling of data. Automated enforcement of data policies becomes more straightforward when data is properly classified and tagged.

Integration of DLP Solutions with Existing Systems

Integrating DLP solutions with existing systems in a SaaS environment requires careful planning and execution to ensure seamless operation and comprehensive data protection. Here are key considerations for successful integration.

You should always ensure that the DLP solution is compatible with the existing SaaS applications and infrastructure. Look for solutions that offer APIs and connectors for easy integration. You should implement a centralized DLP management platform to oversee and coordinate DLP policies across various SaaS applications and services. Doing so simplifies administration and ensures consistent policy enforcement.

Use automated tools (such as Umbrella Security Internet Gateway and CloudLock) to enforce DLP policies in real time, reducing the reliance on manual interventions and minimizing the risk of human error. Automation also ensures that policies are applied uniformly across the organization. You should also integrate continuous monitoring capabilities to detect and respond to potential data loss incidents promptly. Ensure that the DLP solution provides detailed reporting and analytics to support compliance and auditing requirements. Integrate the DLP solution with the organization's incident response framework. This integration enables quick escalation and resolution of data loss incidents, minimizing potential damage.

Another best practice is to choose a DLP solution that can scale with the organization's growth and adapt to evolving security needs. Flexibility is crucial for accommodating new SaaS applications and changes in the threat landscape.

DLP is essential for securing sensitive information and ensuring compliance in a SaaS environment. By leveraging a combination of tools and techniques such as encryption, access controls, data masking, activity monitoring, and cloud access security brokers (CASBs), organizations can effectively prevent data loss and protect their valuable assets.

Note

You will learn more about CASBs later in this chapter.

Identity and Access Management (IAM)

The convenience and flexibility of SaaS come with significant security challenges, particularly around identity and access management. Effective IAM practices are essential to protect sensitive data, ensure regulatory compliance, and prevent unauthorized access. Let's go over some of the best practices for implementing robust IAM in a SaaS environment.

Centralizing Identity Management

Centralizing identity management simplifies user administration, enhances security, and provides a single point of control for user access. Many organizations use a centralized IAM solution to manage all user identities across various SaaS applications. You can integrate your IAM system with an enterprise directory service (e.g., Microsoft Active Directory, LDAP) to streamline user provisioning and deprovisioning.

Account Takeover Attacks

Account takeover (ATO) attacks occur when a malicious actor gains unauthorized access to a user's online account. This type of attack is often achieved through various methods such as phishing, credential stuffing, social engineering, or exploiting security vulnerabilities. Once attackers have access, they can misuse the account in numerous ways, including stealing sensitive information, committing fraud, or launching further attacks.

How Account Takeover Attacks Work

- **Phishing:** Attackers trick users into providing their login credentials by masquerading as a legitimate entity.
- **Credential Stuffing:** This type of attack uses lists of stolen username/password combinations from one breach to attempt access on other platforms, exploiting users who reuse passwords.
- **Social Engineering:** This approach involves manipulating individuals into revealing confidential information.

- **Exploiting Security Vulnerabilities:** Attackers take advantage of weaknesses in software or security protocols to gain access.

Impact on SaaS Environments

Unauthorized access can lead to the exposure of sensitive data stored within the SaaS platform, affecting not only the compromised account but potentially others as well. Attackers may commit fraud, causing financial loss to both the users and the SaaS provider. This attack can include unauthorized purchases, financial transactions, or fraudulent use of services.

SaaS providers may suffer significant damage to their reputation by these or similar attacks, leading to loss of customer trust and potential business losses. If sensitive data is compromised, SaaS providers may face penalties under data protection regulations such as GDPR or CCPA. Multifactor authentication (MFA) is one of the best preventive measures for SaaS providers. We will discuss MFA in detail later in this chapter.

Centralized identity management in SaaS environments plays a critical role in strengthening security and reducing the risk of ATO attacks. By providing a unified approach to managing user identities, access control, and authentication, these systems streamline how organizations govern access across multiple applications and services.

A major advantage of this method is the ability to enforce multifactor authentication consistently across all integrated applications. This additional layer of security goes beyond passwords and significantly reduces the likelihood of credential compromise. Single sign-on (SSO) further enhances both security and user experience by allowing users to authenticate once and seamlessly access all authorized resources, reducing the attack surface for phishing and password reuse.

Modern identity platforms also enable adaptive authentication, dynamically adjusting authentication requirements based on contextual risk factors such as user behavior, device, or location. These systems provide unified monitoring and logging of user activity across all connected applications, making it

easier to spot anomalies or suspicious activity. Advanced analytics can flag unusual patterns and trigger alerts or automated responses to mitigate potential threats in real time.

Automated Provisioning and Deprovisioning

Automated provisioning and deprovisioning in SaaS environments refer to the use of predefined workflows and policies to create (provision) and remove (deprovision) user accounts and access permissions without manual intervention. When a new employee joins an organization, these systems can instantly create accounts and assign the correct level of access across all relevant SaaS applications. This approach can also apply to customers, contractors, and business partners, ensuring they receive only the access they need.

In most mature organizations, this process is driven by role- or team-based provisioning, where access is automatically aligned with the user's job function. This approach is typically implemented using role-based access control (RBAC), which maps permissions to roles (e.g., "HR Analyst" or "Sales Manager") rather than to individual users. By doing so, organizations maintain consistency, reduce errors, and make access reviews and audits much simpler.

Automated deprovisioning is equally critical: When a user leaves the organization, changes roles, or a contractor engagement ends, access is revoked automatically across all systems. Deprovisioning greatly reduces the risk of orphaned accounts or excessive permissions—both of which are common attack vectors for insider threats or compromised credentials.

Figure 4-10 illustrates centralized identity management services in SaaS environments.

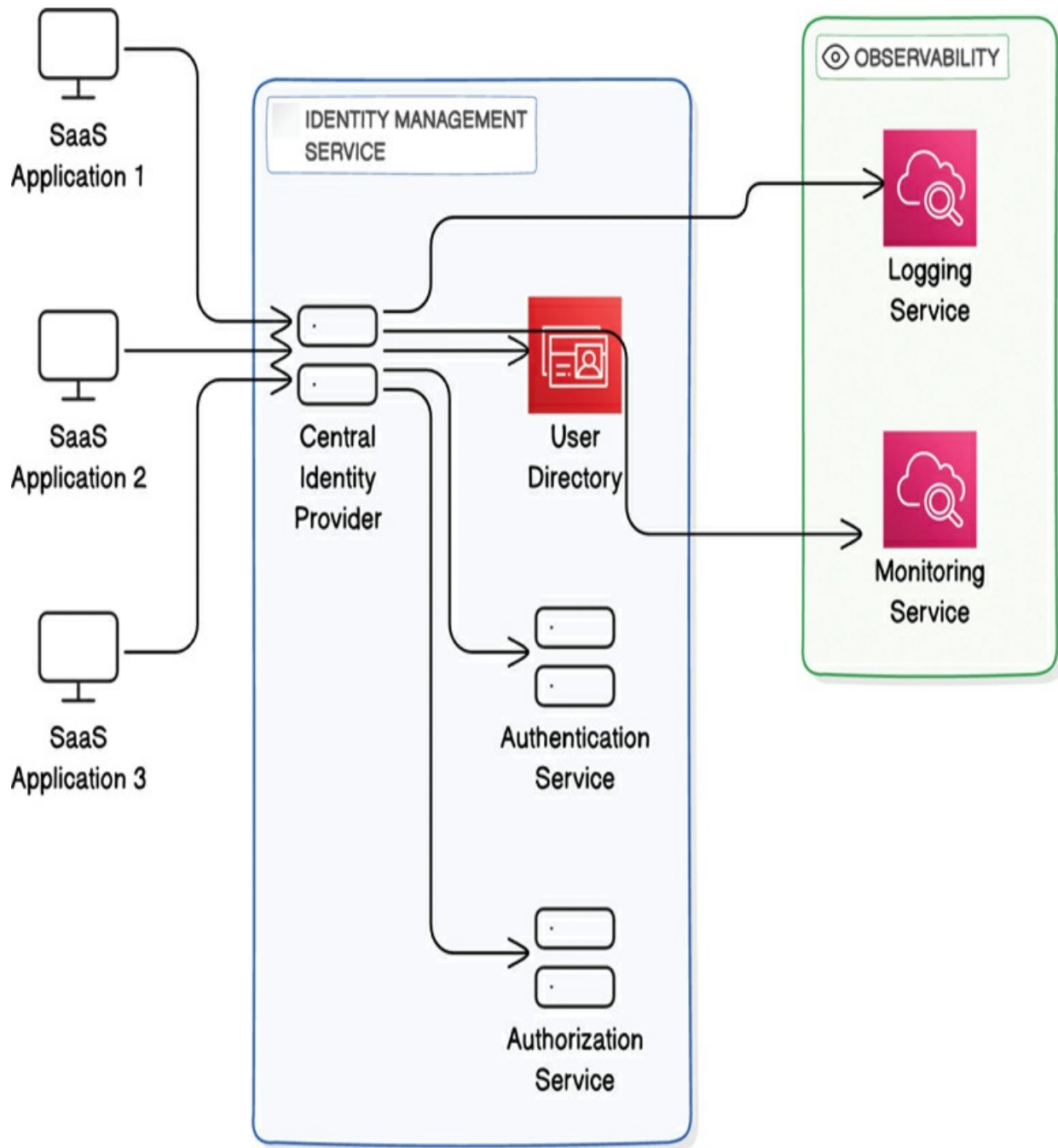


Figure 4-10 Centralized Identity Management

Implementing a Zero Trust security model requires a clear understanding of user identities and the resources they need access to. This implementation necessitates creating an accurate user inventory. The Center for Internet Security (CIS) advises keeping a precise inventory of both authorized and unauthorized devices and users to ensure that only authorized users can access the system. Without an accurate user inventory, identifying and

addressing security risks becomes challenging. [Figure 4-11](#) shows the user inventory and identification maturity level.

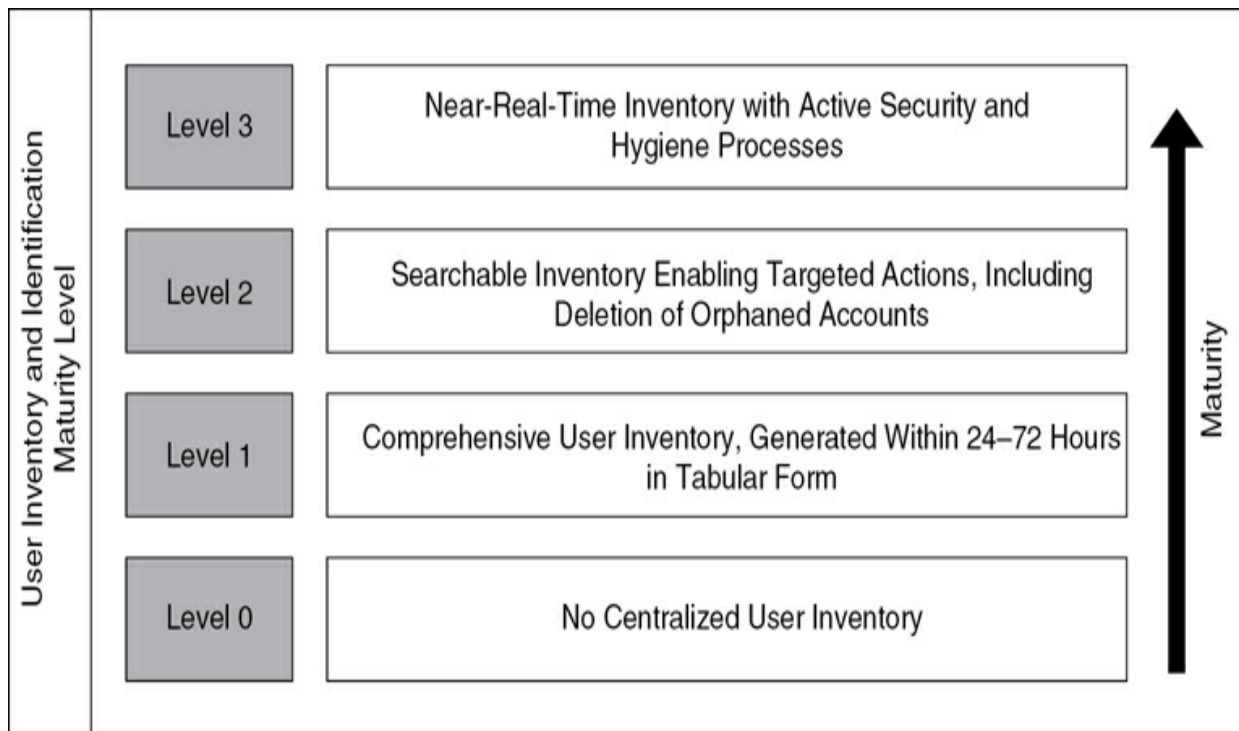


Figure 4-11 User Inventory and Identification Maturity Level

Identity Threat Detection, Response, and Oort

Cisco acquired a company named Oort that provides advanced identity threat detection and response (ITDR) capabilities. With the rise in identity-based attacks, companies today must be capable of detecting when enterprise identities are misused, exploited, or stolen. This task is particularly important as businesses rapidly adopt public cloud services and the number of both human and nonhuman identities grows exponentially. Detecting identity-based activities is crucial given attackers' tendency to leverage credentials and access key identity providers like Active Directory (AD), Okta, and Azure AD.

ITDR is a newly emerging security category that complements other detection solutions such as network detection and response (NDR), extended detection and response (XDR), and endpoint

detection and response (EDR).

ITDR focuses on the protection of access, privileges, and credentials, as well as the systems that manage them. ITDR represents a significant advancement, introducing a new suite of security tools that enhance identity protection.

Enforcing Multifactor Authentication (MFA)

You should make multifactor authentication mandatory for accessing all critical SaaS applications. Why does it matter? MFA adds an additional layer of security by requiring users to provide two or more verification factors to access SaaS applications. Most commonly, these verification factors include something you know (such as a password) combined with something you have (such as a mobile device for push notifications, a time-based one-time password app, or a physical security key). By combining factors, MFA significantly reduces the risk of account compromise, even if a password is stolen through phishing, credential stuffing, or other attacks.

[Figure 4-12](#) shows the different maturity levels of MFA adoption within an organization.

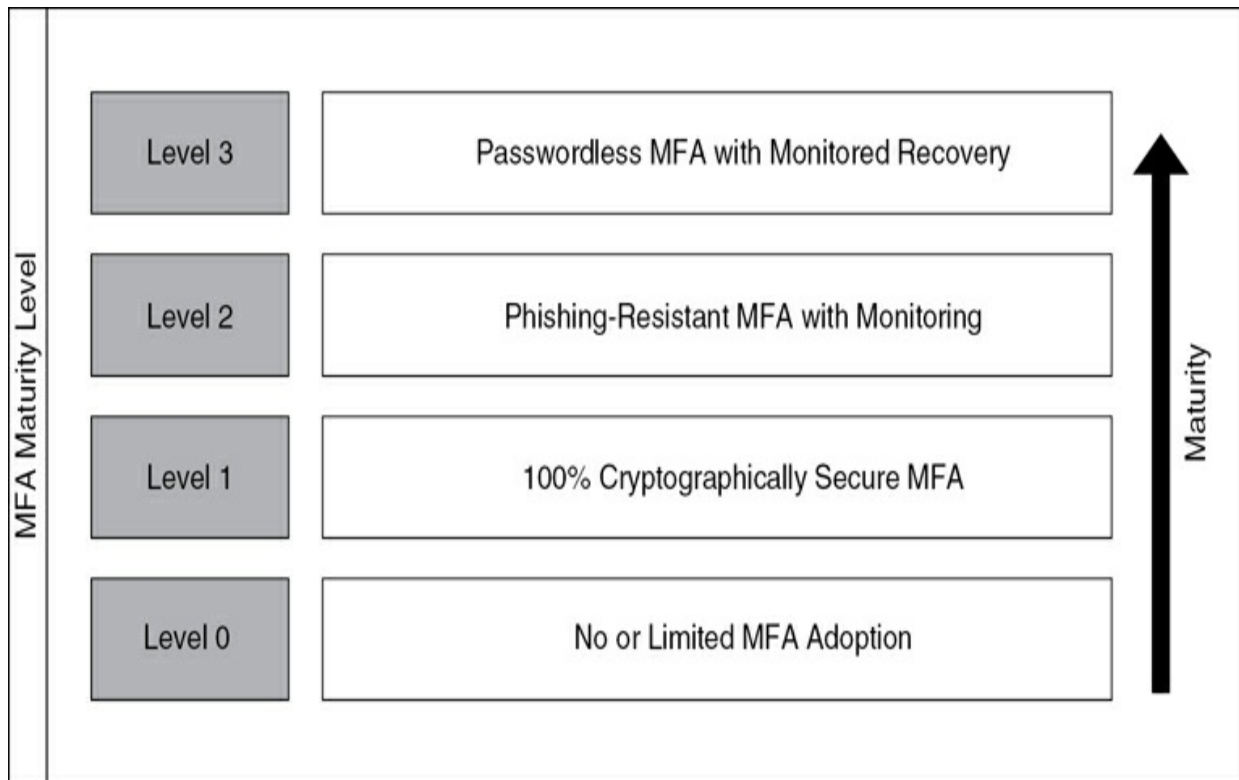


Figure 4-12 MFA Adoption Maturity Level

The following is an explanation of the different maturity levels of MFA adoption within an organization, as shown in [Figure 4-12](#):

- **Level 0:** At this base level, the organization either has not implemented MFA or has only limited adoption. This means there is minimal additional security beyond basic passwords, making the system more vulnerable to attacks.
- **Level 1:** At this stage, the organization has implemented MFA across the board, ensuring that all access points are protected by cryptographically secure methods. There are also clear exception policies in place, detailing specific cases where MFA might not be applicable.
- **Level 2:** The organization employs MFA methods that are resistant to phishing attacks. Additionally, there is active monitoring of user activity and MFA-based threat detection in place. This monitoring enhances security by identifying and mitigating potential threats in real time.
- **Level 3:** At the highest level, the organization has adopted a

passwordless MFA policy, reducing the reliance on traditional passwords. There is also a monitored recovery process in place, ensuring that any issues with MFA can be tracked and resolved efficiently. This level represents the most advanced and secure form of MFA implementation.

Using Cisco Duo and MFA

Cisco Duo is a cloud-based security platform designed to protect access to applications and data by verifying the identity of users and the health of their devices before granting access. It offers MFA, SSO, and adaptive access policies to secure both cloud and on-premises applications.

Adaptive access policies enable dynamic access controls based on user behavior, device health, location, and other contextual factors to ensure secure access. The Cisco Duo Device Trust feature verifies the security posture of devices attempting to access resources, ensuring they meet compliance and security requirements.

Cisco Duo also supports integration with a wide range of applications and services, both cloud-based and on-premises, ensuring comprehensive security coverage.

Cisco Duo Tutorials

Being a cloud-based solution, Cisco Duo can be rapidly deployed and scaled to meet the needs of growing organizations without significant infrastructure investments. Duo has an amazing collection of videos and tutorials at <https://duo.com/resources/videos> and <https://duo.com/resources/videos/archive/technical-setup>.

Understanding FIDO Technology

Fast Identity Online (FIDO) technology is a set of open standards developed by the FIDO Alliance (fidoalliance.org) for strong authentication. FIDO aims to reduce reliance on passwords by enabling secure and easy-to-use

authentication methods. The core of FIDO's approach is to use public-key cryptography for authentication, ensuring that authentication data is never shared with or stored by online services, reducing the risk of credential theft.

The following are the key components of FIDO:

- **FIDO Universal Authentication Framework (UAF):** This protocol allows users to register and authenticate to online services using local biometric solutions (e.g., fingerprint, facial recognition) without needing a password.
- **FIDO Universal 2nd Factor (U2F):** This protocol enables two-factor authentication using a hardware device, such as a USB security key, to provide an additional layer of security.
- **FIDO2:** This standard combines W3C's Web Authentication (WebAuth) and FIDO Alliance's Client-to-Authenticator Protocol (CTAP) to enable passwordless authentication across the web. FIDO2 supports both biometric and hardware-based authentication.

How can FIDO technology help SaaS?

FIDO technology leverages public-key cryptography to deliver secure, phishing-resistant authentication for SaaS applications. In a FIDO-based system, the private key remains securely stored on the user's device—such as a hardware security key, phone, or laptop secure enclave—while the public key is registered with the SaaS provider. When a user logs in, the server issues a cryptographic challenge that can only be signed by the private key on the user's device. This process is resistant to phishing, on-path attacks (also known as man-in-the-middle [MITM]), and replay attacks because there are no shared secrets (like passwords) that can be intercepted or reused.

Beyond security, FIDO dramatically improves user convenience by allowing login via biometrics (fingerprint, face scan) or hardware tokens, eliminating the need to type passwords. This capability not only reduces friction during login but also addresses the long-standing issues associated with traditional password-based authentication.

Passwords have been the default authentication mechanism for decades, but they come with significant drawbacks:

- **Reuse Across Multiple Accounts:** Users often reuse the same password (or slight variations) across work and personal accounts, making them vulnerable to credential-stuffing attacks if any single site is breached.
- **Complexity Requirements:** Traditional password policies that demand frequent changes, minimum complexity, and arbitrary rules often lead to weaker security. Users may resort to predictable patterns or write down passwords, increasing risk.
- **Password Fatigue:** Constant password resets and complex requirements create frustration, slow productivity, and increase help desk tickets—costing organizations time and money.

Until full passwordless adoption is possible, password managers can reduce risk by securely storing and generating unique, complex passwords for each account. This approach mitigates the risk of reuse and weak credentials, while reducing the cognitive burden on users.

Transitioning to passwordless authentication with FIDO/WebAuthn offers even greater benefits:

- **Phishing-Resistant Authentication:** No shared secret is transmitted, so attackers cannot steal or replay credentials.
- **Lower Operational Overhead:** Fewer password resets mean less work for IT and reduced costs.
- **Better User Experience:** Faster, simpler logins improve adoption and reduce friction.
- **Scalability:** Passwordless authentication is ideal for SaaS environments with large, distributed user bases—from employees to customers and partners.

FIDO can be easily integrated into existing SaaS platforms and scaled as the organization grows, ensuring consistent security across all applications.

FIDO standards are widely adopted and supported by major browsers and operating systems, ensuring broad compatibility and ease of integration into various SaaS applications. FIDO works across different devices and platforms, providing a seamless and secure authentication experience for users regardless of their device or operating system.

Figure 4-13 shows a high-level overview of the FIDO authentication process.

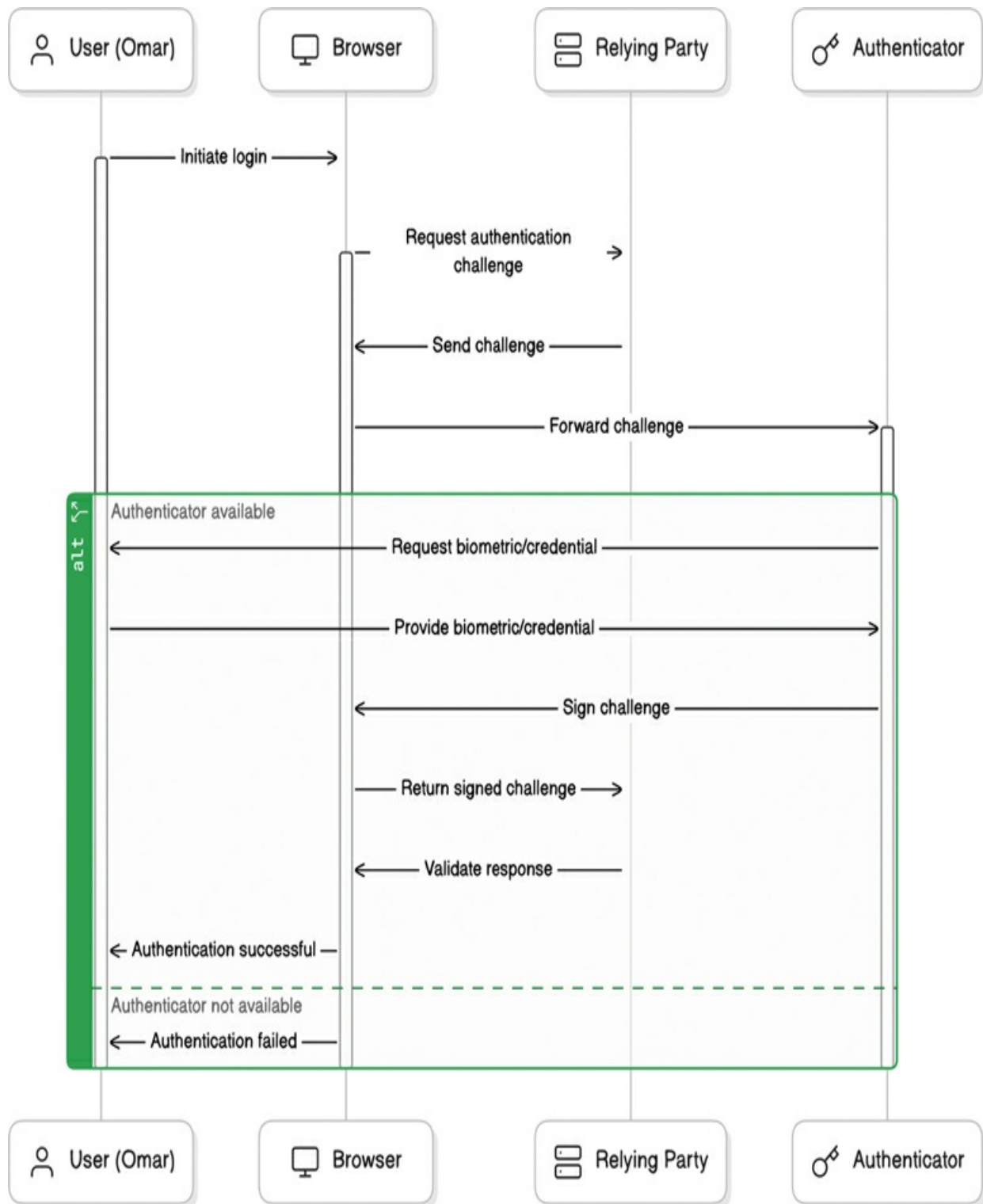


Figure 4-13 High-Level FIDO Authentication Process

The following are the high-level steps of the FIDO authentication process illustrated in [Figure 4-13](#):

1. The user, Omar, initiates the login process via the browser.
2. The browser sends a request to the relying party for an authentication challenge.
3. The relying party generates a challenge and sends it back to the browser.
4. The browser forwards the challenge to the authenticator.
5. If authenticator is available, the authenticator prompts the user to provide a biometric verification (e.g., fingerprint, facial recognition) or another form of credential.
6. The user provides the required biometric data or credential.
7. The authenticator uses the user's private key to sign the challenge.
8. The signed challenge is returned to the browser.
9. The browser forwards the signed challenge to the relying party, which uses the stored public key to validate the signature.
10. If the signature is valid, the relying party confirms that the authentication is successful, granting the user access.
11. If the authenticator is not available or the biometric/credential verification fails, the authentication process terminates, and the user is denied access.

[Figure 4-14](#) shows the FIDO registration process.

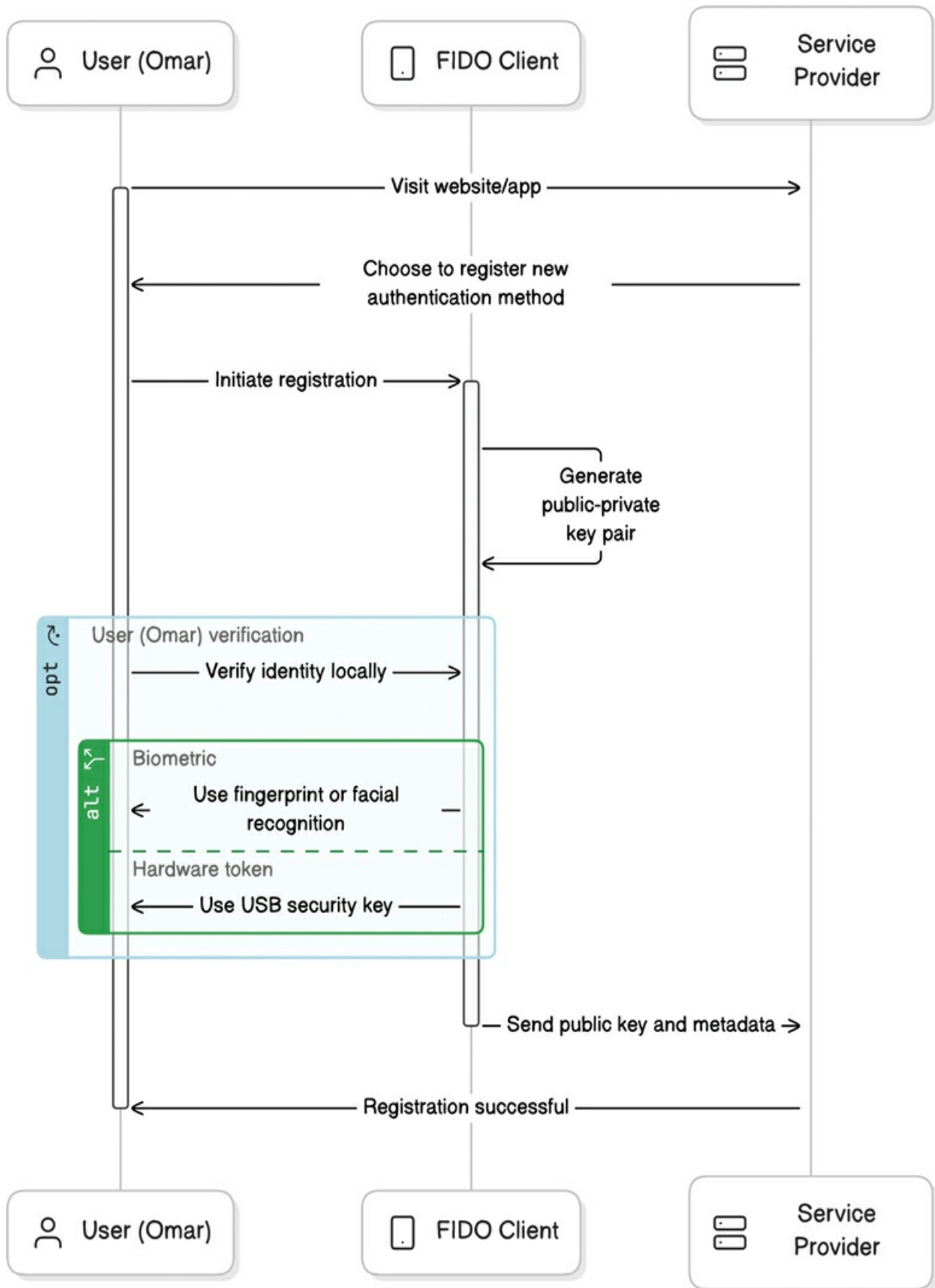


Figure 4-14 High-Level FIDO Registration Process

In [Figure 4-14](#) the user visits a service provider's website or application and chooses to register a new authentication method. The FIDO client (e.g., a browser or mobile app) on the user's device generates a public-private key pair. The user verifies their identity locally using a biometric method (like fingerprint or facial recognition) or a hardware token (like a USB security key). The public key is sent to the service provider, along with metadata about the device and user's verification method. The private key remains securely stored on the user's device and is never shared.

[Figure 4-15](#) shows the user device's FIDO registration process.

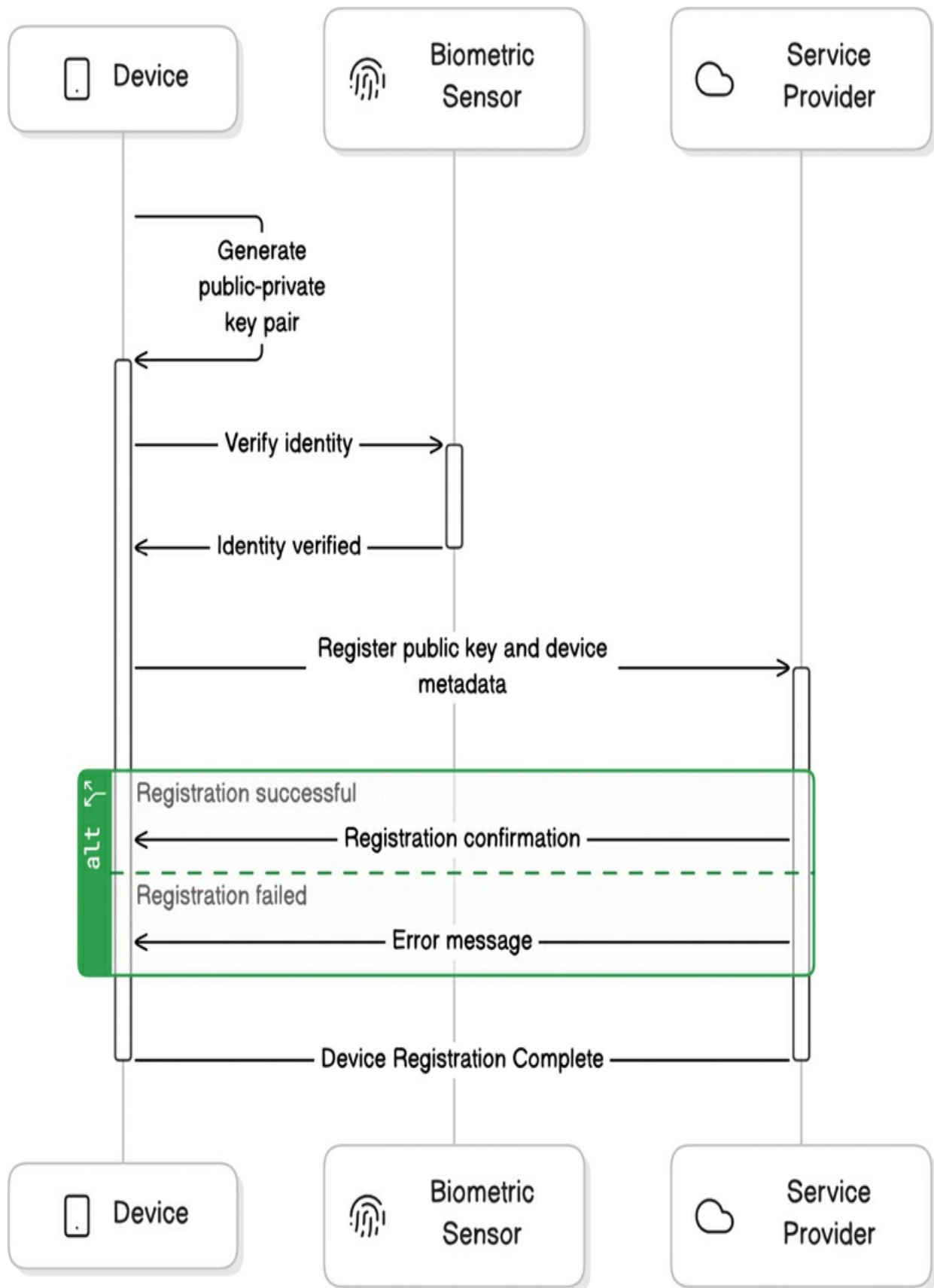


Figure 4-15 High-Level FIDO Device Registration Process

In [Figure 4-15](#), the user's device generates a unique public-private key pair. The user verifies their identity locally using a biometric sensor or hardware token. The public key, along with device metadata, is registered with the service provider. The private key never leaves the user's device.

Implementing Single Sign-On (SSO)

Single sign-on enhances user convenience by allowing access to multiple SaaS applications with a single set of credentials, reducing password fatigue and improving security. [Figure 4-16](#) shows a high-level overview of an SSO authentication process in a SaaS application such as Webex.

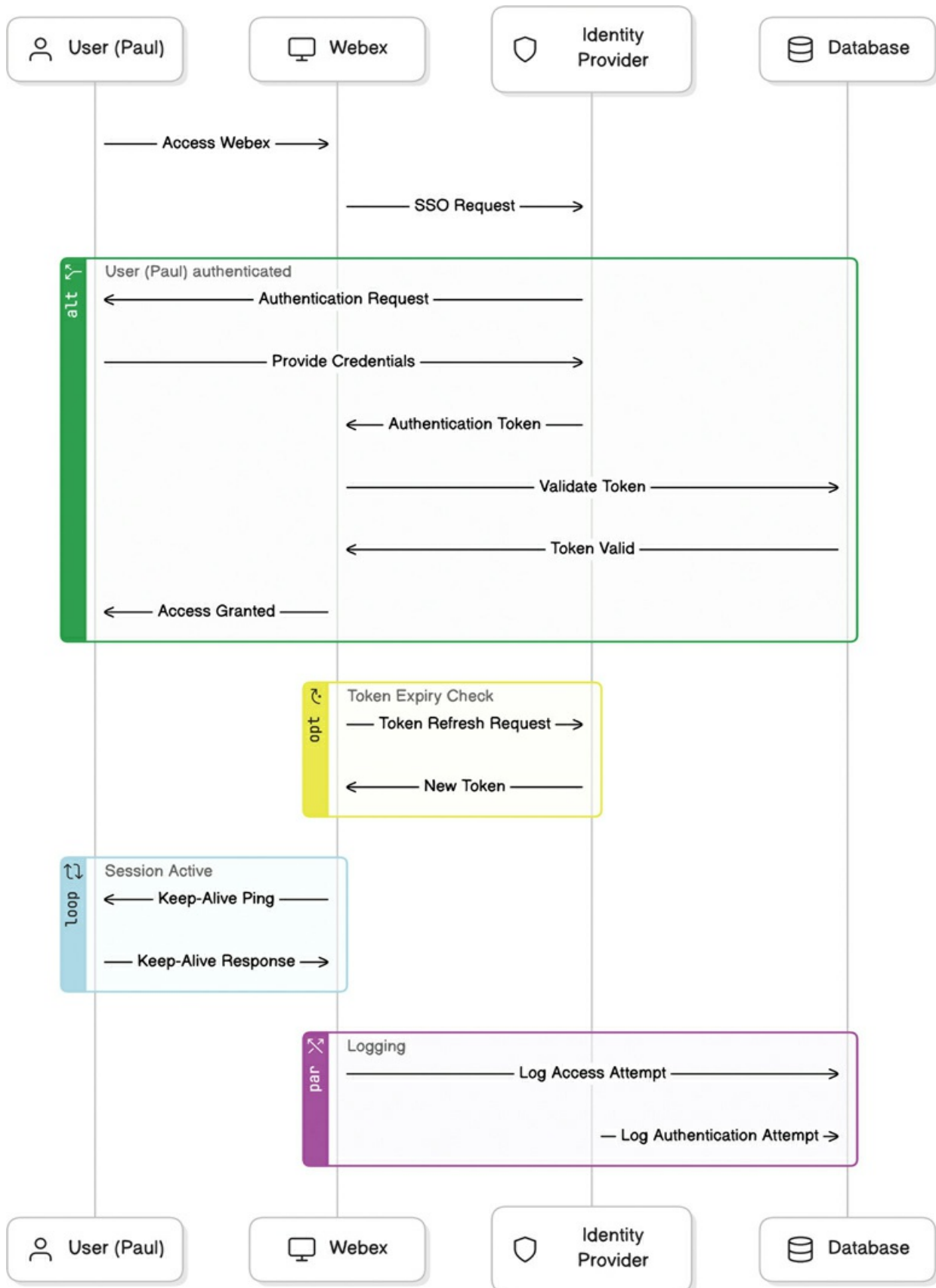


Figure 4-16 High-Level SSO Authentication Process in a SaaS Application

Your organization should adopt SSO solutions to provide seamless access across SaaS applications. Ensure the SSO solution supports standard protocols like Security Assertion Markup Language (SAML), OAuth, or OpenID Connect for interoperability.

Looking at SAML, OAuth, and OpenID Connect

SAML is an open standard for exchanging authentication and authorization data between parties, specifically between an identity provider (IdP) and a service provider (SP). It allows users to authenticate once and gain access to multiple applications, providing single sign-on.

Let's look at [Figure 4-16](#) again. When the user tries to access a SaaS application, the SP redirects the user to the IdP for authentication. The user authenticates with the IdP. In the case of SAML, the IdP generates a SAML assertion (a secure XML document including token information) and sends it back to the SP. The SP validates the assertion and grants the user access to the application.

Why does SAML matter in SaaS? SAML is widely adopted and supported by many SaaS applications, ensuring interoperability between different systems. It allows organizations to manage authentication centrally, enhancing security and simplifying user management. Users can access multiple SaaS applications with a single set of credentials, reducing password fatigue and improving productivity.

OAuth is an open standard for access delegation, commonly used to grant websites, APIs, or applications limited access to user information without exposing passwords. It focuses on authorization, allowing users to grant third-party applications access to their resources hosted by another service. [Figure 4-17](#) shows how OAuth works from a high-level perspective.

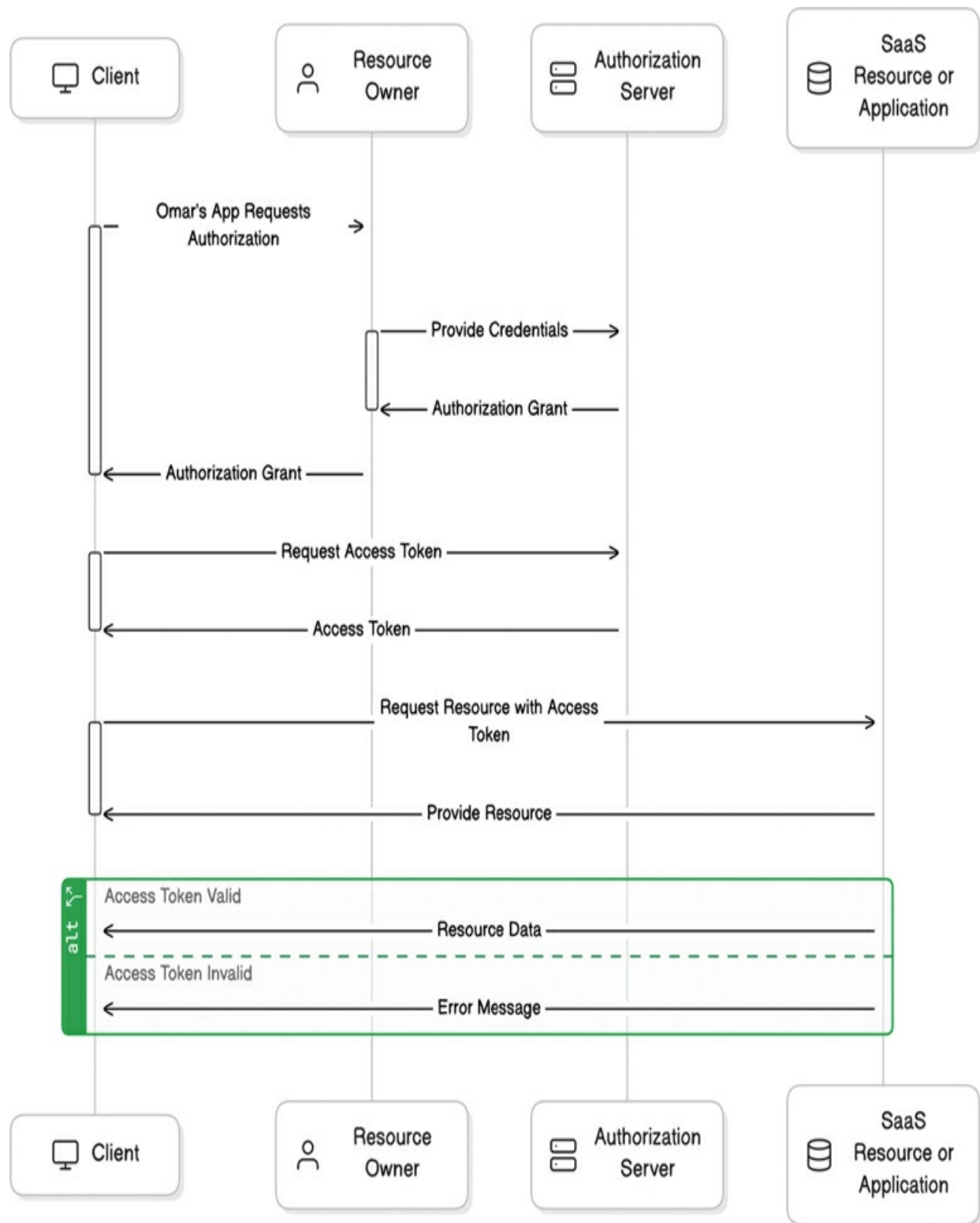


Figure 4-17 High-Level OAuth Process

In [Figure 4-17](#), the user grants permission to a third-party application

(Omar's app) to access their resources. The third-party application receives an authorization grant (usually a code) and exchanges the authorization grant for an access token from the authorization server. The authorization server issues an access token, and the third-party application uses the access token to access the user's resources from the SaaS resource server.

OAuth allows users to grant specific permissions to third-party applications, improving security by limiting access to only necessary data. It also enables seamless integration between SaaS applications and third-party services, enhancing functionality and user experience. By not sharing passwords and using tokens, OAuth reduces the risk of credential compromise.

OpenID Connect is an identity layer built on top of the OAuth 2.0 protocol. It enables clients to verify the identity of users and obtain basic profile information. It provides a standardized way to authenticate users, complementing OAuth's authorization capabilities with authentication features. [Figure 4-18](#) shows the high-level process of OpenID Connect in a SaaS environment.

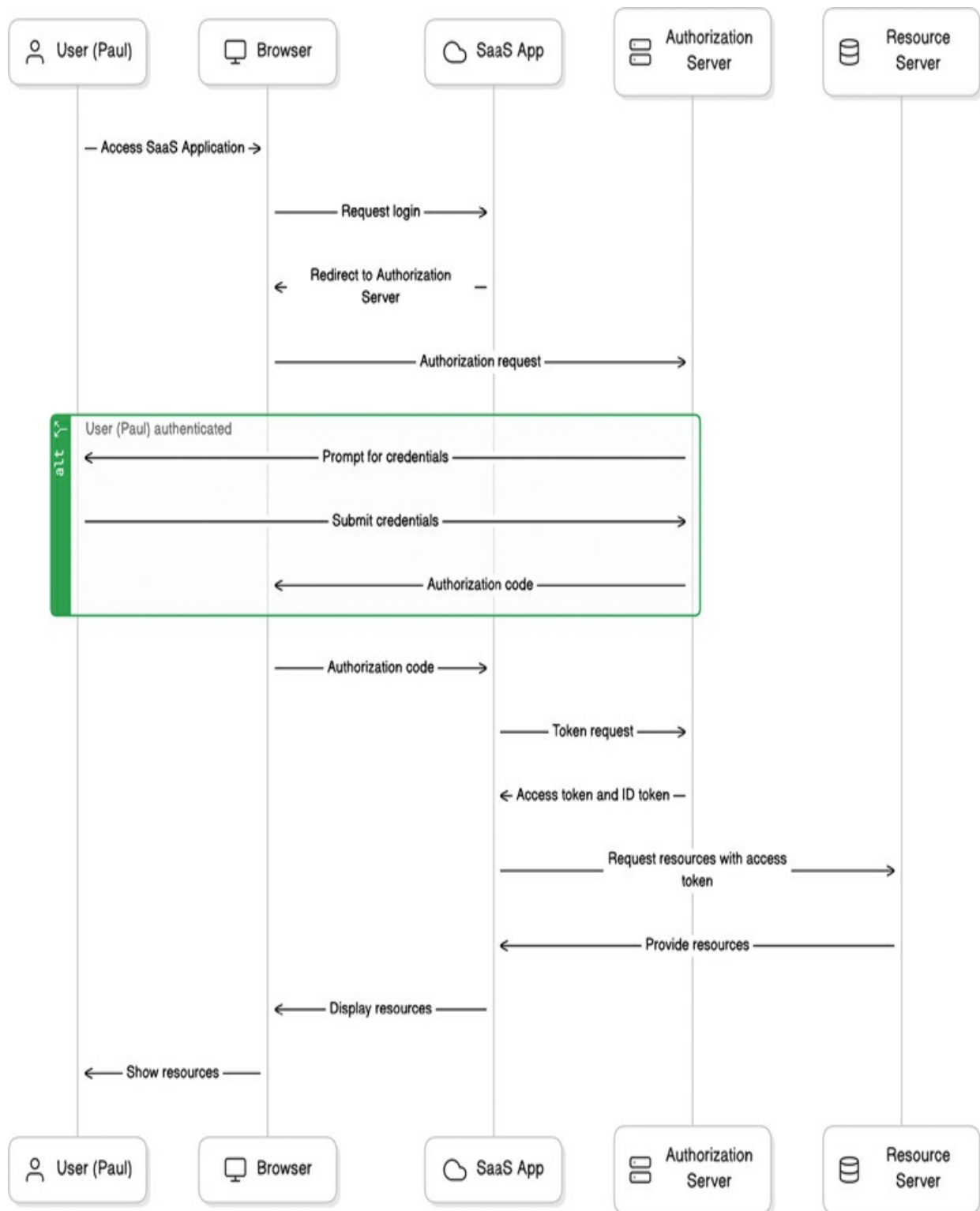


Figure 4-18 High-Level OpenID Connect Process

Figure 4-18 illustrates the OpenID Connect (OIDC) authentication process

involving a user (Paul), a browser, a SaaS application, an authorization server, and a resource server. The following is a step-by-step explanation of the process:

1. The user (Paul) attempts to access the SaaS application through their browser.
2. The SaaS application detects that the user is not authenticated and initiates an authentication request.
3. The SaaS application redirects the user to the authorization server for authentication.
4. The browser sends an authorization request to the authorization server.
5. The authorization server prompts the user to provide their login credentials.
6. The user submits their credentials (e.g., username and password).
7. Upon successful authentication, the authorization server issues an authorization code and sends it to the browser.
8. The browser sends the authorization code to the SaaS application.
9. The SaaS application sends the authorization code to the authorization server in exchange for tokens.
10. The authorization server issues an access token and an ID token and sends them back to the SaaS application.
11. The SaaS application uses the access token to request resources from the resource server.
12. The resource server validates the access token and provides the requested resources to the SaaS application.
13. The SaaS application displays the resources to the user through the browser.
14. The user can now view and interact with the resources within the SaaS application.

Why does OpenID Connect matter in SaaS? OpenID Connect provides both authentication (user identity verification) and authorization (access control) in

a unified protocol. It simplifies the login experience for users by allowing them to use a single identity across multiple SaaS applications. It also ensures secure and scalable authentication and authorization processes, leveraging the robustness of OAuth 2.0.

In short, SAML is ideal for traditional enterprise SSO use cases with robust support for centralized authentication. OAuth excels in delegated authorization scenarios, allowing users to grant specific permissions to third-party applications without sharing credentials. OpenID Connect builds on OAuth 2.0 to provide a comprehensive solution for both authentication and authorization, enhancing security and user experience.

Monitoring and Analyzing User and Application Account Activities

Continuous monitoring of user activities helps detect and respond to suspicious behavior or potential security incidents in real time. However, user accounts are not the only thing that you need to monitor. You must also monitor application (or service) accounts. Before we go in depth, let's define what service accounts are. Service accounts, also known as application accounts, are special types of nonhuman accounts used to run applications or services in a cloud environment. They are designed to provide a way for applications or services to authenticate and interact with other services, databases, APIs, and resources without human intervention.

Service accounts are not tied to individual users but to applications or services. They are used for automated processes, allowing applications to perform tasks without manual intervention. Service accounts typically have specific permissions and roles that grant them access only to the resources and actions they need. They are used to authenticate and authorize applications or services to interact with cloud resources securely.

Service accounts can be used to access databases, storage buckets, and other cloud services. They can also enable secure communication between different services within a cloud environment. Continuous integration and continuous deployment (CI/CD) pipelines often use service accounts to access and deploy resources automatically.

Service Accounts in Major Cloud Providers

Even though this book is about SaaS, it is good to know how service accounts operate in the major cloud providers. In Google Cloud Platform (GCP), service accounts are a type of identity used by Google Cloud services. They are represented by an email address. In AWS, service accounts can be represented by IAM roles, which are assumed by services or applications. In Azure, service principals are used as service accounts, allowing applications to access resources securely.

Monitoring account activities helps detect unauthorized access, potential security breaches, and malicious activities. By analyzing logs and user behavior, organizations can identify and respond to security threats in real time, reducing the risk of data breaches and other security incidents.

Many industries are subject to regulatory requirements that mandate the monitoring and logging of user activities. Ensuring compliance with standards such as GDPR, HIPAA, FedRAMP, and PCI-DSS requires robust monitoring mechanisms to track and audit user and application actions.

Monitoring for Operational Efficiency

Monitoring activities provide insights into the usage patterns and performance of applications and services. This data can be used to optimize resource allocation, troubleshoot issues, and enhance overall operational efficiency.

Best Practices for Monitoring and Analyzing Activities

Ensure that all user and application activities are logged comprehensively. This information includes login attempts, resource access, configuration changes, and any other relevant actions. Logs should be detailed and include timestamps, user IDs, IP addresses, and other contextually relevant information.

As previously discussed in this chapter, you should adopt centralized logging

solutions that aggregate logs from various sources into a single repository. Doing so makes it easier to analyze and correlate events across different systems and services. Tools like AWS CloudTrail, Google Cloud Logging, and Azure Monitor are excellent for centralized log management.

Integrating Splunk with AWS CloudTrail, Google Cloud Logging, and Azure Monitor

You can integrate Splunk with AWS CloudTrail, Google Cloud Logging, and Azure Monitor to centralize and analyze log data from these cloud services. CloudTrail captures API calls and sends log files to an S3 bucket. You can then create an S3 bucket to store the CloudTrail logs. Once you make sure that you have appropriate permissions for the bucket, you can create an IAM role with permissions to read from the S3 bucket and attach the necessary policies for S3 access.

In Splunk, install the Splunk Add-on for AWS CloudTrail from Splunkbase (<https://splunkbase.splunk.com/>). Splunkbase is an online marketplace for apps and add-ons that extend the functionality of Splunk, a powerful platform for searching, monitoring, and analyzing machine-generated big data. In the Splunk Add-on for AWS CloudTrail, you can configure the data inputs and provide the S3 bucket details and IAM role credentials. Once the integration is set up, Splunk will start ingesting CloudTrail logs.

You should always configure alerts and notifications for critical events, such as failed login attempts, unauthorized access, and changes to sensitive configurations. Timely alerts allow for immediate investigation and response to potential issues.

Also, perform regular audits and reviews of logs and account activities. This step helps identify any overlooked anomalies, ensures compliance with policies, and enhances the overall security posture.

You should also implement access management solutions like Duo, Okta, AWS IAM, and Azure AD to enforce policies and monitor account activities.

These tools offer detailed logs and reports on user access and actions.

Access Control Mechanisms

Access control mechanisms are important for ensuring that only authorized users can access specific resources within SaaS applications. These mechanisms help protect sensitive data, maintain compliance with regulations, and secure the overall application environment. The most popular access control mechanisms in SaaS include

- **Role-Based Access Control (RBAC):** RBAC assigns permissions to users based on their roles within the organization. Each role is granted specific access rights, and users are assigned roles according to their job functions. This mechanism is ideal for organizations with clearly defined roles and responsibilities. It is commonly used in enterprise SaaS applications like CRM and ERP systems. For example, a software engineer will have access to source code and related tools, whereas a sales representative has access to customer data and sales tools.
- **Attribute-Based Access Control (ABAC):** ABAC grants access based on user attributes, resource attributes, and environmental conditions. Policies are defined using attributes such as user department, job function, data sensitivity, and time of access. ABAC is suitable for dynamic and complex environments where access decisions need to consider multiple attributes and conditions. An employee can access a document if they are in the HR department, and it is during business hours.
- **Discretionary Access Control (DAC):** DAC allows resource owners to control access to their resources. The owner can grant or revoke access permissions to other users at their discretion. For example, a project lead can grant access to project files to team members as needed. DAC is the most fundamental access control model in this list.
- **Mandatory Access Control (MAC):** MAC enforces access policies based on the sensitivity of the information and the clearance level of the users. Access decisions are made by a central authority rather than by the resource owner. MAC is commonly used in government and military applications where strict access controls and data classification are

required. Examples of MAC implementations include SELinux, AppArmor, and others. For a comparison of several Linux Security Modules (LSMs) such as SELinux, AppArmor, Yama, TOMOYO Linux, and Smack, see <https://becomingahacker.org/bf7f0a1789cf>.

Continuous Monitoring and Incident Response

We previously discussed monitoring user and service accounts. As organizations increasingly rely on SaaS applications to run their operations, the need for robust monitoring and auditing mechanisms becomes paramount. Continuous monitoring and regular auditing of all activities (not only user and service accounts) are essential for maintaining security, ensuring compliance with industry regulations, and protecting sensitive data from unauthorized access.

Continuous monitoring involves tracking and analyzing all activities within a SaaS application in real time. Doing so helps in identifying and responding to security threats before they can cause significant damage.

Regular auditing involves systematically reviewing and verifying logs and activities to ensure compliance with security policies and regulatory requirements. Auditing helps in identifying discrepancies, detecting potential security breaches, and ensuring that all operations align with established guidelines.

Implementing a Comprehensive Logging Mechanism

Log every activity! Why does it matter? Logging every activity within the system provides a detailed record of user actions, system events, and application transactions. This data is important for troubleshooting, forensic analysis, and compliance reporting. Enable logging for all components, including user logins, data access, configuration changes, and API calls. Ensure that logs capture sufficient detail to provide context for each event, such as timestamps, user IDs, and IP addresses.

Implement security measures to protect the centralized logging infrastructure from tampering, such as access controls and

encryption. Regularly scheduled audits help ensure ongoing compliance with security policies and regulatory requirements. They also help in identifying and mitigating potential security risks. Define a regular audit schedule (e.g., monthly, quarterly) based on the organization's risk profile and regulatory requirements. Develop comprehensive audit checklists to guide the review process, covering key areas such as access controls, data integrity, and configuration settings.

In addition to scheduled audits, conducting ad hoc audits in response to specific events or incidents helps address immediate concerns and prevents security lapses. Trigger ad hoc audits based on specific triggers, such as unusual user behavior, security alerts, or changes in regulatory requirements. Document the findings and corrective actions taken during ad hoc audits to ensure transparency and accountability.

You can also use cryptographic hashing to verify the integrity of log files. You should establish data retention policies to determine how long log data should be stored, balancing regulatory requirements with storage costs. Taking these steps not only protects the organization but also develops trust with customers and stakeholders.

The Challenges of Incident Response in SaaS

Unlike traditional software that requires installation on local machines, SaaS solutions are hosted on the service provider's infrastructure. This model offers numerous benefits, including scalability, accessibility, and reduced IT overhead. However, it also introduces unique security challenges, particularly in incident response.

Organizations often have limited visibility and control over the underlying infrastructure of SaaS applications. This limitation can hinder their ability to detect and respond to incidents promptly. In a SaaS environment, security responsibilities are shared between the service provider and the customer. Understanding and clearly defining these responsibilities are necessary for effective incident

response.

SaaS applications are frequently integrated with other systems and services, creating complex interdependencies. An incident in one application can have cascading effects on others, complicating the response efforts.

Best Practices for SaaS Incident Response

Let's go over some of the key best practices for SaaS incident response. You should always define and document the incident response responsibilities of both the SaaS provider and the customer. Ensure that there is a clear understanding of who is responsible for detecting, reporting, and mitigating incidents.

As mentioned earlier, you should deploy continuous monitoring tools to gain visibility into the SaaS environment. Solutions such as Splunk can aggregate and analyze logs from various sources to detect anomalies and potential security incidents.

Service outages and security incidents often require rapid response and clear communication. However, they are fundamentally different in their root cause, trust implications, and stakeholder expectations. Conflating the two can create confusion and erode confidence with customers, executives, and regulators.

Create a comprehensive incident response plan that includes

- **Preparation:** Establish roles, responsibilities, and communication channels. Ensure all stakeholders are aware of the plan.
- **Detection and Analysis:** Define procedures for identifying and analyzing incidents. Use automated tools to detect suspicious activities and gather relevant data.
- **Containment, Eradication, and Recovery:** Outline steps to contain the incident, eliminate the root cause, and restore normal operations.
- **Post-Incident Activities:** Conduct post-incident reviews to identify lessons learned and improve the incident response process.

Additional Resources to Develop a Good Incident Response Program

One of the best resources available is NIST Special Publication 800-61, which you can obtain from <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>.

Having a good incident response plan and incident response process will help you minimize loss or theft of information and disruption of services caused by incidents. It will also help you enhance your incident response program by using lessons learned and information obtained during the security incident. Section 2.3 of NIST Special Publication 800-61 describes the incident response policies, plans, and procedures, including information on how to coordinate incidents and interact with outside parties.

The Forum of Incident Response and Security Teams (FIRST) developed the incident response Services Frameworks. The Services Frameworks are comprehensive documents outlining the potential services that computer security incident response teams (CSIRTs) and product security incident response teams (PSIRTs) can offer. You can access them at <https://www.first.org/standards/frameworks>. These frameworks were created by acknowledged experts within the FIRST community, and they incorporate feedback from a wide range of sectors, including national CSIRTs, private sector CSIRTs, PSIRTs, and other relevant stakeholders.

You should also implement good data backup and recovery solutions to protect against data loss and ensure business continuity. Regularly test your backup and recovery procedures to verify their effectiveness.

Training is crucial! Train your incident response team on the specific challenges and procedures related to SaaS environments. Conduct regular drills and simulations to ensure preparedness and identify areas for improvement. Incident response in SaaS environments presents unique challenges that require a tailored approach. By understanding the shared responsibility model, implementing continuous monitoring, and developing a

comprehensive incident response plan, organizations can effectively manage and mitigate the risks associated with SaaS applications.

The Automation of Everything in SaaS to Enhance Security and Efficiency

Automation enables quick mitigation of security-rule violations, ensuring that undesired changes are promptly reverted to maintain the desired configuration. Let's explore the importance of automation in SaaS, how it can be applied, and the role of DevSecOps in integrating security throughout the application lifecycle.

Automation in SaaS environments is essential for several reasons:

- **Speed and Efficiency:** Automation enables rapid response to security incidents and configuration changes, reducing the window of vulnerability.
- **Consistency:** Automated processes ensure that security policies and configurations are applied uniformly across the environment.
- **Scalability:** Automation helps scale security operations as the organization grows, without requiring a proportional increase in manual effort.
- **Cost-Effectiveness:** By reducing the need for manual interventions, automation lowers operational costs and minimizes human error.

Use automation tools like Ansible or Terraform to enforce configuration management policies.

Case Study: Applying Automation for Security and Configuration Management in SaaS

OmarTech Inc. is a mid-sized company specializing in providing cloud-based solutions to several industries. With a portfolio of more than 20 SaaS applications, the company serves thousands of customers globally. Ensuring the security of its applications and maintaining consistent configuration across its cloud environment

was becoming a significant challenge.

Configuration management was handled manually, leading to inconsistencies and increased potential for human error. Frequent configuration drift caused application downtime and degraded performance. The lack of automated security checks resulted in delayed identification of vulnerabilities. Manual patch management processes were slow and often lagged behind emerging threats.

As the company grew, scaling its security and configuration management processes became increasingly difficult. The team struggled to keep up with the demand for rapid deployment and updates.

To address these challenges, OmarTech Inc. decided to implement a comprehensive automation strategy using industry-leading tools and best practices. The company started adopting Infrastructure as Code (IaC) principles to automate the provisioning and management of its cloud infrastructure. Terraform was used for defining and provisioning infrastructure, while Ansible managed configuration tasks.

The company also used systems like Cisco Vulnerability Management (formerly known as Kenna Security) to prioritize vulnerabilities and then implement automated patch management to keep systems up to date. You can find information about Cisco Vulnerability Management at <https://www.cisco.com/site/us/en/products/security/vulnerability-management/index.html>.

CI/CD pipelines were established to automate the deployment process. These pipelines included automated testing, security checks, and configuration validation steps before deploying to production. The benefits included rapid and reliable deployments, improved code quality, and early detection of issues.

OmarTech Inc. used automated security checks combining tools like Snyk, OWASP ZAP, and Splunk. Automated security tools were integrated into the CI/CD pipelines to scan for vulnerabilities in code and applications. Additionally, Splunk was used for real-

time monitoring and alerting on security incidents. This approach enabled proactive identification of vulnerabilities, faster remediation, and enhanced security visibility.

By adopting automation for security and configuration management, the company significantly improved its operational efficiency, enhanced security, and ensured consistent configurations across its SaaS applications.

SaaS Security Management

By now you already know that traditional security measures often fall short in addressing the unique challenges in SaaS applications. Two additional technologies can be used to protect SaaS environments: SaaS security posture management (SSPM) and cloud access security brokers (CASB). Now, let's go over the concepts, importance, and functionalities of SSPM and CASB in securing SaaS environments.

SaaS Security Posture Management (SSPM)

SaaS security posture management refers to a set of practices, tools, and methodologies designed to continuously monitor and manage the security posture of SaaS applications. The goal of SSPM is to ensure that SaaS environments are configured correctly, secure, and compliant with relevant regulations and policies. SSPM solutions provide visibility into the security status of SaaS applications, identify vulnerabilities and misconfigurations, and offer recommendations for remediation.

Organizations often struggle with a lack of visibility and control over these environments. SSPM provides comprehensive insights into the security configurations and usage of SaaS applications, enabling organizations to identify and address potential risks.

Regulatory requirements such as GDPR, HIPAA, and SOC 2 mandate strict controls over data security and privacy. SSPM helps organizations ensure compliance by continuously monitoring SaaS applications for adherence to these regulations and generating audit-ready reports.

Misconfigurations and vulnerabilities in SaaS applications can lead to data breaches and other security incidents. SSPM solutions identify and highlight these issues, allowing organizations to take proactive measures to mitigate risks and protect sensitive data. SSPM solutions automate the process of security monitoring and management, reducing the burden on IT and security teams. Automated alerts and remediation workflows ensure that security issues are addressed promptly.

SSPM solutions continuously monitor SaaS applications for security posture, identifying misconfigurations, vulnerabilities, and compliance issues in real time. These solutions assess the security risks associated with SaaS applications and provide a risk score, helping organizations prioritize remediation efforts based on the severity of the risks.

Cloud Access Security Broker (CASB)

A cloud access security broker is a security policy enforcement point positioned between cloud service consumers and cloud service providers. CASBs help organizations extend their security policies from on-premises infrastructure to the cloud, providing visibility and control over cloud application usage, ensuring data security, compliance, and threat protection.

CASBs provide comprehensive visibility into cloud application usage, including which applications are being accessed, by whom, and from where. This information helps in identifying shadow IT and ensuring that only approved applications are used.

CASBs offer data loss prevention (DLP) capabilities to safeguard data from breaches and unauthorized access, ensuring that data is encrypted and managed according to organizational policies.

CASBs enhance threat protection by detecting and mitigating cloud-specific threats, such as account hijacking, insider threats, and malware. They provide advanced threat detection and response capabilities tailored for cloud environments.

What about Shadow IT discovery? CASBs discover and monitor unauthorized cloud applications being used within the organization, providing visibility into shadow IT and helping enforce the use of sanctioned

applications.

CASBs can also integrate with identity and access management (IAM) solutions to ensure seamless user authentication and authorization across cloud services.

Cisco Cloudlock

Cisco Cloudlock is a cloud-native CASB that provides a comprehensive approach to securing cloud applications. It focuses on user and data security by offering capabilities that are crucial for managing and protecting SaaS environments.

It supports advanced user and entity behavior analytics (UEBA). Cisco Cloudlock uses machine learning algorithms to analyze user and entity behavior across cloud environments. It detects anomalies and suspicious activities indicative of account compromises or malicious insider actions. Cisco Cloudlock aggregates and analyzes activities across various platforms, including SaaS, IaaS, PaaS, and IDaaS, providing a holistic view of user behavior and potential security threats.

Cloudlock continuously monitors cloud user activities, data usage trends, and connected cloud applications to identify potential security risks. It detects suspicious user behavior, such as abnormal login patterns, high volumes of data downloads or deletions, and unexpected spikes in uploads and downloads.

Cloudlock also provides visibility into OAuth-connected apps, assessing the risk associated with these apps and the permissions they request. This capability helps in identifying potential cloud-native threats posed by these connected applications.

When anomalies are detected, Cloudlock enables a range of automated remediation actions, including administrative alerting, end-user notifications, and enforcing step-up authentication through integrations with IDaaS solutions. The solution moves beyond visibility to offer policy-based enforcement capabilities, allowing security teams to revoke risky applications based on their permission sets and risk scores.

In addition, Cloudlock integrates with malware detection and threat emulation services to identify and respond to cloud-resident malware. It provides automated threat response workflows, including alerting, end-user notification, and file quarantine. This solution offers visibility and control over cloud applications connected to corporate systems, ensuring that even off-network cloud app usage is monitored and managed effectively.

Summary

In this chapter, we explored many aspects of security and privacy within SaaS environments. We began by highlighting the importance of safeguarding sensitive data in SaaS platforms, identifying the types of data most at risk and the common threats they faced. In addition, we examined the significance of major certifications like FedRAMP and regulations such as GDPR, HIPAA, and CCPA, discussing their impact on SaaS applications and how compliance influenced the management and operational strategies of SaaS providers.

You learned about the integration of industry best practices and frameworks, including ISO/IEC 27001 and NIST. Additionally, we covered techniques such as data partitioning and tenant isolation, explaining how these methods contributed to robust data security. The concept of Zero Trust was introduced as a critical approach for protecting cloud implementations, alongside strategies for data loss prevention and the mechanisms behind various data encryption methods.

We provided a comprehensive comparison of physical, logical, and application-level isolation models, detailing their uses in securing SaaS applications and evaluating their benefits and limitations in terms of security enhancement. You learned about best practices for effective detection and response strategies for promptly addressing security incidents, alongside robust identity and access management practices, including role-based access control and multifactor authentication. Then we discussed the challenges and best practices for managing access in cloud-based environments, underscoring the importance of regular security audits and continuous improvements to adapt to evolving threats.

Finally, we explored the roles of SSPM and CASB implementations in SaaS environments. Overall, this chapter provided you with guidance for securing SaaS environments, ensuring compliance with regulations, and implementing best practices to protect sensitive data and privacy and to maintain good security posture.

Part II: SaaS Solutions

Chapter 5. Collaboration: Webex Meetings and Messaging

The ability for people to communicate and collaborate with each other has been one of the driving forces in the creation of many innovations and the birth of the Internet itself. Email and Internet Relay Chat (IRC) channels enabled users to communicate in ways that most had never experienced. Suddenly, people from all over the world could send messages to each other “instantly” instead of having to make an expensive phone call or send a letter in the mail.

IRC quickly evolved into other messaging platforms such as ICQ (which is not an acronym; it’s a play on the phrase “I seek you”) and AOL Instant Messenger (AIM). These were some of the earliest platforms that leveraged centralized Internet-hosted services to provide messaging capabilities to users. There are many examples of early online services that allowed users to “chat” with each other; however, they were primarily focused on consumers and the tech-savvy.

In the late 1990s and early 2000s, the open-source community sought an open, federated model that required no centralized services for instant messaging communication, and the Jabber protocol (later renamed Extensible Messaging and Presence Protocol, or XMPP) was born. Various commercial offerings based on XMPP came to market, including an implementation by the company Jabber, Inc. Cisco acquired Jabber, Inc. in 2008, and its product became the foundation for Cisco’s first serious entry into the Instant Messaging and Presence (IM&P) market, but this solution, like many solutions at the time, relied on the customer managing servers in their network to provide IM services to their users.

Like the early on-premises–based messaging solutions in the enterprise, audio conferencing and eventually web-based meetings started primarily as an on-premises service, or a service provided by a large service provider that provided PSTN-based conferencing. In 2003 Cisco acquired Latitude Communications for its MeetingPlace product, one of the first commercially available on-premises audio and web conferencing solutions. This product evolved into other products like MeetingPlace Express, which was one of the first software-only meeting solutions on the market (the full MeetingPlace product relied on dedicated hardware) to transition meetings to voice over IP (VoIP).

While meetings started off primarily as an on-premises solution, voice and video communication over the Internet has been around for a very long time (relatively speaking in Internet time). Cisco entered the VoIP market in the late 1990s and were the pioneers in telepresence at a time when most connections to the Internet did not have sufficient bandwidth to carry high bit-rate video, and codecs, such as MPEG2, did not offer the bandwidth efficiency and quality of modern codecs, such as H.264, H.265, and AV1. These limitations did not stop Webex Communications from introducing one of the first Software-as-a-Service (SaaS) platforms. Founded in 1995, Webex’s platform allowed users to not just meet on an audio phone call but also share content, see a visual list of participants, and even use VoIP to cut the costs of connecting to meetings. Because of the relative unreliability of the Internet at the time, Webex built its own network backbone to carry customer meeting traffic from location to location, helping to increase quality and performance.

Cisco acquired Webex in 2007, and it has morphed into a premier collaboration platform, offering messaging, meetings, calling, and much more, delivered as Software as a Service. Webex is considered to be one of the first applications delivered through a SaaS model. The architecture of the Webex platform looks very different than it did in 2008 and is a great example of how SaaS products evolve and change over time. In fact, very little of the original Webex platform remains today. In addition to the internal innovation, many Cisco acquisitions were made for technologies and other SaaS applications that made their way into the Webex suite, including Acano, Socio, Involvio, Slido, BabbleLabs, Voicea, Accompany, Broadsoft, Worklife, Tropo, Tandberg, and [Collaborate.com](https://collaborate.com).

In this chapter, we will explore the capabilities of Cisco's Webex platform and will focus on the messaging and meetings features. Although calling is also part of the Webex platform, we will cover it in the next chapter because it presents its own unique considerations, but Webex Calling builds on many of the capabilities discussed in this chapter.

Product Capabilities

The Webex suite provides a variety of collaboration capabilities, including the following:

- Text-based persistent 1:1 and group messaging
- Content/document sharing
- Scheduled and ad hoc video-enabled multiparty meetings with content sharing
- Large events/webinars
- Interactive and persistent whiteboarding
- Polling/audience engagement/Q&A
- Meeting recording, transcription, and summarization
- AI Assistants
- 1:1 calling, including PSTN calling
- Asynchronous video (Vidcast)

We will mainly focus on the meeting and messaging-related capabilities of the Webex suite, and in the next chapter, we will dive into the calling capabilities of the Webex suite because calling presents its own unique challenges and requirements for a SaaS solution.

Webex Meetings

The original Webex platform started as an online meetings platform. Webex initially allowed users to connect into an audio conference call either over the PSTN or using VoIP and allowed users to share content (screen or file

sharing). Since then, the Webex platform has added hundreds of features to provide a best-in-class collaboration solution. [Figure 5-1](#) shows a meeting window for a Webex meeting.

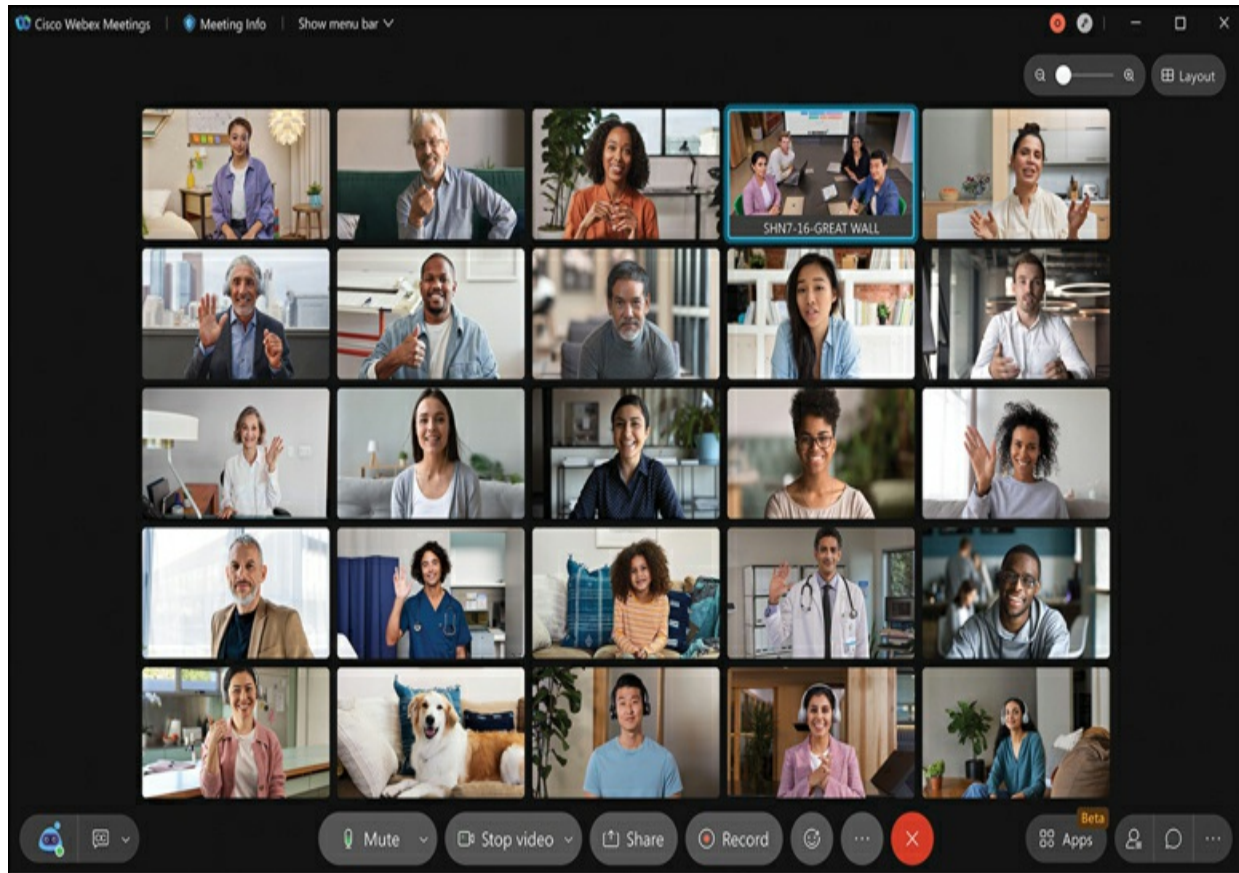


Figure 5-1 A Webex Meeting Window

We will first explore some of the capabilities of the Webex platform that enable users to collaborate and then dig deeper into the services required to create such a platform and how those services fit into the architectural model outlined in [Chapter 2](#), “[SaaS Architecture](#).”

The first major function that the Webex meetings platform provides is allowing users to connect to a multiparty conference to collaborate with audio, video, content sharing, chat, polling, and more—or more generally, meetings. These meetings can be prescheduled or started in an ad hoc fashion.

To schedule a meeting, Webex provides built-in scheduling capabilities; however, many customers make use of integrations with their enterprise

calendaring system so that users can seamlessly schedule meetings from their preferred calendaring application. The two major calendar integrations that the Webex platform supports are Microsoft (both on-premises and cloud-connected) and Google. For both platforms, Webex provides plug-ins that are written for the respective calendar providers to enable the integration.

Ad hoc meetings can be started in a variety of ways. Users are assigned a personal meeting room, and a meeting host can share their personal meeting room information with other users via email, text, or other messaging and instantly start a meeting. An ad hoc meeting can also be started by any member of a Webex space, which instantly notifies all members of the space of the meeting. We will discuss Webex spaces later in this chapter when we talk about messaging. [Figure 5-2](#) shows the Meet button on the upper corner of a space that can be used to start an instant meeting for all participants in that space.

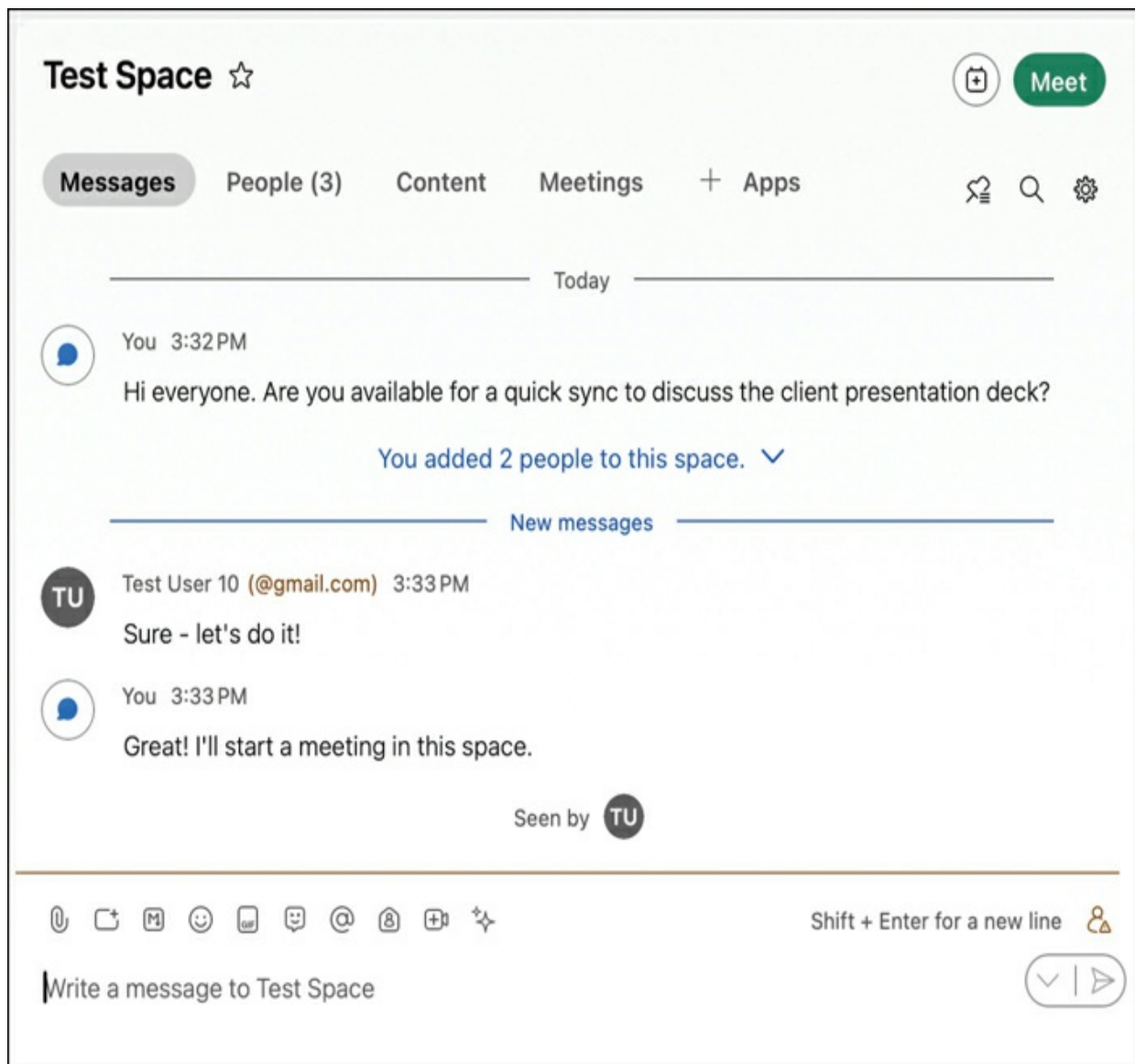


Figure 5-2 Starting an Instant Meeting from a Webex Space

When deciding to join a meeting either as a host or as an attendee, a user has a variety of ways to join the meeting:

- Webex desktop app (macOS, Windows, Linux)
- Webex mobile app (iOS, Android)
- Webex web app (WebRTC)
- Webex Room/Desk devices (RoomOS)

Most users join meetings using either the desktop or mobile app. This is

where the lines between a cloud application and a desktop application start to blur. When a customer subscribes to a SaaS application, it is not uncommon for the service to include access to applications running on a user's device or even to have purpose-built devices like Webex Room and Desk devices that not only interact with the cloud software platform but are actively managed and maintained by the cloud.

If a user is invited to join a meeting and joins from a device that does not have the Webex desktop or mobile app installed, they will be prompted to download and install the application at that time. Once installed, the application is automatically kept up to date by the cloud as new releases are made available. Webex typically releases new software every month, adding both new features and resolving any known defects. Having a cloud-delivered software platform eases the burden of maintaining software releases.

If a user does not have the Webex app installed and chooses not to install it (or perhaps does not have permissions to install it due to security policy enforcement), they can join a meeting using capabilities native to their web browser, provided that the browser supports WebRTC (which all modern browsers support). WebRTC is a complex set of standards that enable real-time communication in a web browser without the need for plug-ins or external applications. The Webex WebRTC client allows users to join and participate in a meeting with just their web browser, but some features are not available when using WebRTC because of the inherent limitations of the browser. For the most feature-rich experience, users should join using the Webex app on a desktop or mobile device.

Regardless of which client a user decides to use to join a meeting, the user must first be identified. They can either be a user with a Webex account, or they can be a guest. A user on the Webex platform has a single identity linked to their email address. Unlike some other meeting and messaging platforms, this identity is global and enables the user to communicate and collaborate with users in any other organization on the Webex platform (barring any administrative restrictions that may be enforced). In this way, it is easy for users in different organizations to collaborate. This ability to have a global identity is one of the powers of a SaaS platform that make federation between tenants easy.

When joining a meeting, the user is asked to authenticate if they have not

previously logged in. A user can be authenticated in a variety of ways. Administrators can use Webex's built-in authentication mechanism, or they can configure the Webex platform to integrate with their own customer-managed identity provider (IdP) to authenticate users. Using a customer IdP allows users to have a single sign-on (SSO) experience. Once Webex has determined the identity of the user, it can look up what roles and licenses are assigned to that user. For example, one user might be licensed as a full-featured meeting host, whereas another has basic meeting capabilities allowing them to participate in meetings hosted by others or to host time-limited meetings.

If the authenticated user is the host of a meeting, they can start the meeting. If they are not the host (or co-host), they can join if the meeting has started but may have to wait in a lobby before being admitted to the meeting.

Whether the user is the host or an attendee, after being authenticated, they are presented with a pre-join window. This window allows a user to select and test their audio and video peripherals, such as their microphone, speaker, and camera. It also allows them to choose whether to enable their microphone or camera prior to joining the meeting. They can also adjust settings such as virtual backgrounds or background noise removal. When they are confident that their settings are configured properly, they can join or start the meeting. [Figure 5-3](#) shows the meeting join window from a Webex desktop application.

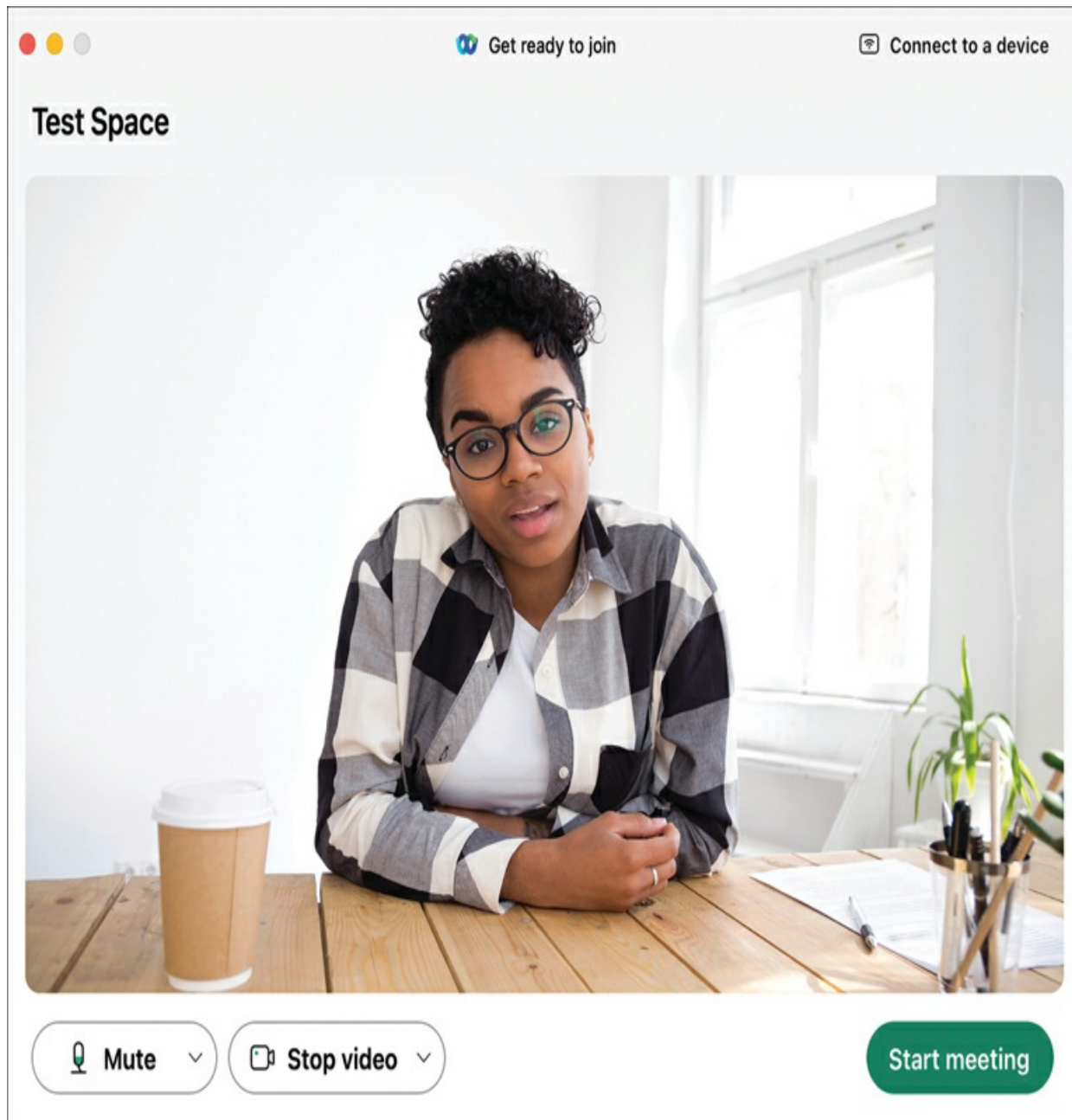


Figure 5-3 The Webex Meeting Join Window

While the join experience is simple for the end user, significant complexity is happening in the background. When we dive deeper into the cloud architecture, we will discuss how Webex media services in the cloud handle media like audio and video. During this phase of the join process, Webex is not only authenticating the user and determining their role in the meeting but also trying to find the nearest media services to use for the meeting. We will go into more detail on this process later.

Once the user has joined the meeting, they are now able to share audio and video and collaborate with other users in the meeting. At the time of this writing, Webex meetings can have up to 1,000 participants, and Webex webinars can scale up to 100,000 participants. Webinars are like meetings but are geared toward sessions that have a few presenters and many participants who are mainly there to watch and listen to the presenters.

Once in the meeting, users have access to a wide variety of ever-expanding in-meeting features. They can manage their layout to get the view of participants they want, share and view content, use interactive whiteboards, chat with other users in the meeting, or enable the polling features delivered by Slido. Audio intelligence allows meeting participants to hear crystal-clear audio free of background noise through the AI noise removal technology enabled by Cisco's acquisition of BabbleLabs. All this functionality is delivered through a hybrid of software running on the user's client and the cloud working in concert, with data being processed either locally or in the cloud, depending on which one provides the best user experience. Chat content during the meeting can be captured for follow-up after the meeting. Advanced audio and video codecs ensure the highest quality experience for audio, video, and presentation sharing.

The meeting can be recorded and summarized through Cisco's AI Assistant. The AI Assistant provides closed captioning in real time through speech-to-text translation in the cloud and can even provide translation into more than 100 languages in real time. During the meeting, the AI Assistant keeps a transcript of the meeting and can summarize the content of the meeting. If you are late for a meeting, it can quickly catch you up on what you missed. Similarly, if you need to step away for a moment, it can tell you what you missed and let you know if you were mentioned while you were gone. After the meeting, the AI Assistant summarizes the meeting and provides insights such as action items. These AI capabilities leverage large language models (LLMs) running on powerful hardware in the cloud.

Meeting users can leverage both Cisco and third-party applications natively inside a meeting. The Webex App Hub (<https://apphub.webex.com/>) lists hundreds of integrations that can enhance the meeting experience from interactive collaboration boards for brainstorming and planning to integrations with medical systems for specialized collaboration in medical

environments. Webex also integrates native applications like Slido for polling and Q&A functionality.

Webex ensures a high-quality voice and video experience with both advanced audio and video codecs as well as media resiliency features that allow meetings to continue functioning in poor-quality network conditions. These features can do only so much in the most severe network conditions, however, so Webex also provides extensive troubleshooting features in Webex Control Hub to give administrators the data they need to locate meetings with poor quality and determine the root cause. Integrations with Cisco ThousandEyes and Cisco Meraki enhance these capabilities. These products will be discussed in more detail later in this chapter.

Webex Messaging

Meetings are just one workload of the Webex platform, with messaging being another major workload on the platform. Cisco went through various iterations of messaging platforms, both on-premises and in the cloud, but the current messaging capabilities of the Webex platform began as Project Squared, which was announced in late 2014; it was rebranded as Cisco Spark in early 2015. Cisco Spark would eventually become Webex Teams in 2018. What was Webex Teams and Webex Meetings would come together as a unified Webex application in 2020. Here, we will focus on the currently existing messaging capabilities that evolved from the original Project Squared. Before Project Squared, XMPP-based messaging capabilities were hosted in the Webex cloud and were known as Webex Connect, but these features have since been deprecated. To make matters more confusing, a new set of features in the Webex Collaboration Platform as a Service (CPaaS) is called Webex Connect, but it has nothing to do with this original messaging platform.

Cisco set out to build a modern, secure, and scalable messaging application with Project Squared. From the beginning, security was of utmost importance. As a result, you will see that what might seem like unnecessary complexity is there to ensure that all communications using Webex are secure and give customers a high degree of control over how messages are encrypted and secured throughout the platform.

As with meetings, users authenticate to log in to the Webex application on desktop, mobile, or web versions of the application. The authentication mechanisms are the same as those for meetings. Once authenticated, assuming the administrator has given the user access to messaging features, the user can collaborate with other users in *spaces* and *teams*. A space can be a 1:1 direct conversation with another Webex user, or it can be a group space with multiple participants. For a brief time, Webex spaces were named *rooms*, but the official term is *space*. In fact, the Webex API for spaces is still called the *Rooms API* so that it does not break backward compatibility with applications using the Rooms API.

Any user can create a group space and add other users to the space. All 1:1 conversations have their own space that is persistent, meaning that messages exchanged between the two users in the 1:1 space will be stored in the cloud and persist through the data retention policy configured by administrators. Group spaces are also persistent but can have up to several thousand users participating in the space.

A user can add users from any organization to a space. This capability enables easy and secure inter-company collaboration. Webex will indicate to the user when a space contains users from other organizations and even allows users to tag spaces with data privacy policies to indicate what types of information can be discussed in a space.

Spaces allow for far more than just text-based messaging. Besides support for rich-text formatting and the ability to mention other users to notify them of a message, for example, Webex messaging capabilities allow users to share content and co-edit documents that have integrations with third-party cloud-based file-hosting services such as Microsoft OneDrive and SharePoint. Users can also choose from a wide variety of third-party apps to add additional functionality to a space. In addition to files, users can also collaborate in real time on whiteboards and share web links that get embedded into the space as tabs for easy access to all in the space. [Figure 5-4](#) shows the Messages tab of the Webex app.

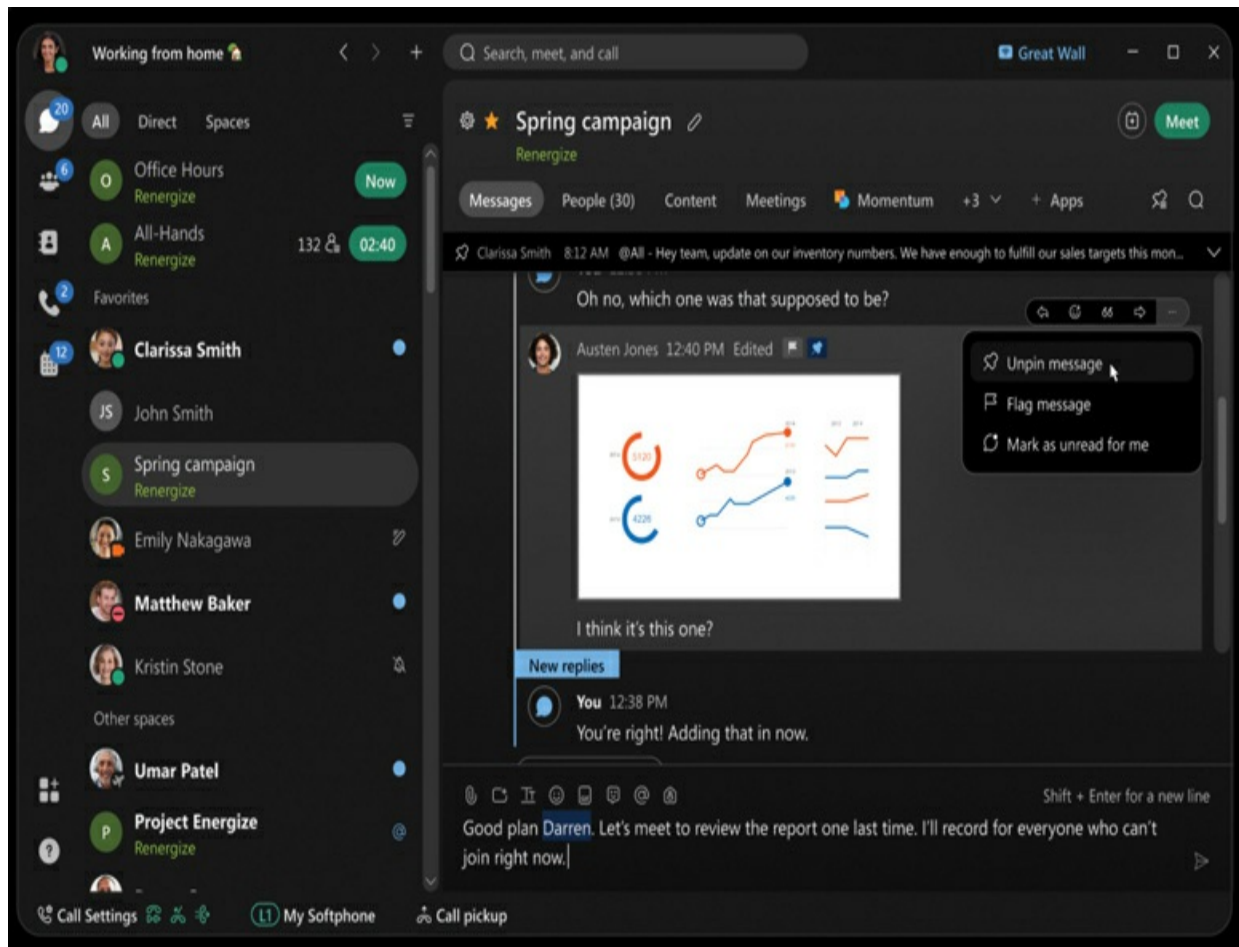


Figure 5-4 Messages Tab of the Webex App

For most spaces, a user can become a member of the space only by creating the space or by having someone who is in the space add them to that space. Users can also choose to make a space into a moderated space, which means that only moderators can add or remove users from the space. Moderators can take things a step further and make a space an announcement-only space, whereby only moderators can post messages to the space. This capability is useful for spaces with large numbers of users that are used to post announcements (hence the name). Spaces can also be tagged as public, which means that anyone in the user's organization can find and join the space without being invited.

In addition to spaces, Webex utilizes the concept of teams. A team is a logical grouping of spaces that generally share a common purpose. Users who are added to a team can see all the spaces that have been added to the team (and can themselves add spaces to the team). Users in the team are not

automatically added to the spaces in the team but can add and remove themselves as needed. Because all the information in a space is persistent, a user who joins a space (whether it be part of a team or not) can always see all the previous discussions and content in that space. Webex allows administrators to configure data retention policies so that data beyond a certain duration is automatically removed from spaces. The default retention policy is 360 days, but Webex Pro Pack customers can extend that further up to 3,600 days.

In addition to interacting with other users, Webex users can interact with *Webex bots*. A Webex bot is a special kind of user that is under the control of a third-party application. Anyone can build a Webex bot by registering it and following the instructions on <https://developer.webex.com>. Bots enable a wide variety of functionality—from ChatOps bots that allow developers to interact with alerting and monitoring platforms to make it easier to collaborate with a team supporting an application or bots that interface with enterprise-specific tooling, making it easier for employees to access important data. Bots can leverage adaptive cards to provide interactive user interfaces inside a space. [Figure 5-5](#) shows the Cisco Ask Licensing bot, which assists customers with managing various aspects of their Cisco licenses.

Ask Licensing ☆



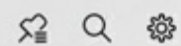
Messages

Profile

Content

Meetings

+ Apps



Ask Licensing 10:04 PM

Hello **Paul Giralt**, Welcome to Ask Licensing, and I am here to help you with licensing questions. Let's get started!



Please select a licensing category you need help with.

Smart Account Administration ^

License Management v

License Delivery v

Device Management v

More Options v

Smart Account Administration covers everything associated with creating, gaining access, organizing, and managing your Smart and / or Virtual Account. Below are the top categories for Smart Account Administration, click More Options button below to see more options (if available). I need help to/with:

- ☐ Request access to an existing Smart Account
- ☐ Manage users in Smart Account
- ☐ Manage Virtual Accounts
- ☐ Smart Account Management issue
- ☐ Manage Smart Account
- ☐ Other - None of these options help me



Submit

End My Session



Shift + Enter for a new line

Write a message to Ask Licensing



Figure 5-5 Ask Licensing Webex Bot

Like bots, Webex also enables a rich developer ecosystem through an extensive library of APIs. These APIs can not only be used for bots and provisioning tasks but can also be used for *integrations*. An integration allows an application to perform tasks on behalf of an individual Webex user. For example, an integration can use Webex APIs to manage users in spaces moderated by the user or schedule meetings on the user's behalf. Applications can also use guest issuer features to embed meeting and messaging functionality into applications for guests to interact with Webex users. For example, a website can provide a medical patient with a link to speak to their doctor. After the patient clicks the link, the website uses the Webex APIs to facilitate a call to the doctor, who is using a Webex application or device.

Asynchronous Video (Vidcast)

Sitting between meetings and messaging is the Cisco Webex Vidcast feature. Vidcast enables asynchronous video collaboration by allowing users to quickly and easily record videos, including audio, video, and screen sharing, and then share that video with users and gather feedback. Vidcast is asynchronous like messaging, but it allows for rich media such as video and presentation sharing, similar to sharing in a meeting. [Figure 5-6](#) shows the Vidcast home page.

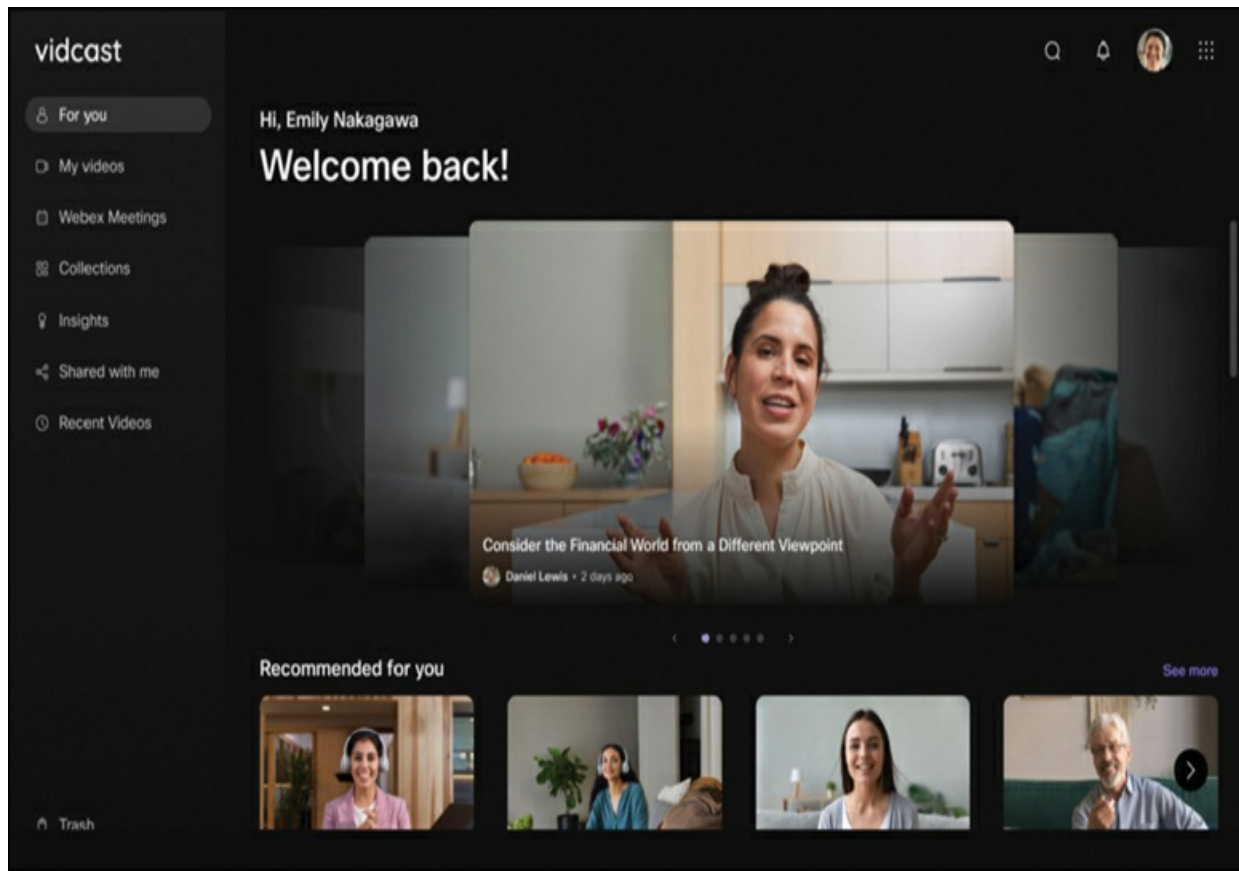


Figure 5-6 Vidcast Home Page

Vidcast is integrated natively into the Webex app but can also be accessed from any web browser at <https://vidcast.io>. It has deep integration with Webex messaging so that you can share messages with people in a team. Using asynchronous video can be a more productive use of time than holding meetings with large numbers of people.

When a user views a video that was recorded in Vidcast (or uploaded to Vidcast), the default setting is to play the video back at 1.2X speed. Most users can easily deal with the slight increase in playback speed and, as a result, end up saving time. The decision to use the 1.2X value again highlights the power of SaaS; the team working on the feature was able to gather data across the entire user base to determine which speed users preferred and then used that data to select the best default value. Traditional on-premises products do not provide the rich usage data that SaaS platforms afford. Additionally, viewers can comment or like parts of the video and give the creator feedback.

Videos that appear in Vidcast can either be created directly from the client PC using the Vidcast application, uploaded from any supported video format, or imported directly from a Webex meeting recording. After the content is uploaded, Vidcast has editing tools that allow the user to trim, cut, or stitch videos to share only the pieces they want to share. Users can even edit collaboratively with others they want to share editing privileges with.

AI features in Vidcast also create transcripts and summaries; plus, they can automatically generate chapters for the different sections of the video. The creator has control over the content generated by AI and can modify it as needed. For example, if the automatically created closed captioning has errors, the creator can modify the captioning to correct it. Vidcast also tracks metrics such as the number of times a video has been watched to give the creator feedback on how much engagement they are getting out of a given video.

The Webex Platform

Now that we have discussed the meetings and messaging features of the Webex suite, let's look at the Webex platform and how it fits into the overall SaaS architectural model introduced back in [Chapter 2](#) and shown again here in [Figure 5-7](#).

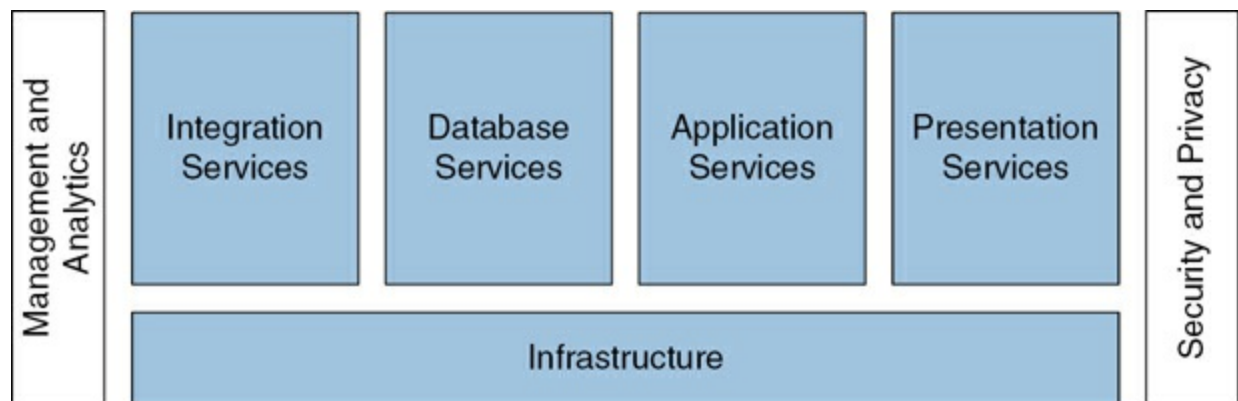


Figure 5-7 SaaS Architectural Model

As with any SaaS application, Webex leverages components that fit into each of the architectural blocks. Let's look at how the Webex platform leverages these different services to provide a reliable, scalable collaboration platform.

Note that the intention of this chapter (and this book) is not to give you detailed information on the inner workings of Cisco's cloud services but rather to give you an overall understanding of the capabilities needed to deliver a collaboration platform such as the Webex suite. After reading through these sections, you should have a good idea of what it takes to build a cloud collaboration service like Webex at a high level.

Infrastructure

As with any SaaS application, the foundation is the infrastructure. The Webex platform leverages a hybrid multicloud architecture that makes use of both private and public clouds along with on-premises edge services to bring cloud-native services directly to the customer premises, such as Webex Edge Video Mesh and other edge services.

As a user of a SaaS platform like Webex, you should, in theory, not have to worry about the details of what is happening in the cloud. You could think of the application as being something as simple as what is shown in [Figure 5-8](#). End users use a variety of devices to connect to the cloud service, and they are able to collaborate. A good SaaS application should abstract all the complexity of what is happening in the cloud and allow you to treat the service as something this simple.

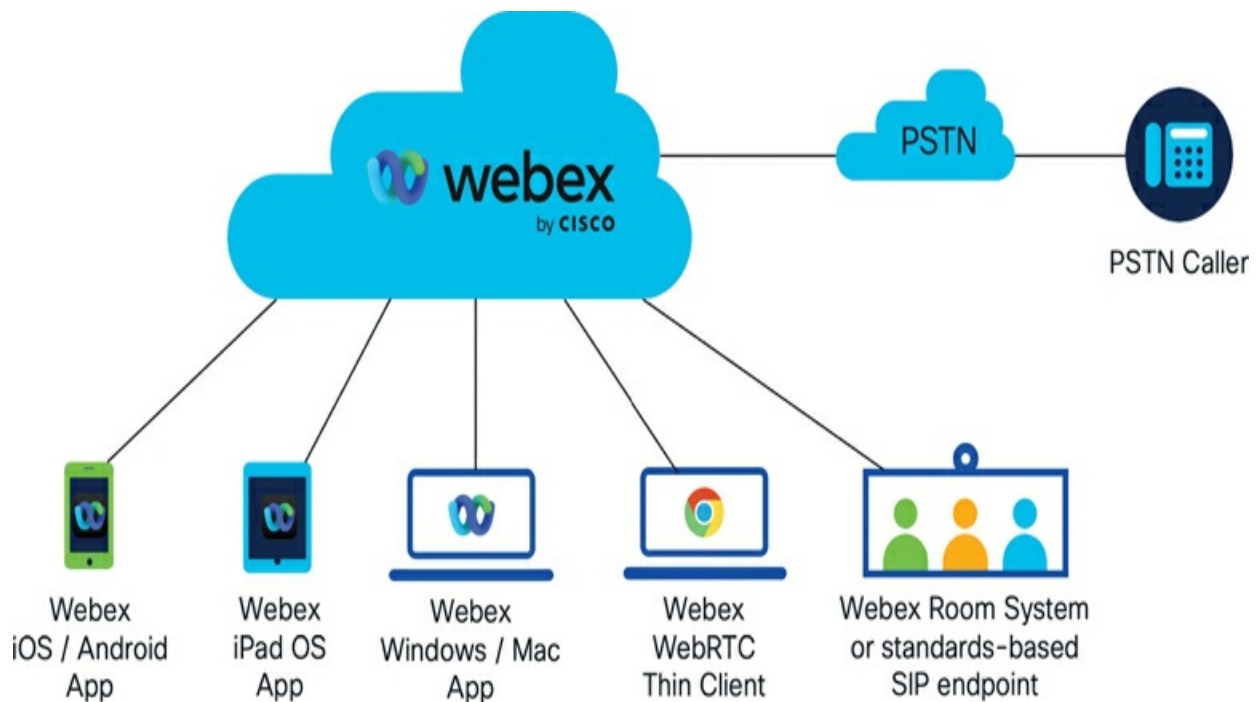


Figure 5-8 Webex Meeting and Messaging Services in the Cloud

While generally true that you, as a customer or end user, should not have to worry about the inner workings of the cloud platform, in some situations, having a deeper understanding of what is happening in the cloud is important for troubleshooting problems, setting up network security policies, and managing traffic. This understanding is especially important for real-time applications where you need to also ensure your connectivity to the cloud meets the standards to provide the optimal level of service.

The services needed to provide meeting and messaging services might look more like what is shown in [Figure 5-9](#). This figure is not intended to show what the actual Webex cloud looks like but rather is just an illustrative example of what kinds of services you might find in a Webex data center or public cloud instance.

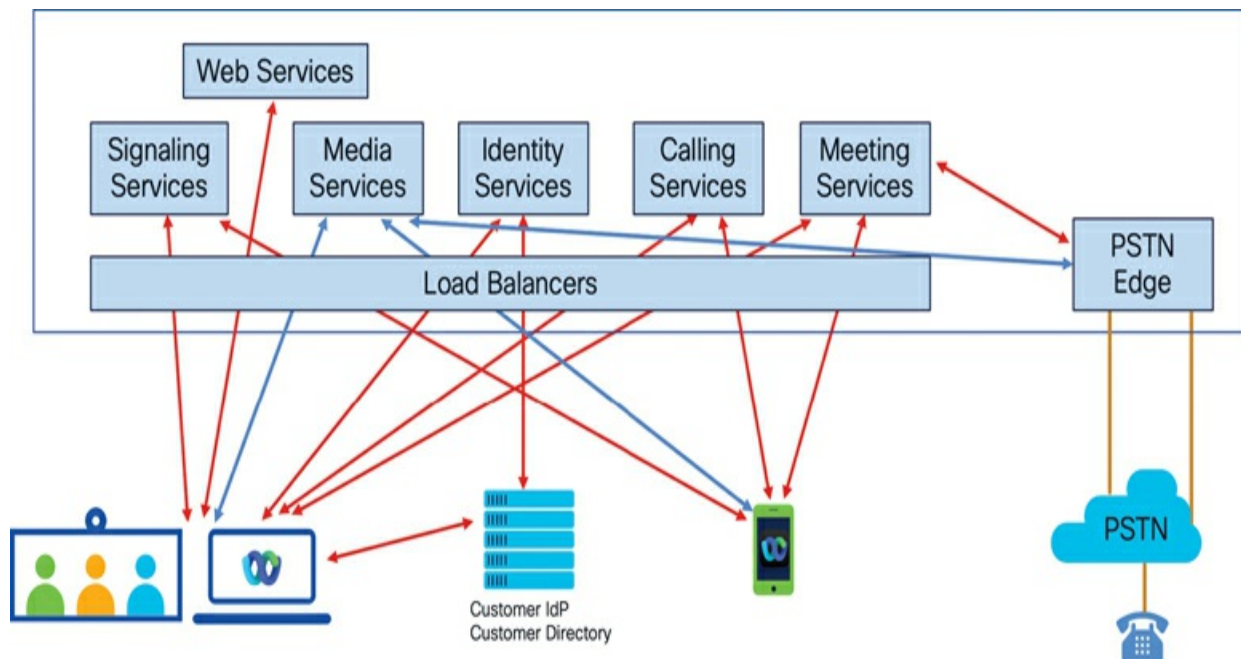


Figure 5-9 A Cloud Conferencing Application Architecture

The Webex platform does use different public cloud providers to host some of the services it provides, but the specific public cloud providers used by the platform are not important to this discussion and could change at any time. What is important to note is that Cisco has the flexibility to deploy and move workloads between cloud providers to meet performance, scale, and resilience objectives.

As with many modern SaaS applications, the platform makes heavy use of containerized microservices that are deployed using continuous integration/continuous deployment (CI/CD) pipelines. This means that new functionality or bug fixes can be deployed into the production environment on a regular basis and undergo routine, automated testing before going into production. Although some Webex services do operate in the public cloud, a significant number of services are also hosted in private clouds built in Cisco-managed data centers around the world.

Many factors go into deciding where to place a given microservice, and some of these factors are specific to real-time collaboration applications like Webex. For example, real-time media such as voice and video require low latency and packet loss.

The Webex meetings platform differs from many other SaaS applications in

that it facilitates real-time communication that can be highly sensitive to network impairments. If a user is watching a movie from a video streaming service, that service can deal with network impairments by buffering several seconds of data before displaying it to the user. This buffering gives the application time to deal with any packet loss by requesting retransmissions or adapting to high or variable latency in the connection path. Real-time communications do not have this luxury. They require low-latency, low-loss connectivity to ensure the best experience.

Imagine speaking in a conversation where the person you are speaking to doesn't hear what you say for several seconds after you have said it. This is what would happen if meeting services like Webex were to add significant buffering to deal with packet loss and jitter, leading to a poor user experience. There are various ways to deal with these stringent requirements, and they begin at the infrastructure layer by ensuring that network and compute infrastructure is available as close as possible to end users and has sufficient bandwidth to users to avoid network congestion and latency. Other features in the application layer such as adaptive codecs and media resiliency features discussed later can also help compensate for any potential issues at the infrastructure layer, but a robust, reliable infrastructure serves as a foundation on which the application services are built.

To help ensure a high-quality experience, the media services that require low latency and jitter might be deployed in a private cloud where Cisco has total control over the infrastructure, allowing it to be interconnected to other private data centers through a private network, thereby ensuring that traffic between media services has the highest level of quality that may not be achievable when traversing the public Internet. In contrast, a web server hosting a configuration web page or a service providing messaging services could be hosted somewhere that does not have the same level of latency and jitter guarantees.

In some cases, hosting a service closer to the user in a public cloud might provide a better experience because private cloud services are not available near the user's location. Elasticity and proximity to end users can affect decisions to put services in a public cloud that might be able to dynamically grow quicker or provide lower latency to a user. Deciding where to deploy services is an ongoing process of optimizing the service that any cloud-based

collaboration platform needs to balance to account for cost, resiliency, performance, and data residency or other security requirements (for example, hosting services in a FedRAMP-approved data center for U.S. government customers).

To enhance the experience even further, customers may choose to deploy one or more Webex Edge Video Mesh nodes in their network. When we discuss media services in the next section, just remember that these media services can be deployed not only in a public or private cloud, but also as a service running on the customer's premises. In this case, the infrastructure becomes the responsibility of the customer because they are hosting the service.

Application Services

The Webex platform includes a variety of application services, each responsible for a particular aspect of enabling real-time collaboration for meetings, messaging, calling, and more. The following service categories in the cloud enable messaging and meeting services on the Webex platform:

- Authentication and authorization
- Messaging conversations
- Message attachments and document transcoding
- Messaging indexing and search
- Meeting scheduling
- Meeting management and control
- Media management and control
- Media termination and switching
- Media mixing and transcoding
- Speech-to-text and translation
- Meeting summarization and other AI features
- Subscription management and billing
- Web services

- Encryption and key management
- Recording
- Management and administration
- Configuration management
- Calendar integration
- Directory integration
- Metrics and logging
- Notifications
- API gateways
- Device management

This list is by no means exhaustive, and most of these services are further subdivided into smaller component microservices that work together to provide the higher-level service. For example, media termination and switching are handled by a suite of services that work together to provide that capability. As another example, meeting management and control may have services to handle screen sharing, whiteboarding, managing participant lists, and managing WebRTC sessions.

We will not cover each of these services in detail but will focus on some of the ones that are most important to enable meetings and messaging on the Webex platform. Additional services for enabling Calling capabilities are discussed in the next chapter.

Authentication and Authorization

The authentication and authorization infrastructure in the Webex cloud relies heavily on the OAuth 2 standard, which was discussed in [Chapter 4](#), “[Security and Privacy for SaaS](#).” Services in the Webex cloud rely on OAuth bearer tokens to authenticate against each other. This use of OAuth extends beyond services in the cloud to clients as well, which also use OAuth tokens for authentication purposes.

The tokens granted to clients and services are limited to specific scopes. A scope defines what the bearer of the token is allowed to do. For example, a

service in the cloud might have a token that gives it permission to read the settings for all organizations in the Webex cloud, but a token granted to an administrator might be able to read and write settings only for the organization that this person is an administrator for. Similarly, an end user's token might be allowed to read certain settings for the organization they are a part of to allow the organization's clients to know how they should be configured based on organization settings. OAuth scopes can be very granular.

Many customers want to use their own identity provider to authenticate their users. In this way, users can leverage single-sign-on with whatever provider the company uses. For example, a company might want to use Cisco Duo for authenticating against its Microsoft Entra ID directory. To enable these integrations, Webex can use Security Assertion Markup Language (SAML) to integrate with the customer's IdP. More modern integrations rely on OpenID Connect (OIDC), as described in [Chapter 4](#).

Regardless of which identity provider is used to authenticate a user or service, in the end, the user ends up with an OAuth bearer token that is used to authenticate against services. The services themselves must also authenticate against other services, but services are not users, so how does that work? Most services have *machine accounts*, which are internal accounts assigned to a service that grant them specific permissions. The credentials for these machine accounts must be stored securely and are typically rotated often. The concept of least privilege is a common practice, whereby services have only the permissions they need to perform their function and no more. This ensures that in the unlikely event that a service is compromised, there are limits as to what damage an attacker can do with the compromised service.

At a higher level, the Webex platform must maintain a database of authorized users and settings associated with those users. The Webex common identity platform stitches together the authentication and authorization components as well as the storage of identity information. These services interface with provisioning services to maintain information like a user's name or email address and licensing information.

Encryption and Key Management

A critical component to the operation and security of the Webex platform is the key management services that enable secure, encrypted communication for both data in transit and data at rest.

Figure 5-10 shows a high-level view of the key management service (KMS) and some of the services with which it interacts.

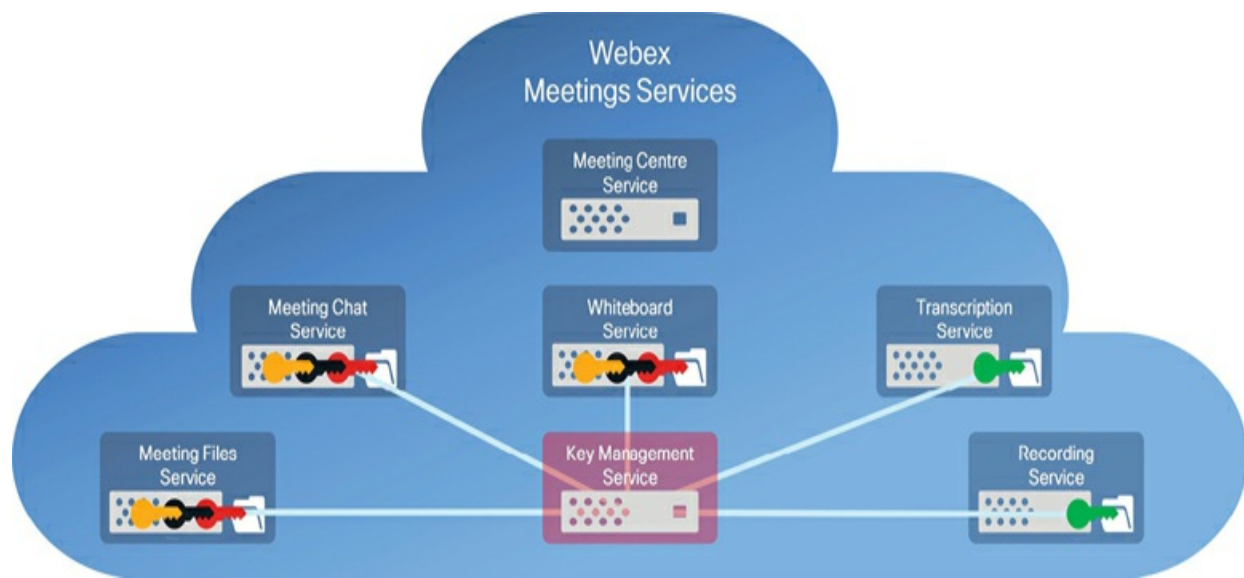


Figure 5-10 Key Management Service Interactions

All data in the Webex cloud is encrypted, and the KMS manages all the keys to encrypt and decrypt the data. Each organization's keys are isolated from other organizations, and access to keys is strictly controlled through role-based access. Logically, you can think of it as each org having its own KMS. Keys are used throughout the Webex platform. For example, each Webex meeting is assigned its own encryption key that is shared as needed with meeting participants. When a user is added to a Webex space, the user is granted permission to retrieve the encryption key or keys needed to decrypt messages in that space. These are just a couple of examples of the many scenarios where KMS is involved in granting keys to users or services.

Certain events can cause encryption keys to be rotated or revoked, and this process is handled by the KMS. One key feature that is enabled by KMS is the Webex Zero-Trust Security for Meetings feature. With this feature enabled for a meeting, all media for a meeting is encrypted in such a way that the cloud has no way to decrypt it for any reason. This means that features

like network-based recording or PSTN connectivity are unavailable for these kinds of meetings because these features require that services in the cloud be able to decrypt the media.

Because each organization manages its own keys, what happens if two users from different organizations want to communicate or two users from different organizations are in the same Webex space? In this case, users might need to retrieve keys from the KMS owned by another organization. This seamless, secure sharing of keys is what enables users across organizations to easily collaborate while still giving administrators control over who and how their own users are allowed to communicate with other organizations.

When the Webex app needs a key, it requests the key from its KMS. Each key has a URL that describes which KMS the key belongs to. If the URL indicates that it is stored on another KMS, the user's KMS retrieves the key from the KMS that owns the key on behalf of the user, so users only ever communicate with their own KMS. Each key has an associated access control list (ACL) that identifies the users that are allowed to access the key. Before granting access to a key, the KMS storing the key verifies that the requesting user is on the ACL. If it receives a request from another KMS, it also verifies that the requesting KMS is authorized to access the key.

As an example, assume Alice and Bob, in different organizations, attempt to send a message to each other. [Figure 5-11](#) shows how keys are exchanged to facilitate this conversation.

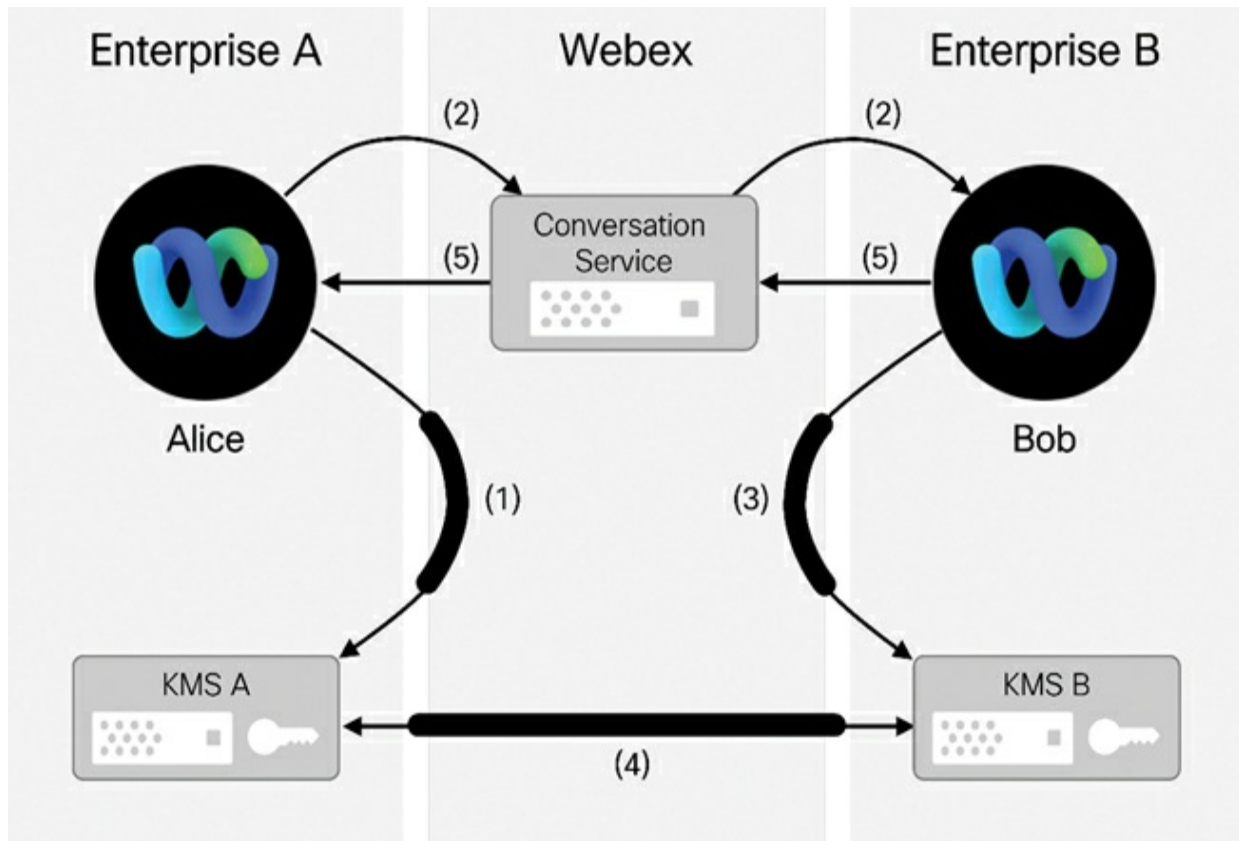


Figure 5-11 Key Management for Conversation Between Users in Different Organizations

When Alice creates a conversation with Bob, Alice's Webex app gets a key for the conversation from KMS A. This app also notifies KMS A that Bob is authorized (1). When Alice's Webex app creates the conversation with the Webex Messaging service, it also provides the key URL for the conversation, which the conversation service relays to Bob's Webex app (2). When Bob's Webex app joins the conversation, it requests the key from the KMS for Bob's enterprise (KMS B) (3). Bob's KMS sees that the key is stored on KMS A and forwards the request (4). KMS A checks that Bob's Webex app is authorized to receive the requested key and that KMS B is authorized to represent Bob. If these checks pass, KMS A provides the key to KMS B, which in turn provides it to Bob's Webex app. Bob's Webex app then uses the key to encrypt a message for Alice and safely send it to the conversation service in Webex (5), which will then store it and forward it to Alice's Webex app when it comes online (and likewise for any other participants in the space). Since Alice's Webex app has the same key, it can decrypt the message and display it.

For customers that require it, Webex allows customers to manage their own keys and the KMS itself in their own infrastructure. This feature is called bring your own key (BYOK). When a customer enables this feature, they now become solely responsible for maintaining the encryption keys used by KMS. This means that if the customer loses the keys for whatever reason, there is no way that Cisco can retrieve any of the data in their org, such as messages, meeting recordings, and more. Most customers choose to let Cisco manage the keys in the cloud, but this option is available for customers who want the ultimate level of control over their keys and are willing to take the responsibility of protecting those keys.

Meeting Management and Control/Media Services

A real-time collaboration application generally needs to deal with two broad categories of data: signaling and media. Signaling is communication between clients and application services or between two or more services that indicates a request for an action, a response to a request, or a notification of an event. Often, the purpose of the signaling traffic is to establish a media channel between two or more devices. Media traffic transports some type of real-time communication such as audio, video, or a screen share. Sometimes the lines between signaling and media blur; for example, dual-tone multi-frequency (DTMF) relay is often carried as part of a media stream, but we could argue that this is signaling more than it is media. Another example is Session Traversal Utilities for NAT (STUN) packets that are carried as part of the media stream and are used to signal a request for information about the other endpoint involved in the media stream.

Services within a collaboration application like Webex that are used to enable a collaboration session are divided into signaling services and media services. This is not to say that there are no other types of services such as security, monitoring, and management services, for example, but the services whose primary purpose is providing the collaboration service can generally be divided into these two categories. The media services will have signaling components that are used to communicate with the signaling services. The majority of signaling services within the Webex meetings and messaging services use HTTPS to transport REST API calls as the basis of their signaling. Some services use the Session Initiation Protocol (SIP) or H.323 to

communicate with external services such as third-party collaboration devices. SIP will become more important in the next chapter when we discuss Webex Calling.

As mentioned earlier, when deciding to join a meeting, a user must first authenticate and indicate the meeting they wish to join. A meeting could be either a scheduled or ad hoc meeting. If scheduled, the meeting could be synchronized from an external calendar service. There are services dedicated to managing scheduled meetings and synchronization of meeting scheduling data.

To join a user to the meeting, the Webex client uses a variety of signaling messages to establish the media connection. First, the Webex client determines the reachability of media servers. The Webex client is provided with a list of available media services through signaling messages, and the client sends STUN binding requests to the different media servers through the media channel, trying to determine which ones are reachable and closest to the client by measuring round-trip delay times. Once the client has determined the media server with the best reachability, it communicates with the signaling services in the cloud to set up the media connection to that server. At a very high level, [Figure 5-12](#) shows how this process occurs.

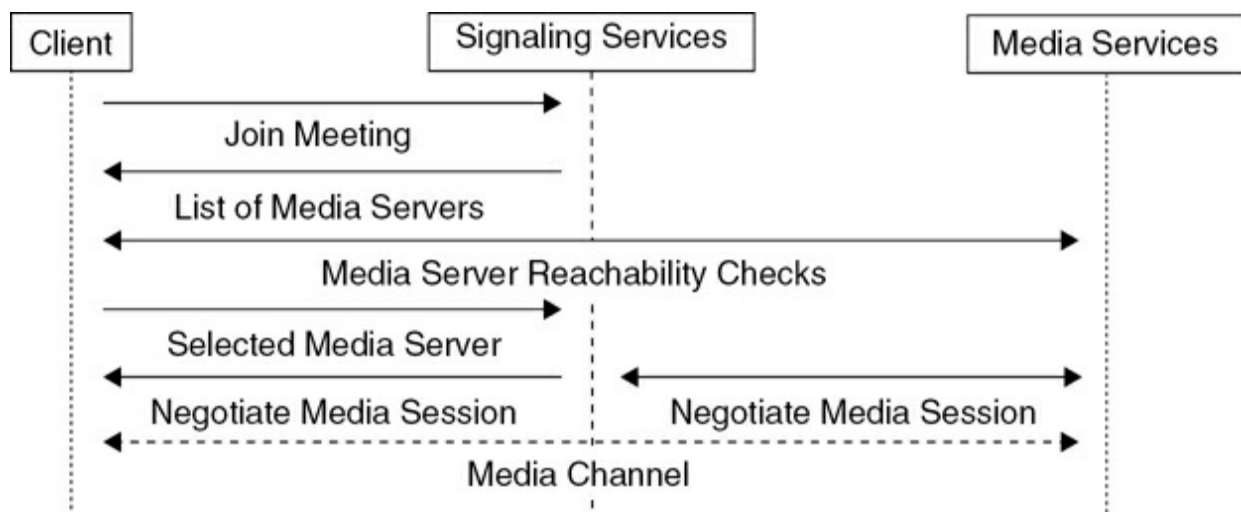


Figure 5-12 Relationship Between Client, Signaling, and Media Services

This figure shows a simplified view of how setting up a connection to a meeting works. In reality, many different signaling services are involved to perform tasks such as load-balancing traffic across different media services

and tying the signaling of this one client with the other participants in the meeting. In some cases, the signaling services may determine that other services like transcoding are required to devices with incompatible media to communicate with each other.

Earlier in this chapter we mentioned Webex Edge Video Mesh nodes. These are virtual machines (VMs) that a customer can install on their premises to allow media to terminate within their corporate network instead of going to the cloud. These Video Mesh nodes basically run some of the same signaling and media services that run in the cloud. When a client is provided the list of media servers, as shown in [Figure 5-12](#), this list includes any Video Mesh nodes in the customer's organization. If these nodes happen to return the best reachability information, they are chosen for the meeting.

In general, Webex meetings primarily leverage media switching technology as opposed to media transcoding. This means that when a media stream—say an audio or video stream—is sent from a client to the cloud, the cloud does not decode that media stream, but rather just sends that stream to participants who have requested to receive that stream. For audio streams, this decision is relatively straightforward: The cloud will always send clients in a meeting the audio from the three loudest speakers at any point in time. For video, it is far more complicated and depends on the device being used, the number of participants in the meeting, and the layout the user has selected on their client. It can also depend on the amount of available bandwidth detected. In a few limited scenarios, the cloud will transcode video or audio from one format to another, but as a general rule, media streams are switched as described in the following paragraphs.

Webex clients can send up to three video streams at different resolutions but can receive many more, depending on the layout and number of participants. [Figure 5-13](#) shows the media streams for four of the participants in a meeting with 16 participants.

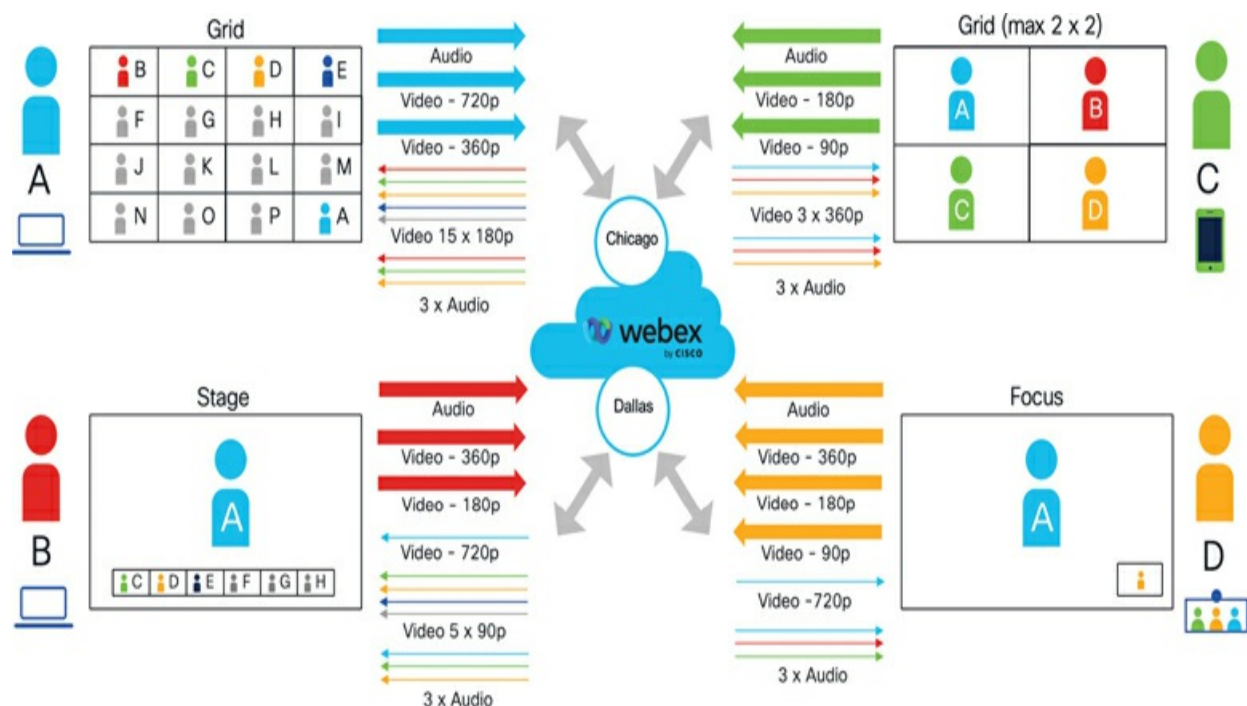


Figure 5-13 Media Stream Switching in a Webex Meeting

In [Figure 5-13](#), the four participants are labeled A, B, C, and D. User A is on a laptop and has a Webex client configured for a grid layout, allowing them to see 15 other users and themselves. User B is also on a laptop but in Stage view, which shows the loudest speaker full sized and then the next six loudest speakers in a “film strip” view. User C is on a mobile device configured with a 2x2 layout, and D is on a video device set to the focus layout, which shows only the loudest speaker and themselves.

Because of these different layouts, sending a high-definition video stream for all 16 participants to all the other participants doesn’t make sense. Similarly, it makes no sense to send the audio stream for all participants either. It doesn’t even make sense for all participants to send high-definition video to the cloud at the same time because there might not be anyone who needs to see that stream at a given time. When a client knows it wants to display the video for a given user, it asks the cloud for the appropriate resolution for that stream. This is the resolution that is large enough to provide good quality for the size of the window displayed on the screen without asking for a resolution that would go to waste. In the example, User A’s client has requested a 180p resolution stream for 15 participants while Users B and D have requested a 720p high-definition stream for User A because User A happens to be the

loudest speaker at this point.

You can also see that Users A and C are connecting to the Chicago Webex data center, whereas Users B and D are connecting to Dallas. The media servers in these two data centers will “cascade” the media sessions between each other so that users connecting to any data center can communicate with users connected to other data centers seamlessly. These cascade media streams are typically transported across a private connection to ensure the highest quality. Webex Edge Video Mesh nodes can also serve the same function as these data centers and can cascade streams to the cloud to enable seamless interoperability.

Other services can be introduced into a meeting to facilitate additional features. For example, a call recording service can be brought into the meeting to record all the streams in the meeting. Translation or closed-captioning services can be brought in to convert the audio streams to text and optionally translated into different languages.

An additional component used extensively in services such as Webex is a message or event bus. As mentioned in [Chapter 2](#), an event bus is useful when one microservice wants to share information with other services. For example, if a user presses the mute button on their client, all the signaling services for other users in that meeting should be notified so that they can notify their respective clients of the mute event. This would then cause the mute icon to show up next to the participant’s name. While a seemingly trivial task, being able to do this at scale across millions of meetings a day requires something like an event bus.

Another use for an event bus is for one service to produce an event without knowing who’s going to consume it. As new use cases arise, new consumers can easily be added to the application without having to modify the producer. For example, Webex clients produce media quality metrics every minute. This kind of data can be sent on an event bus like Kafka, and then various services could consume it. Perhaps a metrics service consumes it for logging and monitoring. Another service could consume it to store in a troubleshooting database so that administrators can view quality metrics. Yet another service could aggregate and summarize the data to include in a voice quality report. The capability to disaggregate producers and consumers of data in a cloud service leads to faster time to market for new features and

allows development teams to work in a way that is more loosely coupled.

Messaging services make extensive use of event buses. For example, if a user types a message into a Webex space with 1,000 users, those 1,000 users must be notified of the incoming message. The same applies for other events such as being added to a space, indicating that someone has started or stopped typing, or notifying that a user has written a stroke on a whiteboard. A message bus allows these services to be distributed for scale and resiliency.

Now, let's dig a bit deeper into how the signaling connections between clients like a Webex application or web browser and the cloud services are established. Most clients on the public Internet sit behind some kind of firewall and/or router performing Network Address Translation (NAT) and usually have a dynamic IP address (or multiple dynamic IP addresses) that can change at any time. Services in the cloud, on the other hand, generally have public IP addresses that do not change; however, services in the cloud can come and go based on how the cloud is being managed.

Clients need to know how to reach a service, and those services need a way to be able to communicate back to the client as needed. For example, if a client is sitting idle and another user sends a message to that client, the signaling services need a way to notify the client of this event. If the client's address is constantly changing and likely cannot be reached directly from the Internet due to firewall rules, how does the cloud service communicate back with the client? For the most part, Webex leverages HTTPS WebSocket connections to create a bidirectional signaling channel and re-establishes that channel if network conditions indicate a change, such as an IP address or network connection change. We briefly discussed WebSockets in [Chapter 2](#). [Figure 5-14](#) shows at a high level how a WebSocket connection is established.

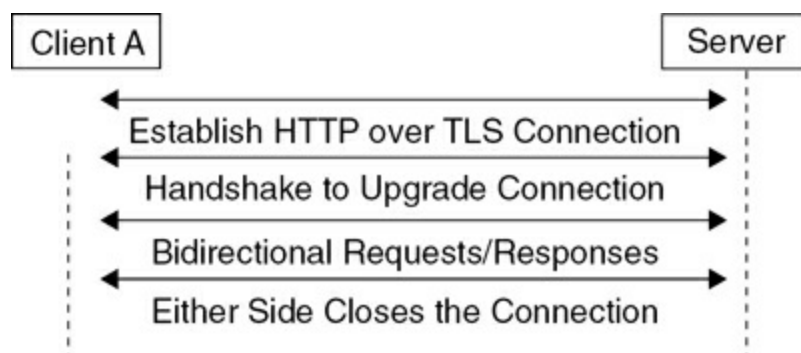


Figure 5-14 Establishing a WebSocket Connection

The important part to note here is that the client reaches out to the cloud service through a well-known DNS address—for example, `meetingservice.webex.com` (this is not the actual name of the service but is an example of what it could be). The DNS entry resolves to an IP address that is typically configured on some kind of network load-balancing device. The DNS resolution itself can also be dynamic and respond with different addresses of different load-balancers based on factors such as geolocation of the source IP address or due to some redundancy or load-balancing configuration (yes, load-balancing between load-balancers). The key here is that the connection is outbound from the client to the server. The load-balancer then finds the appropriate microservice in the cloud to route the request to. The WebSocket connection is then persistent, allowing either side to communicate with each other as long as the WebSocket connection is up.

This kind of bidirectional connectivity can be a problem for mobile devices. Most mobile operating systems do not allow applications to maintain a persistent WebSocket connection when the application is in the background. These restrictions are in place largely to improve battery life on the devices. Because of these limitations, cloud services like Webex must also make heavy use of push notifications. Both Apple and Google have push notification services for iOS and Android, respectively. These services allow an application like Webex to indirectly send a message to a user's device via the push notification service. [Figure 5-15](#) shows how a push notification can be used to inform a mobile user (Client B) that another user (Client A) has sent a message.

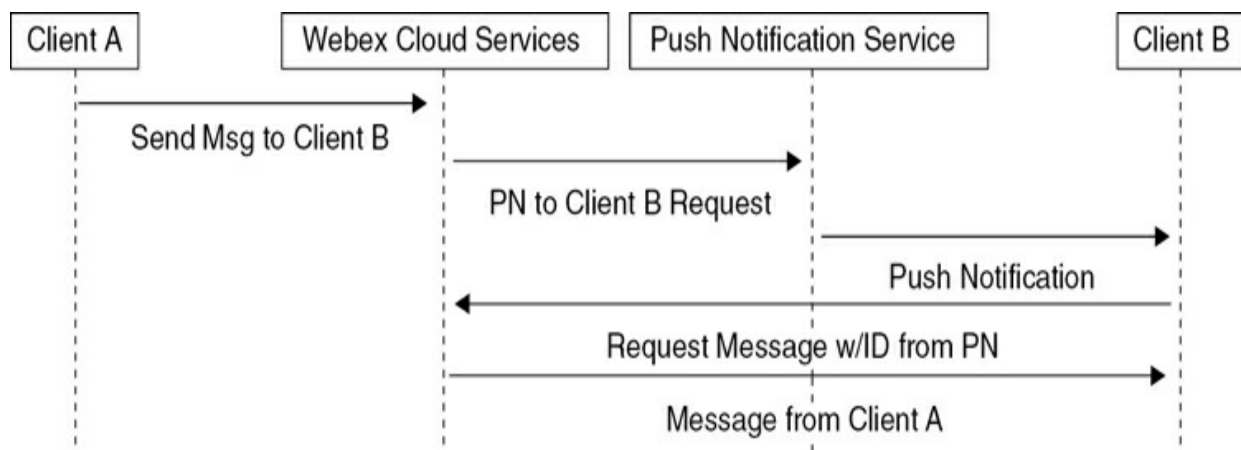


Figure 5-15 Using Push Notification to Deliver a Message

As shown in [Figure 5-15](#), when Client A sends a message to Client B and the user of Client B is not actively using their Webex app, the Webex cloud needs a mechanism to inform the user of the message. The Webex cloud knows that it does not have an active WebSocket connection to the client, so it sends a message to Apple or Google (depending on what kind of mobile device is being used) requesting that a push notification be sent to Client B indicating an incoming message. The push notification is limited in what data it can carry, but it has enough information so that when Client B clicks on the notification, the Webex app opens, reconnects to the cloud, and then asks for the details of the message. This same mechanism is used for other things like a new meeting notification or call notifications. In the background, Apple and Google maintain something like a WebSocket connection that they manage between their cloud and the mobile device, but by having one channel for notifications instead of each app having its own background connection, battery life and performance are significantly improved.

Messaging Services

Messaging services largely serve as a conduit between the messaging database layer that stores messages and the APIs it exposes to clients to be able to write, read, and modify those messages. The messaging services keep track of spaces, teams, conversations, threads, and more. They also interface with the encryption and key management services to encrypt messages, store them, and decrypt them as needed by retrieving the appropriate keys.

Along with text-based messaging, the messaging services must also provide the capability to store file-based content. Documents are generally not stored in a database, but rather in some kind of data store. The Webex services that provide for file storage abstract the back-end data store by allowing a customer to use different data stores. For example, customers can integrate with Microsoft 365 and store all content in a space on a content store managed by Microsoft. Webex can also natively store content in its own data stores. The messaging services hide much of this complexity from the front-end clients that expose these features.

One important requirement of any messaging service is the ability to search

through that message data to find the information a user is looking for. How do you build an application where all the data is encrypted and then provide a way to search through that encrypted data? Webex makes use of encrypted search indexes using proprietary methods that allow a customer to control their own encryption keys, and the encrypted search services index the data in such a way that the Webex search services do not have to decrypt the data to perform a search. Clients retrieve keys from KMS to decrypt the search results.

Figure 5-16 shows how the KMS, indexing, and compliance services can be located either in the Webex cloud or entirely managed by a customer in their own environment.

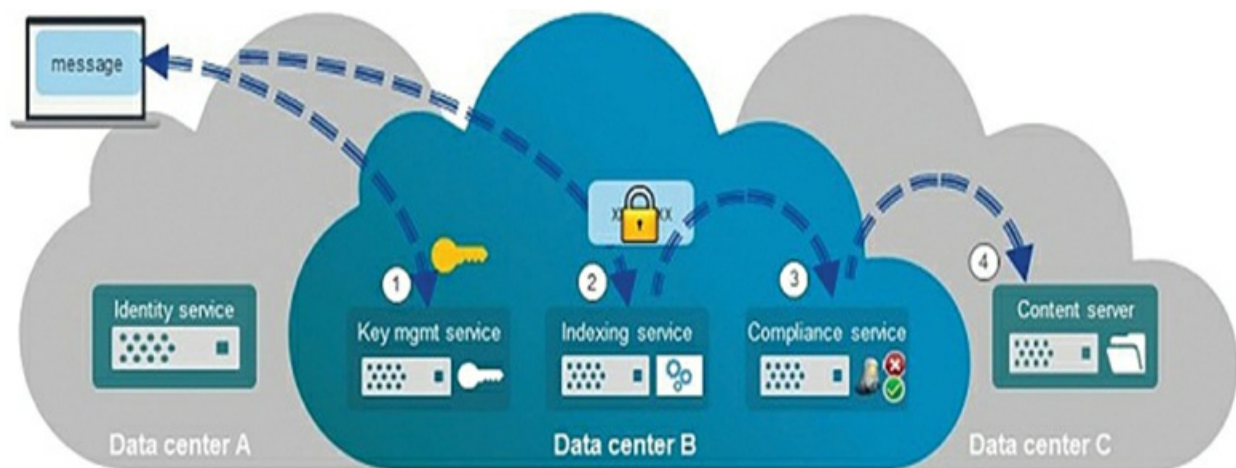


Figure 5-16 Security Domain for KMS, Indexing, and Compliance Services

Data center B, shown in Figure 5-16, can be a customer's data center if they are managing their own KMS or could be a Cisco- or customer-hosted public or private cloud. The services all work the same way, but the distinction is who is managing the services used for key management, indexing, and compliance.

Compliance services also generally fall into the area of messaging services because many customers have specific compliance and data loss prevention (DLP) requirements. The compliance services interface with the other messaging services to check a file for malware before allowing a user to download the file, for example, or checking messages for personally identifiable information (PII) like Social Security numbers or credit card numbers and removing them from messages before allowing a user to see

them. One such integration that Cisco supports for DLP is with Cisco Cloudlock.

Recording, Speech, and AI Services

Webex provides an evolving set of capabilities that take content generated on the platform and process that data for later retrieval or insights. One of the key features of any SaaS meeting platform is the capability to record meetings. Webex not only records meetings but also provides intelligence and AI features during and after the meetings. Various services are responsible for these capabilities.

During a live meeting, transcription and translation services receive the audio stream for participants in a call just like any other client in the meeting. They take the audio stream and process it through a speech-to-text engine to obtain text of what is being said in the meeting in real time. This text can then be used to provide real-time closed captions in the meeting or can be passed to an AI translation service that translates the text to one of more than 100 languages so that participants can see closed captions in their language of choice.

Services like speech to text and transcription require low latency and high processing power to be able to display the text as close to real time as possible. Because of the complexity of language, sometimes there is a delay in translating text because sentence structure and phrasing are different in different languages. As a result, sometimes the translation engine needs to wait for the whole sentence to be provided before it can create an accurate translation.

In addition to the real-time features, a recording service also receives a copy of the media streams for the meeting so that it can save them for later viewing. This service must store the audio, video, and content sharing for the meeting, along with the transcripts from the transcription service.

After a meeting has finished, the meeting summarization services parse through the meeting transcript and attempt to extract important points, action items, and other insights from the meeting content. These services can also categorize sections of the meeting so that the meeting recording is broken up based on the sections and can be synchronized in such a way that a user can

select a part of the transcript and go directly to the recording at that time. Many of these services rely on message busses where they are listening for events and then taking action when a meeting starts or ends, for example, or a recording is started or ended.

Presentation Services

We have spoken extensively about the Webex app on desktop and mobile machines, as well as other clients like WebRTC-based web apps. These applications generally provide the user interface for end users and can be considered to fit into the presentation services block of the architecture.

The Webex platform makes extensive use of RESTful APIs to provide an interface for the presentation services to communicate with all the back-end services needed to provide features and functionality. One advantage of this approach is that the clients providing the presentation services are not tightly coupled with the backend, so two different clients can present very different user interfaces while relying on the same back-end services. For example, a WebRTC application might use HTML, CSS, and JavaScript to create a user interface for checking messages or joining a meeting, whereas a native iOS app might use Swift UI and the Swift programming language to build the UI, but in both cases, the applications will use the same RESTful APIs to retrieve and send messages.

Database Services

The Webex platform must store huge amounts of data; therefore, it provides a variety of database services that meet the requirements of various application services and provides the data needed for management, analytics, and security monitoring, all while taking security and privacy considerations into account.

We will not cover the specific databases used to support Webex services here; however, we will go through the various types of databases needed and the considerations that go into deciding the type of database needed for a particular task. In [Chapter 2](#), we discussed relational and nonrelational databases that can store structured, unstructured, or semi-structured data. You

will see how Webex has the need for all these different database and data types.

Provisioning Database

One obvious place where a database is needed is to store structured data, such as users, licenses, devices, and other provisioning and configuration data. Webex also needs a place to store elements like scheduled meetings and other user data that can change over time.

Organization and user-level settings must be stored securely. Usernames, passwords, and other sensitive credentials must be stored as well. Webex does not have a single database that stores this data, but rather each service makes use of the database that best fits the needs of that service. For example, the service that stores user data might use a relational database, whereas the one that stores settings might use a nonrelational database that offers greater flexibility for the data stored in that database.

Messages Database

Another key data storage requirement is the storage of messaging/conversation data. For example, if a user sends a message in a space, that message must be stored somewhere in a way that is quick and easy to retrieve and can be stored for a long period of time. Time-based retrieval of messages is important because typically a user wants to retrieve all the messages that have happened since a point in time. When retention policies are being enforced, the service responsible for purging messages outside the retention period would want to find all messages that occurred prior to a certain time.

One key decision to make when choosing any database is whether a relational database is required. NoSQL databases generally lend themselves to large datasets, are easy to scale out, and offer high performance for the types of queries needed to offer a messaging service, but they do not necessarily provide the level of structure that a traditional relational database offers. That said, many NoSQL databases still allow for the creation of a schema and having relations between tables, but they usually lack the ability to JOIN between tables or provide for stored procedures. Because messaging features

require such huge volumes of data, a messaging database is well suited to a NoSQL database where structure and relations are present, but the need for things like JOINS is not.

One other commonly used element in cloud services like Webex is implementing a caching layer. These caching layers improve the performance of reading information that is stored in a database, usually by storing the data in-memory, whereas the back-end database must use disk-based storage to ensure persistence of data (although most databases usually do some caching as well). Many well-known caching products like Redis, Memcached, and similar services are provided as services on major cloud providers. These types of caching layers are used to improve performance of various aspects of the Webex platform.

Management and Analytics Data Storage

Although we have not addressed management and analytics yet, these two areas require extensive capabilities to store, index, and retrieve large datasets. For example, a Webex client in a meeting sends media metrics to the cloud every minute. This large dataset includes not only packet loss, jitter, and latency but also details such as which microphone is being used, the model of the PC, and how much CPU and memory are in use, along with many, many more metrics. As you can imagine, when the cloud platform processes millions of meetings a day, the size of this dataset is huge.

Logging Data

Along with the data needed for analytics such as the per-minute media metrics, logging is an important part of any cloud application, allowing developers to quickly diagnose and monitor issues as they occur. NoSQL databases like Splunk and Opensearch are heavily utilized for logging infrastructure. One key consideration of the databases used to store this data is the cost associated with retaining data. This means that these databases must have a way to easily discard data after a certain period of time. For some services that generate huge amounts of logging data, this means the amount of data retained is usually measured in days.

Metric data also leverages similar infrastructure, but instead of discarding old

metric data, it is sometimes aggregated or saved in a lower fidelity. For example, some services might log something like memory utilization every minute, but after seven days, the data is aggregated to hourly metrics and perhaps, after several months, aggregated further to daily metrics. In this way, cloud engineers can visualize and monitor long-term trending while also getting operational metrics for troubleshooting issues that arise where more precise granularity is needed.

Where data is stored is becoming increasingly important to many customers and carries legal ramifications in many countries. For example, the European Union's General Data Protection Regulation (GDPR) sets strict requirements on where PII may be stored. These requirements complicate how databases and other data stores are used because services need to be aware of which users are in which regions and make use of the appropriate databases as needed.

One way to avoid these restrictions is by removing PII when possible. For example, certain logging data stores might have PII such as names or email addresses removed from the logs before being stored. Removing this information allows for more flexibility in where the data is kept at the expense of losing data that might be needed for certain troubleshooting scenarios. Cisco is transparent in how and where it stores data and takes great care to adhere to these guidelines. Because these regulations are constantly changing and evolving, the Cisco Trust Portal (<https://trustportal.cisco.com/>) provides up-to-date documentation on regulations, and data is managed throughout all products, including Webex.

Integration Services

Integration services serve an important role in the Webex platform. Within the Webex platform, most services leverage RESTful APIs to communicate with each other, and most front-end web pages make extensive use of APIs. You can get a good feel for how any web-based application works by opening the developer tools in your web browser of choice and examining the network traffic generated by the website. Most modern web applications use this approach, and Webex is no exception. Webex makes use of many internal APIs that are not documented—primarily because they are subject to change at any time. However, Webex provides a rich set of public APIs that

allow customers, partners, and other developers to integrate with and extend the capabilities of Webex.

The Cisco Webex for Developers site at <https://developer.webex.com> provides extensive documentation on the various APIs available for the platform. The list of APIs is constantly being updated as new features are introduced into the platform. The developer site provides documentation on how the APIs work, how to authenticate against the APIs, and many examples. The site also allows you to log in and interact live with the APIs directly from the website without the need for external tools. This capability makes it easy for you to get a feel for how the APIs work. Additionally, Cisco DevNet maintains several GitHub repositories with code samples to give developers a starting point for writing their own applications.

Webex also provides a variety of software development kits (SDKs) that allow for integration of Webex services into custom applications. For example, there are SDKs for the major mobile platforms (iOS and Android), along with a browser SDK that allows you to embed functionality into your own websites.

Most SaaS applications provide similar API capabilities that allow you to integrate capabilities across disparate SaaS applications. Webex makes use of APIs provided by other SaaS platforms to extend the capabilities of Webex. For example, Webex can integrate with Microsoft 365 for directory synchronization, authentication, and calendar integration. These capabilities are made possible by Webex implementing features that adhere to the APIs documented by third parties. Similarly, other SaaS applications can make use of Webex APIs to integrate its capabilities into their own applications. This is one of the most powerful capabilities of SaaS over traditional privately hosted applications. By having public-facing APIs, customers can more seamlessly integrate their business processes across applications.

One such example of integrating with other products is the Webex capability to integrate with education or learning management systems (LMSs). Standards such as the Learning Tools Interoperability (LTI) standard facilitate these types of integrations by the industry agreeing on one way of performing these integrations. This means that a provider like Webex only needs to provide an integration compatible with LTI, and then any LTI-compatible LMS can interoperate with Webex.

Similarly, the System for Cross-domain Identity Management (SCIM) has become an industry standard for exchanging identity information between SaaS applications. At the time of this writing, Webex supports the SCIM 2.0 specification, allowing for synchronization of identity information in a standardized way. As the industry continues to agree on standards like SCIM and LTI, interoperability between SaaS applications will continue to grow.

Management and Analytics

Webex provides extensive management and analytics features, primarily through Webex Control Hub. Control Hub is the administrative web interface to the Webex SaaS applications. [Figure 5-17](#) shows an example of a configuration page for a user in Webex Control Hub.

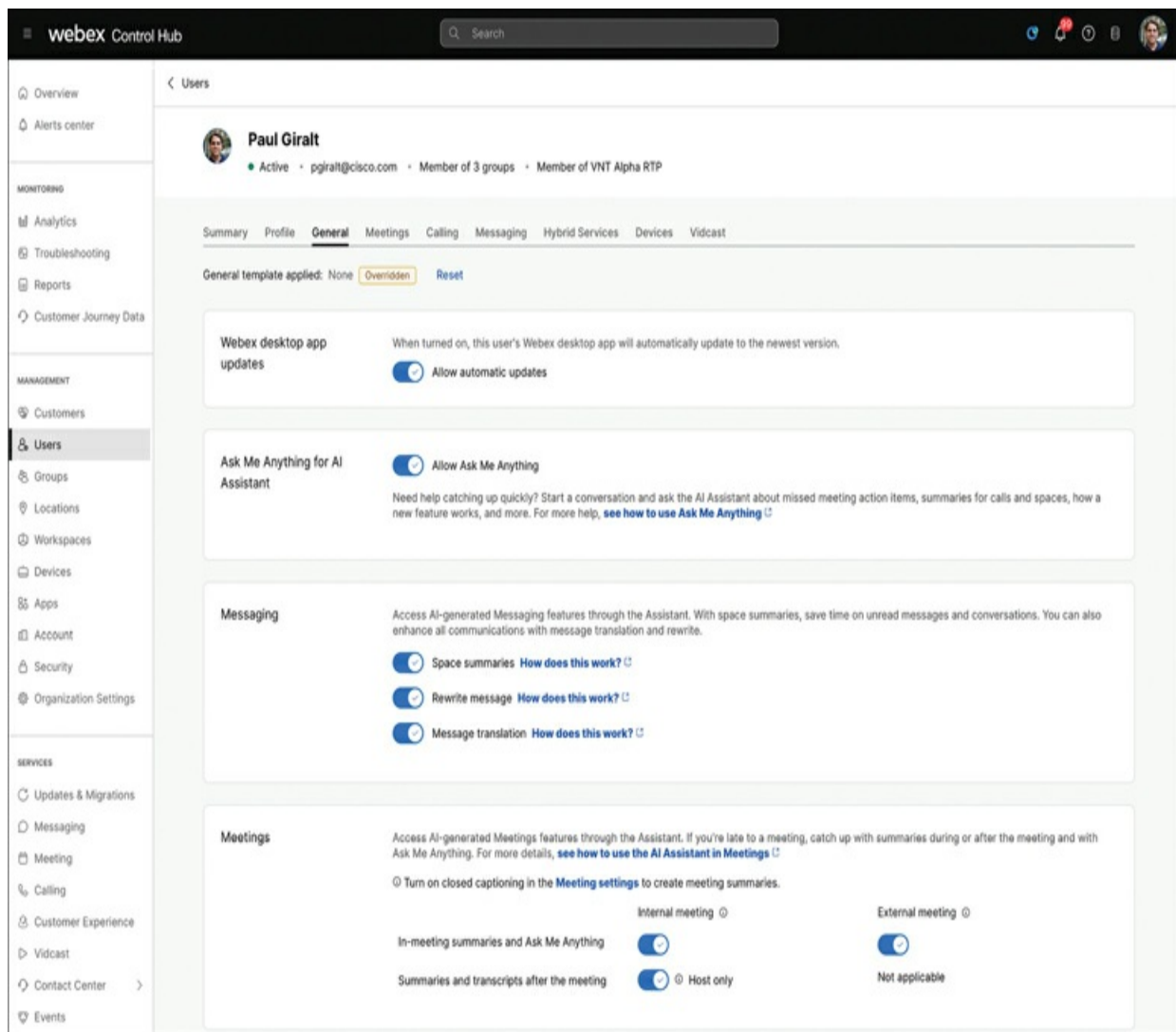


Figure 5-17 User Configuration in Webex Control Hub

The primary use of Control Hub for most administrators is for configuration and management of the various features in Webex. Control Hub provides a unified interface for configuring all meeting, messaging, calling, and contact center features. The navigation on the left provides access to various features like Users, Groups, Locations, Workspaces, and Devices. It also provides access to organizationwide settings and other global settings like security settings. Control Hub also provides access to service-specific configurations for the various workloads such as messaging, meetings, calling, and contact center. It also provides bulk administration capabilities to facilitate large import or change operations.

In addition to configuration management, Control Hub provides a series of

analytics features that help administrators look at trends over time to determine how Webex is being used in their enterprise and expose potential issues that might need resolution. [Figure 5-18](#) shows an analytics dashboard in Webex Control Hub.

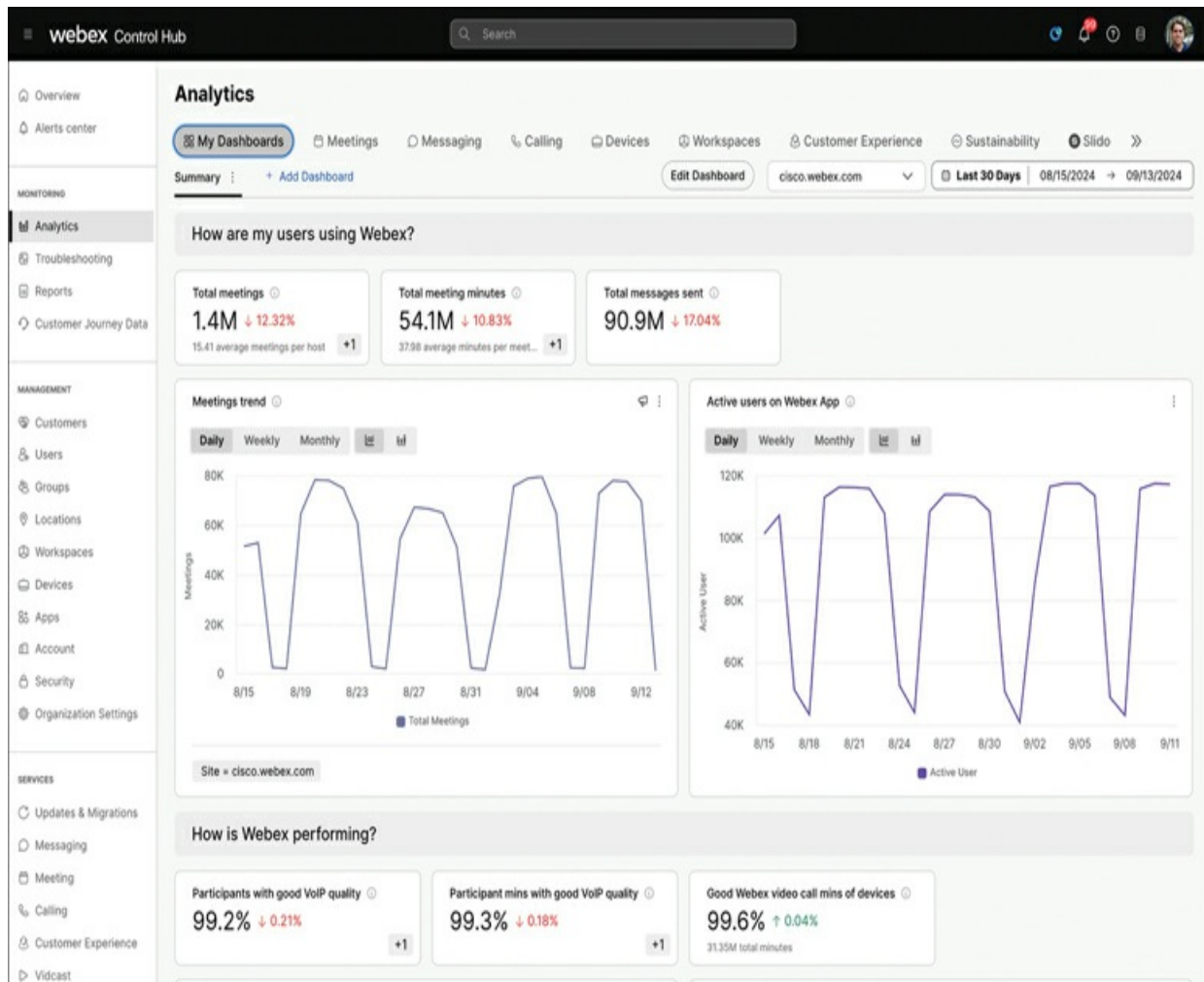


Figure 5-18 Webex Control Hub Analytics Dashboard

This dashboard shows the trend of meetings over the last 30 days as well as active user counts for the Webex app. The analytics dashboard also highlights information on trends in call quality. Several of the dashboards allow users to drill down further and filter on certain criteria.

For customers making use of Webex Room and Desk devices, Control Hub provides powerful insights into how meeting rooms are being used. This tool gives building managers the data they need to decide whether rooms are

being used optimally. For example, [Figure 5-19](#) shows the Workspaces tab in the Analytics section of Control Hub.

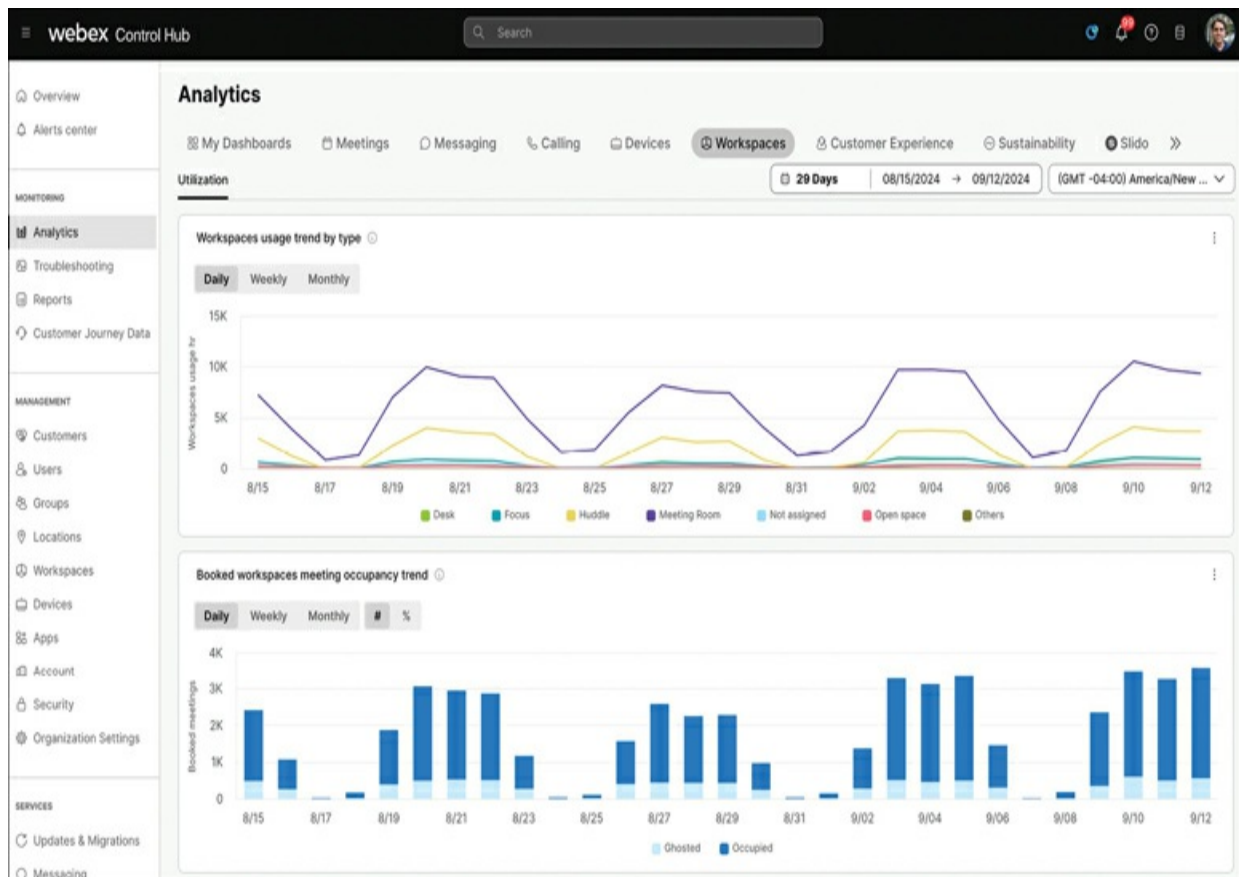


Figure 5-19 Workspaces in Webex Control Hub Analytics

The second graph on this page shows the overall trends in how often rooms are being used, but also the number of *ghost meetings*, meaning meetings where someone schedules a room but the room system never detects anyone in the room during the time of the meeting. Several features in Webex can help free up ghost meetings, and the analytics pages can help administrators understand where they might need to educate employees to book only rooms they need to ensure they are available to others who might need them.

For rooms with devices equipped with environmental sensors, Control Hub also provides detailed information on environmental conditions and shows trends over time. This information can be used to pinpoint problems in rooms with temperature, humidity, air quality, or ambient noise. [Figure 5-20](#) shows the 30-day graph of air quality in a meeting room with a Room Kit EQ.

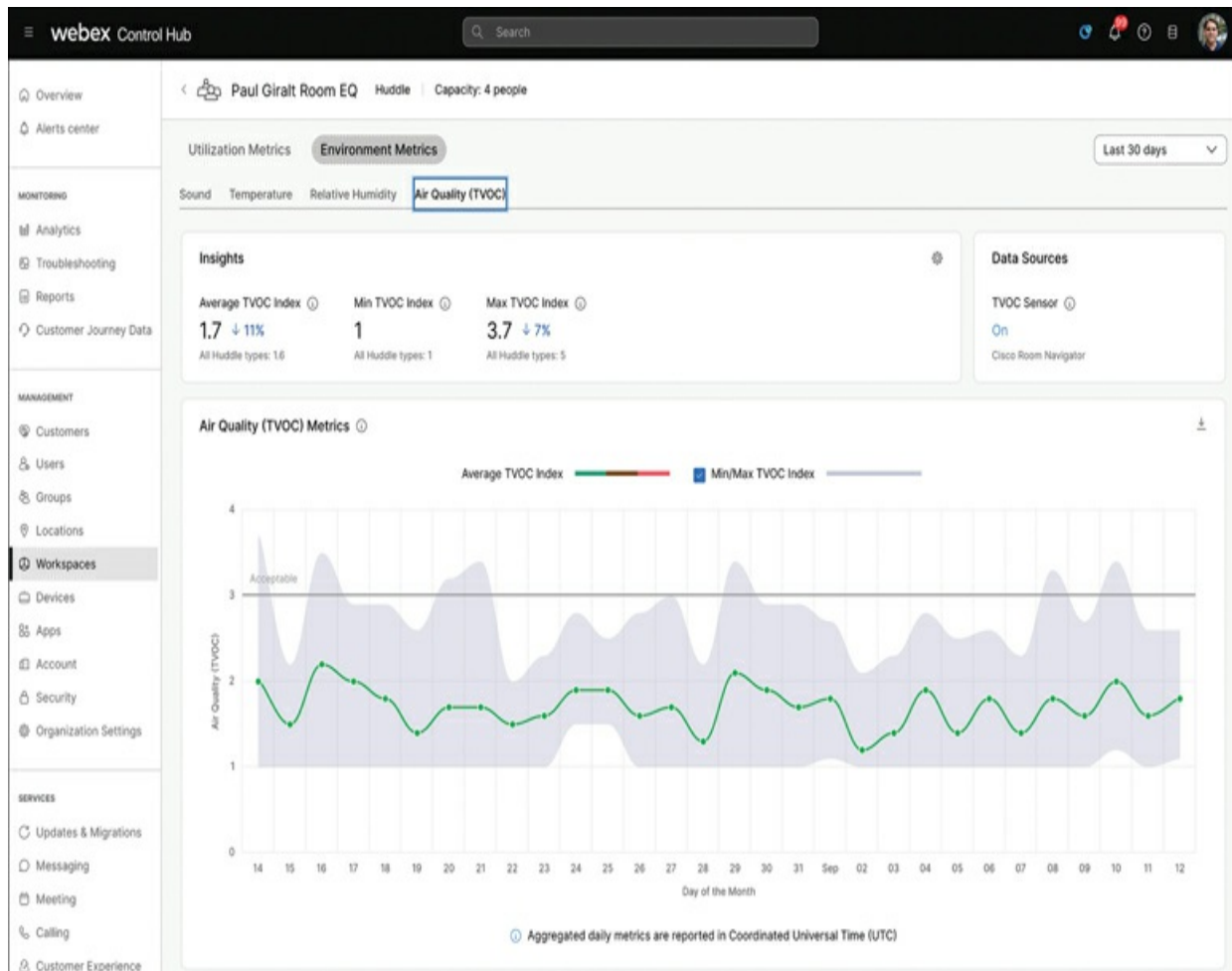


Figure 5-20 Room Air Quality Trends in Webex Control Hub

Although Control Hub is useful for analyzing this data, Webex can also integrate with external applications to provide occupancy and environmental data to applications such as Cisco Spaces, which can combine this information with network and sensor-based information to create interactive signage suited for today's hybrid work environments. API integration services are responsible for facilitating these integrations, which are also configured through Webex Control Hub.

In addition to management and analytics, Webex Control Hub also provides extensive troubleshooting capabilities. These features allow administrators to search for meetings or calls and view media-quality metrics and other diagnostic information for each user in the meeting or call. Administrators can further drill down into individual participants and view detailed information about packet loss, jitter, and more, as shown in [Figure 5-21](#).

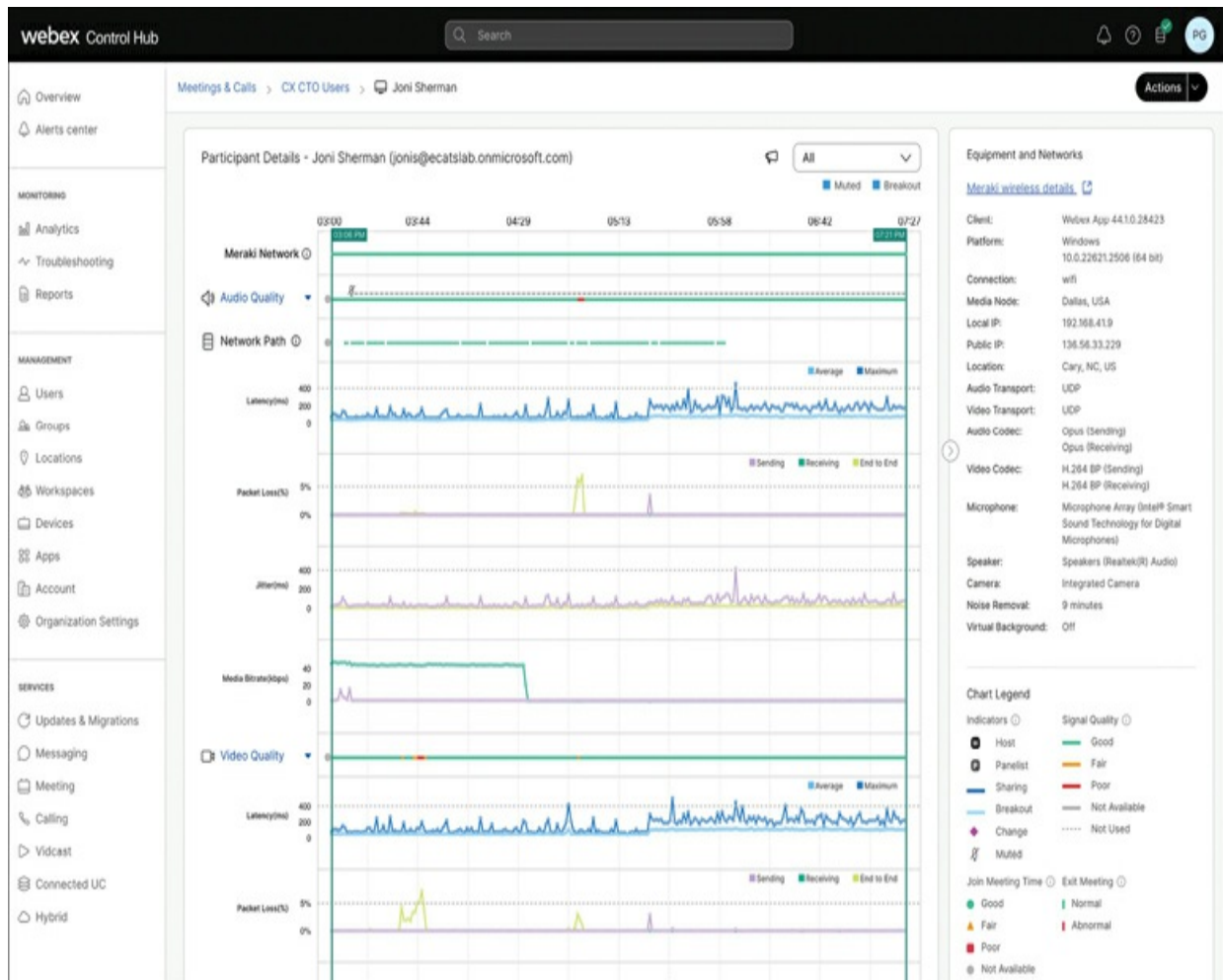


Figure 5-21 Control Hub Troubleshooting for Meetings

Control Hub allows administrators to integrate Webex with both Meraki Dashboard and ThousandEyes to augment the information displayed in Control Hub and provide additional insight when troubleshooting. In the detailed user metrics shown in [Figure 5-21](#), a line for Meraki Network comes from data in Meraki Dashboard, and another line for Network Path comes from ThousandEyes. Those two lines allow for additional drill-down for more information and, if necessary, a cross-launch to those products for even more details. This seamless, in-context integration is easy to enable because all three platforms are SaaS applications that provide open APIs to facilitate such integrations.

Security and Privacy

We have already touched on several aspects of security and privacy in the Webex platform when we discussed some of the various services that enable many of the security and privacy-related features, including the identity, authentication, and authorization services as well as the key management and encryption services.

We also have already discussed how key management and encryption permeate across the Webex platform with all data being encrypted when stored at rest as well as in transit and allowing customers to manage their own key management servers if they choose to do so. One of the features enabled by this security infrastructure is Webex end-to-end encrypted (E2EE) meetings. E2EE meetings not only ensure that all communication is encrypted and intermediaries cannot decrypt the media, but also allow attendees to verify the identity of participants in a meeting to ensure that they can trust everyone there. The identity can be verified either by the Webex certificate authority (CA) or a third-party CA or identity provider.

In addition to the infrastructure and platform security features, the Webex platform also provides many features that enable administrators to control how and who users can interact with. This capability starts with role-based access where users in meetings are hosts, co-hosts, authenticated attendees, or guests. Webex spaces and teams include moderation features so that only certain users have certain privileges in each space. Administrators can control whether users can exchange messages with or attend meetings with users from other organizations. Administrators can also control which third-party applications, integrations, and bots the users in an organization can interact with so that customers who want to certify any third-party application before allowing users to use them can do so.

As mentioned earlier, Webex can integrate with data loss prevention platforms like Cisco Cloudlock to ensure that messages exchanged on the platform do not inadvertently (or intentionally) leak sensitive data by allowing all messages and attachments to be scanned before being available for download or viewing. Attachments are also scanned for malware.

Webex Control Hub provides extensive auditing and compliance features to log all administrative activities on the platform and enable eDiscovery through a dedicated compliance officer role.

In the authentication section, we discussed how the Webex identity infrastructure in the Webex platform makes heavy use of OAuth 2. We also mentioned that services make use of machine accounts so that the services can authenticate against each other. The scopes of these accounts are tightly controlled to ensure that a service has access only to the things that it needs access to.

At the time of this writing, the Webex platform had achieved the following security and privacy certifications:

- SOC2 Type II and SOC 3
- ISO 27001/27017/27018/27701
- ISO 9001 certificate
- Cloud Computing Compliance Controls Catalog (C5)
- HITRUST
- FedRAMP

It also meets the following regulations for privacy and security:

- Health Insurance Portability and Accountability Act (HIPAA)
- General Data Protection Regulation (GDPR)
- Family Educational Rights and Privacy Act (FERPA)
- Children's Online Privacy Protection Rule (COPPA)
- California Consumer Privacy Act (CCPA)

Cisco is committed to responsible AI and takes great precautions to ensure security and privacy of all AI-related features on the platform. One example is the facial recognition feature. This feature allows users to upload their photo, which is then used for detecting users in conference rooms using Webex devices. The Webex devices never send a copy of the faces they detect in the room. Instead, local AI algorithms run on the device and then use the outputs of the algorithms to search through metadata in the cloud that is a mathematical model of facial features but cannot be used to reconstruct the image of the face. Because the feature itself is entirely opt-in, users who do not feel comfortable having their face detected do not need to upload their

photo. You can find a full version of the Cisco Responsible AI Framework by searching the web for “Cisco Responsible AI Framework.”

Summary

The Cisco Webex platform provides market-leading meeting and messaging features through a cloud-delivered Software as a Service. In this chapter, we discussed how the various components and services of the Webex platform line up to our SaaS architecture and how these services are used to enable meeting and messaging features. The Webex platform is complex, built from decades of development, and deployed in many global data centers making use of both public and private cloud services. It is a service that clearly takes advantage of the many advantages of the capabilities available when a product is delivered as SaaS. In this chapter, we specifically discussed the meeting and messaging features of the Webex platform, but calling is also an important workload offered by the platform and that is covered in the next chapter.

References

- Webex App and Features: <https://www.webex.com/all-new-webex>
- Webex messaging security: Cloud collaboration security technical paper (2022):
https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/cloudCollat/spark-security-white-paper.pdf

Chapter 6. Collaboration: Webex Calling

The ability for humans to communicate with each other across vast distances has evolved incredibly over the past hundred years, but over that time, one basic need has remained constant: the ability for someone to pick up a phone and call another person by dialing their phone number. While many platforms have enabled voice and video communication over the Internet, these capabilities often require using special applications, navigating to websites, or leveraging proprietary protocols. Nonetheless, phone numbers remain the universal method to connect with virtually anyone in the world.

While the user experience of picking up a phone and dialing a number to reach another person is a simple one, the infrastructure required to make that call connect and maintain good quality for the duration of the call is far from simple. Phone networks have evolved from analog phone systems dependent on human operators to connect calls to modern digital, IP-based telephony systems, but what has made telephony successful has been the ability for different telephony service providers to interconnect and interoperate to facilitate the simple user experience.

The focus of this chapter is not on the evolution of the public telephone network, but rather the evolution of private telephone networks that use the public network to communicate with each other and how this capability has led to SaaS platforms that deliver telephony features. As telephony evolved, companies needed the ability to enable their employees to make calls.

Purchasing a phone line from a service provider for each employee seemed impractical and expensive; in addition, it did not provide the flexibility companies needed. Some forms of private phone networks date back to the

days of Alexander Graham Bell, but private branch exchanges (PBXs) as we know them today did not become popular until the 1970s with the introduction of electronic switching. PBXs allowed companies to have their own private phone system for calling within the company while also allowing users to place and receive calls from the public switched telephone network (PSTN). Calls between phones in the company did not require traversing the PSTN, saving on costs for individual phone lines for each employee while also, over time, adding many features that allowed employees to more efficiently handle calls. These early PBXs were still analog but eventually evolved again to incorporate time-division multiplexing (TDM) circuits. Companies like AT&T, Lucent, Nortel, Siemens, and Mitel built large, refrigerator-sized cabinets of equipment that terminated copper connections from phones to enable communications between those devices and connect them to the PSTN.

PBXs brought companies a new level of flexibility, allowing them to share expensive phone lines among a large population of employees. These PBXs enabled features such as private dial plans where users could call each other using an extension instead of a phone number and allowed large companies to interconnect their PBXs through private circuits to bypass the PSTN entirely when dialing between offices. PBXs also implemented a variety of productivity-enhancing features like the ability to transfer, conference, or forward a call. Over time, vendors added hundreds of features that allowed customers to customize how calls were routed and presented to their users.

For Cisco, the journey into telephony began in the late 1990s when it led in developing products to convert analog or digital TDM voice to packet-based networks. Packet-based networks allowed both data and voice to share the same circuits in a way that was not possible with TDM voice circuits. Technologies such as voice over frame relay (VoFR) and voice over asynchronous transfer mode (VoATM) were initially used to carry voice traffic over private packet networks, but this technology evolved to voice over IP (VoIP), allowing for interconnection of different packet networks over a common IP transport and even over the public Internet. This capability was particularly appealing because of the ubiquitous nature of IP networks. Transporting voice over IP networks gave customers a way to carry both voice and data over the same network, leading to lower costs, while standards like H.323 and SIP allowed for interoperability between different VoIP

platforms, further driving innovation and lowering costs.

In 1998, Cisco acquired Selsius Systems, the company that arguably built the first commercially successful IP phone. The technology that came from Selsius combined with Cisco's existing VoIP gateway experience and innovations in power over Ethernet (PoE) led to Cisco dominating the enterprise voice market. Over time with other acquisitions such as Active Voice and Tandberg, this technology evolved into an extensive on-premises portfolio of products such as Cisco Unified Communications Manager (CUCM; previously known as Cisco CallManager), Cisco Unity Connection, and Cisco Expressway, along with a variety of products in the video and contact center spaces.

By the early 2010s, with cloud-based meetings like Cisco Webex becoming popular, there was a growing interest from customers wanting to simplify the management of their collaboration services by relinquishing control of their calling infrastructure in much the same way as they had done transitioning their meetings infrastructure to the cloud. For this reason, many customers turned to managed service providers who would provide Unified Communications as a Service (UCaaS), the first evolution toward SaaS. In this transition period, Cisco partners would host the on-premises versions of the collaboration portfolio in their data centers and then offer these capabilities to customers as a service. Cisco built software capabilities that allowed these partners to manage these deployments at scale. This approach allowed customers to pay for their collaboration services on a subscription basis and leave the day-to-day management of the infrastructure to someone else, but these partners were still just operating individual calling environments for each customer.

The move to a true SaaS model for calling services began with Cisco's Spark Call service, introduced in the late 2010s. This was Cisco's first move toward a cloud-native infrastructure for calling, leveraging the infrastructure it had built for the Cisco Spark service (which would eventually become Webex Teams and then just the messaging features of Webex). In 2018, Cisco acquired Broadsoft, a company with a long history of highly scalable, multitenant, service provider class calling solutions. Over time, technology from the Broadsoft acquisition was used as the foundation for what is now Webex Calling, and customers on the Spark Call platform were transitioned

to Webex Calling.

Today, Webex Calling is a highly scalable, feature-rich platform that allows customers to provide calling capabilities to their users from a pure multitenant SaaS platform. Webex Calling is fully integrated into the overall Webex platform; however, in this chapter we will discuss the capabilities specific to calling and the architectural considerations unique to providing a calling platform from the cloud.

Product Capabilities

Webex Calling is an integral part of the Webex platform, leveraging much of the infrastructure discussed in [Chapter 5, “Collaboration: Webex Meetings and Messaging,”](#) for tasks such as user management, licensing, configuration management, monitoring, and many other microservices required to provide a cloud calling platform. Webex Calling adds additional calling-specific services to provide calling capabilities to the Webex app and devices like IP phones. [Figure 6-1](#) shows some of the calling features incorporated into the Webex app, such as the ability to dial a contact from a chat, view recent and missed calls, and check voicemails.

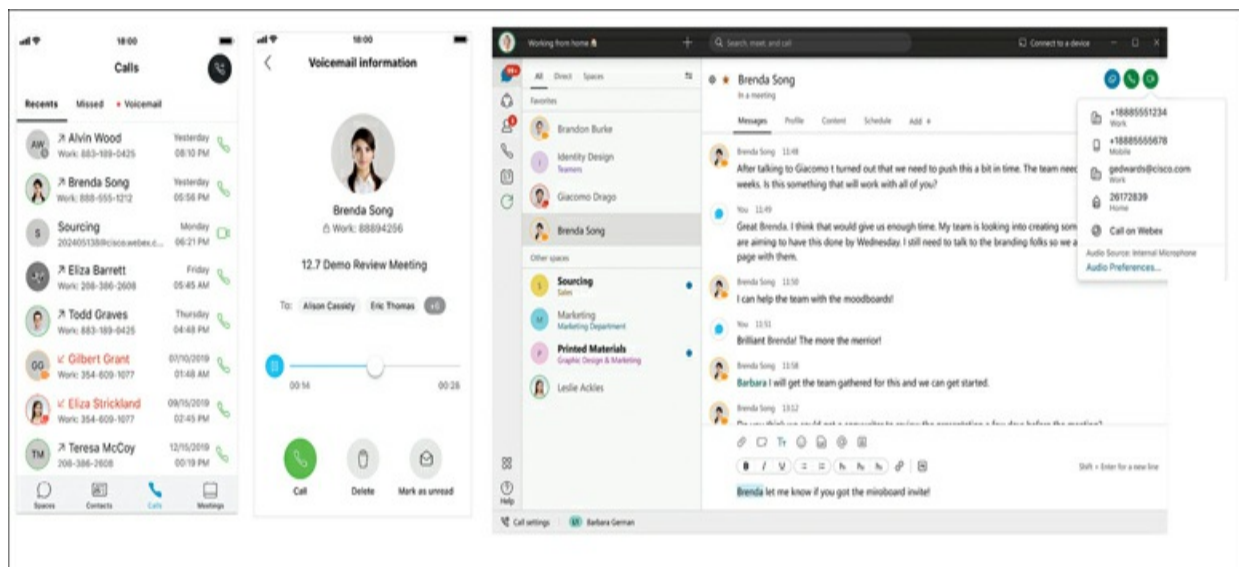


Figure 6-1 Calling Features in Webex App

While we discuss the cloud-native Webex Calling platform in much of the chapter, it's worth mentioning that Cisco also offers a product called Webex

Calling Dedicated Instance (DI). While Webex Calling DI leverages some of the same cloud services as the rest of the Webex platform, it is more akin to the UCaaS platforms that evolved during the transition to SaaS. With Webex Calling DI, Cisco operates and maintains dedicated instances of the same software that customers can run on-premises such as Cisco Unified Communications Manager, Unity Connection, and Cisco Expressway. Cisco has built some cloud native services around this software and has done some work to integrate it with management tools like Control Hub, but it still remains architecturally more like UCaaS. For example, because the call control for Webex Calling DI is Cisco Unified Communications Manager, new features to the calling platform rely on new UCM releases, which happen at a much less frequent pace than a modern SaaS platform.

When we say *Webex Calling*, we are referring to the multitenant SaaS product, and we will specifically use the term *Webex Calling DI* to refer to the dedicated instance product. We will still discuss some of the architectural pieces to tie the Webex Calling DI infrastructure to the Webex Calling components, but we will largely focus on Webex Calling and not DI in this chapter.

Some of the key capabilities we will discuss include

- Call routing and dial plan management
- User management
- PSTN connectivity
- Emergency calling
- Calling endpoint support and management
- Calling features
- Customer experience features
- Interoperability
- Analytics and troubleshooting
- Security and compliance
- High availability and survivability

- Artificial intelligence (AI) features

Now let's explore some of the capabilities of Webex Calling.

Call Routing and Dial Plan Management

The most basic function of a calling platform is to take information from a user that indicates the destination they wish to call and routing the call to the destination. The desired destination is communicated in the form of a number—a phone number, extension, or some other string of digits that indicates the destination. A calling platform must interpret these digits and determine the appropriate end device (or devices) to alert of an incoming call.

While this task might seem relatively straightforward, the complexity of global dial plans and nuances in how customers want to route calls can lead to complex logic needed to facilitate call routing.

Webex Calling has a concept of a *location*, which represents a logical grouping of calling users and devices, oftentimes tied to a specific physical location. For example, a retail customer might create a location for each retail outlet and then a location for each of its back-office campuses or offices. Multinational corporations might have locations all over the world. By grouping devices and users into a location, administrators can configure certain call routing policies and dial plan entries based on the location of the device or user.

Users in Webex Calling can be assigned an extension, a telephone number (TN), or both. A TN corresponds to a globally unique phone number as defined by the E.164 international standard. The Webex Calling platform has detailed information of the dial plans for every country in the world and can ensure that TNs adhere to the requirements of that country. These phone numbers are typically represented in +E.164 format, meaning that the digit string begins with the plus (+) character followed by the country code and the national number based on that country's dial plan. The length of the country code and national number can vary from country to country. For example, in the United States, the country code is 1 and national numbers are 10-digits in length, so a TN for a U.S.-based user would look something like +19195550123. The United Kingdom uses similar length national numbers

but uses 44 as the country code, leading to a number that looks like +442079460123. Because TNs are globally unique, no two users can have the same TN, even if they are in different locations.

Different parts of a national number can have different meanings within a given country. For example, in the U.S., 10-digit national numbers are subdivided into a 3-digit area code and 7-digit number. The 7-digit number is further subdivided into a 3-digit office code and 4-digit subscriber number. Webex Calling understands these nuances of dial plans within each country, which allows it to make intelligent decisions such as allowing administrators to restrict specific types of calls based on country-specific dial plan components. For example, [Figure 6-2](#) shows the outbound calling permissions settings for a Webex Calling user in the U.S.

Permissions by type

Manage the permissions by call type for this user. Different countries and long distance calls require calling plans with specific prefixes. See [calling plans by country](#) for more information.

☐ Location settings

The default settings are based on VNT Alpha RTP. To change the default permissions for all users, [manage the VNT Alpha RTP settings](#).

☒ Custom settings

Manually set this user's outgoing call permissions by type.

Call type	Permission	Allow transfers / forwards ⓘ
Internal	<div>Allow</div>	<input checked="" type="checkbox"/>
Toll-free ⓘ	<div>Allow</div>	<input checked="" type="checkbox"/>
National	<div>Allow</div>	<input checked="" type="checkbox"/>
International	<div>Allow</div>	<input type="checkbox"/>
Operator Assistance ⓘ	<div>Allow</div>	<input checked="" type="checkbox"/>
Chargeable Directory Assistance ⓘ	<div>Block</div>	<input checked="" type="checkbox"/>
Special Services I ⓘ	<div>Require authorization code</div>	<input checked="" type="checkbox"/>
Special Services II	<div>Auto-Transfer to</div>	<input checked="" type="checkbox"/>
Premium Services I	<div>Auto-Transfer to</div>	<input type="checkbox"/>
Premium Services II ⓘ	<div>Auto-Transfer to</div>	<input type="checkbox"/>

Restore to default location permissions

Figure 6-2 Webex Calling Outgoing Call Permissions for a U.S.-Based User

If the user had been in another country, the call types listed on the permissions page would be different because each country has its own unique

call types. For each call type, calls can be allowed, blocked, set to require an authorization code, or transferred to a specific destination. Administrators can also control whether forwarding or transfers are allowed to different call types. These permissions can be applied globally, per location, or per user.

Users can also be assigned an extension in addition to or in place of their TN. Webex Calling requires a fixed extension length for an organization. Extensions can be anywhere from 2 to 10 digits in length and can overlap between locations. For example, if a customer chooses a 4-digit extension length, that customer can assign extension 1000 to several users if those users are in different locations.

While Webex Calling does allow for extension dialing between locations, you can see how this could be problematic if there are overlaps, so customers are advised to use location routing prefixes with a steering digit for routing calls between locations. For example, a customer might choose the digit 8 as the routing prefix and choose 4-digit location routing prefixes (sometimes referred to as site codes). If location A is assigned prefix 8222 and location B is assigned 8333, then the users at extension 1000 at those locations can be reached by the numbers 82221000 and 83331000, respectively.

In addition to numbers assigned to users or phones, a variety of features can have numbers as well. For example, users can dial into voicemail or auto attendants. Auto attendants are often used to reach users who have only an extension and no TN, because callers from the PSTN cannot reach an extension directly. A TN is assigned to the auto attendant, which prompts the caller to select whom they wish to reach. Hunt groups and call queues allow for distributing calls to multiple phones by calling into a single number. Webex Calling also provides virtual extensions that allow users to dial an extension that routes a call out to a +E.164 destination on the PSTN.

Managing phone numbers is an important piece of dial plan management, and Webex Calling provides extensive capabilities to make it make it easy for administrators to obtain, provision, and manage the lifecycle of phone numbers. We will discuss this issue further later in this chapter when we address PSTN connectivity.

User Management

Closely related to dial plan management is user management. Webex Calling makes use of the same common identity infrastructure as the rest of the Webex platform, so everything we discussed in [Chapter 5](#) describes how users are configured, authenticated, and optionally synchronized from an external user database like Microsoft Entra or Active Directory.

When a Webex Calling license is assigned to a user in Webex Control Hub, this allows the admin to configure a variety of additional settings for the user. We will discuss several of the features in Webex Calling later in this chapter, but several features require user-level configuration, which can be performed through Control Hub or, in many cases, via Webex APIs. For example, users in Webex Calling must be assigned a telephone number and/or extension. The ability to provision calling-related settings is seamlessly integrated into the user provisioning mechanisms in Control Hub; however, this integration hides much of the complexity needed to enable the calling services in response to these configuration changes.

PSTN Connectivity

Arguably the most important capability of a calling platform is the ability to place calls to and receive calls from the PSTN, enabling users to reach anyone in the world with a phone number. The global reach of the PSTN means that a calling platform needs to accommodate various requirements set forth by different countries or other municipalities. A properly designed calling infrastructure hides the complexity of PSTN from users.

Webex Calling supports three primary methods of connecting to the PSTN:

- Cisco Calling Plans
- Cloud-connected calling provider
- Premises-based PSTN

As mentioned earlier, Webex Calling groups users and devices into locations. Webex Calling provides flexibility in PSTN connectivity by allowing each location to make use of one of the three different PSTN connectivity methods. This means one location can use a Cisco Calling Plan while other locations use premises-based PSTN, for example. The PSTN capabilities for

one location can also be shared with other locations if desired. Next, let's explore each of these capabilities in more detail.

Cisco Calling Plans

Cisco Calling Plans provide the easiest and most seamless experience for customers because Cisco provides customers with PSTN connectivity and phone numbers as part of the Webex Calling service. Cisco bills customers for these services, so they do not have to interact with any telephone company.

If customers need additional phone numbers, they can easily request them from Control Hub. As shown in [Figure 6-3](#), a user simply selects the location to which they would like to add a number (or numbers); picks the region and, depending on the country, some other filtering criteria such as area code; and then selects from the list of available numbers. These numbers are immediately provisioned for use by the customer, making it easy to add new numbers as needed. Customers can also port phone numbers from another provider into Webex Calling from the Control Hub interface.

Add Numbers (Site1)

Progress: Select a Location (active) | Select Numbers | Done

What kind of numbers do you need?

- Regular phone numbers** (Selected)
 - Individual numbers that can be assigned to users, devices, call features, etc.
- Toll-free numbers
 - Non-local numbers that don't charge the caller.

What area should these numbers be from?

We'll find you numbers in the area code or city of your choice. If you don't find the area code/city you are looking for, you can [open a Cisco Calling Plans support case](#) for more options.

Country: United States of America

State/Province/Region: North Carolina

Search by: Area Code | Area Code: 919 | Prefix: Select an option

How many numbers do you want auto-selected for you?

We can choose up to 10 non-consecutive numbers for you. You will be able to see and change the numbers before submitting the order.

1

Cart [Clear All](#)

You haven't added any numbers. Search and click the displayed numbers to add them here.

Total: 0/10

[Open a Cisco Calling Plans support case](#) [Back](#) [Order](#)

Figure 6-3 Adding Phone Numbers to Webex Calling Using a Cisco Calling Plan

Cisco Calling Plans also offer the ability to provision toll-free numbers and services numbers (e.g., 211, 411 in the U.S.) in some countries and route those calls to a particular device or feature in Webex Calling. Cisco Calling Plans also offer Business Texting services, where the phone number allocated can also make use of Short Message Service (SMS) to send text messages to mobile devices. While this experience is simple and easy for end users to use, the back-end capabilities needed to facilitate these transactions are complex.

Cloud-Connected PSTN Partners (CCPP)

The second PSTN connectivity method available in Webex Calling is a connection through a Cloud-Connected PSTN Partner (CCPP). In some ways,

these services are like the Cisco PSTN Calling Plans in that they are entirely cloud-hosted and require no on-premises infrastructure. Cisco has prearranged peering agreements with a variety of PSTN partners that allow for easy integration into a customer's Webex Calling environment. Cisco and the partner work together to ensure the services are operational and highly available. The peering connections are typically private circuits between the partner and Cisco, ensuring high quality for the calls. [Figure 6-4](#) shows how Cisco Webex peers directly with the cloud PSTN provider and maintains this connection on behalf of the user. The partner then peers to the PSTN provider (or in some cases the partner is also the PSTN provider themselves).

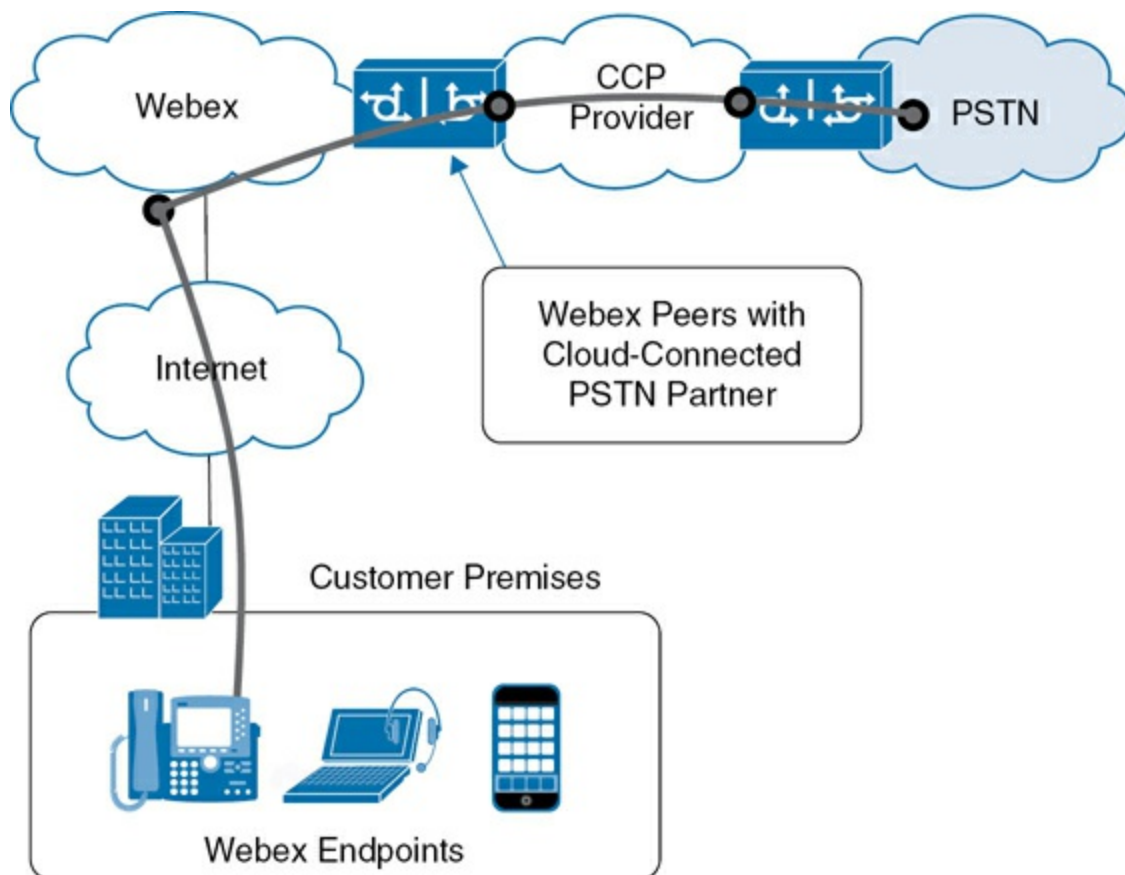


Figure 6-4 Cloud-Connected PSTN Provider with Webex Calling

When using a CCPP, the customer is billed for PSTN services by the partner, not by Cisco. Some partners also have integrated provisioning via Control Hub; this capability allows administrators to request new phone numbers directly from Control Hub, whereas others require that the administrator contact the CCPP directly to obtain new phone numbers. Some partners

support toll-free and services numbers, whereas others provide only standard numbers. Control Hub provides a current list of all supported PSTN partners and their individual capabilities, allowing customers to make an informed decision on which provider to choose.

Again, customers have the flexibility to use not only different PSTN types but also different partners on a per-location basis. If one partner provides services in one part of the world, but a different partner provides better services in a different part of the world, the customer has the flexibility to use both in whatever arrangement works best.

Premises-Based PSTN

The final method of PSTN connectivity available in Webex Calling is premises-based PSTN. In this model, customers are responsible for providing connectivity to the PSTN through whatever means they choose. There are a variety of options, such as a legacy Integrated Services Digital Network (ISDN) circuit or, more commonly, a Session Initiation Protocol (SIP) trunk to a provider of choice. Customers must configure a session border controller (SBC) in a network they manage to provide a connection from the Webex cloud to the local PSTN connection. This SBC is referred to as a local gateway (LGW) and can be either a Cisco Unified Border Element (CUBE; Cisco's SBC product) or one of several certified third-party SBCs.

[Figure 6-5](#) shows how a local gateway might be used to provide PSTN connectivity to users at a site. There is flexibility in how the local gateway is deployed, and in this example, calls from the local gateway are routed through an on-premises Cisco Unified Communications Manager before being sent to the PSTN. This approach is entirely optional, and the local gateway and PSTN gateway can either talk directly between each other or even be the same device providing connections to both the PSTN and Webex.

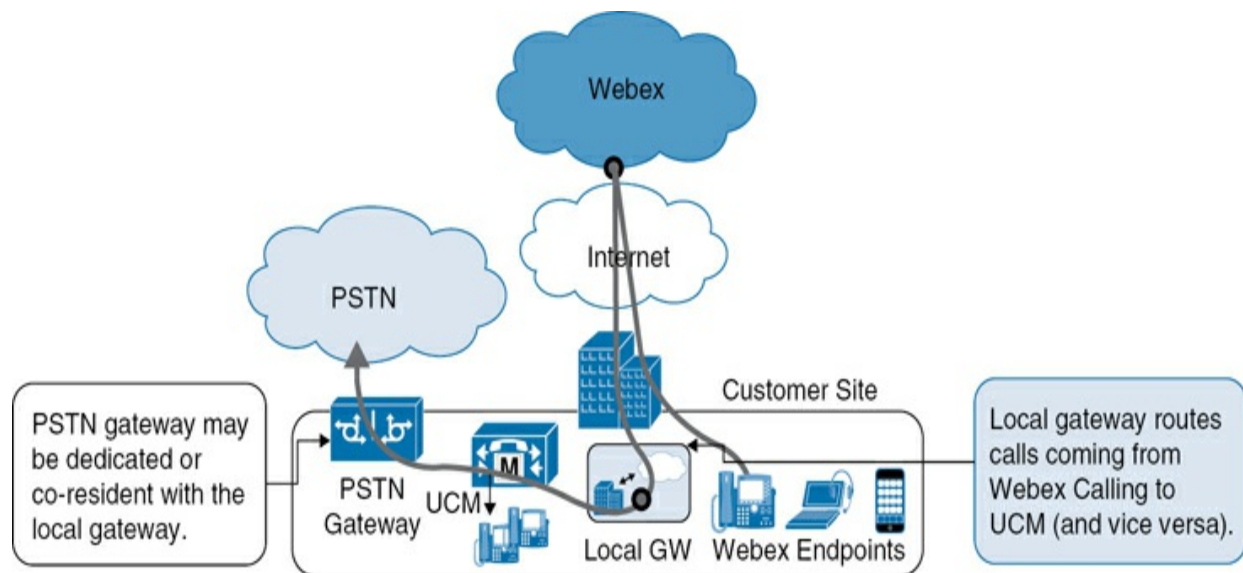


Figure 6-5 Local Gateway at a Customer Site to Provide PSTN Connectivity for Webex Calling

Figure 6-5 also shows how local gateway can be used to provide connectivity to not only the PSTN but also on-premises registered phones and devices. Webex Calling can be configured to route certain calls to the local gateway, and the local gateway can be configured to route those calls to Unified CM instead of the PSTN. This allows for a seamless hybrid integration and can also be used for cloud migrations.

Some partners that are not CCPP providers may choose to host a local gateway on behalf of a customer and provide this LGW as a managed service. In these scenarios, a partner would connect the local gateway to Webex as if they are the customer. The partner may host the local gateway in its own partner-managed network or even in a co-location facility to make interconnecting the local gateway with the PSTN circuits easier. Regardless of which route a customer takes, the customer is responsible for the PSTN circuits and contacting the provider for number provisioning and billing, for example.

A local gateway can be provisioned to connect to the Webex Calling network either over the public Internet or through a direct connection to a Webex Calling data center. We will discuss these deployment options in more detail later in this chapter.

Emergency Calling

In addition to normal local, national, and international number dialing, the ability to dial emergency numbers is crucial for any calling service. Different countries and even municipalities have different laws regarding emergency services. In the U.S., phone systems must comply with E911 regulations that allow emergency services to know the location of a caller. While this problem is generally easy to solve for mobile network providers because of global positioning system (GPS) capabilities in mobile phones as well as the ability to triangulate location using cell towers, determining the location of a user who might move an IP phone from one location to another or who places a call from a soft client like Webex app on a laptop can be more challenging.

In the U.S., RAY BAUM'S Act requires that emergency calls send a dispatch address for any nonfixed or nomadic devices. Cisco partnered with a third party, RedSky, to provide these capabilities to Webex Calling users. This is the power of Webex Calling being a SaaS platform. Webex enabled this functionality for users by peering the Webex cloud with RedSky's cloud services for all Webex Calling users without the customers having to set up their own peering with RedSky.

The E911 services in Webex allow the Webex app to detect when a user changes locations and send the correct location information to emergency services when an emergency call is placed. The app can do this by using the HTTP Enabled Location Delivery (HELD) protocol, which allows a device to report network environment information such as the network switch or wireless access point to which it is connected. If the device cannot determine its location, it will prompt the user to enter their address and remember that address any time the same network environment is detected. When E911 services are enabled, emergency calls do not traverse the configured PSTN provider but rather are sent directly to RedSky for processing. [Figure 6-6](#) shows how E911 calls are routed directly to RedSky while PSTN calls can take other paths, such as through a Cisco Calling Plan or through a local gateway providing premises-based PSTN.

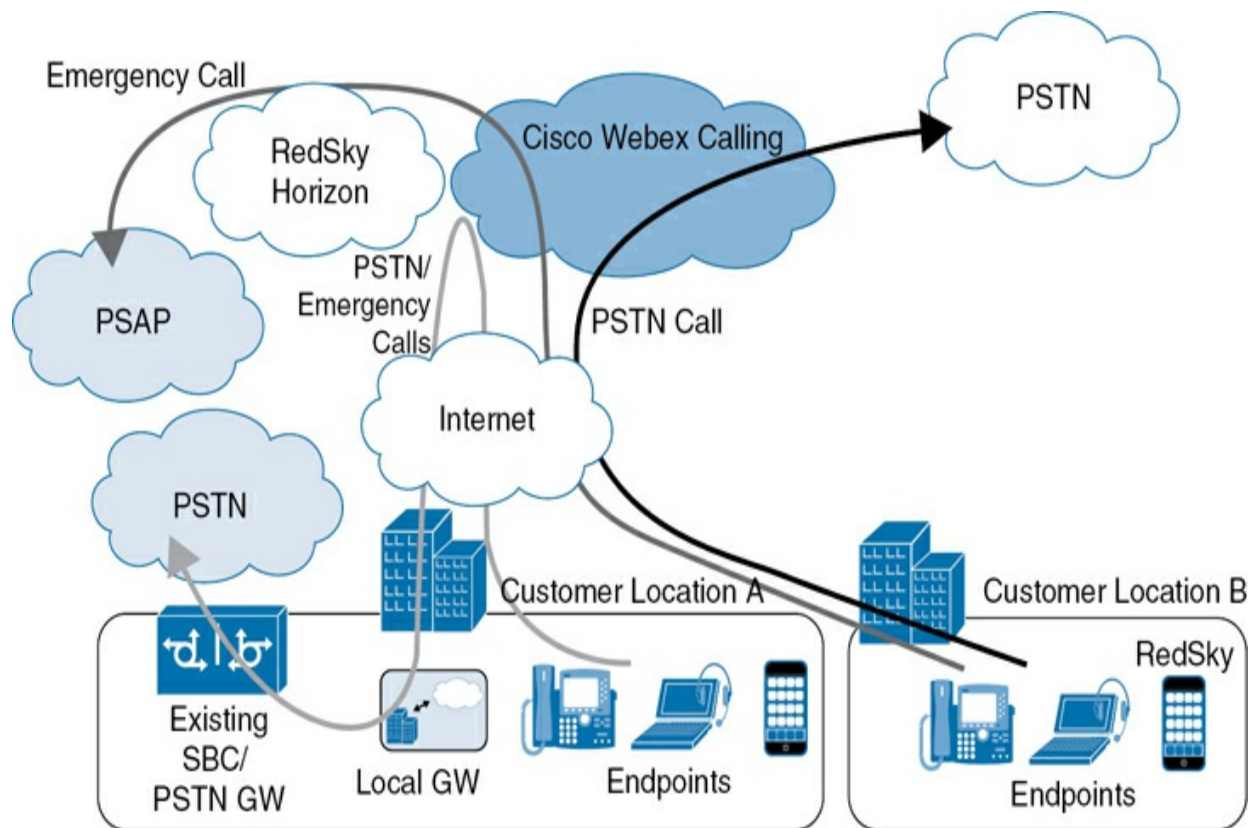


Figure 6-6 Routing E911 Calls Directly to RedSky

By now, you should realize that calling services are different from many other SaaS products in that they interconnect a variety of different devices and networks. While many SaaS products provide services to users on laptops or mobile devices, calling services require connectivity to a variety of PSTN providers; third-party services such as RedSky; and devices such as a local gateway, IP phones, and other calling devices.

Calling Endpoint Support and Management

Another key capability of a calling platform is managing endpoints. Endpoints can either be software-based like the Webex app or physical devices like IP phones.

Webex Calling supports a variety of endpoints and provides advanced provisioning and onboarding features for Cisco devices. It also offers the ability to integrate with third-party devices; however, management and provisioning of third-party devices are not as tightly integrated into the

platform. This means that Cisco devices are generally much easier to deploy at scale.

Most Cisco IP phones can be provisioned either by MAC address or activation code. When using MAC address–based provisioning, the administrator configures the device in Control Hub and enters the MAC address. When the device connects to the Internet and attempts to register, it is permitted to do so based on its MAC address, making it easy to deploy phones if an administrator knows which physical device needs specific provisioning.

In some cases, a customer might want to deploy phones at a site and then assign them a provisioning profile later. In these cases, activation code-based provisioning is typically a better solution. An administrator first provisions the settings for a given phone profile. For example, if a lobby phone for Location A needs to be provisioned with extension 1000, no TN, and an outbound calling profile that restricts calls to only national numbers, the administrator can create the phone device with this configuration without a MAC address. The administrator then requests an activation code from Control Hub and provides it to the user who will activate the phone. This 16-digit number is used to tie a physical phone to the configuration profile. To provision the physical phone, any phone can be sent to the site, and when it boots up, it will ask for an activation code. The user enters the appropriate activation code, and that phone now takes on the profile that was associated with that activation code.

The tight integration between Cisco devices and the SaaS platform means that all provisioning changes are pushed to devices automatically. Additionally, firmware updates for devices are managed through Control Hub by placing a device into an “upgrade channel” that automatically updates devices as new releases are made available. A SaaS platform like Webex makes this process completely seamless to the administrator and end user.

In addition to IP phones, Webex Calling has support for analog gateways for environments that require legacy analog phones. This can include fax machines and phones that are specialized for certain environments such as hotel rooms, or emergency call boxes, which typically still use analog phone technology.

Calling Features

Webex Calling has an extensive list of features beyond just basic calling between two endpoints. Some of these features have existed in PBXs and even the PSTN for a long time, such as caller ID, redial, transfer, conference, hold, music on hold, call blocking, call forward (on various conditions such as busy, no answer, or always), call park, call pickup, call waiting, and many more. It also supports more advanced features such as call recording, location-based E911, shared lines, paging, call monitoring, hoteling (the ability to log in to a phone to make and receive calls using your own number), executive/admin features, business texting, single number reach, advanced analytics, and visual voicemail, just to name a few. For a full list of features, search <https://webex.com> for “Webex Calling feature support matrix.”

Although “calling” is typically associated with audio calls, the Webex Calling platform allows for video calls if the endpoints involved in the call can negotiate video capacities.

Webex Calling also provides for advanced call routing features like auto attendants and call queues. [Figure 6-7](#) shows the configuration page for an auto attendant in Webex Calling.

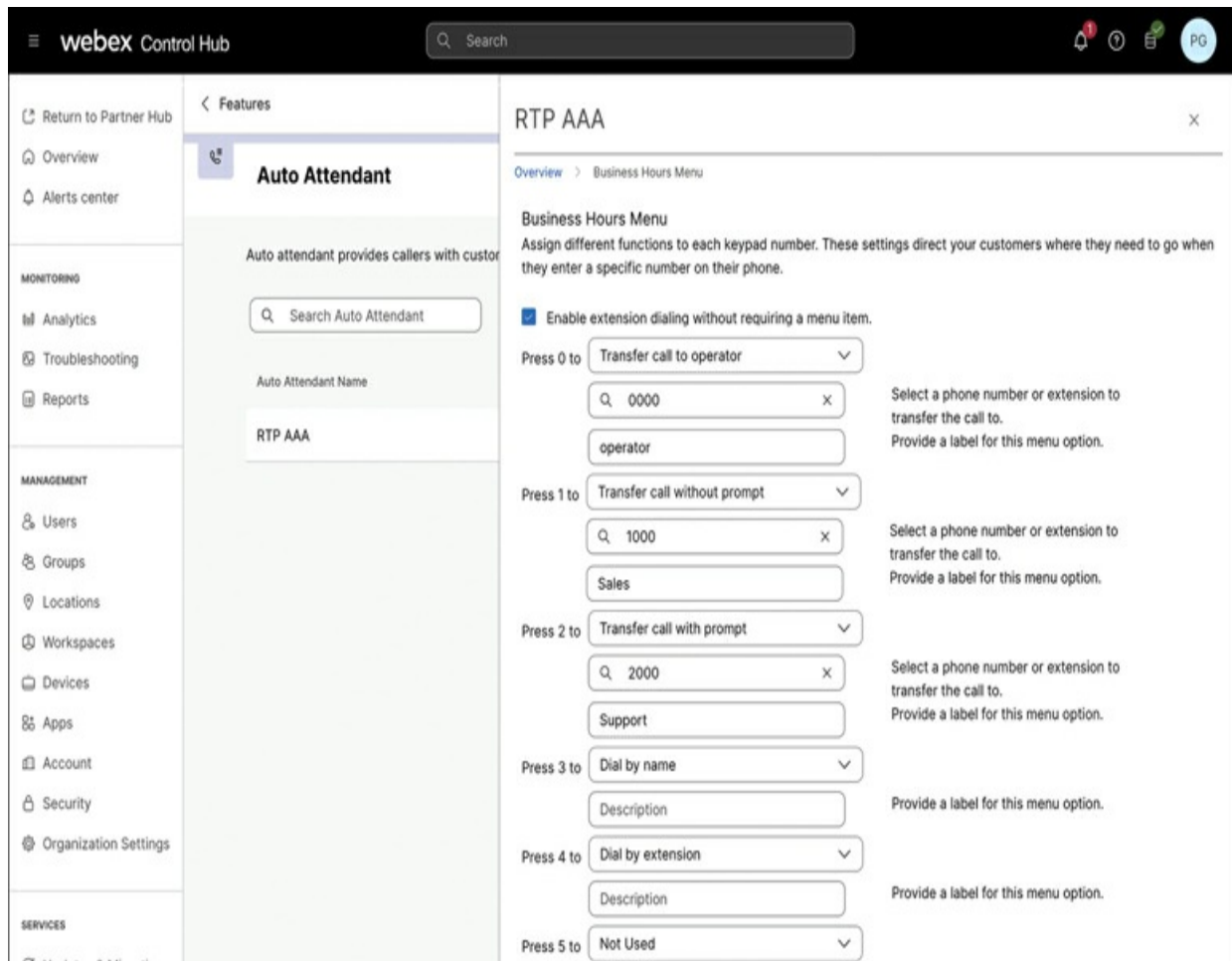


Figure 6-7 Auto Attendant Configuration in Webex Calling

We won't go into the details on all the features in Webex Calling, but some of the available options for auto attendant calls highlight some of sophisticated call routing that Webex Calling enables. For example, an administrator can configure different menu choices based on schedules that determine business hours or after-hours treatment. Different auto attendants can be configured on a per-location basis, or they can be assigned globally to the organization. This capability enables the administrator to customize time zones, for example, and limit extension and user lookups to only those within a location.

Another advanced feature available in Webex Calling is Webex Go, which allows an enterprise to extend Webex Calling to the native calling features on a mobile device. It works by registering a secondary eSim on the mobile device, which adds a second line to the phone. This feature allows Webex

Calling to provide cellular service to the phone like any mobile carrier would, but this line is tied back to Webex Calling and operates just like the line on an IP phone or other endpoint associated with the user. This means that the user can place calls using the same dial habits as used in the office, such as dialing extensions. The user can also receive calls just as any other Webex Calling phone would receive, such as receiving a call in a hunt group or a call directed by an auto attendant. [Figure 6-8](#) shows how a mobile device has two separate cellular connections when using Webex Go.

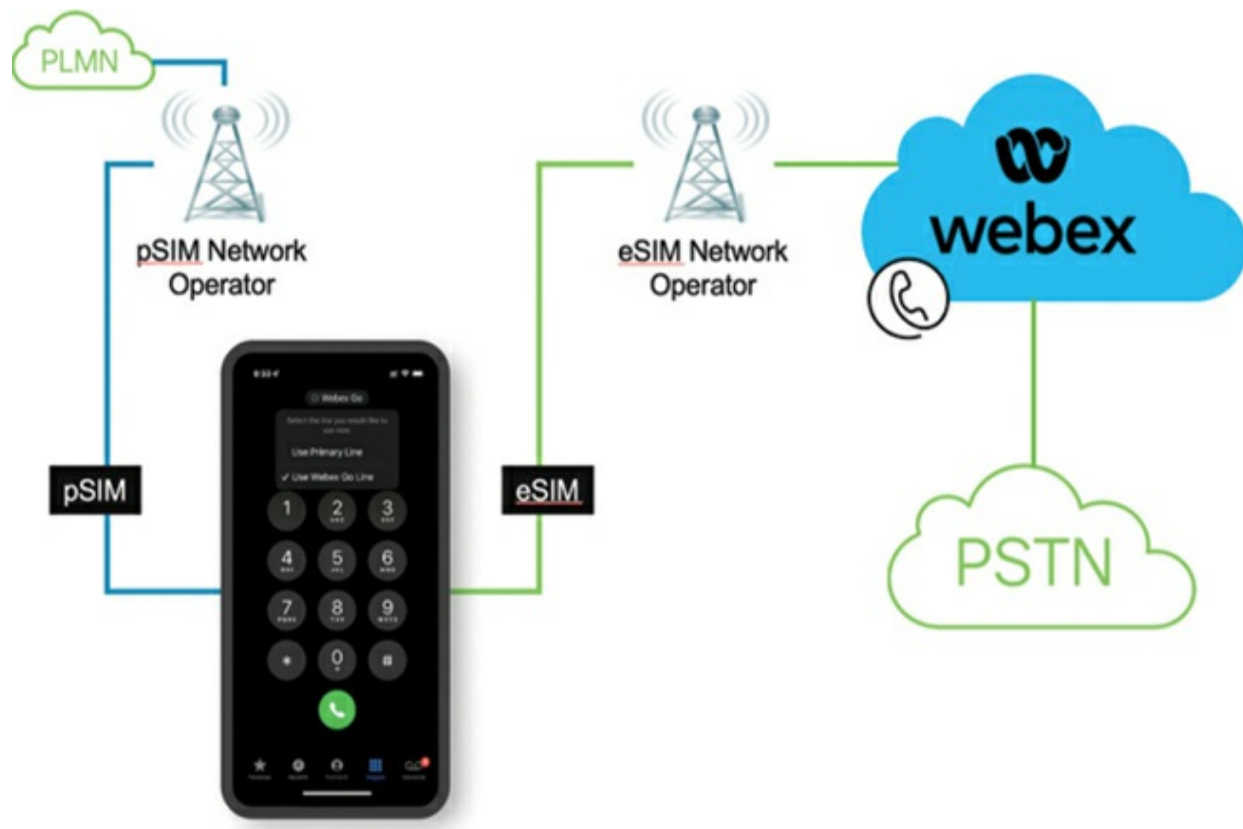


Figure 6-8 A Mobile Phone Using Webex Go

By separating a user's business and personal lines, the end user's privacy is protected because their personal number is not used for business calls. Webex Go does not require any application to be installed on the mobile device and uses the native dialer functionality. Because all calls are anchored in the Webex Calling cloud, administrators can apply any security or compliance policies, including recording all calls made to or from a user. Activating Webex Go on a mobile device is as easy as generating a QR code in Control Hub and scanning it on the phone, making the onboarding process easy.

Webex Calling also offers features that integrate with the rest of the Webex platform. For example, [Figure 6-9](#) shows how a user can click an Invite and meet button from a call in the Webex app and instantly escalate the call to a fully featured Webex Meeting.

Images



Figure 6-9 Escalating a Call to a Webex Meeting in Webex App

The option illustrated in the figure is far easier to enable in a SaaS platform like Webex as opposed to traditional software running in an on-premises environment. To enable this experience, the system needs to create a meeting and then move the participants on the call into the meeting as individual meeting participants. The process of escalating a call in Webex Calling to a meeting in Webex Meetings is complex, but SaaS allows for the two systems to be integrated seamlessly to enable a simple user experience.

Customer Experience Features

Webex provides a fully featured contact center offer as part of the Webex

Contact Center product; however, some customers have basic needs that do not warrant a full contact center solution. For these customers, Webex Calling includes a set of features called Webex Customer Experience Basic and Webex Customer Experience Essentials.

Webex Customer Experience Basic includes features such as voice queues with skills-based routing and customer callback, voice queue analytics and reporting with live queue agent stats, and a native agent experience in the Webex app, just to name a few. Webex Customer Experience Essentials builds on the Basic feature set and adds screen pops, supervisors, and enhanced reporting and analytics. [Figure 6-10](#) shows the real-time queue statistics in the Webex app for the Webex Customer Experience Essentials feature.

Images

Figure 6-10 Webex Customer Experience Essentials Live Queue Statistics

For many customers, the features included in Webex Calling provide all they need for their limited contact center requirements. For customers who need additional capabilities, Cisco offers several contact center products, such as Webex Contact Center, which are discussed in [Chapter 7, “Collaboration: Webex Contact Center and Webex Connect.”](#)

Interoperability

While many Webex Calling customers embrace the entire Webex suite of products, others choose to use other products for their messaging or meeting workloads. Webex Calling features allow it to interoperate with other platforms, such as Microsoft Teams. [Figure 6-11](#) shows how Webex Calling can be integrated into the Microsoft Teams client to provide a seamless calling experience from within the Microsoft Teams client.

Images



Figure 6-11 Webex Calling Integrated with Microsoft Teams Client

When the integration is enabled, users see a Cisco Call button in their Microsoft Teams application and can perform all calling actions as if they were using the Webex app. Call history, voicemail, and forwarding configuration are available, as are speed dials. When a call is active, a call window appears from the Webex app that is also running in the background on the user's machine, but the end user is not aware that they are using a

separate app when the call window shows up.

Webex Calling also provides a WebRTC client to allow for calling from a web browser. The client is provided as a Chrome browser extension, allowing Chrome web browser or ChromeOS device users to make use of native Webex Calling features without having to install an application (because ChromeOS does not allow for application installation beyond browser plug-ins). The plug-in also enables click-to-call functionality from any phone numbers found on a web page, making it easy to call a number right from the browser.

Additional integrations allow for embedding calling features into productivity applications. For example, the Webex app integration for Salesforce allows users to make and receive calls directly from a Salesforce record, increasing employee productivity. These types of integrations are made possible by SaaS platforms like Webex and Salesforce, providing open APIs that allow for extending the capabilities of one platform with another.

Analytics and Troubleshooting

Webex Calling provides various capabilities that allow administrators to gain insights into how their environment is performing, along with troubleshooting and diagnostic features to resolve potential problems.

We discussed the analytics features of Control Hub in [Chapter 5](#) when discussing the rest of the Webex platform, and these capabilities extend to Webex Calling. Control Hub has several analytics dashboards specific to calling features. For example, [Figure 6-12](#) shows an analytics dashboard for auto attendants in Webex Calling.

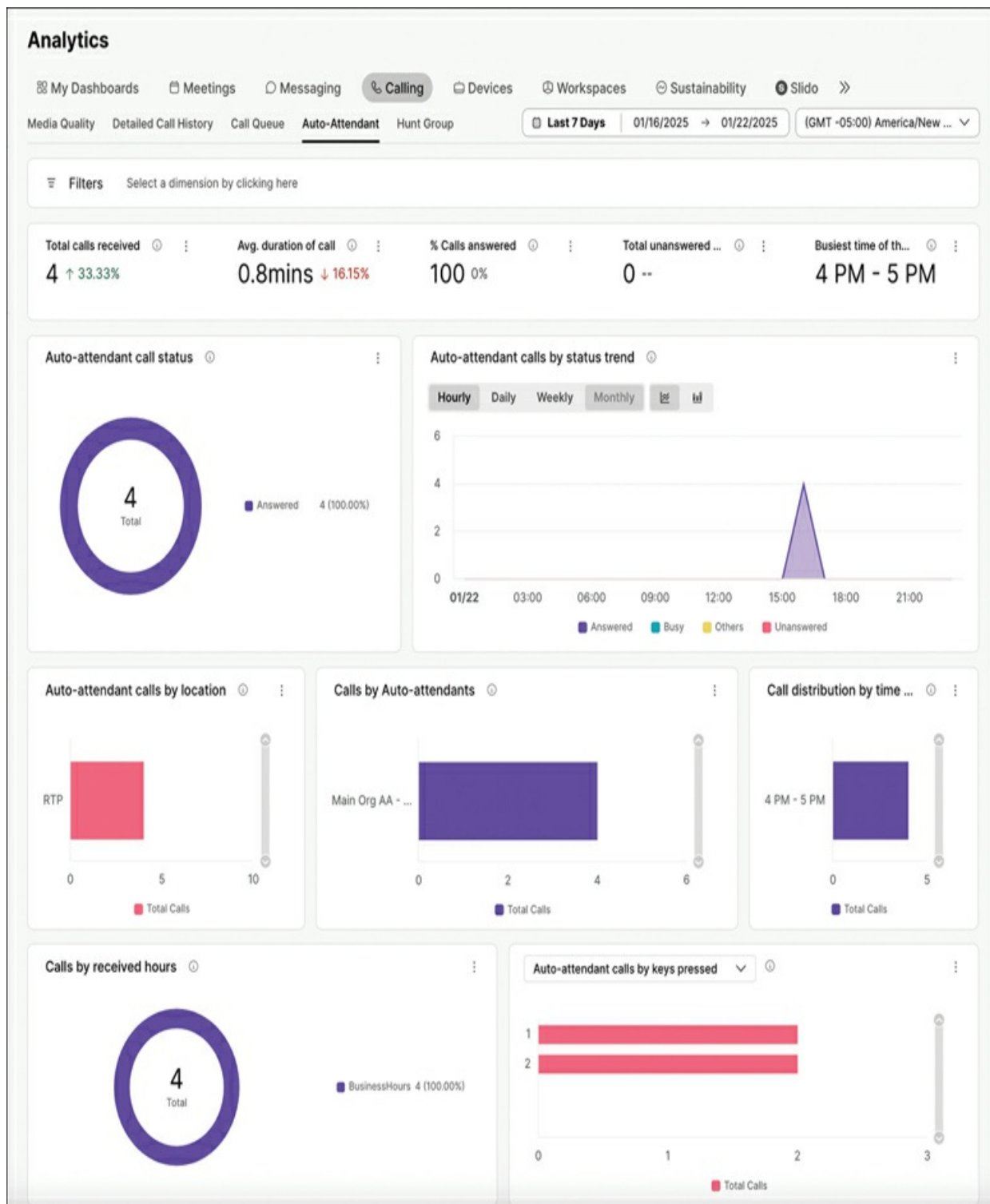


Figure 6-12 Analytics for Auto Attendants in Webex Calling

The analytics dashboard for auto attendants provides insights such as which auto attendants are busiest; which options callers tend to press more often;

and the times of day, day of week, and day of month that are busiest. Not shown in [Figure 6-12](#) is the list of all auto attendants and metrics for answered, unanswered, and busy call counts for each one, as well as answer rate and average call duration. These metrics can help administrators tune their configuration and get a better understanding of how users are leveraging their auto attendants.

In a traditional on-premises environment, a customer may have had to look at these statistics to determine whether to add more ports to their auto attendants, which would typically require additional hardware and licenses. In a SaaS environment, however, Cisco ensures there is sufficient capacity and scales up resources automatically in response to increased load.

Various Webex Calling features enable customers to troubleshoot problems with the service. Customers who choose to use a local gateway for PSTN connectivity need to be able to troubleshoot problems if they are having issues. Control Hub provides the status of the local gateways (either Online, Offline, or Impaired). If the local gateway is not online, an error reason indicates what the problem might be and provides details on how to resolve the issue. For example, [Figure 6-13](#) shows the Trunk Status page in Control Hub for a local gateway that is in an impaired state.

Images

Figure 6-13 Local Gateway Status in Control Hub

You can see that the status is impaired; a detailed message indicates “TLS connection from Local Gateway to Webex Calling failed as we were unable to trust the Certificate Authority.” This message provides a clear indication as to what the problem is, and the “Learn more” link takes the administrator to a page with actions they can take to resolve the problem.

Diagnosing call flow issues can be challenging, especially when calls involve calling features like hunt groups and call transfers. Control Hub provides extensive troubleshooting features that enable administrators to easily understand exactly what happened for a given call. [Figure 6-14](#) shows a calling troubleshooting page after searching for the phone number and selecting the call. In this example, you can easily see that a call arrived at a hunt group and was answered by the user Teller 2. The user then performed a consultative transfer to the Home Mortgage hunt group, which went unanswered and was forwarded on no answer to the Branch Managers hunt group, where Bob answered the call. Understanding this call flow from call detail records alone can take a significant amount of time and effort. The Troubleshooting page in Control Hub makes this task easy.

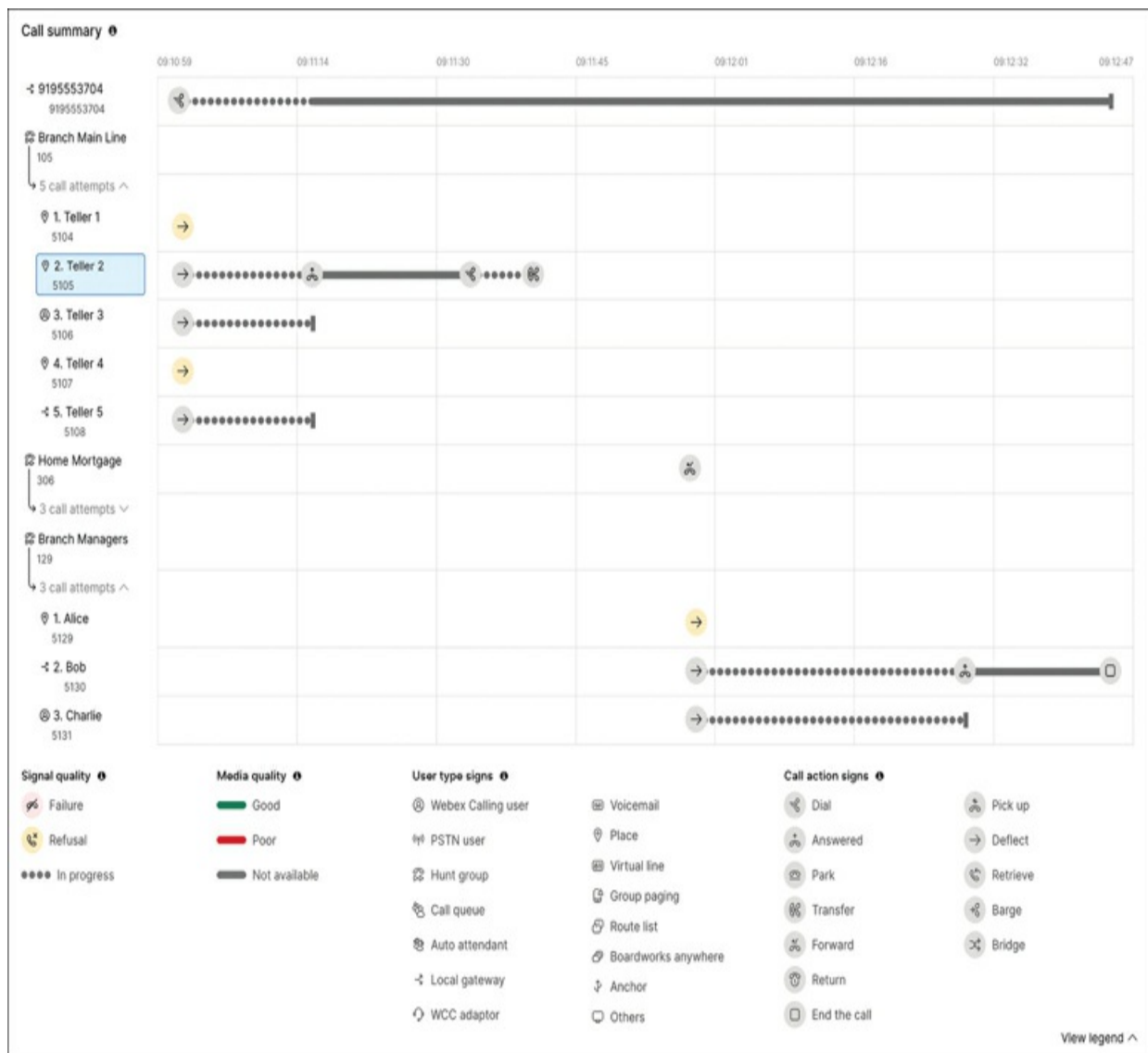


Figure 6-14 Webex Calling Call Flow Troubleshooting in Control Hub

Control Hub also provides details on calls to help diagnose media quality issues. For each call leg shown in [Figure 6-14](#), the administrator can select the call leg (in this example, the Teller 2 call leg is selected) and then scroll down to see the media quality details for that call leg. For example, [Figure 6-15](#) shows the details for a call as seen in Control Hub for the call leg between the inbound caller and Teller 2.

Images

Figure 6-15 Call Details for a Webex Calling Call in Control Hub

You can see that Control Hub provides details on the endpoints involved in the call and indicates the quality of the call for call legs from the endpoints to the cloud and from the cloud to the endpoints. Depending on what kind of endpoint is involved, some call legs might not show information.

High Availability and Survivability

Calling services, in general, must be highly available, but in some use cases, calling services are mission-critical and customers cannot afford any downtime. One challenge with SaaS services is the reliance on connectivity to the cloud for the service to function. IP phones and other endpoints must register with call services in the Webex cloud to be able to make and receive calls.

Because of the dependency on cloud connectivity, most customers have redundant paths to the Internet to ensure high availability. The Webex cloud hosts services across redundant data centers, as we will discuss in the “[Infrastructure](#)” section later in this chapter. For some customers, this amount of redundancy is still not sufficient. Some customers require that their phones continue to operate even when a site has no Internet connectivity. For these customers, Webex Calling provides the ability to host a local survivability gateway on the customer premises. This survivability gateway operates as a last resort for call processing capabilities if phones cannot communicate with the Webex cloud. The capabilities offered by the survivability gateway are limited; it does not offer the full feature set of Webex Calling, but it allows for the most important calling capabilities to be available while the Internet connectivity is down.

For users to be able to make and receive calls from the PSTN, the survivability gateway must have access to the PSTN. This typically means that customers using a survivability gateway are also using a local gateway for PSTN connectivity and the local gateway is co-located with the survivability gateway. In fact, in some environments, it is permitted to run both functions on the same device. Survivability gateways are assigned to a location, and all the user and device information from that location gets automatically synchronized daily with the survivability gateway.

When a user’s device is connected to the survivability gateway, the user is notified through a message such as the one in [Figure 6-16](#) where an IP phone indicates that there is a service interruption and some features may be unavailable. Other endpoints like the Webex app show a similar message, ensuring that users are not surprised by missing features while in this state.



Figure 6-16 IP Phone Indicating It Is Registered to a Survivability Gateway

The survivability gateway is like the Video Mesh nodes we discussed in [Chapter 5](#) in that these devices exist on the customer's network but are managed and provisioned from the cloud, operating as an extension of the SaaS platform.

Artificial Intelligence (AI) Features

Webex is constantly adding new AI-enabled capabilities, and Webex Calling is no exception. These features show the power of a SaaS platform by allowing AI processing to happen in the cloud as needed. Webex Calling allows calls to be automatically close captioned and transcribed. This capability can be useful for capturing a text record of a conversation. In addition to transcription, Webex Calling can also summarize a conversation, which is useful for capturing what was discussed on a call without needing a full transcript.

One of the more powerful AI features related to transcription and summarization is summarization on transfer. Imagine a caller speaks to someone at your company, and the person who received the call needs to transfer the call to someone else. Typically, this means that the caller must either repeat what they told the first person they spoke to or the person who is transferring the call must consult with the second party and brief them on

what they discussed with the caller. The summarization on transfer feature will create a summary of the conversation the caller had with the first person they spoke to and provide that summary to the person the call was transferred to. In this way, the person receiving the call has some context about what was discussed already, saving everyone involved valuable time.

Additional AI features will be added over time, providing more advanced analytics and diagnostics capabilities on the Webex Calling platform.

The Webex Calling Platform

Now that we have discussed the features of Webex Calling, we can take a similar approach as previous chapters to examine the platform and how it fits into the overall SaaS architectural model introduced back in [Chapter 2](#), “[SaaS Architectures](#),” and shown again here in [Figure 6-17](#).

Images

The diagram area is currently blank, showing only the placeholder text 'Images' in the top-left corner.

Figure 6-17 SaaS Architectural Model

As with previous chapters, the intention of this chapter is not to give you detailed information on the inner workings of Cisco’s cloud services but rather to give you an overall understanding of services needed to enable calling features of the Webex platform and specifically dig into the calling-related services.

Infrastructure

Webex Calling builds on the same Webex platform discussed in [Chapter 5](#). It makes use of the same hybrid multicloud architecture, leveraging both private

and public clouds; however, the introduction of PSTN-related services brings additional requirements to the infrastructure. Some data centers are dedicated specifically to Webex Calling–related services. [Figure 6-18](#) shows the map of Webex Calling data centers at the time of this writing.



Figure 6-18 Webex Calling Data Centers

Webex Calling is delivered from redundant data centers in six regions: U.S. (Dallas, Chicago), Canada (Vancouver, Toronto), Europe (Frankfurt, Amsterdam), UK (redundant data centers in London), Australia (Melbourne, Sydney), and Japan (Tokyo, Osaka). These data centers support call control and some media services. The data centers in New York and Singapore provide additional media services to optimize media roundtrip times. A customer tenant is assigned to a region, and signaling traffic for that customer is hosted by services in the redundant data centers for that region. All the data centers are interconnected through a high-speed, redundant network backbone used to transport both signaling and media traffic, removing the dependency on the public Internet for calls traversing between regions, thereby ensuring media quality.

[Figure 6-19](#) shows a high-level view of some of the services needed to

provide calling services within these data centers.

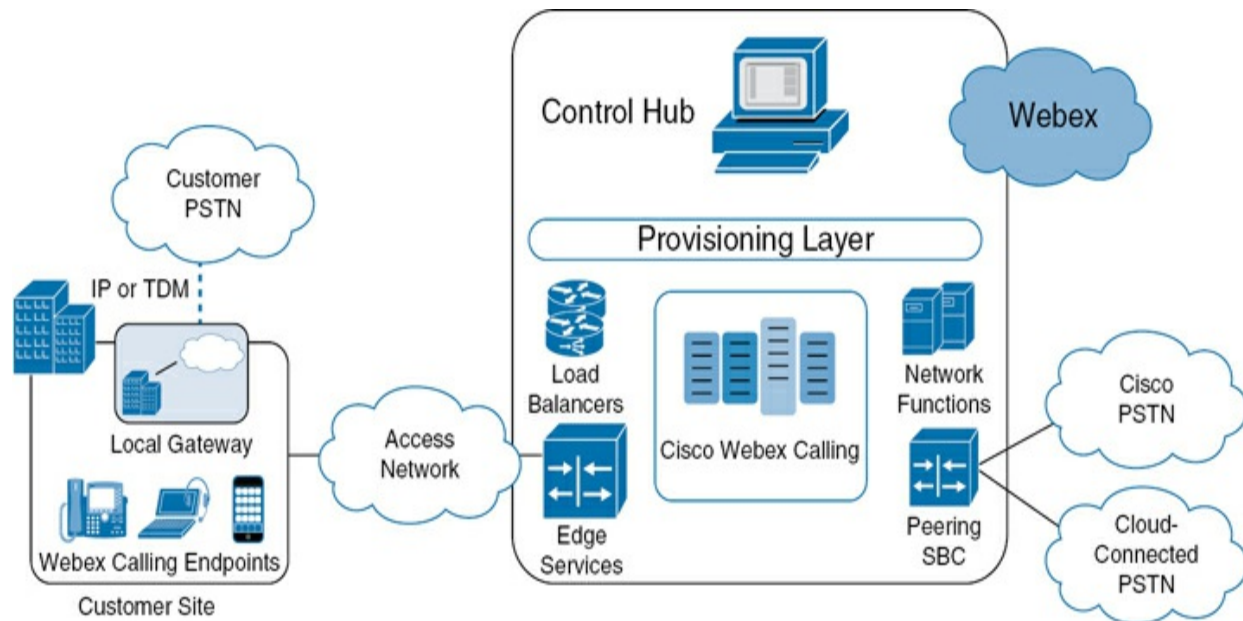


Figure 6-19 Webex Calling High-Level View

Notice that Webex Calling interconnects to the customer premises through a series of edge services and to PSTN providers through a peering session border controller. We will discuss these services later, but these connections necessitate infrastructure requirements for the peering connections. Most of these connections require private peering links with various telephony providers and are not transported over the public Internet. To facilitate such connectivity, peering points are needed. One common way to do this is by setting up infrastructure in co-location facilities where multiple providers manage infrastructure in the same location and then interconnect between their devices at that location. Webex Calling has a global presence and therefore requires peering connections with many service providers around the globe. Earlier in this chapter we discussed Cloud-Connected PSTN Partners. Each of these partners must have reliable, redundant connections into the Cisco data centers for these PSTN connections.

As with the media services in the Webex meetings platform, calling services require low-latency, high-bandwidth, highly reliable data centers to host the services that facilitate call setup and media termination. Webex Calling aims to achieve “5-nines” (or 99.999%) availability. This means less than 5 minutes of downtime per year. This is the same standard that traditional

telephone service providers aim for. To achieve these levels of availability, the infrastructure needs to be reliable, but the services also need to provide the ability to fail over to redundant services in other parts of the infrastructure should part of the infrastructure fail.

Webex Calling provides for three access methods: over the top via the Internet, Webex Edge Connect, or Private Network Connect (PNC). Most customers connect directly over the Internet to the nearest Webex Calling data center; however, larger customers may choose to use Webex Edge Connect or PNC for performance and reliability reasons because these options eliminate the unknowns of the public Internet.

Webex Edge Connect allows customers to peer their networks to the Webex network backbone through an Equinix Cloud Exchange (ECX) location. Webex has a presence in the ECX network, so a customer just needs to connect to an ECX location and set up peering with Webex through ECX. This approach provides guaranteed bandwidth and quality of service for the traffic.

Certain very large customers may choose to establish direct peering connections with the Webex cloud through the PNC service. In this case, redundant direct connections between Webex and the customer's network are provisioned to provide dedicated bandwidth and quality of service guarantees, like the capabilities provided through ECX. Customers may choose to mix these connectivity methods, depending on their needs. For example, customers may use a private connection for their local gateways connecting to their local PSTN connections but then use the public Internet for end-user connections.

Application Services

Webex Calling has many application services that handle the core functionality of routing calls, invoking calling features, supporting provisioning, and managing devices. These services comprise the bulk of the services dedicated to Webex Calling; many of the other services are shared with other Webex services already described in [Chapter 5](#).

- Call control services

- Edge services
- Media services
- Provisioning services
- Device management services

Call Control Services

Webex Calling has several services that are responsible for the core call control capabilities of the platform. These services perform tasks like dial plan resolution, media negotiation, mid-call calling features (e.g., transfer, hold, conference), music on hold, feature access codes, auto attendants, hunt groups, voicemail, call queues, and many others. These call control services must be highly redundant, scalable, and performant. The services for a customer are distributed across multiple Cisco data centers within a region to ensure high availability.

At its core, the call control services communicate using the Session Initiation Protocol for call signaling and media negotiation. Media negotiation is the act of exchanging media capabilities and IP addresses so that the endpoints involved in the call can send each other media using the Secure Real-Time Protocol (SRTP). SIP is defined in RFC 3261, along with many other related RFCs, and is recognized as the standard for most IP-based call signaling in the industry. While call control services facilitate the establishment of media, they are largely not involved in terminating media streams. The exceptions are the components that handle voicemail, auto attendants, and announcements or music played while on hold. The call control services use the Session Description Protocol (SDP) in SIP messages to negotiate media parameters between devices. Webex Calling endpoints then use SRTP to securely carry encrypted media between devices.

[Figure 6-20](#) shows how SIP is used to communicate between different components in Webex Calling. This figure is greatly simplified; many different services that exchange SIP signaling within the Webex Calling cloud to set up a call are not depicted.

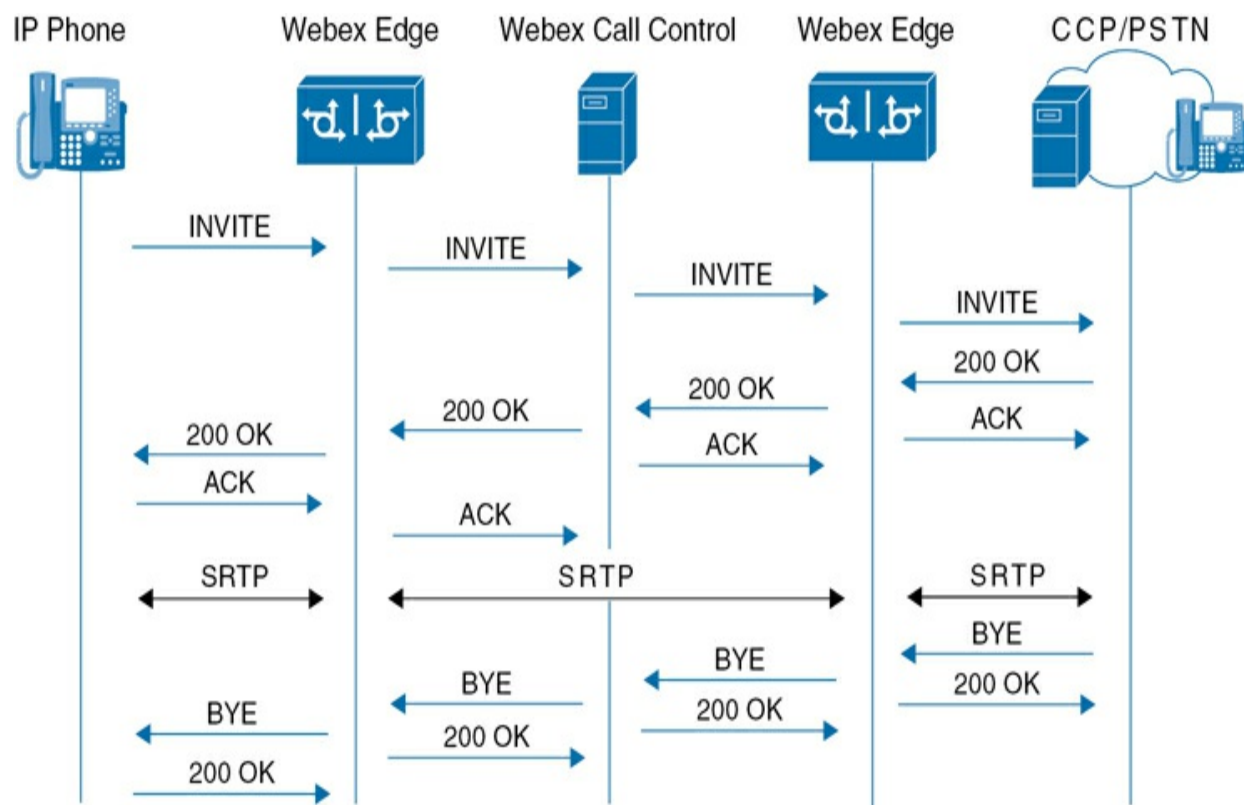


Figure 6-20 SIP Signaling

You can see that a call originates from an IP phone as a SIP INVITE message and is received by the Webex edge component to which that phone has registered. (Webex edge components are described in more detail in the next section.) Once the signaling has been verified by the edge, it is forwarded to the call control layer inside the cloud. Although only a single component is depicted here, calls will flow between various components in the cloud, depending on the features involved in the call and other dial plan considerations. Eventually, the call control needs to route the call out to a destination—in this case, to the PSTN—so it sends the call to the edge responsible for the terminating device. The edge then sends a SIP INVITE to the destination PSTN provider, whether it be Cisco PSTN, CCP, or a local gateway.

By default, media streams between two endpoints are anchored through the cloud. This means that the media from one device, like an IP phone, will be sent to and received from an edge node in the cloud. Notice that the SRTP media streams flow directly from edge to edge because the call control components are not involved in the media stream. The Interactive

Connectivity Establishment (ICE) protocol suite (which includes the Session Traversal Utilities for NAT [STUN] and Traversal Using Relay NAT [TURN] protocols) can be used to further optimize this media and potentially allow for the two endpoints in the call to directly exchange SRTP media streams without traversing the cloud at all, assuming the two endpoints have direct IP connectivity between each other. ICE not only provides a better, low-latency experience but also saves on bandwidth to and from the Internet.

The call control services are also responsible for producing call detail records (CDRs) that provide details of every call that is processed by the platform (both successful and unsuccessful calls). CDRs are published to customers via reports available in Control Hub as well as an API-based feed that customers can poll periodically to retrieve CDRs. We will discuss CDRs in more detail later in the “[Management and Analytics](#)” section of this chapter.

Edge Services

While the call control services are the “brains” of the Webex Calling platform, user devices generally do not communicate directly with the call control services. Instead, they rely on a series of edge services that handle signaling and media termination from end devices. We already touched on these services in the previous section when describing how signaling and media traverse the Webex Calling cloud.

For the most part, these edge services use SIP for signaling and SRTP for media. Webex Calling has separate services for the signaling and media termination components at the edge for scalability and redundancy purposes. This means that if you look at the traffic from a Webex client, the SIP signaling will terminate on one IP address while the SRTP media terminates on another. In this way, these services can be scaled correctly for the individual workloads. For example, media and signaling workloads impose unique requirements on the network and compute infrastructure. Media processing involves high network throughput but does not require significant amounts of processing of the media payloads. In contrast, signaling traffic uses significantly less bandwidth than media but is much more compute-intensive because of the need to process the signaling messages as well as the termination of TLS connections, which require the services to decrypt and encrypt the signaling data.

The edge services provide TLS connection termination and verification; security services, including denial-of-service protections; traffic load balancing; and high availability. Because these services terminate the media connections at the edge of the Webex network, they are also responsible for keeping track of statistics for the media connections that eventually get logged in to call media reports (CMRs) and exposed in the troubleshooting pages of Control Hub.

Edge services handle two distinct types of traffic: client device traffic, such as that from an IP phone or Webex app; and trunk traffic, which is used to connect to service providers or peer with on-premises infrastructure. The trunk-side edge services can be further subdivided between those serving connections from customers versus those terminating connections to other service providers or partners.

The edge services might generically be referred to as a session border controller function, but these services are built in a cloud-native microservices architecture in Webex Calling, which is different than a traditional SBC. Traditionally, an SBC is a purpose-built physical or virtual device used for terminating SIP sessions, but the Webex Calling edge services are built in a much more distributed, cloud-native fashion, allowing them to be scaled easily as demand for services increases.

Customer local gateway connections terminate on these edge services. Typically, a customer local gateway is provided with a DNS SRV record that contains four peering locations in the cloud—two in one data center and two in a second data center. The Webex cloud also establishes four connections from these peers back to the customer SBC for both high availability and scale.

Similarly, IP phones and other endpoints are given a list of edge services they can use to connect to the Webex cloud. The edge services terminate the SIP connections from all endpoints and relay the signaling data back to the core call control services.

Because every endpoint and trunk communicating with the Webex cloud must traverse the Webex edge services, these services are all highly scalable and distributed for redundancy. Their global presence also allows media to be anchored as close to an endpoint as possible, a crucial factor for maintaining

low-latency calling.

Media Services

A series of services within the Webex Calling cloud handles the termination of media. We have already covered the Webex edge services that terminate media in the cloud to relay it to other endpoints, but there are a few services in the cloud that serve as the endpoint for the media. They include interactive voice response (IVR), announcement, voicemail, and call recording services.

When Webex Calling determines a call is destined to an IVR, it sends the call to a service that is responsible for playing prompts, accepting user input, and then indicating the user preferences back to the call control layer. For example, if a user selects an option in an IVR that requires the call be transferred to another destination, this information is communicated back to the call control layer, and the call control layer uses SIP to reconnect the call to the appropriate destination (after applying any policies that may permit or deny that call). A SIP call typically passes DTMF digits dialed by a user in the media stream using a method specified in RFC 4733 (first introduced in RFC 2833, so often still referred to by the RFC 2833 name). This means the media layer must communicate the presence of these digits to the call control layers for processing when a digit is detected in the media stream.

Similarly, if Webex Calling needs to play an announcement to a user, the call control layer sends the call to a service responsible for announcements and then indicates which announcement needs to be played. This can include announcements like “The number you have dialed cannot be reached” or similar messages. This can also include music on hold when a caller is placed on hold. An administrator can customize the announcements by uploading audio files through Control Hub.

Another role of media services is for media mixing to handle multiparty conference calls. Webex Calling offers a series of services that can not only terminate media but also can intelligently mix the audio streams to allow for conference calling. These services can also be invoked when media needs to be forked to an additional destination, such as the recording or transcription services.

The last set of media services we will touch on are recording and

transcription services. When a call is configured to be recorded, the media streams for the participants in the call are forked to a call recording service responsible for keeping a copy of the media streams and saving them to persistent storage for retrieval. These recordings can then be sent to a variety of third-party recording services based on customer preference. The transcription services can also obtain a copy of the media stream to process them through an AI speech-to-text system to enable features like live closed captioning and transcription. The transcription can also be sent to additional AI services that can perform actions such as call summarization, both after a call and in real time.

Provisioning Services

Control Hub serves as the primary customer interface for provisioning of Webex Calling services. Customers can also use a variety of RESTful APIs documented on <https://developer.webex.com> to provision Webex Calling services that are like those discussed in [Chapter 5](#) for all other Webex services.

Once a customer has initiated some provisioning action through these public interfaces, several back-end provisioning services make the necessary changes to the various services that need to act on that provisioning request. For example, most Webex Calling provisioning requires notifying the call control services of the change for tasks such as adding new subscribers or numbers, changing dial plan, or modifying call policy.

Control Hub allows customers to manage provisioning of phone numbers through third-party cloud-connected PSTN services, as described earlier in this chapter. To provide for a seamless user experience, when a user makes provisioning changes on Control Hub related to a third-party service, the provisioning services send those requests to the appropriate third parties and return the results to the user. This task can include features like PSTN number/trunk provisioning and call recording provisioning through third-party recording providers. These provisioning services largely serve as API gateways with business logic to direct requests to the appropriate APIs implemented by these third parties in a secure, reliable way.

Device Management Services

Webex Calling supports both software-based clients like the Webex app and physical devices like IP phones and analog gateways. Physical devices like IP phones are supported by services dedicated to managing those devices. The following are some of the functions that the device management services provide:

- Onboarding
- Provisioning
- Firmware management
- Diagnostics

As mentioned earlier, IP phones can be provisioned either by MAC address or activation code. When a Cisco IP phone first boots up, it does not have a configuration and does not know how it has been provisioned. The phone contacts the activation service in the cloud to determine how it should proceed. The phone authenticates the cloud connection and provides its MAC address. If the MAC address has already been provisioned, it requests its configuration file information, checks for firmware updates, and then proceeds to register using a SIP REGISTER message to the call control services.

If the MAC address has not been provisioned by the administrator, the phone displays an activation screen prompting the user for an activation code. The user enters the activation code that the phone provides to the activation service. This process connects the phone's MAC address to the device assigned to that activation code and enters the MAC address into the provisioning service database. The registration process then proceeds as before, with the phone downloading its provisioning configuration, firmware updates, and finally registering with the call control services. These services allow a phone to easily be onboarded.

In addition to the onboarding and registration, the device management services maintain status and diagnostic information for devices. This information allows administrators to see the state of the devices, such as whether the device is online, is offline, or has issues. These services (along with some of the call control services) feed data to the management layer services to provide administrators with real-time information on the state of

devices in their network, as shown in [Figure 6-21](#).

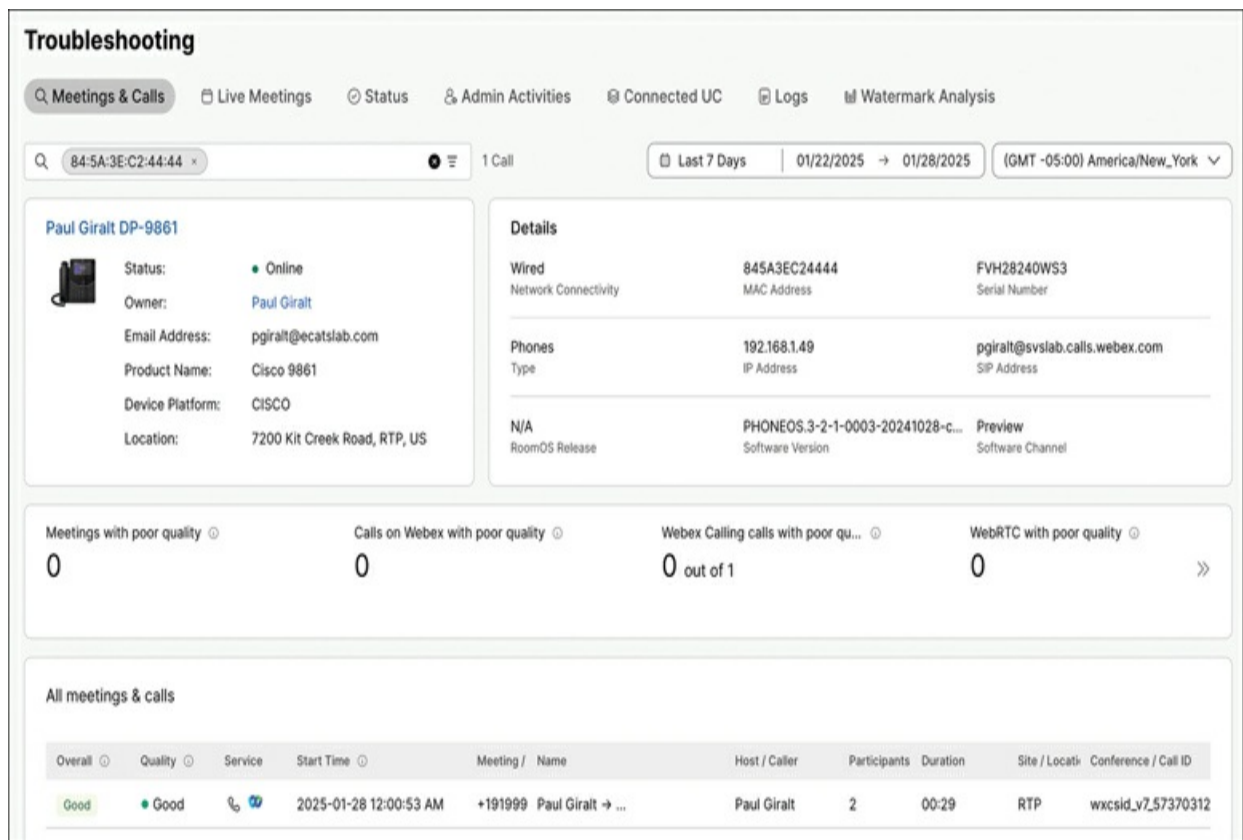


Figure 6-21 Real-Time IP Phone Information in Control Hub

The device management services also facilitate generation and retrieval of diagnostic logging data from devices if an administrator needs to download logs to troubleshoot a problem on a phone.

Presentation Services

The edge services described earlier could arguably be put into the category of presentation services because they are responsible for exposing the various call control capabilities to external devices; however, because they are such a core piece of the application layer for Webex Calling, we did not place them here. There are some additional user-facing services that can be categorized as presentation services.

Just as with other Webex services, the Webex app on desktop and mobile devices provides the end-user experience for Webex Calling, but users can

also use Webex Calling through physical devices such as IP phones. Although an IP phone is not what you would normally consider to be a cloud service, the firmware running on the phones is fully managed and provisioned by the cloud device management services and therefore becomes an extension of the cloud in many ways.

We previously discussed how Webex Calling can be used natively in a web browser through WebRTC. Several microservices responsible for exposing this WebRTC interface could also be presentation services.

As with the rest of the Webex platform, various RESTful APIs are responsible for providing many of the capabilities of the platform, such as device onboarding and provisioning.

Database Services

Webex Calling relies on some of the same database services as the rest of the Webex platform for capabilities such as user provisioning, logging, and analytics, as discussed in [Chapter 5](#), so we won't repeat them here. In addition to these services, other database services are used by the Webex Calling services.

Calling devices typically rely on a SIP REGISTER message to tell the call control platform information about where they are located on the network so they can receive and place calls using SIP. The calling platform must maintain a runtime database of these registrations to determine where to find an endpoint when trying to route a call to that device. The Webex Calling platform must maintain millions of these connections and must be able to look up the connection details quickly. As a result, in-memory database technology is used to store registration information for highly performant lookups.

Maintaining call detail records is a critical function of Webex Calling; therefore, it must store this data in a database that is not only performant but also highly available so that records are maintained even if the call processing services run into issues. Once the records are written by the call processing services, they are also sent downstream to the various analytics and management services to allow customers to retrieve them via reports or APIs.

Integration Services

Webex Calling is like the rest of the Webex platform in that it has an extensive set of open APIs. Webex Calling provides APIs not only for provisioning but also for placing and manipulating calls in real time. These services and APIs allow third-party integrations to exist, like the Salesforce integration mentioned earlier. Third parties can use the APIs to perform call control actions such as placing, answering, or ending a call. The APIs can also invoke more advanced features such as call recording, park, transfer, and conference, to name a few.

The provisioning-related APIs available in the Webex Calling platform allow not only for end-user provisioning changes but also for the ability to change dial plan configuration. In this way, administrators can potentially integrate dial plan provisioning into an automation framework that can be used for tasks like turning up a new site. You can find all the Webex Calling APIs at <https://developer.webex.com>.

Management and Analytics

As with the rest of the Webex platform, Webex Calling provides Control Hub as the primary management and analytics portal. Various services are responsible for providing this functionality, as discussed in [Chapter 5](#); however, some additional management and analytics services are specific to Webex Calling.

We already mentioned that the call control services generate call detail records for every call processed by the platform. The management and analytics layers are responsible for receiving the raw CDR data and processing them for user consumption. CDRs can be accessed either through RESTful APIs, in the Analytics section of Control Hub, or via reports that can be generated from Control Hub. The management services are responsible for combining the data from the CDRs generated by the call control services with the CMRs generated by the edge services to provide a holistic view of calls.

[Figure 6-22](#) shows the Detailed Call History tab in the Calling Analytics section of Control Hub. This tab provides key performance information

related to Webex Calling and then breaks down calling trends by type, location, and trends over time by date and hour of day. The page also displays a list of all the calls that have been placed and can be searched by calling or called number, the user who placed the call, or other filters like location, country, or whether the call was answered or not. This information allows administrators to quickly browse through their call data without needing to resort to a report or API access to the CDR data.

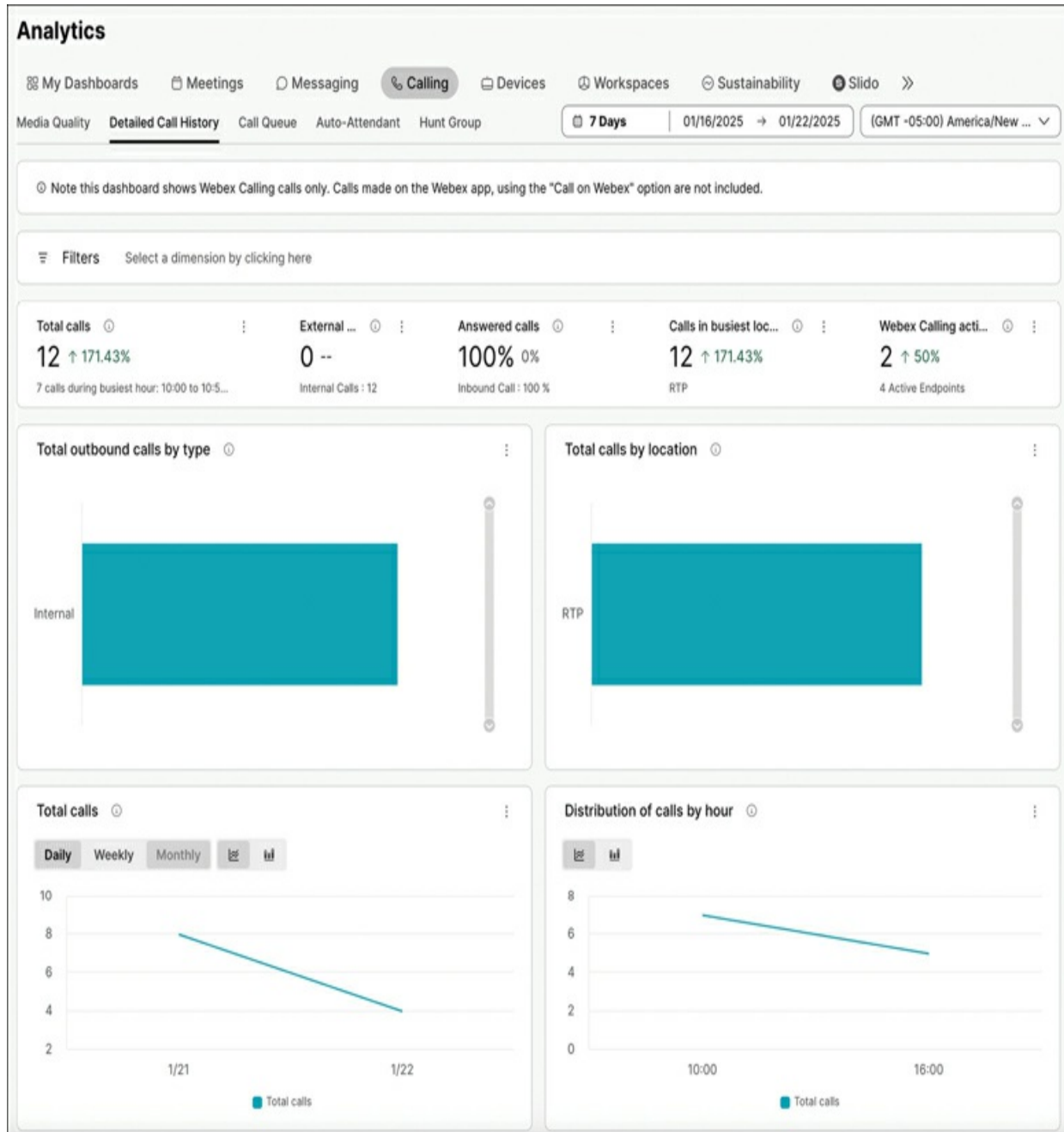


Figure 6-22 Detailed Call History in Control Hub Calling Analytics

Administrators need to be able to monitor and maintain the end devices they have deployed to provide calling services to their users.

Security and Privacy

A calling platform has certain unique security and compliance requirements. For example, toll fraud remains a challenge for any provider offering calling services. Webex Calling has automatic fraud detection mechanisms that can detect and stop call traffic that appears to be fraudulent.

Webex Calling provides granular control over calling permissions to ensure users are allowed to place calls only to destinations to which they are authorized. This includes controlling access to international calling or special toll numbers. Webex Calling has a detailed understanding of country dial plans to enforce these restrictions. For example, country code 1 is shared by the United States, Canada, and several countries in the Caribbean, so some international calls look like national calls when dialed from a phone in the U.S. and can enforce calling protections based on the specific area codes that are used in the different countries.

Webex Calling requires encryption for all its public interfaces, so SIP traffic is always encrypted over TLS, and media is always carried over SRTP. All HTTP-based interfaces are always carried over HTTPS.

As with the rest of the Webex platform, the network infrastructure for Webex Calling is protected by network security and monitoring services, including firewalling and DDoS protection. This protection is especially important because of the number of connections to other third-party services such as cloud calling partners.

Webex Calling manages the firmware of all connected devices, ensuring that any firmware-related defects or security vulnerabilities are patched in a timely manner, eliminating the need for administrators to manually manage firmware updates on their devices.

Like the rest of the Webex platform, Webex Calling conforms to a variety of industry standards including

- ISO 27001/27017/27018
- SOC2 Type II and SOC 3

Webex Calling also complies with the following privacy and compliance regulations:

- Health Insurance Portability and Accountability Act (HIPAA)
- General Data Protection Regulation (GDPR)
- PCI DSS v.3.2.1

Summary

Webex Calling extends the Webex platform beyond meetings and messaging to include a best-in-class cloud calling platform delivered through a SaaS model. In this chapter, we explored the extensive capabilities of the platform and the services required in the cloud to deliver these capabilities. Webex Calling allows customers to easily provide calling services to a global workforce without the need for on-premises infrastructure.

References

- Cisco Webex Calling:
<https://www.cisco.com/c/en/us/solutions/collaboration/webex-calling/index.html>
- Cisco calling plans for Webex Calling:
<https://www.webex.com/products/enterprise-calling-plans.html>
- Data center locations for Webex Calling: <https://help.webex.com/en-us/article/n70gvam/Data-Center-locations-for-Webex-Calling>

Chapter 7. Collaboration: Webex

Contact Center and Webex Connect

Have you ever had to call another company for support, or to inquire about a product, or perhaps to cancel your cable TV plan? During this call, you may have had to navigate your way through a phone tree (press 1 for sales, press 2 for support, and so on), and maybe you were even put on hold with an automated system telling you how many callers were ahead of you in the queue. This experience was likely orchestrated by a contact center solution. Contact center software allows for businesses to centrally manage customer communications through several different channels, including phone calls.

You may have heard the terms *call center* and *contact center*, possibly even used interchangeably. Although they are similar, they are not quite the same thing. In a call center, phone calls are handled by an organization, either inbound from customers or outbound through calling campaigns. All interactions through the call center are handled over the phone. A contact center, however, is a platform that supports multiple modalities for customer communication. This communication could include voice calls but also expands to other methods such as email, SMS, and chat clients, just to name a few. Therefore, a call center can be considered a specialized subset of a contact center, focused solely on voice interactions.

In the late 1800s, calls were routed via switchboard operators who answered and routed calls to the appropriate destination by plugging in phone cords to the right jack, as seen in [Figure 7-1](#).



Figure 7-1 Switchboard Operator Routing Calls

It wasn't until the 1950s that automatic call distribution (ACD) systems emerged, which many consider to be the technology catalyst that marked the starting point for modern contact center solutions today. ACD systems allowed incoming calls to be automatically routed to agents. This technology allowed companies to scale incoming call volume at a much greater capacity without needing a human to answer each call and route it appropriately.

In the 1960s, a UK-based company named Birmingham Press and Mail installed a General Electric Company (GEC) Private Automated Business Exchange (PABX) 4 system, which is considered one of the first call center solutions to be used by a company in the UK. By the 1970s, companies were beginning to include ACD technology in PABX systems, giving way to larger-scale call centers.

In 1993 a company named GeoTel was founded. GeoTel developed software solutions for call routing, primarily used in contact center solutions for both enterprise and service provider customers. GeoTel's software was mainly focused on Computer Telephony Integration (CTI). CTI enabled customers to leverage data from the network to enhance phone interactions. For example, through a CTI application, you could pull up data from a database about a customer based on caller ID. In 1999 Cisco acquired GeoTel for \$2 billion to accelerate its voice over IP (VoIP) business. This acquisition laid the groundwork for what would become Cisco's first contact center software product called Cisco Intelligent Contact Management (ICM).

In 2007 Cisco also acquired Webex, which was Cisco's first entrance into the

SaaS market for collaboration. The introduction in [Chapter 5, “Collaboration: Webex Meetings and Messaging,”](#) covers the history and acquisition of Webex in more detail. Webex enabled Cisco to enter the SaaS market with a product that allowed for meetings, messaging, and calling all as a hosted cloud service. This meant that companies would not have to manage on-premises servers but instead could pay for a subscription that would grant them access to these services. However, at the time, Webex did not have a contact center solution. This situation changed in 2018 when Cisco made another acquisition, Broadsoft, which specifically accelerated its cloud calling contact center business. This Broadsoft acquisition was the starting point of Cisco’s entrance into the cloud contact center offer. Since then, Cisco has made several other acquisitions to strengthen its cloud contact center business, including CloudCherry and IMImobile.

In this chapter, we will explore the capabilities of Cisco’s Webex Contact Center platform, details about its SaaS architecture, and how this solution has evolved over time.

Cisco Cloud Contact Center Products

Cisco Cloud Contact Center is really broken up into two distinct product offerings: Webex Contact Center and Webex Contact Center Enterprise. Webex Contact Center is the primary offering for most businesses of varying size and works well for small- to medium-sized deployments. Webex Contact Center Enterprise, on the other hand, is targeted specifically for large-scale contact center deployments where larger capacity and even more advanced call routing capabilities are needed. Webex Contact Center Enterprise is capable of scaling up to 36,000 concurrent agents.

Another unique thing about Webex Contact Center Enterprise is that it is supported as both a SaaS offering hosted in Webex’s multitenant cloud, or it can also be hosted in Webex Dedicated Instance (DI), which is Webex’s Infrastructure-as-a-Service (IaaS) offering for calling and unified communications (UC). Webex DI gives customers a dedicated cloud instance for hosting their UC infrastructure and makes transitioning from on-premises systems like Cisco Unified Communications Manager (CUCM) to the cloud seamless. It is essentially like having your CUCM instance moved from on-premises hardware to running in the cloud for simplified maintenance and

support. A Webex DI deployment for Webex Contact Center Enterprise means that you will have more support for customization and control over the platform, but you will not have feature parity and speed of feature development as you will with a SaaS offering for Webex Contact Center Enterprise. Because Webex DI allows for greater customization and control, this means that software updates are less frequent than those of multitenant SaaS offerings, which results in there not being full feature parity between the platforms. However, you do gain greater control and flexibility over scaling of the infrastructure; plus, managing the services in Webex DI should be much more familiar to users who are used to managing traditional on-premises collaboration services. Finally, there is Webex Contact Center Enterprise for Government, which is a FedRAMP-approved deployment of Webex Contact Center Enterprise, allowing it to be used for federal and state customers that require stricter security policies. Webex for Government is hosted in a separate cloud instance that meets stricter security regulations and requirements than traditional Webex deployments at all levels of the offering—from product requirements to hardening of the hosting environment and everything in between. Webex for Government deployments do not have all the same features and capabilities as a traditional Webex Contact Center deployment, because not all features and capabilities are FedRAMP approved. However, as new features are developed and eventually pass specific security requirements, they can be introduced to the government platform later.

In this chapter, we will primarily focus on Webex Contact Center and the SaaS offering and not dive into much detail about Webex Contact Center Enterprise, Dedicated Instance, or Webex for Government.

Product Capabilities

Before we jump into the SaaS make-up of Webex Contact Center, it would probably be helpful to first explain the capabilities of this solution and then afterward discuss the architecture and components that make this a SaaS product.

Webex Contact Center is a cloud-native communications platform that allows for companies to connect with customers through several different channels, including voice, SMS, chat, and even social media platforms. It is an as-a-

Service platform, often referred to as Contact Center as a Service (CCaaS), not to be confused with Communication Platform as a Service (CPaaS). CPaaS is a fully programmable platform that allows you to build quick applications to handle customer interactions across many digital platforms. Think of CPaaS as a suite of tools to develop and launch real-time communications capabilities into existing services or applications. It can do this by exposing APIs and supporting SDKs that make it simple to integrate and embed communications services practically anywhere. Both CCaaS and CPaaS are SaaS products and, in the context of Webex, are closely related and often integrated. For example, it is common for Webex CCaaS to leverage Webex CPaaS (specifically Webex Connect) for omnichannel communication.

The following are some of the features and capabilities of a Webex Contact Center deployment:

- Omnichannel communication (voice, email, chat, SMS, social media)
- Advanced call routing
- AI-powered chatbot and virtual assistants
- Analytics and reporting
- Workforce Optimization
- Integrations (CRM, Webex meetings, messaging and calling, APIs)
- Agent desktop application
- Call recording and music on hold
- Self-service interactive voice response (IVR)
- Outbound calling campaigns

We will not cover all these capabilities here but will cover some of the most important ones. First, we will explore some of the capabilities of the Webex Contact Center platform and then dig deeper into the services required to create such a platform and how those services align to the SaaS principles outlined in [Chapters 1 through 4](#).

Omnichannel Communication

When you think about the core component of a contact center solution, it is communication. That communication can take many different forms. Perhaps the most well-known form of communication in a contact center solution is a phone call. The reason for this call could be customers calling in for support or to purchase a product, as an example. But with Webex Contact Center omnichannel capabilities, customers can select how they want to communicate, whether that is chat, text, call, email, or social media.

This omnichannel capability is possible because of Webex's CPaaS offering called Webex Connect. Webex Connect came from Cisco's acquisition of IMI Mobile in December 2020. IMI Mobile was a cloud communications software platform that enabled organizations to connect with clients through various interactive channels such as social media, text message, and voice. Through the acquisition of IMI Mobile, Cisco was able to expand its Webex Contact Center capabilities by now offering this omnichannel communication capability.

Webex Connect integration with Webex Contact Center makes it seamless for you to configure contact flows that integrate various digital communication methods to allow for simplified customer self-service and support options within your contact center business. It supports digital channels such as Facebook Messenger, SMS, live chat, email, WhatsApp, and Apple Messages for Business so that no matter what platform your customer is on, you can engage with them.

What makes omnichannel communication so important? And is it just a buzzword that means you can talk to your customers in several different ways? And hasn't that capability already existed in the contact center space for a while already? The key differentiation between omnichannel communication and what many other vendors provide—which you could call multichannel communication—is that with omnichannel communication all communication is centrally managed and accessible. This means that if a customer begins their communication with you over text and that communication later progresses to a phone call, the agent working with this client does not lose the communication that happened over text. The agent would retain all the communication history, regardless of what channel was used.

From a customer experience perspective, this situation is much more ideal.

As a customer, nothing is more frustrating than having to repeat yourself every time you get transferred from one system or agent to another. Webex Contact Center's omnichannel communication allows contact center agents to retain that communication history over the lifecycle of the customer's dialogue, across various communication channels, giving the customer a much smoother and less frustrating experience.

Omnichannel communications are all brought together for contact center agents through the Webex Contact Center Agent Desktop application, as shown in [Figure 7-2](#).

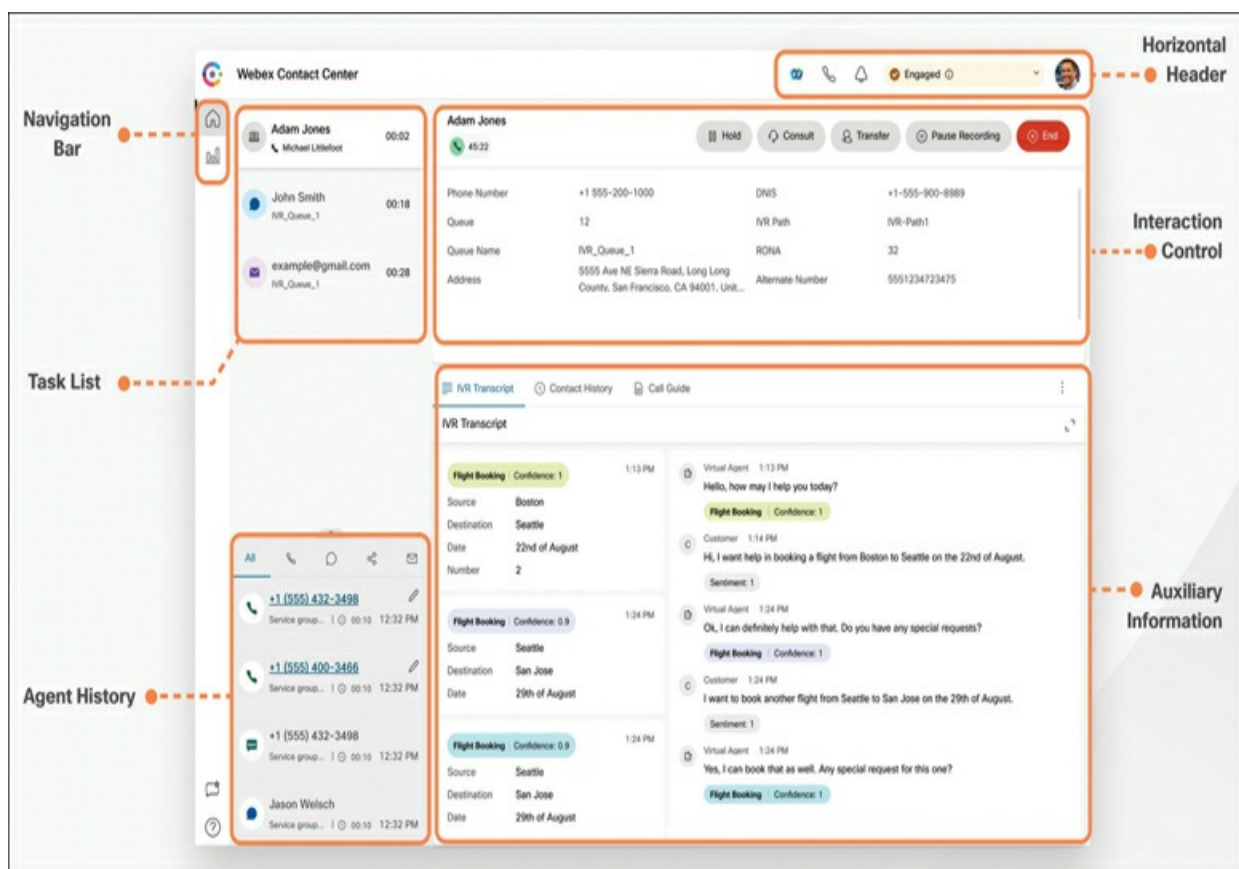


Figure 7-2 Webex Contact Center Agent Desktop Application

The client shown in this figure brings together the communication history for a user in a single application, enabling support agents to understand the customer's journey up to that point. That could mean understanding how long a contact waited on an IVR, what communication happened in the IVR chat, whether the IVR interaction resulted in a call back or the contact dropped out, any previous communication the contact may have had with another agent

before being moved to a new agent, and more.

AI-Powered Features

AI is transforming the way that businesses deliver services to customers, and within the Webex Contact Center space, it is no different. AI is used to provide many different capabilities, such as these:

- Intelligent chatbot communication
- Speech-to-text transcription
- Automated note taking
- Action item follow-up
- Customer insights
- AI Assistant to help agents answer customers' questions
- Custom-developed solutions using AI and Webex APIs

Virtual Agents

Possibly one of the most practical uses of AI in Webex Contact Center solutions is in virtual agents. Virtual agents allow you to orchestrate and automate interactions between customers and your organization. These interactions could come in a few different forms. It could be a chatbot that is used to triage and possibly solve basic questions and tasks. Or it could be a virtual voice assistant that is used to do the same thing as a chatbot but over a phone call. Virtual assistants are essential tools to allow for a contact center deployment to scale without having to add more human agents. By eliminating initial call routing, triage, phone trees, and questions using a virtual agent, human agents can then focus on helping customers with more challenging and potentially time-consuming tasks.

Virtual agents are designed using Dialogflow CX, which is a platform offered in Google Cloud to design and develop conversational interfaces. An example of a Dialogflow flow is shown in [Figure 7-3](#).

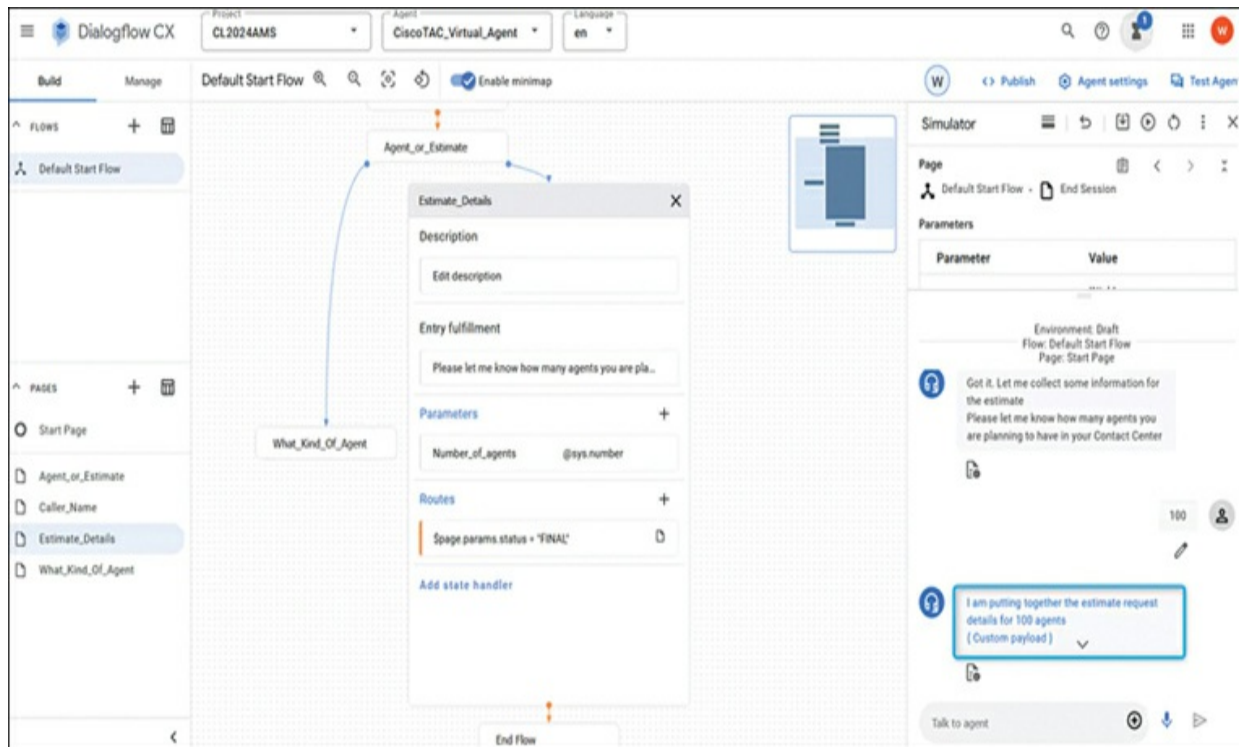


Figure 7-3 Dialogflow CX Configuration for a Virtual Agent

Dialogflow leverages AI and natural language processing to determine intent and supports like audio or text. This means that a Dialogflow design can be used in both chat interfaces, and through a phone interaction such as an IVR phone system. Dialogflows are built in a graphical interface that allows you to visualize the conversational flow and configure the actions to take.

Cisco AI Assistant

Cisco's AI Assistant is also integrated with Webex Contact Center. Agents can utilize it to assist with tasks such as call transfer summaries, automated action follow-up, or note taking during a call. Each of these actions helps contact center agents focus on customers instead of trying to do multiple tasks at once. Imagine a call that is being transferred to you from another agent, and although you may have access to the transcript of the call that is being transferred to you, having a way to quickly summarize the transcript enables you to be fully informed when you take the call. [Figure 7-4](#) shows an example of how the Cisco AI Assistant can provide call transfer summaries to do exactly that.

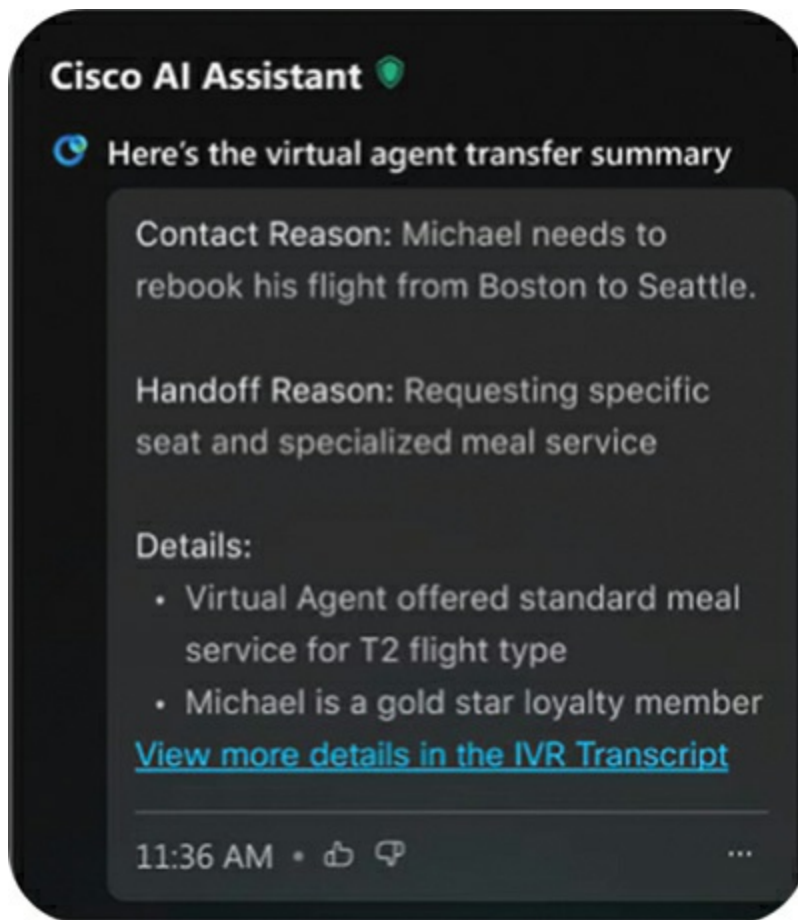


Figure 7-4 Cisco AI Assistant Virtual Agent Transfer Summary

In this example, you can see that the IVR transcript was provided by a virtual agent. That means this call was originally handled through a phone call that a customer placed, and the initial dialogue with that customer was handled by a virtual agent. The details from that interaction were then summarized and sent to the human agent when the call was transferred.

Advanced Call Routing

You now know that Webex Contact Center is capable of omnichannel communication, allowing you to handle conversations from customers through various channels all within a single interface. You also know some of the AI capabilities that are supported and how they can simplify and improve customer experience. But how do these different inbound communication channels get routed to the right virtual agent or human agent?

A contact center solution must be able to provide advanced levels of call routing functionality to be able to route calls to the appropriate destination, handle large volumes of traffic, determine how to distribute call loads based on staffing and shift schedules, handle off-hour communications, and distribute calls based on skill and availability. As you can imagine, this task can get complex quickly. Although we will not dive deep into the configurations necessary to deploy and manage a Webex Contact Center deployment, we will cover the basic concepts to help you understand how the system handles call routing.

Before we dive into the call routing specifics, let's look at some of the core terminology of a Webex Contact Center deployment.

A Webex Contact Center tenant is an enterprise that has contact centers at one or more sites. The enterprise also has entry points for incoming contacts that are associated with queues. Incoming contacts can be toll-free numbers for voice calls, designated email addresses for emails, or chats with agents. For example, an enterprise that is named NetCorp might have an entry point that is named Welcome. Welcome classifies contacts into NetCorpSupport and distributes to teams of agents in San Jose, London, and Sydney.

Each Webex Contact Center tenant profile consists of sites, teams, entry points, and queues:

- A *site* is a physical contact center location under the control of the enterprise or an outsourcer. For example, Acme might have sites in San Jose, London, and Sydney.
- A *team* is a group of agents at a specific site who handle a particular type of contact. For example, NetCorp might have teams at their San Jose site that are named SJC_Support, SJC_CustomerSuccess, and SJC_AccountManagement, and teams at their Sydney site named Syd_Support, Syd-AccountManagement, and Syd-TechnicalSupport. Agents can be assigned to more than one team, but an agent can service only one team at a time.
- An *entry point* is the initial landing place for the customer contacts on the Webex Contact Center system. For the voice contacts, typically one or more toll-free or dial numbers are associated with an entry point. IVR call treatment is performed while a call is in the entry point.

- A *queue* is where active contacts are kept while they await handling by an agent. Contacts are moved from the entry point into a queue and are distributed to agents.

Figure 7-5 illustrates how these concepts are organized within a Webex Contact Center deployment. Imagine that a customer calls into NetCorp. That call comes in through an entry point configured on the tenant. This entry point is pointed to the support teams. Based on the time of day and what sites are currently active, the call is then routed to a support team at a site that is currently active, and the call is accepted by an agent on one of those teams. The queue is used as a holding place for contacts to wait for an agent to become available.

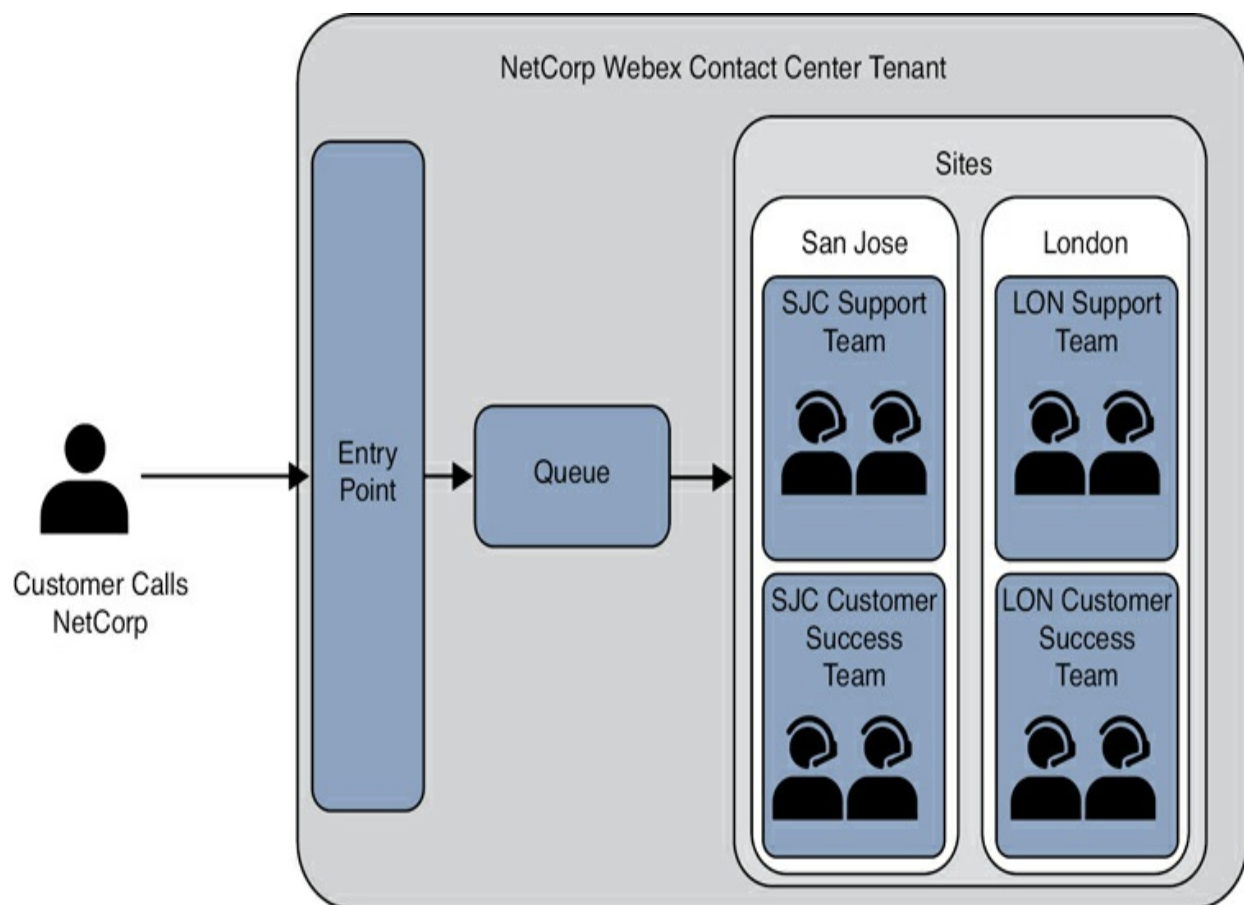


Figure 7-5 Organization of Sites, Teams, Entry Points, and Queues in a Webex Contact Center Tenant

What are the different strategies that can be used to route a call to an agent? When a call hits an entry point, before it is forwarded to an agent, it is first

evaluated by a routing strategy. A routing strategy allows you to configure more discrete parameters for how you want the contact to route to an agent. There are two primary contact routing options that an administrator can configure: skills-based routing and agent-based routing.

Skills-Based Routing

Skills-based routing is a routing option that allows you to try to map the needs of the contact with an agent who has the skills to handle those needs. Continuing with the example provided in [Figure 7-5](#), say that the customer called into NetCorp and wanted some assistance with support. With skills-based routing, the routing algorithm would look at available agents who are skilled with “support” and then route the call to the appropriate queue to get to those agents. You could also leverage skills-based routing to help with factors such as language preferences to ensure that the agent to whom the call gets routed can speak the language of the contact. Skills-based routing has two methods for routing contacts to an agent: longest available agent and best available agent.

The longest available agent routing strategy routes contacts to the agent who has been available for the longest. The goal of this routing strategy is to ensure that no one agent is getting longer breaks between contacts than others. However, it does not mean that the contact distribution will be even. If an agent handles contacts much faster than other agents, that agent may become available sooner than other agents and might consequentially also receive more contacts routed to them.

The best available agent routing strategy routes contact to the agents with the highest skill proficiency that matches the contact’s needs. To do this, obviously each agent needs to be appropriately skilled in their agent profiles for the routing to work correctly.

Agent-Based Routing

Agent-based routing, unlike skills-based routing, attempts to route contacts to a preferred agent directly, instead of considering a queue of individuals and choosing the agent based on some routing methodology to distribute the load. This type of routing works by mapping a contact with an agent. The mapping

is not done directly in Webex Contact Center but rather is managed in an external application that you can integrate with your Webex Contact Center environment. This task is usually handled by an integration with some type of customer relationship management (CRM) software because this is often the source of truth for customer account information, allowing Webex Contact Center to leverage existing platforms over an API to derive the proper mapping of customer to agent for routing a call.

When using an agent-based routing strategy in a contact flow, the system can make an HTTP request to this external service to look up any agent mappings to that contact. If there is a match, the contact can be directly routed to that agent. If, for some reason, the agent that the contact is mapped to is not immediately available, there are also ways to queue that call with that agent to wait for them to become available before routing the contact to them.

One example of how agent-based routing can be configured is to make it so that if a customer is reaching out to a contact center multiple times, that contact could be routed to the agent who handled their contact previously to ensure that the customer has a better experience and has more familiarity with that user.

The routing strategy that is selected can have a major impact on how agents receive contacts from customers and how the contacts are distributed to different queues and agents. Although we did not cover all the different routing configurations available in this section, you should now have a better idea of the options that are available in a Webex Contact Center configuration.

The analytics and reporting capabilities available in a Webex Contact Center deployment are key to ensuring that the configuration that is used in a deployment is effective. By maintaining information on contacts coming in and how they are distributed across teams, you can ensure that an efficient and effective strategy is in place or make modifications based on what the metrics are telling you.

Flow Designer

Flow Designer is a web tool that enables administrators to configure and

manage how contacts are handled once they enter the contact center environment. The interface is a drag-and-drop utility that allows you to quickly build and visualize a flow and execution steps. Flow Designer is a powerful utility that has many built-in components and activities that you can drag and drop onto the flow canvas, connect with other components or activities, and define and set variables for use in the flow, among many other things. [Figure 7-6](#) shows an example of a flow in the Flow Designer interface.

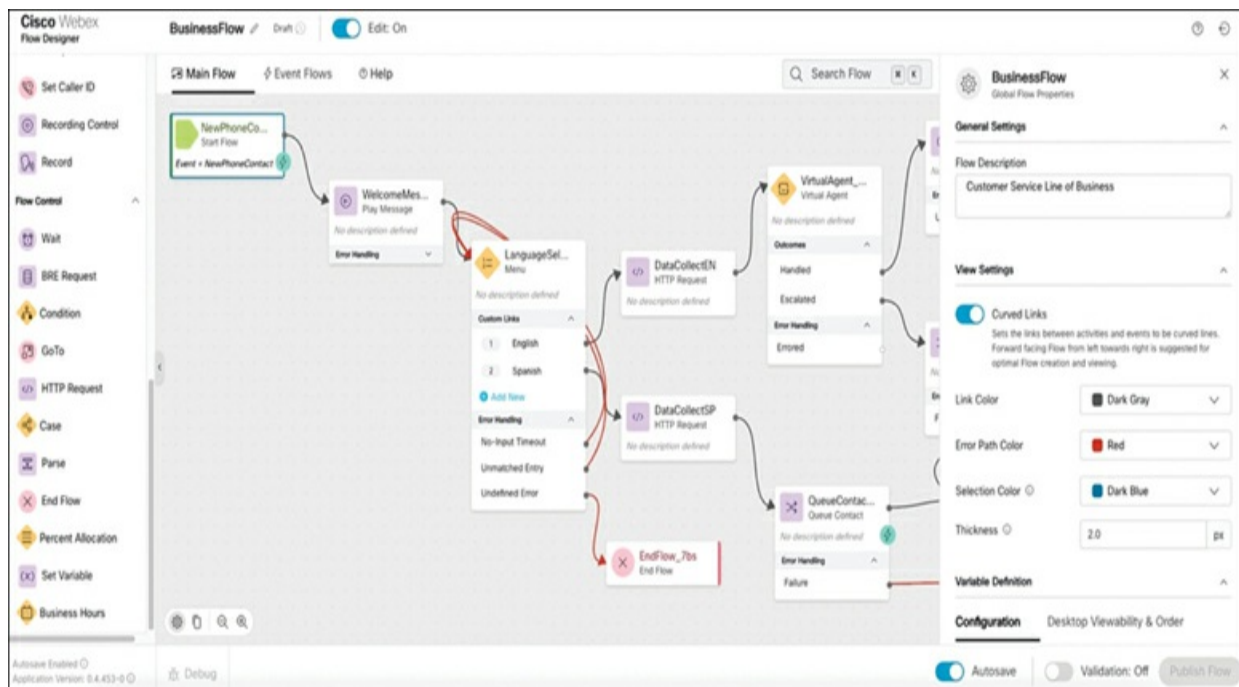


Figure 7-6 Flow in Flow Designer

When you're designing a flow, you really need to understand four key components:

- **Activity:** A single step in a flow. Each box in the flow in [Figure 7-6](#) is an activity. For example, this could be to collect digits from a user or queue the contact to an agent.
- **Event:** Something that causes the flow to be executed. This could be an HTTP event (API call) or user input from an IVR.
- **Flow:** A sequence of activities that get executed.
- **Link:** A bond that connects activities together. In Flow Designer, links

are literally arrows between activities.

Flow Designer supports conditionals to allow for complex activity flows based on varying types of events or inputs that the flow might receive. For example, when you call into an IVR and it gives you a list of options for which teams you want to talk to, depending on the digit you press on your phone, your call will be routed to that specific team. You could use conditional logic in Flow Designer to account for each of the possible digit inputs from a user or define a catchall should the user input a digit that you were not expecting.

Some inputs you receive from customers may be sensitive, or personally identifiable information (PII), so you would not want to expose this data. Flow Designer also supports storing information in a secure variable, which prevents any logging or storing of information related to that variable. The data in that variable is available only at runtime to the flow but is not accessible later.

To help users get started for the first time, Flow Designer also has several different flow templates available. They give you a basic flow that is prebuilt, and you can modify and repurpose it to meet your businesses goals.

Another helpful feature of Flow Designer is that it allows for subflows to be created and reused by other flows. Let's say you want to reduce a single large flow into a set of subflows and then string those together in a smaller flow to reduce the size and visual complexity of a single flow, or perhaps there is a portion of your flow that you consistently use in many different flows that you design. You could modularize your flows such that common tasks are developed as subflows and become available for reuse by your organization; that way, you can avoid having to repeat common flows.

Analytics and Reporting

Among the great things about most SaaS platforms are the analytics and reporting capabilities that you get out of the box. Webex Contact Center is no different in this regard. Through the Webex Contact Center administration portal, you can choose from multiple options for viewing data about a Webex Contact Center deployment. The primary tool used for delivering analytics is

called Analyzer.

Administrators of a Webex Contact Center tenant can log in to Webex Control Hub to view the details and configuration related to a deployment. From Webex Control Hub, an admin user can cross-launch over to Analyzer. Analyzer is a web-based analytics platform developed specifically for Webex Contact Center; it supports both off-the-shelf dashboards and telemetry, as well as the ability to build custom reports and visualizations. [Figure 7-7](#) shows what the Analyzer interface looks like. The following are some examples of the stock reports available in Analyzer:

- Agent reports
- Auxiliary reports
- Business metrics
- Callback reports
- Contact center overview
- Multimedia reports
- My Team and queue stats
- Transition reports

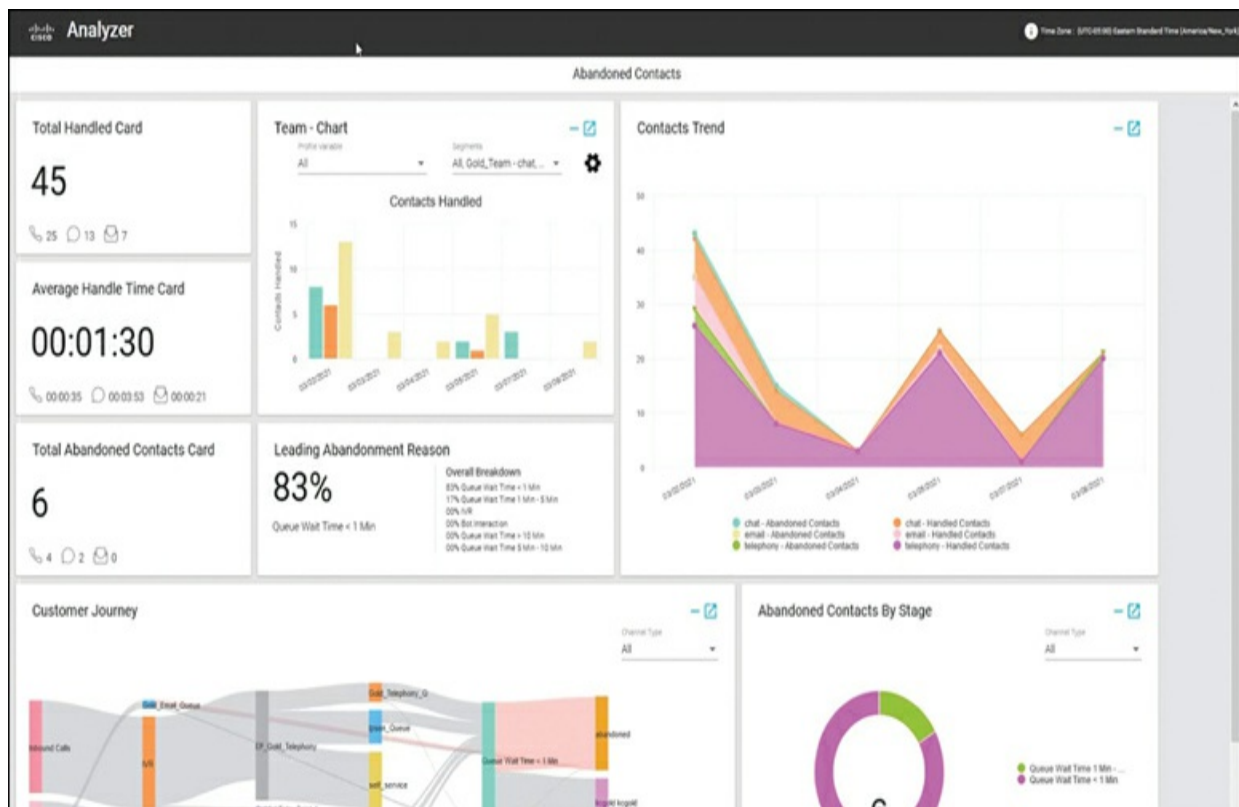


Figure 7-7 Analyzer Dashboard for a Webex Contact Center Deployment

These reports make it quick and easy for you to understand the performance, efficiency, customer sentiment, and overall queue management of a contact center deployment. If one of these reports does not show the detail that you're looking for, you can generate a custom report within Analyzer and can also share that report with other users.

In addition to the web interface, Analyzer has a set of APIs available to it to allow you to integrate with other systems and tools. Specifically, a Search API is a GraphQL (Graph Query Language) endpoint that enables you to fetch telemetry from a Webex Contact Center deployment. This API exposes all the fields that are available in the Analyzer dashboard, supporting both historical and real-time data collection. You can find more details on this Search API at this Webex Contact Center developer site:

<https://developer.webex-cx.com/documentation/guides/getting-started-with-search-api>.

Another option for viewing reports is directly through the agent desktop client. If configured, agent performance reports can be exposed directly on

the desktop client, allowing agents to see real-time metrics regarding their performance, their team's performance, or even queue statistics. These reports enable agents to have a quick view into their work statistics, such as total contacts handled, average response times, and types of contacts handled, and compare that information to their team's to understand how many contacts others on the team have handled over the same period. These reports can be customized and developed in the Analyzer web interface and then exposed on the agent desktop. [Figure 7-8](#) shows what the agent performance metrics view looks like within the Webex Contact Center desktop client.

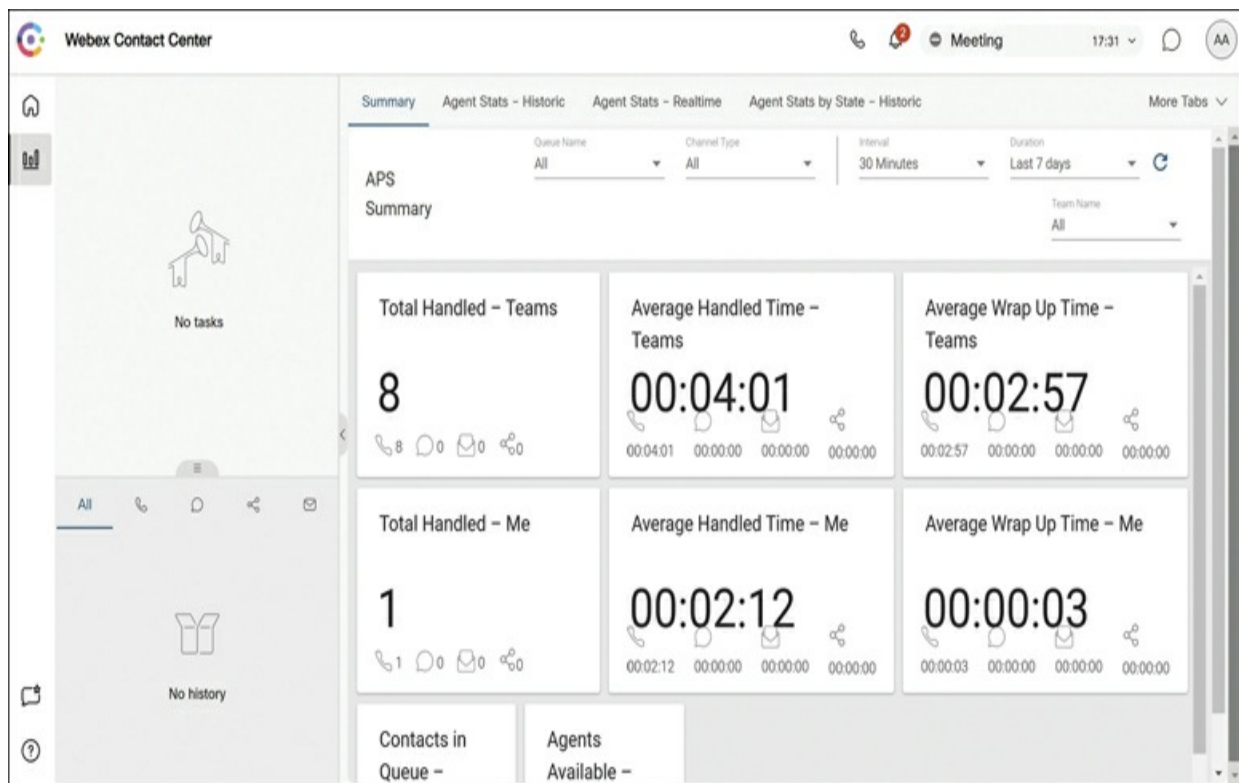


Figure 7-8 Agent Performance Metrics Within Webex Contact Center Desktop

Integrations

The Webex Contact Center solution also can be integrated into various other systems to enable agents to view and take work directly from another system. It is common to see a Webex Contact Center deployment integrated with a CRM system like Salesforce, Microsoft Dynamics, or Zendesk. By integrating your contact center deployment into a CRM tool, you can shrink

down the number of tools an agent needs to work from.

Let's say that you are using Salesforce as your CRM tool to manage all customer cases, account information, sales and booking information, and customer history. When you integrate your Webex Contact Center deployment into Salesforce, agents who are working in both systems, Salesforce and Webex Contact Center, can now log in to a single application, Salesforce, and have control of their Webex Contact Center desktop directly from the Salesforce app. When a customer reaches out, the agent can accept that contact from the Salesforce integration and make notes directly to the Salesforce case, plus see customer account information or related and historical cases for that customer in Salesforce. Additionally, the agent can see all the details related to the contact center interaction with the customer including IVR transcript (if any), transfer a call, place a call on hold, and so on.

This integration provides a powerful interface to allow for a more seamless interaction between a contact and an agent by having all the details about that customer whittled down to a single unified interface. The same is true across other CRM applications like Zendesk and Microsoft Dynamics. At the time of writing, Webex Contact Center supports a few different CRMs:

- Salesforce
- ServiceNow
- Microsoft Dynamics 365
- Zendesk
- Freshdesk

The simplest way to think about integrating Webex Contact Center with a CRM is to take the contact center desktop application and expose that application directly within your CRM. [Figure 7-9](#) shows what this integration could look like.

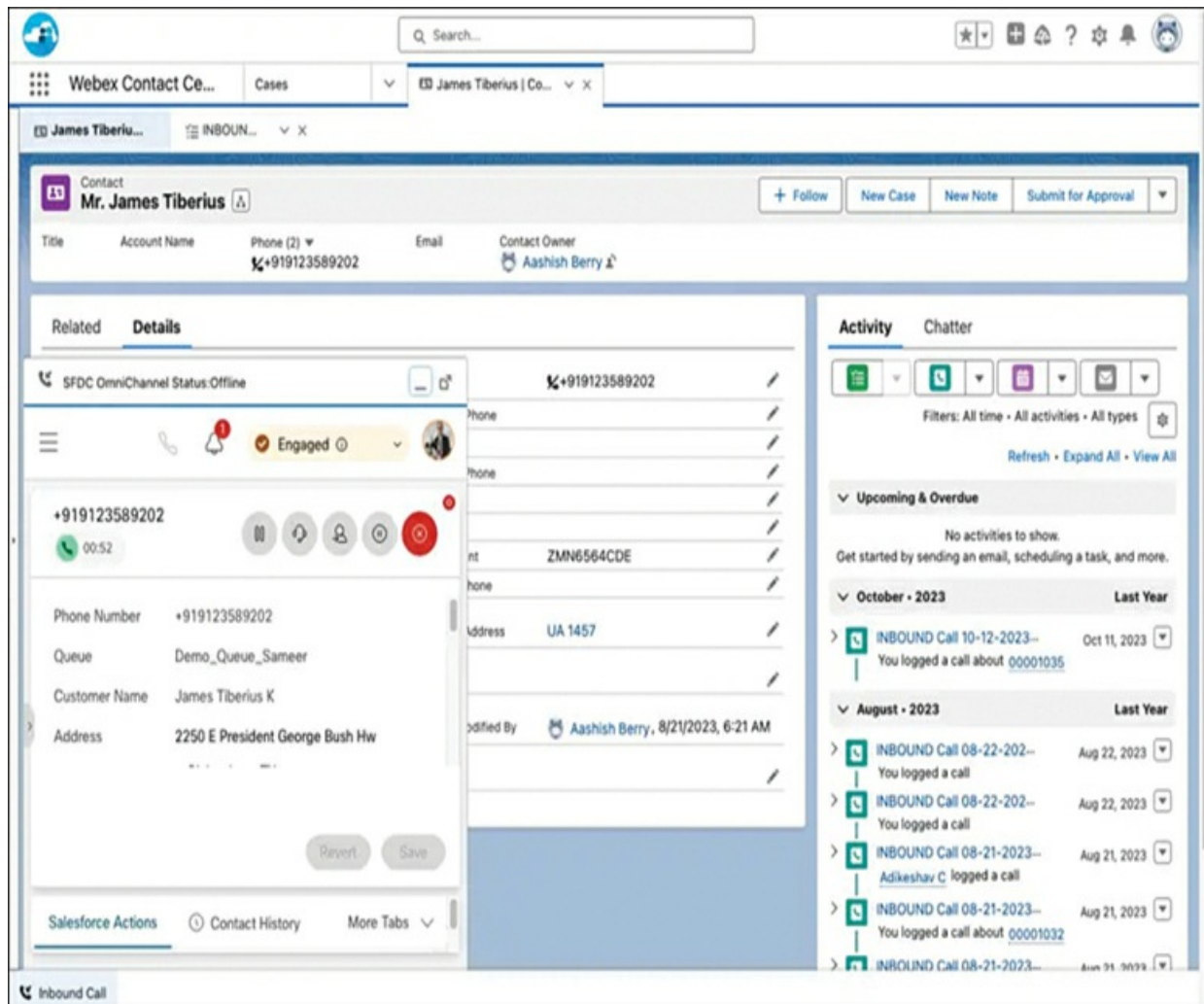


Figure 7-9 Webex Contact Center Integration with Salesforce

Another integration that is commonly used is between Webex Contact Center and Webex Calling, Messaging, and Meetings. Because these applications are all a part of the Webex platform, it makes sense that they would have a tight integration, making integrations between these different systems simple and more effective.

If your organization has licenses for Webex Contact Center and Messaging, Meeting, or Calling, you can enable an integration between the Webex App, as discussed in [Chapter 5](#), and the Webex Contact Center desktop application. With this integration, users do not have to sign into both the Webex App and the contact center desktop application. Within the contact center desktop application, a Webex App icon will appear in the header of the application, allowing agents to open the Webex App for Messaging, Meetings, or Calling,

and all notifications that would normally go to the Webex App desktop client will now come through the contact center desktop application. This integration makes it simple for agents to chat with other users through the contact center desktop application, join Webex meetings, and if they are assigned a number through Webex Calling, accept and place calls using Webex Calling. [Figure 7-10](#) shows how this integration is accessed through the contact center desktop application.

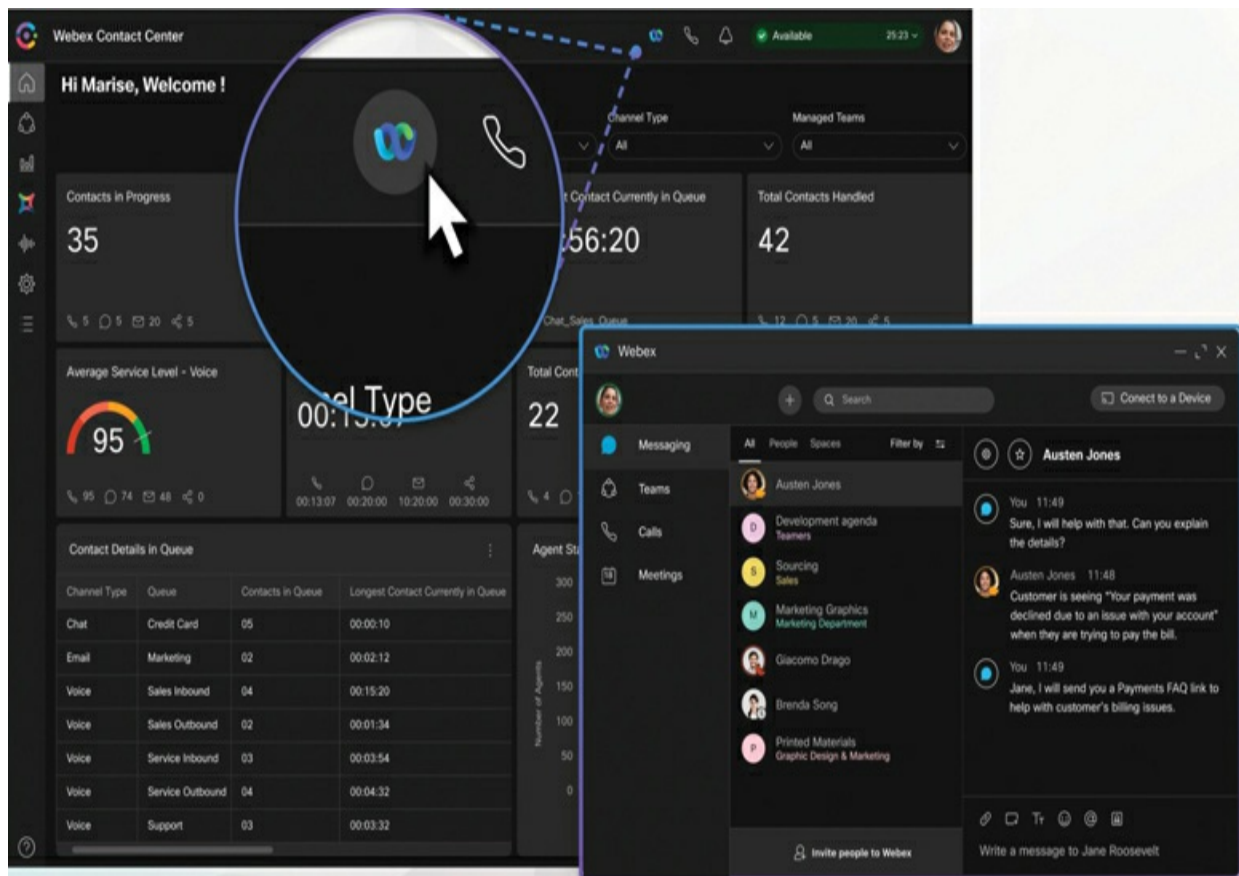


Figure 7-10 Webex Contact Center Desktop App Integration with Webex Messaging, Meeting, and Calling

The Webex Calling integration with Webex Contact Center, depending on your deployment, plays an even more vital role because agents can leverage Webex Calling for their phone extensions to be reached via PSTN calls. When a call comes into Webex Contact Center and needs to be routed to an agent, that agent must have a phone extension for the call to be routed to. This extension can be assigned through the Webex Calling service to allow the agent to have a phone extension assigned, allowing them to receive and

place calls both internally and externally. [Chapter 6, “Collaboration: Webex Calling,”](#) covered the Webex Calling platform and infrastructure in more depth.

The last, and arguably most important, integration we will touch on in this chapter is API integrations. Webex Contact Center is truly a pure-cloud as-a-Service offering that enables enterprises to customize their contact center to meet the demands of their business. We will discuss this issue in more detail later in this chapter when we discuss the architecture of a Webex Contact Center deployment and how it fits into the SaaS model defined in [Chapter 2, “SaaS Architectures.”](#)

Webex Contact Center has a robust set of APIs that allow for virtually every aspect of the deployment to be configured and managed via those APIs. This means that from the deployment, configuration, call routing, and analytics down to the desktop application, each aspect of the deployment can be completely customized and tailored to suit the environment that it is being deployed in. Perhaps instead of using the Webex Contact Center desktop application, you want to custom-develop your own application or integrate the desktop application controls for Webex Contact Center into an existing application you own; if so, you could do that. Or perhaps instead of using Analyzer to build and view dashboard for telemetry related to your deployment, you would prefer to use the Search API to query and display that data in another system entirely. Again, that is completely doable.

When Webex Contact Center was designed, it was designed to be API accessible from almost every component, such that each service that is running in the Webex cloud can be accessed and controlled independently via the API services. These APIs are documented on the Webex Contact Center for Developers portal (<https://developer.webex.com/webex-contact-center/docs/webex-contact-center>). There is verbose documentation for each API, from how to authenticate to the API to how to develop integrations; plus, there is an API reference doc to show the parameters and usage for each available API.

An API-first approach allows for the Webex Contact Center solution to be customizable to meet almost any need a customer may have. Obviously, the catch with using the APIs is that it requires custom development to leverage the functionality of those APIs. Organizations that do not have the in-house

talent to develop a custom solution can rely on third-party vendors who specialize in developing custom contact center solutions, leveraging these APIs.

Webex Workforce Optimization

Another solution closely related to Webex Contact Center is called Webex Workforce Optimization (WFO). WFO is not directly built into the Webex Contact Center product but is an integration that supports this product, as well as many other contact center solutions. The purpose of WFO is to empower organizations to have a strategy for maintaining an efficient and effective contact center. WFO consists of three products within its suite: workforce management, quality management, and analytics.

The workforce management arm of WFO provides a set of tools to help organizations plan and manage their contact center operations using automation, scheduling, forecasting, and other tools to help you ensure that you can deliver a consistent experience through your contact center solution. Workforce management's connector connects to your existing contact center solution to extract telemetry and generate reports and insights.

You could consider workforce management to be the main hub of the WFO solution. Workforce management enables your users to view and update their shift schedules, swap schedules with other users, submit vacation requests, manage overtime, view data and analytics about your contact center, employ gamification techniques to motivate employees, and many other tasks. And it does all of this in real time. Consequently, you won't need to wait for a daily or weekly report to see and detect issues or problems that need to be fixed. You can be notified instantly to avoid any disruptions or issues throughout the day. Additionally, workforce management can be connected to third-party calendar services (Gmail, Microsoft Outlook, and so on), set up to send SMS notifications, and connected to payroll systems.

Another product available in the WFO suite is quality management. Quality management delivers a set of tools to help you uncover areas for improvement in your contact center deployment. This is achieved by automating the analysis of every contact center interaction, helping to uncover areas for potential growth and improvement. This outcome is

achieved through call recording of all phone interactions and real-time or historical screenshots of the agent desktop. With the transcripts and audio recordings of all contact interactions in your contact center, automated and custom evaluation tooling is applied to each contact to derive insights about the interaction.

Quality management enables you to view and interact with each contact center interaction to better understand the health and metrics about your customers' interactions and, when integrated with WFO analytics, uncover ways to improve. [Figure 7-11](#) shows how to review a recording, display the text from the conversation, listen to the audio recording, see the agent desktop application, and even get AI-generated sentiment analysis.

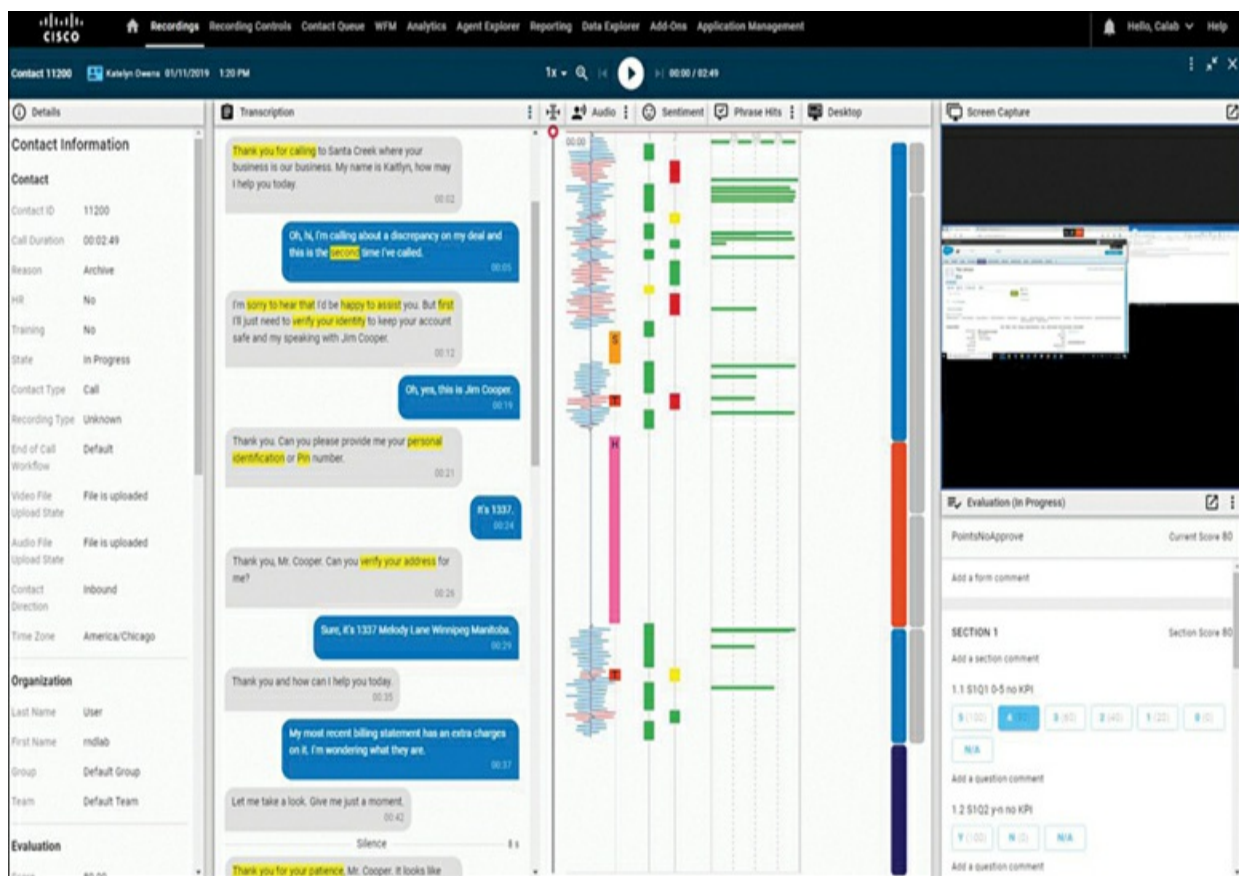


Figure 7-11 Webex WFO Quality Management Recording Analysis

The third and final product in the WFO suite is WFO analytics. Analytics takes the data from your contact center deployment, along with telemetry from WFO quality management, to help you understand the state of your

business and ultimately improve your revenue. WFO analytics enables you to view and interact with your data in both an automated and manual way that makes uncovering insights easy. It supports the following features:

- Speech analysis
- Speech-to-text for audio recordings
- Text analytics for omni-channel communication
- Agent desktop application usage analytics
- Automated sentiment analysis for calls
- Built-in and customizable dashboard for quick consumption of data
- Predictive net promoter scores (NPS)
- Advanced speech and text search

The WFO suite of tools gives you a lot more data and actionable insights about your contact center environment to both empower your employees to manage their schedules directly and to help you deliver a great customer experience through both automated and custom tooling delivered in WFO quality management and analytics.

Webex Contact Center Platform

Now that we've addressed some of the features and capabilities of a Webex Contact Center deployment, let's examine the Webex Contact Center platform and how it fits into the overall SaaS architecture model introduced back in [Chapter 2](#) and shown here again in [Figure 7-12](#).

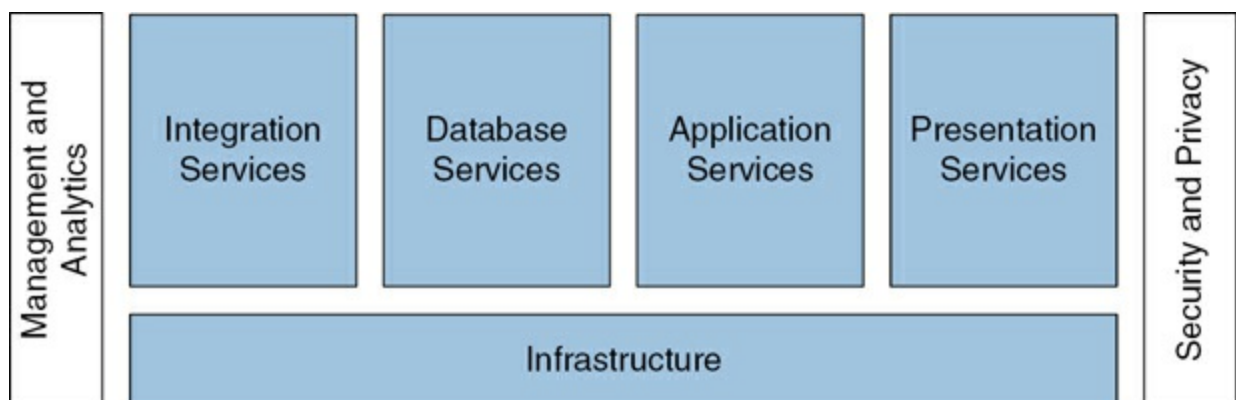


Figure 7-12 SaaS Architectural Model

Like any SaaS application, Webex Contact Center’s architecture has services and components that fit into each of the blocks in [Figure 7-12](#). Let’s look at how the Webex Contact Center platform leverages these different services to provide a reliable, scalable contact center platform. Note that the intention of this chapter (and this book) is not to give you detailed information on the inner workings of Cisco’s cloud services but rather to give you an overall understanding of the capabilities needed to deliver a collaboration platform such as Webex Contact Center. After reading through this section, you should have a good idea of what it takes to build a cloud service like Webex Contact Center at a high level.

Infrastructure

At the core of any SaaS application is the infrastructure. The infrastructure that your application is built on must be constructed in a way that scales to users on demand, is resilient to handle outages and failures, can meet the demands of users quickly, and has enough flexibility to support future innovation and growth. Although the underlying infrastructure should never be something that an end user should have to care about, there are times when having a deeper knowledge of SaaS infrastructure concepts could help you troubleshoot issues with your network or improve performance when using a SaaS service.

Although the goal of this chapter is not to show you the exact infrastructure and inner workings of a Webex Contact Center deployment, you should have a deeper understanding of the type of SaaS architecture that a platform like Webex Contact Center would need.

From an end-user perspective, you should be able to think of a Webex Contact Center architecture as simply as that shown in [Figure 7-13](#). You use a hosted software application that solves a business need—in this case, supporting contacts from customers through various digital channels. What happens in the SaaS is abstracted away from the end user and it simply “works.”

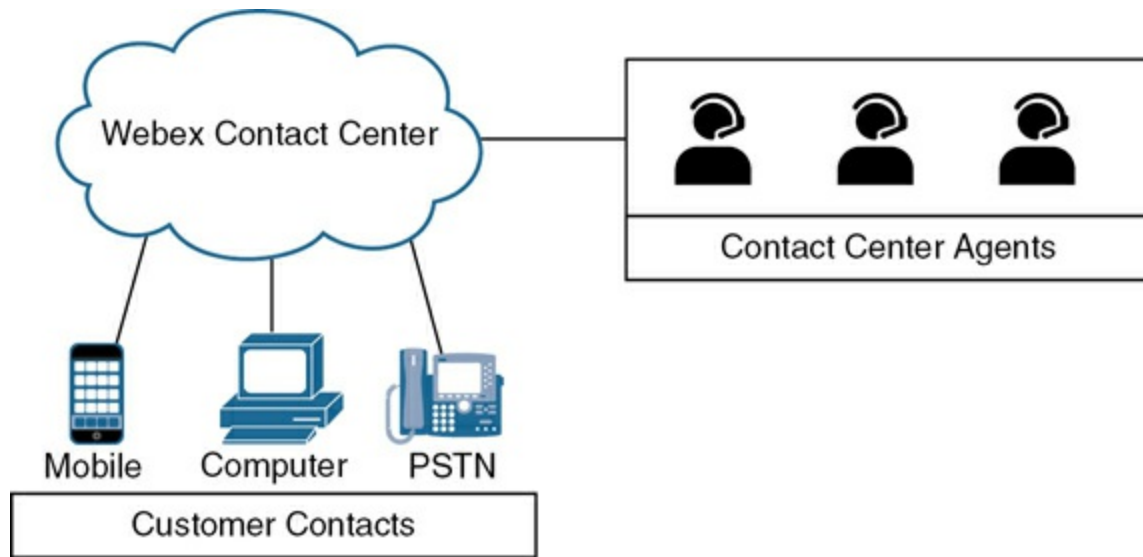


Figure 7-13 Webex Contact Center Services in the Cloud

As you have learned about the various components and features of a Webex Contact Center deployment, you likely understand that there is a lot more beneath the surface that is required to make a complex software platform like Webex Contact Center work seamlessly from the cloud. If you were to look behind the scenes at the infrastructure, what you would really find is an architecture that more closely resembles [Figure 7-14](#).

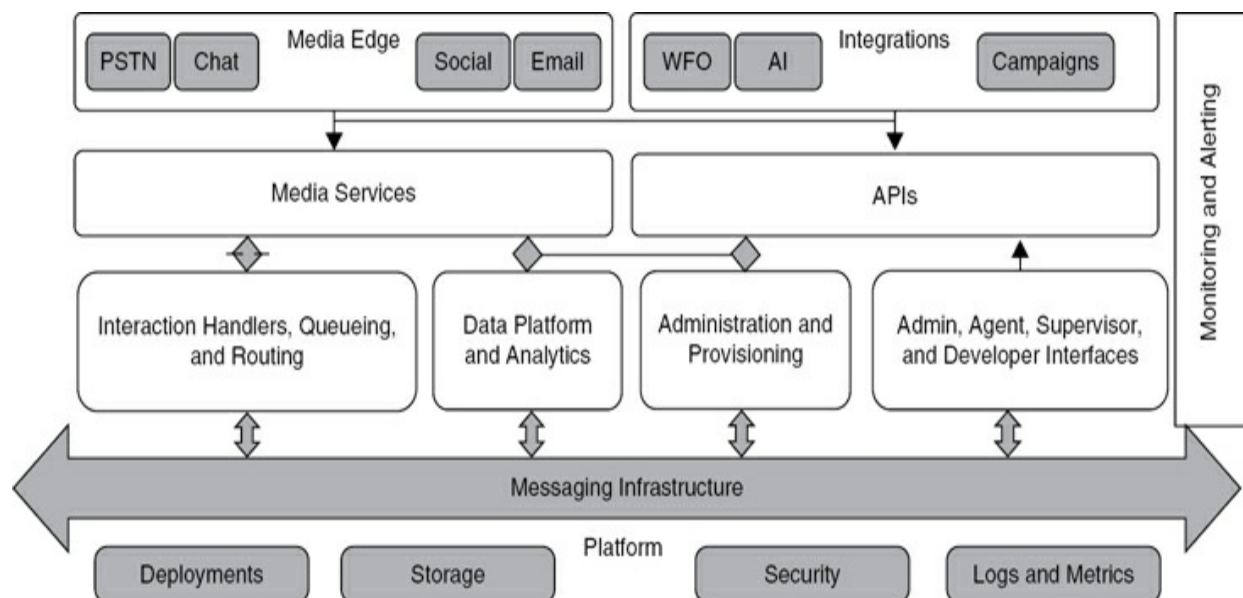


Figure 7-14 Logical Architecture of Webex Contact Center

You can see that many different services are all working in tandem to

orchestrate all the various features provided by the platform. At the core of the deployment, you have your platform components. They include databases and storage, for example, for the handling of configurations, backups, version retention, audit logging, and so on. You have services handling deployments to allow for routine upgrades, version changes, and rollouts of new features within an environment. You have security services to handle components from networking security into and out of your cloud environment down to security services to handle authentication and authorization into your services from contacts to APIs. And finally, you have logging and metrics to give you observability into the state of the deployment, to help aid in troubleshooting errant services, and to allow you to receive alerts and warnings if things start to get out of control.

Webex Contact Center is deployed in Amazon Web Services (AWS) and available in multiple regions to allow for deployments to be closer to the customer, and for some countries to fulfill regulatory requirements for data sovereignty. Inside of AWS, like most SaaS services, Webex Contact Center is composed of many different microservices, all containerized and deployed via build pipelines that do automated testing of new software and automated deployment of new services and versions. This structure ensures that deployments are done consistently and can be done quickly. Additionally, using containerized services gives the system elasticity to be able to scale up and down on demand. At times of higher load, tools like Amazon Elastic Kubernetes Services (EKS) can scale up and down the number of microservices running at a given time to be able to handle the additional volume of traffic. Or they can scale down when the load on the system is lower.

Like both Webex Meetings and Webex Calling, covered in [Chapters 5 and 6](#), Webex Contact Center has a real-time media component to it when handling phone interactions with users. Real-time media adds another layer of complexity to cloud SaaS deployments because the service needs to be able to handle this traffic with low latency. This means that services that are dealing with any real-time media must be deployed as close to the customer as possible and be architected in a way to handle media without adding additional buffering or latency; otherwise, it would impair the phone interaction.

Application Services

Webex Contact Center hosts many different services, called microservices, each of which is responsible for specific functions related to a contact center deployment. These services deliver capabilities such as supporting inbound contacts from customers, interacting with integrations like Salesforce and Zendesk, delivering API capabilities, and designing flows for various contact center interactions. The following are different service categories in the Webex Contact Center platform:

- Authentication and Authorization
- Media Termination and Switching
- Media Processing
- API Gateway
- Call Routing and Queueing
- User Administration
- Configuration Management
- Analytics
- Metrics and Logging
- Agent Desktop Interface
- Supervisor Functions
- Contact Interactions
- Call Recording
- Notifications
- Data Encryption
- Scheduling
- Flow Engine
- Web Services

Although this list is not exhaustive, and some of these categories may have a

few different services working together to deliver a single capability, it gives you a better understanding of the amount of orchestration and planning that needs to go into delivering the high-level functionality that is exposed to end users.

We will not explore each of these services here, but we will touch on some of the more fundamental services required for running the Webex Contact Center platform. Some services, such as authentication and authorization, media handling, and encryption, are covered in [Chapter 5](#) as a part of the Webex suite.

Contact Interaction Management

The set of services that work together to support the ingress or egress handling of contacts to and/or from a contact center is what we mean by contact interaction management. With a Webex Contact Center deployment, customers likely have multiple ingress options for connecting with an agent. They could call into a phone number, click to chat from a messaging application, send an email, or chat from an application like WhatsApp. Depending on the path that users take to connect with the contact center, they may hit a different service. For example, let's say that a user dials a number to reach support. This number is routed to your contact center. One of the first services to come into play during this interaction is your media termination service. [Figure 7-15](#) highlights an example of an inbound PSTN call from a customer to a number configured to receive inbound calls in your Webex Contact Center configuration.

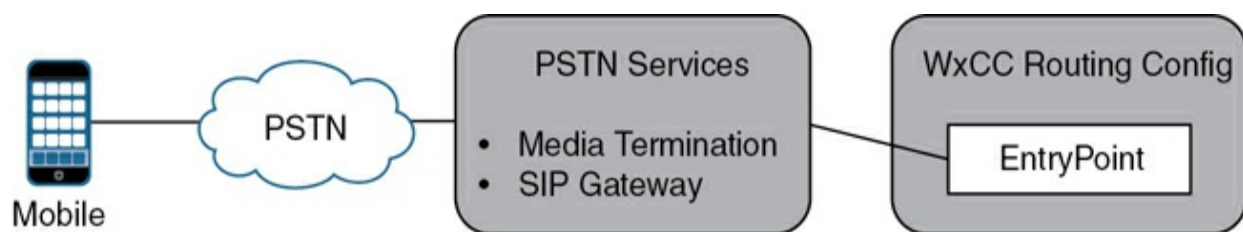


Figure 7-15 Ingress Call Path for Webex Contact Center

The call first is routed through the PSTN, and then the call is sent to Webex Contact Center's PSTN services. The SIP Gateway handles the call signaling to negotiate the connection details for the call, and the media for the call will be sent to the media termination point from the PSTN when the call is

established. In the Webex Contact Center configuration, the call is routed to the EntryPoint that is associated with the number the customer dialed and ultimately gets routed to an agent that the customer can speak with.

But what about a scenario where a customer is connecting from another digital channel that is not a phone call. Perhaps the customer sends an email or connects via a chat service, as shown in [Figure 7-16](#).

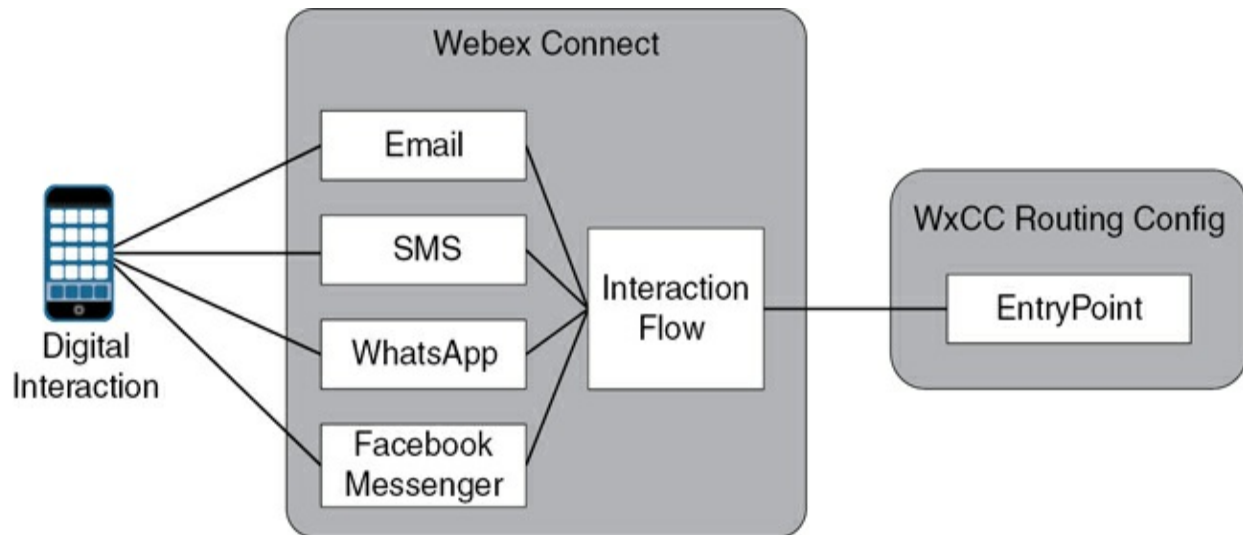


Figure 7-16 Ingress Digital Interaction Path for Webex Contact Center

In this example, digital interactions that are not phone calls are handled through the Webex Connect service. Within Webex Connect, you configure the various services you want to support for digital communication channels, and you also set up interaction flows for each of those services. The interaction flows determine how the conversation is handled and when to escalate the interaction to a live agent if necessary.

For each of these components to work properly, you must have a few different services working together to handle these interactions. You will need a web service to handle connections with each of the supported digital channels. You also will need a database to store the configurations used for those connections, as well as for the interaction flows that are generated. Plus, you will need an event service to handle communication between these different microservices and to maintain the state of each communication thread. In [Figure 7-15](#), you can see that the left and right sides of the diagram are identical to what is shown in [Figure 7-16](#); the difference is the services in

between that handle the interaction.

Queues and Routing

After a contact is routed to a Webex Contact Center EntryPoint, it is handed off to the flow engine. The flow engine executes the flow script that has been created for the entry point to determine how to handle the contact interaction. If the flow engine determines the contact should be routed to an agent, the queue and routing services come into play. The queueing service is responsible for routing a contact to an available agent using the routing strategy that was configured by the administrator. For example, this could be a skill-based routing or longest available agent routing. Teams can be assigned to a queue via what is called call distribution groups. A call distribution group is a collection of teams. When a contact hits the queue, the queueing service can tell the notification service to inform all available agents that there is a contact in the queue. When an agent is available, the contact is routed to the agent via the routing service. The interactions between these services are essentially API calls and event streams that allow each service to communicate to the other services independently. This flow of communication is illustrated in [Figure 7-17](#), where you can see how a contact is pushed to the queue from the flow engine and ultimately routed to the desktop agent for handling.

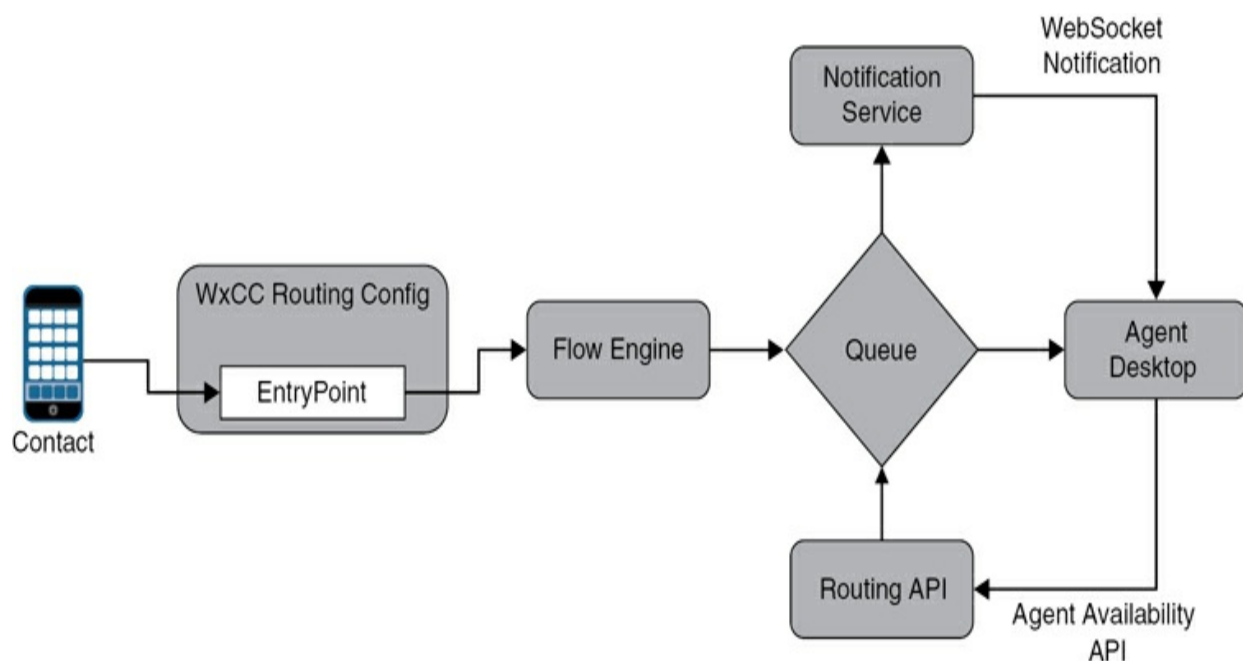


Figure 7-17 Webex Contact Center Routing and Queue Services

If a contact were to be routed to a queue and there were no available agents, the contact handler in the flow engine would maintain control of the contact until an agent became available. The agent's availability can be updated from the agent desktop via the routing API. When the routing API is updated with an agent's availability, the queue service will become aware of this fact and can then proceed to route the contact to the available agent.

Desktop Agent

The desktop agent application was uniquely designed to be highly modular and customizable so that enterprises could tailor the layout and functionality of the application to meet their needs. This customizability is accomplished by essentially making the browser-based application support the addition of custom widgets. These widgets are just HTML components that can be custom-developed and injected onto the application.

The desktop agent application leverages a combination of WebSockets and API communication to communicate with the Webex Contact Center backend to receive events and notifications from the cloud and to pull down data as needed via the API. Most commonly, the desktop agent leverages asynchronous APIs where it will make the request to the Webex Contact Center API gateway, and the result of that request will be sent back over the WebSocket connection. [Figure 7-18](#) shows the logical architecture for the desktop agent.

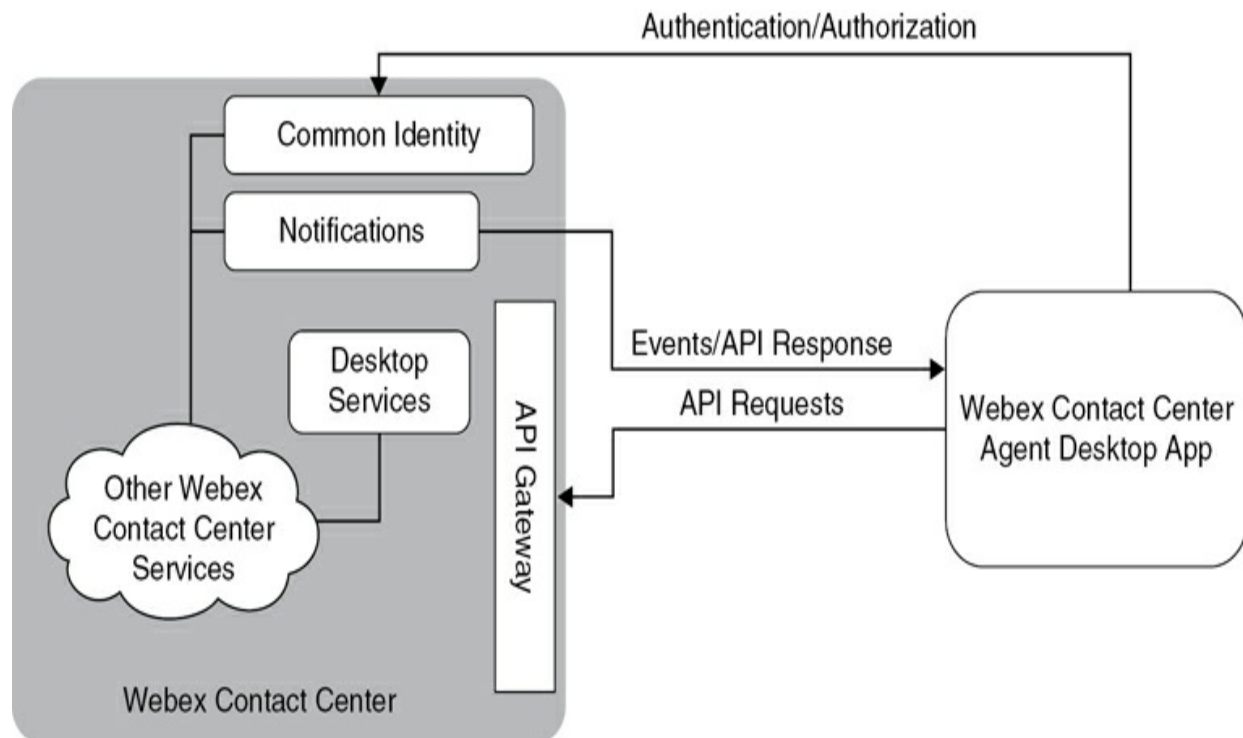


Figure 7-18 Webex Agent Desktop Logical Architecture

When an agent logs in to the agent desktop application, first the user is authenticated via the Common Identity service. This service maintains user information and performs authentication services for all of Webex, including Webex Calling, Meetings, and Messaging. After the user is authorized, the application will receive a bearer token. This token can then be used for communicating with the API gateway and setting up the WebSocket between the desktop app and the Webex Contact Center cloud service. When the client establishes a connection with the Webex Contact Center cloud, the desktop widgets and layout are pushed to the client to instruct it what data it should show and how to organize the widgets on the application. Again, you can almost think of the application as a web browser, and the Webex Contact Center cloud is telling the client what HTML to render to the user.

Presentation Services

A consumer of a contact center service is likely to never interface with any presentation services from that contact center service. Their interactions will all take place through a digital channel through another application or service or via a phone call into the system. However, the agents, supervisors, and

administrators of a Webex Contact Center service will primarily spend their time utilizing the presentation services.

For Webex Contact Center, three primary interfaces are utilized. First, you have the presentation services for the administration and configuration of your Webex Contact Center deployment. This contact is done primarily through a site called Control Hub. You also have the management portal for Webex Contact Center deployments that can be accessed through Control Hub or directly via another URL. Next, you have presentation services for analytics and reporting, which is accessed through the Analyzer web interface, which also can be accessed directly or from a link within Control Hub. And finally, you have the agent presentation services, which are primarily the agent desktop application. This application is either exposed through an integration service like a CRM platform or accessed directly through the desktop application.

Like the rest of the Webex Suite, Webex Contact Centers presentation services are heavily driven through REST APIs. While the frontend development is done with a mixture of HTML, CSS, and JavaScript, the presentation services communicate with the backend services via APIs. In this way, various services and components within the Webex Contact Center architecture can be exposed in a variety of different methods. Regardless of how the frontend services are developed, they can all access the backend data and services through REST API calls.

Database Services

As we have mentioned previously, many of the application services are dependent on some type of database service to be able to function properly. In fact, database services are critical to almost any SaaS offering. The ability to store and retrieve data—whether it is configuration data, user information, logging, or metrics—is critical to almost any cloud-native application. This service also must be handled in a secure and scalable manner.

In [Chapter 2](#) you learned about the various types of databases that are leveraged in a SaaS architecture. While the purpose of this chapter is not to go over the specific databases that are used for Webex Contact Center, we will look at how different database types are needed for a Webex Contact

Center deployment.

Flow Configurations

As complex flows are built in the Flow Designer interface, those flows must be stored and retrieved, and they must be able to be updated at any time. Flows need to be stored in a structured, relational database because they are utilized by many different services in the system and need to be able to relate a flow configuration to a routing strategy and an entry point, for example. When they are stored in a structured database, this means that there is a defined schema by which that data must be formatted and stored. If you think about the Flow Designer interface, you can imagine that it is essentially building out this data to match the necessary schema required by the database.

User Profile Information

Agents within Webex Contact Center are mapped to a team, which is mapped to a site. Each entity is related to one another in some way, and the data that is needed for each agent is bounded. This approach lends itself well to a structured relational database because each entity (user, team, site) needs to have a relational structure to one another in the database. [Figure 7-19](#) shows three different tables all using a relational structure to relate the data to one another. Each table stores data for a distinct entity—in this case, user, team, and site data.

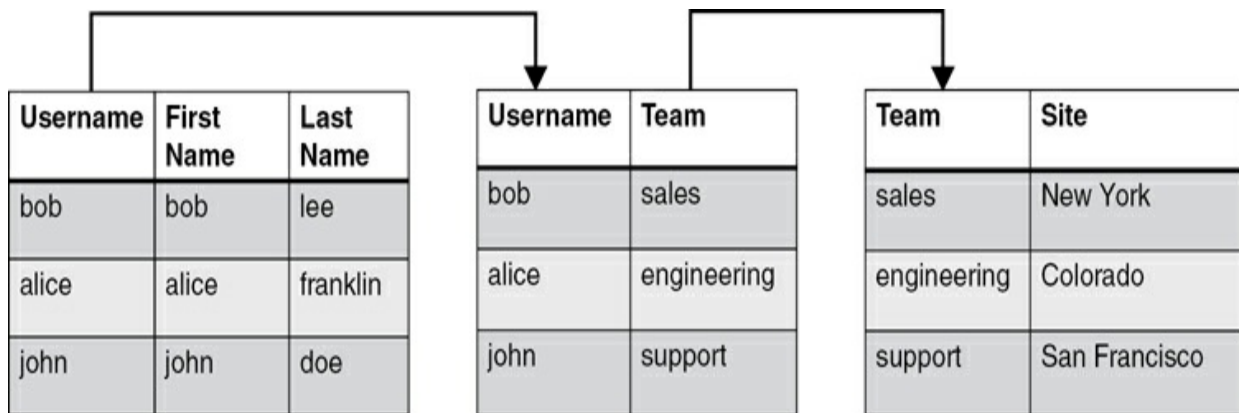


Figure 7-19 Relational Data Structure for Users, Sites, and Teams

Logging and Telemetry

An example of data that would be stored in a nonrelational database is logging data. Logging data plays a critical role in supporting a cloud application by taking logging data from various components and services and storing it in a centralized location. This allows the developers of the application to easily troubleshoot and resolve issues.

As you can imagine, a large-scale cloud service could generate a large number of logs in a short period of time. This means that the services that are handling these logs need to be able to hold massive amounts of data but also have retention policies in place to automatically offload data after it reaches a specific threshold. That threshold could be a specific memory size or the age of the data. Often services will keep logging data on hand for a certain period of days, and anything outside of that configured range is automatically discarded.

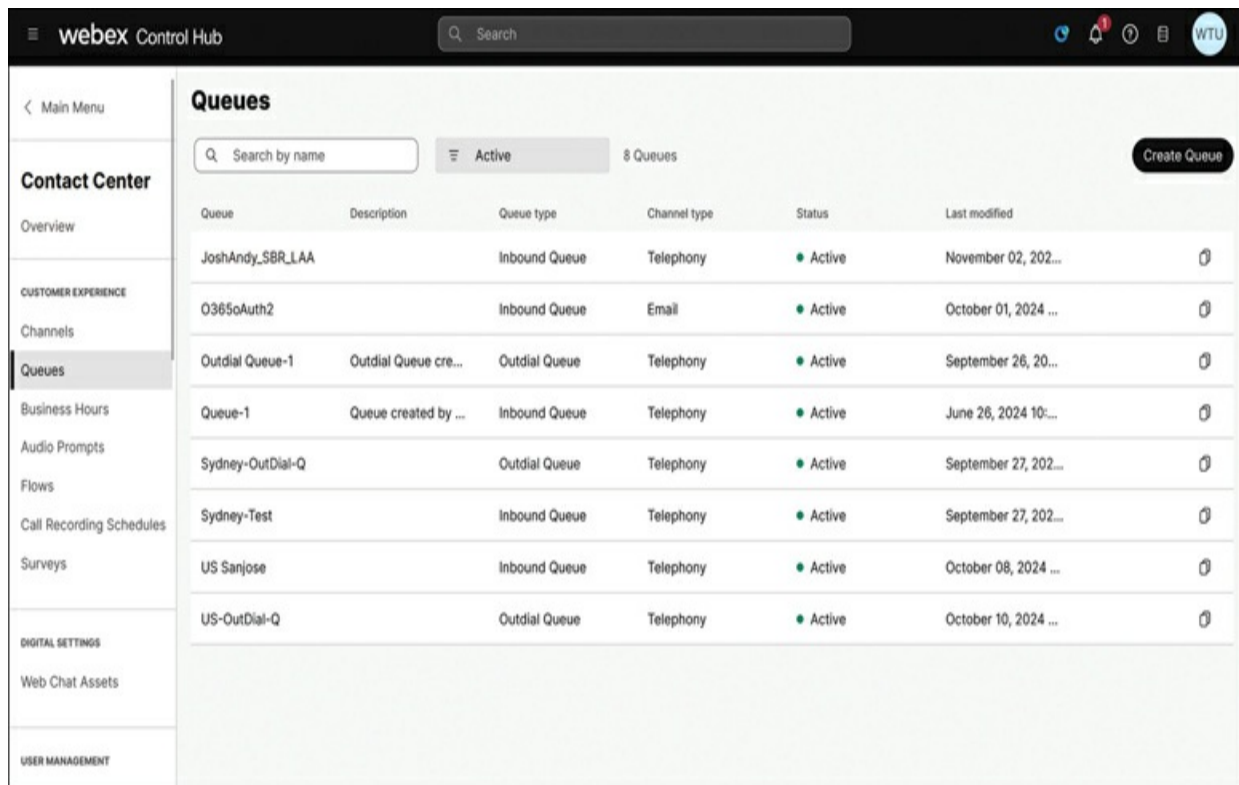
There are many platforms available to index and store large quantities of logs, such as Splunk and Opensearch. These platforms not only give you a way to store logs but also a way to visualize and search through those logs; plus, you can set up automated alerting and monitoring on those logs.

In addition to storing logs, storing telemetry from various systems is another critical service to monitor to be able to maintain a high-performance system. For a Webex Contact Center deployment, this could be things like total contact interactions in a day, average amount of time to resolve a contact engagement, or number of contacts in a queue for a given period. This could also be infrastructure metrics to assist with keeping your system running smoothly, like CPU, memory, and latency of a given microservice. Again, the quantity of metrics that could be stored for a large-scale deployment could be massive, so having a system that can handle large amounts of data—and storing that data quickly—is important.

Management and Analytics

Another core pillar of any SaaS deployment is having an administrative interface that allows you to manage your SaaS tenant, as well as for viewing metrics or analytics about the usage of the service for your tenant. For

Webex, the primary management interface for all Webex products, including Webex Contact Center, is Control Hub. Control Hub is the place where details about your Webex platform usage will appear, including all subscriptions and services that you are leveraging on the Webex platform, where you manage users, and the licenses assigned to those users. It has organization-level configurations that you can manage for your entire Webex suite. [Figure 7-20](#) shows Webex Contact Center queues in Control Hub.



Queue	Description	Queue type	Channel type	Status	Last modified
JoshAndy_SBR_LAA		Inbound Queue	Telephony	Active	November 02, 202...
0365oAuth2		Inbound Queue	Email	Active	October 01, 2024 ...
Outdial Queue-1	Outdial Queue cre...	Outdial Queue	Telephony	Active	September 26, 20...
Queue-1	Queue created by ...	Inbound Queue	Telephony	Active	June 26, 2024 10:...
Sydney-Outdial-Q		Outdial Queue	Telephony	Active	September 27, 202...
Sydney-Test		Inbound Queue	Telephony	Active	September 27, 202...
US Sanjose		Inbound Queue	Telephony	Active	October 08, 2024 ...
US-Outdial-Q		Outdial Queue	Telephony	Active	October 10, 2024 ...

Figure 7-20 Queues for Webex Contact Center in Control Hub

Although some contact center configurations can be handled from Control Hub, the primary site for configuring a Webex Contact Center deployment is through the management portal. The Webex Contact Center management portal can be accessed via a cross-launch link directly from Control Hub or via a direct URL.

Note

The Webex Control Hub management portal has several different URLs, each based on the location where your contact center tenant is located. For example, if your Webex Contact Center is in a U.S.-

based tenant, the management portal URL is different from a tenant that is in a UK-based tenant. Using the cross-launch link from Control Hub is the easiest option for accessing the management portal to the correct location.

This site is the place where most configurations are done; it also enables you to view real-time data, monitor interactions between contacts and agents, and view agents' status in real time. [Figure 7-21](#) shows the landing page of the Webex Contact Center management portal.

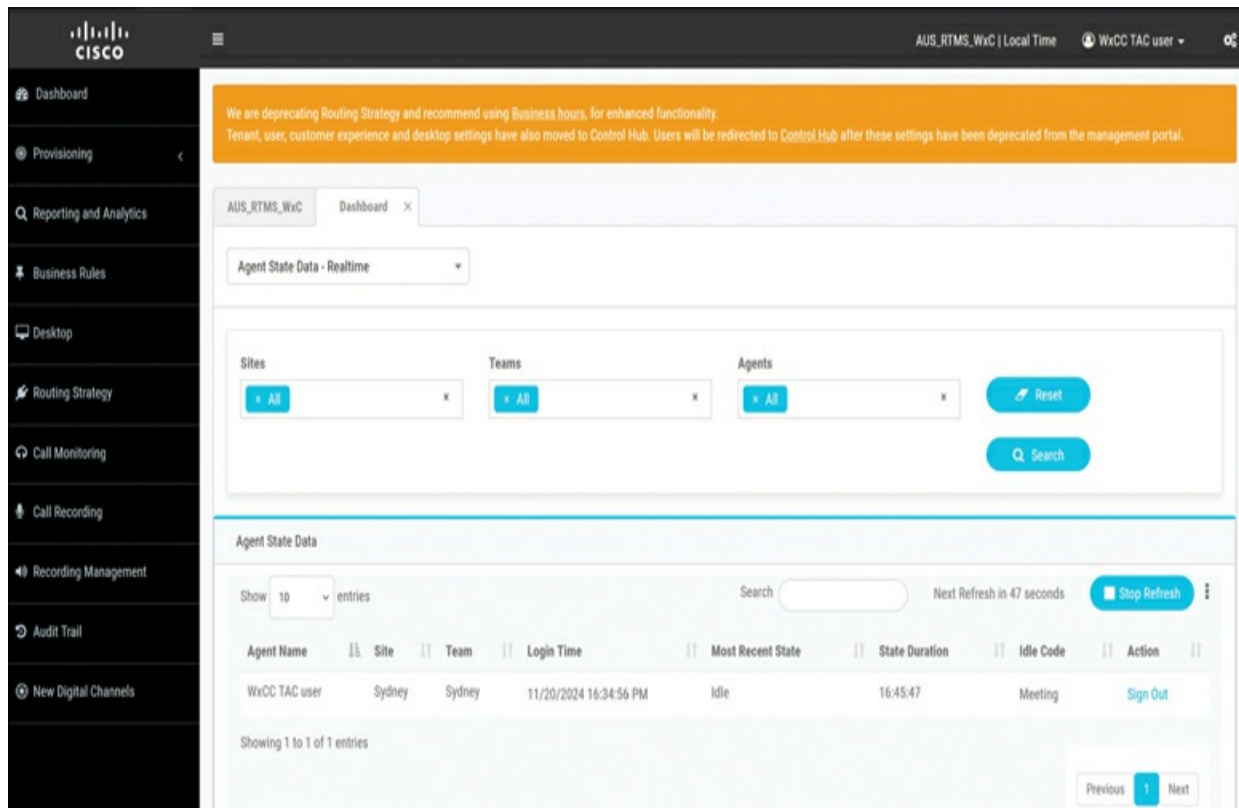


Figure 7-21 Webex Contact Center Management Portal

In addition to the configuration and management portals, there is also the Analyzer portal. Analyzer is Webex Contact Center's primary interface for all metrics and telemetry for a contact center deployment. It enables you to create and manage custom dashboards, view prebuilt dashboards for common metrics, visualize data in various ways, as well as use calculations and functions like those in Excel to derive data from various sources. The Analyzer portal, Control Hub, and the management portal are all web applications that can be accessed by administrative users or via an API.

Integration Services

We discussed the integration capabilities of Webex Contact Center earlier in this chapter. Webex Contact Center deployments make extensive use of the ability to integrate with other services. The primary interface for allowing other services to work with a Webex Contact Center deployment is through RESTful APIs. Webex's APIs leverage OAuth2 to grant services an access token to authenticate with against the Webex APIs. The public APIs for a Webex Contact Center deployment are extensive and allow you to do just about anything you can do from the web interface using the APIs instead. This means that you can virtually customize your entire contact center deployment, from configuration to the actual desktop interface, all using APIs.

The Webex Contact Center for Developers site at <https://developer.webex.com/webex-contact-center/docs/webex-contact-center> has documentation on gaining access to the APIs, authentication, and API reference docs showing you each available API that you can use and the specifications for that API. It also includes sample code and allows you to test the API from the developer portal to test the APIs without having to write any code.

Note

For more details on API security, including how OAuth works, see [Chapter 4, “Security and Privacy for SaaS.”](#)

Webex Contact Center also supports integrations using connectors, which allow the service to connect to other third-party applications with a few clicks. This includes services such as Google Contact Center Artificial Intelligence (CCAI), Salesforce, Microsoft Teams, and the Webex Experience Management platform. These connectors are custom-built tools built directly in Webex Control Hub that make it easy to set up a connection between your Webex Contact Center tenant and another cloud service you want to leverage or may already use in your organization.

Security and Privacy

Webex as a platform is dedicated to building security into the core of its products. That spans from the infrastructure to the presentation services for users and everything in between. It includes things like data security, authentication and authorization practices, and hardening services to restrict access to only what is needed.

At the core of Webex's identity and user authentication services is a service called Common Identity, which leverages OAuth for granting access to services using a scope that allows a user or service to access only the specific services that are needed and nothing else. This follows the principle of least-privilege security, which says that a user or service should be given access only to the minimum set of services or data needed to perform a job. For more information on these principles and the OAuth standard, see [Chapter 4](#).

In addition to users or services accessing services through an API or integration externally, there are also all the microservices running in AWS that talk to one another. It is common practice for each service to be given a machine account to allow it to authenticate and talk to other services, and this machine account is given only the specific access that is required for that service to function.

Webex Control Hub also exposes various security controls for administrators to configure and leverage themselves. For example, in Control Hub, an administrator can enable a service called privacy shield (which is enabled by default) that allows agents to not record sensitive information on a call as needed. This service protects both the customer and Webex from holding sensitive information for a customer. Control Hub also enables administrators to set up content security policies to define specific domains that you want to allow access to your Webex Contact Center applications and to block all others.

As with any SaaS application, securing data both in transit and at rest is critical to ensuring the safety of customers' data and protecting against the loss of sensitive data. For Webex Contact Center, all data transactions external to the Webex Contact Center cloud are done over HTTPS/TLS, and none of the interfaces for the internal microservices are exposed externally. If a service requires external communication, either inbound or outbound, that

service's communication is routed through a load balancer to ensure that the traffic is encrypted. In addition to data in transit, all data at rest is encrypted. This includes both internal data storage for services in AWS, as well as services like MongoDB and Amazon S3 for storage. [Figure 7-22](#) shows what the data security strategy looks like for Webex Contact Center.

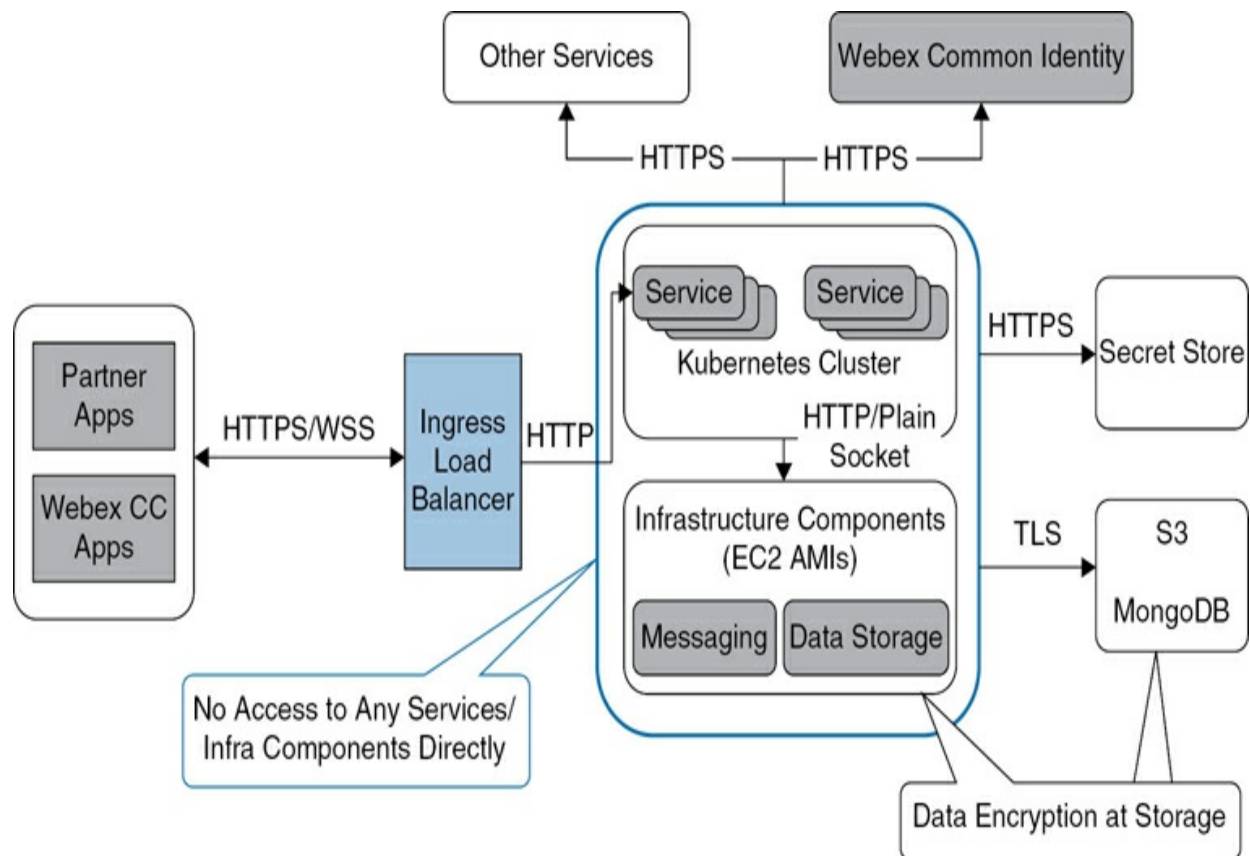


Figure 7-22 Webex Contact Center Data Flow and Security Model

For more information on the security certifications and regulatory requirements that Webex adheres to, as well as information on Cisco's responsible AI framework, check out the "[Security and Privacy](#)" section in [Chapter 5](#).

Summary

In this chapter, we described some of the key features and components of Webex's Contact Center as a Service (CCaaS) product. You learned about product capabilities like omnichannel communication powered by Webex

Connect. You also learned about some of the AI features included with Webex Contact Center, including AI powered virtual agents and the Cisco AI Assistant's integration with the product. In addition, you learned about the advanced call routing capabilities included in a Webex Contact Center deployment and how the Flow Designer utility is an extremely powerful tool. You also gained a better understanding of the integration and analytics capabilities of Webex Contact Center. Next, you learned about Workforce Optimization and how it adds additional value to the product.

After describing the fundamental concepts of the product, we applied those concepts to the SaaS framework that was covered in [Chapter 2](#) and applied that framework to the Webex Contact Center architecture. After reading this chapter, you should have a deeper understanding of the Webex Contact Center product, as well as a better understanding for what type of SaaS architecture is used by Webex Contact Center.

References

- GEC PABX No. 4 (PB4200) Private Telephone System:
<https://www.britishtelephones.com/gec/pabx4.htm>
- The history of the call centre—updated, Jonty Pearce:
<https://www.callcentrehelper.com/the-history-of-the-call-centre-15085.htm>
- Cisco Systems to acquire GeoTel Communications Corp:
<https://newsroom.cisco.com/c/r/newsroom/en/us/a/y1999/m04/cisco-systems-to-acquire-geotel-communications-corp.html>
- Cisco completes acquisition of BroadSoft:
<https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2018/m02/cisco-completes-acquisition-of-broadsoft.html>
- What is a contact center?:
<https://www.cisco.com/c/en/us/solutions/collaboration/what-is-a-contact-center.html>
- Dialogflow CX: <https://cloud.google.com/dialogflow/docs>
- Webex contact center architecture: <https://help.webex.com/en-us/article/utqcm7/Webex-Contact-Center-Architecture>

- Webex contact center setup and administration guide:
<https://help.webex.com/en-us/article/n5595zd/Webex-Contact-Center-Setup-and-Administration-Guide>

Chapter 8. Security: Identity and Access Management

Identity and access management (IAM) is a foundational element of cybersecurity that enables the right individuals to access the right resources at the right times for the right reasons. It includes the policies, processes, and technologies used to manage digital identities and control user access to sensitive information and systems within an organization. At its core, IAM is about defining and managing the roles and access privileges of individual users and the circumstances under which those users are granted or denied those privileges. As organizations increasingly adopt cloud computing and mobile technologies, robust IAM systems have become critical for protecting against unauthorized access and data breaches.

Many people say that “identity is the new firewall.” Cloud computing and remote work have dissolved traditional network boundaries, and the centralized firewall model is no longer adequate. Modern security architectures require controls to be attached directly to every user, device, and workload. Verified identity now serves as the dynamic perimeter that governs access to resources (regardless of physical location, network, or device). This zero-trust approach ensures that authentication and authorization happen at every access point, making identity verification the cornerstone of contemporary security strategy.

The primary components of an IAM framework work together to provide a comprehensive security posture. *Authentication* is the process of verifying a user’s identity, typically through something they know (a password or PIN), something they have (a security token or smartphone), or something they are (a fingerprint or facial scan). However, passwordless deployments are significantly better than traditional username and password systems. When

you implement a passwordless IAM program, you eradicate the most common vector for cyber attacks, such as phishing, credential stuffing, and brute-force attacks, thus significantly strengthening your security posture. For your users, this means a more convenient and frictionless login experience, free from the burden of creating, remembering, and frequently resetting complex passwords.

Authorization, on the other hand, is the process of granting a verified user permission to access specific resources. This process is often managed through policies and access control lists that define what an authenticated user is allowed to do. Together, authentication and authorization ensure that only legitimate users can access the systems and data they are explicitly permitted to view or modify.

Accounting is the “last A in AAA.” AAA stands for authentication, authorization, and accounting. Accounting can help with incident response and auditing. It creates a detailed log of user activity, which is important for security analysis, incident response, and compliance with regulations. For example, it can track which files users accessed or what commands they executed on a server. Logs generated by the accounting process can also help administrators diagnose and resolve network or system issues that users might have experienced.

Modern IAM solutions often include several advanced components to address the complexity of today’s fast-moving technological landscape. Single sign-on (SSO) simplifies the user experience by allowing individuals to access multiple applications with a single set of credentials. Multifactor authentication (MFA) enhances security by requiring two or more verification methods. Identity governance and administration (IGA) tools provide for the management of identity and access rights across multiple systems. Privileged access management (PAM) focuses on securing, controlling, and monitoring access to an organization’s most critical assets. These components collectively enable organizations to maintain robust security postures while ensuring seamless user experiences and regulatory compliance.

Cisco continues to aggressively rearchitect its security portfolio to deliver a comprehensive, unified identity fabric. This strategy is built on three core technological pillars: Cisco Duo; Cisco Identity Services Engine (ISE), which

is the granular on-premises network policy enforcer; and technologies from the identity threat detection and response (ITDR) platform from a company called Oort that Cisco acquired. All these components now power Cisco Identity Intelligence.

In this chapter, we will provide a comprehensive introduction of this evolving portfolio. You will learn the foundational principles of Cisco's Zero Trust philosophy, which dictates the integration and function of these components. We will then perform a technical deep dive into the individual architectures of Duo, ISE, and Oort. Although we will concentrate on the SaaS solutions, we will examine their core capabilities, deployment models, and strategic roles within the broader ecosystem.

Cisco's Zero Trust and Continuous Trust Philosophy

To comprehend the architecture and strategic direction of Cisco's identity portfolio, you must first understand the foundational security principles that govern its design. These principles are not just marketing concepts; they are a direct response to a fundamental shift in the cybersecurity landscape and provide the strategic “why” behind the development and integration of products like Duo and Cisco Identity Intelligence.

The Shift to Identity as the Perimeter

The traditional model of enterprise security, which centered on a heavily fortified corporate network perimeter, has become obsolete. This dissolution is a direct consequence of several irreversible technology and business trends: workforce mobility (even after the worldwide COVID-19 pandemic), the proliferation of bring-your-own-device (BYOD) policies, the mass adoption of cloud services and applications, and the rise of the Internet of Things (IoT). In this new, perimeterless world, users, devices, and applications are distributed globally, accessing resources from anywhere at any time. Look at the example in [Figure 8-1](#).

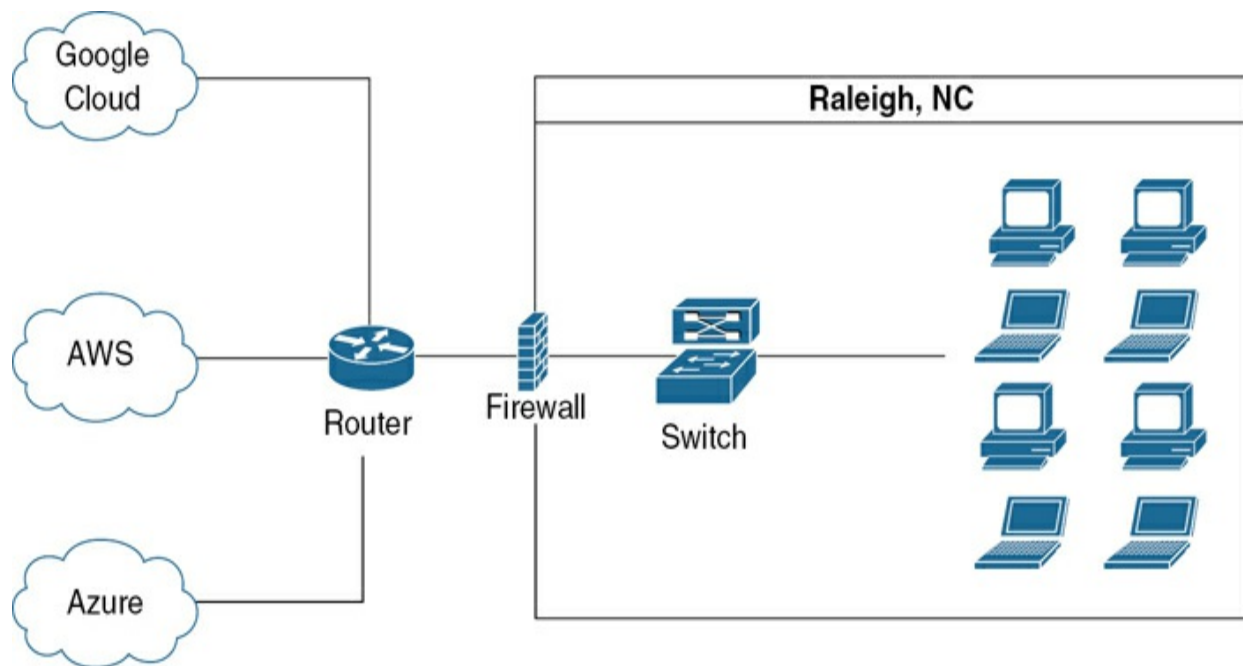


Figure 8-1 A High-Level Architecture of a Typical Business Network

Figure 8-1 shows a common business network architecture. On the left, you have public cloud environments (Google Cloud, AWS, and Azure) where the company runs applications and stores data. On the right is the company's physical office in Raleigh, North Carolina, with employee workstations and laptops connected to a local network switch. A central router manages traffic between the office, the cloud providers, and the Internet, with a firewall positioned to protect the local office network.

In this architecture, the firewall is only protecting the physical office in Raleigh. It inspects traffic coming into and going out of the local office network, but it provides no protection for the corporate applications and data running in AWS, Azure, or Google Cloud. Because the cloud resources are outside the firewall's perimeter, they are directly exposed to the Internet and rely on separate, cloud-native security controls for protection.

However, even if you deploy virtual firewalls in the cloud, there is huge challenge in most deployments: identity management. Even in today's era of AI, our world is dominated by identity-based attacks, where adversaries no longer need to hack in through complex network exploits when they can simply log in using compromised credentials. Many breaches involve the use of lost or stolen credentials. Cisco's own internal analysis and public

statements from its leadership echo this sentiment, recognizing that identity is now the first and most critical line of defense against cyber threats. This fundamental understanding of the modern identity crisis is the driving force behind Cisco's strategic acquisitions of Duo, Oort, and others; and their integration into a unified security platform.

Cisco's Zero Trust Framework

In response to the dissolution of the traditional perimeter, Cisco has adopted and championed a Zero Trust security model. This model is not a single product but a strategic approach and a comprehensive architecture designed to secure the modern, distributed enterprise.

Tip

You can access Cisco's Zero Trust Architecture Guide at <https://www.cisco.com/c/en/us/solutions/collateral/enterprise/design-zone-security/zt-ag.html>.

The core tenet of Zero Trust is the elimination of implicit trust: No user, device, or application is trusted by default, regardless of its location (inside or outside the old corporate network). Every access attempt must be continuously verified against a granular security policy before access is granted.

To translate this philosophy into a tangible architecture, Cisco has structured its Zero Trust framework around three foundational pillars, each addressing a critical domain of the enterprise:

- **Workforce (User and Device Security):** This pillar focuses on establishing and maintaining trust in the users and devices that access corporate resources. It seeks to answer fundamental questions: Is the user who they claim to be? Is the device they are using secure and compliant? This domain is the primary responsibility of Cisco Duo, which provides robust multifactor authentication, device health checks, and adaptive access policies to secure the workforce.
- **Workplace (Network and Cloud Security):** This pillar addresses the security of the underlying network infrastructure, both on-premises and

in the cloud. It involves controlling access to the network and dynamically segmenting traffic to limit the blast radius of a potential breach. This is the traditional domain of the Cisco Identity Services Engine, which provides granular network access control (NAC) for wired, wireless, and VPN connections.

- **Workloads (Application and Data Security):** This pillar is concerned with securing the applications themselves, including the communication between microservices, APIs, and containers, whether they are hosted in a data center or a multicloud environment. It aims to prevent unauthorized access and lateral movement within application environments.

This three-pillar framework provides the strategic blueprint for Cisco's entire security portfolio, dictating how individual products are designed and, more importantly, how they must interoperate to deliver a holistic Zero Trust outcome. [Figure 8-2](#) shows an overview of the Cisco's Zero Trust Framework.

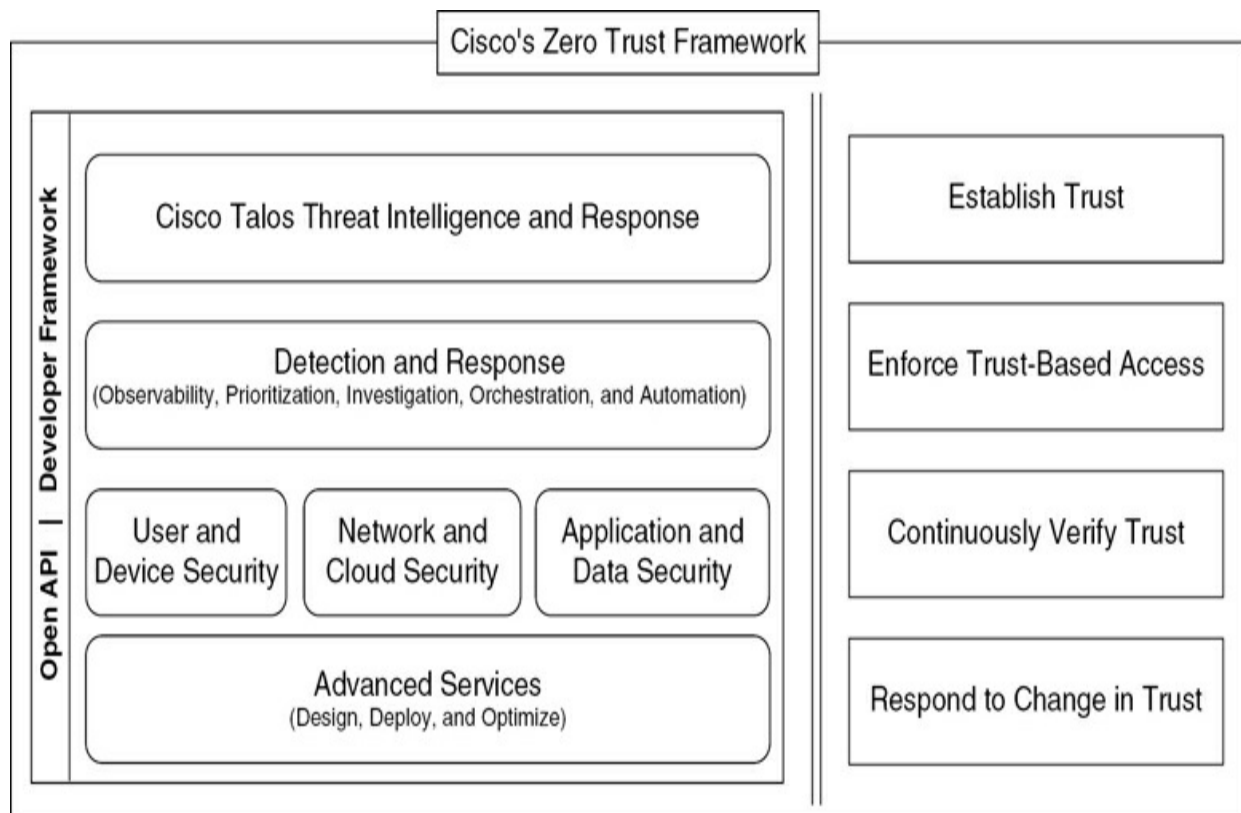


Figure 8-2 Cisco's Zero Trust Framework Overview

Figure 8-2 is divided into two main sections: The left side focuses on the foundational components and enablers (built on an Open API—Developer API Framework), while the right side outlines the iterative process for managing trust. The left section of Figure 8-2 represents the layered architecture that supports Zero Trust implementation. It starts from the bottom (foundation) and builds upward to advanced threat response capabilities.

Cisco Talos Threat Intelligence and Response is at the top; this integrates Cisco's global threat intelligence platform (Talos), which shares insights on emerging threats, vulnerabilities, and attack patterns. It enhances all layers by providing proactive intelligence to inform detection, response, and policy enforcement.

In Figure 8-2, you can also see the three core pillars you learned earlier, arranged horizontally (User & Device Security; Network & Cloud Security; and Application & Data Security). The Detection & Response (Observability, Prioritization, Investigation, Orchestration and Automation) layer provides tools for real-time monitoring and reaction. It leverages analytics for anomaly detection, flow analysis, and automated responses, ensuring threats are identified and mitigated quickly. Capabilities here include security orchestration, automation, and response (SOAR) and network detection and response (NDR).

The right side of the diagram in Figure 8-2 shows the Trust Management Process. This vertical stack outlines the dynamic, ongoing process for handling trust in a Zero Trust environment. It's a cycle that ensures security isn't a one-time check but a continuous evaluation:

- **Establish Trust:** The initial step verifies identity and context for users, devices, and requests at every access attempt. This uses tools like identity authorization, MFA, and SSO to confirm legitimacy before any access is granted.
- **Enforce Trust-Based Access:** After trust is established, adaptive policies apply least-privilege rules. This includes microsegmentation, policy audits, and change management to restrict access to only what's necessary, based on real-time context.
- **Continuously Verify Trust:** Trust isn't static; it's monitored ongoing

through anomaly detection, device posture assessment, and flow analytics. This approach ensures any changes in behavior or context (e.g., a device becoming compromised) are flagged immediately.

- **Respond to Change in Trust:** If trust levels drop (e.g., due to a detected threat), automated responses kick in. This leverages SOAR, threat intelligence from Talos, and incident response tools to isolate, remediate, or revoke access swiftly.

The Principle of Continuous Trusted Access

A critical and dynamic extension of the Zero Trust model is the principle of Continuous Trusted Access. This concept suggests that trust is not a static, binary state established at the moment of login. An initial successful authentication is necessary but not sufficient to guarantee security for the duration of a session. Trust must be continuously earned, verified, and reevaluated based on real-time contextual signals, as shown in [Figure 8-3](#).

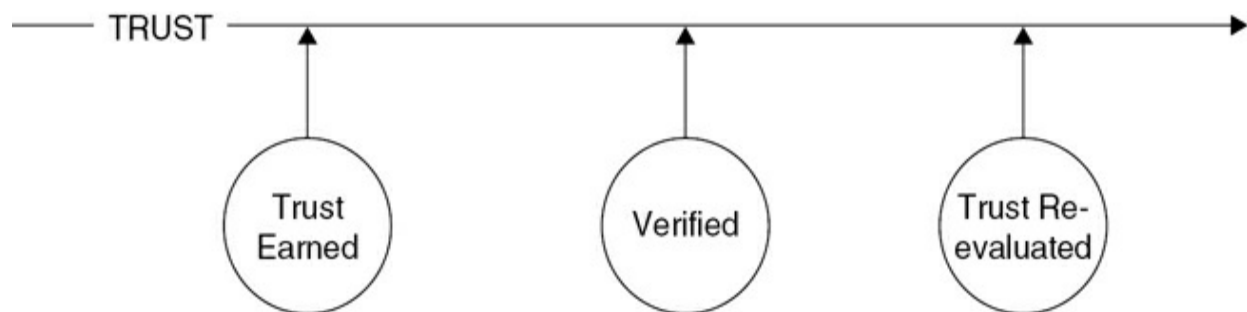


Figure 8-3 Trust Earned, Verified, and Reevaluated

This principle transforms security from a one-time gatekeeping function into an ongoing, dynamic process. The analogy of airport security is suitable: A passenger is checked at ticketing, at the security checkpoint, and again at the gate. Similarly, in a Continuous Trusted Access model, an access session is monitored for changes in context that might alter its risk profile. Examples of such contextual changes include

- A change in the device's security posture (e.g., its firewall being disabled)
- A user attempting to access a highly sensitive application at an unusual time or from a new geographic location

- Anomalous network behavior, such as a CCTV camera that typically receives connections suddenly initiating scans of the data center

When such deviations from a trusted baseline are detected, the system can dynamically adjust the trust level and enforce a new policy, such as requiring step-up authentication or terminating the session altogether. This principle of continuous verification is the conceptual foundation for Cisco Identity Intelligence, which provides the analytical engine required to detect these subtle shifts in context and risk.

The strategic decision to build a portfolio around the Zero Trust framework and the principle of Continuous Trusted Access provides a coherent narrative for Cisco's acquisition and integration roadmap. Although the market is saturated with the term *Zero Trust*, Cisco defines it with a clear, three-pillar structure that maps directly to its core product offerings. Duo is explicitly the solution for the Workforce pillar, ISE is the solution for the Workplace pillar, and Oort's technology underpins the entire model by enabling Continuous Trusted Access through its advanced threat detection and posture management capabilities. This demonstrates that the acquisitions of Duo and Oort were not merely opportunistic but were deliberate, strategic steps to build the technological capabilities required to deliver on a predefined architectural vision.

An Introduction to Cisco Duo

Cisco Duo is the centerpiece of Cisco's strategy for securing the Workforce pillar of Zero Trust. It has evolved from a best-of-breed MFA tool into a comprehensive access security platform. Its design philosophy is rooted in the belief that strong security can and should be user-friendly, a principle that has driven its wide adoption and serves as a key differentiator in the market.

From Traditional MFA to Phishing-Resistant Passwordless

Duo's capabilities are layered to provide a defense-in-depth approach to access security, addressing a wide spectrum of threats from simple credential theft to sophisticated phishing attacks.

Multifactor authentication is a foundational capability of Duo's ecosystem. However, Duo offers a diverse range of authentication methods to suit different user needs, device capabilities, and security contexts:

- **User-Friendly Methods:** The most adopted method is Duo Push, a real-time notification sent to the Duo Mobile app that users can approve with a single tap. Other simple methods include passcodes generated by the app (which work offline) or delivered via SMS and phone calls. However, it is recommended to prioritize phishing-resistant methods, as described in the next paragraph.
- **Phishing-Resistant Methods:** To combat the growing threat of MFA bypass attacks, Duo has heavily invested in phishing-resistant authenticators. Verified Duo Push enhances the standard push by requiring the user to enter a numeric code displayed on the login screen into the mobile app prompt, ensuring user attentiveness and mitigating "push fatigue" attacks. For the highest level of security, Duo supports the FIDO2/WebAuthn standard, allowing the use of device-native biometrics like Apple's Touch ID/Face ID and Windows Hello, as well as hardware security keys (e.g., YubiKeys). These methods are resistant to phishing because the cryptographic exchange is bound to the specific origin of the login request, making it impossible for a fake website to capture and replay the credential.

Overview of FIDO2 and WebAuthn

FIDO2 (Fast Identity Online 2) and WebAuthn together form a modern standard for secure, passwordless authentication on the web and beyond. FIDO2 is the broader framework developed by the FIDO Alliance to enable phishing-resistant authentication using everyday devices, while WebAuthn is a specific component of FIDO2 that provides the web browser API for implementing this authentication. Often referred to interchangeably or as FIDO2/WebAuthn, they work in tandem to replace traditional passwords with cryptographic credentials, improving security, privacy, and user experience. This standard addresses vulnerabilities in legacy methods like passwords or SMS one-time codes by ensuring credentials are never stored on servers and are resistant to common attacks.

Note

You can access the FIDO2 specification at <https://fidoalliance.org/fido2>.

FIDO2 has two primary specifications:

- **Web Authentication (WebAuthn):** This W3C standard defines the JavaScript API for web browsers to interact with authenticators. It handles the creation and use of credentials on the client side.
- **Client-to-Authenticator Protocol (CTAP):** Developed by the FIDO Alliance, this protocol governs communication between the client device (e.g., browser or OS) and the authenticator (e.g., a hardware key or built-in sensor). CTAP supports transports like USB, NFC, Bluetooth Low Energy (BLE), and more.

These components enable integration across devices and backward compatibility with earlier FIDO standards like U2F. Note that multidevice credential syncing (e.g., syncing across phones and laptops) is an implementation feature provided by specific authenticator vendors (such as Apple's iCloud Keychain or Google Password Manager), not an inherent capability of the FIDO2/WebAuthn standard itself.

WebAuthn is a W3C specification that defines an API for web applications to create and use public key-based credentials for user authentication. It's essentially the browser-facing part of FIDO2, extending the Credential Management API in browsers to support strong, attested credentials. The main purpose of WebAuthn is to enable passwordless, second-factor, or multifactor authentication in web apps by allowing the registration and assertion of public key credentials. It focuses on two main processes:

- **Registration:** A new public key credential is created and associated with the user's account on a specific Relying Party (e.g., a website).
- **Authentication:** The user proves possession of the credential by generating a signed assertion, verifying their presence and consent.

WebAuthn features public key credentials, where each credential consists of a key pair; the private key remains securely on the authenticator, while the public key is shared with the server. These credentials are scoped to a single

Relying Party to prevent misuse across different sites. The standard supports various authenticators, including platform authenticators built into devices such as phones or laptops, as well as roaming authenticators like external hardware keys (e.g., YubiKey). User verification options can be set as required, such as through biometrics, preferred, or discouraged based on the implementation needs.

During registration, attestation allows authenticators to provide proof of their properties, including model and security level, to the server, thereby enhancing overall trust in the system. User consent and privacy are prioritized, with all operations requiring explicit user approval, and the API being mediated by the browser to safeguard against unauthorized access. Overall, the specification is stable and widely deployed, with ongoing feedback and improvements managed through GitHub.

Case Study: Implementing FIDO2/WebAuthn with Cisco Duo for Passwordless Authentication at Cisco

Cisco has more than 130,000 full-time employees and contractors. To enhance security, reduce phishing risks, and improve user experience, Cisco adopted Duo Passwordless MFA, leveraging the FIDO2/WebAuthn standards. This implementation resulted in zero password-related incidents and a 93 percent reduction in authentication steps, demonstrating the effectiveness of passwordless solutions in a large-scale enterprise environment.

Note

You can gain additional insights about this case study at <https://www.cisco.com/site/us/en/solutions/cisco-on-cisco/duo-passwordless-mfa.html>.

Cisco operates in a highly dynamic threat landscape, where passwords have long been a weak link susceptible to phishing, credential stuffing, and breaches. With a large workforce of users accessing hundreds of applications and more than 170,000 devices, the company sought a Zero Trust approach to authentication. Traditional MFA methods, while effective, still relied on passwords as the primary factor, leading to user friction and administrative

overhead. FIDO2/WebAuthn emerged as a promising standard for passwordless authentication, using public key cryptography to enable secure, phishing-resistant logins via biometrics or security keys. Cisco's own Duo Security provided an ideal platform to integrate these standards, supporting FIDO2-compliant authenticators for seamless passwordless access.

There were many challenges:

- Passwords were vulnerable to attacks, contributing to potential data breaches and compliance risks in a Zero Trust model.
- Employees dealt with complex password requirements, frequent resets, and multistep MFA processes, leading to frustration and reduced productivity.
- Managing authentication for a global, diverse workforce across on-premises, cloud, and hybrid environments required a flexible, interoperable solution.
- Traditional methods were insufficient against advanced threats, necessitating phishing-resistant authentication.

Cisco Duo Passwordless integrates FIDO2/WebAuthn to enable passwordless authentication. Duo uses WebAuthn for browser-based public key credential creation and assertion, combined with FIDO2's Client-to-Authenticator Protocol for communication with authenticators. This approach supports platform authenticators (e.g., Apple's Touch ID, Windows Hello, Android biometrics) and roaming authenticators (e.g., YubiKey security keys).

Duo Passwordless allows for complete passwordless logins (no fallback) or hybrid modes. It supports Duo Push on mobile devices with biometric verification, remembered devices for up to seven days, and self-service management. Integration with Duo SSO federates SAML/OIDC applications, ensuring compatibility with identity providers like Microsoft Entra ID or Okta.

By requiring channel binding and cryptographic proofs, Duo prevents session hijacking and on-path attacks. This solution aligns with Cisco's Zero Trust framework, providing continuous verification without shared secrets.

Implementation Process

Cisco rolled out Duo Passwordless in phases, as shown in [Figure 8-4](#).



Figure 8-4 Cisco Passwordless Implementation Phases

The process shown in [Figure 8-4](#) starts with the planning and configuration phase. Cisco configured Duo SSO with existing directories and applied passwordless policies to pilot groups, enabling FIDO2-compatible methods like platform authenticators and security keys.

The second phase was user enrollment. Users enrolled via self-service portals, registering authenticators such as biometrics or hardware keys. Duo supported flexible enrollment from CSV imports or external directories.

In phase 3, Cisco integrated and federated key applications (e.g., Office 365) with Duo SSO, ensuring seamless logins. Phase 4 is the monitoring and rollout phase. Cisco used Duo’s Admin Panel for adoption tracking, authentication logs, and policy adjustments. This phased expansion minimized disruptions.

The last phase was the optimization process. During this implementation, Cisco highlighted change management, communication, and support, with only 1 percent of users needing assistance. Cisco also reported that there were zero (none) password-related incidents post-implementation, with enhanced phishing resistance via FIDO2. There was a 93 percent reduction in authentication steps, enabling single-gesture logins (e.g., biometric scans), boosting productivity and satisfaction.

Shortly after this deployment, Cisco experienced significant operational scale and visibility improvements. The system processed more than 5 million monthly access attempts across a diverse ecosystem of 170,000 devices, providing comprehensive real-time visibility into device health metrics, security posture, and potential risk indicators. This centralized monitoring

capability enabled IT teams to proactively identify and address security concerns before they escalated into incidents. Additionally, the passwordless authentication approach dramatically reduced the administrative burden associated with password management, virtually eliminating the time-consuming and resource-intensive process of handling password reset requests from users who had forgotten or needed to update their credentials.

Passwordless Implementation Lessons Learned and Recommendations

The following are the lessons learned that Cisco reported after its passwordless implementation:

- Start with pilot groups to refine policies and gather feedback.
- Prioritize user education on FIDO2 authenticators to drive adoption.
- Integrate with existing SSO for minimal disruption.
- Monitor metrics like adoption rates and authentication failures via tools like Duo's reporting.

This case study highlights how FIDO2/WebAuthn, through Cisco Duo, can transform enterprise security, offering a blueprint for organizations pursuing passwordless authentication.

Single Sign-On (SSO) and Duo

Duo provides a cloud-hosted SSO identity provider (IdP). It allows users to authenticate once through Duo (secured with MFA) and then gain seamless access to multiple integrated cloud applications (like Microsoft 365, Salesforce, and Webex) that support the SAML 2.0 standard, without reentering credentials for each service. This approach improves user productivity while centralizing access control and policy enforcement.

Passwordless Authentication

Representing a key strategic direction for Cisco, passwordless authentication

aims to eliminate the root cause of most breaches: the password itself. Duo enables a truly passwordless login experience where users can authenticate to applications using only a strong, phishing-resistant factor like a biometric verifier (Face ID, fingerprint) or a security key. This approach not only provides superior security but also dramatically improves the user experience by removing the friction of password management. As you learned, Cisco's own internal deployment of Duo Passwordless serves as a powerful testament to its benefits, resulting in a 93 percent reduction in authentication actions and zero password-related security incidents.

Device Trust and Health

Device trust and health is a core component of Duo's Zero Trust functionality. It provides administrators with complete visibility into every device (corporate-managed or personal devices) that is attempting to access protected applications. More importantly, it allows the creation of policies that enforce device health checks before granting access. These checks can verify that a device has an up-to-date operating system and browser, an enabled screen lock, active disk encryption, and a host firewall.

Note

If a device fails a check, Duo can guide the user through self-remediation steps, reducing the burden on IT support. This capability is tiered across Duo's plans, with more advanced checks available in higher tiers.

Duo's adaptive policies enable dynamic, context-aware access control decisions (moving beyond static, one-size-fits-all rules). Available in the Advantage and Premier tiers, these policies can evaluate a rich set of contextual signals in real time, including the user's role and group, geographical location, network trust (e.g., on-campus vs. public Wi-Fi), and the device's trust status. Based on this context and a real-time risk assessment powered by Cisco Identity Intelligence (with the integration of Oort), Duo can dynamically adjust the authentication requirements. For example, a low-risk login from a trusted device on the corporate network might be seamless, whereas a higher-risk attempt from an unknown device in a new country could trigger a prompt for a strong, phishing-resistant authenticator.

[Table 8-1](#) provides a clear comparison of the features available across Duo's primary commercial editions, which is a good resource for strategic planning and budgeting.

Table 8-1 Duo's Feature Comparison Across Levels

Feature	Duo Free	Duo Essentials	Duo Advantage	Duo Premier
MFA (Standard)	Yes (10 users)	Yes	Yes	Yes
Single Sign-On (SSO)	No	Yes	Yes	Yes
Passwordless	No	Yes	Yes	Yes
Phishing-Resistant MFA	No	Yes	Yes	Yes
Trusted Endpoints	No	Yes	Yes	Yes
Device Health Checks	No	No	Yes	Yes
Adaptive Access Policies	No	No	Yes	Yes
Risk-Based Authentication	No	No	Yes	Yes
Cisco Identity Intelligence (ITDR/ISPM)	No	No	Yes	Yes
Duo Network Gateway (VPN-less Access)	No	No	No	Yes

Getting Familiar with Cisco Duo via the Free Tier

Cisco Duo's free tier is an excellent entry point to explore its capabilities hands-on, especially for individuals, small teams, or those evaluating the tool before committing to paid plans. Accessing this tier allows you to experience core functionalities without financial risk, build familiarity with the interface, and test integrations in a real-world setting, all while addressing basic security needs.

The Duo Free tier is a perpetual, no-cost plan priced at \$0 per user per month, making it accessible for small-scale use without any time limits (unlike the 30-day free trial available for premium). It supports up to 10 users, focusing on essential MFA to safeguard applications and credentials from attacks like phishing, malware, and ransomware.

The Duo Cloud-Native Platform and On-Premises Components

Duo's architecture is designed for flexibility and scalability, combining a robust cloud-native service with essential on-premises components that enable integration with legacy infrastructure.

At its core, Duo is a Software-as-a-Service (SaaS) platform. This cloud-native architecture provides inherent benefits, including high availability, automatic updates, and massive scalability, allowing it to support organizations ranging from small businesses to the largest global enterprises without requiring customers to manage complex server infrastructure.

Duo Authentication Proxy

Many of Duo's application integrations do not require any local components. However, certain services do require a local Authentication Proxy service. The Authentication Proxy is a lightweight software service installed on a server within the customer's network. It functions as a versatile bridge between legacy systems and Duo's modern cloud service. It can act as a RADIUS server to add Duo MFA to VPNs (like Cisco ASA and Palo Alto) and network hardware, or as an LDAP proxy to secure on-premises

applications that authenticate against directories like Active Directory. This proxy is the key that unlocks Duo protection for a vast ecosystem of technologies that do not natively support modern authentication protocols like SAML or OIDC. [Figure 8-5](#) shows a high-level architecture of the Duo Authentication Proxy.

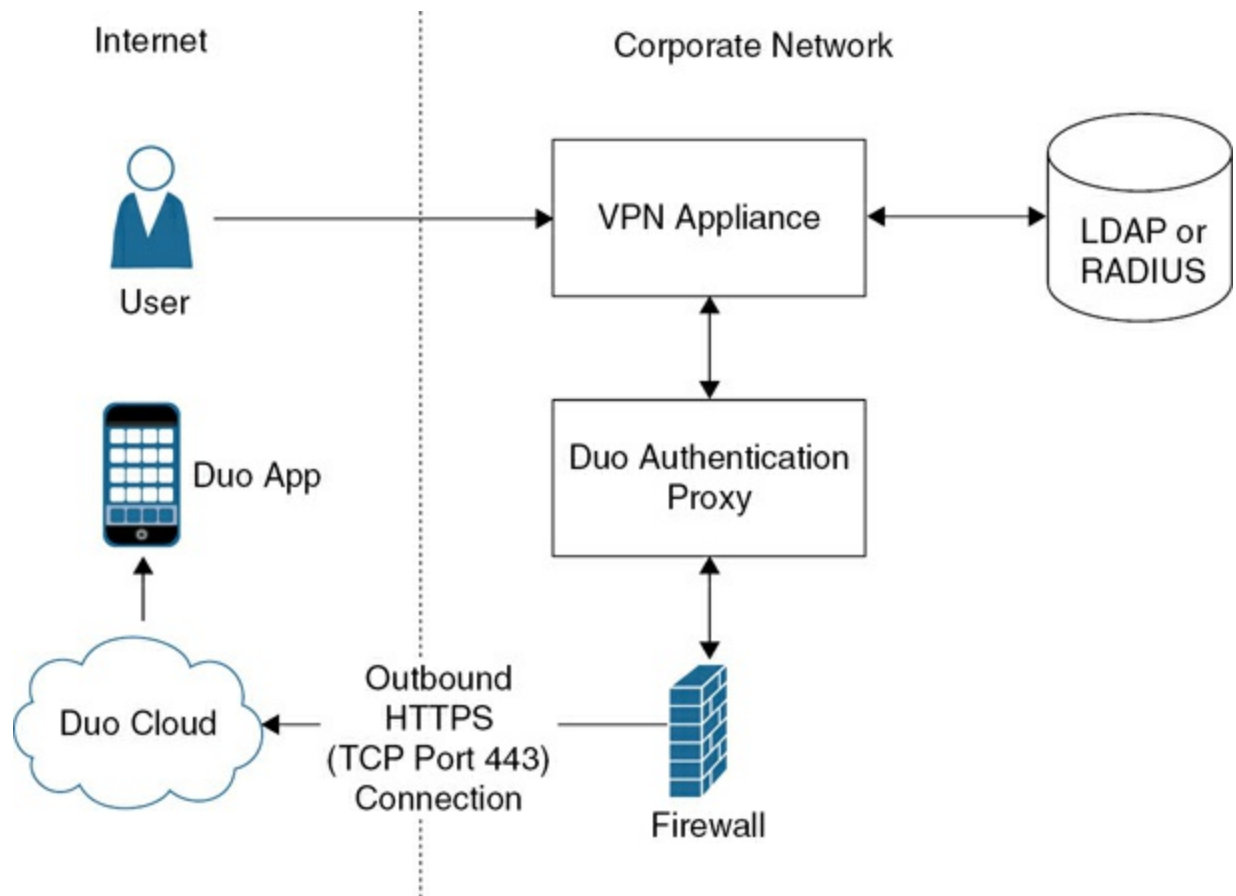


Figure 8-5 Duo Authentication Proxy

[Figure 8-5](#) illustrates the network diagram for integrating Duo with a RADIUS-based application or LDAP service, such as a VPN appliance, using the Duo Authentication Proxy. This setup inserts the proxy between the VPN device and the primary authentication server (e.g., Active Directory/LDAP or another RADIUS server), enabling primary credential validation followed by secondary Duo MFA via the cloud. The proxy runs on-premises, ensuring secure communication without exposing primary credentials to the Internet.

Let's go over a step-by-step breakdown of the process. This aligns with Duo's official RADIUS integration guidelines and assumes users are

preenrolled in Duo (e.g., via directory sync or manual addition).

1. Primary Authentication Initiated to the Application or Service:

The process begins when a user attempts to log in to the VPN appliance (or similar RADIUS client) from the Internet side, entering their username and primary password. The user may also have the Duo Mobile app on their device for secondary authentication. This step initiates the overall authentication request.

2. Application or Service Sends Authentication Request to the Duo Authentication Proxy:

The VPN appliance forwards a RADIUS Access-Request packet containing the user's credentials to the on-premises Duo Authentication Proxy. The proxy is configured as the RADIUS server for the appliance (e.g., listening on UDP port 1812), acting as an intermediary to handle both primary and secondary authentication without requiring changes to the user's login experience.

3. Primary Authentication Using Active Directory or RADIUS:

The Duo Authentication Proxy validates the user's primary credentials against the configured back-end server, such as an Active Directory/LDAP directory or another RADIUS server. This step checks the username and password for validity. If primary authentication fails, the process halts with a rejection; if it succeeds, the proxy proceeds to secondary authentication.

4. Duo Authentication Proxy Establishes Connection to Duo Security over TCP Port 443:

Upon successful primary authentication, the proxy initiates an outbound HTTPS connection (over TCP port 443) through the corporate firewall to Duo's cloud service. This step uses the integration key (*ikey*), secret key (*skey*), and API hostname specified in the proxy's configuration. The firewall must allow this outbound traffic to Duo's, ensuring secure, encrypted communication without inbound ports needing to be opened.

5. Secondary Authentication via Duo Security's Service:

Duo's cloud service processes the secondary authentication request and prompts the user for MFA. This prompt could involve an automatic Duo Push notification to the user's mobile app, a phone call, SMS passcode, or hardware token (based on the user's enrolled methods and proxy settings). The user approves or responds to the challenge (e.g., by

tapping Approve in the app). Alternatively, users can leverage append mode by concatenating their password with a comma and their desired Duo factor at login time (e.g., typing **password,push** or **password,sms** directly into the password field), allowing the proxy to automatically select and trigger that specific authentication method without additional prompts. This step adds phishing-resistant protection without sharing primary credentials with Duo.

6. Duo Authentication Proxy Receives Authentication Response:

Once the user completes the MFA challenge, Duo's cloud sends a success or failure response back to the proxy over the established HTTPS connection. The proxy logs the event and, if configured with *failmode=safe*, may allow fallback to primary authentication only if Duo's service is unreachable; otherwise, in secure mode, both factors are mandatory.

7. Application or Service Access Granted: If it succeeds, the Duo Authentication Proxy returns a RADIUS Access-Accept message to the VPN appliance, optionally passing through attributes like group memberships from the primary server. The VPN appliance then grants the user secure access to the corporate network. If any step fails, access is denied, and the event is logged for auditing.

Duo Network Gateway (DNG)

The Duo Network Gateway is another on-premises component, typically deployed using Docker containers. It provides secure, VPN-less remote access to specific internal resources, such as internal web applications, SSH servers, and RDP desktops. Instead of granting broad network access like a traditional VPN, the DNG brokers access on a per-application basis, aligning perfectly with Zero Trust's principle of least privilege.

More Than 400,000 VPN Connections per Month

The "Cisco on Cisco" case study at the following link provides a powerful metric for the DNG's impact, showing that its deployment eliminated the need for more than 410,000 VPN connections each month, significantly improving user experience and reducing infrastructure load:

<https://resources.duo.com/explore/assets/cisco?contentType=case-study>.

Duo provides a comprehensive suite of developer tools, including RESTful APIs (Admin API for management, Auth API for authentication, Device API for trusted devices) and software development kits (SDKs) for various languages (Python, Java, .NET, and so on). This offering demonstrates the platform's openness and allows organizations to embed Duo's security capabilities deeply into any custom-built application, ensuring that no part of the IT estate is left unprotected.

Figure 8-6 shows a high-level architecture of a Cisco Duo Network Gateway deployment.

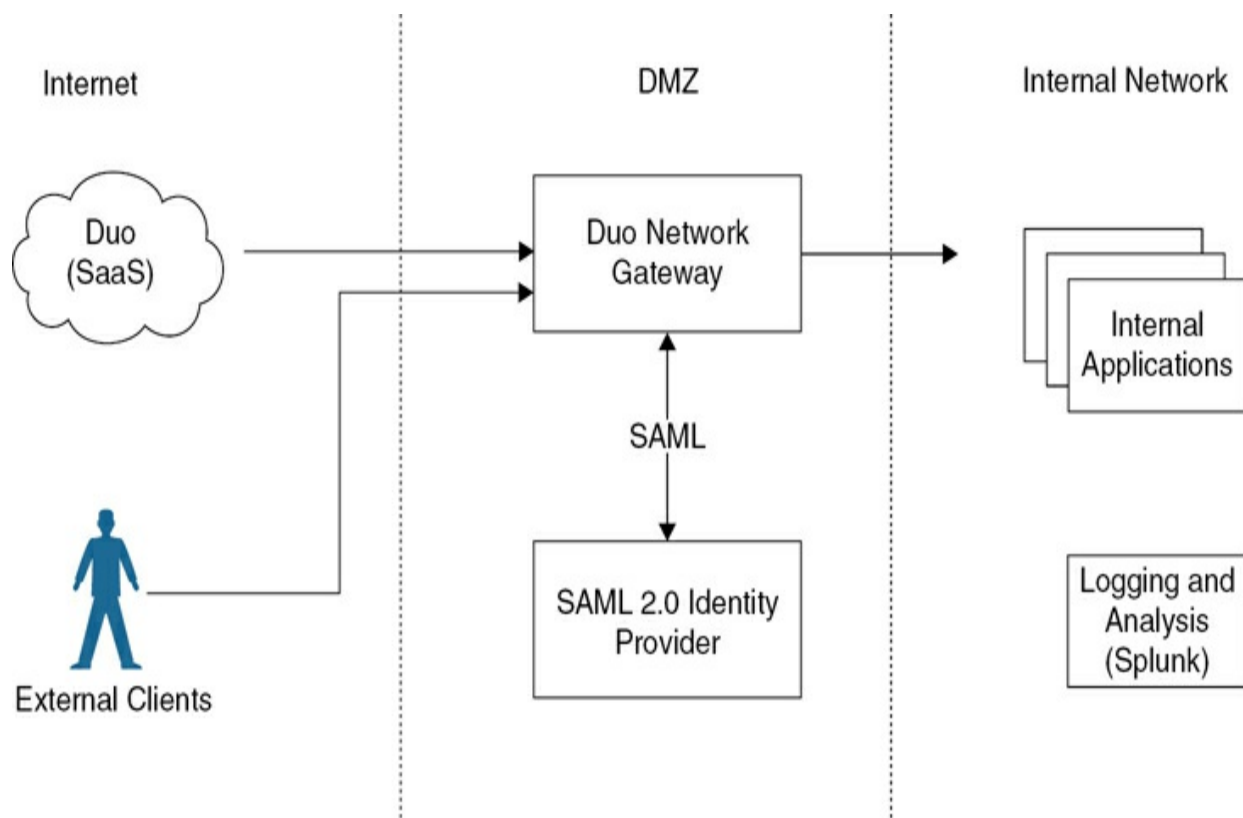


Figure 8-6 Cisco Duo Network Gateway

Figure 8-6 is divided into three zones: Internet, DMZ (Demilitarized Zone), and Internal Network. This figure illustrates how external clients gain protected access to on-premises assets through layered authentication and proxying. DNG acts as a reverse proxy in the DMZ, integrating SAML for primary authentication, Duo SaaS for MFA, and logging for monitoring. This

setup supports features like passwordless authentication (e.g., via FIDO2/WebAuthn passkeys) and aligns with Zero Trust principles by verifying every access request without exposing internal networks directly to the Internet.

Users or devices are accessing the system from outside the corporate network, typically via a web browser or client applications like DuoConnect or Duo Desktop. The Duo Network Gateway can be deployed as a Docker container. It serves as the entry point for external traffic, proxying requests to internal resources. Positioned under the DNG and connected via a bidirectional SAML communication, there is an on-premises (or it can also be a cloud-based) IdP (such as Active Directory Federation Services [AD FS], Okta, OneLogin, or Duo Single Sign-On). The DNG uses SAML 2.0 for primary authentication, exchanging metadata (e.g., Entity ID, ACS URL, SLO URL) and signing certificates to verify user identities before invoking Duo MFA.

Note

The DNG can generate logs via Syslog that can be forwarded to Splunk for real-time analysis, auditing, and compliance.

The Cisco Duo Identity and Access Management (IAM) Platform

Throughout the years, the Cisco Duo platform has transformed from a supplementary access security tool into a full-stack, security-first IAM solution. One of the foundation components of the Duo IAM platform is the native Duo Directory. This component allows Duo to act as a standalone identity provider, capable of storing and managing user identities (usernames, attributes, roles) directly within its cloud service. This capability is a fundamental shift, because it reduces or eliminates the dependency on external directories like Microsoft Active Directory or Okta Universal Directory, offering a fully self-contained IAM solution.

Duo Identity Routing Engine

One of the most powerful and strategically astute features of the Duo platform is the Identity Routing Engine. This engine allows Duo to function as a sophisticated identity broker. An organization can place Duo in front of its existing, heterogeneous identity solutions. For instance, many organizations might include Entra ID, Okta, and on-premises AD simultaneously due to mergers and acquisitions, as illustrated in [Figure 8-7](#).

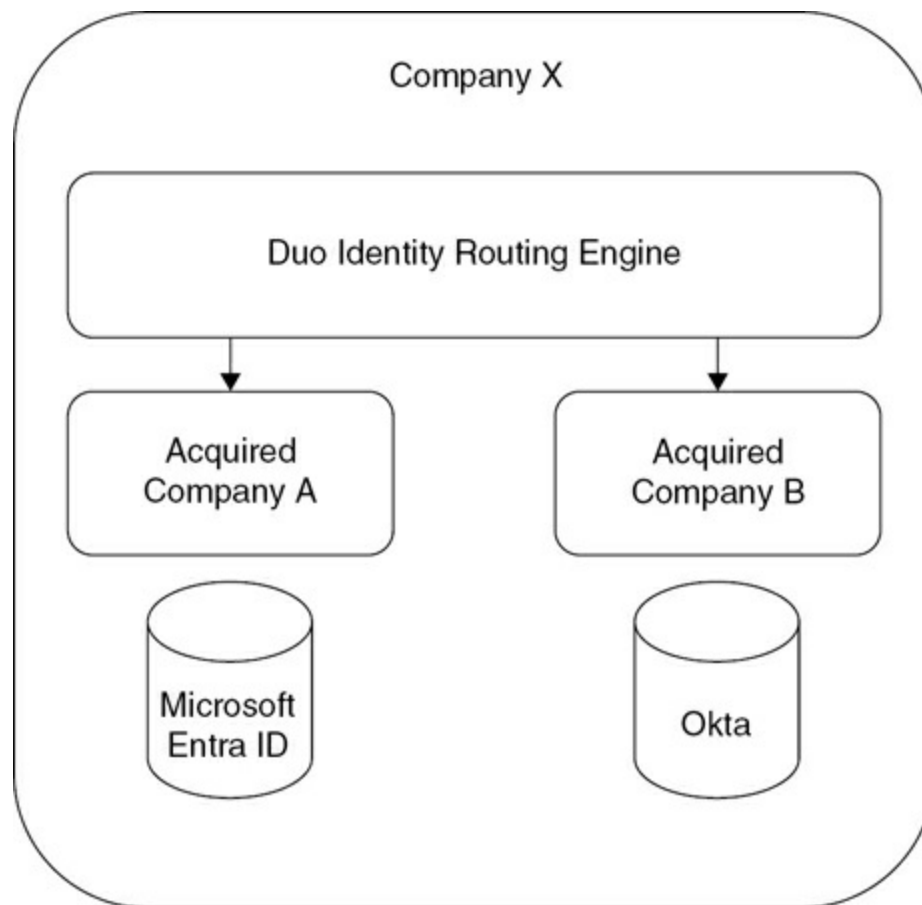


Figure 8-7 Using the Duo Identity Routing Engine

[Figure 8-7](#) illustrates the architecture of Cisco Duo’s Identity Routing Engine within Company X, a parent organization managing multiple acquired subsidiaries with diverse IdPs (Microsoft Entra ID and Okta). Duo routes authentication requests to the appropriate IdP based on user attributes, such as email domains, enabling seamless integration across heterogeneous systems often resulting from mergers and acquisitions.

Hardware-Free Phishing Resistance

The Duo IAM platform doubled down on phishing resistance by introducing Proximity Verification. This feature uses Bluetooth Low Energy (BLE) to cryptographically verify that the user's mobile device (running the Duo Mobile app) is in physical proximity to the laptop or desktop from which the login attempt is originating. This feature prevents remote on-path phishing attacks, where an attacker on another continent tricks a user into approving a push notification. It provides a strong, phishing-resistant authentication factor without the cost and logistical overhead of deploying physical hardware security keys to the entire workforce.

The launch of Duo IAM represents a direct assault on the conventional IAM market. Cisco is leveraging Duo's sterling reputation for security and usability to make a compelling argument. Traditional IAM solutions were born in an era focused on productivity and access enablement but have fundamentally failed to keep pace with the modern threat landscape in today's AI-driven world. They often treat security capabilities like adaptive MFA and threat detection as complex, expensive, and bolted-on afterthoughts. In contrast, Cisco introduced Duo IAM as "security-first," with advanced protections like end-to-end phishing resistance built in as foundational, out-of-the-box features.

The Identity Routing Engine was a good strategic move. It dramatically lowers the barrier to entry for prospective customers by allowing them to immediately benefit from Duo's security layer across their entire organization, irrespective of their existing IdP vendor. This approach creates a powerful "land and expand" motion. Once Duo is established as the universal "secure front door," the pathway to gradually migrating users and applications to the native Duo Directory becomes significantly simpler, creating a long-term opportunity for Cisco to displace legacy IdPs and become the central identity authority for the enterprise.

More on Cisco Identity Intelligence

As we discussed earlier in this chapter, the acquisition and integration of Oort represented one of the most forward-looking components of Cisco's modern identity strategy. Oort, which now powers the Cisco Identity Intelligence engine, provides the proactive detection and response capabilities that are essential in a threat landscape where prevention alone is insufficient. It acts

as the “brain” of the identity fabric, continuously analyzing identity behavior to find weaknesses and detect active threats that can bypass traditional access controls.

Oort’s technology was centered on two emerging and complementary security disciplines: identity threat detection and response (ITDR) and identity security posture management (ISPM).

Summary

In this chapter, we covered the fundamental concepts of identity and access management. Now you understand why many people refer to IAM as “the new firewall.” We explored the foundational components of IAM, including authentication (verifying identity), authorization (granting access), and accounting (logging activities for auditing and incident response).

We described the strategic shift towards passwordless authentication to protect against cyber attacks like AI-driven spear phishing and credential stuffing, and how this shift improves both security and user experience. We also covered advanced IAM solutions such as single sign-on, multifactor authentication, identity governance and administration, and privileged access management.

In addition, we covered Cisco’s unified identity fabric, powered by Cisco Duo, Cisco Identity Services Engine, and Cisco Identity Intelligence. As a result, you gained an understanding of Cisco’s Zero Trust philosophy, which reinforces these technologies by removing implicit trust and continuously verifying every access attempt across your workforce, workplace, and workloads. You also learned about the dynamic principle of Continuous Trusted Access, where trust is constantly reevaluated based on real-time contextual signals. You also learned about Duo’s architectural components like the Authentication Proxy and Network Gateway (DNG), as well as the innovative Identity Routing Engine and Proximity Verification for enhanced security.

References

- Cisco Zero Trust Architecture Guide:

<https://www.cisco.com/c/en/us/solutions/collateral/enterprise/design-zone-security/zt-ag.html>

- FIDO Alliance, "FIDO2 Specifications": <https://fidoalliance.org/fido2>
- W3C, "Web Authentication (WebAuthn) Level 2": <https://www.w3.org/TR/webauthn-2/>
- Cisco, "Cisco on Cisco: Duo Passwordless MFA Implementation Case Study": <https://www.cisco.com/site/us/en/solutions/cisco-on-cisco/duo-passwordless-mfa.html>
- Cisco Duo, "Cisco Duo Customer Case Study": <https://resources.duo.com/explore/assets/cisco?contentType=case-study>
- Cisco Duo Documentation: <https://duo.com/docs>
- Cisco Identity Services Engine (ISE) Documentation: <https://www.cisco.com/c/en/us/products/security/identity-services-engine/index.html>
- NIST Special Publication 800-63-3, "Digital Identity Guidelines": <https://pages.nist.gov/800-63-3/>
- NIST Special Publication 800-207, "Zero Trust Architecture": <https://csrc.nist.gov/publications/detail/sp/800-207/final>

Chapter 9. Security: Cisco Umbrella and Cisco AI Defense

Enterprise IT is being reshaped by two concurrent and powerful transformations. The first is the mass migration to the cloud during the last couple of decades, and the second is the dawn of the generative AI era. The first revolution is driven by the adoption of Software-as-a-Service (SaaS) applications and the rise of the hybrid workforce, as you have learned throughout this book. Security can no longer be predicated on a model of a fortified corporate data center, because users, devices, and data are now distributed globally. We needed a new approach to security, one that is delivered from the cloud, follows the user, and provides consistent protection regardless of location.

The second revolution continues to unfold with unprecedented speed. AI has transitioned from a theoretical concept to a transformative business tool, promising to unlock new levels of productivity and innovation. However, this rapid adoption has also introduced a new and complex attack surface. Enterprises now face a dual challenge: mitigating the risks of employees using third-party AI tools (aka *Shadow AI*) and securing the in-house AI applications they are developing and deploying. Traditional cybersecurity solutions (designed for a pre-AI world) are not equipped to address newer threats like prompt injection, data poisoning, and algorithmic vulnerabilities.

In this chapter you will learn a comprehensive modern security strategy to address both of these transformations through a unified, network-centric approach. This strategy requires a two-pronged defense: first, securing the access to the cloud and the Internet, and second, securing the development and use of AI itself. In this chapter, we will provide an exhaustive examination of two cornerstone solutions in the Cisco Security portfolio that

are designed to meet these challenges: Cisco Umbrella and the newly unveiled Cisco AI Defense. The analysis will demonstrate that Cisco's overarching strategy is to leverage the network as the ultimate point of visibility and control, providing a consistent security fabric for both the established challenges of the cloud era and the emerging frontiers of the AI revolution.

From OpenDNS to Cisco Umbrella

The story of Cisco Umbrella is a compelling case study. Its origins lie with a company (OpenDNS) that fundamentally understood the power of the DNS as a ubiquitous control point for Internet traffic. This foundation, built on a massive global data pipeline, proved to be the essential ingredient for building a predictive, cloud-native security platform.

OpenDNS was founded in 2006. The company's main goal was to provide a faster, safer, and more reliable recursive DNS service for both home and business users. OpenDNS rapidly built a massive global user base by offering a free alternative to the often slow and unreliable DNS services provided by Internet service providers (ISPs). The company's user base was not just a customer list; it was the source of an immense and diverse stream of real-time data on global Internet activity—a strategic asset whose full value would become apparent over time.

The company's first significant move into the enterprise security market came in 2009 with the launch of OpenDNS Enterprise. This offering introduced enterprise-grade features such as shared management, audit logging, expanded malware protection, and customizable block pages. However, the true precursor to the modern platform arrived in 2012 with the introduction of the Umbrella service. Umbrella was specifically designed to address a growing security gap: the protection of mobile employees and their roaming devices (such as laptops and smartphones) when they were operating beyond the confines of the corporate network and its traditional security stack.

Cisco acquired OpenDNS in 2015. The acquisition of OpenDNS provided a proven cloud-native security platform, a team of talented researchers and engineers with deep expertise in DNS and data science, and most

importantly, one of the largest and most diverse real-time Internet traffic datasets in the world.

Following the acquisition, Cisco renamed OpenDNS Umbrella as Cisco Umbrella. The popular consumer and home products, which continued to provide DNS-based content filtering and parental controls, retained the OpenDNS brand.

In the years following the acquisition, Cisco has systematically expanded Umbrella's capabilities, transforming it from a powerful DNS security tool into a comprehensive, multifunction security platform. This evolution has mirrored the broader industry shift toward service consolidation and cloud-native architectures.

Secure Internet Gateway and Secure Access Service Edge

Umbrella's feature set grew to encompass the core functions of a Secure Internet Gateway (SIG). This included the addition of a full proxy Secure Web Gateway (SWG) for deeper inspection of web traffic, a cloud-delivered firewall (CDFW) for nonweb traffic, and cloud access security broker (CASB) functionality to discover and control SaaS applications.

This collection of services positioned Umbrella as a leading security service edge (SSE) solution, which is the security-focused feature set of the secure access service edge (SASE) framework. As the core security component of Cisco's SASE architecture, Umbrella now provides the unified stack of security services that are converged with Cisco's SD-WAN networking solutions and Cisco Secure Access to deliver secure, optimized access for users anywhere. [Figure 9-1](#) shows a high-level architecture of some of the capabilities of Cisco Umbrella and other security components.

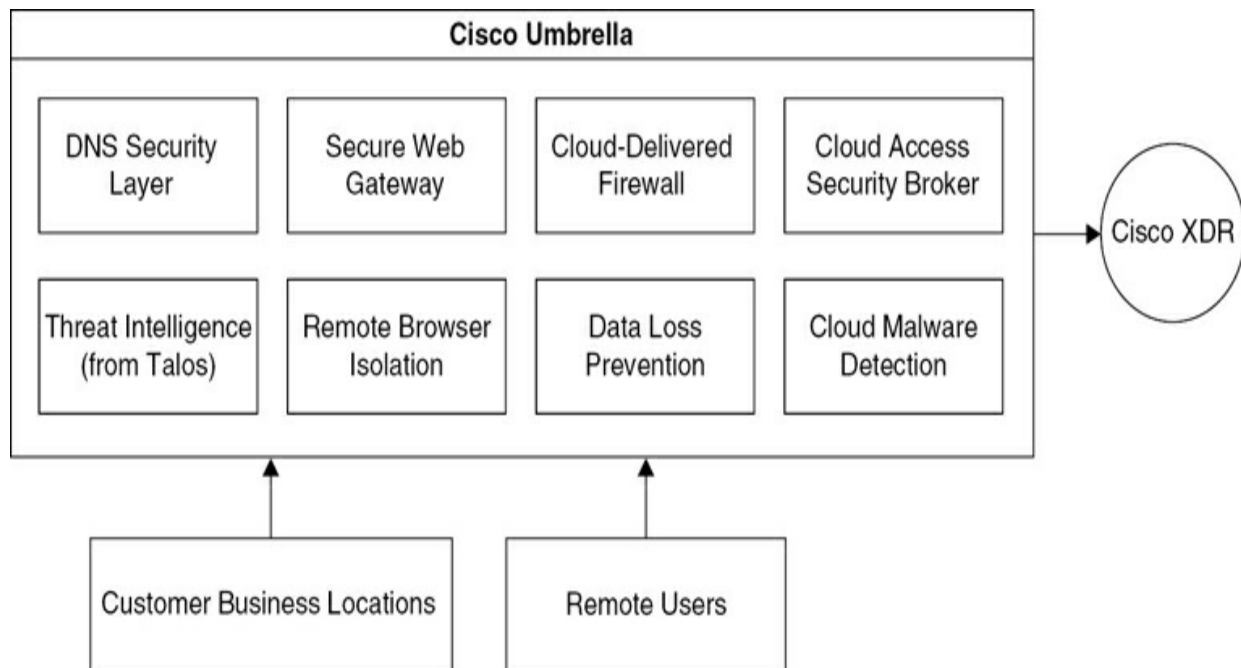


Figure 9-1 Cisco Umbrella and Other Components of the Broader Cisco Security Ecosystem

Figure 9-1 illustrates Cisco Umbrella's role as a cloud-delivered security platform and its integration into the broader Cisco security ecosystem. Cisco Umbrella provides multiple layers of protection for both on-site business locations and remote users, ensuring consistent security regardless of where users connect from. It begins with a DNS security layer, which blocks access to malicious domains at the DNS level before connections are established. The Secure Web Gateway inspects and filters web traffic, enforcing acceptable use policies and blocking malicious content. A cloud-delivered firewall provides network-level access control without requiring on-premises appliances, while the cloud access security broker monitors and manages the use of cloud applications to prevent data exposure and shadow IT risks.

Cisco Umbrella also incorporates threat intelligence from Cisco Talos, which continuously updates Umbrella with the latest information on emerging threats. Remote Browser Isolation allows users to safely access risky websites by executing browsing sessions in a remote environment and streaming only safe content to endpoints. The data loss prevention (DLP) feature inspects outbound traffic to prevent sensitive data from leaving the organization, helping to meet compliance requirements, while cloud malware detection identifies and blocks malicious files in real time through advanced

analysis techniques such as sandboxing.

All of this telemetry and protection data is fed into Cisco Extended Detection and Response (XDR), which correlates information from multiple Cisco security tools, enabling unified visibility, faster threat detection, and automated response. Together, these components form a comprehensive, cloud-native security solution that protects users everywhere and strengthens an organization's overall security posture.

An Analysis of Umbrella's Converged Functions

The power of the Umbrella platform lies in its layered, multifunction architecture. It intelligently steers traffic to the appropriate inspection engine based on the protocol and the assessed risk, optimizing for both security efficacy and performance.

- **DNS-Layer Security:** This is the foundational first line of defense. By inspecting every DNS query, Umbrella can block requests to malicious or unwanted destinations before an IP connection is ever established. This method of enforcement is incredibly efficient, capable of stopping the vast majority of threats (including malware, ransomware, phishing, and botnet command-and-control callbacks) across all ports and protocols with negligible latency. This initial filter significantly reduces the volume of traffic that requires deeper, more resource-intensive inspection.
- **Secure Web Gateway (SWG):** For web traffic (HTTP/HTTPS on ports 80 and 443) destined for domains that are not definitively known to be malicious but are deemed risky or uncategorized, Umbrella's proxy routes the connection through its cloud-based SWG. The SWG acts as a full proxy, providing deep packet inspection and a rich set of controls. Its capabilities include full visibility into web traffic, granular URL and application-level controls, SSL/TLS decryption for inspecting encrypted traffic, antivirus scanning, and advanced malware protection through file sandboxing with Cisco Secure Malware Analytics (formerly Threat Grid). It also enables comprehensive web content filtering based on dozens of categories.
- **Cloud-Delivered Firewall (CDFW):** This feature provides visibility

and control for all other Internet-bound traffic, regardless of the port or protocol. By establishing an IPsec tunnel from a network device (like an SD-WAN router) to the Umbrella cloud, organizations can enforce firewall policies at the cloud edge. The CDFW allows for policy enforcement based on Layer 3/4 rules (IP addresses, ports, protocols) and Layer 7 application-aware rules. This setup effectively replaces the need for traditional firewall appliances at every branch office, providing consistent policy enforcement and logging for all outbound traffic.

- **Cloud Access Security Broker (CASB):** This feature discovers and reports on the cloud applications being used across the organization, helping IT teams identify Shadow IT. Security administrators can view risk information for each application and create policies to block or allow specific apps or app categories, thereby better managing cloud adoption and reducing the risk of data leakage.
- **Data Loss Prevention (DLP) and Remote Browser Isolation (RBI):** Umbrella's SSE capabilities are further enhanced by advanced security services. The inline DLP engine analyzes sensitive data (such as PII, PHI, PCI data, and other regulated content) to ensure compliance with security policies and regulatory frameworks. The DLP engine provides visibility and control, blocking the exfiltration of confidential information before it leaves the organization. For situations where users need to access potentially risky websites for legitimate business purposes, remote browser isolation (RBI) provides an "air gap." RBI executes the web session in a disposable container in the cloud and streams a safe rendering of the page to the user's browser, isolating their device from any potential browser-based threats like malware or exploits.

This tiered architectural methodology is a great design principle. The fast and efficient DNS layer serves as the initial, broad-spectrum filter. This allows the more computationally expensive services, such as the full proxy SWG with SSL decryption and the CDFW, to be applied more selectively to the smaller subset of traffic that requires deeper inspection. This intelligent routing of traffic optimizes the trade-off between security and performance, ensuring robust protection without imposing a significant negative impact on the user experience. [Table 9-1](#) summarizes the core components of the Cisco Umbrella SASE architecture.

Table 9-1 Core Components of the Cisco Umbrella SASE Architecture

Component	Primary Function	Traffic Inspected	Key Use Case
DNS-Layer Security	Blocks malicious destinations before a connection is made	All DNS queries (UDP/TCP Port 53)	First line of defense against malware, phishing, and C2 callbacks
SGW	Provides deep inspection and control for web traffic	Web traffic (HTTP/HTTPS Ports 80, 443)	Full visibility, URL/content filtering, malware sandboxing for risky sites
CDFW	Provides visibility and control for nonweb traffic	All nonweb Internet traffic (all ports and protocols)	Consistent L3/L4 and L7 policy enforcement for direct Internet access
CASB	Discovers and controls the use of cloud (SaaS) applications	Web and application traffic	Uncovering Shadow IT, blocking unsanctioned apps, reducing cloud risk
DLP	Prevents the exfiltration of sensitive data	Data in motion (inline inspection)	Protecting intellectual property and ensuring data compliance
RBI	Isolates high-risk browsing sessions from the endpoint	Web traffic to potentially risky or uncategorized sites	Safe access to necessary but untrusted websites without risk of malware infection

Predictive Security with AI, ML, and Talos

The efficacy of any modern security solution is directly proportional to the quality and scale of the data it analyzes. Cisco Umbrella's primary differentiator is not merely a single feature but its underlying intelligence architecture, which leverages one of the world's largest security datasets to power a predictive engine. This engine combines the scale of machine learning (ML) with the expertise of human researchers to identify and block threats before they can cause harm, representing a fundamental shift from a reactive to a proactive security posture.

The Power of Big Data: A Global View of the Internet

The foundation of Umbrella's intelligence is the sheer volume and diversity of its data. By operating a global recursive DNS network, Umbrella processes billions of Internet requests every day, originating from more than 30,000 organizations and millions of individual users across 190 countries. This provides an unparalleled, real-time view into the Internet's activity, capturing the spatial and temporal relationships between every domain name, IP address, malware files, and networks.

Unlike solutions that rely on telemetry from a limited set of customers or specific threat vectors (like email or endpoints), Umbrella's DNS-level view captures the very first step of nearly every Internet connection, regardless of the port or protocol being used. This breadth of visibility allows the system to uncover patterns and detect anomalies that would be invisible to more narrowly focused tools.

The core of Umbrella's predictive engine is the application of sophisticated analytical techniques to its massive data stream. A team of security researchers, data scientists, and mathematicians develops and maintains a suite of statistical and machine learning models that continuously analyze Internet activity to automatically score and classify data, detect anomalies, and uncover both known and emergent threats.

This solution is fundamentally proactive. Traditional security methods are

often reactive. They require a sample of an attack (such as a malware file or a malicious URL) to create a signature or a rule for blocking. This means an organization must be victimized first for protection to be developed.

Umbrella's methodology is more modern. Its models are learning from live Internet activity patterns and can identify the infrastructure that attackers are staging for future campaigns. Similar to how an e-commerce platform learns from shopping patterns to recommend products, Umbrella learns from Internet activity patterns to automatically identify attacker infrastructure being staged for the next threat. In this way, Umbrella can block malicious domains and IPs before they are even used in an attack, effectively moving protection to the earliest possible point in the attack chain.

The power of this machine learning approach is most evident in its ability to detect threats that are specifically designed to evade traditional security controls. Two examples are domain generation algorithms (DGAs) and command-and-control (C2) callbacks.

Domain Generation Algorithms (DGAs)

Malware often uses DGAs to maintain communication with its operators while evading simple blacklists. A DGA is an algorithm embedded within the malware that generates hundreds or thousands of pseudorandom domain names per day. The attacker registers only one or two of these domains, which the malware then attempts to contact to receive commands. Because the list of potential domains is vast and changes constantly, blocking them all manually is nearly impossible.

Cisco Umbrella employs a sophisticated, multifaceted approach to detect these domains. Its statistical models analyze the linguistic and structural characteristics of requested domain names. Features used in this analysis include character frequency distribution, the presence of meaningful words or substrings (by checking against a massive set of dictionaries in various languages), and the overall "randomness" of the domain string.

Based on these features, the model calculates a score that determines the likelihood of the domain being algorithmically generated. This capability allows Umbrella to block entire families of DGA-based malware without needing to see every individual domain. This technology is backed by

significant research and development, as evidenced by patents covering the detection of DGA activity through the correlation of DNS request patterns and hostname-based classifiers. This concept is illustrated in [Figure 9-2](#).

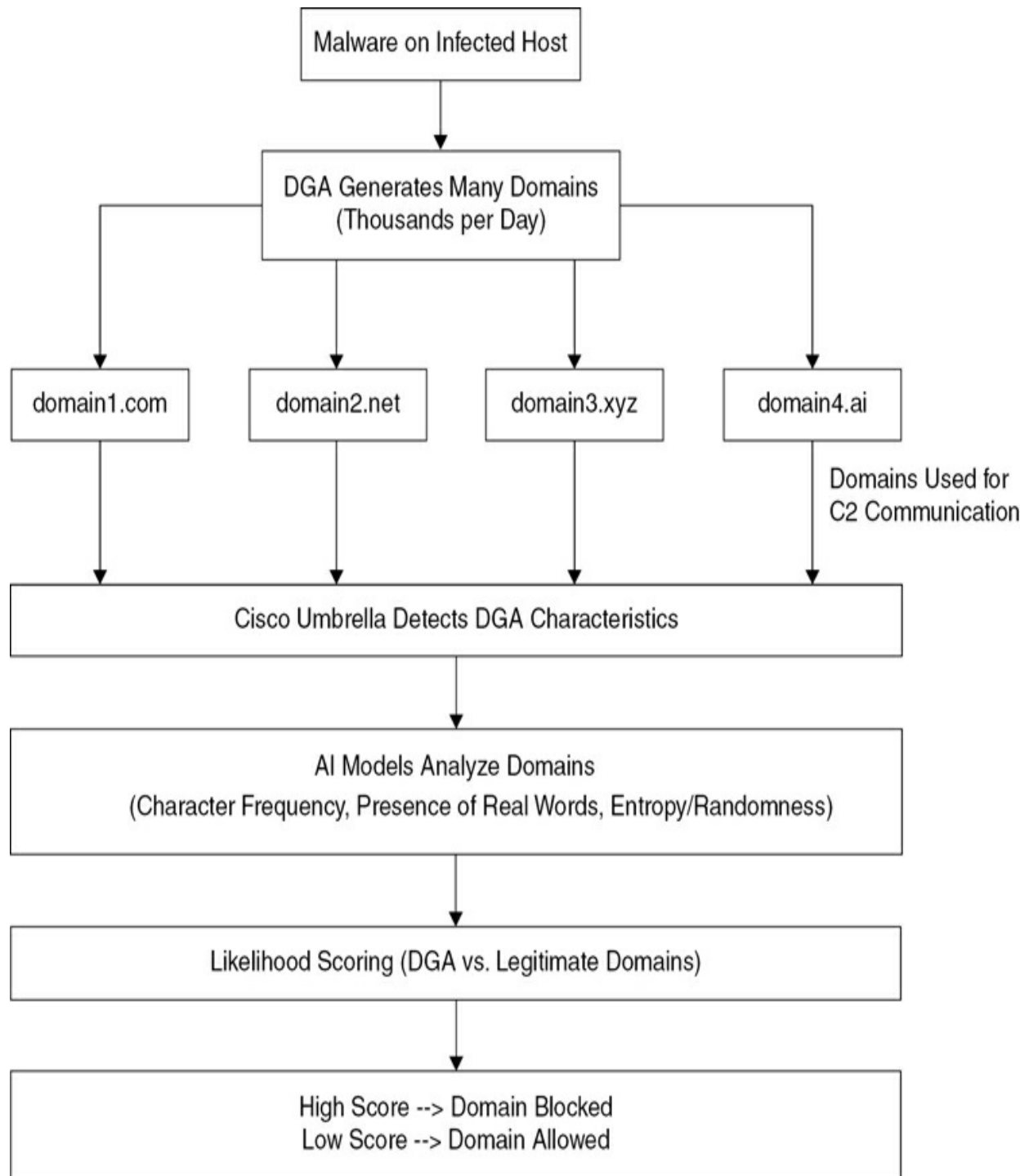


Figure 9-2 DGA Detection

Command-and-Control (C2) Callbacks

Once a device is compromised, malware will attempt to “call back” to a C2 server to receive instructions or exfiltrate data. These C2 communications are a critical indicator of an active infection. Umbrella is uniquely positioned to detect these callbacks because they almost always begin with a DNS request. The platform’s models analyze the request patterns from individual devices. If a device suddenly begins making requests to a number of known-malicious domains, newly seen domains with suspicious characteristics, or domains identified as part of a DGA family, it can be flagged as compromised. Because this detection happens at the DNS layer, Umbrella can block these C2 callbacks over any port or protocol, preventing the malware from activating, even if the initial infection vector was missed. This concept is illustrated in [Figure 9-3](#).

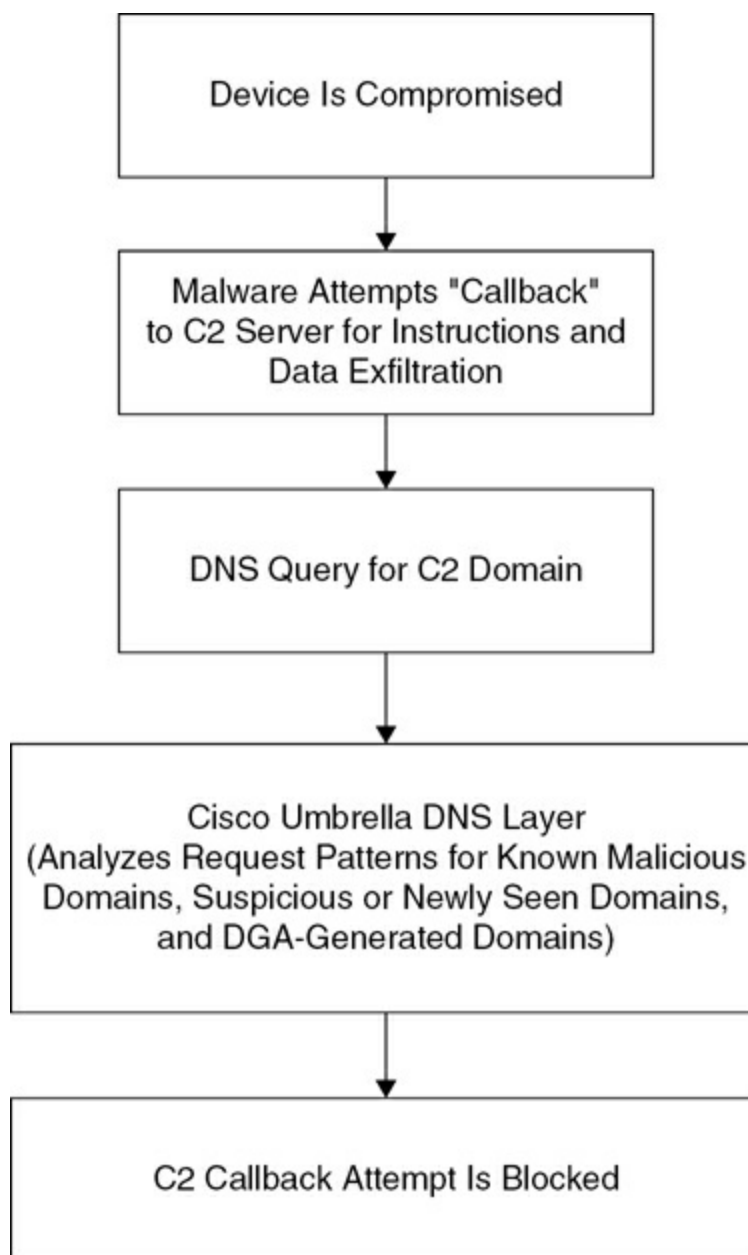


Figure 9-3 Umbrella Blocking C2 Callback Attempts

In [Figure 9-3](#), the flow illustrates how Cisco Umbrella detects and blocks C2 callbacks at the DNS layer. It starts with a device being compromised by malware. Once infected, the malware attempts to “call back” to a C2 server to receive further instructions or to exfiltrate data. This callback process begins with a DNS query for the C2 domain. Cisco Umbrella intercepts this query at the DNS layer and analyzes it using its threat intelligence models. These models look for indicators such as known malicious domains, suspicious or newly seen domains, and domains that match patterns generated by DGAs. If

the request is determined to be malicious or high risk, Umbrella blocks the connection attempt, effectively stopping the malware from communicating with its operator and preventing data theft or activation of additional payloads. This proactive methodology works across all ports and protocols, mitigating threats even if the initial compromise was missed.

Umbrella Investigate

Umbrella Investigate is an interactive console and API that gives security analysts direct access to the live and historical data that powers Umbrella's detections. It is a powerful tool for threat hunting and incident response, allowing analysts to move beyond simple alerts to a deep understanding of an attack's context.

With Investigate, a security analyst can take a single indicator, such as a suspicious domain, and instantly see a wealth of contextual information: a risk score indicating its likelihood of being malicious, a timeline of its DNS query volume, its passive DNS history (which IP addresses it has resolved to over time), its WHOIS registration data, and co-occurrences (other domains that are frequently requested by the same clients). This tool allows analysts to pivot through an attacker's infrastructure, uncovering related domains, IPs, and malware samples. By providing this rich, correlated context in a single interface, Investigate dramatically accelerates the incident investigation and response process, with many customers reporting a reduction in investigation time by 50 percent or more.

Case Study: Using Cisco Umbrella Investigate for Threat Hunting and Incident Response

A large enterprise in Raleigh, North Carolina, has more than 150,000 employees across multiple branch offices and remote workers. Its security operations center (SOC) noticed unusual activity flagged by its SIEM system. Several endpoint alerts suggested potential malware infections, but the alerts lacked enough context to confirm whether the threat was active or which domains were involved. The company's SOC team needed a way to quickly pivot from suspicious DNS queries to actionable

intelligence to contain the incident and prevent further compromise.

Traditional security tools provided indicators such as IP addresses and domains but did not give deeper insight into their relationships, hosting infrastructure, or reputation over time. The SOC analysts faced three key challenges: speed, context, and visibility. They needed to investigate and block malicious domains before more devices could be compromised. They needed to know whether the domains seen in logs were truly malicious, newly registered, or part of a larger campaign. They wanted to understand the full scope of potentially infected endpoints communicating with these domains.

The security team used Cisco Umbrella Investigate, a threat intelligence platform that provides real-time and historical visibility into domain and IP activity. This is how they approached the problem.

They extracted the suspicious domains from their SIEM and ran them through Cisco Umbrella Investigate. Investigate returned detailed information about each domain, including domain age and registration details (WHOIS), co-occurrence (other domains queried by the same devices), global and internal query volume trends, and security scores (likelihood of being malicious or DGA-generated).

The team identified that multiple suspicious domains resolved to the same IP range, indicating shared malicious hosting infrastructure. Investigate also revealed that these domains were part of a fast-flux network associated with a known malware family.

Using Investigate's pivoting capability, the SOC team discovered related domains that had not yet been queried in their environment but were highly likely to be used by the same attacker. The team added these domains and IPs to their blocklists in Cisco Umbrella to proactively prevent future communication attempts. They also isolated the infected devices and performed forensics to confirm and remove the malware.

By using Cisco Umbrella Investigate, the SOC team reduced

investigation time by 70 percent. Analysts were able to quickly distinguish between benign and malicious domains without waiting for third-party reports. They prevented the malware from successfully contacting its C2 servers. The related infrastructure data allowed them to preemptively block new domains before they were queried by other endpoints.

Cisco Umbrella Deployment Scenarios

Cisco Umbrella provides several deployment models, enabling organizations to adopt cloud security in a phased approach that aligns with their existing infrastructure and security maturity.

- **Network-Level Deployment:** This is the simplest and fastest way to protect an entire location. It involves redirecting the DNS queries from a network device (e.g., a router, firewall, or DHCP server) to Umbrella's global network IP addresses (e.g., 208.67.222.222 and 208.67.220.220). After this change is made, any device on that network that uses the local DHCP server for DNS resolution is automatically protected. This method is ideal for securing corporate offices, branch locations, and guest Wi-Fi networks with minimal configuration.
- **Virtual Appliances (VAs):** For organizations requiring more granular visibility and policy control within their internal network, Umbrella offers on-premises virtual appliances. These VAs are deployed as virtual machines on hypervisors like VMware or Hyper-V and function as conditional DNS forwarders. They intercept all DNS requests from the local network. Requests for internal domains are forwarded to the local Active Directory DNS servers, while requests for external domains are sent to the Umbrella cloud, appended with metadata that includes the internal IP address of the originating client. This process provides visibility into which specific device on the network made a request. Furthermore, by integrating with an Active Directory Connector, the VAs can map internal IPs to AD usernames, enabling user- and group-based policy enforcement. Best practices dictate that VAs should always be deployed in redundant pairs on separate physical hosts to ensure high availability.

- **Endpoint Client Deployment:** To protect users when they are outside the corporate network, Umbrella relies on an endpoint agent. This functionality is now integrated into the Cisco Secure Client, a unified agent that consolidates multiple security services. The client, available for Windows, macOS, iOS, Android, and ChromeOS, ensures that all DNS requests from the device are sent directly to Umbrella, regardless of what network the user is connected to. This approach provides “always-on” protection for roaming and remote workers, enforcing the same security and content policies whether the user is in the office, at home, or connected to public Wi-Fi. [Figure 9-4](#) shows the Cisco Secure Client and the Umbrella module active. [Figure 9-5](#) shows the Cisco Secure Client Umbrella statistics window.

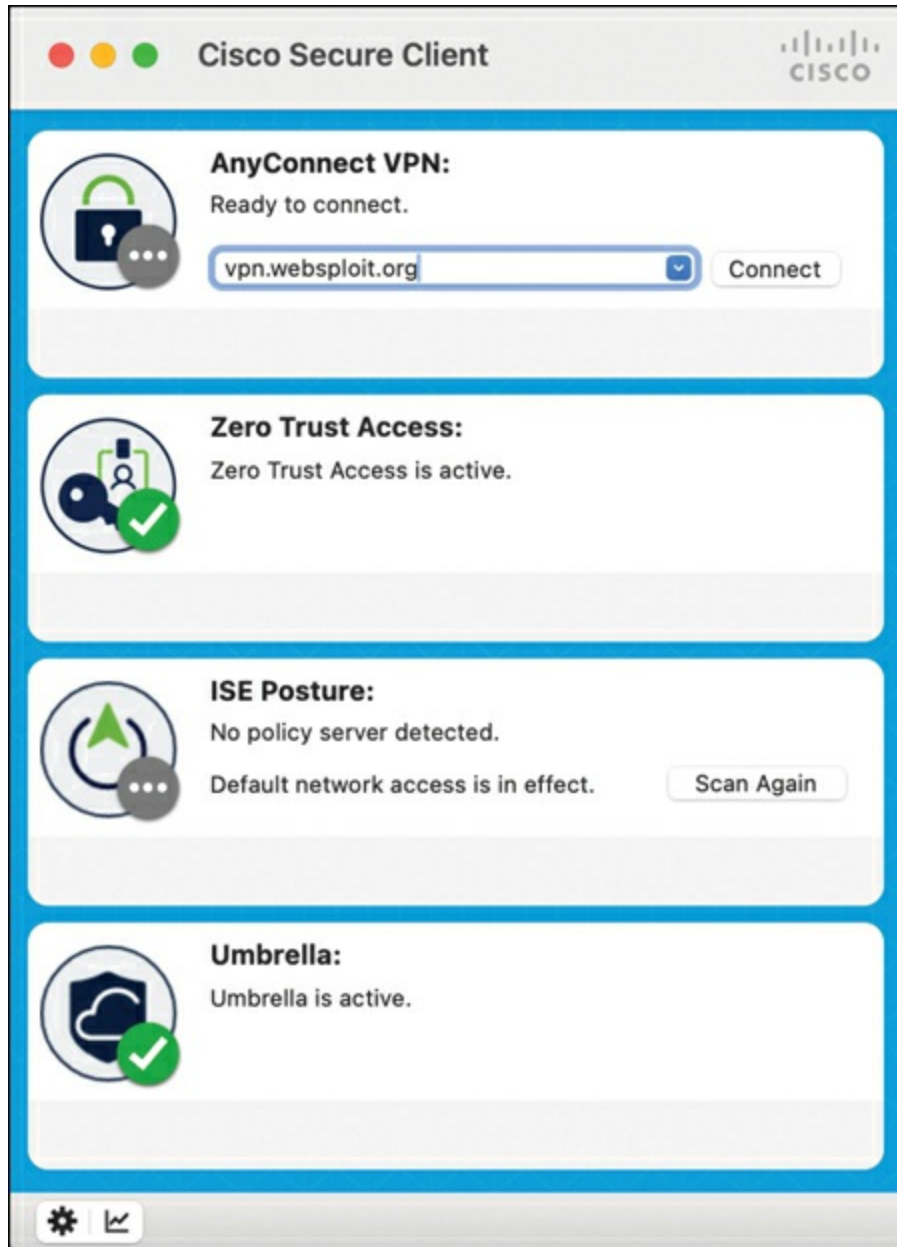


Figure 9-4 Cisco Secure Client

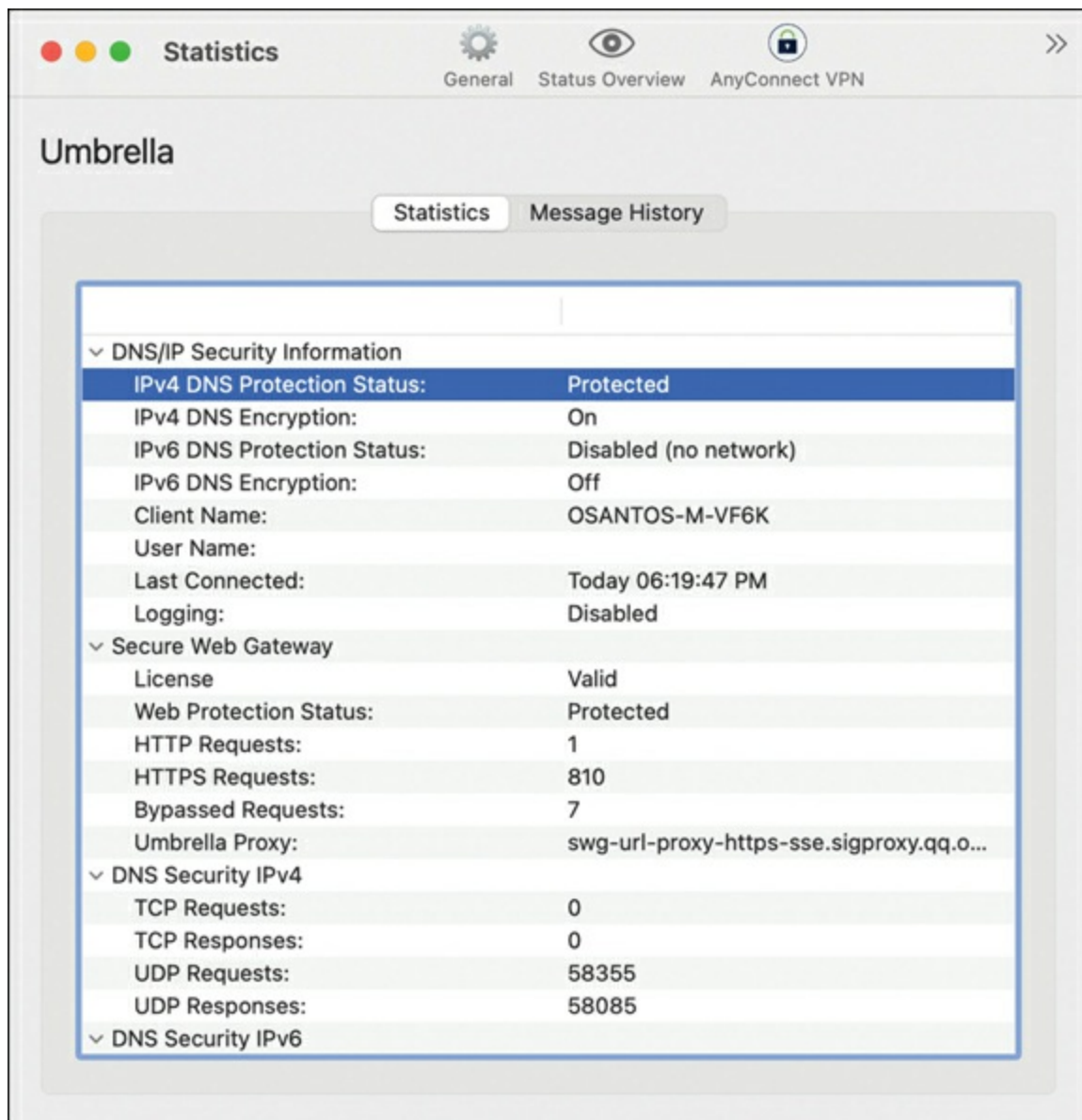


Figure 9-5 Cisco Secure Client Umbrella Statistics

Policy Configuration Best Practices

Umbrella's policy engine allows for the creation of granular rules based on identity (network, user, device), destination (security category, content category, specific domain), and location.

- **Remote Workers:** A common and effective strategy for securing remote workers is to create layered, location-aware policies.

- **Default Roaming Policy:** A baseline policy should be created for all roaming computers. This policy typically enforces all security categories (malware, phishing, C2 callbacks, and so on) but may have more lenient content filtering rules. This approach maintains the user's privacy while ensuring they are always protected from threats, acknowledging that work devices are often used for some personal browsing when off-network.
- **On-Network Policy:** A more restrictive policy is then created for the corporate network identities. This policy is placed at a higher priority in the policy list. It would include stricter content filtering rules aligned with corporate acceptable use policies (e.g., blocking social media, unapproved AI tools, streaming video).
- **Policy Precedence:** When a user with the roaming client is on the corporate network, their device is identified as being part of that network, and the higher-priority on-network policy is applied. When the user leaves the office, the default roaming policy takes effect. This approach provides a seamless transition between security postures without any user intervention.
- **Guest Wi-Fi Networks:** Securing guest Wi-Fi is crucial for protecting both the guests from threats and the organization's brand reputation. A dedicated policy should be created for the guest network identity (e.g., the guest VLAN's public IP address).
- **Security Settings:** All security categories should be enabled to prevent guests from accessing malicious sites and to stop any compromised guest devices from communicating with C2 servers over the network.
- **Content Filtering:** A restrictive content filtering policy is typically applied, blocking categories such as pornography, gambling, hate speech, and other content deemed inappropriate. Custom allow/block lists can be used to fine-tune access. The goal is to provide a safe and brand-appropriate browsing experience for all guests.
- **Simplified Deployment via Meraki:** For organizations using Cisco Meraki wireless access points, this deployment is exceptionally simple. The Meraki dashboard can be linked to Umbrella via an API key, and policies can be applied directly to specific SSIDs, such as the Guest-

WiFi SSID, from within the Meraki interface.

You can leverage these flexible deployment models and granular policy controls to build a robust and adaptable security posture that effectively protects all users and locations in the modern, distributed enterprise.

Cisco AI Defense: Securing the AI Revolution

Although Cisco Umbrella and the SASE architecture address the security challenges of accessing the cloud-centric world, the rapid proliferation of generative AI has created an entirely new and distinct security frontier. The use of AI introduces a dual-risk scenario: the threats posed by employees using external AI tools and the vulnerabilities inherent in the AI applications developed in-house. Cisco introduced Cisco AI Defense as a comprehensive platform designed to secure the entire AI lifecycle.

The Rise of Shadow AI

The mainstream adoption of powerful generative AI tools has been a double-edged sword for enterprises. While offering immense potential for productivity and innovation, it has also opened a Pandora's box of security and data governance challenges that traditional security frameworks are not equipped to handle.

The term *Shadow AI* began to describe the widespread, unsanctioned use of third-party generative AI tools and applications by employees. Driven by a desire to improve efficiency, employees are increasingly turning to public large language models (LLMs) to summarize documents, write code, draft emails, and perform a variety of other tasks. While often well-intentioned, this behavior introduces significant risks.

The most immediate danger is the inadvertent leakage of sensitive data. When an employee pastes proprietary source code, confidential financial data, customer personally identifiable information (PII), or strategic planning documents into a public AI tool, that data is transmitted to a third-party provider, often with little to no organizational oversight or control. This situation can lead to serious data breaches, intellectual property loss, and regulatory compliance violations. Furthermore, the outputs generated by

these AI tools can also pose a threat, potentially containing malware, misinformation, or biased content that is then brought back inside the organization. A blunt approach of blocking all AI tools is often untenable because it hinders the very innovation and productivity that businesses seek to adopt.

The Developer's Dilemma: Securing In-House AI Applications

Beyond the use of external tools, organizations are increasingly developing their own AI-powered applications. This tactic creates an entirely new and complex attack surface that exists within the AI models themselves. These models are not like traditional software; they are susceptible to a new class of attacks that exploit their probabilistic nature and training data.

These AI-specific threats include

- **Prompt Injection and Jailbreaking:** Attackers can craft malicious inputs (prompts) that trick an AI model into bypassing its safety guardrails, causing it to generate harmful, biased, or inappropriate content, or to execute unintended actions.
- **Data Poisoning:** Adversaries can intentionally introduce malicious data into the training set of an AI model, creating hidden backdoors or biases that can be exploited later.
- **Model Evasion:** Attackers can subtly manipulate inputs to cause a model to misclassify data, for example, tricking a malware detection model into classifying a malicious file as benign.
- **Data Exfiltration:** A model can be prompted in such a way that it reveals sensitive information that was part of its training data, leading to data leakage.

These vulnerabilities exist at every stage of the AI lifecycle, from data collection and training to fine-tuning and deployment. Securing this lifecycle requires a new set of specialized tools for vulnerability testing and real-time protection. Cisco AI Defense protects against the most common risks outlined in the OWASP Top 10 for LLM Applications.

OWASP GenAI Security Project

The OWASP GenAI Security Project (available at <https://genai.owasp.org>) is a community-led initiative dedicated to helping developers, security professionals, and organizations understand and mitigate the risks associated with generative AI. The project provides open-source, practical guidance and tools.

The OWASP Top 10 for LLM Applications is the flagship effort of the project and is designed to be the essential starting point for anyone building or defending systems that use LLMs. Its primary goal is to raise awareness of the most critical security risks.

Cisco AI Defense Four-Pillar Framework for AI Security

Cisco AI Defense is engineered as a single end-to-end solution that provides visibility and protection across the entire AI landscape. Its architecture is built on the pioneering technology acquired from Robust Intelligence, a company at the forefront of the AI security space, and is enriched with threat intelligence from Cisco Talos and detection models powered by Scale AI.

The platform's core architectural principle is to embed AI-specific security controls directly into the network fabric. This means that, rather than requiring specialized agents on endpoints or custom instrumentation for individual AI models, Cisco AI Defense leverages the same network infrastructure that already mediates data flows between users, applications, and cloud services. The enforcement and inspection capabilities are integrated into key control points such as Cisco firewalls, secure web gateways, cloud-native proxies, and existing policy enforcement nodes. By doing so, AI Defense operates at the natural chokepoints where AI inference requests, data exchanges, and model interactions occur, enabling governance and real-time detection without adding operational overhead.

By leveraging the existing visibility and enforcement points within the Cisco Security Cloud, AI Defense can provide a consistent and reliable layer of protection that is agnostic to the specific AI models being used or the cloud environments they are hosted in. This network-centric approach avoids the

complexity of agent-based or model-specific solutions and leverages Cisco's core strengths in networking.

The solution is structured around four distinct but interconnected pillars, each designed to address a specific aspect of AI risk.

Figure 9-6 shows the four major pillars of AI Defense.

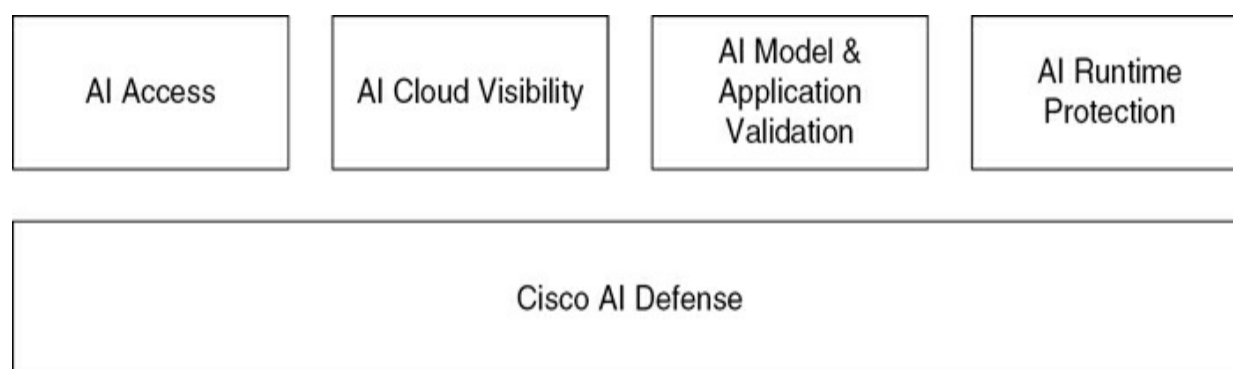


Figure 9-6 The Four Major Pillars of AI Defense

The following are the details of each pillar shown in Figure 9-6 (see also Table 9-2).

- **AI Access:** This pillar is designed to tackle the problem of Shadow AI. It provides security teams with comprehensive visibility into the use of third-party AI applications across the organization. The system can automatically discover and assess the risk of more than 750 generative AI applications, providing context on their usage patterns and potential security implications. Armed with these insights, administrators can enforce granular, context-aware access policies. Rather than a simple block or allow, they can create nuanced rules, such as permitting the use of a sanctioned tool like DeepSeek, Qwen, and others while applying DLP guardrails to prevent employees from submitting sensitive or proprietary information in their prompts.
- **AI Cloud Visibility:** As organizations move from being multicloud to being multicloud and multimodel, understanding their AI footprint becomes a critical first step. This pillar focuses on discovering, inventorying, and assessing an organization's own AI assets. It can identify AI workloads, models, agents, and connected data sources across distributed cloud environments, including native integrations with

platforms like AWS Bedrock, Google Vertex AI, Microsoft’s Azure AI Foundry, and others. This pillar provides a single pane of glass for security and governance teams to understand their AI posture, gauge risk, and identify unmanaged or unprotected AI systems.

- **AI Model and Application Validation:** This pillar functions as an automated “algorithmic red teaming” service for an organization’s in-house AI applications. Before an AI model is deployed into production, the validation engine rigorously tests it against hundreds of known safety and security vulnerabilities. It probes the model for weaknesses such as susceptibility to prompt injection, algorithmic bias, generation of toxic content, and other potential failure modes. The system then generates detailed reports on any vulnerabilities found and recommends specific guardrails that can be implemented to mitigate them, ensuring that models are hardened against threats before they are exposed to users.
- **AI Runtime Protection:** This is the enforcement arm of the platform, acting as a real-time “AI firewall” for applications in production. Based on the vulnerabilities identified during the validation phase, this pillar deploys tailored security guardrails that inspect the traffic flowing to and from the AI model in real time. It can detect and block active threats as they occur, including malicious prompt injection attacks, attempts to cause denial of service, and prompts designed to exfiltrate sensitive data from the model’s knowledge base. This pillar provides a good layer of defense that protects live applications from the rapidly evolving landscape of adversarial AI attacks.

Table 9-2 Cisco AI Defense Components and Their Functions

Pillar	Target	Function	Threats Mitigated
AI Access	Employee use of third-party AI tools	Discovers, assesses risk, and applies access/DLP policies to hundreds of GenAI apps	Shadow AI, sensitive data leakage to public models, use of unsanctioned tools
AI Cloud Visibility	In-house AI infrastructure across multicloud environments	Creates a comprehensive inventory of AI assets (models, data sources, workloads)	Unmanaged AI risk, lack of visibility into AI footprint, compliance gaps
AI Model and Application Validation	Preproduction AI models and applications	Performs automated, algorithmic red teaming to test for hundreds of vulnerabilities	Model vulnerabilities, bias, toxicity, susceptibility to jailbreaking and prompt injection
AI Runtime Protection	Live, production AI applications	Applies real-time security guardrails to monitor and block malicious traffic	Active prompt injection attacks, denial of service (DoS), and sensitive data exfiltration

Summary

The dual disruptions of cloud adoption and the generative AI revolution have forever altered the enterprise security landscape. The traditional perimeter has vanished, replaced by a distributed ecosystem of users, devices, and applications. Simultaneously, AI has emerged as both a powerful engine of

innovation and a new attack surface. Navigating this complex new reality requires a security strategy that is equally adaptive, intelligent, and integrated.

This chapter has provided an overview of two of Cisco's important security platforms. Cisco Umbrella (born from the work of OpenDNS) is integrated into the core of Cisco's SASE architecture. Through its cloud-delivered suite of services (from foundational DNS-layer security to advanced SWG, CDFW, and CASB capabilities), it provides the essential first line of defense, applying consistent, intelligent protection for any user, on any device, anywhere. Cisco has a massive global data set, which fuels a predictive intelligence engine that uses machine learning and the expertise of Cisco Talos to block threats before they can compromise your systems.

You also learned that Cisco AI Defense is a solution designed to secure the new AI-powered applications and workflows within the enterprise. It addresses the full lifecycle of AI risk, providing the visibility to manage Shadow AI, the tools to validate the security of in-house models, and the real-time guardrails to protect production applications from novel, AI-specific attacks.

References

- Cisco acquisition of OpenDNS. Cisco Newsroom:
<https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2015/m08/cisco-completes-acquisition-of-opendns.html>
- Cisco AI Defense:
<https://www.cisco.com/site/us/en/products/security/ai-defense/index.html>
- Cisco Umbrella enterprise security packages:
<https://umbrella.cisco.com/products/umbrella-enterprise-security-packages>
- Comprehensive cloud security services for business:
<https://umbrella.cisco.com/products/cloud-security-service>
- Cisco Umbrella global cloud architecture:
<https://umbrella.cisco.com/cisco-umbrella-global-cloud-architecture>

Chapter 10. Security: Cisco XDR, Splunk, and Cisco Vulnerability Management

In previous chapters, you learned that the landscape of cybersecurity has undergone a fundamental transformation over the past decade. Security operations have evolved from a reactive posture, dependent on a collection of disparate and siloed tools (e.g., standalone antimalware, endpoint detection and response solutions, firewalls, and intrusion detection systems), to a proactive and integrated discipline.

Modern enterprises have been operating in hybrid cloud environments for several years and with an ever-expanding digital footprint. They face a lot of pressure from increasingly sophisticated adversaries using AI and highly complex attacks. This complexity has introduced significant challenges for the modern security operations center (SOC). The most notable is the overwhelming volume of alerts generated by disconnected security products. This issue is also known as *alert fatigue*. This fatigue makes it nearly impossible for security teams to manually correlate threats between systems, leading to manual, ad hoc investigation processes and, critically, an increased risk of missing genuine threats among the noise.

The industry's initial response was to centralize data, leading to the rise of the security information and event management (SIEM) platform as the foundation of the SOC. This is illustrated in [Figure 10-1](#).

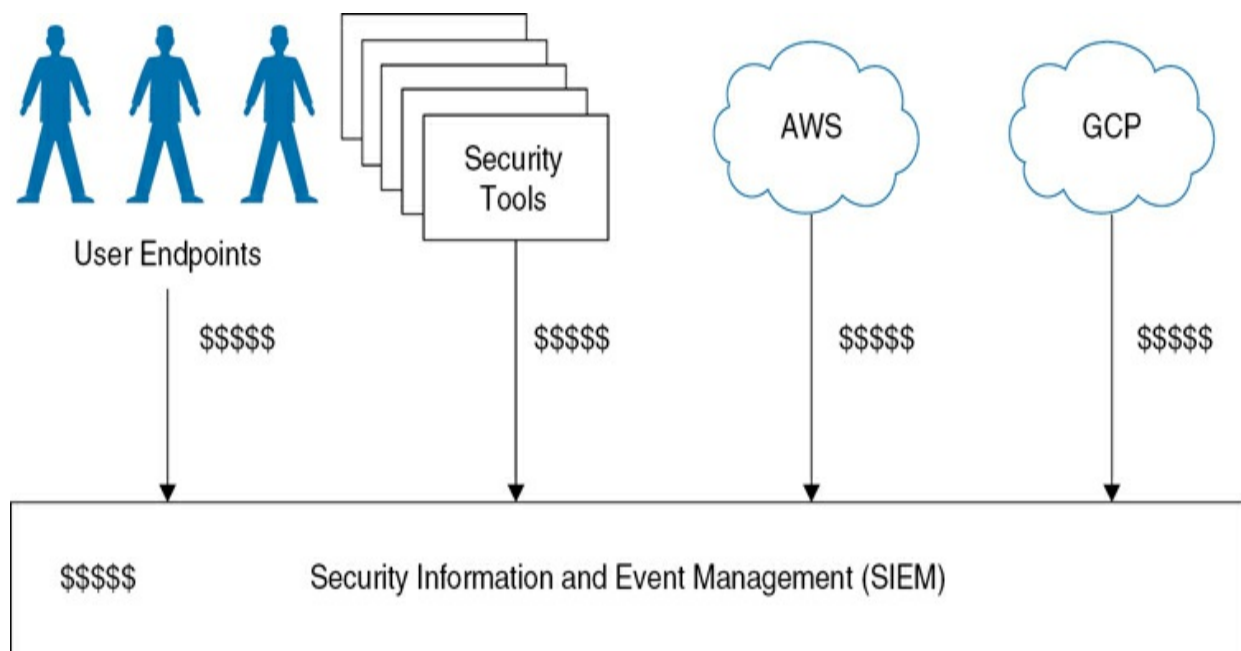


Figure 10-1 A Traditional SIEM Processing Data from Different Sources

The huge volume and velocity of data in contemporary environments have exposed the limitations of monolithic systems. Sending all telemetry (from every endpoint, network device, and cloud service) to a central SIEM for analysis can be prohibitively expensive (shown with the \$ signs in [Figure 10-1](#)) and can slow down query performance, hindering the rapid response required to combat modern attacks.

This challenge has required a more intelligent, federated architectural approach. Instead of a single data lake, a modern security architecture requires specialized engines optimized for distinct tasks (such as real-time detection and response, deep forensic analysis and threat hunting, and proactive risk management). This architectural choice is a strategic response to the economic and operational limitations of previous security models, acknowledging that a single platform cannot efficiently serve these diverse needs.

This chapter covers three core technology pillars that form the foundation of this modern, federated security strategy: Cisco Extended Detection and Response (XDR), the Splunk ecosystem of tools, and Cisco Vulnerability Management. Each represents a distinct and critical function within a mature security program, and their integration creates a powerful, synergistic defense ecosystem.

XDR represents the evolution of endpoint detection and response (EDR), extending its principles across multiple security domains. It functions as a real-time, high-fidelity threat detection and response engine. Its primary purpose is to ingest and correlate telemetry from a curated set of critical sources (such as endpoints, networks, email, and identity systems) to automatically connect disparate, low-confidence signals into a single, high-confidence incident. This process dramatically accelerates detection and enables automated containment actions, serving as the SOC's rapid response capability.

The SIEM remains the comprehensive data analytics and long-term storage platform of the SOC. Its role is to ingest and normalize vast quantities of machine data from virtually any source across the enterprise for deep investigation, compliance reporting, and proactive threat hunting. Unlike the curated, real-time focus of XDR, the SIEM provides the “big data” backend, empowering analysts with a powerful query language and extensive historical context to uncover complex, low-and-slow attacks and fulfill stringent regulatory requirements.

Risk-based vulnerability management (RBVM) is a proactive, intelligence-driven discipline that shifts vulnerability management from a reactive, volume-based patching exercise to a strategic, risk-focused program. By leveraging data science, machine learning (ML), and real-world threat intelligence, RBVM platforms identify and prioritize the small subset of vulnerabilities that pose a genuine, measurable threat of exploitation. In this way, organizations can focus their limited remediation resources on the weaknesses that matter most, effectively reducing the attack surface before an incident can occur.

The strategic integration of Cisco XDR, the Splunk platform, and Cisco Vulnerability Management is working toward a unified, AI-driven security operations platform. This combination is engineered to deliver unprecedented visibility, context, and automation across the entire threat detection, investigation, and response (TDIR) lifecycle. In this chapter, we will provide an exhaustive analysis of this security triumvirate. We will begin with a deep dive into the architecture and capabilities of each individual component: Cisco XDR as the engine for accelerated response, the Splunk ecosystem as the universe of data-driven insights, and Cisco Vulnerability Management as

the data science–powered risk prioritizer. Subsequently, we will explore the powerful, synergistic integrations between these platforms, demonstrating through practical workflows how they combine to create a security ecosystem that is far greater than the sum of its parts. The final analysis will synthesize these findings, offering a forward-looking perspective on the future of the integrated, AI-driven SOC.

Unifying Telemetry for Accelerated Response Using Cisco XDR

Cisco XDR is engineered as a direct response to the complexity and fragmentation that plague modern security operations. It is not designed to be another data lake but rather a robust security platform built for speed and efficacy. Its value proposition lies in its curated, high-speed correlation and response capabilities, operating within a well-defined ecosystem of both Cisco and third-party security tools. This is very different in nature of a traditional SIEM and enables Cisco XDR to solve for the most common and high-impact threats with extreme efficiency.

Cisco XDR Architectural Deep Dive

Cisco XDR is a cloud-native platform that functions as a central hub for security analytics and response, creating an integrated ecosystem from what are often isolated security tools. Its architecture is founded on three key components: data ingestion and normalization, a central analytics engine, and a tiered licensing model that scales its capabilities.

Data Ingestion and Normalizations

The Cisco XDR platform’s effectiveness begins with its ability to ingest high-value security telemetry from a broad range of sources. It natively integrates with the Cisco Secure portfolio, pulling data from key control points including endpoints (Cisco Secure Endpoint), networks (Cisco Secure Firewall, Meraki devices), email (Email Threat Defense), identity and access management (Duo), and DNS security (Umbrella), as illustrated in [Figure 10-2](#).

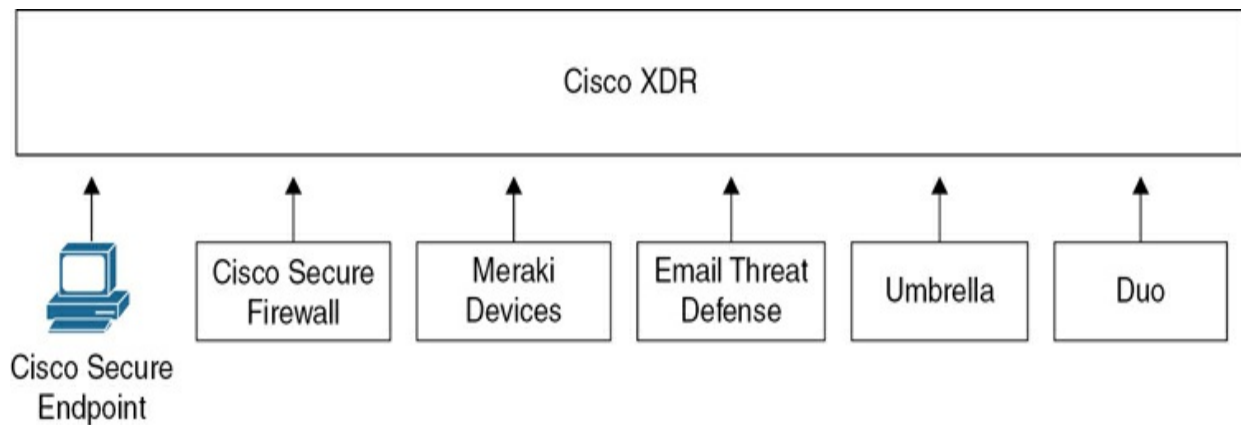


Figure 10-2 Cisco XDR Ingestion of High-Value Security Telemetry

Only a few security environments are homogeneous. This is why Cisco XDR also supports integrations with select third-party (non-Cisco) security tools. The top six data sources that organizations consider essential for an XDR solution are endpoint security, network, firewall, identity, email, and DNS.

Once ingested, this telemetry is normalized into a common, structured format. This step is very important, because it allows the analytics engine to correlate events and behaviors across different domains—for example, linking a suspicious login event from an identity provider with an anomalous process execution on an endpoint and a command-and-control (C2) beacon detected on the network. This is illustrated in [Figure 10-3](#).

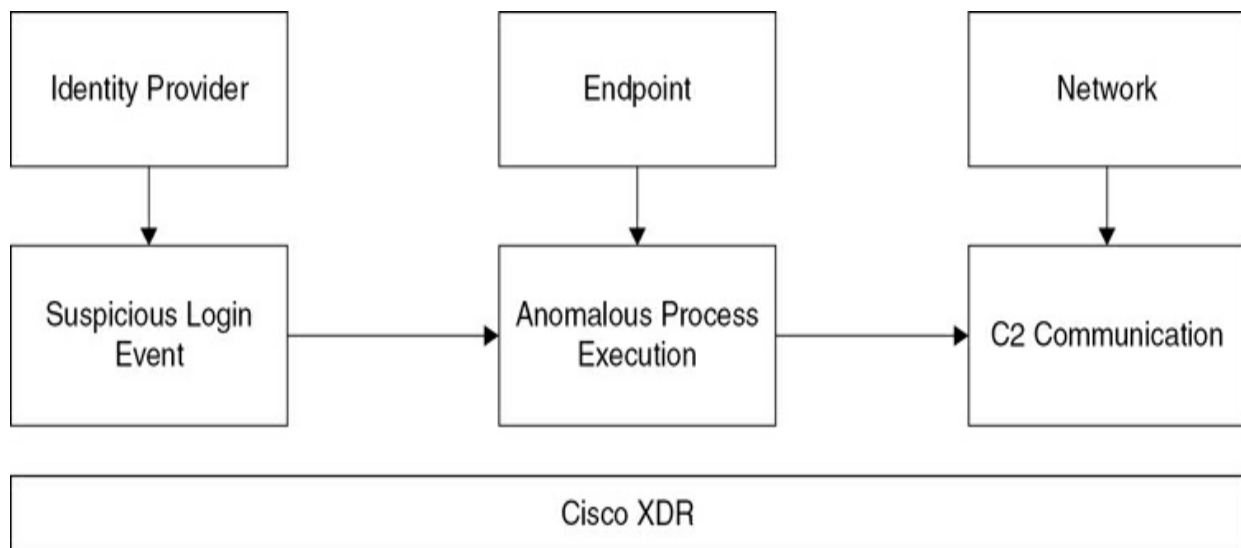


Figure 10-3 Cisco XDR Event Correlation

The Analytics and Correlation Engine

As shown in a high-level example in [Figure 10-2](#), the heart of Cisco XDR is its analytics and correlation engine. The engine processes the normalized data to produce prioritized, high-fidelity detections. The engine is designed to solve one of the biggest problems in security operations: the inability to connect weak signals from multiple security products into a strong indicator of malicious intent. For example, a single failed login is a weak signal, as is a single PowerShell script execution. However, when these events occur in sequence on the same endpoint, initiated by a user connecting from an unusual location, the combination becomes a strong signal of a potential breach. The Cisco XDR engine automates this cross-domain correlation, chaining events together to build a comprehensive picture of an attack.

Licensing Tiers (Essentials, Advantage, Premier)

Cisco offers XDR in three distinct licensing tiers to meet varying organizational needs and maturity levels:

- **Essentials:** This tier provides the core security analytics and correlation capabilities, focusing primarily on telemetry from the native Cisco Secure product suite.
- **Advantage:** This tier builds on Essentials by adding Cisco-curated integrations with select third-party security tools, allowing organizations to extend XDR's visibility and correlation into their existing multivendor environments.
- **Premier:** This tier delivers the full capabilities of the Advantage license as a Managed Service provided by Cisco security experts. It includes additional services such as security validation through penetration testing and select incident response services from Cisco Talos.

Cisco XDR Core Capabilities and AI-Powered Detection

The “how” of Cisco XDR's threat detection is rooted in a multilayered approach that combines advanced machine learning, continuous threat

intelligence, and a deep understanding of cross-product context to move beyond simple signature-based alerts.

Cisco XDR uses advanced machine learning (ML) and user and entity behavior analytics (UEBA) to establish a baseline of normal activity for every user and device in the environment. This continuous monitoring allows the platform to learn what constitutes “normal” behavior (e.g., typical login times, common applications used, and regular network traffic patterns). Once this baseline is established, the AI/ML engines can identify subtle anomalies and sophisticated attack patterns that would evade traditional security tools.

Note

The use of AI is critical for detecting modern threats like insider threats, compromised credentials, and file-less malware, while simultaneously minimizing the false positives that contribute to alert fatigue.

Cross-Product Context, Incident Prioritization, and Cisco Talos Threat Intelligence Integration

A great capability of Cisco XDR is its functionality for enriching alerts with cross-product context. When an alert is generated, the system automatically gathers a detailed view of the surrounding environment. It assesses the posture of the involved assets, relevant policy settings, recent user behavior, and the overall IT posture. This rich contextual data allows the platform to automatically score and prioritize security events based on tangible risk factors. Cisco XDR (using AI) can surface the most critical threats that warrant immediate attention, instead of presenting analysts with a flat list of thousands of alerts. This approach dramatically improves SOC efficiency.

The platform’s detection capabilities are continuously improved by real-time threat intelligence updates from Cisco Talos, Cisco’s very reputable threat intelligence organization. This integration ensures that Cisco XDR is always aware of the latest malware signatures, malicious domains, exploits, and adversary tactics, techniques, and procedures (TTPs). Alerts generated by the platform are automatically mapped to the MITRE ATT&CK framework, providing analysts with standardized context around the attacker’s objectives

and methods, which aids in both investigation and strategic defense planning. MITRE ATT&CK is a publicly available knowledge base that catalogs the TTPs used by real-world cyber adversaries. It's essentially a structured framework that describes how attackers operate, step by step, across the entire lifecycle of an intrusion.

Automated Guided Response and Containment Actions

Detecting a threat quickly is only half the battle; responding effectively is equally critical. Cisco XDR is built to drastically reduce the time between detection and response (mean time to respond, or MTTR) through a combination of full automation and guided, analyst-driven actions.

For high-confidence threats, Cisco XDR can execute preconfigured response actions automatically, without requiring human intervention. This capability is very important for containing fast-moving attacks like ransomware before they can spread laterally across the network. These automated actions are tied directly to the control points within the integrated ecosystem and can include

- Isolating a compromised endpoint using Cisco Secure Endpoint to prevent lateral movement.
- Blocking malicious IP addresses, domains, and URLs at the DNS layer using Cisco Umbrella.
- Disabling a compromised user account or enforcing multifactor authentication (MFA) challenges via Cisco Duo.
- Triggering custom remediation scripts to eliminate malware and restore system integrity.

Guided Playbooks, Workflows, and Forensics

For incidents that require human analysis and decision-making, Cisco XDR provides orchestration capabilities through a library of prebuilt and customizable playbooks. These workflows guide analysts through the necessary investigation and response steps, ensuring a consistent and effective process. Using a drag-and-drop interface, security teams can build

their own playbooks, assigning tasks to different groups and automating sequences of actions across multiple integrated products. This functionality streamlines complex response efforts and reduces the potential for human error during high-pressure situations.

After a threat has been contained, understanding its full scope and root cause is essential for improving long-term security posture. Cisco XDR provides powerful forensic tools to support post-incident investigation and analysis. The platform offers a visual representation of the entire attack chain, often referred to as a “visual forensic” summary. This allows analysts to trace the complete flow of an incident, from the initial point of entry to the final impact. They can see precisely how an attacker gained access, what lateral movements the attacker conducted across the network, which systems were compromised, and what data may have been exfiltrated. This clear, graphical depiction of an attack is invaluable for understanding complex incidents and communicating their scope to stakeholders.

Understanding Attacker TTPs

Beyond just showing the sequence of events, Cisco XDR provides a summary of the specific TTPs used by the attacker. It is very important that the security team understands the adversary’s methods. For example, did they use a phishing email for initial access, PowerShell for execution, and Remote Desktop Protocol (RDP) for lateral movement? With this knowledge, you can identify and remediate the underlying security gaps that allowed the attack to succeed. This historical attack analysis helps organizations continuously improve their security posture over time.

Data-Driven Insights with the Splunk Ecosystem

You learned that Cisco XDR is a great platform designed for speed and precision within a curated ecosystem. The Splunk platform represents a similar philosophy: an agnostic, all-inclusive data engine capable of ingesting, analyzing, and visualizing machine data from any source. Splunk’s security strategy is built on a “platform-first” model, where the value of its individual security applications is exponentially amplified by the power and

flexibility of this core data platform. This creates an integrated ecosystem. For many organizations, this investment turns Splunk into the central nervous system for their operational and security data, making it an indispensable part of their technology stack.

At the core of the Splunk ecosystem are its foundational data platforms, which serve as the engine for all of its security, observability, and IT operations solutions.

Splunk Enterprise and Splunk Cloud Platform

Splunk Enterprise and Splunk Cloud are the two deployment models for Splunk's core technology, as highlighted in [Figure 10-4](#).

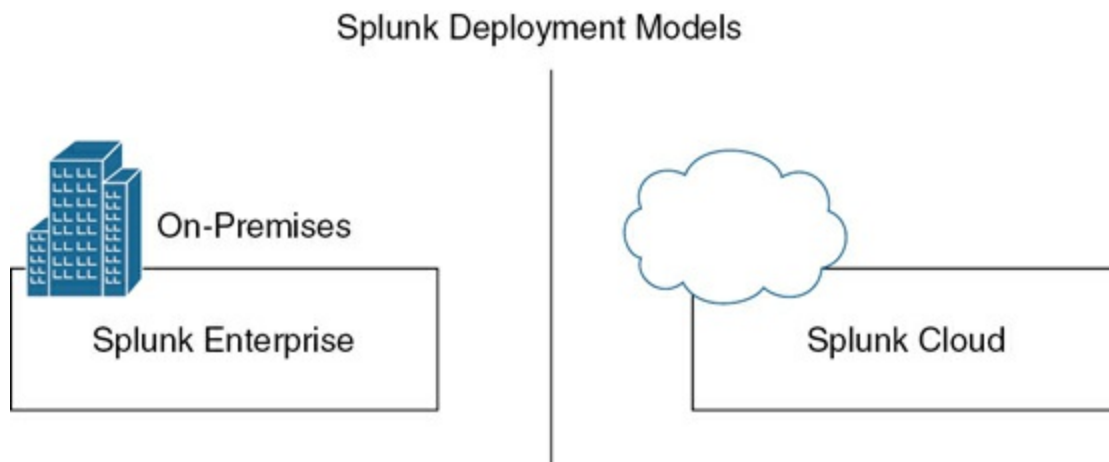


Figure 10-4 Splunk Enterprise vs. Splunk Cloud

As shown in [Figure 10-4](#), Splunk Enterprise is the self-hosted, on-premises version, whereas Splunk Cloud Platform is the SaaS offering. Both are designed to perform the same fundamental task: to capture, index, and correlate massive volumes of real-time and historical machine-generated data in a searchable repository. This data can come from virtually any source (servers, network devices, applications, IoT sensors, cloud services, and more). The platform's power lies in its ability to turn this unstructured or semi-structured data into actionable insights through its proprietary Search Processing Language (SPL). For security teams, this capability makes Splunk the definitive "single source of truth" for long-term log retention, deep forensic investigation, compliance auditing, and unrestricted threat hunting.

Splunk for Security Operations and Splunk Enterprise Security

Built upon this powerful data foundation is a comprehensive suite of security products designed to address the entire threat detection, investigation, and response (TDIR) lifecycle. These are not standalone tools but rather applications that leverage the full power of the underlying platform.

Splunk Enterprise Security (ES) is the market-leading SIEM solution. It operates as an application layer on top of the core Splunk platform, providing a security-specific framework for monitoring, detection, and compliance. ES ingests and normalizes data from a vast array of security technologies, including network devices, endpoints, identity systems, malware sandboxes, and vulnerability scanners, to provide a unified view of an organization's security posture.

ES is equipped with a rich set of features designed for the modern SOC. These features include hundreds of prebuilt correlation rules, customizable dashboards for real-time monitoring, and seamless integration with third-party threat intelligence feeds. A powerful capability of Splunk is the risk-based alerting (RBA) feature. Instead of generating a separate alert for every suspicious event, RBA attributes risk to users and systems as events are observed. Once an entity's cumulative risk score crosses a certain threshold, a single, high-context alert is generated. This methodology can reduce alert volumes by 80 to 90 percent, allowing analysts to focus on the highest-risk threats rather than chasing individual, low-fidelity events. ES also provides an integrated investigation workbench and threat topology mapping to visualize the scope of an incident.

[Figure 10-5](#) shows an example of the many dashboards that you can create and customize in Splunk.

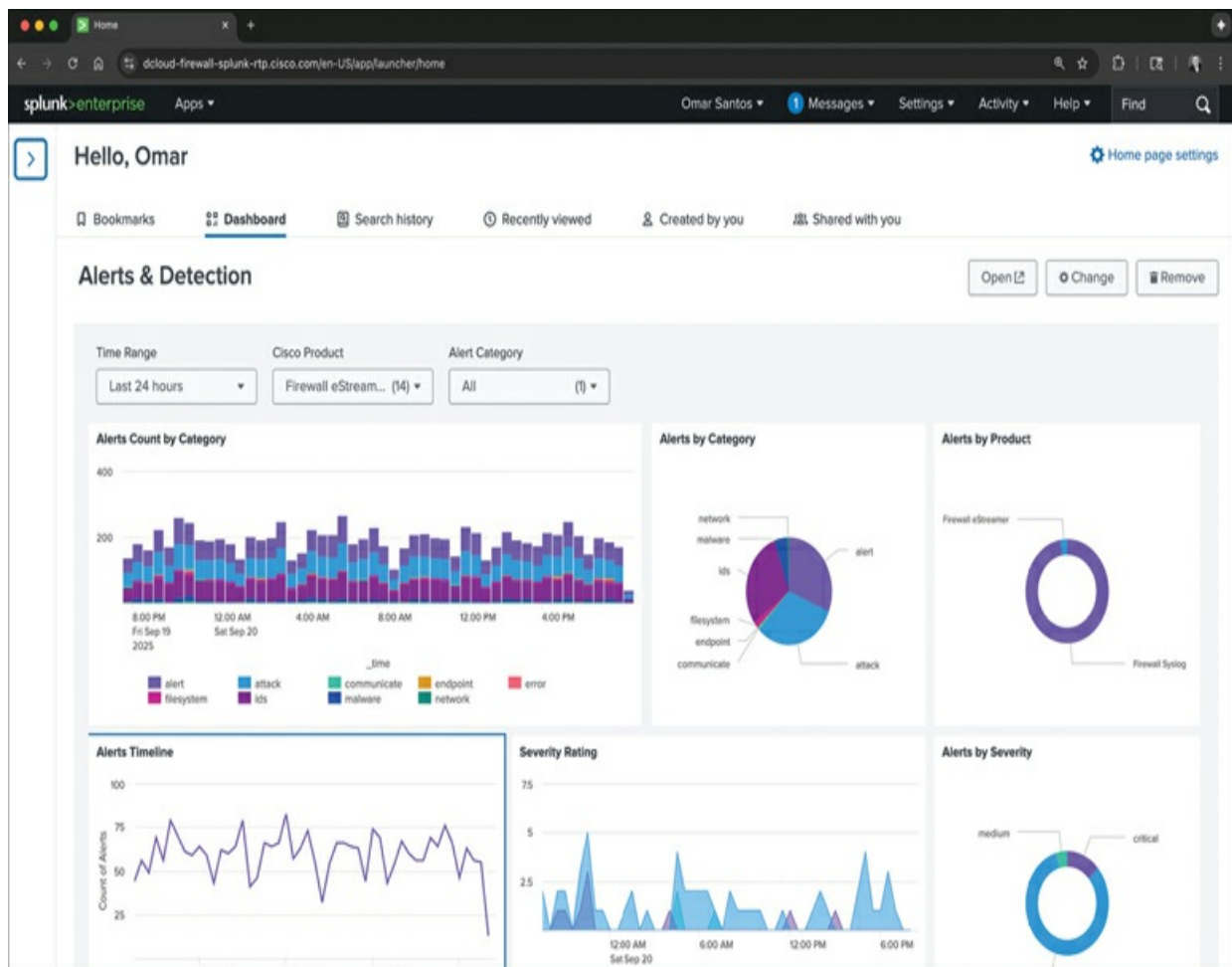


Figure 10-5 Splunk Enterprise vs. Splunk Cloud

Figure 10-5 shows a Splunk Enterprise dashboard focused on Alerts & Detection for Cisco firewall logs. The dashboard is a security operations monitoring interface that consolidates security alerts from multiple sources and visualizes them for rapid analysis. Its main purpose is to help analysts detect, investigate, and respond to security events in near real time. It identifies patterns over time, such as surges in attacks or unusual behavior.

Splunk SOAR (Security Orchestration, Automation, and Response), formerly known as Phantom, is designed to automate and orchestrate incident response workflows, force-multiplying the effectiveness of the security team. The core of Splunk SOAR is its Automated Playbooks. These are customizable workflows that codify an organization's standard operating procedures for responding to specific types of incidents, such as phishing emails, malware infections, or credential abuse. A user-friendly Visual Playbook Editor allows

analysts to build and modify these playbooks using a drag-and-drop interface with prebuilt code blocks, making automation accessible even to those without extensive coding knowledge. The platform's power is extended through its extensive library of App Integrations, supporting over 300 third-party tools and more than 2,800 distinct automated actions. In this way, Splunk SOAR can act as a central coordinator, executing actions across a multivendor security stack. Finally, it includes robust case management features, enabling teams to track, document, and collaborate on investigations from detection through resolution. Splunk also integrates with numerous ticketing systems such as ServiceNow.

Splunk offers several specialized security solutions that integrate seamlessly into the broader platform:

- **Splunk User and Entity Behavior Analytics (UEBA):** This dedicated solution uses machine learning and advanced behavioral analytics to detect subtle anomalies that may indicate insider threats, compromised accounts, or lateral movement. By baselining normal user and entity behavior, it can flag deviations that traditional, rule-based detection systems would miss.
- **Splunk Attack Analyzer:** This tool provides automated analysis of potential threats like malware and credential phishing links. When a suspicious file or URL is submitted, this tool detonates it in a secure sandbox environment and provides a detailed report on its behavior, helping analysts quickly determine its maliciousness without manual reverse engineering.
- **Splunk Asset and Risk Intelligence:** This solution addresses the foundational security challenge of asset management. It provides capabilities for continuous asset discovery, identification, and compliance monitoring, ensuring that the SOC has an accurate and up-to-date inventory of what it needs to protect.

Going Beyond Security with Observability and IT Operations

The Splunk platform extends far beyond security, as shown in [Table 10-1](#).

This versatility is a key part of its value proposition because data ingested for one purpose (e.g., IT troubleshooting) can often be leveraged for another (e.g., security investigation).

- **Splunk Observability Cloud:** This comprehensive, integrated solution monitors the performance and health of modern, cloud-native applications and infrastructure. It combines infrastructure monitoring, application performance monitoring (APM), real user monitoring (RUM), and log investigation into a single platform, giving engineering and ITOps teams full-stack visibility.
- **Splunk AppDynamics:** Acquired by Cisco several years ago and now deeply integrated with Splunk, AppDynamics is a powerful APM platform focused on observing, securing, and correlating application performance with business outcomes, particularly in complex hybrid and on-premises environments.
- **Splunk IT Service Intelligence (ITSI):** This is Splunk's AIOps (AI for IT Operations) solution. ITSI monitors the health of critical IT services rather than individual components. It uses machine learning to predict and prevent outages, reduce event noise, and pinpoint the root cause of service degradation, helping IT teams ensure business continuity.

Table 10-1 The Splunk Product Portfolio

Product/Solution	Category	Core Function/Purpose
Splunk Cloud Platform	Platform	A SaaS-based data platform for ingesting, searching, analyzing, and visualizing machine data at scale across hybrid environments
Splunk Enterprise	Platform	A self-hosted data platform providing the same core data analysis and visualization capabilities as Splunk Cloud for on-premises deployments
Splunk Enterprise Security (ES)	Security	A market-leading SIEM application built on the Splunk platform for advanced threat detection, security monitoring, and compliance
Splunk SOAR	Security	A security orchestration, automation, and response platform for automating incident response workflows and coordinating actions across tools
Splunk User Behavior Analytics (UEBA)	Security	A machine learning-driven solution for detecting insider threats and compromised accounts through anomalous behavior analysis
Splunk Attack Analyzer	Security	An automated analysis tool for investigating complex malware and credential phishing threats in a sandbox environment

Splunk Asset and Risk Intelligence	Security	An application that provides continuous asset discovery, inventory, and compliance monitoring to identify and manage the organizational attack surface
Splunk Security Essentials	Security	An application that extends Splunk Enterprise/Cloud with prebuilt security content, detections, and guidance to accelerate security use cases
Splunk Observability Cloud	Observability	A unified SaaS platform for infrastructure monitoring, APM, digital experience monitoring (DEM), and log investigation for cloud-native environments
Splunk AppDynamics	Observability	An application performance monitoring (APM) solution for observing and securing hybrid and on-prem applications and correlating performance with business outcomes
Splunk IT Service Intelligence (ITSI)	IT Operations	An AIOps solution that uses machine learning to monitor service health, predict outages, and accelerate root cause analysis for critical IT services
Splunk Universal Forwarder	Extensibility	A lightweight agent for securely collecting and forwarding data from remote sources to a Splunk deployment
Splunkbase	Extensibility	A marketplace with over 2,400 apps, add-ons, and integrations from Splunk, partners, and the community to extend platform functionality

Splunk AI Toolkit: Operational Machine Learning on the Splunk Platform

The Splunk AI Toolkit enables teams to create, validate, manage, and operationalize AI/ML models directly on the Splunk platform using a guided

user interface. It is not a turnkey, one-click solution. To be successful, you need domain knowledge, proficiency with Splunk Search Processing Language, familiarity with the Splunk platform, and foundational data science skills. When used well, the toolkit accelerates time-to-value by tightly coupling ML workflows with the data engine teams already use for security, IT, and business analytics.

Splunk provides AI/ML across the portfolio. Some features are embedded in premium products (e.g., Enterprise Security and ITSI), while the data platform allows you to build custom ML solutions. For custom ML, Splunk offers three tiers:

- **(Tier 1) Core Splunk (no additional apps):** This tier enables you to use built-in SPL commands for statistics, correlation, anomaly detection, clustering, prediction, and trending. It is best for lightweight analytics, prototyping, and operational searches that should run completely within core SPL.
- **(Tier 2) Machine Learning Toolkit (MLTK) with Python for Scientific Computing (PSC):** This tier provides Guided Assistants to build, validate, and version models; access to 30+ bundled algorithms and 300+ open-source algorithms via PSC; ML-SPL API for custom algorithms; and ONNX model import. It is best for productionizing classical ML within Splunk.
- **(Tier 3) Splunk App for Data Science and Deep Learning (DSDL) with PSC and MLTK:** This tier provides access to an external container environment connected to Splunk. It enables you to develop in JupyterLab with any open-source library, leverage multi-CPU/GPU, and manage production container models. It is best for deep learning and advanced/custom workloads.

Tip

If you are new to Splunk, begin with the Search Tutorial to learn ingestion, search, reporting, and dashboards. If you are new to the AI Toolkit, explore the Showcase examples that walk through end-to-end use cases across IT, security, business, and IoT. These examples prepopulate an Assistant, demonstrate best practices, and show ideal outcomes that you can adapt to your own data.

Figure 10-6 shows the Splunk AI Toolkit Showcase.

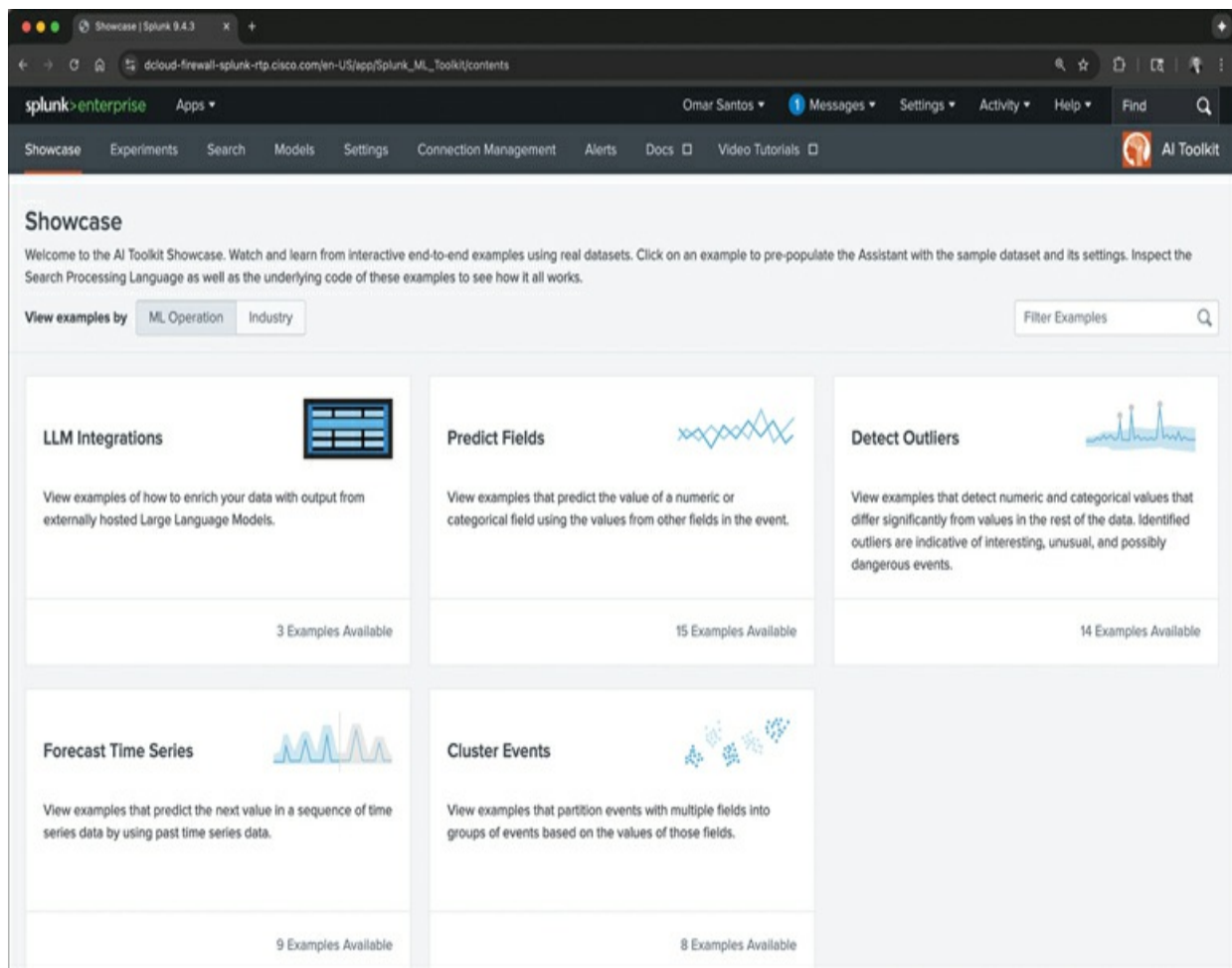


Figure 10-6 Splunk’s AI Toolkit Showcase

The Splunk AI Toolkit Showcase allows you to explore interactive, end-to-end examples using real datasets to learn how to apply ML in Splunk. Each example can be loaded into the Splunk Assistant, which prepopulates the SPL query and model configuration, making it easy to experiment and learn.

There are different available categories (as shown in the tiles on the screenshot in [Figure 10-6](#)). LLM Integrations shows how to enrich data with outputs from externally hosted large language models (LLMs). It demonstrates examples of combining Splunk data with AI-driven contextual insights. Predict Fields provides examples that predict numeric or categorical values based on other fields in events. This feature is useful for scenarios like predicting network latency, error rates, or risk scores. The Detect Outliers

contains examples for identifying unusual or anomalous data points (both numeric and categorical). This is great for security use cases like detecting abnormal login patterns, spikes in network traffic, or suspicious firewall events.

The Forecast Time Series demonstrates the prediction of future values from time-series data. This tool can be used for capacity planning, threat forecasting, or performance monitoring. You can also group events based on similar field values, enabling analysts to identify patterns and relationships. This capability can be applied to threat hunting by clustering related alerts or correlating incidents.

Cisco Vulnerability Management: Prioritizing Risk with Data Science

Cisco Vulnerability Management (CVM), formerly known as the Kenna Security platform, introduced capabilities for organizations to manage security vulnerabilities in systems and applications. This tool is built on the philosophy that not all vulnerabilities are created equal and that focusing on raw counts or static severity scores is an inefficient and ineffective strategy. Instead of just using Common Vulnerability Scoring System (CVSS) data, CVM functions as a data-driven translation layer using the Exploit Predictability Scoring System (EPSS), other data, and AI/ML. It converts the abstract and often overwhelming language of CVE and CVSS scores into the concrete, prioritized language of quantifiable risk and remediation efficiency. This method is designed to improve security operations (SecOps) and IT operations (ITOps) effectively.

CVSS and EPSS

CVSS and EPSS are both standards maintained by the Forum of Incident Response and Security Teams (FIRST). You can find the latest version of the CVSS standard and associated tools at <https://first.org/cvss>. You can find the latest version of the EPSS standard, data model, and API at <https://first.org/epss>.

Risk-Based Vulnerability Management (RBVM)

The foundation of CVM is the concept of risk-based vulnerability management (RBVM), which directly addresses the shortcomings of traditional vulnerability management practices. As mentioned earlier, for years, the industry standard for rating vulnerabilities has been the CVSS. While useful for understanding the technical severity of a vulnerability based on its intrinsic characteristics (e.g., attack vector; complexity; impact on confidentiality, integrity, and availability), CVSS has a few limitations in a real-world operational context. A CVSS score measures severity, not risk, and is not a predictor of whether a vulnerability will actually be exploited by adversaries.

This shortcoming leads to a serious operational problem: You are faced with millions of vulnerabilities across their environments, and CVSS often flags thousands of them as “High” or “Critical.” However, extensive research performed by many organizations shows that only a very small fraction (typically between 2 and 5 percent) of all published vulnerabilities are ever seen being exploited in the wild. Yet this may change over time because attackers are now using AI to accelerate exploitation. However, the majority of vulnerabilities disclosed are not being exploited in real-world attacks.

Because most organizations can only remediate about 10 to 15 percent of their open vulnerabilities in any given month, a strategy based solely on CVSS scores forces teams to expend effort on many vulnerabilities that pose little to no actual threat, while potentially neglecting lower-scored vulnerabilities that attackers are actively targeting.

The RBVM approach, as implemented by CVM, flips this model. Instead of asking “How severe is this vulnerability?” it asks “How likely is this vulnerability to be exploited in my environment, and what is the business impact if it is?” By focusing remediation efforts on the small subset of vulnerabilities that pose a genuine, measurable risk, CVM allows organizations to make their vulnerability management programs dramatically more effective and efficient, reducing the most risk with the least amount of effort.

CVSS, EPSS, and the Cisco Security Risk Score

CVSS and EPSS are still good tools for initial assessment of a vulnerability, but the power that CVM introduces is a better approach for vulnerability management. The core of the CVM platform is its proprietary risk scoring engine, which produces the dynamic and context-aware Cisco Security Risk Score. This score is the output of a sophisticated data science process that synthesizes vast amounts of internal and external data.

Data Ingestion and Intelligence Feeds

The accuracy of the risk score is directly proportional to the breadth and quality of the data the engine analyzes. CVM ingests and processes tens of billions of data points from more than 55 sources. This includes

- **Internal Vulnerability Data:** The platform integrates with all major vulnerability scanners (e.g., Qualys, Tenable, Rapid7), application security testing tools (DAST/SAST), and penetration testing results to understand the specific vulnerabilities present within an organization's unique environment.
- **External Threat Intelligence:** CVM continuously processes data from more than 18 external threat and exploit intelligence feeds. This includes information from commercial sources, open-source databases, exploit kits available on the dark web, and real-time "early warning chatter" from security forums and social media that may indicate a vulnerability is about to be weaponized.
- **Global Attack Telemetry:** The platform analyzes data from more than 12.7 billion managed vulnerabilities across its global customer base to understand which vulnerabilities are actively being exploited in real-world attacks, tracking their volume and velocity.

CVM uses AI and predictive modeling to analyze this massive dataset. It doesn't just look at past exploits; it forecasts the future risk of newly discovered vulnerabilities. A key input to this model is EPSS. CVM's own predictive models have demonstrated over 94 percent accuracy in forecasting which vulnerabilities will see active exploitation, allowing organizations to proactively manage risk.

The Scoring Mechanism, Asset Criticality, and Context

The final Cisco Security Risk Score is a dynamic, quantifiable score, typically on a 100-point scale, that represents the real-world risk a vulnerability poses to the organization at that moment in time. The calculation starts with a normalized CVSS score but then layers on all the real-time context from the intelligence feeds and predictive models. The score is continuously updated as the threat landscape changes. For example, if a new exploit for a previously low-risk vulnerability is published, its risk score in CVM will increase almost immediately. The platform also provides clear, qualitative context, such as whether a vulnerability is easily exploitable (i.e., part of a known exploit kit) or is actively being used by malware or ransomware.

A great differentiator for CVM is its ability to incorporate the business context of the asset on which a vulnerability resides. A “Critical” vulnerability on a developer’s laptop does not pose the same level of risk as the same vulnerability on a mission-critical, Internet-facing production database. CVM allows organizations to set an asset priority value (typically on a scale of 1 to 10) for individual assets or groups of assets. This priority value acts as a multiplier on the vulnerability scores for that asset. By increasing the priority of critical systems (e.g., those in the DMZ, part of a PCI environment, or those containing sensitive data), organizations ensure that remediation efforts are automatically directed toward their most important assets first, providing a true business-risk perspective.

Operationalizing Vulnerability Management

The act of identifying and scoring risk is only the first step. The CVM platform includes a suite of features designed to help organizations operationalize this intelligence and drive remediation.

The CVM platform provides highly customizable dashboards and risk meters that offer a real-time, aggregated view of risk across the organization. Risk meters can be configured to track the risk score for specific groups of assets, such as those belonging to a particular business unit, geographic location, or

application owner. This capability allows different teams to have a view of risk that is relevant to them and enables leadership to get a holistic view of the organization's overall risk posture.

To make remediation as efficient as possible, CVM provides a feature called Top Fix Groups. Instead of simply providing a long list of vulnerabilities to patch, the platform analyzes the available solutions (e.g., patches, configuration changes) and identifies the specific fixes that will reduce the most amount of risk with the least amount of effort. For example, it might determine that applying a single Microsoft patch will remediate 20 different vulnerabilities across 500 critical servers, thereby reducing the organization's overall risk score by a significant margin. This actionable guidance is precisely the information ITOps teams need to prioritize their work effectively.

CVM integrates directly with common IT service management (ITSM) and ticketing systems like ServiceNow and Jira. This integration allows the platform to automatically create and assign remediation tickets to the appropriate IT teams, complete with all the necessary context and prioritization information. This automated workflow eliminates manual processes and ensures that prioritized vulnerabilities are actioned quickly. The platform also offers comprehensive reporting and peer benchmarking capabilities, allowing organizations to measure the success of their VM program, track metrics like mean time to remediate (MTTR), and demonstrate tangible risk reduction to auditors and executive leadership.

[Figure 10-7](#) shows an example of a CVM dashboard. This dashboard is designed for risk-based vulnerability management. It allows security teams to view, prioritize, and act on vulnerabilities based on risk scores, exploitability, and asset context. The primary goal is to help focus remediation efforts on the most critical vulnerabilities that pose the highest risk to the organization. The All Groups selected here could represent business units, asset groups, or environments.

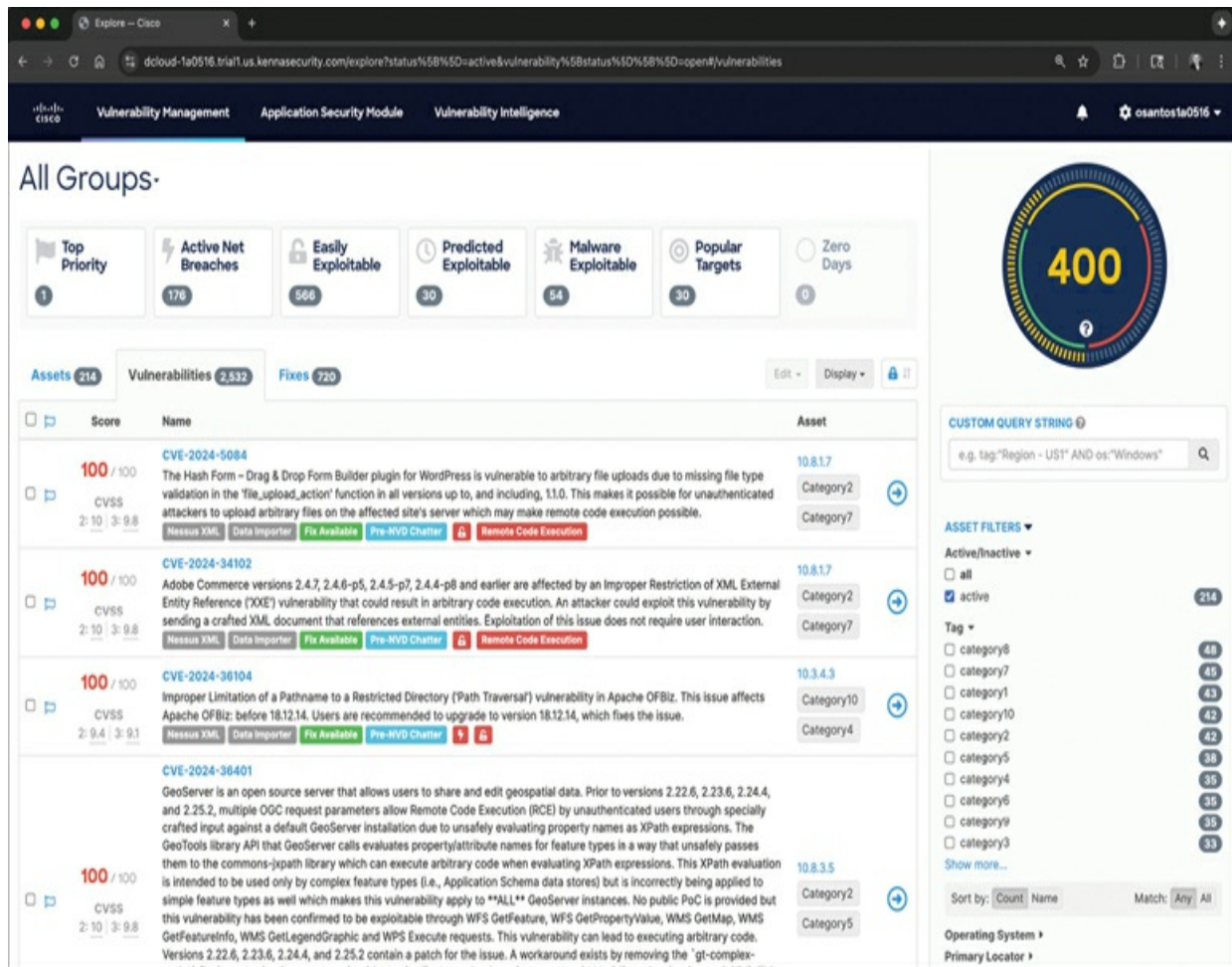


Figure 10-7 CVM Vulnerability Management Dashboard

The navigation bar (at the top of the figure) provides access to

- Vulnerability Management
- Application Security Module
- Vulnerability Intelligence
- Switching between different views and modules within the platform

The main vulnerability table displays a list of vulnerabilities (CVE-based), including

- CVM Risk Score (0–100, here showing critical scores of 100)
- CVSS base score and vector string
- CVE ID with vulnerability description

- Links to assets, tags, and exploit references (Nessus XML, Fix Available, Remote Code Execution)
- Whether a patch or workaround exists

In this figure, you can see several old vulnerabilities that have not been patched within the organization's systems.

The Power of Integration: A Unified Security Strategy

The true power of the Cisco security triumvirate is realized not when its components operate in isolation, but when they are deeply integrated, creating a security ecosystem that is far greater than the sum of its parts. This integration transforms a linear security workflow (detect, investigate, respond, patch) into a continuous, self-improving virtuous cycle of data enrichment. Each platform performs its core function while simultaneously generating context that makes the other platforms more intelligent and effective. Cisco Vulnerability Management enriches Cisco XDR's detections; XDR provides high-fidelity alerts that make Splunk's investigations more efficient; and Splunk's deep analysis can uncover new patterns that feed back into and refine future detection and prioritization strategies.

Following Cisco's acquisition of Splunk, the strategic vision has become clear: The two platforms are positioned as complementary, not convergent. This layered architecture is designed to provide organizations with both the speed required for real-time tactical response and the depth needed for strategic investigation and compliance, avoiding the compromises inherent in a single-platform approach.

The core of the relationship lies in their distinct but coactive roles. Cisco XDR excels at real-time correlation and tactical response. It is optimized to quickly process curated telemetry from critical security controls, identify high-confidence threats like ransomware or lateral movement, and execute immediate, automated containment actions. It is the SOC's "fast data" engine.

Note

This unified approach is especially powerful in a SaaS-delivered security model. Because each platform is cloud-native and continuously updated, integration doesn't require complex professional services engagements, lengthy upgrade cycles, or manual data engineering to keep systems aligned. Instead, customers benefit from an always-improving security fabric where new detections, analytics models, and threat intelligence appear organically across the stack. As Cisco and Splunk evolve their platforms, those enhancements instantly amplify the entire ecosystem. This tactic reduces operational overhead, accelerates time-to-value, and ensures organizations are always leveraging the most current defenses—not just the ones they've had time to deploy.

Splunk excels at deep search, custom analytics, and long-term data retention. It serves as the enterprise's system of record for all machine data, enabling deep forensic investigations, complex threat hunting using the powerful SPL, and comprehensive compliance reporting. It is the SOC's "big data" engine.

This relationship enables several powerful, practical workflows that leverage the strengths of both platforms:

- **Detection Handoff:** A typical workflow begins in Cisco XDR. The platform's analytics engine detects a sophisticated, multistage attack and generates a single, high-fidelity incident, complete with a visual map of the attack chain. This contextualized incident is then automatically forwarded to Splunk Enterprise Security. A senior analyst in the SOC can then pivot directly into the Splunk console to conduct a deeper investigation, correlating the incident with months or even years of historical log data from sources outside the XDR's purview to determine the full scope and timeline of the breach.
- **Enrichment and Custom Detection:** The workflow can also operate in the reverse direction. An analyst in Splunk ES might create a highly specific, custom correlation search to detect an attack pattern unique to their industry or organization. When this custom detection triggers an alert in Splunk, it can be forwarded to Cisco XDR, which can then initiate a coordinated, automated response action, such as quarantining the involved endpoints across the enterprise.

- **Unified Response Orchestration:** The two platforms can work in concert to deliver a comprehensive response. Cisco XDR can handle the initial, rapid containment actions (e.g., isolating a host, blocking a malicious domain). Simultaneously, the alert can trigger a more complex playbook in Splunk SOAR. This SOAR playbook could orchestrate a multistep workflow that involves creating a high-priority ticket in ServiceNow, sending detailed notifications to the legal and HR departments, and querying third-party threat intelligence services for additional indicators of compromise (while maintaining a detailed case log for post-incident review).

Splunk, Cisco XDR, and CVM Technical Integration Points

The technical backbone for this interoperability is the Cisco Security Cloud app for Splunk. This application, available on Splunkbase, provides a seamless integration experience. It includes modular data inputs for various Cisco security products, ensuring that telemetry and alerts are ingested efficiently. Critically, the app maps the incoming Cisco data to Splunk's Common Information Model (CIM). This is what allows the data to be immediately usable within Splunk Enterprise Security's correlation rules, dashboards, and analytics without requiring extensive custom parsing, thus accelerating time-to-value.

The integration of Cisco Vulnerability Management makes Cisco XDR a significantly more intelligent and context-aware platform. By understanding the vulnerability posture of assets involved in a security incident, analysts can make faster and more accurate decisions. When Cisco XDR is integrated with CVM, it can ingest asset and vulnerability data, including the all-important Cisco Security Risk Score. This vulnerability context becomes a critical input for XDR's own incident scoring and prioritization engine. For example, an alert indicating suspicious behavior on a server might normally be assigned a "Medium" priority. However, if CVM data shows that this specific server is running an outdated operating system with multiple high-risk vulnerabilities that are known to be actively exploited by ransomware gangs, Cisco XDR can automatically elevate the incident's priority to "Critical." This approach ensures that analysts are always focusing on the incidents that represent the

greatest potential business impact.

During an incident investigation within the Cisco XDR console, analysts are presented with a consolidated view of the device, which now includes its vulnerability data from CVM. With a single click, they can see the asset's overall risk score and a list of its specific CVEs. This immediate access to vulnerability context helps them quickly understand potential attack vectors—how the adversary might have gained entry or escalated privileges—and identify systemic weaknesses in security controls that need to be addressed. This shared context between the incident response and vulnerability management teams is vital for ensuring that both teams are focused on mitigating risk on the most critical assets.

Leveraging Vulnerability Intelligence in Splunk

The rich, data science-driven intelligence from Cisco Vulnerability Management (CVM) can also be leveraged directly within the Splunk platform to enhance threat hunting and investigation.

CVM provides a dedicated threat intelligence feed that can be ingested by Splunk. This feed contains a wealth of information, including detailed CVE data, associated CVSS scores, real-world exploit data, trends in vulnerability chatter, and historical changes to the Cisco Security Risk Score. This intelligence is typically stored in Splunk as lookup tables, making it easily accessible for correlation with other data sources.

Correlating Threats with Vulnerabilities

Within Splunk Enterprise Security, analysts can use this vulnerability intelligence to create powerful correlation searches. They can correlate real-time security events (such as an intrusion alert from a firewall, a malware detection from an endpoint, or a suspicious web request from a proxy) with the known vulnerabilities on the involved assets. This information allows them to answer critical questions that provide deep investigative context, such as

- “Is this network attack targeting a specific vulnerability that our CVM data shows is unpatched and high risk on the destination server?”

- “Show me all endpoints that have communicated with this malicious IP address in the last 24 hours AND have a critical, remotely exploitable vulnerability.”
- “Alert me if any of our Internet-facing servers have a new vulnerability for which CVM has detected active, in-the-wild exploitation.”

A Practical Workflow: From Vulnerability Awareness to Orchestrated Response

To demonstrate the power of this three-way integration, consider the following step-by-step workflow for a common attack scenario:

- **Discovery and Prioritization (Cisco Vulnerability Management):** A security researcher publicly discloses a new, critical remote code execution vulnerability in a widely used web server application. Within hours, CVM’s intelligence engine detects a significant spike in chatter about the vulnerability on security forums and observes the publication of a proof-of-concept exploit. Its predictive models flag the vulnerability as highly likely to be exploited. The platform automatically assigns it a high Cisco Security Risk Score. Simultaneously, CVM cross-references this with the organization’s internal vulnerability scanner data and identifies three Internet-facing production servers that are vulnerable.
- **Detection and Context (Cisco XDR):** An attacker, using the newly published exploit, successfully compromises one of the vulnerable web servers. The attacker then begins to perform reconnaissance and attempt to move laterally to a database server on the internal network. Cisco XDR’s behavioral analytics engine, which has baselined normal traffic patterns, immediately detects this anomalous east-west traffic. It generates a high-fidelity incident, automatically chaining together the initial exploit, the command-and-control communication, and the lateral movement attempt. The XDR incident is instantly and automatically enriched with data from CVM, showing the responding analyst that the compromised server has the critical, actively exploited vulnerability that was identified just hours before.
- **Investigation (Splunk Enterprise Security):** The enriched incident

from Cisco XDR is forwarded to Splunk ES, appearing on the primary incident review dashboard. A senior SOC analyst takes ownership and pivots into the Splunk console for a deep-dive investigation. Using SPL, the analyst searches through months of historical logs from firewalls, proxies, and authentication servers to determine if the attacker's IP address has been seen before or if other assets are exhibiting similar, more subtle signs of compromise. The analyst uses the CVM vulnerability intelligence lookup table within Splunk to quickly generate a list of all other assets in the environment that have the same critical vulnerability, identifying potential targets for the attacker.

- **Orchestrated Response (Cisco XDR and Splunk SOAR):** Based on the investigation, the analyst confirms a critical breach is in progress and initiates a Splunk SOAR playbook designed for this scenario. The playbook executes a series of automated and coordinated actions across the ecosystem:
 1. It makes an API call to Cisco XDR to immediately isolate the compromised web server from the network and block the attacker's source IP address at the perimeter firewall.
 2. It automatically creates a "Critical–Remediate Now" ticket in the company's ITSM tool, assigning it to the server administration team. The ticket is prepopulated with the exact patch information recommended by CVM's "Top Fix Groups" feature.
 3. It sends a high-priority notification to the cybersecurity leadership team via email and a secure chat channel, providing a summary of the incident and the actions taken.

The entire workflow, including all actions taken and evidence collected, is documented in the Splunk SOAR case file for post-incident analysis and reporting.

Cisco's AI Assistant

The Cisco AI Assistant is a generative AI– and natural language–driven interface layered across Cisco's security and networking portfolios. It accesses a large scale of data to more intelligently guide and inform decision-

making. In this way, you are able to work faster, safer, and smarter.

With that foundation, the AI Assistant becomes a key enabler of the unified security strategy. The following are the major capability buckets in which the AI Assistant plays a role, aligned to the integrated security lifecycle:

- The AI Assistant surfaces deeper insights by leveraging broad telemetry and context: devices, applications, networks, security events, and Internet-wide data. For example, when a firewall administrator is asked which policies are in place for an application or what rules might be shadowed, the AI Assistant can respond in seconds rather than manual digging.
- It enables decision-makers (SecOps, network operations, compliance) to ask natural-language prompts such as “What rules apply to SalesApp outbound?” or “Which policies need cleanup?” and receive actionable answers.
- The AI Assistant supports automating tasks such as policy creation, rule optimization, and change management. For instance, administrators can use it to generate firewall access rules from English language prompts.
- It flags redundant, duplicate, or conflicting rules and suggests optimizations (e.g., remove obsolete rules, tighten access).
- This workflow automation reduces the manual burden on security operations teams and thereby speeds response and service delivery.
- The AI Assistant amplifies human analysts by preprocessing context, surfacing relevant data and elevating high-confidence actions. For example, “Which firewall rule conflicts may be causing latency?”
- It helps shorten the investigation loop: Instead of analysts manually gathering logs, flows, configurations, and asset dependencies, the AI Assistant guides them via conversational interfaces and prioritized suggestions. Because it sits within the integrated stack, insight from one domain (e.g., endpoint, network) can be brought to bear more quickly across domains, supporting the “data enrichment” cycle in your earlier text.
- The AI Assistant isn’t just static: Via feedback mechanisms (thumbs up/down, revise prompt) it learns from administrator behavior and gets

smarter over time. With full access to Cisco's native telemetry (e.g., more than 550 billion security events processed per day), the system can apply machine-scale learning in security operations. As part of your unified architecture, this means the AI Assistant helps fuel the "self-improving virtuous cycle" you described: Enrichments lead to better detections, which lead to better investigations and policies, which feed back into analytics and tuning.

The AI Assistant helps simplify complexity around policies and rules. It can assist with compliance-oriented tasks such as retrieving support case history, checking rule lifecycles, and identifying policy drift. It also operates under Cisco's data-governance and privacy framework, giving administrators control over data, disabling features if desired, and ensuring that training does not consume customer-specific PII. For sophisticated environments (e.g., regulated industries), this means the AI Assistant can reduce audit burden, help maintain policy hygiene, and support consistent enforcement.

Summary

The strategic combination of Cisco XDR, the Splunk platform, and Cisco Vulnerability Management creates a security operations ecosystem that is unequivocally greater than the sum of its parts. This integrated triumvirate provides a comprehensive solution that addresses the full threat detection, investigation, and response (TDIR) lifecycle in a way that siloed products cannot. It begins with the proactive, intelligence-driven risk reduction of Cisco Vulnerability Management, which narrows the attack surface by focusing remediation on the threats that matter most. It continues with the real-time, high-speed detection and automated containment provided by Cisco XDR, which excels at stopping common, high-impact attacks in their tracks. Finally, it offers the unparalleled depth of the Splunk platform for deep forensic investigation, custom threat hunting, and long-term compliance, ensuring that no threat, no matter how esoteric, is beyond the reach of analysis. This layered, symbiotic architecture delivers both the speed necessary for tactical defense and the depth required for strategic security intelligence.

Pervading this entire ecosystem is the transformative power of AI. This is not a future promise but a present reality, with AI and machine learning

embedded into the core of each platform. Cisco Vulnerability Management uses predictive AI to forecast which vulnerabilities will be exploited. Cisco XDR leverages machine learning for behavioral analysis and offers an AI Assistant to guide analysts through complex investigations and responses. The Splunk platform utilizes AI for intelligent event correlation in ITSI and is increasingly incorporating generative AI to allow analysts to query vast datasets using natural language.

Looking ahead, the continued integration of these platforms under the umbrella of the Cisco Security Cloud points toward a future of more predictive, automated, and resilient security operations. As the platforms share data and intelligence more seamlessly, the virtuous cycle of enrichment will accelerate. The SOC of the future will not be defined by its ability to react to thousands of alerts, but by its ability to proactively reduce risk, automatically contain the threats that do emerge, and leverage AI to augment human expertise for the most complex challenges. The integrated suite of Cisco XDR, Splunk, and Cisco Vulnerability Management provides a powerful and adaptable foundation for building that future, purpose-built for the complexity of the AI era.

References

- Cisco XDR:
<https://www.cisco.com/site/us/en/products/security/xdr/index.html>
- Automated Ransomware Recovery with Cisco XDR At a Glance:
<https://www.cisco.com/c/en/us/products/collateral/security/xdr/automated-ransomware-recovery-with-xdr-aag.html>
- Cisco XDR Demos and Webinars:
<https://www.cisco.com/site/us/en/products/security/xdr/demos.html>
- Cisco XDR Integrations:
<https://www.cisco.com/site/us/en/products/security/xdr/integrations.html>
- Splunk documentation: <https://help.splunk.com/en>
- Splunk Lantern Customer Success Center: <https://lantern.splunk.com/>
- Splunk Community: <https://community.splunk.com/>

- Splunk getting started guidance:
https://community.splunk.com/t5/Welcome-Center/Navigating-the-Splunk-Community/ta-p/755847/redirect_from_archived_page/true
- Splunk tech talks: <https://community.splunk.com/t5/Splunk-Tech-Talks/bg-p/splunktechtalks>

Chapter 11. Observability and Monitoring: AppDynamics and Splunk

The pace of innovation is rapidly increasing in today's digital landscape. As companies shift more workloads to the cloud, IT is left to deal with the challenges of managing workloads across multiple domains. Often, applications may be spread across on-premises, public cloud, private cloud, and SaaS domains. This arrangement results in increased complexity and can hinder an organization's ability to draw business insights from this disparate data.

Application performance monitoring (APM) solutions have been around since the '90s, with companies like Precise, Wily, and Quest Software being the early leaders in the APM market. However, it wasn't until around 2008 that companies like Dynatrace, AppDynamics, and New Relic were releasing more modern APM solutions. Early on, AppDynamics saw that SaaS would be a key differentiator in the industry and that companies would migrate their workloads to the cloud. So, it developed its APM software to support SaaS solutions while also offering an on-premises deployment option.

In 2017, Cisco acquired AppDynamics, an industry leader in application monitoring. The AppDynamics APM suite allows customers to monitor business applications' performance and interactions with other applications and backends (i.e., database, third-party web services, messaging servers). This telemetry can give customers greater visibility into their overall application health and performance while delivering real-time insights to help them achieve more significant outcomes. With AppDynamics, Cisco could now provide solutions that enable customers to gain visibility across their network, data center, security, applications, and end-user monitoring.

In March 2023, Cisco acquired Splunk to enhance its security and observability capabilities further. The integration of the companies, along with Cisco's already industry-leading monitoring and observability solutions with AppDynamics, would give Cisco the ability to offer customers solutions that could not be rivaled by anyone else. Splunk + AppDynamics + AI is a solution that gives customers visibility and insights across their entire digital footprint. AppDynamics and Splunk are uniquely positioned in today's growing SaaS market to help customers understand their data spread across on-premises and cloud environments.

If you are interested in Cisco's monitoring and observability solution using AppDynamics and Splunk, this chapter is for you. Specifically, we will cover the following topics:

- **What Is Full-Stack Observability?:** We'll cover full-stack observability (FSO), why it is essential in the industry now, and how FSO differs from application monitoring.
- **MELT:** We'll introduce the concept of MELT, which stands for metrics, events, logs, and traces, and how each of these is used in observability solutions.
- **OpenTelemetry:** We'll introduce Open Telemetry, an open-source, open-standard project leveraged by many industry observability platforms.
- **AppDynamics and Splunk Overview:** We also will cover how Splunk and AppDynamics deliver a stronger monitoring and observability solution for customers.
- **Security Practices Used by Splunk and AppDynamics:** We'll cover some of the SaaS security practices used by Splunk and AppDynamics.
- **Observability Architecture:** We'll provide a brief overview of the architectural differences between Splunk and AppDynamics deployments.
- **Core Observability Features:** Finally, we'll cover the most used observability features across both Splunk and AppDynamics. This discussion includes how these features work and why they should be utilized.

The Basics

Before we discuss Splunk and AppDynamics' specifics on monitoring and observability, it is helpful to address key terms and technologies that power these solutions. These technologies or terms are foundational elements to understand before we dive into specific tooling leveraged by Splunk and AppDynamics.

What Is Full-Stack Observability?

When you hear the term *full-stack observability (FSO)*, what comes to mind? If you were to ask a software developer to define full stack, they would likely tell you that this term describes both client- and server-side development. This requires a working knowledge of front-end (HTML, CSS, JavaScript, etc.), back-end (Java, C++, Python, etc.), and database development. Each of these items combined would then represent the full stack.

While full-stack observability may encompass many items, it is not limited to application monitoring and visibility. Instead, it combines data from various domains, such as network, security, infrastructure, applications, and user experience, to help customers derive real-time insights about their business.

Modern digital applications are spread across various platforms and services, and correlating the data between them can be challenging for IT, Ops, and business groups. Organizations often have operationally siloed teams that focus on a specific area. These teams often have a set of tools and telemetry that they leverage to perform their job. This means that the telemetry that SecOps cares about may not be the same telemetry that DevOps cares about. [Figure 11-1](#) shows teams that may operationally work to support the same application but have minimal overlap in scope and are often siloed from one another.

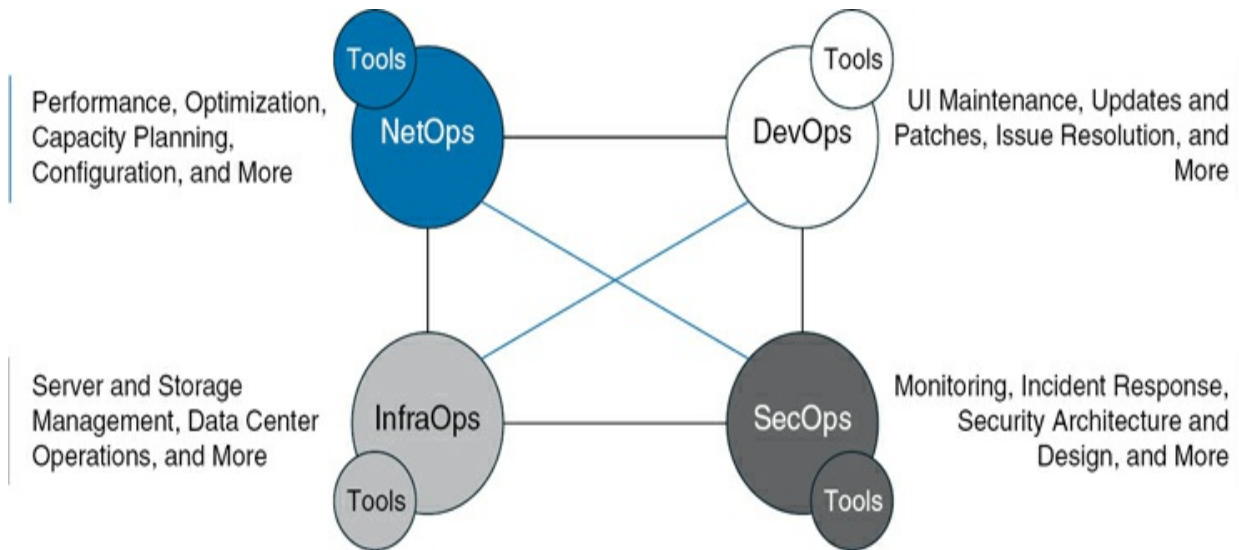


Figure 11-1 Operations Teams Within an Organization

Full-stack observability delivers visibility into your technology stack, allows you to gain valuable insights from telemetry, and helps you prioritize actions to improve your business goals. Instead of making business decisions based on siloed pools of data, teams can make smarter business decisions by correlating data across their technology stack.

Observability vs. Monitoring

Two terms that you may hear a lot in the observability space are *monitoring* and *observability*. You might even hear these terms being used seemingly interchangeably in reference to monitoring and observability solutions. However, they have distinct differences that are important to understand.

Monitoring is about understanding the overall performance, user experience, and availability of software applications. A monitoring solution gathers data from systems over time and analyzes this data to understand the overall health of that system. Often, monitoring solutions operate in real time, allowing IT administrators to gain immediate awareness of issues that may be impacting an application's performance or user experience. Some of the key features of monitoring are

- End-user experience monitoring
- Real-time application performance and visibility tracking

- Robust application health and security monitoring
- Metrics (error codes, API calls, CPU, throughput, and so on)
- Traces for application errors to assist with debugging

Observability, unlike monitoring, looks at the output of a system and makes inferences about the internal health of that system based on its outputs over time. The metrics collected by a monitoring solution could be an output of that system and utilized by the observability solution. Observability takes monitoring one step further and seeks to provide insights into the entire infrastructure surrounding an application, including network, storage, compute, containers, databases, and applications. Observability bridges the gaps between siloed operational components and helps IT administrators understand the downstream impact of changes or issues across the entire technology stack.

For example, the team that handles storage and computing at a company may have no visibility or awareness of the impacts on end users when they make a change or when an issue is observed in their environment. A change in the compute environment that causes a CPU spike or an increase in latency would appear problematic in your application monitoring tooling. Still, you may not realize the downstream impact of those issues until later. However, with an observability solution, changes to one place within your infrastructure can be easily correlated to impact across your entire solution, allowing you to quickly isolate, fix, and even prevent the issue from happening again.

Some of the key features of observability are

- The ability to view the entire IT infrastructure
- The ability to correlate data and events across the whole technology stack
- The ability to proactively identify issues or security risks
- Extensibility to allow custom integrations, which can provide more visibility into infrastructure and applications

You could say that monitoring is simply a subset of observability. The better your monitoring solution is at collecting and correlating events for an

application, the better the observability solution will be at correlating that data across your entire technology stack. Observability is about connecting the dots between your data so that IT can make informed, swift, and impactful decisions.

OpenTelemetry and MELT

OpenTelemetry is an open-source, vendor-neutral framework for generating, collecting, managing, and exporting telemetry. It offers a variety of tools and software SDKs for efficiently gathering and exporting telemetry from your applications in a standardized format that downstream observability tools, like the Splunk Observability Cloud, can easily consume. You can find more information on OpenTelemetry at <https://opentelemetry.io>.

OpenTelemetry has a broad library of already-supported languages, allowing you to easily enable the collection and export of telemetry from your existing applications and tools. OpenTelemetry is also extensible, allowing you to develop custom integrations to collect telemetry from any number of data sources if you do not find an existing library or SDK to leverage.

A few benefits of using OpenTelemetry are that it is vendor agnostic, so you aren't susceptible to vendor lock-in. Unlike vendor-specific APM solutions, it also gives you complete control over your data and how it is processed. It also supports sending data to any monitoring backend, again not locking you into a specific vendor.

The primary purpose of OpenTelemetry is to collect and export what OpenTelemetry calls "signals." Signals are outputs from systems such as metrics, events, logs, and traces (MELT). Each signal type provides unique insights into an application's health and behavior. When this data is combined, it gives a holistic view of the application. [Table 11-1](#) shows some data that fits into the MELT datatype.

Table 11-1 Metrics, Events, Logs, and Traces

Signal	Description	Examples
Metrics	Aggregated set of measurements grouped or collected at regular intervals or a given time span; not discrete.	CPU Usage (%), Memory Usage (MB), Average Response Time (ms)
Events	A discrete action happening at a moment in time.	Container Restart, VM Power Off, Sales Transaction, User login
Logs	Strings of text with a discrete timestamp associated with them; unstructured or structured.	[DEBUG] 2025-09-26 23:08:06 MSExchangeServer HandleDBRequest - received request for lookup from [client_ip=x.x.x.x] and processing took 34 ms
Traces	Chains of events (or transactions) between different components in an application. Traces are discrete and irregular in occurrence.	A SQL query execution

One way to think of MELT data and how it is used for monitoring and observability is to think of the metrics, events, logs, and traces as different layers of an iceberg, as seen in [Figure 11-2](#).

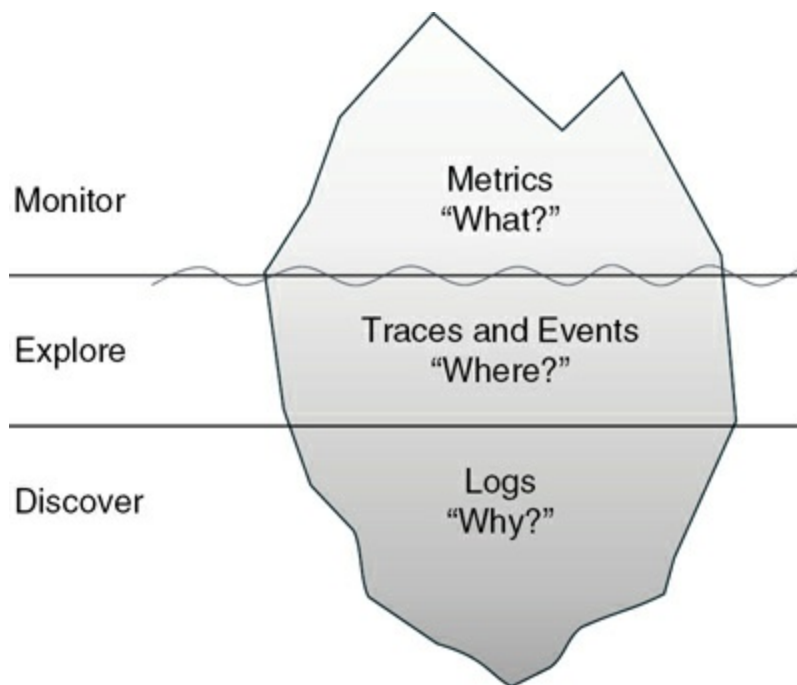


Figure 11-2 Metrics, Events, Traces, and Logs Represented as an Iceberg

At the top of the iceberg, you have your metrics. These are the signals that something is going on. They are the first indicators or metrics that show “what” is happening. For example, this could be a metric showing a spike in CPU utilization. As you look further down the iceberg, you would look to your traces and events to determine “where” the issue is coming from. This is where you would isolate further down in your application to understand what might be causing the CPU increase. And then, finally, you have your logs, which can show the “why” behind the issue.

Note

Although MELT is the commonly used acronym in the observability space for the types of telemetry used in observability solutions, OpenTelemetry has extended these telemetry types to now include two other telemetry types, or signals as OpenTelemetry refers to them: baggage and profiling. You can find more details on the signals used and supported in OpenTelemetry at <https://opentelemetry.io/docs/concepts/signals/>.

The OpenTelemetry framework is a core component of Splunk’s Observability Cloud. This framework allows the Observability Cloud to be vendor-agnostic and highly extensible to support telemetry from any number of systems and tools. It can then leverage this data to provide intelligent insights, allow you to set specific objectives and KPIs around that data, and deploy custom actions based on events occurring in real time.

AppDynamics + Splunk

We cannot discuss Cisco observability solutions without discussing AppDynamics and Splunk. Until the acquisition of Splunk, AppDynamics and Cisco Observability Platform were Cisco’s product offerings for monitoring and observability: AppDynamics hybrid application monitoring for on-premises applications and Cisco Cloud Observability, under the Cisco Observability Platform, for cloud-native applications.

There are many different reasons why Cisco acquired Splunk, including

bolstering its security business and expanding its observability platform. If you think about Cisco and Splunk from a monitoring and observability perspective, Cisco has a footprint in almost every enterprise with its vast range of IT solutions. This data, combined with Splunk's data security and intelligence capabilities, gives Cisco a massive advantage in the observability space.

The Splunk acquisition meant that Cisco owned two of the leading monitoring and observability companies: AppDynamics and Splunk. AppDynamics leads the way with its on-premises and hybrid monitoring capabilities, and Splunk's Observability Cloud would become the primary solution for cloud-native application observability.

Note

With the acquisition of Splunk and the overlap in product features between the Cisco Observability Platform, Cisco Cloud Observability, and Splunk Observability Cloud, the decision was made to consolidate these tools all into the Splunk Observability Cloud. This consolidation simplifies the solution and reduces tool overlap while combining the best of both tools into one platform for customers to consume. All customers leveraging the Cisco Observability Platform or Cisco Cloud Observability will eventually be migrated to the Splunk Observability Cloud. Looking ahead, Cisco is committed to delivering a truly unified observability experience, seamlessly integrating the best-in-class capabilities of both AppDynamics and Splunk into a single, comprehensive platform to provide unparalleled visibility and insights across the entire digital landscape.

Cisco's observability portfolio can be viewed as a combination of several tools, all of which are combined to provide digital intelligence across a software portfolio. [Figure 11-3](#) depicts the available platforms that make up this portfolio.



Figure 11-3 Full-Stack Observability Portfolio

Splunk Enterprise or Splunk Cloud is used as the centralized logging platform, AppDynamics for on-premises/hybrid application observability, Splunk Observability Cloud for cloud-native application observability, and Splunk IT Service Intelligence as the single pane of glass that brings actions and insights from all these components together. This portfolio provides companies with many capabilities, as seen in [Figure 11-4](#).

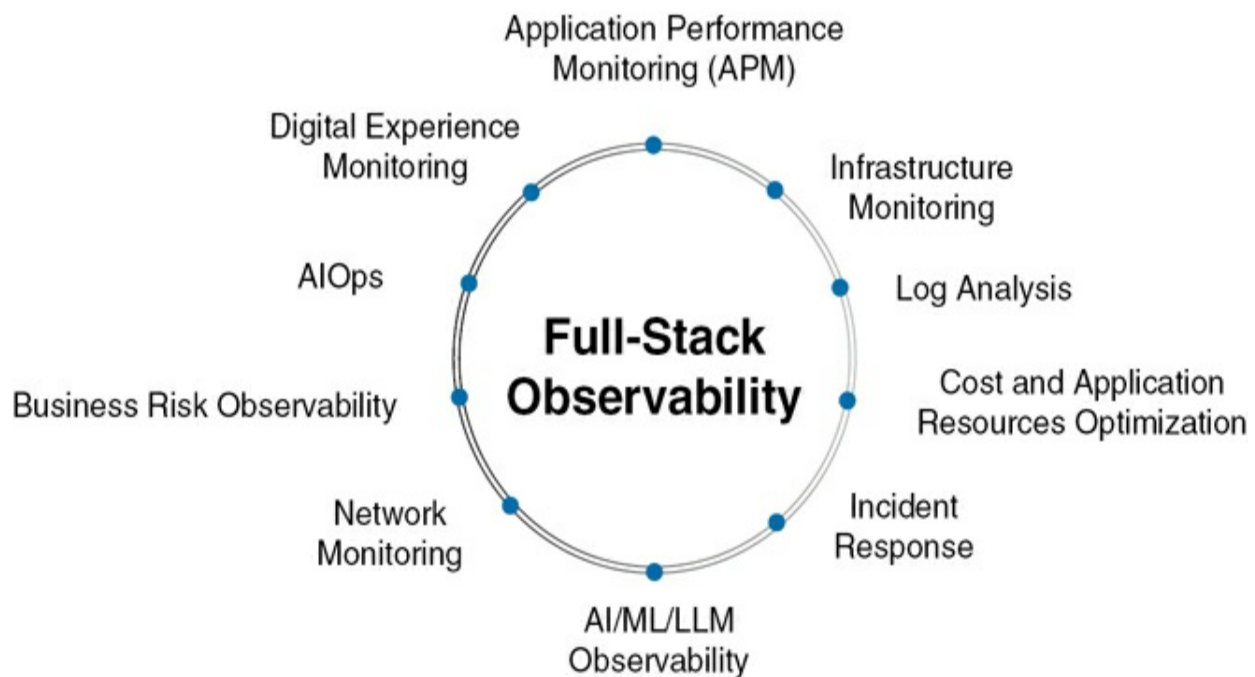


Figure 11-4 FSO Capabilities with Splunk + AppDynamics

- **Application Performance Monitoring:** View real-time metrics and analytics on specific applications running in your network. Performance

metrics include CPU and memory utilization, application flow monitoring to see what systems your application is communicating with, and event monitoring to determine when an issue occurs.

- **Infrastructure Monitoring:** Monitor key infrastructure in your network, such as server monitoring, database visibility, and cloud infrastructure monitoring.
- **Log Analysis:** Apply intelligence to your application and infrastructure logging to determine when issues arise to help IT teams isolate issues faster, contain incidents, and maintain system performance.
- **Cost and Application Resource Optimization:** Manage your cloud computing cost by leveraging intelligent analytics to deliver optimization recommendations for your cloud and Kubernetes (K8s) environments to reduce overall spending.
- **Incident Response:** Take advantage of visibility into key areas of your network, including logging and system performance, and allows incident teams to triage and resolve incidents much faster, reducing your application's exposure threats.
- **AI/ML/LLM Observability:** Monitor your company's use of artificial intelligence, machine learning, and large language model tooling to understand spending, token usage (for LLMs), and latency for connections to services like OpenAI.
- **Network Monitoring:** Monitor the connectivity between your applications to understand whether the network carrying your data is causing issues with application performance and to monitor your network health.
- **Business Risk Observability:** Discover vulnerabilities in your applications and pinpoint specific business transactions to which these vulnerabilities are linked.
- **AIOps:** Leverage the data and analytics captured from your business environment and apply ML and AI to improve IT visibility and predict when issues might occur to enhance IT operations.
- **Digital Experience Monitoring:** Understand and predict application-impacting events through synthetic monitoring, which mimics customer

interactions with your application, and real user monitoring, which records customer interactions to help swiftly diagnose and solve problems.

Note

Splunk offers many other products and features in addition to the ones mentioned in this chapter. Here, we focus specifically on monitoring and observability and the primary Splunk features that enable this. For more details on the features and capabilities available from Splunk, you can visit <https://www.splunk.com>.

SaaS Security Practices—AppDynamics and Splunk

AppDynamics and Splunk offer their services through a SaaS subscription, which allows the infrastructure for the service to be maintained and hosted in the cloud. They also offer enterprise or on-premises services where companies can host and maintain their applications on their own. Still, because this book's primary focus is on SaaS services, the SaaS components will be the focus of this chapter.

In [Chapter 2, “SaaS Architecture,”](#) and [Chapter 4, “Security and Privacy for SaaS,”](#) you learned about SaaS architecture principles, security practices, and challenges. So, let's explore how AppDynamics and Splunk leverage SaaS best practices to ensure your customers' data is secure and to reduce their threat surface for bad actors looking to steal sensitive data.

Data Encryption

Understanding how data is transmitted and stored for SaaS applications is imperative. Data encrypted at rest but unencrypted in transit can expose customer data before it reaches the SaaS services, making the data encryption inside the service pointless. Data encrypted in transit but not encrypted at rest could expose data if the SaaS service was breached by external bad actors.

AppDynamics and Splunk encrypt data in transit by default and support data encryption at rest. They leverage Advanced Encryption Standard (AES) 256-bit encryption, an industry-standard and FIPS-approved algorithm used to

protect digital data, including data backups, personally identifiable information (PII), and customer-identifiable data.

For data transmitted to and from the SaaS tenants, either to other cloud services or to on-premises customer networks over the Internet, SSL/TLS is leveraged to encrypt the data so that services in between cannot read the payload and expose sensitive data.

Access Control

The ability to manage how your data is accessed and to control various levels of access is an essential concept for SaaS because it ensures that users get access only to the data that they need. This approach follows the principle of least privilege, which essentially says that users should have access only to what they need to do their job, no more, no less. For example, this could mean that someone in finance should not be able to access source code for software engineering projects, and someone in engineering should not be able to access sensitive financial data.

Both Splunk and AppDynamics support role-based access control (RBAC), which allows administrators to create customized levels of access and assign those roles to users to ensure that users have access only to what they need. By defining roles with access to specific resources, you can assign the necessary roles to users to grant them access to the resources they need.

[Figure 11-5](#) depicts how RBAC limits resource access based on a given role.

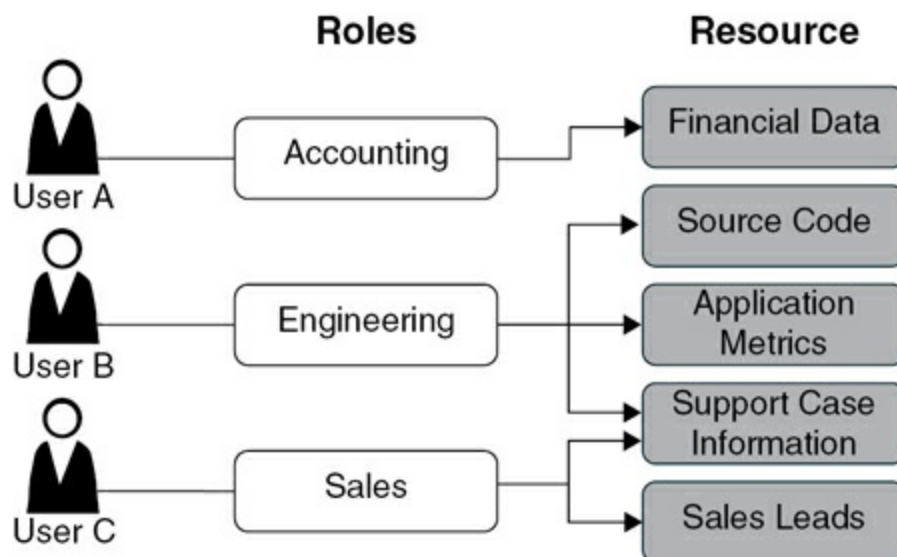


Figure 11-5 Role-Based Access Control

Disaster Recovery

Disaster recovery is the practice of restoring services and data if a “disaster” or outage were to take place. Since SaaS solutions are hosted in the cloud, and a single SaaS solution hosts many different customers on its platform, having an outage—or worse, some type of loss of your data—has a huge impact on the SaaS provider and for the customers relying on that platform. So, having a plan in place to ensure that your service can still run and data can be recovered should a disaster occur is vital. Many customers ensure that the SaaS providers they select have a disaster recovery strategy and plan before they even consider using that SaaS solution.

Splunk Cloud leverages a primary and secondary environment for customer organizations. When an organization is set up in Splunk Cloud, a secondary environment in a different region is also created, and data is continually replicated between these environments. If a disaster happens, the primary environment fails over the secondary environment, ensuring continuity of access. You can find more details on Splunk Cloud’s disaster recovery strategy at

<https://docs.splunk.com/Documentation/SplunkCloud/latest/DR/HowitWorks>.

Figure 11-6 depicts what a disaster recovery failover would look like for a Splunk deployment.

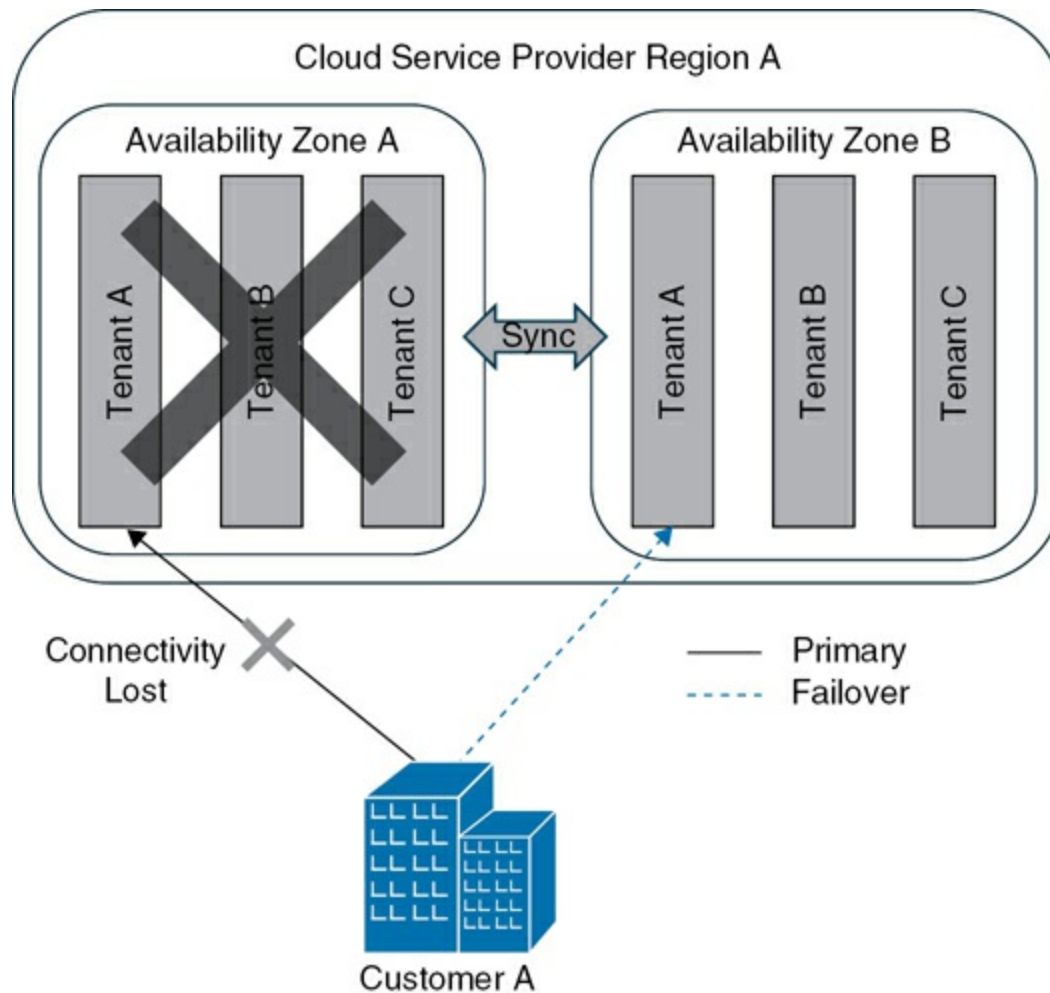


Figure 11-6 Disaster Recovery

In this example, Customer A's connectivity to their tenant was lost due to an issue with the availability zone where the tenant was hosted. When this happened, all traffic was automatically redirected to their backup/failover tenant in a different availability zone that was not impacted.

Note

Availability zones and regions are common cloud service provider (CSP) services that allow for redundancy and continuity of service if there is an issue with a data center in a specific region. Most large CSPs offer these services. For more details on availability zones and how they are leveraged for SaaS applications, check out this article: https://www.splunk.com/en_us/blog/learn/availability-zones.html.

AppDynamics leverages a similar high-availability environment for SaaS customers, allowing them to fail over customer organizations to a new location if there is an issue with the region in which they are currently hosted. You can find more details about AppDynamics's high availability at <https://www.appdynamics.com/trust-center/operations>.

Audit Logging

Audit logging is necessary for many reasons, both for product security and for understanding how a product is being used. Regulatory standards that may be imposed on certain customers may also require audit logging. SOC1 and ISO27001, for example, both require audit logging.

Splunk and AppDynamics support audit logging for their SaaS solutions, allowing their customers to get insight into how the product is being used, monitor them for unusual behavior, and ultimately act as a record of the actions taken on the system.

Multitenancy

[Chapter 2](#) covered the concept of multitenancy to help you better understand how SaaS solutions can segment customers' data on a cloud-hosted solution. AppDynamics and Splunk are also SaaS offers that rely on multitenancy within their cloud platforms to segment customer tenants from one another. Let's explore what this looks like specifically for AppDynamics and Splunk.

For AppDynamics, customers purchase a SaaS license that will give them a tenant on a SaaS controller (we will discuss controllers in more depth later in this chapter). These customers can add users and administrators to this tenant to manage and view the data coming into the controller. Additionally, they can set up agents to send data to a specific controller and tenant. What is important to note about this controller is that multiple customers could be using the same SaaS controller. However, customer data is segmented into distinct tenants on that controller. [Figure 11-7](#) highlights what this scenario would look like.

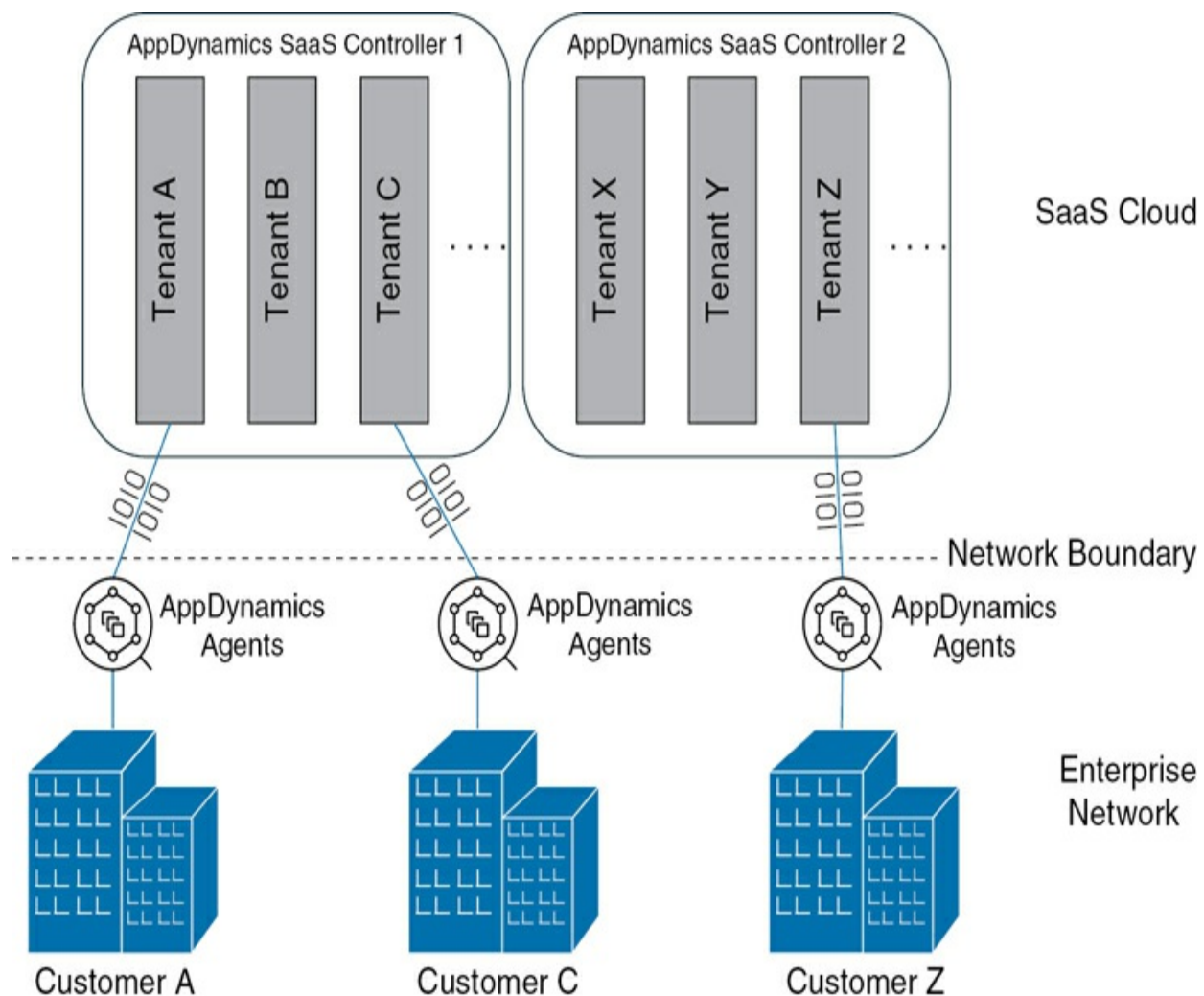


Figure 11-7 AppDynamics SaaS Multitenancy

In [Figure 11-7](#), you can see that Customer A has deployed AppDynamics agents in its enterprise network to pull data from applications and infrastructure. This data is transmitted to a specific tenant on the SaaS controller in the cloud. Although this controller is a shared environment with multiple tenants used by other customers, each customer's data is shared only with their specific tenant.

Splunk also leverages a multitenancy configuration for its cloud SaaS offering. Most SaaS solutions leverage a multitenant deployment because it allows for shared resources across multiple customers, which makes the solution scalable and much easier to maintain and deploy. The basic idea is that not every customer gets a dedicated hardware/software stack when they are onboarded. Instead, each Splunk instance is hosted on a shared resource

while data stored on each instance is separated using various secure data segmentation techniques.

Splunk Observability Cloud—Architecture Overview

While Splunk Observability Cloud and AppDynamics are both observability solutions, their architecture differs in important ways. Here, we will cover how the Splunk Observability Cloud architecture is pieced together to deliver its Observability solution. Later, we will cover the AppDynamics architecture, which will give you a better understanding of the differences between these solutions and how they are similar in many ways.

Cloud-Native Observability

The first point to note about Splunk Observability Cloud is that it is a SaaS/cloud-only solution. There is no on-premises version of the Splunk Observability Cloud that you can purchase and host in your data center. It is a cloud offer where the core components of the Observability solution are hosted in Splunk's SaaS cloud offering. The only on-premises components are the software instrumentation and OpenTelemetry components you may deploy to collect data from your applications to send to the Observability Cloud. [Figure 11-8](#) shows the high-level architecture of the Splunk Observability Cloud.

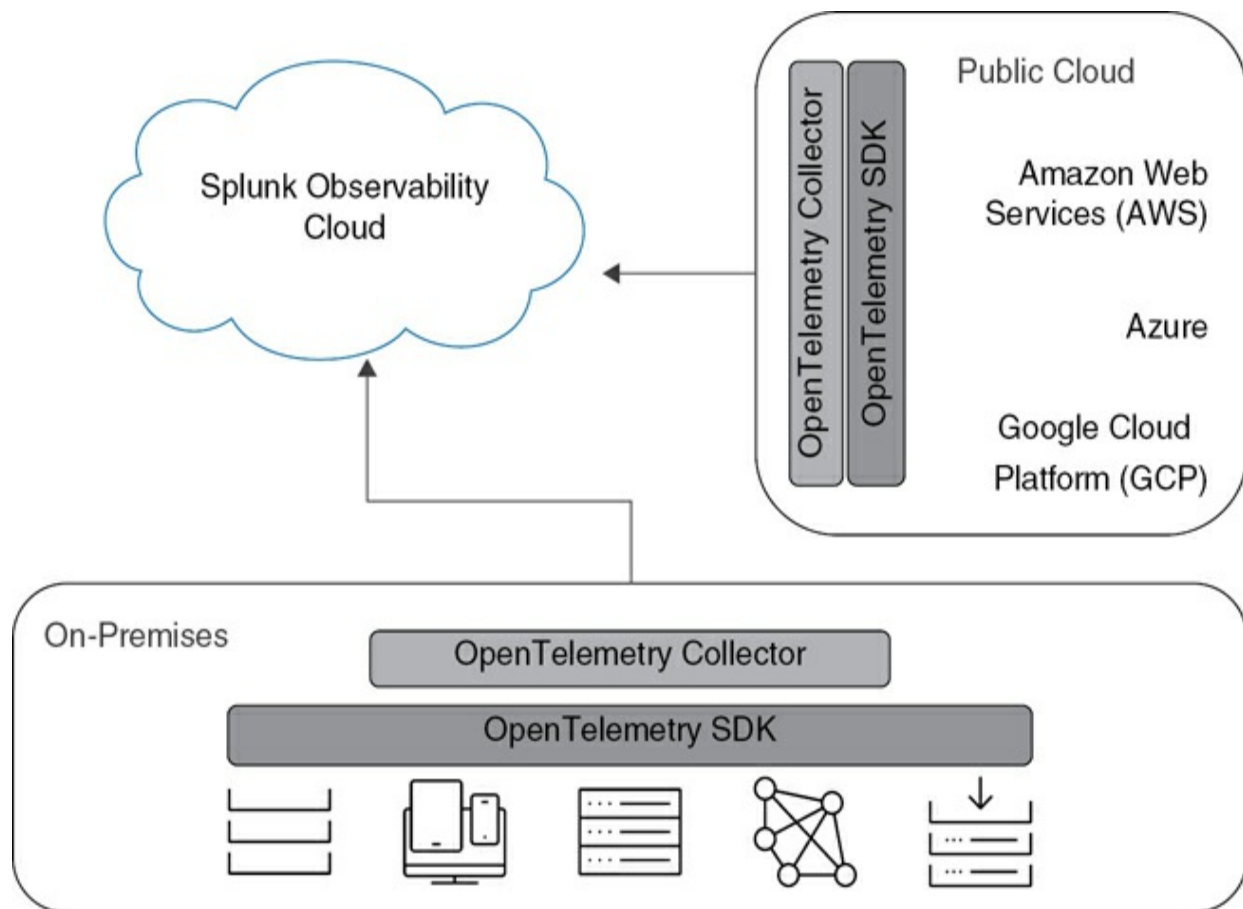


Figure 11-8 Splunk Observability Cloud High-Level Architecture

Data Collection

As you can see from [Figure 11-8](#), the primary method of collecting data is through OpenTelemetry instrumentation. Splunk Observability Cloud was built around the OpenTelemetry standard, allowing customers to customize their APM deployments and what data is collected. It also avoids vendor lock-in because OpenTelemetry data can be sent to any Observability backend.

For on-premises infrastructure, the OpenTelemetry SDK and OpenTelemetry Collector are the standard approaches for collecting telemetry for the Observability Cloud. Applications, devices, containers, and databases are all collected through OpenTelemetry deployments and configuration.

For public cloud infrastructure, there are multiple means of collecting data. The first is through direct integrations between Splunk Observability Cloud

and the public cloud where your hosts and infrastructure reside. The cloud integration between Splunk Observability Cloud and public clouds like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Azure primarily collects metrics related to your deployment and infrastructure. The metrics collected, by default, will depend on the public cloud you connect to and what metrics they support. After you have set up an integration with a public cloud, Splunk will also provide prebuilt dashboards for metrics around the infrastructure it is monitoring. Figure 11-9 shows an example of a prebuilt Splunk dashboard for an AWS cloud integration with Splunk Observability Cloud.

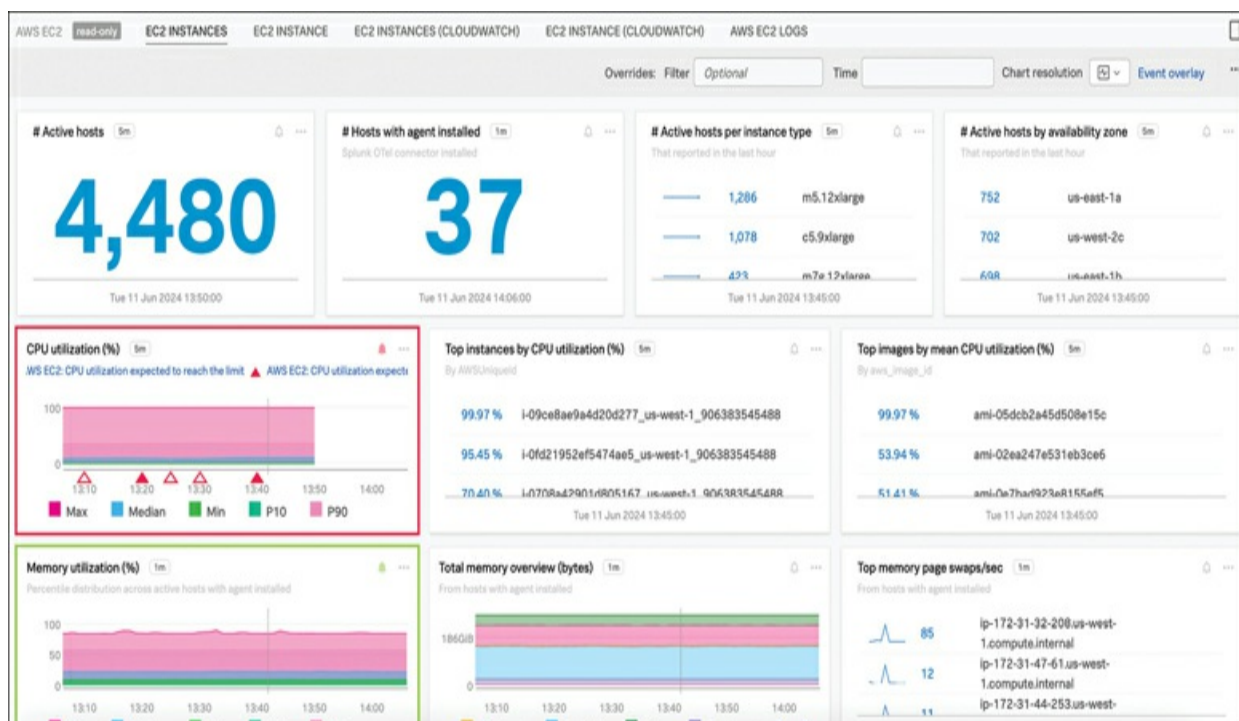


Figure 11-9 Splunk Observability Cloud AWS Prebuilt Dashboard

In addition to the native integration between Splunk Observability Cloud and AWS, GCP, and Azure, OpenTelemetry Collectors need to be deployed to gather additional metrics and logs from infrastructure that the cloud-to-cloud integration cannot collect. These OpenTelemetry Collectors are needed to gather deeper metrics and logs for containers, virtual machines, OSs, and so on. Additionally, OpenTelemetry SDKs can be used to collect application-specific metrics and traces for any software applications running in the environment.

Data collection is primarily done through OpenTelemetry Collectors and SDKs for on-premises deployments and infrastructure because there are no direct integration capabilities between Splunk Observability Cloud and on-premises environments.

AppDynamics—Architecture Overview

Similar to the way we covered Splunk’s architecture, it is helpful to begin with some of the core concepts behind AppDynamics architecture before jumping into the core features of an AppDynamics deployment. After you understand some of the larger pieces that make up a typical AppDynamics deployment architecture, you will be better prepared to understand the product’s various features.

SaaS and On-Premises Observability

AppDynamics has two primary offerings for hybrid application monitoring: SaaS and on-premises. In this way, customers have the flexibility to choose whether they want the service to be cloud-hosted and managed or to own and manage their own AppDynamics application. One key difference between an AppDynamics hybrid application monitoring deployment and a Splunk Observability Cloud deployment is that AppDynamics, while fully capable of monitoring applications and their underlying compute infrastructure deployed within cloud environments, does not offer the same level of native integration with cloud provider–managed services and platform-level infrastructure metadata as Splunk Observability Cloud. [Figure 11-10](#) shows what the architecture for an AppDynamics SaaS deployment looks like.

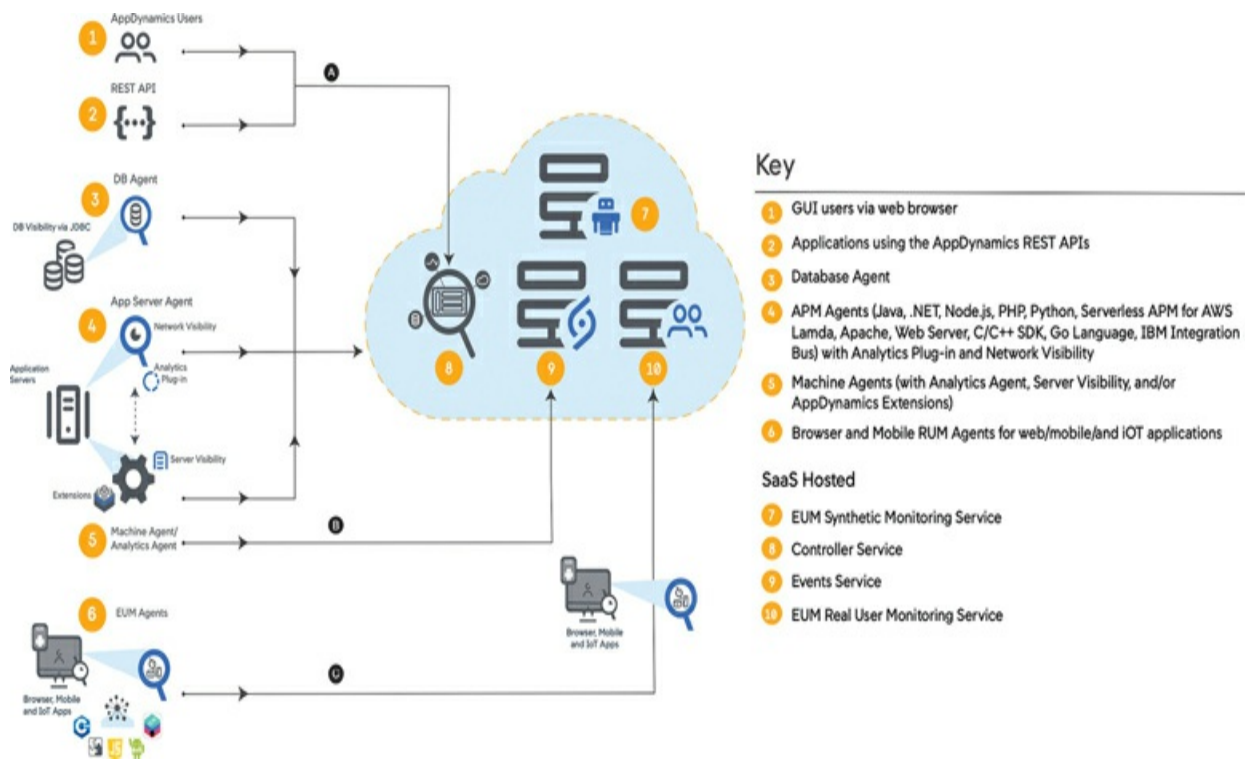


Figure 11-10 AppDynamics' SaaS Deployment Architecture

In [Figure 11-10](#), you can see that the core infrastructure for the deployment is all hosted in the cloud. The main components deployed on-premises are the AppDynamics agents, which are used to collect telemetry from various components in the network. We will discuss agents in more detail later in this chapter. The benefit of a SaaS AppDynamics deployment is that there is no required maintenance or deployment of the AppDynamics Controller or various other services. Instead, those are all maintained and hosted as a part of the SaaS offering.

On-premises deployments of AppDynamics require, at a minimum, a deployment of an AppDynamics Controller, which can be seen in [Figure 11-11](#) as key number 9. The other services, such as the EUM monitoring service, custom EUM geo server, and EUM synthetic server, are required only if your deployment will include an end-user monitoring setup or synthetic monitoring. We will discuss these features in more detail later in this chapter. [Figure 11-11](#) shows what an on-premises AppDynamics architecture looks like.

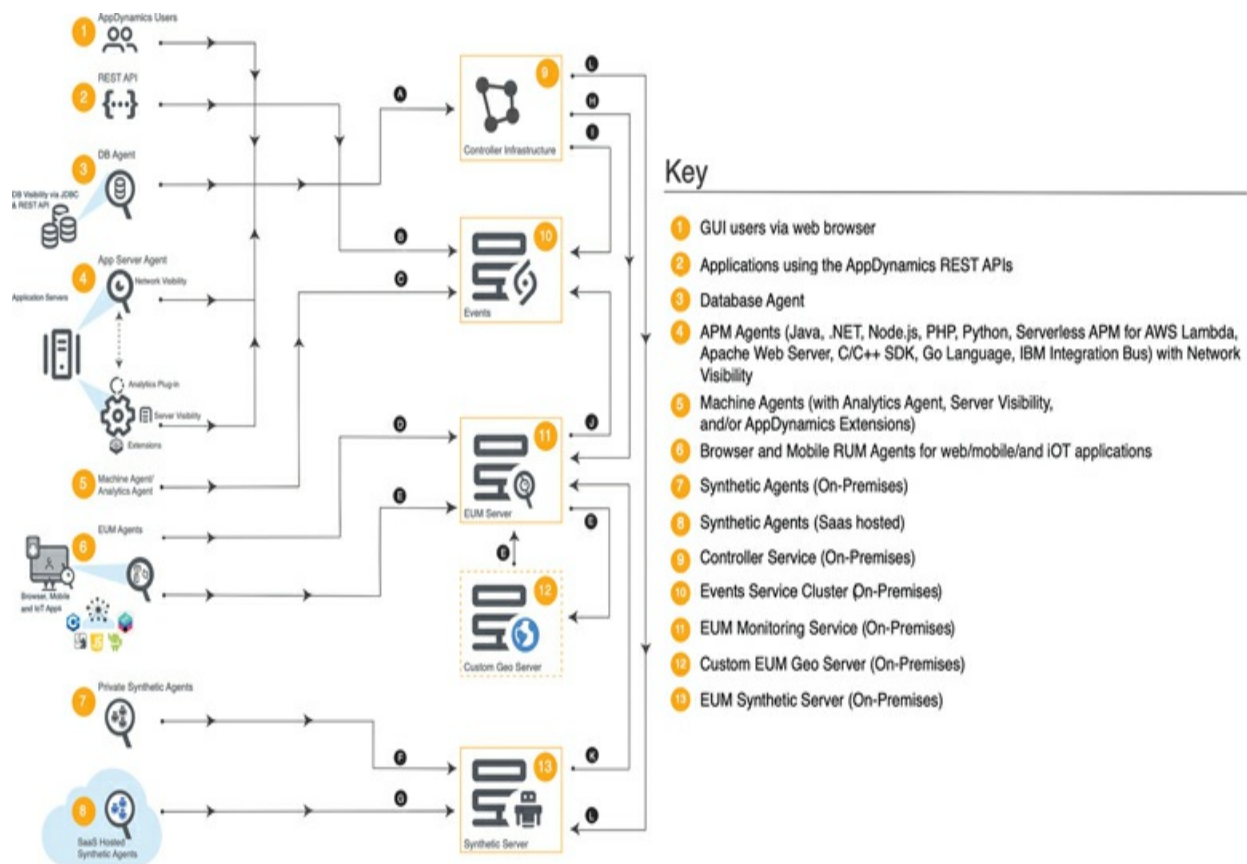


Figure 11-11 AppDynamics' On-Premises Deployment Architecture

Now that you have a better understanding of AppDynamics' primary deployment types, let's explore two key components of any AppDynamics deployment: controllers and agents.

Controllers and Agents

Unlike Splunk, AppDynamics uses controllers and agents to collect APM data from applications. Controllers and agents are fundamental when considering or deploying an AppDynamics solution. They act as the primary source for data collection, storage, and processing, so understanding how each operates is essential. This contrasts how Splunk leverages OpenTelemetry to collect telemetry and metrics.

Controllers

Controllers are the primary collection point for AppDynamics agents. Data is

collected from various applications and infrastructure using AppDynamics agents and sent to a controller. There are two types of AppDynamics controllers: SaaS and on-premises. The SaaS controllers are cloud-hosted offerings that allow customers not to have to manage and host the controller to which their data is being sent. As you might imagine, the on-premises controller requires the customer to host and manage the software on their network.

So why might you choose one over the other? Security and compliance are the biggest reasons you may select an on-premises controller over a SaaS cloud controller. By hosting the controller on-premises, you maintain control over the protection of the data hosted on that controller. For some organizations with strict data regulations, on-premises deployments may be their preferred option. The SaaS cloud controller, by default, leverages a multitenant model where multiple customers may be hosted on a controller instance in the cloud. However, if a customer requires it, a customer can be provided a dedicated controller and tenant so that it is not shared with other customers. For multitenant deployments, each customer is assigned to a tenant that segments their data and users from other customers. This solution is more approachable for some organizations, so they don't have to manage software versions, hosting, and maintenance costs for an on-premises controller instance.

Now that you understand the types of controllers available from AppDynamics, let's examine what controllers are used for. You can think of the controller as the brain of your AppDynamics deployment. Agents are deployed on hosts (physical, virtual, or cloud) or containers (orchestrated with Docker, K8s, Amazon ECS, and so on) running applications, databases, and so on, to collect data, just as humans gather data from their surroundings through touch, smell, taste, sound, and sight. This data is then sent back to the brain, or the controller for AppDynamics, to be processed, and decisions can then be made about that data.

Continuing with the human senses analogy, if you were to reach out your hand over a fire, the warmth of the fire against your hand would be data that the nerves in your hand are collecting, and this data would be sent back to the brain to interpret and make decisions about. The brain would determine that this heat could harm you, telling you to move your hand away from the fire.

Similarly, AppDynamics agents collect data from applications, physical/virtual/cloud machines, containers, and databases, which are sent back to the controller for processing. If it notices that the data is abnormal or outside of standard thresholds, it can create an alert to notify an administrator that something is wrong.

The controller is also the management and dashboard interface where all the data for your AppDynamics deployment, including application performance monitoring data, is displayed. [Figure 11-12](#) shows an example of AppDynamics controller deployments. In this figure, the Python and Java applications report data to a SaaS controller tenant, and the Ruby application sends data to an on-premises controller.

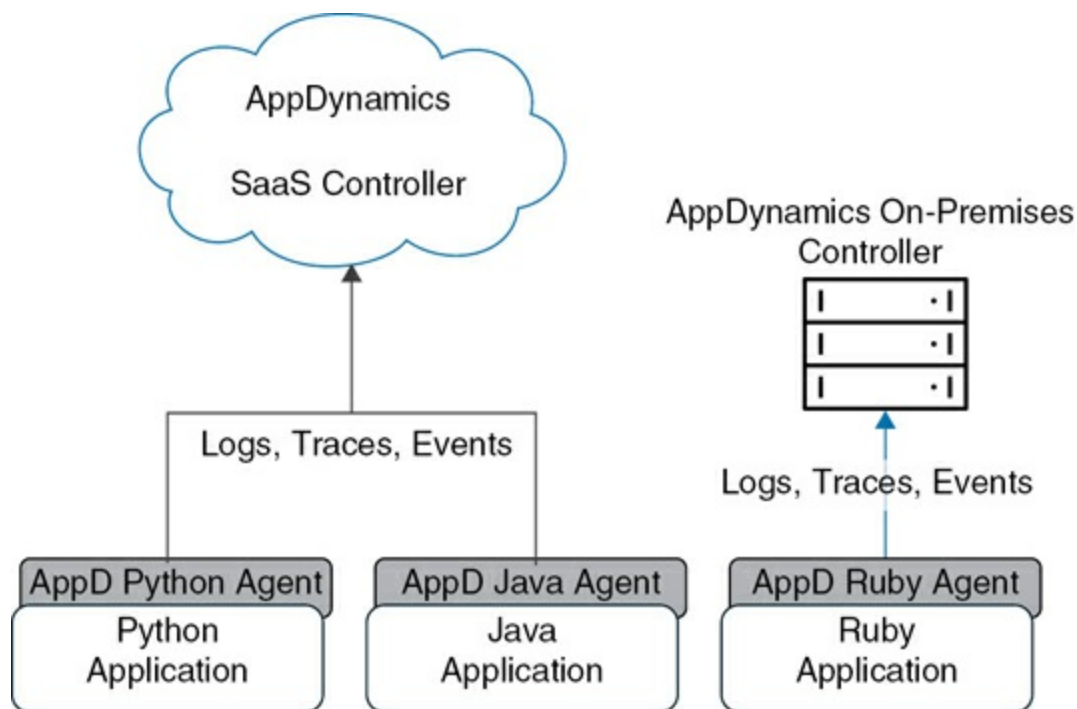


Figure 11-12 AppDynamics Controller Deployments

Note

It is not common for a single customer to have both a SaaS tenant and an on-premises tenant. [Figure 11-12](#) shows examples of what a SaaS and on-premises controller deployment might look like for application performance monitoring.

Agents

For AppDynamics, agents are among the most critical components of a deployment. There are many different agent types to explore. However, the primary purpose of an agent is to collect data. That data could be about the performance of an application, infrastructure and servers, K8s and docker deployments, or even databases. AppDynamics has developed several different agents to support a wide range of products. You can break down the AppDynamics agents into six different agent types:

- **Application Agent:** Attaches to a software application to gather runtime information, stack traces, errors, and business transactions from the application. AppDynamics supports many different programming languages and operating systems for the application agent.
- **Database Agent:** Provides performance metrics about your database, monitoring SQL query performance, wait times for locks, queries, sorting, and so on. It also provides server-level metrics for the hardware the database is running on, including disk, CPU, and memory utilization.
- **Machine Agent and Cluster Agent:** Provides hardware metrics from the infrastructure (that is, on-premises physical/virtual host, cloud VMs, K8s clusters) your applications or business processes are flowing through or running on.
- **Analytics Agent:** Captures data to help describe business information from your application, such as the number of transactions, unique browser sessions, browser and mobile user count, browser session bounce rate, and many other items.
- **JavaScript Agent:** Is injected into a user's browser session to gather more data about that user's session on the web page.
- **Synthetic Agent:** Tests and simulates end-user workflows using an application, allowing you to test for performance and errors occurring in your application without relying on end users to report an issue.

Note

Although agents are not the only method of collecting data for observability, they are the primary means of data collection for

AppDynamics deployments. There are, however, APIs and SDKs that can also be leveraged to collect data from devices like mobile and Internet of Things devices. We will cover this topic in more detail later in the “[End-User Monitoring](#)” section.

Depending on the architecture you are looking to deploy AppDynamics within and the areas in which you need observability, the agents you deploy would differ. Almost all deployments will have some application agent to give insights into the specific application you want to monitor. Past that, you may deploy database agents to provide end-to-end visibility into database calls from your application. If your application has a front-end web page, you can deploy the JavaScript agent for end-user monitoring capabilities. There is no one-size-fits-all deployment when it comes to observability. The key is having access to a set of tools that will allow you to achieve your business objectives. AppDynamics agents, APIs, and SDKs allow for a wide range of capabilities that suit most observability needs.

Core Observability Features

Across the Splunk Observability Cloud and AppDynamics hybrid application monitoring solutions, a set of core features allows these solutions to collect data across various network components, applications, and infrastructure to deliver observability and insights for enterprise networks.

Note

Because Splunk and AppDynamics are now both Cisco-acquired companies, you will begin to see these solutions combine efforts in overlapping areas of their observability and monitoring solutions. We will take the best of each and combine them to build a platform that will meet the needs of any customer looking for an observability solution.

Full-stack observability requires many components to work together to provide enough telemetry to deliver a full-stack view of a workflow, business application, user experience, and so on. No single feature or tool is used to produce a full-stack observability view for your business. For both Splunk Observability Cloud and AppDynamics, several major features work together

to deliver a piece of the full-stack observability solution. [Figure 11-13](#) identifies each feature and where they fit into an FSO deployment.

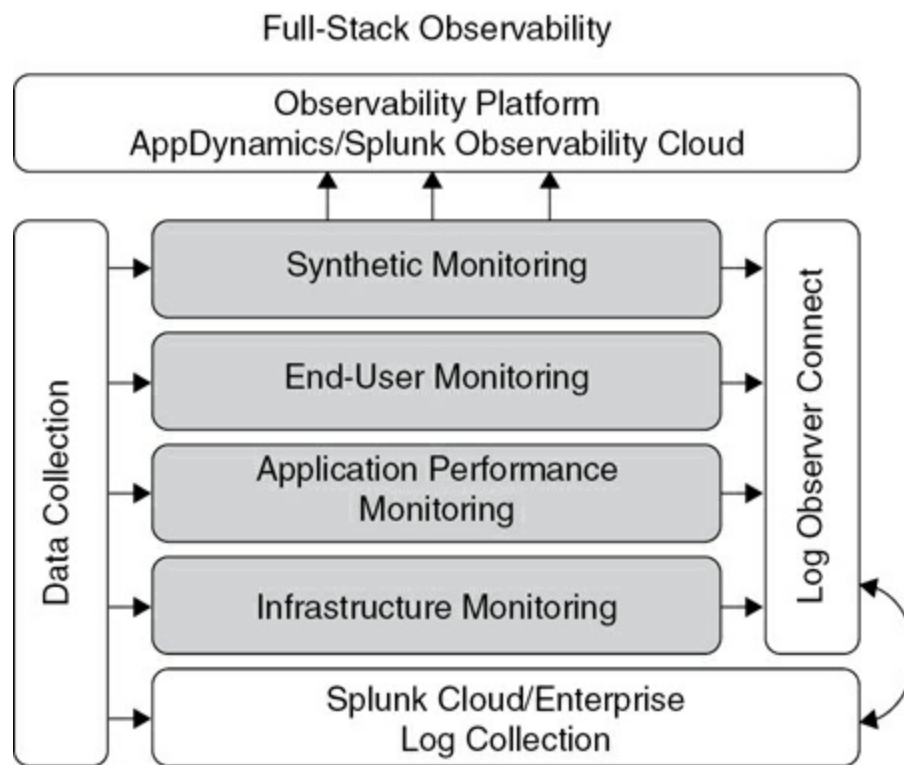


Figure 11-13 Features of a Full-Stack Observability Solution

[Figure 11-13](#) shows the five main components that deliver the full-stack observability capability: synthetic monitoring, end-user monitoring, application performance monitoring, infrastructure monitoring, and log observer connect. Let's explore each of these features and how they work.

Application Performance Monitoring (APM)

One of the most powerful tools in an IT administrator's toolkit is the ability to monitor application performance in real time to understand the overall health of critical applications running in an enterprise. Waiting for end users to report a problem is no longer an acceptable way to manage business-critical applications. Performance or application issues can directly lead to a poor end-user experience; for customer-facing applications, this experience could mean a loss in sales and revenue. Customers expect applications to work reliably and quickly. When that is not happening, it can be detrimental

to your business. Application performance monitoring (APM) solutions, like those offered by AppDynamics and Splunk, provide actionable insights and predictions about the performance and health of your infrastructure and applications, allowing you to deliver excellent customer experiences.

Application performance monitoring is a practice and set of tools that monitor core metrics and telemetry of your applications. This includes monitoring things like

- CPU performance
- Memory utilization
- Error rates
- Response times or delay
- Count of instances
- Request rates
- Application traces and code-profiling
- User experience

The data collected by an APM deployment is then sent back to the monitoring backend to be processed. This allows for tasks such as automatically generating a topology of your application and displaying the various components that an application is communicating with. Or it allows you to measure the end-to-end performance of an application transaction to identify areas that are causing a delay or a problem. You can even set up custom alerts to notify you when an application does not meet specific service-level agreements (SLAs). This data can even allow you to diagnose code-level problems that are causing a problem with your application.

Splunk and AppDynamics offer APM solutions, but as we discussed earlier in this chapter, they each have unique strengths that make each solution viable depending on the type of infrastructure that needs to be monitored. Splunk's APM solution primarily uses OpenTelemetry collectors and instrumentation to collect data from various applications and send that data back to Splunk. AppDynamics, however, leverages custom-built agents that send data directly back to the AppDynamics controller.

Note

Although the primary means of collecting APM data for an AppDynamics deployment is using the AppDynamics agents, AppDynamics does support collecting data via OpenTelemetry deployments. This is done through the service called AppDynamics for OpenTelemetry. You can find more details about this deployment and configuration at

<https://docs.appdynamics.com/appd/24.x/latest/en/application-monitoring/splunk-appdynamics-for-opentelemetry>.

Although the OpenTelemetry deployment gives greater control of what data is collected and allows for preprocessing of the data on the OpenTelemetry collector, it does not support more granular code profiling such as class/method-level details like AppDynamics agents. [Figure 11-14](#) compares what a basic APM deployment would look like for AppDynamics versus Splunk.

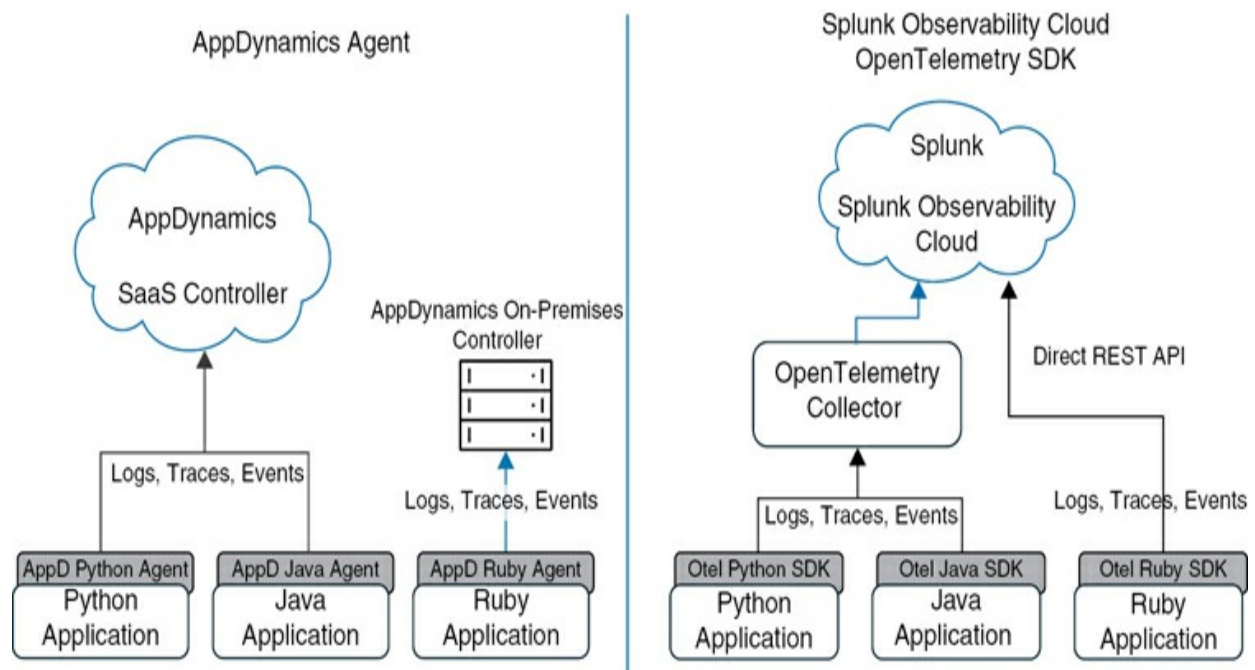


Figure 11-14 Comparison of AppDynamics and Splunk APM Deployments

There are a few points to take note of from [Figure 11-14](#). First, you can see that each application has an OpenTelemetry SDK or AppDynamics agent connected to it. The purpose of these agents or SDKs is to extract telemetry

(logs, traces, events) from the application. The second point to notice is how the data is transmitted. For AppDynamics, the agents are configured with information about the controller to which they should send data, and the data is sent directly from the host application to the controller. For Splunk, the data can be sent to an OpenTelemetry collector, which will preprocess and batch it before sending it to the Splunk Observability Cloud. The data can also be sent directly from the application to the Splunk Observability Cloud via the Splunk REST API.

The preferred approach is to leverage an OpenTelemetry collector to send the data because sending the data directly over the API has more limitations. For example, suppose you want to use Splunk AlwaysOn Profiling to continually collect stack traces and display more granular details about your application's performance. In that case, you must use an OpenTelemetry collector to send this data and not the direct API because the direct API does not support AlwaysOn Profiling.

The real magic happens after APM data is collected and sent to the controller or Observability Cloud. Now, it can be visualized in the controller or Observability Cloud to highlight potential issues; visualize transaction flows and their downstream dependencies; and quickly isolate issues and associate problematic events to their application traces, logs, and even lines of code that may be causing the problem. [Figure 11-15](#) shows how the AppDynamics APM service can automatically build flow maps showing connections between various components of your application and automatically highlighting problematic nodes. Note, however, that Splunk has this same capability. This example shows what the flow maps would look like for AppDynamics because the services are similar.

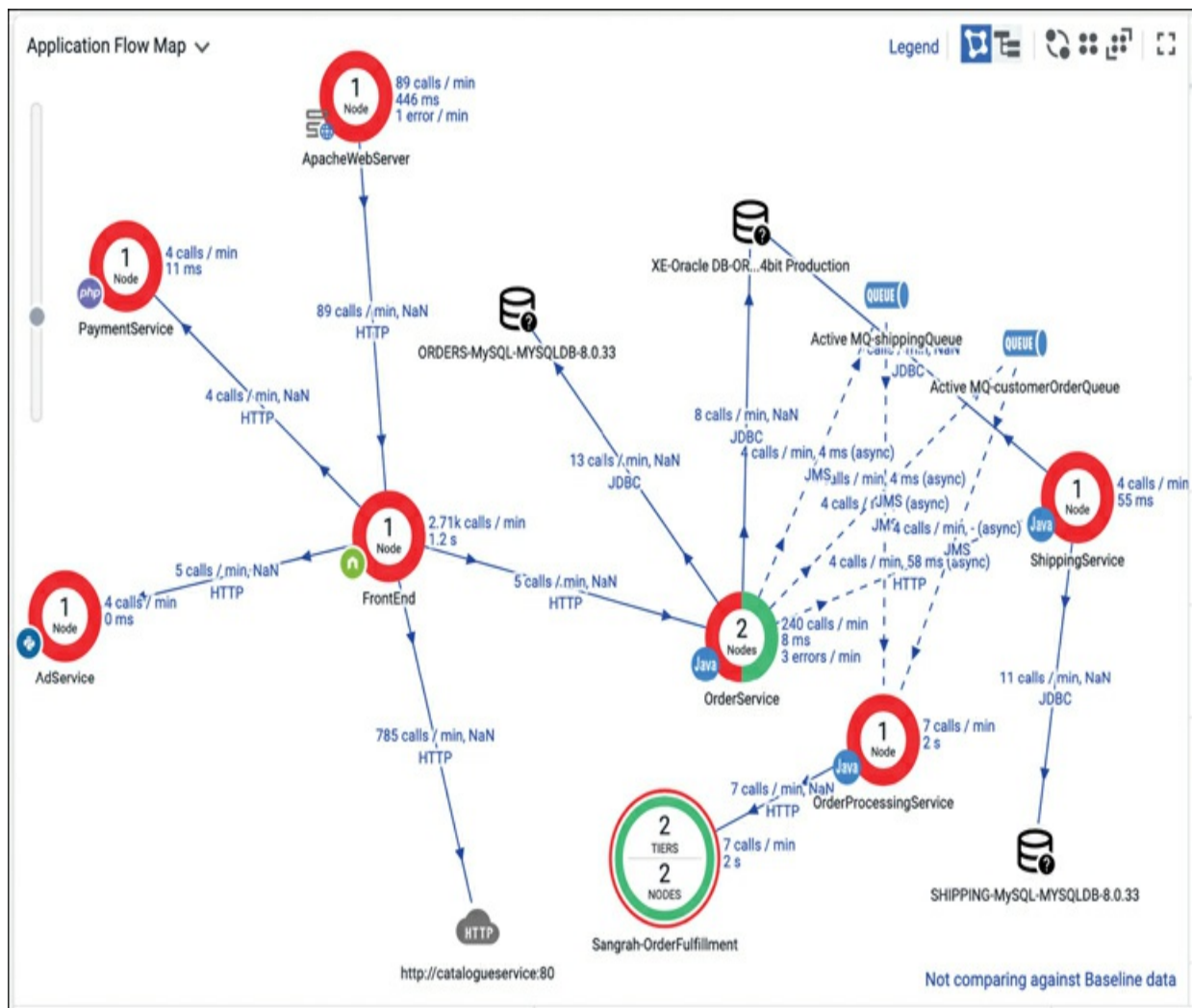


Figure 11-15 AppDynamics APM Flow Map Example

The dynamically generated flow map intelligently combines the data from various applications so that you can visualize transactions between your applications quickly as well as the average delay and errors that may be occurring on those applications. In [Figure 11-15](#), you can see the different types of applications being used—from HTTP endpoints, database connections, queues, and even what language the application is running.

Additionally, you can examine each application's endpoint performance. Endpoint performance can quickly show you the request rate, error rate, latency, HTTP status codes, and so on. [Figure 11-16](#) shows what the endpoint performance dashboard looks like in the Splunk Observability Cloud.

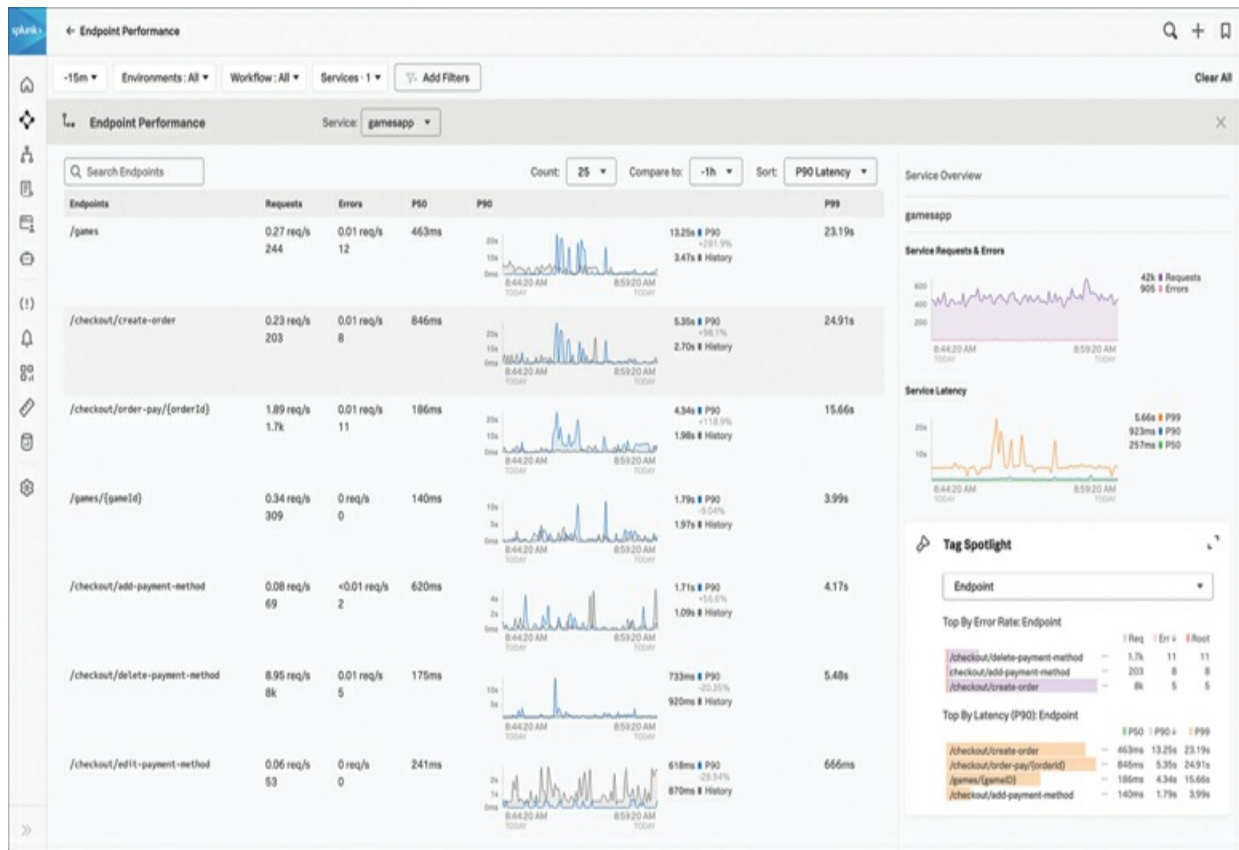


Figure 11-16 APM Endpoint Performance in Splunk Observability Cloud

These are only a few capabilities you get from an application performance monitoring deployment using Splunk or AppDynamics. The main point to remember and understand about an APM service is that it gives you visibility into applications on your network. But to have full-stack observability, you need more than just visibility into your application's performance. You need an understanding of how the infrastructure hosting your application is performing, monitoring end users, and even collecting and correlating logs to various components of your architecture.

Infrastructure Monitoring

Customers' data and applications are now spread across both on-premises and cloud data centers, with the rise of Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) solutions gaining more adoption in the industry. The ability to monitor your infrastructure, regardless of where it resides, is challenging. Without an observability solution like AppDynamics or Splunk,

IT must leverage many tools to assess the state and health of different environments.

Let's consider an example where fake company Acme Co. has data center resources in its on-premises data center and within a cloud provider like AWS. To monitor and view the state of its infrastructure across these sites, Acme would need to manually leverage an AWS portal to view the state of infrastructure in the AWS environment and likely use several different tools to monitor the state and health of its on-premises infrastructure. This disjointed monitoring tactic can lead to unknown outages, teams not collaborating across various disciplines, and slower time to resolution when trying to determine the cause of an outage.

Infrastructure monitoring—a feature supported by Splunk and AppDynamics—enables IT teams to monitor their infrastructure regardless of whether it is in the cloud or on-premises. With this capability, you now have a single pane of glass, giving you insight into your infrastructure's health, errors, and metrics.

Infrastructure monitoring is a way to extend the visibility of your applications' health into the infrastructure hosting your applications. If you have already configured and deployed application monitoring in your environments, adding infrastructure monitoring extends your visibility further down the technology stack. Adding infrastructure monitoring allows you to monitor the hosts, containers, Kubernetes, and cloud instances used across your enterprise.

Splunk and AppDynamics each take a different approach to gathering metrics from various infrastructure components. Splunk, for example, leverages OpenTelemetry collectors and cloud-native integrations with various cloud providers like AWS, GCP, and Azure to collect data. AppDynamics, on the other hand, relies on its network and machine agents to collect this data.

[Figure 11-17](#) shows how AppDynamics leverages three different types of agents to gain visibility from the application to the host infrastructure and network. Each agent collects a key component of metrics needed to have visibility across your application's performance.




This Agent Monitors...		Example Metrics
1	App Agent 	apps app servers JVMs CLR's
2	Network Agent 	network packets TCP connections TCP sockets
3	Machine Agent 	processes services caching swapping paging queueing
	Hardware/OS	disks volumes partitions memory CPU network interfaces
	Server Visibility	Hardware/Software Interrupts Virtual memory/swapping Process faults CPU/disk/memory utilization by process
		CPU busy times Memory utilization Disk reads/writes JVM crashes

Figure 11-17 Agent Overview for Infrastructure Monitoring for AppDynamics

Note

In [Figure 11-17](#), Server Visibility is an extension of machine agents that allows them to collect additional telemetry and unlocks additional capabilities in the AppDynamics controller UI. You can find more details on Server Visibility at <https://docs.appdynamics.com/appd/24.x/latest/en/infrastructure-visibility/server-visibility>.

Splunk's infrastructure monitoring approach comprises cloud-native integration with cloud providers and OpenTelemetry collectors. Unlike

AppDynamics, Splunk has cloud-native integrations with GCP, AWS, and Microsoft Azure. This capability allows Splunk Observability Cloud tenant administrators to authorize their Splunk Observability tenant to collect metrics directly from their cloud provider. This cloud integration is required to collect cloud metadata and display it in the Splunk Observability Cloud.

By connecting your Splunk Observability tenant to the cloud providers you are using, Splunk can now monitor and alert you of incidents that may occur on those platforms, letting you know if the cloud provider is having an issue that may impact your services on that platform.

The second approach to infrastructure monitoring using Splunk Observability Cloud is to leverage OpenTelemetry to capture metrics from the hosts you want to monitor. Splunk Observability Cloud supports Kubernetes, Linux, and Windows integrations to collect infrastructure metrics. This usage is consistent with Splunk's approach to application monitoring using OpenTelemetry collectors. The most significant difference is that the OpenTelemetry deployment monitors the hosts or containers running the applications.

End-User Monitoring

End-user monitoring is all about measuring the performance and health of web and mobile applications, directly correlating with the end user's experience with an application. Understanding how users interact with your application, where they are coming from, how your server responds to user interactions, and what actions or sources are inflicting the most significant load on your application are all critical data points to delivering an excellent user experience.

Several different sources, including web browsers, mobile client applications, and even IoT devices, can be used to collect metrics from end-user interactions with your application.

Browser Monitoring

One of the most common approaches to end-user monitoring is configuring your web application to collect metrics from users' browser sessions while

interacting with your application. This is done by leveraging a JavaScript agent that gets injected into a user's browser when the website is loaded. This JavaScript agent running in the user's browser can collect key metrics about that user's interaction with the web application and send these metrics back to your observability platform. [Figure 11-18](#) illustrates how the JavaScript agent is loaded onto a user's browser when accessing a web application.

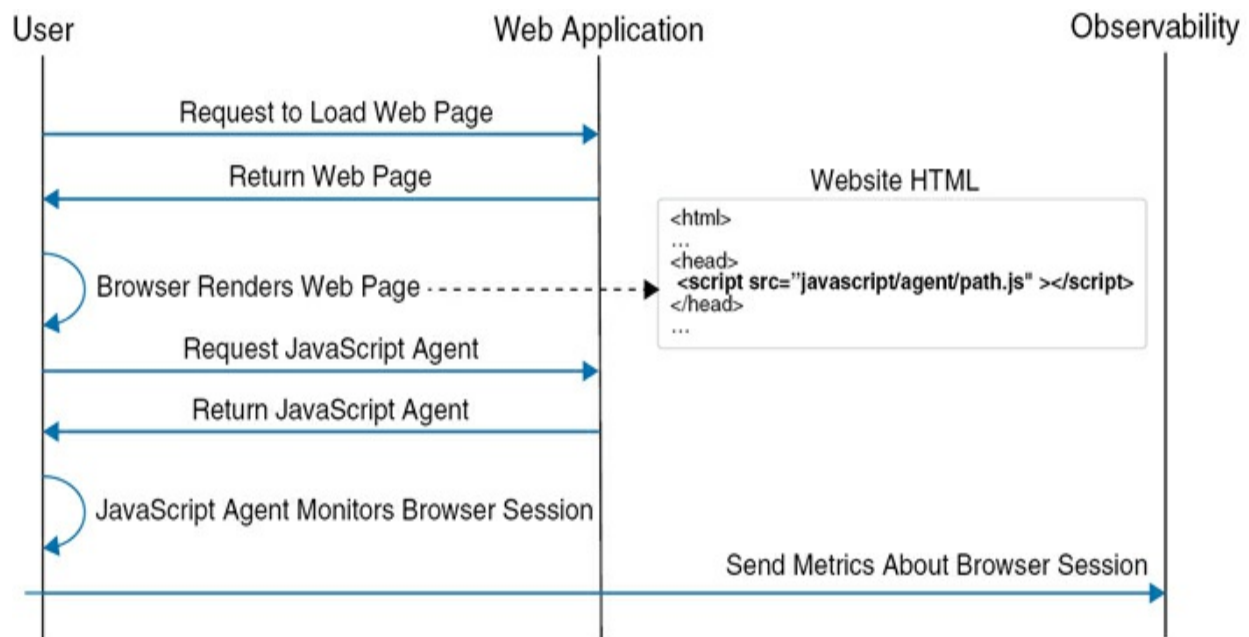


Figure 11-18 Browser Monitoring Process Using JavaScript Agent

[Figure 11-18](#) illustrates the high-level process of injecting the JavaScript agent into a user's browser session. After a browser agent has been configured on a web application, when a user loads the web page on that browser, the web page instructs the browser to fetch the JavaScript agent, which can then run and collect information about that user's browser session on that web page. This JavaScript agent is configured with details about your monitoring backend (Splunk or AppDynamics) and will report metrics and logs back to the monitoring backend, giving you direct insight into the user's experience on the web page. This information includes

- Page load times
- Network request performance
- JavaScript errors

- Browser/client type
- Geographic location of users
- How users navigate through your web page

Suppose you have an application agent running along with the browser agent collecting telemetry from the end user. In that case, you can now correlate events and traces from the browser and application agent to have an end-to-end picture of network request timings, failures, and events.

For example, if you have an application agent running and a browser agent running to collect metrics from users' browsers, and the browser agent detects a high amount of latency when hitting a specific application API, you could also correlate this same browser request to the application to understand what took the application so long to respond to that request. If the application metrics show that the application responded quickly, this would indicate a possible network issue between the client and the application. However, if the application metrics show that it took awhile to process the request, you know there is a performance issue with your application API. All this analysis and insight is done automatically from the observability backend.

Mobile Monitoring

Like browser monitoring, mobile client monitoring is another method of gaining visibility into your user's experience when leveraging a mobile application. Unlike browser monitoring, where a JavaScript agent is injected into the browser session to collect metrics, mobile clients must instrument their application by adding performant code to collect and send telemetry. Splunk and AppDynamics have code libraries that you can import and configure onto your existing mobile application code base.

For example, say that you have an iOS application that you want to configure for mobile client monitoring. If that iOS application is written in Swift, you could install the AppDynamics mobile agent or Splunk OpenTelemetry iOS library in Xcode. Then, you could import that package into your project and configure it to know where to send the telemetry data it collects.

By instrumenting your application with a mobile agent to monitor and collect metrics, you can gain insights into crash data, performance, and health insights about the usage of your mobile application—all instrumented directly into your application.

IoT Monitoring

You might not immediately consider IoT devices a source for end-user monitoring. Still, they fall in the same category as end-user monitoring because they are peripheral devices typically outside the network boundary of your enterprise. IoT devices could range from parking garage sensors to weather stations, smart home devices, and many other things. Collecting metrics from these devices can help better understand these devices' availability, performance, and health.

Like mobile monitoring, IoT monitoring requires you to instrument the software to leverage prebuilt SDKs or REST APIs to send metrics and traces to your monitoring backend. AppDynamics and OpenTelemetry offer a wide range of software-specific SDKs that allow you to add these SDKs to your IoT devices; this way, you can easily instrument the software to collect and send telemetry to your Observability backend. Suppose an SDK is not available for the software your IoT device is running. In that case, there are also direct REST APIs where the telemetry can be sent, allowing you to send the data without using an SDK. [Figure 11-19](#) illustrates how IoT devices can send metrics leveraging either an SDK or REST API.

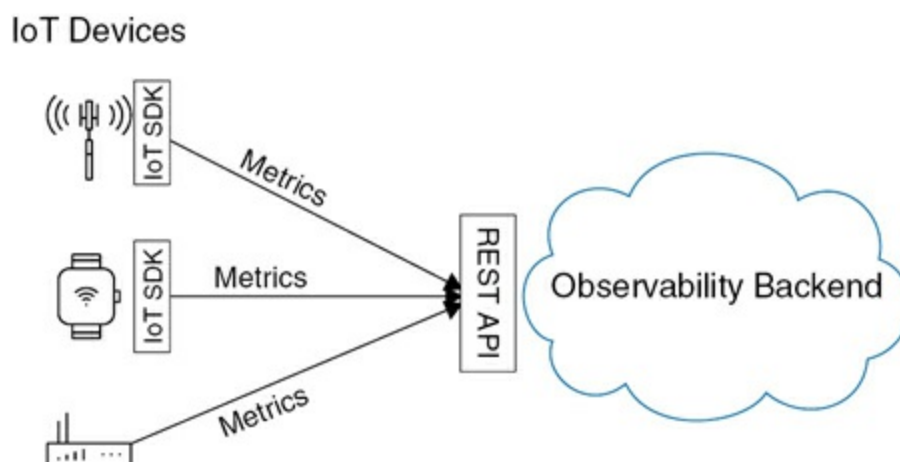


Figure 11-19 IoT Devices Sending Metrics Using SDK or Directly Using

REST API

IoT monitoring gives you clear visibility into latency issues that devices may have connecting to back-end services and allows you to collect application errors or traces to detect and diagnose issues. It also enables you to understand the general use of the device to perform baseline usage analysis to determine when events occur outside of that baseline.

Synthetic Monitoring

Up to this point, you have learned about the most common forms of monitoring, which allow you to gain observability into various locations of your application—from the application directly, from the infrastructure used to host your application, from end users' browsers and mobile clients, and IoT devices. However, there is one more common form of monitoring that allows you to gain visibility into even more of your applications. That is synthetic monitoring.

Synthetic monitoring is a way to generate traffic to test your website's performance artificially. The primary purpose of a synthetic monitoring solution is to help you understand how your site is working over time and to allow you to test all aspects of your site. If you compare a synthetic monitoring solution with browser end-user monitoring, one of the key differences is that end-user monitoring requires real users to access and interact with your site for the data to be collected. Additionally, the data collected from the end user is based on how they interact with the site. On the other hand, synthetic monitoring allows you to automate interactions with your site from a controlled environment. This will enable you to customize how the interactions with your site are done to ensure greater test coverage of your site while simulating what the end user experience would be.

The following are some specific examples of the types of tests you can run with a synthetic monitoring solution:

- **API Performance:** You can configure a synthetic agent to routinely call an API endpoint to test the API's results and performance. With this type of monitoring in place, you will know when your API services are not functioning well or responding.

- **A/B Testing:** A/B testing is a standard test practice that compares the results or performance of two solutions. With a synthetic monitoring solution, you could deploy two different application deployments to compare them against one another. Does one offer better performance than the other?
- **Geographical Impact:** By testing your application from various locations, you can better understand the end users' experience when accessing your site from different locations.
- **Uptime:** Is your website running? Why wait for an end user to tell you it is not responding? Synthetic testing can routinely check to ensure that your service is up and running.
- **User Journey Testing:** You can configure tests to navigate specific paths through your website to ensure it behaves appropriately.
- **Broken Links:** You also can test the links on your web pages to ensure that they still work.

These are just a few of the practical uses of synthetic monitoring. What you choose to monitor may differ depending on the type of application you have deployed and your business-critical workflows. The important point is having a solution that is robust enough to allow you to customize what and how you monitor your application. AppDynamics and Splunk give you those capabilities.

Another essential factor for synthetic monitoring is the capability to test private and public applications. Both Splunk and AppDynamics offer the capability to deploy private synthetic agents in your enterprise that can be used as a location to run a test. You may have some internal APIs or applications that you want to include in your synthetic testing. If these applications are unavailable from the Internet, you must deploy a private agent within your network with access to the necessary applications to perform the testing.

Synthetic monitoring is a powerful tool that can unlock performance and insights about your applications. The ability to monitor and test your application, even when it is not actively used by end users, equips application owners with data to improve the user experience.

Log Observer Connect

One of the capabilities that you have not heard much about regarding observability or monitoring to this point is logging. The logs produced by your application play a critical role in your ability to diagnose and resolve issues related to your application. However, one of the main challenges you face is finding what logs to look at. When a user reports a problem on one of your applications, narrowing down that issue to a finite point in your application or infrastructure is challenging. That is the benefit that comes from observability and monitoring solutions. Now, when a problem is detected with your application, the observability platform gives you insight into the specific location, metrics, and traces related to the issue you are trying to troubleshoot. With this data, you are now equipped with much more context to know what logs you should look at.

That is where Log Observer Connect comes into play. Now, with both Splunk Observability Cloud and AppDynamics, you can view your application logs in context with the insights produced from the observability platform.

This is where the power of using Splunk for logging comes into play. By using Splunk Enterprise or Splunk Cloud for application logging, you get all the benefits of Splunk's logging platform and deep integration between Splunk Observability Cloud or AppDynamics and your logs.

If you talk to IT, SRE, or DevOps teams, you will no doubt hear their frustration over using several different tools to monitor, troubleshoot, and resolve issues with the applications they work with. Having to use multiple tools creates more delay and difficulty when diagnosing problems across multiple systems. Having a single pane of glass that can provide both visibility into their applications and logging context to help with troubleshooting and remediation is a game changer.

For example, let's say you have an application that has recently started showing high latency for a specific business transaction. This issue is automatically highlighted and brought to your attention in the observability platform you are using. As you drill down into that issue, you can see the problem occurring on a specific application node and when a particular API is called. With Log Observer Connect, in both Splunk Observability Cloud or

AppDynamics, you can now jump directly to the logs for that application in Splunk Cloud or Splunk Enterprise, with a dynamically generated SPL query that will show you the logs related to that application node, the timeframe you are looking at, and filtered to the logs for a specific business transaction.

Note

Search Processing Language, or SPL, is the language used in Splunk's logging platform to search for data. SPL was initially based on SQL and UNIX pipeline syntax. It allows for the searching, filtering, modification, filtering, and deletion of data.

Within Splunk Observability Cloud, the logs from your application are viewable in the same interface as the rest of your observability data. With AppDynamics, deep links from the controller UI to Splunk Enterprise or Splunk Cloud are dynamically generated, allowing you to jump to the logs you care about quickly. [Figure 11-20](#) shows the process of navigating to Splunk logs from an AppDynamics controller UI.

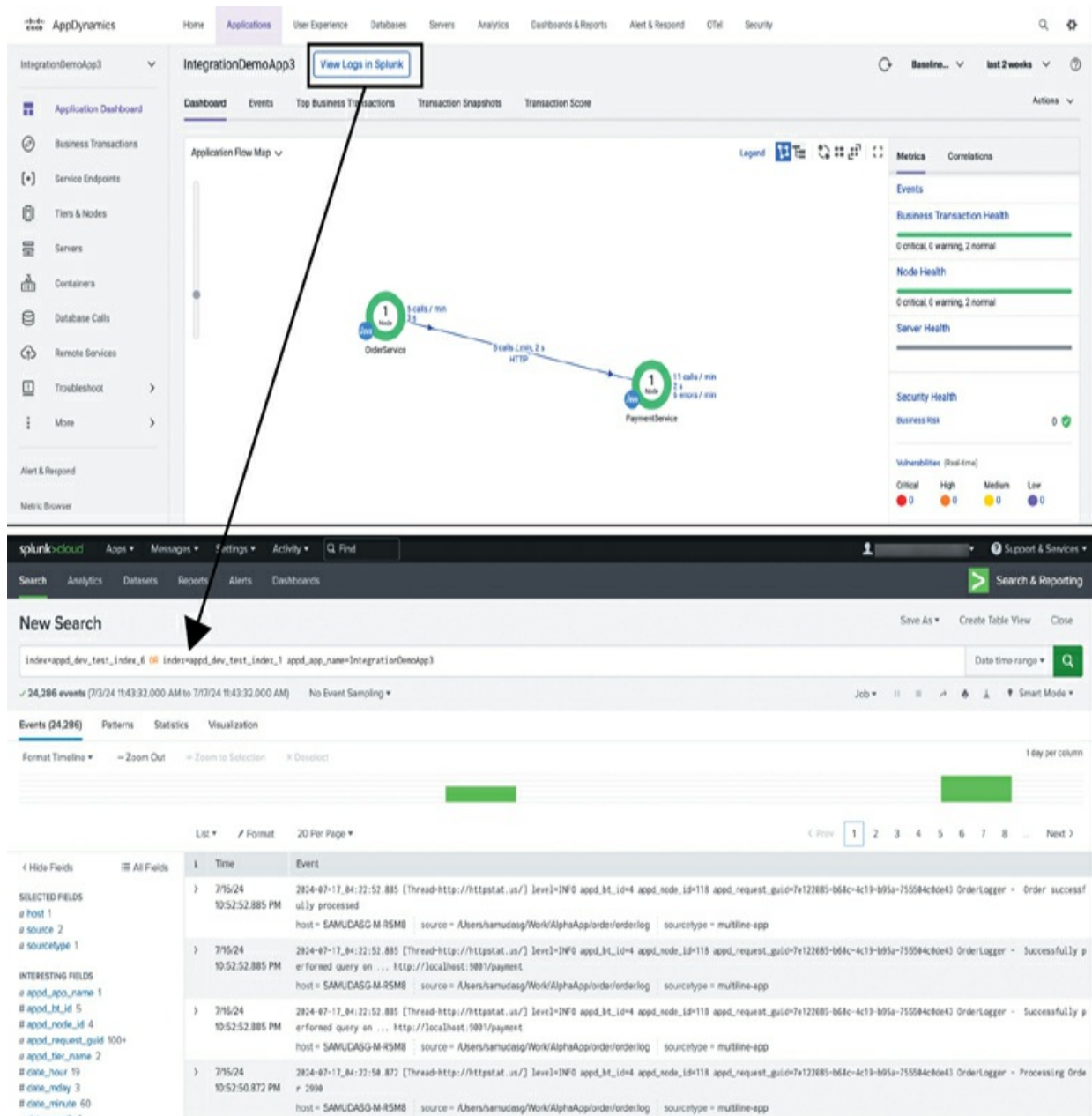


Figure 11-20 Log Observer Connect for AppDynamics

In [Figure 11-20](#), when you are viewing an application or specific application node from an AppDynamics controller, a button shown at the top of the screen will directly take you to the Splunk logs associated with that application you are viewing. As you drill down further into specific tiers or nodes in the AppDynamics controller UI, the deep link will be modified to match your viewing context. Additionally, there is an SSO integration between AppDynamics and Splunk, so you do not have to reauthenticate

Splunk when you click the View Logs in Splunk button. This allows for a more seamless experience.

The Log Observer Connect integration is even deeper for Splunk Observability Cloud. You can display logging directly in the Observability Cloud portal without jumping to another site to view the logs. [Figure 11-21](#) shows what this process looks like in Splunk Observability Cloud.

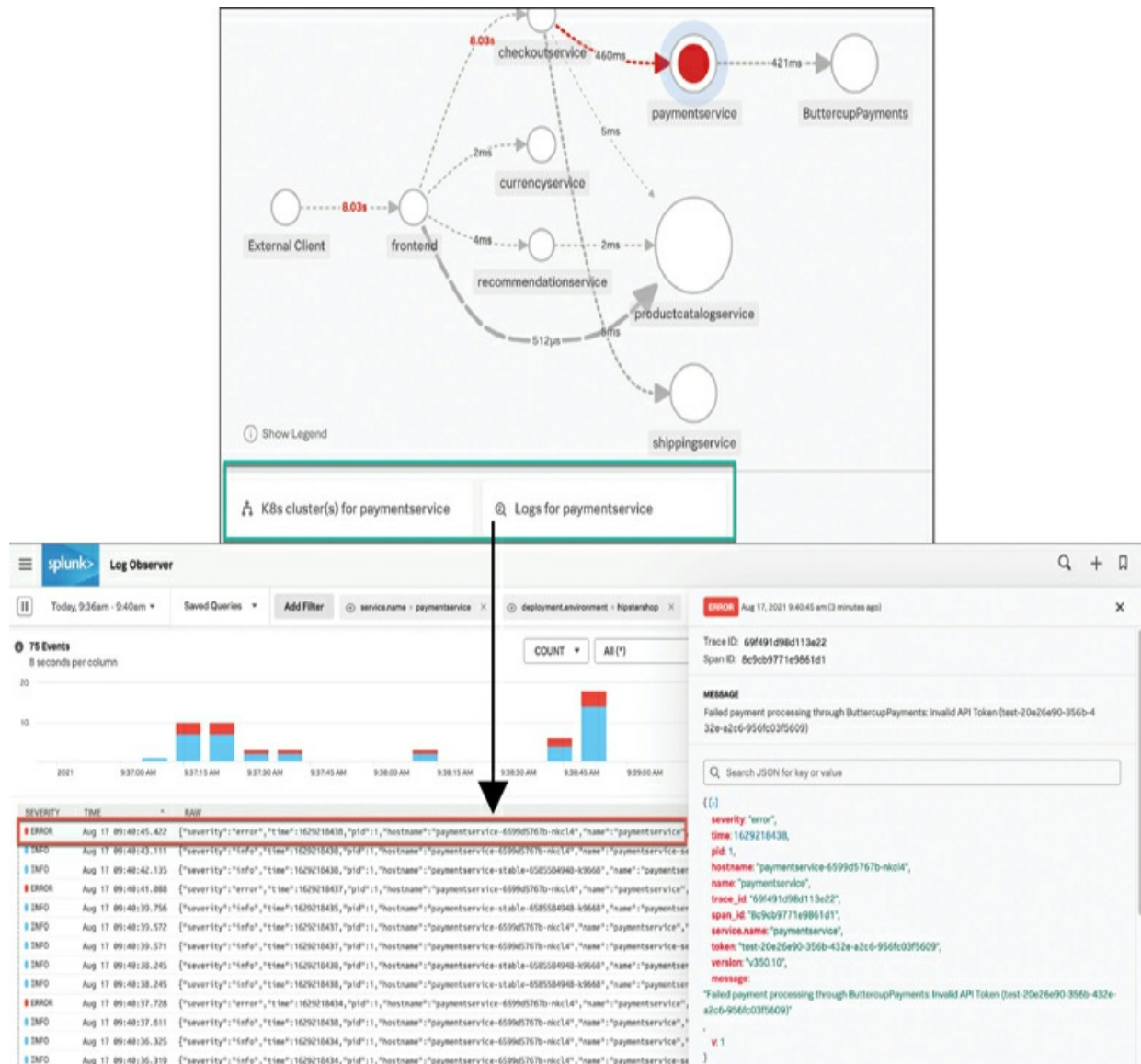


Figure 11-21 Log Observer Connect in Splunk Observability Cloud

In [Figure 11-21](#), the application flow map highlights errors on the paymentservice node. When you click this node, you are given an option to

view the logs for paymentservice. These logs are then pulled up in the Observability Cloud for that specific node and time range you used in the application flow map.

If you have ever had to work through isolating an application issue, wading through various data streams to try to isolate the problem, you can see immediately how effective a solution like this would be at reducing the time needed to resolve an issue.

Summary

Full-stack observability (FSO) is growing in importance for customers. With the rise of more SaaS applications and tools, and applications spread across hybrid networks spanning from on-premises to various cloud service providers, the need to have a consolidated view of your application's health and performance is paramount to delivering a successful business function.

In this chapter, we explored some basic concepts of observability and monitoring, introducing concepts such as MELT and the OpenTelemetry framework and their importance in FSO solutions. We then looked at how Cisco's Observability applications from AppDynamics and Splunk are coming together to deliver a best-in-class observability suite. With Cisco's more recent acquisition of Splunk, these two solutions will become increasingly integrated to take the best of both applications and deliver them seamlessly to customers.

We also covered some fundamental architectural differences between a Splunk and AppDynamics deployment. This discussion set the stage for better understanding how some of the core features of an observability solution works and what they are used for. This discussion covered application performance monitoring, infrastructure monitoring, mobile monitoring, end-user monitoring, and synthetic monitoring. Each of these capabilities delivers critical telemetry from a different part of an application and, when combined, gives you unparalleled visibility into your application's usage, performance, health, and insights.

In the last part of this chapter, we discussed how Log Observer Connect brings together the logging from your application into your Observability

application to combine both the data and telemetry of your applications with your logging. This reduces the number of tools that application administrators must use to discover and remediate application problems.

This chapter was meant to introduce the world of observability within Cisco. However, it was not exhaustive in covering all the features and capabilities offered by AppDynamics and the Splunk Observability Cloud. For a deeper dive into all the capabilities provided by each of these solutions, you should check out the Splunk and AppDynamics web pages.

After reading this chapter, you should have a better understanding of how observability and monitoring solutions work, why they are necessary, and, more specifically, how Cisco delivers these capabilities through Splunk and AppDynamics.

References

- Managing the application experience with Cisco: Full-stack observability, Stefano Gioia and Tjerk Bijlsma:
<https://www.ciscolive.com/c/dam/r/ciscolive/emea/docs/2024/pdf/BRKC2310.pdf>
- Cisco and Splunk coming together to deliver full-stack observability, Ananda Rajagopal and Raja Mukhopadhyay:
<https://www.ciscolive.com/c/dam/r/ciscolive/global-event/docs/2024/pdf/BRKAPP-1512.pdf>
- Splunk Observability Cloud overview:
<https://docs.splunk.com/observability/en/gdi/get-data-in/gdi-guide/infrastructure/integrate-cloud-services.html>
- Deployment architecture diagrams:
<https://docs.appdynamics.com/appd/24.x/latest/en/pdfs/deployment-architecture-diagrams>
- Flow maps:
<https://docs.appdynamics.com/appd/24.x/latest/en/application-monitoring/business-applications/flow-maps>
- Monitor service performance using endpoint performance:

<https://docs.splunk.com/observability/en/apm/apm-scenarios/endpoint-performance.html#apm-scenario-endpoint-performance>

- Overview of infrastructure visibility:
<https://docs.appdynamics.com/appd/24.x/latest/en/infrastructure-visibility/overview-of-infrastructure-visibility>
- Application dashboard:
<https://docs.appdynamics.com/appd/24.x/latest/en/unified-observability-experience-with-splunk/troubleshoot-using-logs/application-dashboard>
- Troubleshoot business transaction failures with Log Observer Connect:
<https://docs.splunk.com/observability/en/logs/LOconnect-scenario.html>
- Splunk AppDynamics SaaS Documentation 25.4:
<https://docs.appdynamics.com/appd/24.x/latest/en>
- Splunk AppDynamics on-premises/on-premises virtual appliance: Learn how to monitor, troubleshoot and optimize the performance of your entire stack: <https://docs.appdynamics.com/appd/onprem/24.x/latest/en>
- Splunk: Observability: <https://docs.splunk.com/observability/en/>
- MELT data: <https://developer.cisco.com/docs/cisco-observability-platform/#!melt-data/melt-data>
- MELT explained: Metrics, events, logs & traces, Austin Chia:
https://www.splunk.com/en_us/blog/learn/melt-metrics-events-logs-traces.html
- Splunk protects: <https://www.appdynamics.com/trust-center/security>
- Cloud Security at Splunk: https://www.splunk.com/en_us/about-splunk/splunk-data-security-and-privacy/cloud-security-at-splunk.html

Chapter 12. Observability and Monitoring: Cisco ThousandEyes

In the preceding chapter, you saw how Splunk and AppDynamics provided observability and monitoring for devices and applications, located both on-premises and in the cloud. However, how do you monitor or “see” your data and traffic in transit as it moves across your entire network, including not only your on-premises links but also the connections on the Internet and in the cloud? For many networks, this is a network visibility gap that is challenging. ThousandEyes addresses this blind spot by providing network intelligence and digital experience monitoring for network connectivity, making it a critical complement to observability solutions like Splunk and AppDynamics.

[Figure 12-1](#) depicts how Splunk, AppDynamics, and ThousandEyes can work together. For hardware and applications in your on-premises network or the cloud, Splunk and AppDynamics agents can be utilized for visibility. ThousandEyes can then be used to monitor the connections between them. Combining these solutions provides a more holistic view of your applications and their performance across the network. Essentially, for a complete observability and monitoring system, you need visibility into the applications, platforms, *and* the network connections tying them all together.

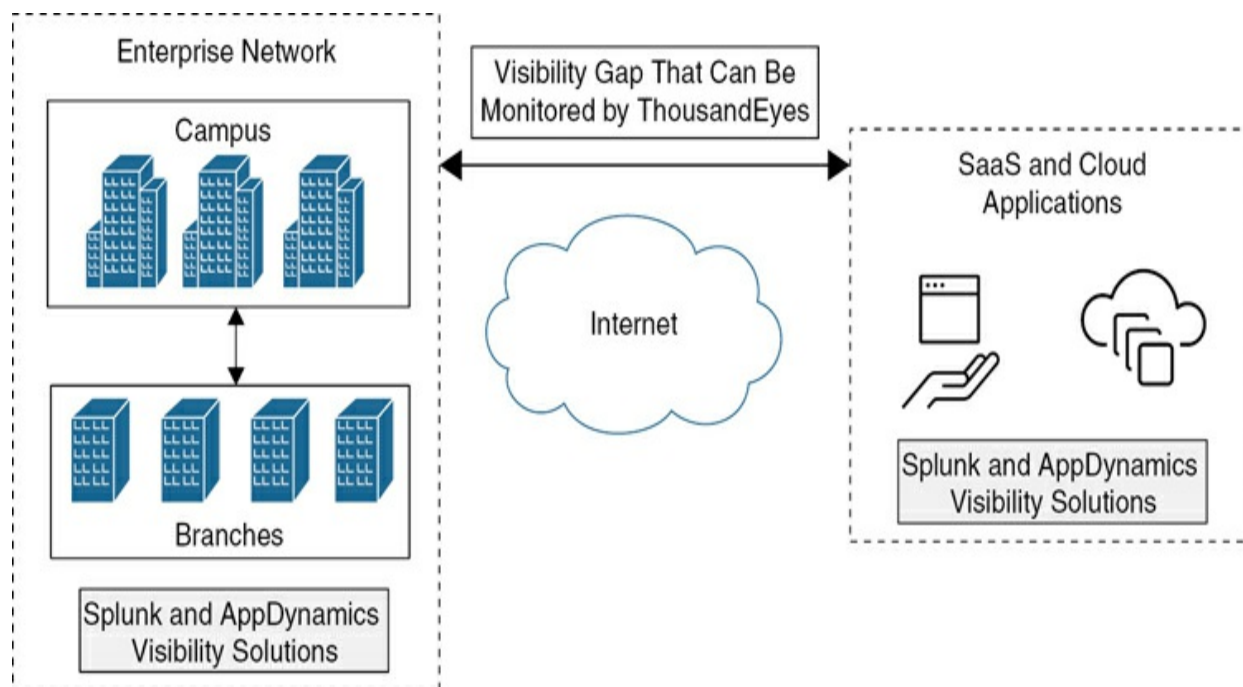


Figure 12-1 ThousandEyes as a Complement to Splunk and AppDynamics Solutions

ThousandEyes is a cloud-based network intelligence tool that provides visibility into the network connections of your infrastructure and is therefore a critical part of any holistic observability and monitoring solution. Applying the concept of agents to the network, the ThousandEyes SaaS solution delivers real-time insights for network performance optimization and the efficient troubleshooting of network-related problems. For example, packet loss or delay could be occurring on routers and switches within your Internet provider's cloud. This issue could negatively impact how users experience your applications and services even if the applications and platforms they reside on are performing as they should. ThousandEyes can help you quickly determine and pinpoint the network segment or device where issues like packet loss or delay are occurring.

In this chapter, we will break down the discussion of ThousandEyes into the following sections:

- **Architectural Overview:** In this section, we'll look at the main building blocks of the ThousandEyes solution as well as some of the underlying software that aligns to the SaaS Architectural Model.

- **Agent Types:** Here, we'll provide details on the different agents utilized by ThousandEyes for data collection, including endpoint agents, enterprise agents, and cloud agents.
- **Agent Tests:** In this section, we'll cover the various tests supported by enterprise and cloud agents and endpoint agents.
- **Path Visualization and Dashboard Snapshots:** Next, we'll highlight two of the more compelling and useful features of ThousandEyes.
- **Internet, WAN, and Cloud Insights:** In this section, we'll provide an overview of these tools and dive deeper into Internet Insights and how it supplements the information provided by agent tests.
- **Integrations:** Finally, we'll provide information on how ThousandEyes can connect with other applications using prebuilt and custom integration methods.

Architectural Overview

The ThousandEyes solution utilizes a network of agents, including cloud agents, enterprise agents, and endpoint agents, to monitor and manage performance from various vantage points. Endpoint and enterprise agents are deployed by users on devices and private networks, whereas cloud agents are deployed and managed by ThousandEyes at important points across the global Internet.

Leveraging these agents, ThousandEyes collects monitoring data to detect real-time Internet outages and application experience problems. It provides cross-correlated visibility in a single view, allowing users to see all layers of service delivery, from synthetic transactions to network paths and global Internet routing feeds. [Figure 12-2](#) provides a high-level overview of the ThousandEyes solution.

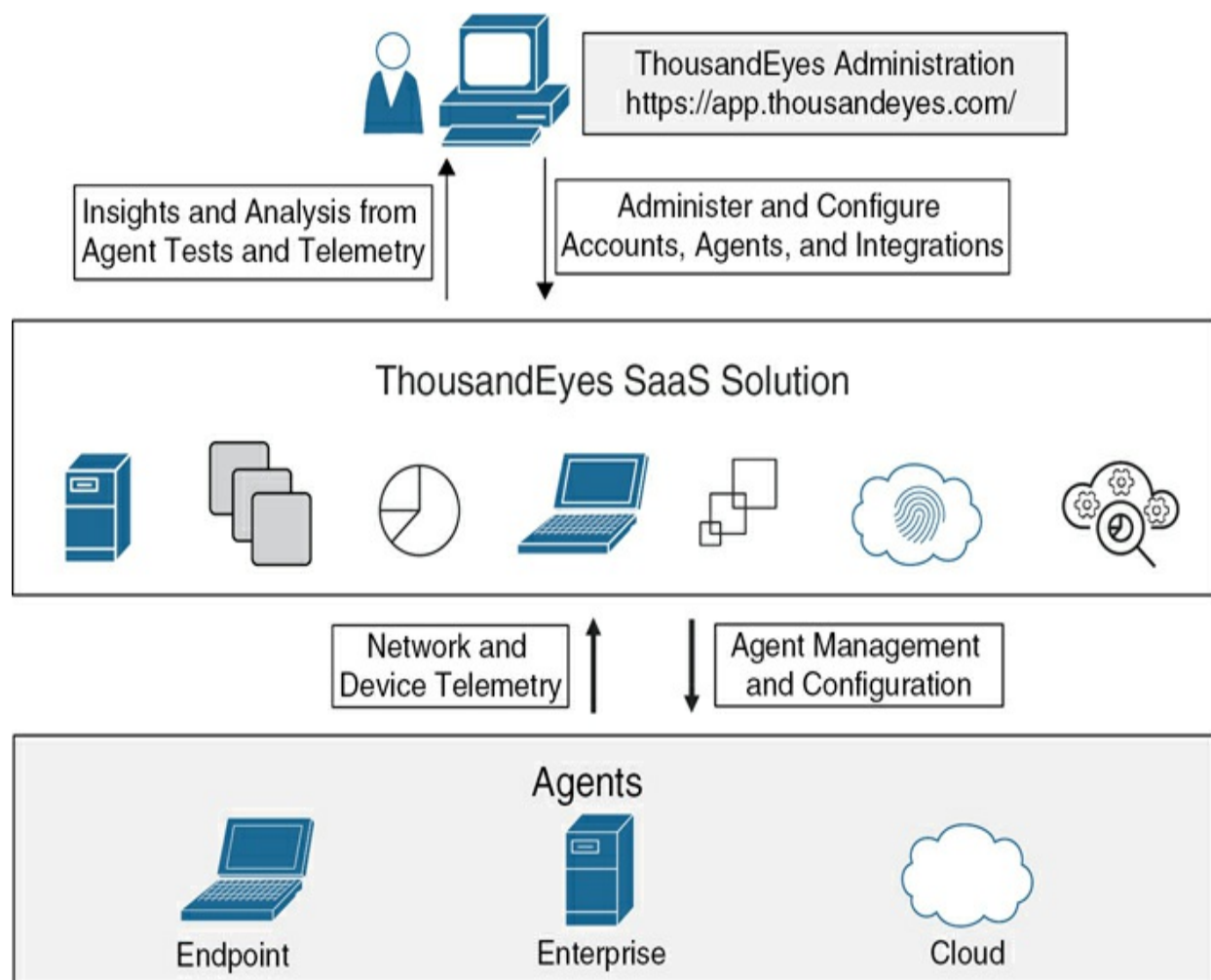


Figure 12-2 ThousandEyes Solution Overview

Being a SaaS solution, ThousandEyes utilizes a subscription model to deliver services from the cloud. Subsequently, access to and administration of ThousandEyes services is provided through a web portal, as shown at the top of [Figure 12-2](#). From this web portal, users can log in and manage and configure their ThousandEyes account, along with all the agents associated with that account. At the same time, ThousandEyes surfaces insights and analysis to the user through the web portal based on the data that has been collected from agents, integrations, and other sources.

At the bottom of [Figure 12-2](#), you can see that ThousandEyes connects with various types of agents. Configuration and management data is pushed down from ThousandEyes to these agents. For example, a user on the ThousandEyes web portal configures a test for an agent. ThousandEyes

converts this user-configured test to instructions for the agent to execute. As agents perform testing and monitoring, they stream this data back to ThousandEyes for analysis. After this analysis, insights and dashboards are displayed back to the user via the ThousandEyes web portal. In the next section, we'll take a deeper dive into the different agent types.

While the exact end-to-end architecture is not publicly disclosed, a few components can be shared to depict how the ThousandEyes solution aligns to the SaaS architectural model presented previously in [Figure 2-8](#) in [Chapter 2](#), “[SaaS Architecture](#).” If you recall, the SaaS architectural model is derived from the NIST cloud computing reference model but makes it a more approachable format with a SaaS focus. In [Figure 12-3](#), some ThousandEyes architectural components are aligned to the SaaS architectural model.

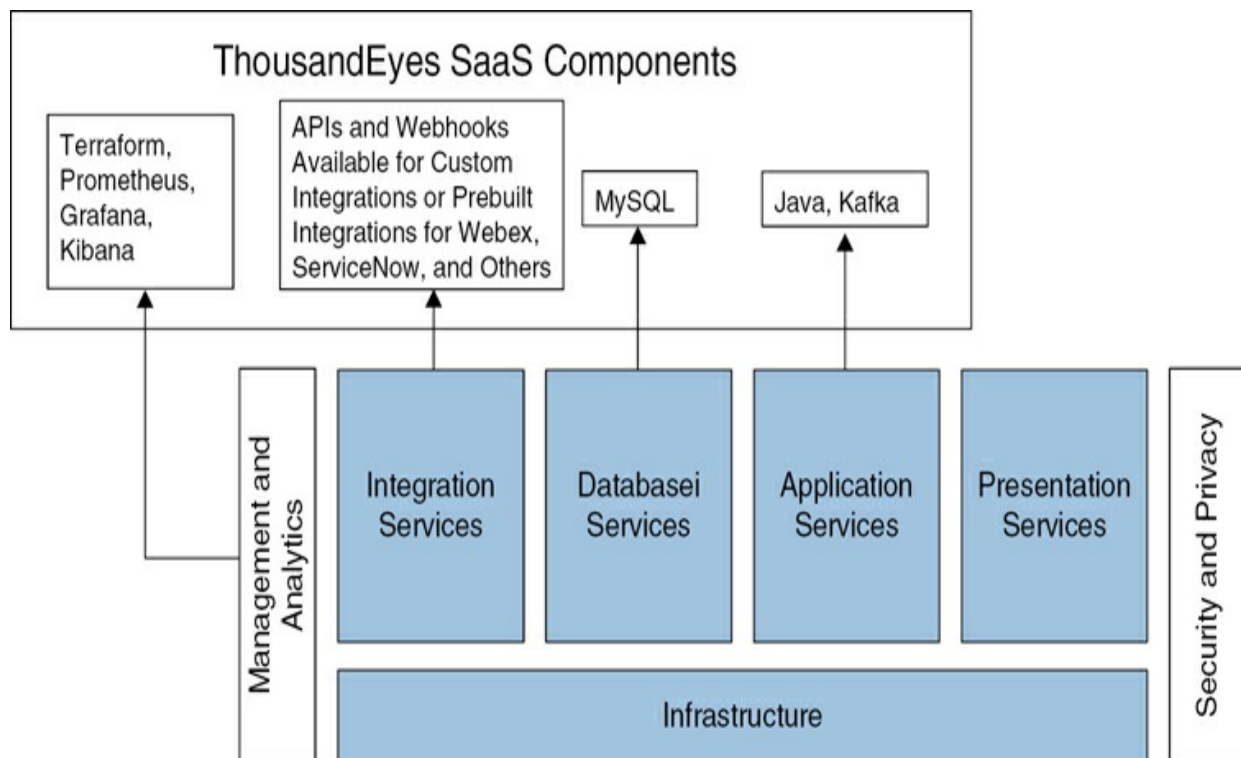


Figure 12-3 Aligning ThousandEyes Component to SaaS Architectural Model

As shown in [Figure 12-3](#), ThousandEyes utilizes Terraform to manage its cloud infrastructure. We introduced and discussed Terraform previously in the “[Management and Analytics](#)” section in [Chapter 2](#). With Terraform, ThousandEyes can define its infrastructure declaratively as a code. This

capability allows for the automatic configuring and provisioning of ThousandEyes cloud resources at scale.

Other software used for management and analytics by ThousandEyes, such as Prometheus, Grafana, and Kibana, is focused on the monitoring and observability of the SaaS infrastructure. Prometheus is an open-source monitoring and alerting toolkit that records all its metrics as time series data. This multidimensional data model works well for highly dynamic SaaS and microservices-driven architectures. ThousandEyes then uses Grafana to visualize Prometheus data in dashboards. ThousandEyes developers use Kibana for troubleshooting and debugging. Services provide logging statements that provide information on program behavior, along with errors and warnings. This information can be easily searched using Kibana.

From an integration services perspective, ThousandEyes has quite a few options. In the “[Integration Services](#)” section in [Chapter 2](#), we provided an overview of custom and prebuilt integrations for SaaS. ThousandEyes supports both types of integrations. If you want to customize, ThousandEyes offers both APIs and webhooks. Numerous prebuilt integrations are available as well, including ones for Webex, Open Telemetry, ServiceNow, and many others. For more details on integrating other applications with ThousandEyes, refer to the “[Integrations](#)” section, later in this chapter.

One database type used by ThousandEyes is MySQL. We covered SQL databases in more depth in the “[Relational and Nonrelational Database Types](#)” section in [Chapter 2](#). This relational or structured database is used to store ThousandEyes account, organization, and test identifiers. With this dataset, ThousandEyes can easily determine which organizations run which tests, for example.

For application services in the SaaS architectural model, two applications used by ThousandEyes are Java and Kafka. Leveraging the Spring Boot framework, Java is a language used for coding services that run in the ThousandEyes backend. Kafka is an event bus application used for communications between microservices. We previously introduced both Kafka and event buses and Java as a back-end programming language in the “[Application Services](#)” section in [Chapter 2](#).

Agent Types

Like the AppDynamics agents discussed in the preceding chapter, ThousandEyes also utilizes agents for data collection. However, ThousandEyes agents are focused on network traffic versus application performance monitoring. Depending on the capabilities needed and location in the network, three primary types of ThousandEyes agents are available: endpoint agents, enterprise agents, and cloud agents. [Figure 12-4](#) provides an overview of these agent types.

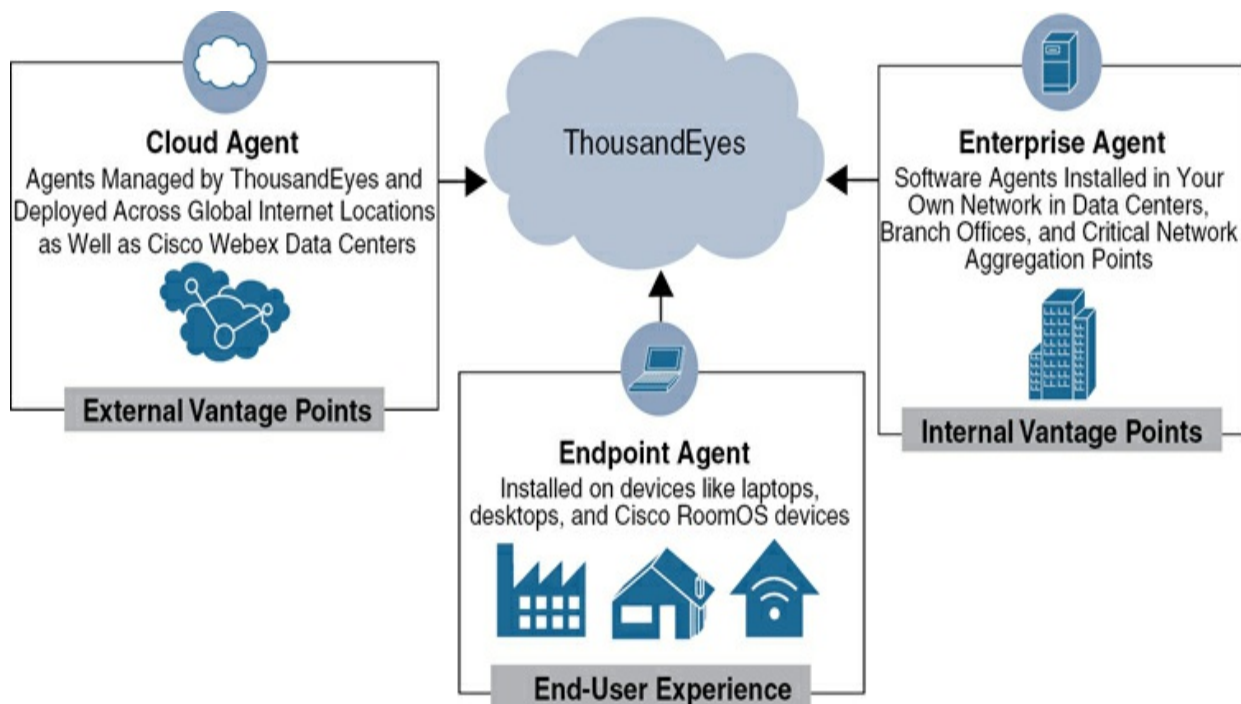


Figure 12-4 Overview of ThousandEyes Agent Types

Although these agents do have some overlapping functionality, they also have some key differences that make them better or even specialized for handling certain use cases. In the following sections, we will cover each agent type individually and highlight the agent functions and unique capabilities to help you better understand when to utilize each agent type.

Note

Cloud, enterprise, and endpoint agents are the main types of agents that you will see most often, but there are others, such as mobile

endpoint agents and device agents. These other agents will often have the same sort of capabilities as one of the main agent types, but there are also some key differences. We will provide more details on mobile endpoint agents and device agents later in this chapter. If you understand the capabilities and functions of the main agent types, it is relatively straightforward to familiarize yourself with other agent types.

Endpoint Agent

The endpoint agent serves two primary functions. First, it conducts tests similar to cloud and enterprise agents, though the tests are a subset of those available to these other agent types. Second, it monitors actual user traffic, deriving user-experience statistics from this data. This traffic includes web and network data, such as Wi-Fi, DNS, gateways, and VPNs. This dual functionality provides a comprehensive view of the user's digital experience. We will cover specific endpoint agent tests that enable the capture of this later in this chapter.

For certain applications, like Webex, preconfigured templates are available for endpoint agents to make setup easier. Tests on endpoint agents can be scheduled or be dynamic and run when certain applications are used. Because ThousandEyes is a SaaS solution, management and configuration of endpoint agents tests for a user or group of users are managed through the cloud-based ThousandEyes web portal.

Endpoint agents are installed on end-user devices like laptops and desktops running Windows or macOS, and then a supported browser, such as Google Chrome, is leveraged. Because they provide direct visibility from the user's device, you can monitor the user's experience from the initial connection leg. This capability is helpful in diagnosing first-hop issues, often happening over a Wi-Fi link. These agents are also part of the RoomOS software running on Webex board, desk, and room series devices but must be enabled through the Webex Control Hub management portal. Webex Control Hub is covered in detail in the [“Management and Analytics”](#) section in [Chapter 5](#), [“Collaboration: Webex Meetings and Messaging.”](#)

Deployment strategies for ThousandEyes endpoint agents vary depending on

customer needs and use cases. In campus environments, it is effective to install the agent on a representative sample of devices because the agent operates only when the host machine is active. For small office/home office (SOHO) settings, deploying the endpoint agent on all devices helps capture transient issues that may affect individual users. Similarly, for nomadic users, installing the agent on all devices is ideal to ensure comprehensive coverage, although deploying on a sample of devices may suffice for detecting broader network issues.

For mobile users, the ThousandEyes mobile endpoint agent simplifies monitoring smartphones by providing enhanced visibility. While it shares some overlapping functionality with classic endpoint agents, the mobile endpoint agent offers additional capabilities focused on cellular and Wi-Fi performance for each mobile device. Given the variety of endpoint agent types available, strategic placement is essential to effectively detect both transient and systemic network problems.

Endpoint agents provide you the granularity to observe performance and experience all the way to the user level. This capability is critical because it enables full visibility and monitoring of performance issues effectively by helping you understand both the connectivity and user experience perspectives at the end device.

Note

The term *device agents* is common when discussing ThousandEyes, but it can have a different meaning depending on the context. It may mean any agent installed on a device in a general sense, or more narrowly, it can mean the router agents and lightweight agents installed on small hardware platforms like Raspberry Pi or x86 devices. These router agents and lightweight agents are often found on consumer home routers as part of a service provider's deployment.

Enterprise Agents

Enterprise agents are test points that you deploy in your own infrastructure to monitor network performance and user experience. Depending on what you

need to monitor, you can locate enterprise agents at critical junctions in your network core or on the edge. These agents are more robust than endpoint agents in their capabilities and versatility. For example, enterprise agents have access to a larger suite of network-related tests. Also, while endpoint agents can only initiate tests for monitoring network connections, enterprise agents can both initiate and terminate tests from other Thousand Eyes agents.

ThousandEyes offers enterprise agents in various form factors or options to accommodate diverse customer preferences and installation. Described in [Table 12-1](#), these installation options include OVA/OVM, Cisco devices, physical appliances, Linux packages, Docker, and cloud templates. When you're adding an enterprise agent from the web portal, these various options will be accessible. You should note that unlike endpoint agents, enterprise agents do not provide support for Windows or MAC operating systems.

Table 12-1 Installation Options for Enterprise Agents

Installation Option	Description
OVA/OVM	An OVA/OVM installation provides a prebuilt enterprise agent for Open Virtual Appliance (OVA) or Open Virtual Machine (OVM) environments, enabling deployment on virtual infrastructure platforms such as VMware, Microsoft Hyper-V, and Oracle VirtualBox.
Cisco Devices	Cisco devices support enterprise agent deployment as a Docker container on select Cisco switching and routing platforms, including Catalyst, Nexus, ASR, and ISR series products.
Physical Appliances	A physical appliance installation provides a turnkey enterprise agent solution for off-the-shelf hardware, such as Intel NUC and Raspberry Pi, requiring only power and a network connection for provisioning.
Linux Package	The Linux package installation enables the enterprise agent to be installed on supported Linux software distributions, such as Red Hat, CentOS, and Ubuntu.
Docker	Docker installation enables the enterprise agent to run as a container in supported 64-bit Linux environments, including Ubuntu, Red Hat, Debian, CentOS, and Fedora.
Cloud Templates	Cloud template installation enables the enterprise agent to be deployed natively from the ThousandEyes web portal in supported Infrastructure-as-a-Service providers, such as Microsoft Azure.

Note

For the latest versions of supported software and hardware for ThousandEyes enterprise agents, refer to the Enterprise Agent Systems Requirements documentation on <https://thousandeyes.com>.

When deploying enterprise agents in your network, prioritize placing them near user communities, critical network junctions, and distribution points to optimize performance monitoring and enable targeted troubleshooting. Agents can be deployed in various branch locations, including offices, retail environments, and manufacturing facilities. Additionally, they can be placed near application hosts to provide a clear vantage point free from network

noise and to support bidirectional testing, which is beneficial in identifying issues in asymmetric network paths. Locating agents at key traffic aggregation and routing or switching points is also helpful for quickly narrowing down any issues. This strategic placement of enterprise agents allows for effective monitoring and troubleshooting of network and application performance.

One interesting capability of enterprise agents is clustering. With clustering, multiple agents can be grouped together as a single entity that functions as a single resource for executing network tests. By clustering agents, customers can enhance test capacity, simplify agent management, and distribute test loads more effectively. Clusters inherit configurations from the initial agent and can be expanded with additional agents, redistributing tests as needed.

Cloud Agents

Cloud agents offer a global perspective on network performance, with deployments in dozens of countries and hundreds of locations around the world. These agents are installed and managed by ThousandEyes, providing customers with comprehensive and reliable monitoring coverage. You do not need to administer these agents but can use them for testing whenever you need them. Also, you do not have to worry about scalability because cloud agent clusters adjust according to overall demand. Cloud agents are a shared resource that can be utilized by any ThousandEyes customer.

ThousandEyes focuses not only on geographic distribution but also on network categorization when positioning cloud agents on the Internet. Cloud agents are strategically positioned across different ISP tiers, broadband networks, and mobile operators to simulate a consumer's view of an application or service. Additionally, agents are placed within major cloud infrastructure providers like AWS, Azure, GCP, and Alibaba, with careful consideration of regions and availability zones to enhance monitoring capabilities.

Note

If you utilize Cisco Webex, you can realize an additional benefit from ThousandEyes cloud agents. Within every Webex data center

worldwide, a cloud agent has also been installed, and this Webex cloud agent can be easily used for tests from your enterprise and endpoint agents. This Webex cloud agent deployment enables customers to gain insights into Webex’s internal operations and better identify issues in asymmetric return paths. Furthermore, cloud agents facilitate RTP tests to the Webex infrastructure for testing and troubleshooting audio and video media problems.

Compared to the “inside-out” approach of enterprise agents, cloud agents primarily serve for “outside-in” monitoring, focusing on applications hosted by customers and accessed by external end users. This includes public-facing apps as well as SaaS applications consumed by enterprises. Additionally, cloud agents are instrumental in inter and intra-cloud monitoring, enabling customers to track performance and availability across different regions, availability zones, or even across different cloud providers’ infrastructures. This versatility allows customers to maintain a robust understanding of their digital experience and service performance from various external vantage points.

Cloud agents, enterprise agents, and endpoint agents are foundational blocks of the overall ThousandEyes solution and are summarized in [Table 12-2](#). Understanding agent capabilities and use cases is critical before you dive into the specific agent tests that each can offer. We’ll take a closer look at agent tests in the next section.

Table 12-2 ThousandEyes Agent Types

Agent Type	Managed By	Function	Host Platform
Endpoint	End users and admins	<ul style="list-style-type: none"> Provides insight into device hardware utilization, local network performance, and the user experience of Internet applications Monitors collaboration applications to detect any issues that might impact a user's experience during a meeting 	User devices, Windows, and macOS supported and some Cisco RoomOS devices
Enterprise	Admins	<ul style="list-style-type: none"> Located within your infrastructure, including data centers, branches, and important network junctions Provides internal or SaaS application monitoring and visibility from the network perspective 	Linux-based with various integration options (e.g., Docker, IaaS, Cisco networking devices)
Cloud	ThousandEyes	<ul style="list-style-type: none"> Strategically placed globally on the Internet as test points for monitoring traffic through ISPs and CSPs Installed and maintained by ThousandEyes 	ThousandEyes servers

Agent Tests

ThousandEyes provides a variety of synthetic tests that customers can select to monitor specific applications or assets. These tests simulate real user traffic, ensuring privacy by not capturing or analyzing actual network packets. This is an important distinction compared to other solutions in this space. Customers define these tests through the ThousandEyes platform, which then distributes the test parameters to the selected agents. The agents execute the tests according to the set frequency, collect the results, and send the data back to the ThousandEyes platform.

Upon receiving the results, ThousandEyes processes the data, correlates it with different test components, and provides real-time access to the

information via the ThousandEyes web app and API. The processed data also integrates into the ThousandEyes alert and notification system, enhancing the monitoring and alerting capabilities for the customers. This approach allows customers to maintain a comprehensive view of their network's performance while ensuring user privacy.

In the following subsections, “[Enterprise and Cloud Agent Tests](#)” and “[Endpoint Agents Tests](#),” we will provide a deeper discussion about the tests that are supported. Both enterprise and cloud agents are dedicated applications that are typically deployed to always be available. Their test capabilities are the same, and that is why it makes sense to talk about them together. Endpoint agents are sometimes offline, and their tests are a little different because of the end-user focus. Therefore, we will discuss endpoint agent tests in a separate subsection.

Enterprise and Cloud Agent Tests

Enterprise and cloud agents are dedicated test points in your network and on the global Internet. They have a robust test suite to cover several different use cases with many customization opportunities as well. [Figure 12-5](#) provides an overview of the types of tests supported by enterprise and cloud agents.

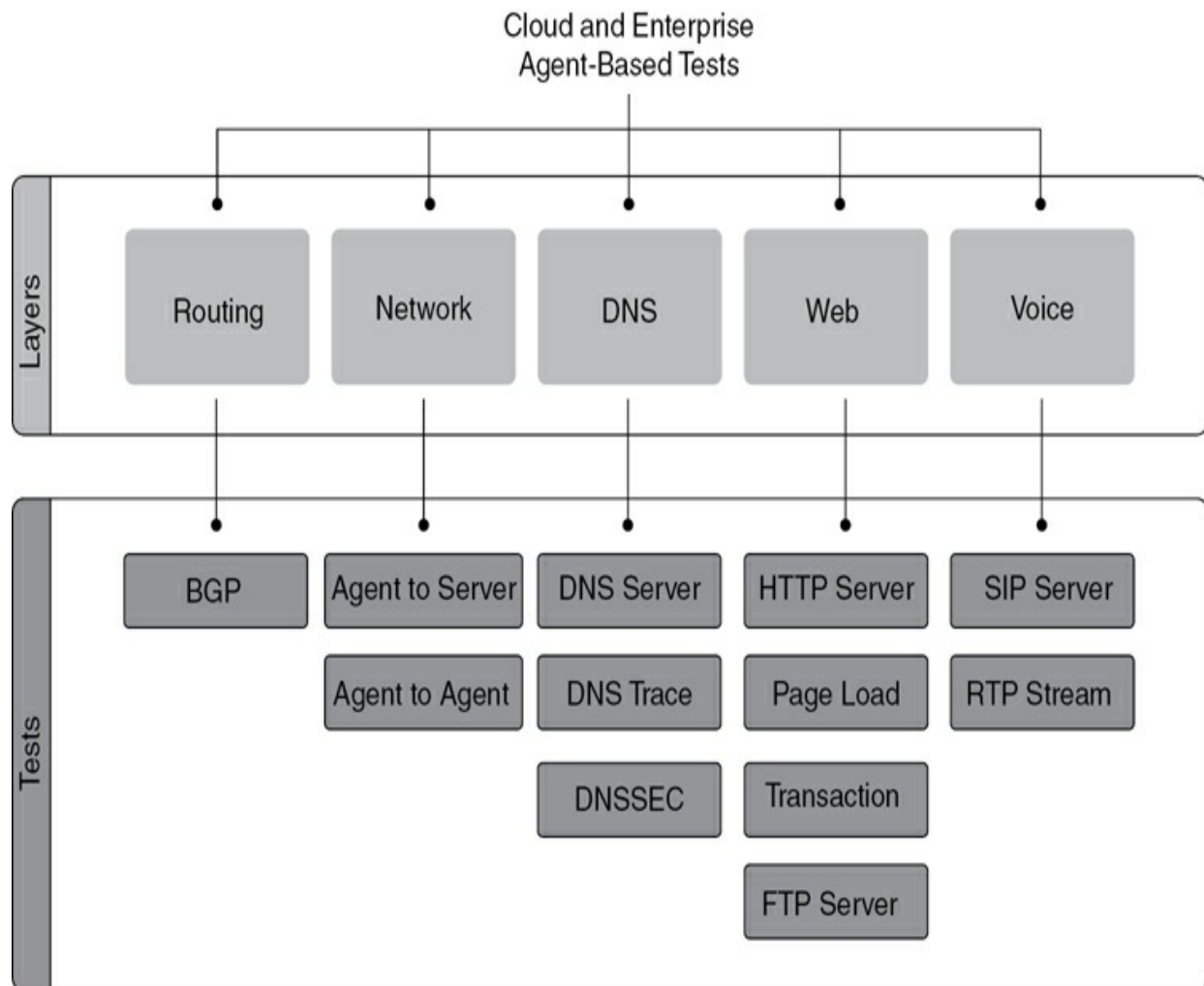


Figure 12-5 Test Options for Enterprise and Cloud Agents

Across the top of [Figure 12-5](#), you can see how ThousandEyes segments its tests into layers. Each of these layers then has one or more tests associated with it that you can configure. In the next few sections, we will provide more details about each layer and its associated test(s).

Routing Test

Routing layer tests provide methods for collecting Internet routing-related information. These tests can measure metrics like routing path changes, reachability, and BGP updates. The available routing test is for Border Gateway Protocol (BGP). As the routing protocol of the Internet backbone, BGP is critical to how all traffic moves across the Internet. The BGP test covers various use cases, including network prefix reachability, detecting and

alerting on route leaks or route hijacking, monitoring upstream network providers, and alerting on unexpected path changes.

To ensure its accuracy when generating its insights and visualizations, the ThousandEyes BGP test collects BGP routing data from both public and private BGP monitors. Public BGP monitors include ones that are deployed by ThousandEyes as well as routing data provided by the University of Oregon's RouteView project and the Routing Information Service (RIS) provided by Réseaux IP Européens (RIPE). All this publicly monitored data is combined with data from private BGP monitors. Private BGP monitors are ones that you have configured to report BGP data directly from your routers to ThousandEyes.

Figure 12-6 shows a BGP test result as seen within the ThousandEyes BGP Route Visualization display. Specifically, it shows a visual rendering of a BGP path change detected by the St. Petersburg public BGP Monitor, where the network path through ASN20485 is replaced by a path through ASN9002.

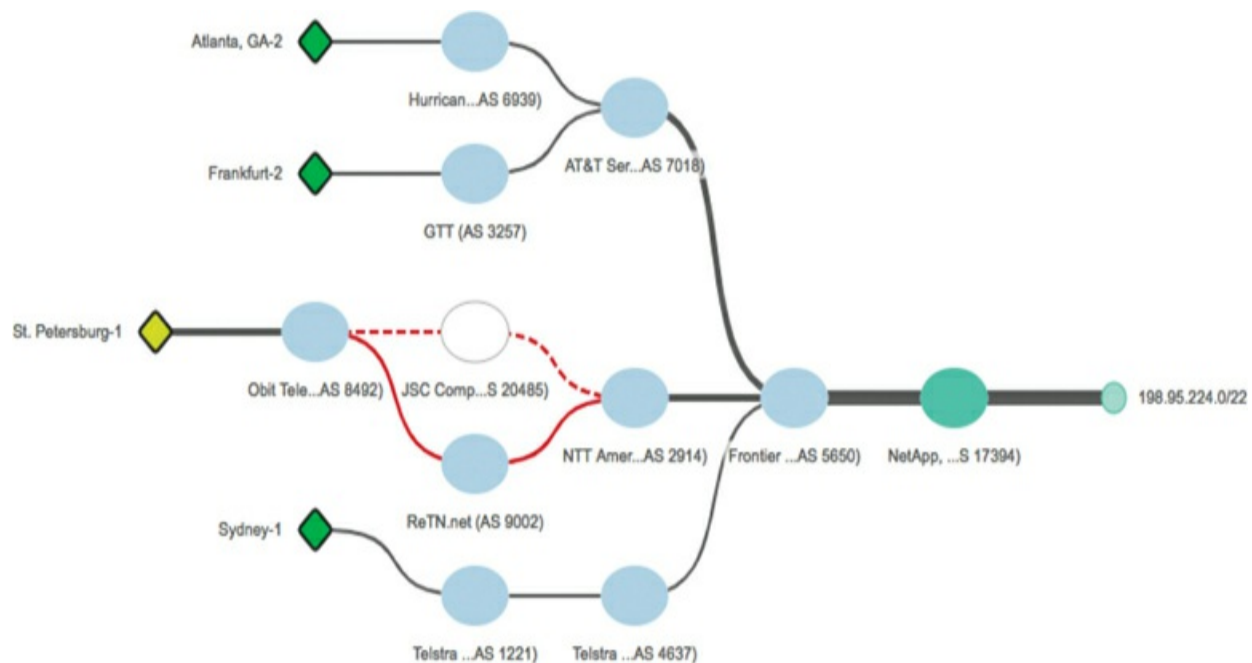


Figure 12-6 BGP Test Results Showing a Change in the Network Path

Network Tests

Network layer tests are aimed at core protocols in the IP stack, such as TCP, UDP, and ICMP. Two options are available when setting up network tests:

Agent to Server and Agent to Agent. The Agent-to-Server test can target any IP node using TCP or ICMP, whereas the Agent-to-Agent test adds support for UDP and allows bidirectional testing, enabling both agents to originate test data. This capability is beneficial for identifying issues on the reverse path due to the Internet's asymmetry.

Use cases for both the Agent-to-Server and Agent-to-Agent test are numerous. [Table 12-3](#) lists some of the more well-known ones, but you should realize that there is a quite a bit of flexibility and customization when configuring network layer tests.

Table 12-3 Network Test Use Cases

Network Test	Common Use Cases
Agent to Server	<ul style="list-style-type: none">• Measuring network performance to a remote server• Understanding network path changes between source to destination• Verifying the availability of a remote target• Identifying degradation along a network path• Confirming network handling of DSCP and MTU parameters• Monitoring ingress traffic load distribution across ISPs
Agent to Agent	<ul style="list-style-type: none">• Measuring bidirectional network performance and throughput• Measuring network connectivity between data centers that are on-premises or in the cloud, as well as regional branch offices connecting to on-premises or cloud data centers• Determining branch office to HQ network quality via VPN• Detecting packet drops and latency on a network path• Evaluating the return path by performing bidirectional testing• Comparing end-to-end network performance and characteristics between the forwarding and return paths• Evaluating network connections for specific applications

Note

One network test case that is particularly useful is monitoring your connections to your IaaS cloud applications. As discussed earlier in this chapter, enterprise agents have multiple installation options in your local network but also can be installed in the cloud. For example, it is relatively easy to get an enterprise agent running in AWS or Azure. With agents in your cloud deployments, you can monitor a hybrid cloud scenario and create operational awareness across the Internet.

In [Figure 12-7](#), you can see some of the results from an Agent-to-Agent network test. This test shows average packet loss over time to an agent located in New Zealand.

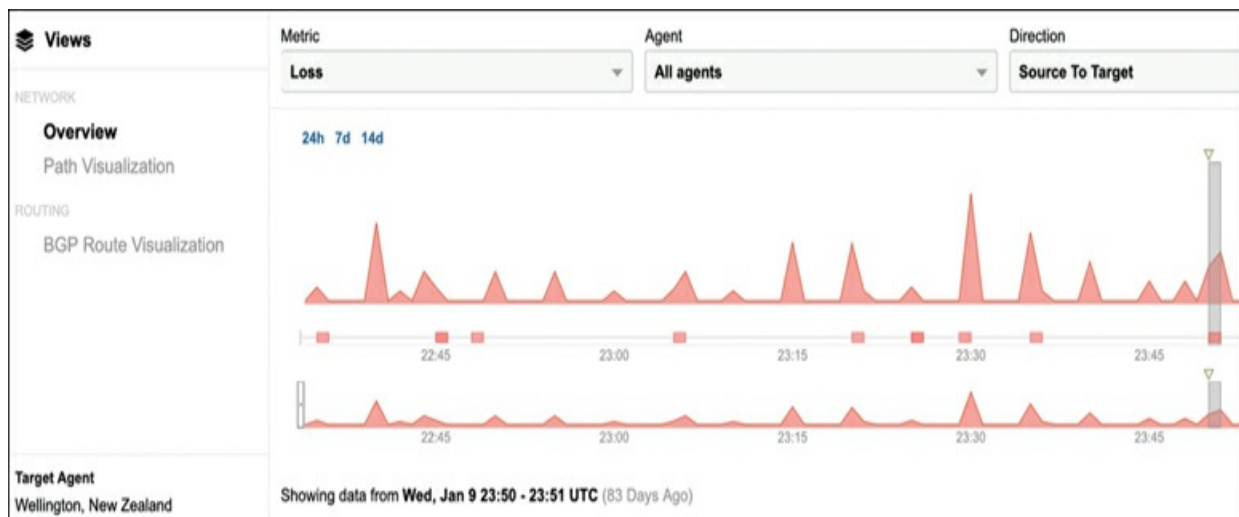


Figure 12-7 Viewing Loss from an Agent-to-Agent Network Layer Test

DNS Tests

Like the BGP protocol discussed in an earlier section, Domain Name Service (DNS) is a foundational Internet protocol. DNS is the protocol that translates a web domain name like www.cisco.com into an IP address for communication across the Internet.

You might have a well-architected app, running on a cutting-edge compute and storage infrastructure and served by a low-latency high-bandwidth efficient network. However, if DNS is not working properly, the user

experience will likely be poor if users are even able to access your application at all. Proper functioning of the DNS infrastructure is essential for the delivery of an application to users.

Three tests are part of the DNS test set: DNS Server, DNS Trace, and DNS SECurity Extensions (DNSSEC). Each is defined, along with some common use cases, in [Table 12-4](#).

Table 12-4 Summary of DNS Tests and Their Common Use Cases

DNS Test	Test Description	Common Use Cases
DNS Server	This test is designed to assess and monitor the performance and availability of DNS servers along with the records that they contain.	<ul style="list-style-type: none">• Alerting on incorrect DNS record mapping• Monitoring network performance between agents and target servers• Comparing DNS results and performance from around the globe
DNS Trace	This test verifies that the delegation of DNS records between parent and child zones is correct.	<ul style="list-style-type: none">• Testing the DNS server hierarchy for a domain, including time to resolve• Observing the DNS hierarchy of a target domain from various vantage points
DNSSEC	This test validates the authenticity of resource records.	<ul style="list-style-type: none">• Verifying that valid DNS signatures are being sent with DNS records• Validating DNS records based in DNSSEC• Observing the DNSSEC Trust Chain and Data Chain

In [Figure 12-8](#), you can see some of the output from a DNS Server test. For this test, iterative queries are made from enterprise agents to authoritative servers for [google.com](#). As you can see, the availability and response time are good. This test is often run for business-critical applications that are utilized over the Internet so that an immediate notification can be received if DNS

problems occur.

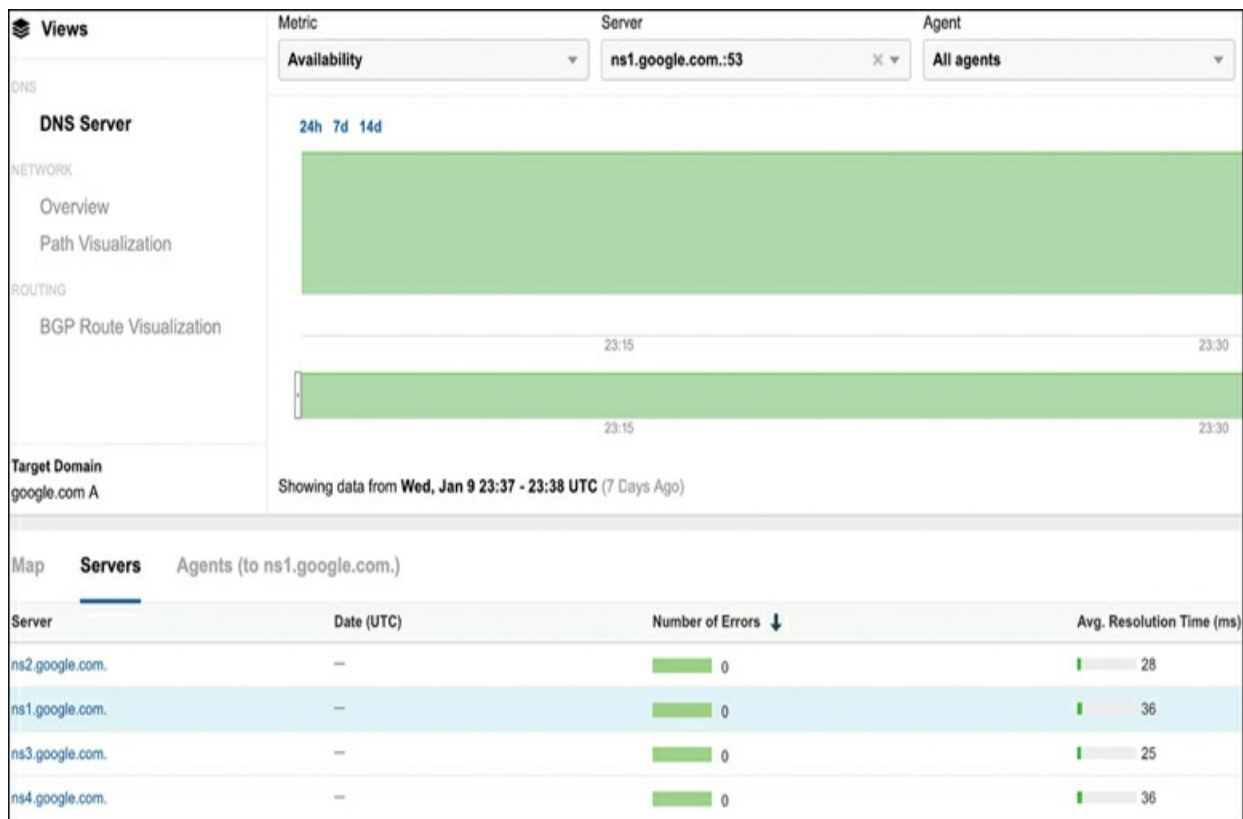


Figure 12-8 DNS Server Test

Web Tests

Web layer tests consist of three tests, each gathering progressively more detail. As the name suggests, these tests apply to application web servers primarily but could also be used to test API endpoints. The test targets may be publicly accessible or internal to the enterprise.

The first and most basic test is the HTTP Server test. As the name suggests, this test looks at the availability and performance, including response time and throughput, of a hosted HTTP service or web server. The next level of testing is the Page Load test. This test is useful for assessing a website from the user perspective. The Page Load test captures not only how quickly a page loads but also how various page elements, such as images, scripts, and stylesheets, contribute to the overall load time.

[Figure 12-9](#) shows the output from a Page Load test for [google.com](#). At the

top of the page, you can see a graph of the average page load time over the span of a few hours.

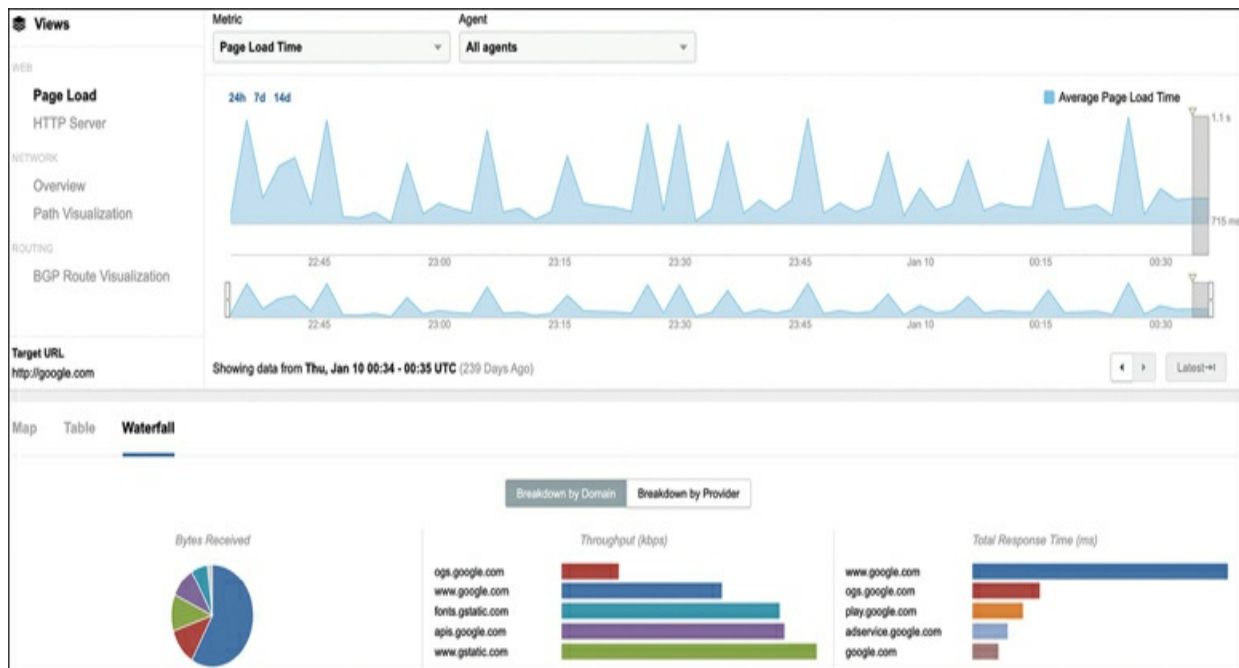


Figure 12-9 Web Layer Page Load Test

The last and most detailed web server test is the Transaction test, which emulates user interactions with a website. The purpose of this test is to dive deeper into the website, beyond the first page, to uncover problems that may be further in a user's workflow. For example, you may want to test an e-commerce site by logging in, selecting an item for purchase, and making sure the checkout screen loads. Using a series of scripted steps, the Transaction test can click buttons, enter data, and so on. [Figure 12-10](#) shows a Transaction test illustrating both a page load and a login.

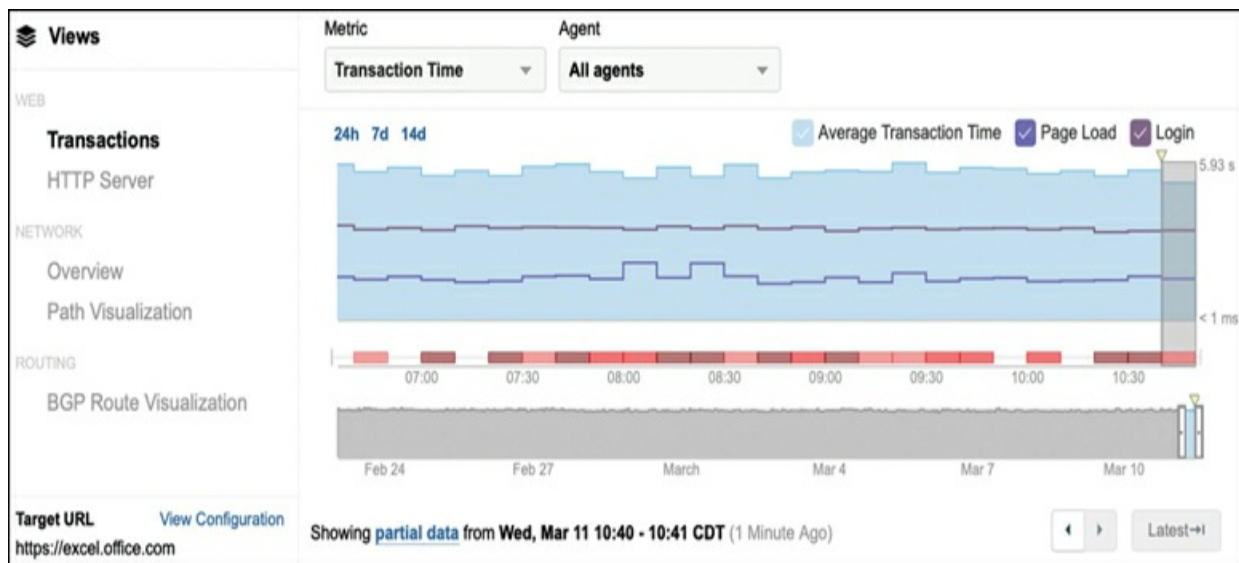


Figure 12-10 Web Layer Transaction Test

A fourth test that is also available as part of the web tests is the File Transfer Protocol (FTP) Server test. The FTP protocol is used for moving files across a network. As a complement to other network tests, you can use the FTP Server test to emulate how large data transfers move between points in your network. Common use cases for the FTP Server test include the following:

- Verifying the availability and performance of an FTP server
- Measuring throughput and bandwidth capacity
- Validating SSH operation by selecting SFTP and performing a list test to any SSH-enabled server

Note

Although FTP itself is an older protocol, more modern versions are supported as part of the FTP Server test, including Secure File Transfer Protocol (SFTP) and FTP Secure (FTPS). Because FTPS uses Secure Sockets Layer (SSL) to secure its sessions, FTPS may also be referenced as FTP-SSL in addition to FTP Secure.

[Figure 12-11](#) highlights the results from an FTP Server test to an external test point, speedtest.tele2.net. The graph across the top shows an average availability of 100 percent across a period of a couple of hours. The map at the bottom details the geographic locations of the destination test point as

well as all the ThousandEyes agents being used to originate the FTP Server test.

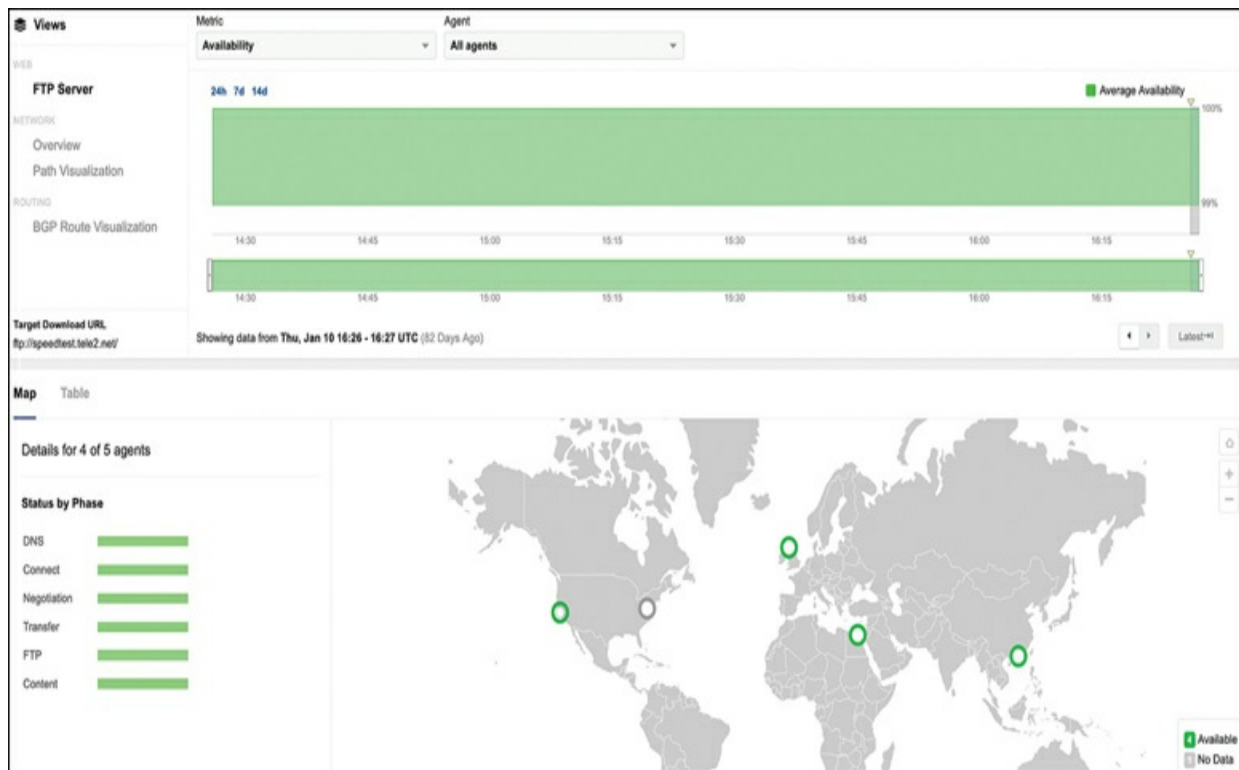


Figure 12-11 Web Layer FTP Server Test

Voice Tests

Businesses today are more connected than ever through voice and video communications, often on a global scale. Being able to proactively test and monitor both internal and external IP media communication paths is critical to identifying and accurately troubleshooting issues. The Voice layer tests from ThousandEyes provide this capability through two tests: SIP Server and RTP Stream.

Just as its name alludes to, the SIP Server test focuses on the control plane, or more specifically, the SIP voice call control. As the most common call control protocol, SIP is responsible for setting up voice and video calls and other voice over IP (VoIP) services by defining the messaging formats and the flows for the exchange of these messages. You should use this test in scenarios where you want to confirm SIP server availability and response time, testing SIP registration flows, and observing SIP Register and Options

request and response. [Figure 12-12](#) shows the results from a SIP Server test.

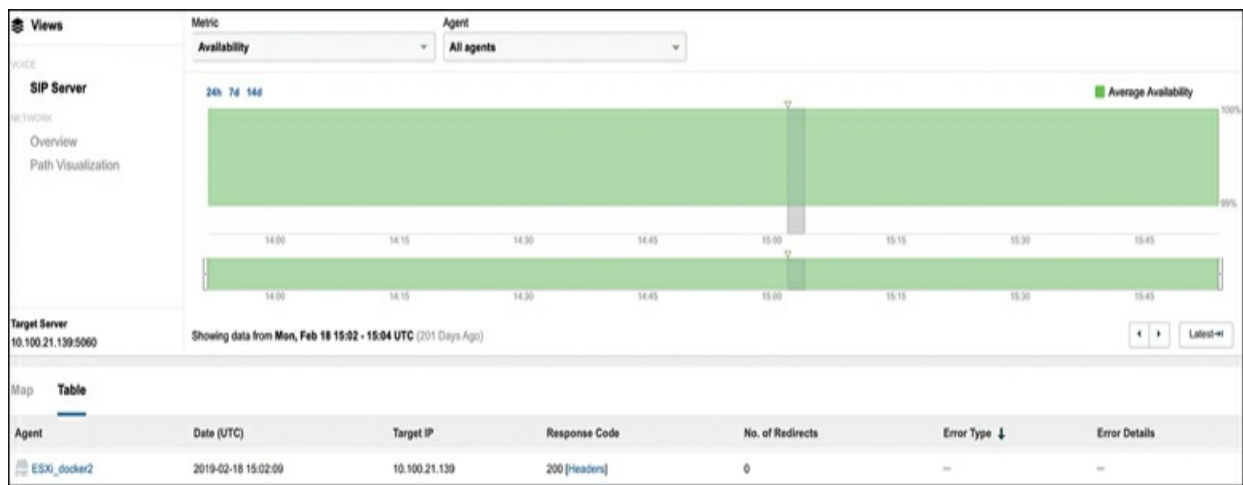


Figure 12-12 Voice Layer SIP Server Test

In [Figure 12-12](#), average availability of a SIP server is shown as 100 percent across the top of the figure. In this example, a single agent is being used, but you can configure multiple agents across various locations for broader coverage.

The second Voice layer test, RTP Stream, does not focus on the control plane but instead covers the data plane. The data plane in this case is the RTP stream that carries the voice or video media. We discussed the concepts of control plane and data plane in [Chapter 2](#), in the “[Review of SDN Logical Model](#)” section.

The RTP Stream test works by creating a simulated voice data stream between two ThousandEyes agents that are acting as VoIP endpoints. This voice data stream is composed of RTP packets that use UDP as the transport protocol. This allows the determination of mean opinion score (MOS), packet loss, discards, latency, and packet delay variation (PDV) metrics. Additionally, you can identify the node that is causing degradation of the RTP stream.

[Figure 12-13](#) shows the output from an RTP Stream test. In this example, the jitter or average packet delay variation is highlighted in the graph at the top of the figure, between the originating agent in Cape Town, South Africa, and a target agent in Albuquerque, New Mexico, in the United States. You can see the DSCP value of EF was configured for RTP packets in this stream, along

with a G.711 codec. In the bottom left of the figure, other metrics like MOS, loss, discards, and latency are also available.

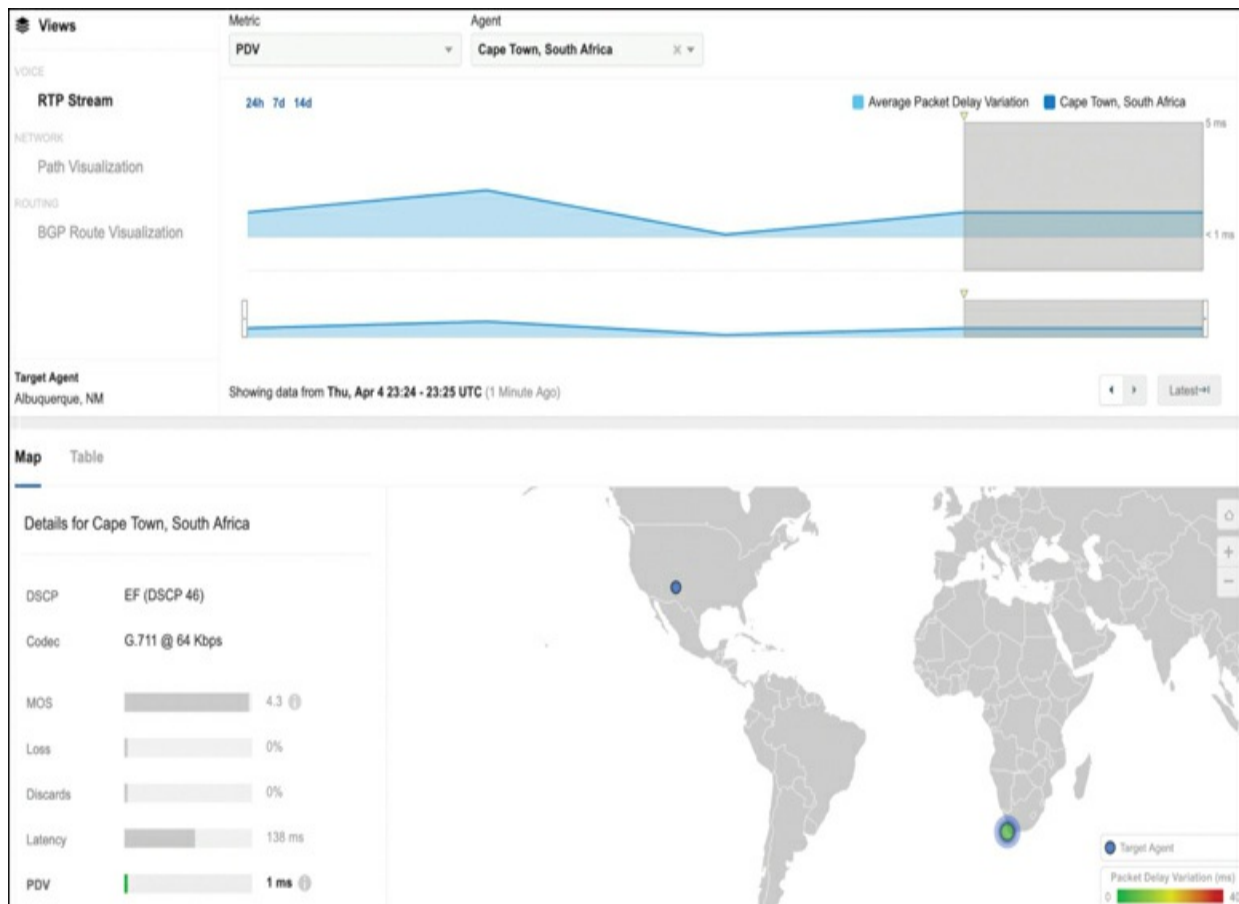


Figure 12-13 Voice Layer RTP Stream Test

Endpoint Agent Tests

In the preceding sections, we provided an overview of the tests for enterprise and cloud agents. These tests were built around five layers: Routing, Network, DNS, Web, and Voice. As you will see, the endpoint agent tests that we cover in this section will have some similar capabilities but also have some differences. As mentioned previously, endpoint agent tests are aimed at the end-user experience and are enabled and configured at the user device level. [Figure 12-14](#) provides an overview of the endpoint agent capabilities.

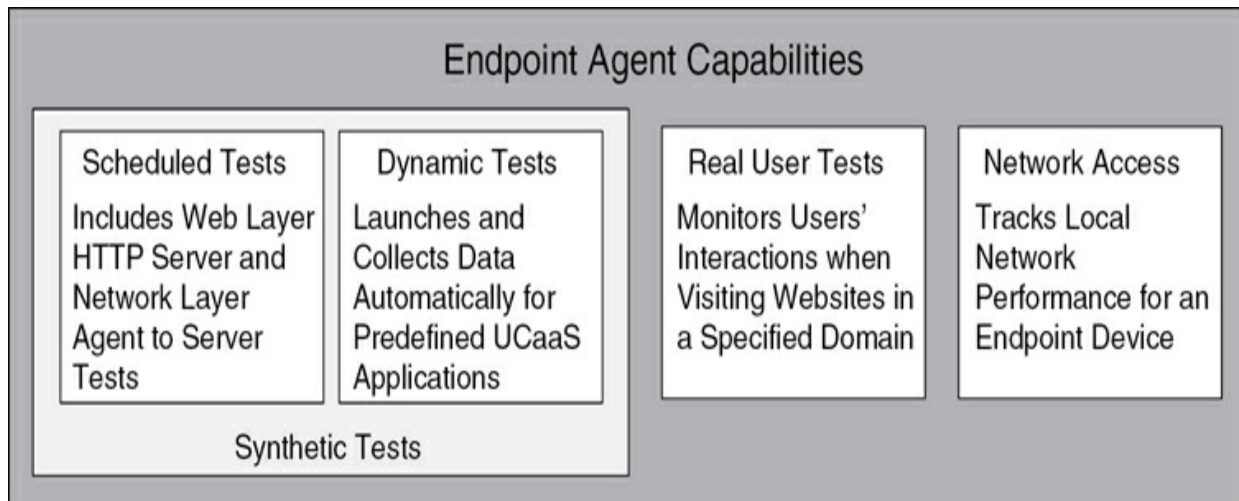


Figure 12-14 Endpoint Agent Capabilities Overview

The endpoint agent capabilities in [Figure 12-14](#) are accessible when you add a new endpoint agent test. Two options are available for adding an endpoint agent test from the ThousandEyes web portal: synthetic tests and real user tests. From the Synthetic test option, you can then easily add scheduled and dynamic tests.

Adding a synthetic test begins with adding an application to monitor. Many well-known applications are preconfigured to save you time in figuring out associated network information and to streamline the test configuration process. [Figure 12-15](#) shows some of the preconfigured tests for well-known SaaS applications. Note that you can search for additional applications or use the custom application option to build a test for your own applications or applications that are not preconfigured.

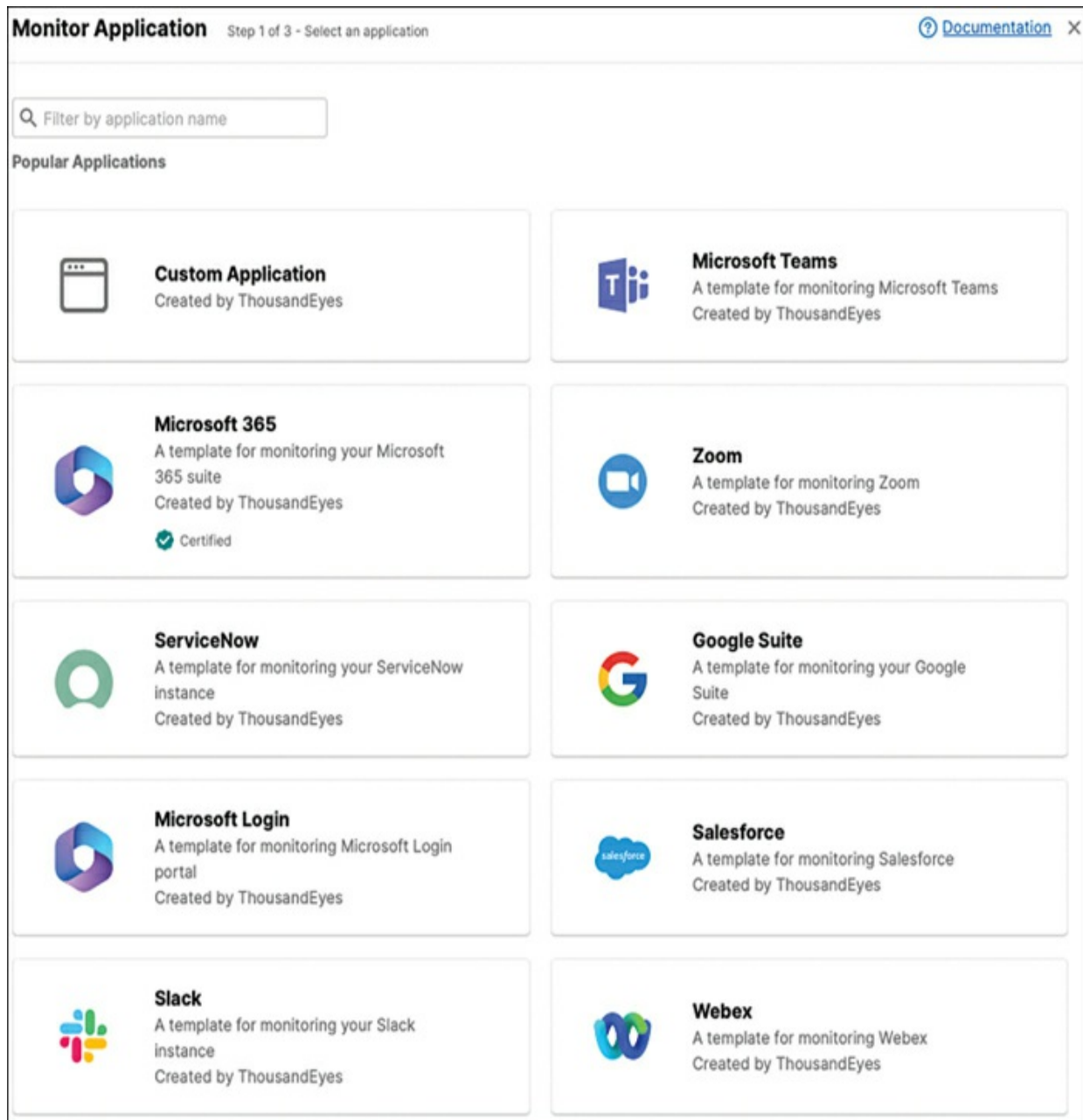


Figure 12-15 Adding an Application to Monitor for an Endpoint Agent Synthetic Test

After selecting a predefined application to monitor or the custom application option, you can now configure your test details. As shown in [Figure 12-14](#), scheduled and dynamic test options are possible when configuring a synthetic test. For scheduled tests, you can choose an HTTP Server test or a Network test. The HTTP Server test is similar to the enterprise and cloud agent test of the same name, and the Network test is like the Agent-to-Server test for

enterprise and cloud agents. We covered both tests earlier in the “[Enterprise and Cloud Agent Tests](#)” section.

Endpoint agent scheduled tests run without user interaction (compared to real user tests, which we will discuss later in this section). You have full control of when and under what conditions scheduled tests run. For example, a scheduled test deployed to a laptop can be switched on or off as the laptop transits through different locations, wireless and wired networks, VPNs, and proxies. [Figure 12-16](#) shows a scheduled HTTP Server test for a group of endpoint agents. The top of the HTTP Server test in [Figure 12-16](#) highlights the server availability and average number of agents participating in the test.

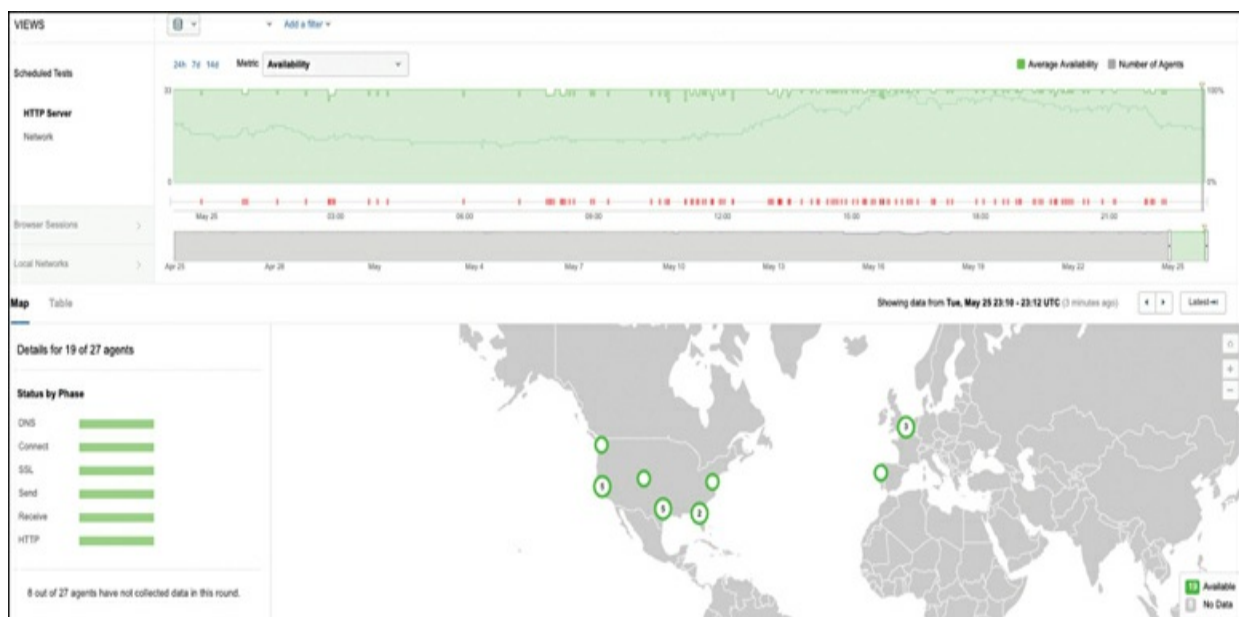


Figure 12-16 HTTP Server Test Results for an Endpoint Agent

The other part of synthetic testing for endpoint agents is dynamic testing. Dynamic testing is valid for UCaaS applications, like Cisco Webex. What makes these tests dynamic is that they run only when the specific UCaaS application that the test has been enabled for is launched on the endpoint. This approach optimizes both system and network resources. The data collected from when a dynamic test runs on an endpoint is handy for troubleshooting issues that the user faced while using the application. [Figure 12-17](#) shows a dynamic test for Cisco Webex.

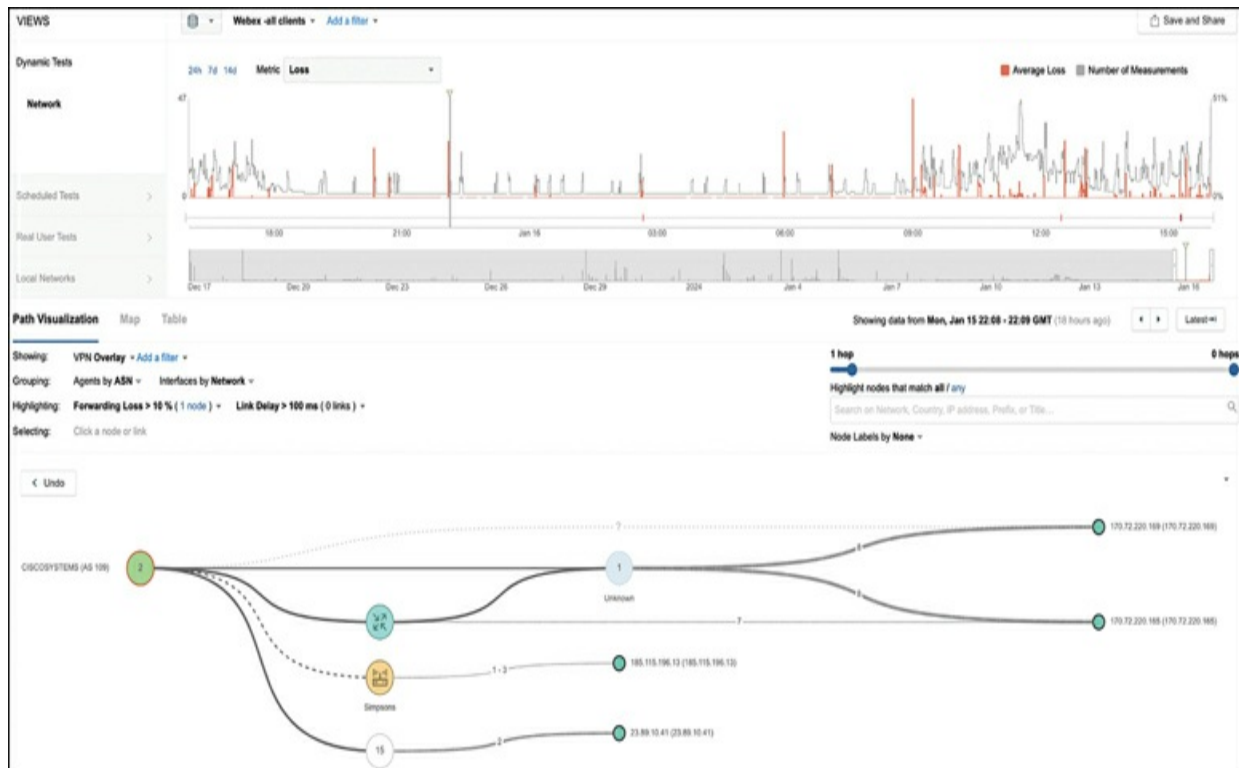


Figure 12-17 Dynamic Test Results for an Endpoint Agent

In [Figure 12-17](#), the top graph shows the average loss by endpoint agents participating in the test. In the bottom portion of [Figure 12-17](#), you can see a path visualization of the agents and their connection to Cisco Webex. We will discuss path visualization more in the next section.

Real user tests are enabled simply by specifying a domain to be monitored. When an endpoint agent detects a user connecting with this domain, the test will run. This test collects performance data from actual users and provides insights into how applications and websites are performing in real-world conditions.

The last capability from [Figure 12-14](#) is network access. This endpoint agent feature gathers local network connectivity data and is helpful in troubleshooting local network problems, especially when related to Wi-Fi connectivity. Network access insights can be found under Local Network as part of Endpoint Agent Views. From network topology maps showing how endpoint agents are connected to gateways, proxies, DNS servers, and so on to endpoint wireless information highlighting parameters like signal strength and retransmissions, the network access function is critical for determining

when issues may be caused by local network problems versus problems further downstream.

Path Visualization and Dashboard Snapshots

While ThousandEyes has numerous features, capabilities, and types of tests that make it a critical tool for monitoring your network, two key features around data visualization deserve special attention. These features are path visualization and dashboard snapshots.

The path visualization view offered in ThousandEyes offers a graphical view of network paths for certain tests. If you are familiar with the ICMP traceroute function, you can think of path visualization as a graphical version of this function with a lot more data and metrics. The path visualization view is included as part of results when the test that you are running captures network metrics. This means that tests like Network, HTTP Server, Page Load, and DNS Server tests can provide a path visualization. [Figure 12-18](#) shows a simple path visualization for two agents connecting to AWS. One agent is in Brooklyn, New York, and the other is in Amsterdam, Netherlands.

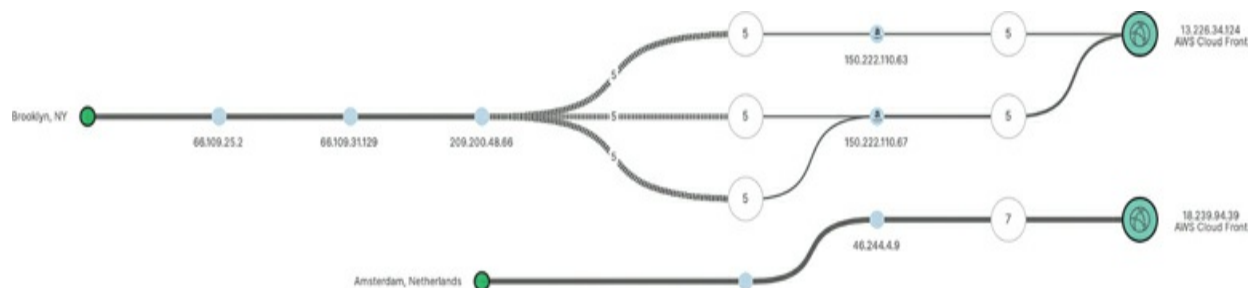


Figure 12-18 Path Visualization View

In [Figure 12-18](#), circles represent nodes and endpoints. Obviously, the circles on the ends are agents or targets. In this example, the agents are on the left, and the AWS targets are on the right. The circles in the middle represent nodes, and some can be expanded because they represent multiple nodes if you want to drill down further into the path. Additionally, you can click an individual node to get more data about it, including IP address, DSCP settings, and response time. [Figure 12-19](#) shows a snippet of this information for a node in a path visualization view.

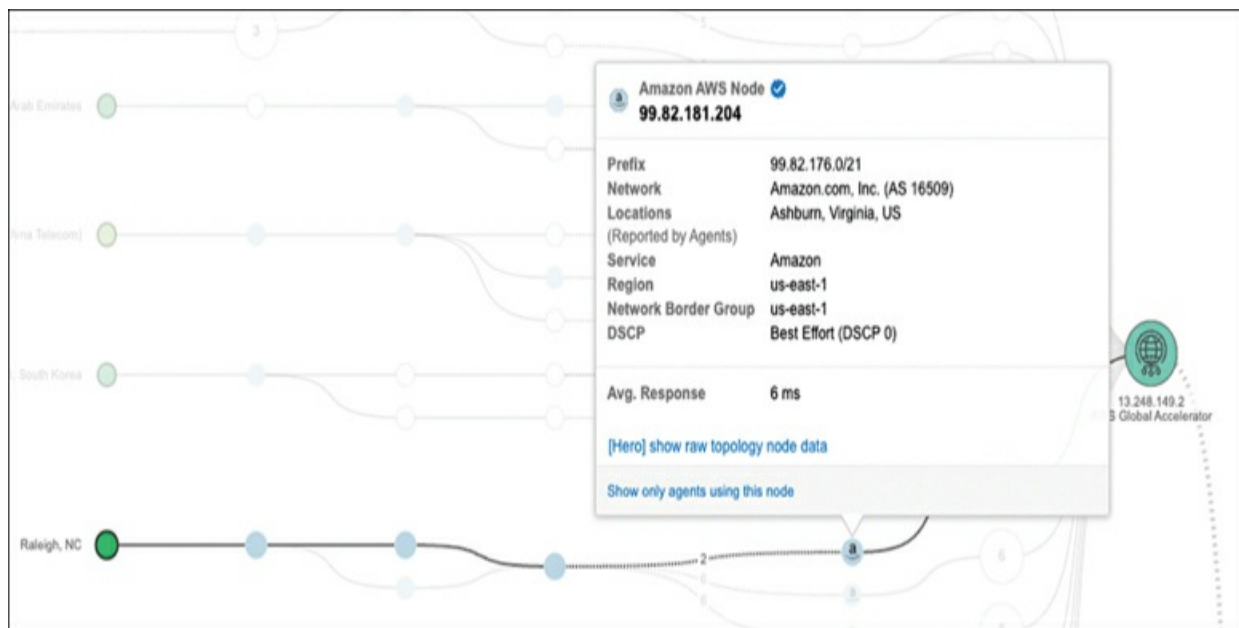


Figure 12-19 Node Details from a Path Visualization View

Errors and other problems, when detected, are also shown in the path visualization view. Using various graphical icons and symbols, you can quickly home in on problems. This end-to-end visibility between an agent and another agent or target make path visualization an indispensable tool for finding and understanding network issues. Furthermore, when issues are found, you can easily share them using the ThousandEyes snapshot feature.

The snapshot feature is a powerful tool for exporting and saving test results. Dashboard snapshots can be scheduled for regular intervals and automatically emailed to a list of recipients. A nice capability of dashboard snapshots is that you can share them with others outside your organization, even if they are not ThousandEyes customers.

For example, you are seeing discards and packet loss using path visualization for an HTTP Server test to a SaaS application hosted on the Internet. If the path visualization shows these issues occurring in your service provider, you can take a dashboard snapshot of the issue that you are seeing and simply send others the link. This makes dashboard snapshots quite useful when troubleshooting with partners and third parties.

Internet, WAN, Cloud, and Traffic Insights

Zooming out from agent tests to wider impacting events and issues, ThousandEyes offers Internet Insights, WAN Insights, Cloud Insights, and Traffic Insights. Each of these capabilities is summarized in [Table 12-5](#).

Table 12-5 Summary of Internet, WAN, Cloud, and Traffic Insights Capabilities

Area of Visibility	Description
Internet Insights	Leveraging collective intelligence based on the cloud and enterprise agent network, Internet Insights provides macro-level visibility into outages that may affect you. A global map is utilized to highlight network and application outages across the global Internet.
WAN Insights	Taking advantage of integrations with Cisco vAnalytics and Cisco vManage, WAN Insights analyzes application data flow records from all routers in the SD-WAN fabric. This allows it to generate path recommendations for application categories.
Cloud Insights	Cloud Insights provides end-to-end visibility into your cloud environments by integrating with cloud providers like AWS to monitor network traffic flows and infrastructure configuration changes within or across your virtual private clouds. With advanced topology visualization and dependency mapping, it enables you to quickly understand relationships between cloud resources, identify potential issues, and optimize performance.
Traffic Insights	Traffic Insights delivers granular visibility into network traffic by analyzing flow data, such as NetFlow, IP Flow Information Export (IPFIX), and sampled Flow (sFlow), from across your environment, enabling you to identify traffic patterns, performance bottlenecks, and anomalies in real time. IPFIX and sFlow are network flow monitoring protocols, similar to NetFlow, that collect and export detailed traffic statistics from network devices to provide visibility into network performance and usage patterns. Leveraging advanced flow-based analytics and intuitive visualizations, it correlates traffic with applications and endpoints, uncovers application dependencies, and empowers you to optimize network performance and troubleshoot issues with precision.

In [Table 12-5](#), WAN Insights, Cloud Insights, and Traffic Insights are similar while Internet Insights is a bit different. WAN Insights, Cloud Insights, and Traffic Insights require specialized integrations for you to utilize them, and they are focused on your own network connections. These have a closer tie with the agent tests covered earlier in this chapter.

Internet Insights gives you macro-level visibility into outages that may affect you, using the collective intelligence of ThousandEyes' entire agent network. It does not require any additional integrations but instead leverages anonymized test data from all the cloud and enterprise agents to provide a lens into third-party networks and applications. This data is aggregated and after analysis is displayed on a global outage map. This map can be accessed in your ThousandEyes account.

Note

ThousandEyes publishes a public version of Internet Insights at <https://www.thousandeyes.com/outages/>. This public version is limited but provides a good overview of the current state of the global Internet. The Internet Outages version available to ThousandEyes customers allows for a deeper inspection of the outage data and other functions and customizations.

The macro viewpoint of Internet Insights is helpful in answering questions like “Am I the only one having a problem with a certain Internet application or service?” or “Is this application down?” Sometimes the problem is not a fault of yours or the application you are using, but an Internet provider or another node out of your control. Internet Insights adds more layers of information to help better evaluate probable causes. Agent tests can show where communications are failing, but Internet Insights can relate a test error to a broader pattern of failures related to geography, providers, domains, and so on. [Figure 12-20](#) captures how Internet Insights, enterprise and cloud agents, and endpoint agents interrelate when it comes to the information and analysis that each provides.

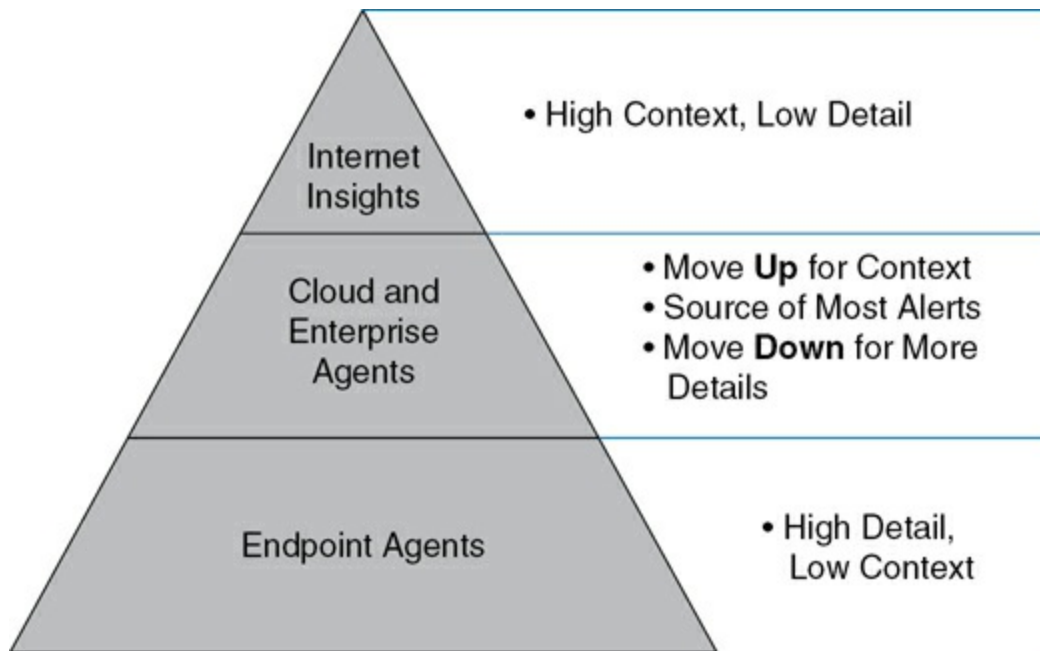


Figure 12-20 Data Context and Detail for Internet Insights and Agents

Integrations

Being able to integrate your SaaS application with other applications greatly increases its usefulness and value. This capability is even more important for a SaaS application like ThousandEyes that is focused on monitoring and visibility. For example, when ThousandEyes detects a problem, an integration with other systems can provide proper alerting or an action to be taken in real time. Automated, complex workflows can be built with the proper integrations.

Note

When discussing ThousandEyes integrations, you will also often see the support of ThousandEyes software on Cisco products also being referred to as an *integration*. We touched on this type of integration earlier in the chapter when agents were first introduced. The reason is that the actual integration is typically installing a ThousandEyes agent on a Cisco hardware platform. Current Cisco product integrations include select Webex devices, select Catalyst and Nexus switches, select ASR and ISR routers, and select Meraki products. This is an area where ThousandEyes continues to expand.

For the most current listing of Cisco product integrations, refer to <https://www.thousandeyes.com/integrations/cisco>.

As mentioned earlier in the chapter, ThousandEyes supports both prebuilt and custom integrations. The prebuilt integrations enable ThousandEyes to be connected with many other applications with minimal configuration and setup. We will discuss more detailed examples of prebuilt integrations in the next section and will cover custom integrations in the section after.

Prebuilt

As you may recall, prebuilt integrations were introduced earlier in [Chapter 2](#) in the “[Prebuilt Integrations with Apps, Connectors, Modules, and Adapters](#)” section. In this section, coverage of this topic will be from the ThousandEyes perspective. Because of its role in monitoring and observability, prebuilt integrations for ThousandEyes focus mainly on alert notifications to other well-known SaaS applications. [Figure 12-21](#) provides an overview of such a scenario.

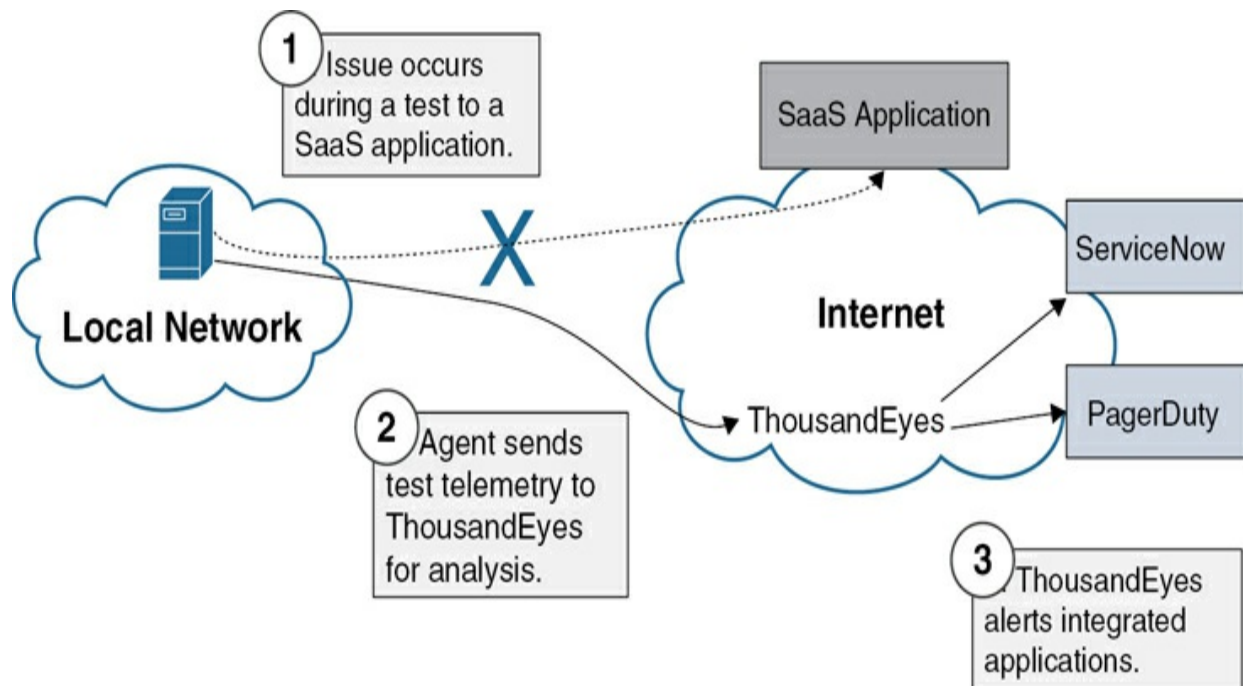


Figure 12-21 ThousandEyes Sending Alert Notifications to Integrated Applications

As discussed in previous sections of this chapter, ThousandEyes can be configured with numerous agents, and then various tests can be assigned to those agents. For the sake of simplicity, let's assume a single test you configured on a ThousandEyes agent encounters an issue, as shown in the first step of [Figure 12-21](#). The agent streams this test telemetry back to the ThousandEyes application for analysis, as indicated in step 2. Based on alert thresholds and configured integrations with ServiceNow and PagerDuty, ThousandEyes can then automatically pass information about this issue to other SaaS applications. This is depicted by step 3 of [Figure 12-21](#). In the case of both ServiceNow and PagerDuty, they are prebuilt integrations from ThousandEyes. You can get more information about available ThousandEyes prebuilt integrations at <https://www.thousandeyes.com/integrations/>.

Prebuilt integrations are always worth looking at first to see if they meet your needs. Although you may lose some ability to customize, if the integration fulfills your needs and use cases, it is usually much less work compared to a custom integration. We will discuss custom integrations for ThousandEyes in the next section.

Custom

For custom integrations, ThousandEyes supports both APIs and webhooks. We provided examples of how APIs and Webhooks work using ThousandEyes in [Chapter 2](#) in the “[Custom Integrations with APIs, Webhooks, and WebSockets](#)” section. In this section, we will take you a step further and show how ThousandEyes can use a custom API integration with another application or service that supports OpenTelemetry (OTel).

As discussed in the “[OpenTelemetry and MELT](#)” section in [Chapter 11](#), “[Observability and Monitoring: Cisco AppDynamics and Splunk](#),” OpenTelemetry is a collection of APIs, SDKs, and tools that allow you to generate, collect, and export telemetry data for more effective observability. Since it is an open-source, vendor-neutral framework, you can use OTel to easily export telemetry to any other application that also supports it.

ThousandEyes supports formatting the analyzed telemetry from its agents into the OTel format, which can then be shared or exported. This integration is typically built using the ThousandEyes API to set up an OpenTelemetry

data stream.

Note

ThousandEyes also makes it possible to set up an OTel integration through its UI. Therefore, you can integrate ThousandEyes with OTel through both a prebuilt integration in the GUI or a traditional custom integration via an API. While the UI offers convenience for setup, the API method is recommended for its flexibility, automation capabilities, and efficiency in managing integrations and tasks programmatically.

[Figure 12-22](#) illustrates an OpenTelemetry integration with ThousandEyes. As you can see, raw data from agents is collected and analyzed and then turned into a stream of OTel data. This OTel data is sent to an OpenTelemetry Collector. The OpenTelemetry Collector is a core element of the OpenTelemetry framework and offers numerous benefits. Most importantly, it is an intermediary that acts as a unified collection point for different types of telemetry as opposed to a dedicated agent or collector for each telemetry signal. While [Figure 12-22](#) just shows ThousandEyes as the sole source of an OTel data stream, there are often many sources and more types of telemetry that make the Collector's role even more important.

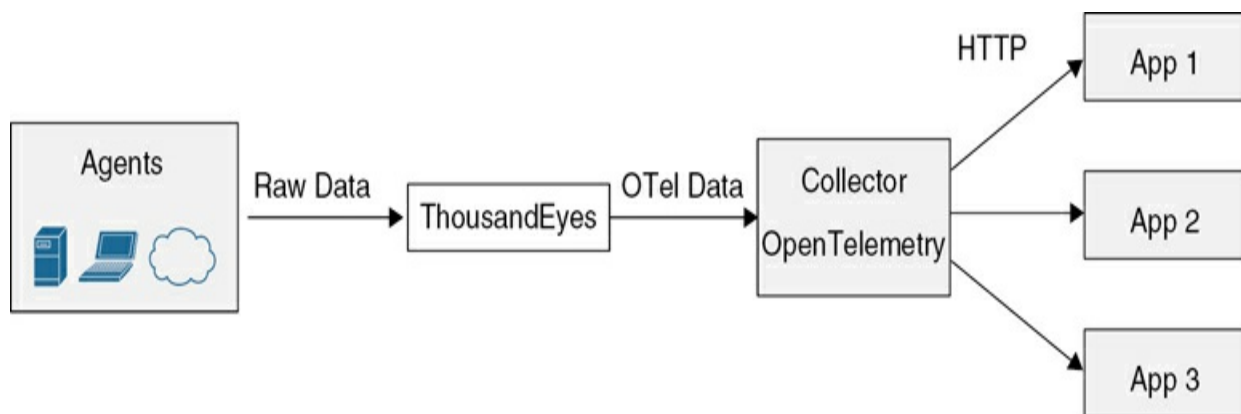


Figure 12-22 OpenTelemetry Custom Integration with ThousandEyes

An OpenTelemetry Collector has an exporter function that converts OpenTelemetry-formatted data into a back-end compatible format. For the scenario depicted in [Figure 12-22](#), the exporter function of the Collector converts to HTTP format for the three back-end applications that are

connected. You should note that formats other than HTTP are also used, depending on the requirements of the back-end application.

The back-end apps shown in [Figure 12-22](#) could vary but most likely would be an observability-related application, like Grafana, Honeycomb, or SigNoz. Cisco AppDynamics or Splunk could also be connected to ThousandEyes in this manner. However, both AppDynamics and Splunk have prebuilt integrations with ThousandEyes that make connecting them much easier.

Cisco AI Assistant Integration

With the integration of the Cisco AI Assistant into the ThousandEyes platform, you can efficiently analyze the large volumes of telemetry data collected by agents. The Cisco AI Assistant uses natural language processing to allow users to ask questions in a conversational manner about their ThousandEyes deployment and receive guidance on troubleshooting network issues. It provides the ability to instantly interpret in-product views and visualizations so users can understand the underlying root cause of issues much faster.

One of the more compelling features of the Cisco AI Assistant is the Views Explanations. This capability provides automated analysis of test results, helping to identify the fault domain and potential root cause of network performance issues. Views Explanations analyzes ThousandEyes test data across multiple layers—network, application, and routing—and provides a summary of the issue in natural language. The feature processes metrics from network paths, BGP monitors, and application tests to determine where in the network path a problem is occurring.

When you select a test round that shows degraded performance or an alert condition, you can click the Explain Selection button. The AI Assistant then examines

- Network layer metrics (packet loss, latency, jitter)
- Application layer metrics (response time, availability)
- BGP routing information
- Path visualization data

- Historical performance baselines

The analysis identifies which network segment or component is responsible for the issue. For example, if packet loss is occurring within an ISP's network between your enterprise agent and a SaaS application, Views Explanations will indicate this loss in its summary. [Figure 12-23](#) shows a View Explanation using the Cisco AI Assistant in ThousandEyes.

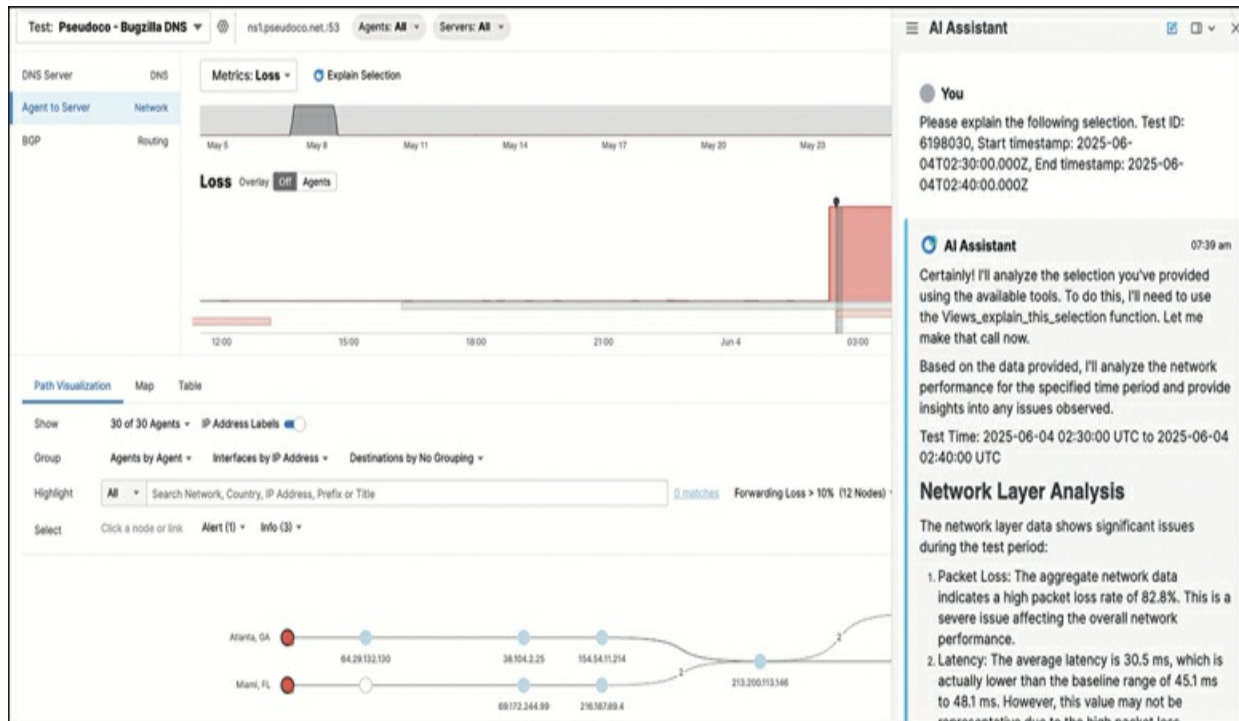


Figure 12-23 Cisco AI Assistant in ThousandEyes Showing a View Explanation

The Cisco AI Assistant also includes an Account Health Check feature that monitors the operational status of your ThousandEyes deployment. This feature checks for common configuration issues such as

- Enterprise agents that are offline or not reporting data
- Agents with high CPU or memory utilization
- Tests that are not running as scheduled
- Missing alert rule configurations

For example, you can ask the AI Assistant questions in natural language,

such as "Are any of my enterprise agents offline?" The system queries your account configuration and agent status, then returns a summary of any issues found along with recommendations for resolution. This capability helps maintain the health of your monitoring infrastructure by identifying potential blind spots before they affect your ability to detect network issues.

By seamlessly integrating the Cisco AI Assistant with ThousandEyes, organizations can dramatically streamline network monitoring and troubleshooting workflows. The Views Explanations feature empowers users to quickly interpret complex telemetry data and pinpoint the root cause of issues without the need for deep technical expertise. This AI-powered feature aims to reduce the time required to identify the root cause of network issues (mean time to identify, or MTTI) and make ThousandEyes' network intelligence more accessible to team members with varying levels of network expertise. Combined with other functions, like proactive account health checks, the AI Assistant not only accelerates problem resolution but also strengthens the overall resilience of your monitoring environment.

Summary

In this chapter, we provided an overview of the Cisco SaaS application, ThousandEyes. ThousandEyes is a tool that provides visibility and monitoring of your traffic, not only in your network but extending out to the public Internet as well. Additionally, ThousandEyes performs analysis on the data it collects to provide real-time insights on network performance and optimization.

In the first section of this chapter, we discussed the high-level architecture. You learned how ThousandEyes collected its telemetry through agents and how insights from these agents were displayed through the ThousandEyes web portal.

In the next section, we took a deeper dive into the types of agents used by Thousand Eyes. For user devices, the endpoint agent is utilized to monitor and measure the user experience. Enterprise agents have a more robust testing capability than endpoint agents and can be in your network infrastructure and cloud deployments. Lastly, cloud agents are located throughout the global Internet. Managed by ThousandEyes, they are available to all customers for

initiating and terminating tests.

In the “[Agent Tests](#)” section, we discussed the various tests associated with the ThousandEyes agents. For enterprise and cloud agents, tests were grouped and then covered by the Routing, Network, DNS, Web, and Voice layers. For endpoint agents, the tests were organized into synthetic tests that included scheduled and dynamic tests and real user tests.

The next section highlighted two compelling features of ThousandEyes: the path visualization view and dashboard snapshots. Path visualization is an element that a user can interact with to trace the flow of data through a network for a better understanding of network traffic and performance as well as troubleshooting. Dashboard snapshots allow for the easy sharing of test data and path visualizations, even with non-ThousandEyes customers.

In the next section, we looked at Internet, WAN, and Cloud Insights. While derived from agent telemetry and raw data, the dashboards associated with these insights allow for a higher-level view and context when monitoring and troubleshooting.

The last section in this chapter was about integrations, and we covered both prebuilt and custom integrations. Prebuilt integrations allow for a more streamlined setup of the connection to another application while custom integrations, like the OpenTelemetry example shown, typically require API access and development but provide more flexibility.

The ThousandEyes SaaS solution offers a complete monitoring and visibility solution for your network traffic and is integral for providing full-stack visibility when paired with other Cisco applications, AppDynamics and Splunk. To learn even more about ThousandEyes, please refer to *Cisco ThousandEyes: Digital Experience Monitoring and Troubleshooting*, by Aaron Trompeter and Robert Webb, which focuses on this topic.

References

- *Cisco ThousandEyes*: <https://www.ciscopress.com/store/cisco-thousandeyes-digital-experience-monitoring-and-9780138309183>
- ThousandEyes product documentation:

<https://docs.thousandeyes.com/product-documentation/getting-started>

- TE uses Terraform to configure and build cloud infrastructure at scale: <https://medium.com/thousandeyes-engineering/scaling-terraform-at-thousandeyes-b2a581b8b0b0>
- TE uses Kafka for messaging: <https://medium.com/thousandeyes-engineering/using-kafka-windowing-and-suppressions-with-heartbeats-in-internet-insights-to-detect-application-3074215af8aa>
- TE uses MySQL databases: <https://medium.com/thousandeyes-engineering/from-single-to-multiple-mysql-datasources-9cf477104a7d>
- Enterprise agent system requirements: <https://docs.thousandeyes.com/product-documentation/global-vantage-points/enterprise-agents/installing/enterprise-agent-system-requirements>
- Accelerate troubleshooting with AI-powered ThousandEyes Views explanations: <https://www.thousandeyes.com/blog/thousandeyes-views-explanations>

Chapter 13. Management: Cisco Meraki

The world of network infrastructure in 2006 was very different than it is today. Managing a network at the time predominately required connecting to routers, switches, and firewalls via a command-line interface (CLI) and manually entering commands to configure and troubleshoot the device. Network management often meant having a network management solution like CiscoWorks or HP OpenView polling network devices using the Simple Network Management Protocol (SNMP) to aggregate data across multiple devices.

Large enterprises could afford to hire top talent to design and manage their networks, but the small and medium business (SMB) market was fragmented and customers struggled with network management. Most customers in this market segment did not require many of the sophisticated and complex features available on Cisco IOS devices. Instead, customers were looking for solutions that were easy to deploy and manage with a simple feature set.

With the rise of cloud infrastructure, Meraki was formed to tackle these challenges by creating a solution where network devices could be configured and managed all from a simple cloud-based SaaS platform. Meraki first focused on cloud-managed wireless networking. This technology was well suited for cloud management because wireless networks are typically used to access the Internet and, even in small networks, usually involve managing many wireless access points, all with very similar configuration. Meraki expanded from wireless access points to switches and security appliances (a.k.a. firewalls) as well as mobile device management and gained significant market share in the SMB market.

Cisco acquired Meraki in 2012, bringing cloud-managed networking to the Cisco portfolio. This acquisition allowed Cisco to bring this technology to a much wider enterprise audience while addressing the gap in reaching the SMB market. Since then, the portfolio of products has expanded significantly beyond wireless, switching, and security appliances to include advanced security protection, software-defined wide area networks (SD-WANs), cameras, environmental sensors, and endpoint management with products that scale from SMB to enterprise environments. The feature set has also grown to be more sophisticated over time but still maintains its ease of use.

While the Meraki products were approaching management from the cloud as a SaaS solution, Cisco was also developing on-premises management solutions for enterprise and campus networks, starting with the Cisco Application Policy Infrastructure Controller Enterprise Module (APIC-EM), which would then morph to Cisco Digital Network Architecture Center (DNAC) to Cisco Catalyst Center. This solution (DNAC/Catalyst Center) allows customers to centrally manage their network without relying on the cloud. Catalyst products were managed through DNAC/Catalyst Center while Meraki products were managed by the Meraki Dashboard. Over time, the two platforms have converged and will likely continue to converge in the future, with Cisco Meraki Dashboard allowing customers to manage Catalyst devices and Catalyst Center providing visibility into cloud-managed devices.

Cisco Meraki is different from many of the other platforms discussed so far in this book in that its primary purpose is not to deliver a software-only solution, but rather to use SaaS as a management platform to deliver the outcome of easy-to-use, reliable networking. This is like how the Cisco Webex cloud can manage phones and video devices on customer premises, as we discussed in [Chapter 5, “Collaboration: Webex Meetings and Messaging,”](#) but in the case of Cisco Meraki, the primary purpose of the service is management and monitoring of on-premises devices, whereas Webex started as a pure software model and later expanded to include device management.

In this chapter, we will provide an overview of the Cisco Meraki platform, highlighting how the cloud enables the delivery of services to devices that live on the customer premises. We will then discuss how the Cisco Meraki platform fits into the SaaS reference architecture we have discussed previously.

Meraki Platform Capabilities

The Cisco Meraki platform provides a scalable, easy-to-use cloud-based platform that delivers cloud-managed networking, including the following technologies:

- Security appliances/routers (including SD-WAN)
- Layer 2 and Layer 3 switches
- Wireless access points
- Cellular WAN gateways
- Network video cameras
- Environmental sensors
- Device management

Before diving into each of these individual areas, we will discuss the key component that provides the primary user interface to the Cisco Meraki platform—the Meraki Dashboard.

Meraki Dashboard

Cisco Meraki Dashboard is the mechanism by which customers access and manage their Cisco Meraki devices. Unlike they do with traditional network devices, administrators do *not* directly connect to a network device like a router or a switch to manage or configure it. Instead, they configure the device in Meraki Dashboard, and when the device connects to the Meraki Cloud over the Internet, it automatically configures itself. Because the device needs to be able to connect to the Internet and reach the Cisco Meraki cloud, some limited configuration is possible directly on devices if they need some provisioning to get to the Internet—for example, if they need a static IP address—but for the most part, all configuration is done in Cisco Meraki Dashboard. [Figure 13-1](#) shows how networks, including campus, branch, retail, and teleworker locations, all connect to the cloud SaaS platform, enabling browser-based management of the network through Cisco Meraki Dashboard.

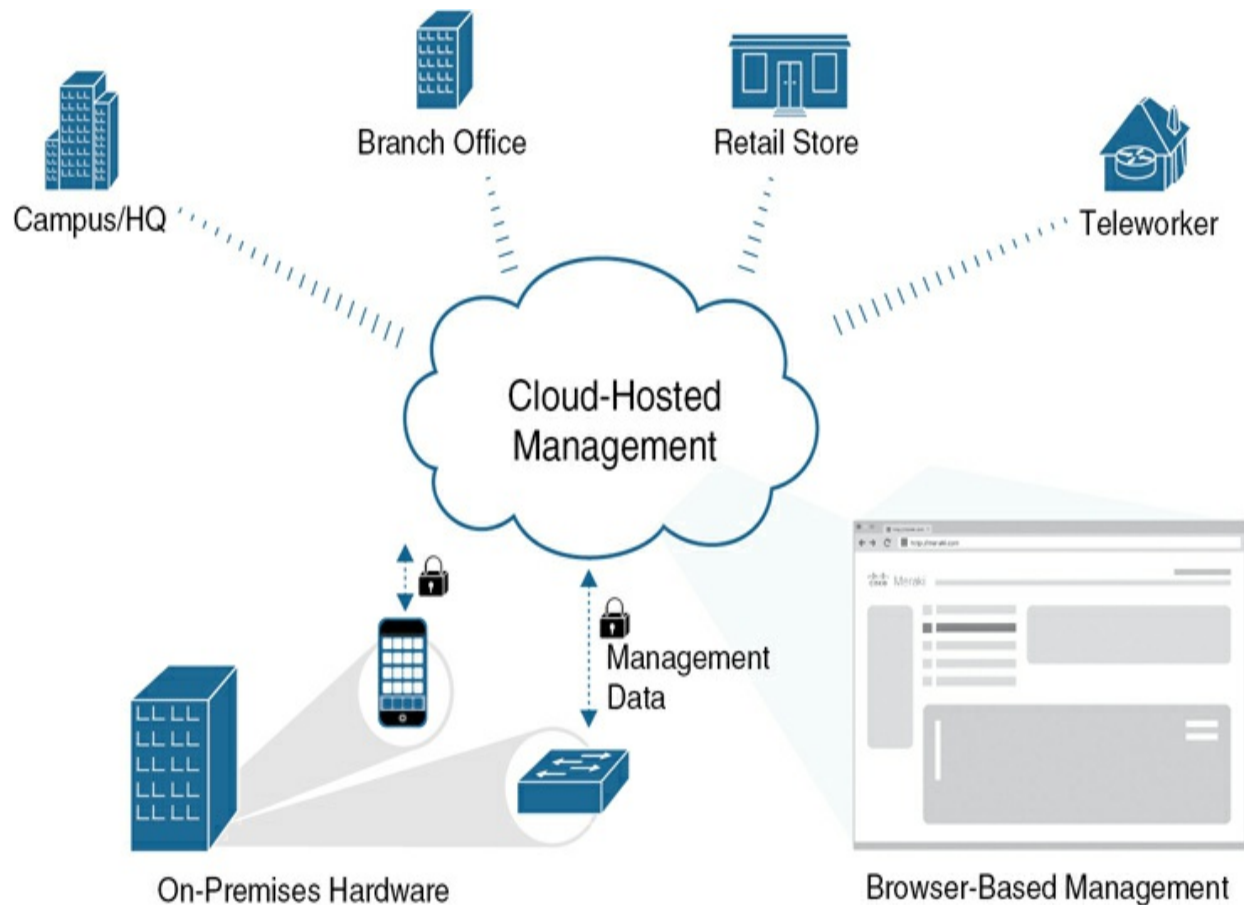


Figure 13-1 Cisco Meraki High-Level Architecture

Figure 13-2 shows the Assurance Overview page from the Cisco Meraki Dashboard, highlighting how the dashboard provides an easy-to-use management interface for the entire network. The left side of the page highlights all the various technologies available for management such as Security & SD-WAN, Switching, and Wireless.

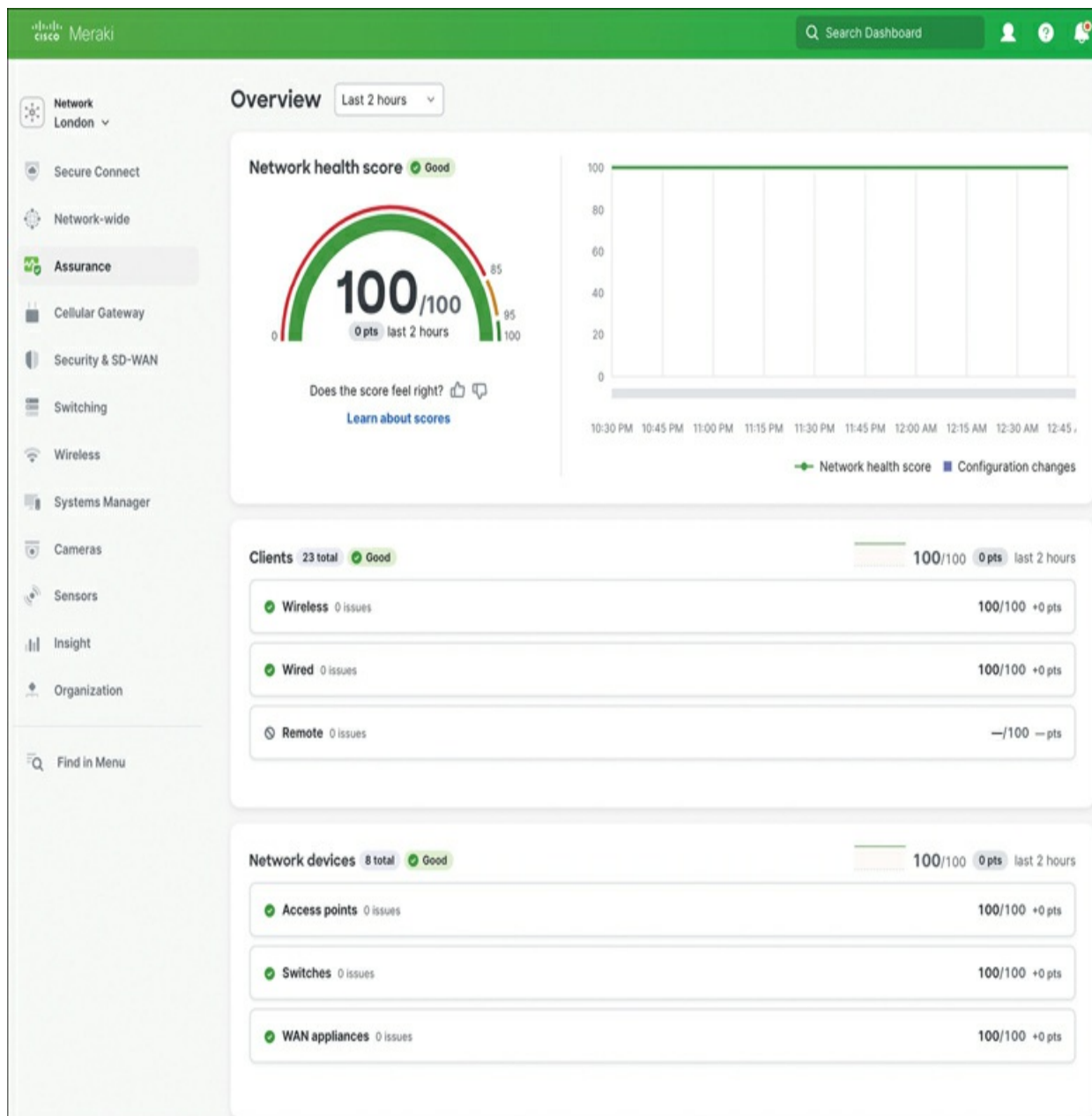


Figure 13-2 The Assurance Overview Page from Meraki Dashboard

When we talk about how Cisco Meraki is a SaaS platform, we are primarily referring to the infrastructure that provides the Cisco Meraki Dashboard and all the back-end services that enable the provisioning and management of the devices that connect to the dashboard. Even though the word *dashboard* generally connotes the web-based GUI, Cisco often refers to the whole back-end platform that enables the UI as Cisco Meraki Dashboard.

Devices in Cisco Meraki Dashboard are grouped in a hierarchy to help

organize customer networks and simplify management. To begin, a customer creates an organization in Meraki Dashboard. An organization typically encompasses all the networks managed by a single customer, but customers may choose to create multiple organizations if they wish to further subdivide management and control of their environment.

Organizations are then further subdivided into networks. An administrator then adds devices to the network. A network typically contains all the managed devices in a single physical location.

There are different types of networks, and each network type has some restrictions on what can be placed in the network. An administrator can create a combined hardware network that allows administrators to add security appliances, wireless access points, switches, and more to the same network. This network type is the most commonly used. There are also options to create networks that support only a single type of device, such as a wireless network or security appliance network. These options primarily exist for legacy reasons before the advent of combined networks.

Within a network, there are restrictions on the number of a device type that can be placed into the network; for example, a network can contain only a single security appliance (or a pair if configured for high availability).

[Figure 13-3](#) shows how devices are grouped into networks, which are then grouped into organizations.

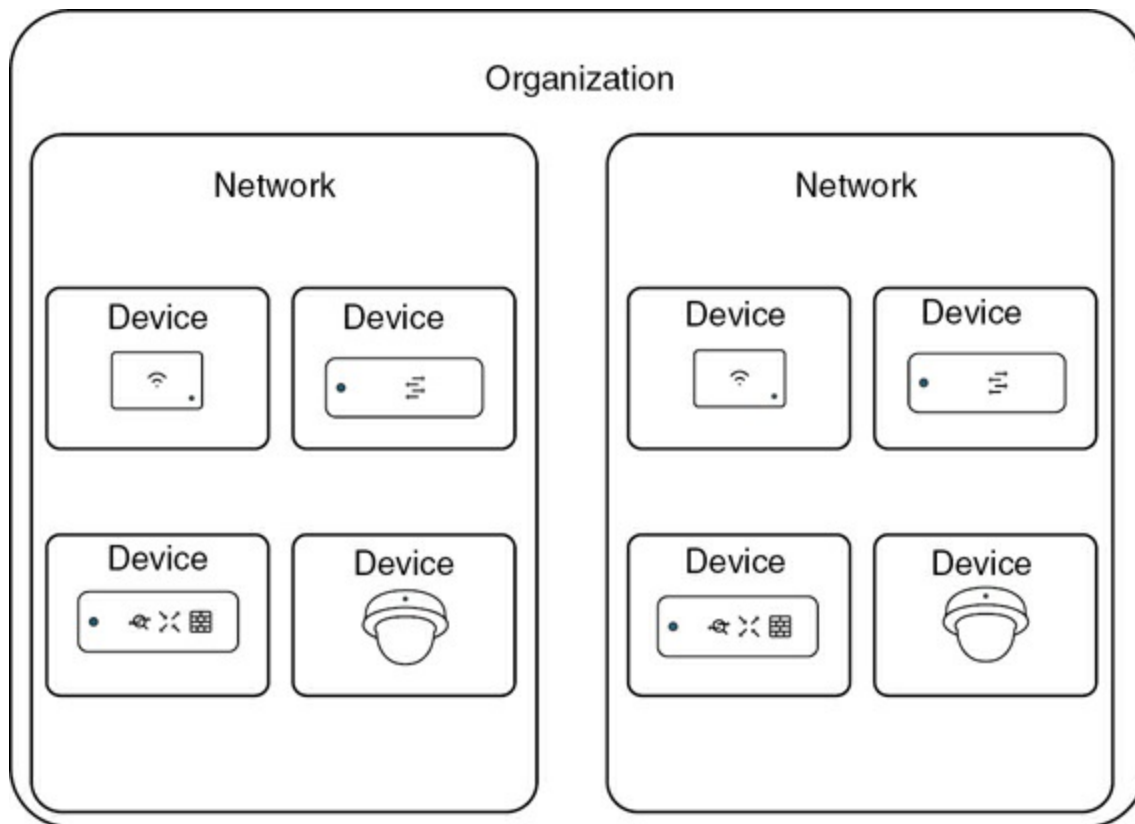


Figure 13-3 Organizational Hierarchy in Meraki Dashboard

This structure presents several advantages. First, administrators can delegate administrative privileges to other administrators but restrict their ability to manage only specific networks. This approach allows for delegation of responsibilities to staff managing a location without having to provide access to the entire organization. Another advantage is that configuration for networks can be performed using templates. This is a powerful feature for customers who have many networks that all have similar configurations—for example, a customer with hundreds or thousands of branches or retail locations. Templates allow a common configuration to be applied uniformly across all networks using a template while still providing variables to modify specific parameters that change between locations, such as IP address allocations.

For example, [Figure 13-4](#) shows the template configuration for IP addressing on a VLAN in a network.

Modify VLAN

Dual-stack Config

Subnetting

Same Unique

IPv4 Config

VLAN interface IP

Auto-generated

Subnet

/24 10.1.0.0/16

IPv6 Config Enabled Disabled

WAN 1

Auto Disabled

WAN 2

Auto Disabled

Independent

Auto Disabled

Back Preview

Figure 13-4 Modify VLAN Configuration in Meraki Dashboard

As you can see in this figure, a network can be configured to either use the same IP subnet for each network that inherits from the template, or a subnet can be autogenerated as sites are added. In this example, the subnetting has been set to Unique, and then the IP address is configured to allocate a /24 network from the /16 block 10.1.0.0/16. This type of template configuration enables you to easily add new networks in a consistent fashion while also allowing you to run site-to-site VPN connections between sites. Other parameters can be similarly templated with the platform automatically allocating the next available item in a set as new sites are added.

Meanwhile, common configurations like firewall rules, wireless network security settings, switch configurations, and SD-WAN settings remain the same across all the sites. Furthermore, if a configuration change is needed—say an update to a firewall rule—a change on the template automatically applies that change across all networks using that template.

Let's walk through the variety of devices supported by Meraki Dashboard and how SaaS enables administrators to easily provision, manage, and

troubleshoot the devices in their network.

Security Appliances (Cisco Meraki MX)

One of the most crucial devices in a Cisco Meraki network is the security appliance, designated by the MX model prefix. Security appliances operate as both routers and stateful firewalls, typically deployed at the edge of a network to connect that network to the Internet or other private WANs. They form the foundation for secure connectivity and SD-WAN within the Cisco Meraki architecture. Security appliances can be paired for high availability (HA), but typically no more than a single active node or HA pair can exist within a given Cisco Meraki network. This architectural choice limits certain complex edge designs but makes the platform exceptionally well suited for organizations with numerous distributed sites like branches, retail locations, or even remote worker setups, each requiring a secure gateway. Having only a single entry point to a network because of this limitation significantly restricts the feature set needed on these devices and therefore simplifies configuration options available to administrators.

Cisco Meraki offers a wide variety of MX security appliance hardware models, scaling to meet diverse performance needs. Smaller models are designed for small branches or remote workers. Larger models scale up significantly to handle the demands of large campus or branch environments. Despite the hardware differences affecting performance and scale, the core feature set available across these platforms remains largely consistent. This consistency is a direct result of the cloud-delivered software capabilities managed via the Cisco Meraki Dashboard. New features and security updates are pushed from the cloud, ensuring that even older hardware can benefit from the latest advancements until the hardware reaches Cisco's announced end-of-life and no longer receives newer updates. This approach demonstrates the SaaS model's power to deliver evolving, complex services like advanced security and SD-WAN through centrally managed software updates.

Cisco Meraki security appliances function as Unified Threat Management (UTM) devices, integrating a comprehensive suite of security services beyond basic routing and firewalling:

- **Next-Generation Firewall (NGFW):** Provides stateful inspection from Layer 3 through Layer 7, enabling application-aware firewall policies based on traffic type rather than just ports and protocols. Geo-IP based firewall rules allow for policies based on the geographic location of source or destination IP addresses.
- **Intrusion Detection and Prevention (IDS/IPS):** Incorporates an engine that monitors network traffic for signatures of known threats and anomalous behavior. Based on configured policies (detection or prevention), it can alert on or actively block malicious traffic. Security events are visualized within the Meraki Dashboard's security center GUI.
- **Advanced Malware Protection (AMP):** Integrates Cisco's Secure Endpoint technology, leveraging the cloud-based Cisco Threat Grid for sophisticated malware analysis and sandboxing. This allows the MX to identify and block known malware and analyze unknown files to determine their threat level. This capability relies on up-to-date cloud intelligence that can protect against recently discovered attacks automatically.
- **Content Filtering:** Enables policy-based blocking or allowing of websites based on category (e.g., social media, gambling) or specific URLs.
- **VPN Services:** Supports both remote access VPN for individual users (using native IPsec or Cisco AnyConnect client VPN) and site-to-site VPN for connecting networks. Site-to-site VPN can be established using standard IPsec to non-Meraki devices or via Meraki's powerful and easy-to-use AutoVPN feature, which uses the cloud to facilitate connectivity between sites automatically.

Beyond these core UTM features, Meraki MX appliances deliver a full suite of SD-WAN capabilities, designed to optimize connectivity and application performance across multiple WAN links:

- **Multiple Uplink Support:** MX devices support various types of WAN connections, including standard Ethernet Internet links, private WAN connections like MPLS, and cellular backhaul. Some models have integrated cellular modems, whereas others can utilize external Meraki

MG cellular gateways (discussed later). The platform supports WAN link balancing to distribute traffic across active links and automatic WAN failover to maintain connectivity if a primary link goes down.

- **Dynamic Path Selection:** The MX can intelligently route traffic over the most appropriate WAN path based on real-time link performance (latency, jitter, packet loss) and predefined policies configured in the Dashboard. Policies can be based on application type (using Layer 7 identification) or custom traffic definitions (e.g., source/destination IP, port, VLAN). If a preferred path degrades, traffic can be dynamically rerouted to a healthier link without the need for any manual intervention.
- **Application Optimization:** The platform provides Layer 7 application visibility and allows for traffic shaping rules to prioritize critical applications (like VoIP or business applications) and limit bandwidth for noncritical traffic. Global bandwidth limits per client can also be enforced.
- **Performance Monitoring:** The Meraki Dashboard provides detailed visibility into WAN uplink performance, tracking key metrics like latency, packet loss, and jitter using configurable uplink statistics probes. Machine learning algorithms and Smart Thresholds are used to analyze this data and identify genuine application performance degradation, reducing alert noise. This allows administrators to quickly assess the health of their WAN connections across all sites. Cisco Meraki can also integrate with Cisco ThousandEyes for even more comprehensive performance monitoring.
- **Multi-Uplink AutoVPN:** AutoVPN tunnels can be established simultaneously over multiple active WAN uplinks, allowing for load sharing and resilience even within the VPN fabric. Flow preferences within SD-WAN policies determine which uplink is preferred for specific VPN traffic flows.

One of the most powerful features of the Cisco Meraki SD-WAN solution is AutoVPN. This feature dramatically simplifies the creation of secure site-to-site VPN tunnels between Meraki MX appliances within the same organization. Traditionally, setting up IPsec VPNs involves complex manual configuration of parameters like encryption algorithms, pre-shared keys (PSKs), and peer IP addresses on both ends of the tunnel. AutoVPN replaces

this with a simple, cloud-orchestrated process.

The Meraki cloud acts as a central broker, specifically using a service called the VPN Registry. When an MX appliance is configured for AutoVPN in the Cisco Meraki Dashboard (simply by designating it as a hub or spoke), it securely registers its WAN IP address(es) and NAT traversal information with the VPN Registry. The registry maintains a dynamic table of all participating MX devices in the organization. When tunnels need to be built, the cloud automatically negotiates the necessary IPsec parameters (using modern standards), securely exchanges security keys, and pushes the required VPN routes to the peer devices. This entire process leverages the existing secure TLS-based communication channel between the MX and the Meraki cloud, which we will discuss later in this chapter.

AutoVPN supports both hub-and-spoke and full mesh topologies, configured via the Meraki Dashboard. In a mesh configuration, tunnels are automatically established between all participating hubs, creating a resilient and scalable network fabric that allows sites to communicate directly. When a new site is added, all existing sites automatically establish a direct connection to the new site. AutoVPN is self-healing because it automatically detects and adapts to changes like dynamic WAN IP address updates or uplink failures, reestablishing tunnels as needed. Keepalive messages sent between MX devices and the VPN Registry help to quickly detect disconnected peers. This automation significantly simplifies VPN deployment for Meraki-only environments. Establishing VPNs to non-Meraki devices still requires traditional manual IPsec configuration.

To further enhance security, Cisco Meraki MX appliances integrate seamlessly with Cisco Umbrella, Cisco's cloud security platform discussed in [Chapter 9, "Cisco Umbrella and Cisco AI Defense."](#) The MX devices have built-in SD-WAN connectivity optimized for tunneling traffic to Umbrella's global data centers. This integration extends security beyond the capabilities of the on-premises appliance by leveraging Umbrella's cloud-delivered services, including

- DNS-layer security for blocking malicious domains before connections are established
- A Secure Web Gateway (SWG) for inspecting web traffic and enforcing

acceptable use policies

- A cloud-delivered firewall (CDFW) for consistent policy enforcement
- Cloud access security broker (CASB) functionality to discover and control cloud application usage
- Advanced features like SSL decryption/inspection for visibility into encrypted traffic, data loss prevention (DLP) to protect sensitive information, and remote browser isolation for mitigating web-based threats

The integration with Cisco Umbrella allows organizations to layer cloud-based security on top of the MX's on-premises capabilities, providing defense-in-depth managed through the unified Meraki Dashboard.

In addition to the physical MX security appliances, Cisco Meraki offers a virtualized version dubbed the vMX. The vMX appliances are available in three sizes—small, medium, and large—and can be deployed on a variety of cloud platforms including AWS, Azure, Google Cloud, and others. These virtual appliances allow customers to easily manage their cloud networks in the same way they manage on-premises infrastructure. vMX appliances are a key component to the network infrastructure in the Cisco Meraki platform itself.

Switches (Cisco Meraki MS)

Complementing the routing and security capabilities of the MX series, the Cisco Meraki MS switches provide cloud-managed access and aggregation layer switching. Like the MX, the MS portfolio encompasses a wide range of hardware models, but the management experience is unified through the Cisco Meraki Dashboard.

As with all Cisco Meraki products, all configuration, monitoring, and troubleshooting tasks are performed through Dashboard. This capability eliminates the need for traditional CLI access for most operations, which may seem foreign to customers accustomed to managing their switching infrastructure from the command line. The cloud-delivered management capabilities provide many benefits over traditionally configured switches, including

- **Zero-Touch Provisioning:** Like security appliances, switches can be claimed and preconfigured in the Cisco Meraki Dashboard using their serial number even before they are physically deployed. When a switch is connected to the network and powers on, it automatically contacts the Meraki cloud, downloads its configuration, and becomes operational. This process drastically simplifies deployment, particularly for organizations with many remote sites or limited on-site IT expertise.
- **Centralized Visibility and Control:** Meraki Dashboard provides a single pane of glass to view the status of all switches and ports across the organization, regardless of their physical location.
- **Simplified Firmware Updates:** Firmware updates are delivered seamlessly via the cloud. Administrators can schedule update windows or allow automatic updates, ensuring switches are kept current with the latest features and security patches with minimal administrative effort.
- **Configuration Templates and Cloning:** Consistent configurations can be applied across multiple switches or sites using templates or by cloning the configuration of an existing switch.

Cisco Meraki switches support a feature called virtual stacking. Unlike traditional physical stacking, which requires dedicated stacking cables and co-located switches, virtual stacking allows administrators to manage potentially thousands of switch ports across the entire organization as if they were part of a single logical stack within the Dashboard, even across different switch models and geographic locations. This capability simplifies configuration management at scale. For example, an administrator can search for all ports configured for VoIP phones across multiple buildings and apply a QoS policy change to all of them simultaneously with just a few clicks. Although virtual stacking provides scalable management scalability, it does not provide the high-bandwidth, low-latency backplane interconnectivity of physical stacking. For customers requiring the advantages of physical stacking, Cisco Meraki also supports traditional physical stacking on specific switch models allowing multiple switches (typically up to eight) in a single wiring closet to be interconnected via dedicated stacking cables, forming a single logical switch with aggregated backplane capacity and providing redundancy. The two features are not mutually exclusive, so a physical stack can participate in a virtual stack along with other switches.

Port configuration in Meraki allows administrators to select individual ports or use the search and filtering capabilities of virtual stacking to select multiple ports across the network for bulk configuration changes. Port-level security is enforced through Access Policies, which can include 802.1X authentication (integrating with RADIUS servers), static MAC address allow lists, or sticky MAC learning to restrict access to authorized devices. [Figure 13-5](#) shows the Switch Configuration page for a Cisco Meraki MS switch.

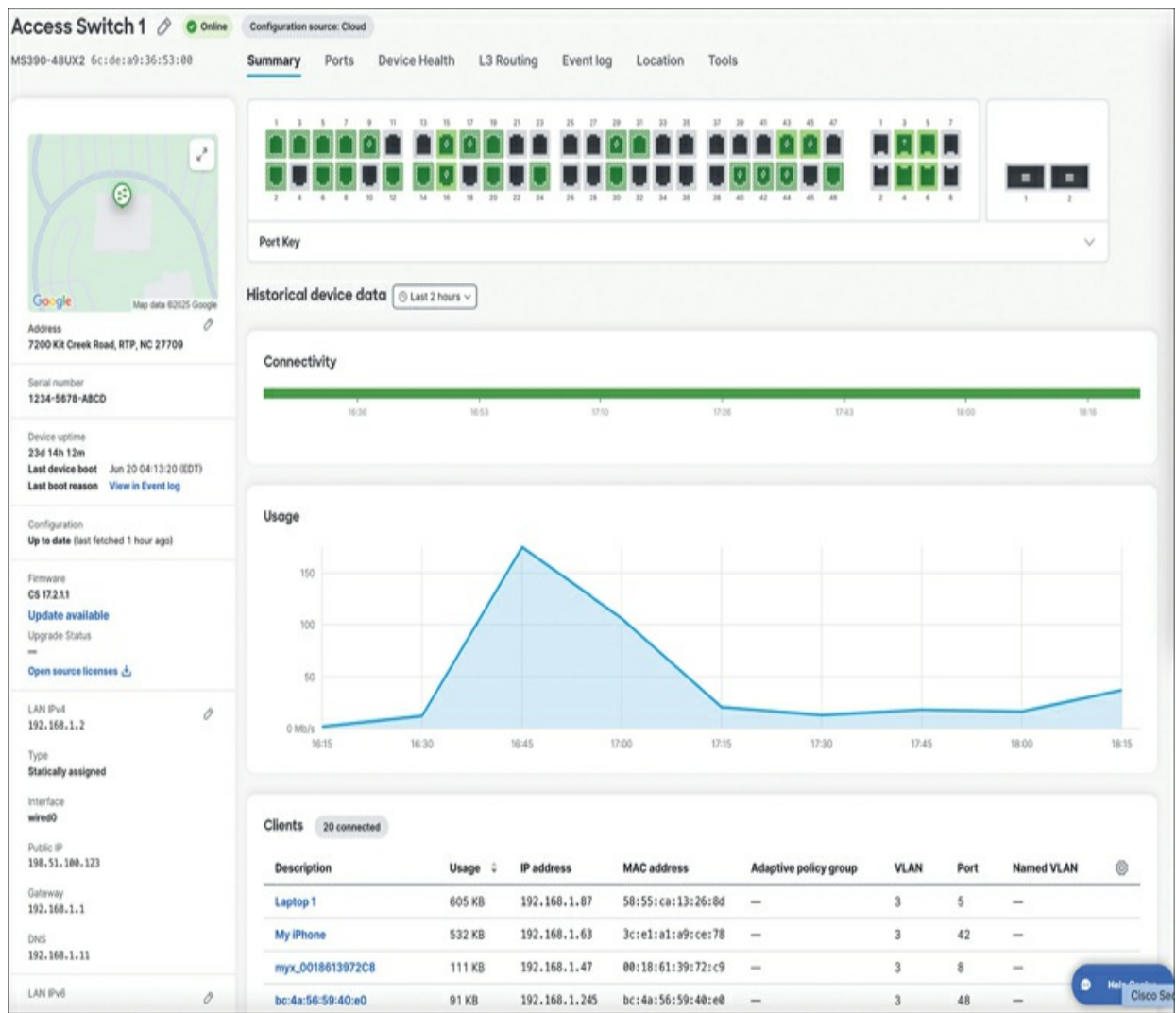


Figure 13-5 Configuring a Cisco Meraki MS Switch in Cisco Meraki Dashboard

You can see how information about the switch ports is available at a glance, offering link status, PoE status, and connectivity metrics along with usage and clients connected to the switch. Clicking on a port allows administrators

to easily drill down into specific metrics and configuration of the port.

Meraki MS switches offer a comprehensive set of Layer 2 features suitable for most access layer deployments, including VLAN support, Spanning Tree enhancements for loop prevention, quality of service (QoS) tagging and prioritization for voice and video traffic, DHCP snooping for security, and access control lists (ACLs) for both IPv4 and IPv6 traffic filtering. For networks requiring routing capabilities, specific MS models provide Layer 3 functionality. They can also provide DHCP server or relay services and support the Virtual Router Redundancy Protocol (VRRP) for first-hop redundancy (providing a warm spare gateway).

The MS hardware portfolio caters to various deployment needs from compact, fanless switches suitable for small offices or quiet environments to high-density 48-port switches for wiring closets, and even ruggedized models for harsh environments. MS switches also provide a variety of uplink and access port configuration options from 1G all the way to 100G ports. Many also have support for Power over Ethernet (PoE) for powering Cisco Meraki MR wireless access points, IP phones, and Cisco Meraki MV cameras or other PoE infrastructure.

Recognizing that some organizations have significant investments in Cisco Catalyst hardware or require specific features unique to the IOS XE operating system, Cisco has enabled integration between Catalyst switches and the Meraki cloud. Specific Catalyst switches can run the Meraki operating software natively and be fully managed by the Meraki Dashboard, offering a familiar hardware platform with the benefits of cloud management. Over time, more Catalyst switches will support being fully managed by Cisco Meraki dashboard, including devices running IOS XE.

A significant advantage of the cloud management model is the availability of remote troubleshooting tools directly within Cisco Meraki Dashboard. These tools allow administrators to diagnose and often resolve issues without needing physical access to the switch. Some of the troubleshooting and diagnostic features include

- **Live Tools:** Meraki Dashboard provides real-time status information for switches and ports, including connectivity status, traffic statistics, power draw, and connected client details. CDP and LLDP neighbor

information can help identify directly connected devices like APs or phones.

- **Remote Cable Tests:** Administrators can initiate a cable test on any copper port directly from the Dashboard. This test checks the integrity of the connected cable pairs and provides an estimated length, helping to quickly diagnose physical layer problems like faulty cables or unterminated pairs.
- **Remote Packet Capture:** One of the most powerful remote tools is the built-in packet capture capability. An administrator can select one or more switch ports and initiate a packet capture remotely. The captured traffic can be viewed directly within the Dashboard's web-based viewer, downloaded as a standard PCAP file for analysis in tools like Wireshark, or saved to the Meraki cloud for later review.

Wireless Access Points (Cisco Meraki MR)

Meraki started as a wireless networking company, creating the first cloud-managed wireless networking platform. The MR series of access points (APs) is therefore a key piece of the overall Cisco Meraki product portfolio and one that has evolved significantly over the years. Unlike some other wireless networking solutions, MR access points eliminate the need for traditional on-premises wireless LAN controllers (WLCs), simplifying deployment, management, and scaling of wireless networks. The cloud provides centralized configuration, monitoring, firmware updates, and advanced control features. This controller-less architecture, where APs tunnel management traffic to the cloud but switch data traffic locally, is well suited for distributed environments and scales from small businesses to large enterprise deployments with thousands of APs. For environments where a WLC is needed, the Cisco Meraki Dashboard also supports management of Cisco wireless networks that use a WLC.

Meraki consistently incorporates the latest wireless standards into its MR portfolio. Most enterprise-grade MR APs feature multiradio designs, commonly including 2.4 GHz, 5 GHz, and potentially 6 GHz client-serving radios, along with a dedicated scanning radio. This scanning radio plays an important role in continuous RF environment monitoring for security

(WIDS/WIPS) and automated radio frequency (RF) optimization.

Optimizing the radio frequency environment is critical for wireless performance. Cisco Meraki provides sophisticated cloud-based RF optimization techniques that leverage the power of the cloud to deliver capabilities to the devices on-premises, including

- **Auto RF:** Using data collected by the dedicated scanning radio, the Meraki cloud automatically adjusts parameters like channel assignments and transmit power levels for each AP to minimize interference and maximize coverage. This process runs continuously, adapting to the dynamic nature of the RF environment.
- **AI-RRM (Artificial Intelligence Radio Resource Management):** This capability represents a more advanced layer of RF optimization, available with specific access points and license tiers. Unlike traditional RRM that uses snapshot data, AI-RRM leverages machine learning and analyzes long-term, historical RF data trends from the specific network and potentially anonymized data from the broader Meraki ecosystem. This capability allows the AI engine to make more intelligent decisions tailored to the network's unique patterns. Key features include
 - **AI Channel Planning:** AI-RRM learns which channels are frequently impacted by interference or dynamic frequency selection (DFS) events and avoids them proactively.
 - **Flexible Radio Assignment (FRA):** In dense deployments, it can intelligently disable redundant 2.4 GHz radios based on historical coverage and interference data to improve overall performance.
 - **Busy Hour Optimization:** AI-RRM identifies the network's peak usage times based on historical client activity and minimizes disruptive RF changes during these periods, enhancing stability. The goal of AI-RRM is to improve user connectivity (e.g., reduced co-channel interference, better signal-to-noise ratios) while simplifying administration. The move toward AI-driven optimization showcases how the SaaS model facilitates the delivery of complex, computationally intensive algorithms as a service, enhancing the performance of the underlying hardware.

Figure 13-6 shows how the Cisco Meraki Dashboard allows administrators to visualize the RF spectrum as seen by a given access point in the network and troubleshoot issues related to channel interference.

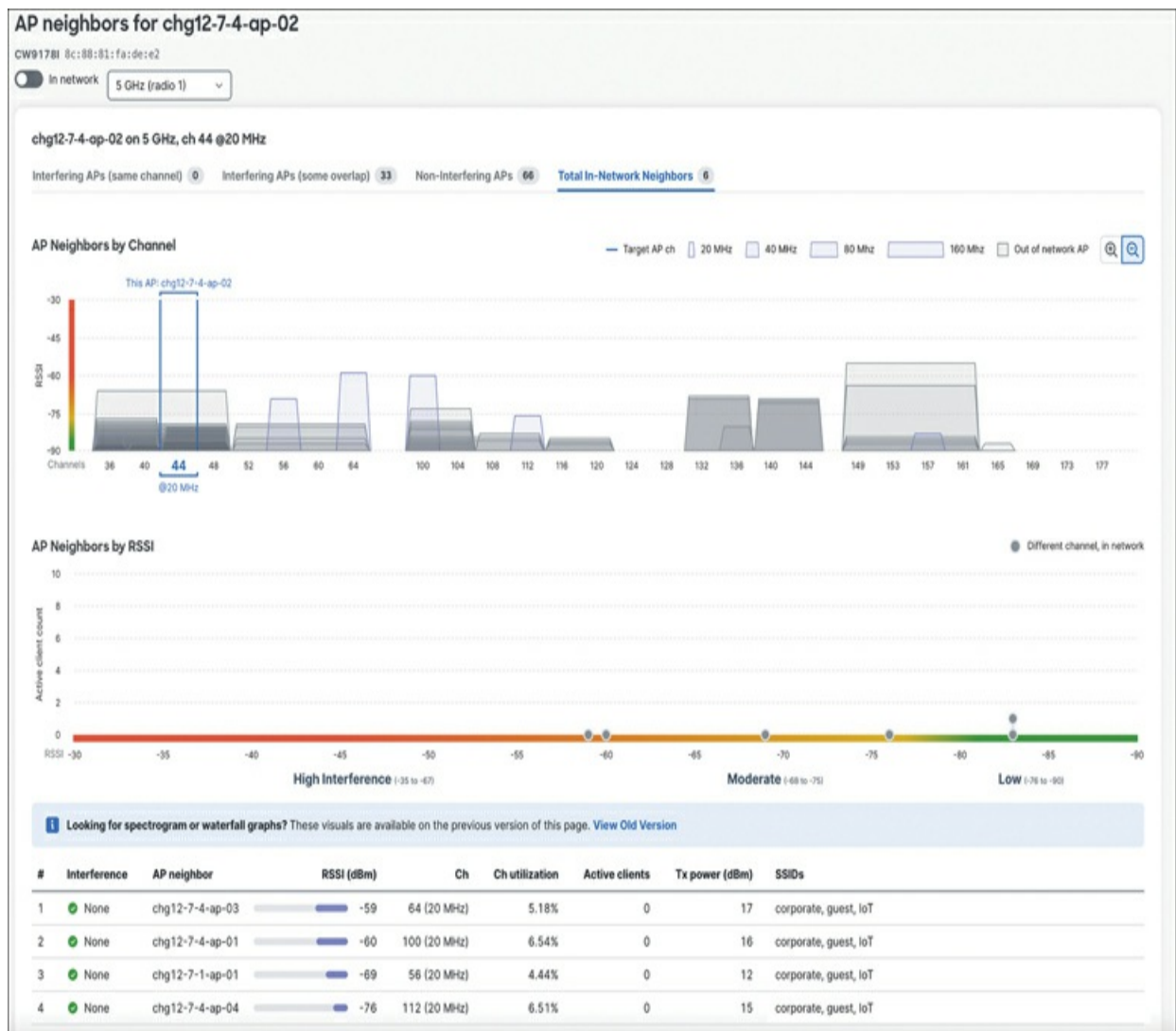


Figure 13-6 Wireless Access Point Network Neighbor Data in Cisco Meraki Dashboard

Meraki MR APs incorporate multiple layers of security features, all configured and managed via Cisco Meraki Dashboard:

- **Encryption and Authentication:** Meraki MR APs provide support for the latest WPA3 security standards for encryption, while maintaining backward compatibility for WPA2. Users can be authenticated with pre-shared keys or via enterprise authentication using 802.1X/EAP, which

can optionally be integrated with RADIUS servers or Active Directory for per-user credentials.

- **Air Marshal:** The dedicated scanning radio powers Air Marshal, Cisco Meraki's integrated wireless intrusion prevention system (WIPS). It continuously scans for threats like rogue access points, malicious broadcasts, and spoofing attacks, providing configurable containment options.
- **Guest Access:** Secure, isolated guest networks can be easily configured with customizable splash pages for user authentication or terms acceptance. Built-in firewall rules ensure guests can access only the Internet, not internal network resources.
- **Integrated Firewall:** APs feature a built-in Layer 7 stateful firewall engine, allowing administrators to apply traffic shaping and firewall rules directly at the AP level based on application type, helping to control bandwidth usage and enforce security policies at the wireless edge. These policies are enforced directly on the access point without any requirement for an MX security appliance.
- **Systems Manager Sentry Integration:** As discussed later, Sentry allows dynamic network access policies based on the security posture of connecting devices managed by Meraki Systems Manager. This includes automatically provisioning secure EAP-TLS Wi-Fi configurations (Sentry Wi-Fi) and potentially quarantining noncompliant devices.

Modern access points increasingly serve as integrated IoT platforms. Many models include a dedicated IoT radio supporting Bluetooth Low Energy (BLE) and sometimes other standards like 802.15.4 (the basis for Zigbee and Thread). Some models also incorporate ultra-wideband (UWB) radios for high-precision indoor location services. This integrated IoT capability allows the MR AP to act as a gateway for various IoT devices, most notably the Meraki MT environmental sensors, collecting their data via BLE and relaying it to the Cisco Meraki Dashboard. This capability eliminates the need for separate IoT gateways, simplifying deployment for smart building applications, asset tracking, location-based services, and environmental monitoring, positioning the AP as more than just a Wi-Fi device but a central connectivity and sensor hub within a physical space. This data can feed into other solutions like Cisco Spaces to provide capabilities like rich maps of

workspaces and indoor wayfinding.

Cellular WAN Gateways (Cisco Meraki MG)

To address the need for connectivity in scenarios where traditional wired options are unavailable or unreliable, Cisco Meraki offers the MG series of cellular WAN gateways. These devices connect to a cellular network (4G LTE or 5G) and provide an Ethernet connection that can be used as a WAN uplink for Meraki MX security appliances or other third-party routers and firewalls. Like all Meraki hardware, MG gateways are managed centrally through the Cisco Meraki Dashboard.

The primary use cases for MG cellular gateways revolve around providing flexible and resilient WAN connectivity:

- **Primary WAN:** In locations lacking broadband infrastructure, or where deployment speed is critical (e.g., temporary sites, pop-up retail, construction), an MG can serve as the primary Internet connection.
- **Backup/Failover WAN:** Perhaps the most common use case is providing a secondary, backup WAN connection for an MX appliance. If the primary wired connection fails, the MX can automatically fail over to the cellular link provided by the MG, ensuring business continuity.
- **SD-WAN Uplink:** An MG can provide an additional diverse path for SD-WAN traffic, allowing the MX to route traffic over cellular based on policy or performance, and enabling site-to-site VPNs (AutoVPN) to be established over the cellular network.

Earlier cellular backup solutions for MX security appliances relied on a USB device physically attached to the MX appliance. This meant making sure there was good cellular coverage in the room where the MX appliance was located or sometimes running an RF cable of some kind to extend the antenna from the modem to a better location. Having an MG gateway as a separate device from the MX offers deployment flexibility. Because cellular signal strength is highly dependent on location and antenna placement, the MG can be installed in a location with optimal cellular reception (e.g., near a window, on a roof, outdoors), while the MX remains secured in a network rack or data center. All MG gateways are IP67-rated, allowing them to be placed

outdoors.

The cloud management aspect via the Meraki Dashboard simplifies the deployment and ongoing operation of cellular WAN. Features include

- **Zero-Touch Provisioning:** Like other Meraki devices, MGs can be added to the Dashboard and will automatically configure themselves upon connection.
- **Monitoring:** The Dashboard provides real-time and historical visibility (up to 30 days) into key cellular metrics, including signal strength (RSSI), signal quality (RSRP, RSRQ, SINR), SIM status, data usage, and the gateway device the MG is connected to. This is crucial for troubleshooting connectivity issues.
- **Configuration:** Failover behavior (when used with an MX) and basic network settings (NAT mode vs. Passthrough mode) are configured through the Dashboard. Passthrough mode allows the downstream router (MX) to receive the public IP address directly from the cellular carrier, while NAT mode provides a private IP address to the downstream device.

Cameras (Cisco Meraki MV)

Cisco Meraki MV cameras provide a cloud-managed video surveillance platform that offers a variety of capabilities not available in traditional video surveillance systems. The MV platform makes use of both cloud computing as well as edge processing on the MV devices to provide a high-performance solution with the ease of cloud management. The core innovation lies in its edge storage architecture. Instead of relying on centralized network video recorders (NVRs) or video management systems (VMS) to store footage, most Meraki MV cameras feature high-endurance solid-state drives (SSDs) directly onboard the camera. Video is recorded and stored locally at the edge and then retrieved by the cloud as needed.

This distributed storage model offers various benefits:

- **Elimination of NVRs/VMS:** This model reduces hardware costs, complexity, and points of failure associated with traditional recording

servers.

- **Simplified Installation and Scaling:** Adding cameras automatically adds storage capacity, simplifying system design and expansion. Installation is easier without needing to connect cameras back to a central recorder over the network for storage.
- **Enhanced Security:** This model removes the NVR as a potential network vulnerability target. Video data at rest on the camera is encrypted.
- **Resilience:** Cameras continue to record footage locally even if the network connection to the cloud is temporarily lost.
- **Bandwidth Efficiency:** Network bandwidth is primarily consumed only when footage is actively being viewed remotely, rather than constant streaming to an NVR.

Figure 13-7 depicts the MV architecture where video is stored locally on devices and can be streamed locally or remotely.

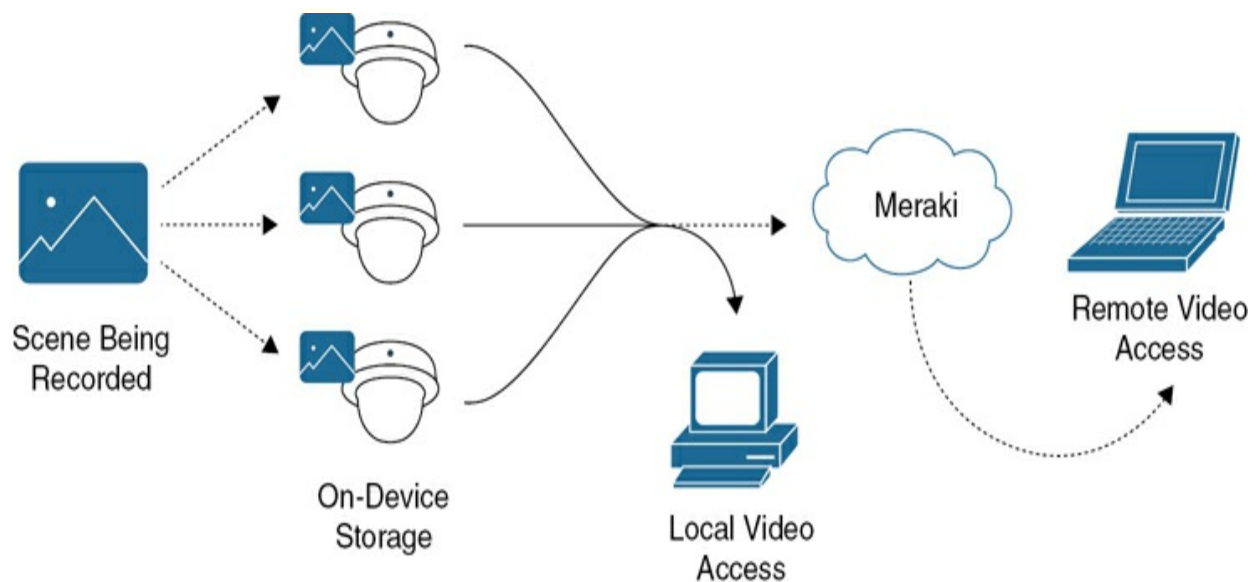


Figure 13-7 Cisco Meraki MV Camera Architecture

While storage is primarily at the edge, the management and access are entirely cloud-based, facilitated by the Cisco Meraki Dashboard and the Meraki Vision portal. Administrators can configure cameras, manage storage settings, view live and recorded video, and access analytics from anywhere

using a web browser or the Meraki mobile app. The Meraki cloud automatically proxies video streams, enabling seamless remote viewing without complex firewall configurations. Granular access controls allow administrators to define precisely who can view which cameras (live or recorded) and whether they can export footage, with detailed audit logs tracking user activity. The Vision portal also allows for the creation and arrangement of custom video walls for monitoring multiple feeds simultaneously.

Beyond simple recording, MV cameras are equipped with powerful AI processing capabilities, enabling onboard video analytics powered by machine learning and computer vision executed directly at the edge. This edge processing allows for analysis without overwhelming network bandwidth or requiring powerful central servers. Built-in analytics capabilities include

- **Object Detection:** Identifying people and vehicles within the frame
- **Motion Detection:** Recording and alerting based on standard motion
- **Motion Heatmaps:** Visualizing patterns of movement over time
- **Attribute Search:** Searching recorded footage based on detected attributes, such as the color of a person's clothing or type of vehicle, dramatically speeding up investigations
- **People Counting:** Estimating the number of people in an area for occupancy analysis
- **Audio Detection:** In some models, detecting specific audio patterns, such as smoke or carbon monoxide (CO) alarms

These analytics are accessed and visualized through the Meraki Dashboard or the Vision portal. Additionally, the platform supports Custom Computer Vision, allowing organizations or partners to deploy their own trained machine learning models onto the cameras. This capability opens possibilities for highly specific analytics tailored to unique business needs, such as detecting personal protective equipment (PPE) compliance, identifying specific objects on a manufacturing line, or analyzing customer behavior in retail environments.

The MV platform is designed for integration. A key native integration is Sensor Sight, which links MV camera footage directly with events from Meraki MT environmental sensors. When an MT sensor (discussed in the next section) triggers an alert, the associated MV camera can automatically capture a snapshot or provide a time-stamped link to the video footage corresponding to that event. This capability provides immediate visual context for environmental alerts, enhancing situational awareness and speeding up response times. For example, security can instantly see who opened a restricted door when the sensor triggers. Setting up Sensor Sight involves simply assigning a relevant camera to one or more sensors within the Meraki Dashboard.

Beyond native integrations, MV cameras offer rich APIs and support for RTSP streaming, enabling integration with a wide range of third-party systems. In this way, video context can be embedded within access control logs (e.g., showing who badged in), building management systems, or custom applications. Analytics results, including those from custom computer vision models, can also be streamed in real-time using the Message Queuing Telemetry Transport (MQTT) protocol to external platforms for further analysis or automation triggers. This programmability transforms the MV camera from a simple security device into an intelligent, programmable sensor platform capable of feeding valuable data into broader business intelligence and automation workflows.

Sensors (Cisco Meraki MT)

Cisco Meraki MT is a recent addition to the Cisco Meraki portfolio that introduced cloud-managed environmental sensors designed to provide real-time visibility into physical conditions within IT closets, data centers, offices, classrooms, and other facilities.

The MT portfolio includes sensors for monitoring a variety of environmental parameters, including temperature and humidity; water leaks; air quality (monitoring temperature, humidity, particulate matter [PM2.5], total volatile organic compounds [TVOC], CO₂, and ambient noise); intrusion/access of open doors, cabinets, or windows; and power monitoring. A push button sensor also can be used as a trigger for automations.

As with other Cisco Meraki products, one significant advantage of MT sensors is the ease of deployment. Most of the sensors can be battery powered, lasting several years before requiring battery replacement; therefore, they require low power usage. To achieve long battery life, they communicate wirelessly using Bluetooth Low Energy. Instead of requiring dedicated sensor gateways, they leverage existing Meraki infrastructure (Meraki MR access points or MV smart cameras), which act as BLE gateways. This means environmental monitoring can be added anywhere that has connectivity to an MR or MV gateway. The MR or MV gateway receives the BLE data from the sensors and securely relays it to the Meraki cloud.

All sensor data is centrally managed and visualized within the Cisco Meraki Dashboard. The Dashboard provides real-time readings, historical data trends (with configurable time ranges), and visualizations. Sensor status can also be viewed on network topology maps or floor plans. Data can be exported for external analysis via into CSV or XLS files, retrieved via API, or streamed via the MQTT protocol. The system includes a variety of alerting mechanisms based on configurable thresholds for each sensor reading. When a threshold is breached, notifications can be sent immediately via email, SMS text message, push notifications to the Meraki mobile app, or programmable webhooks to trigger actions in external systems. [Figure 13-8](#) shows the Psychrometric chart in Cisco Meraki Dashboard that provides a single view of all temperature and humidity sensor data in the network.

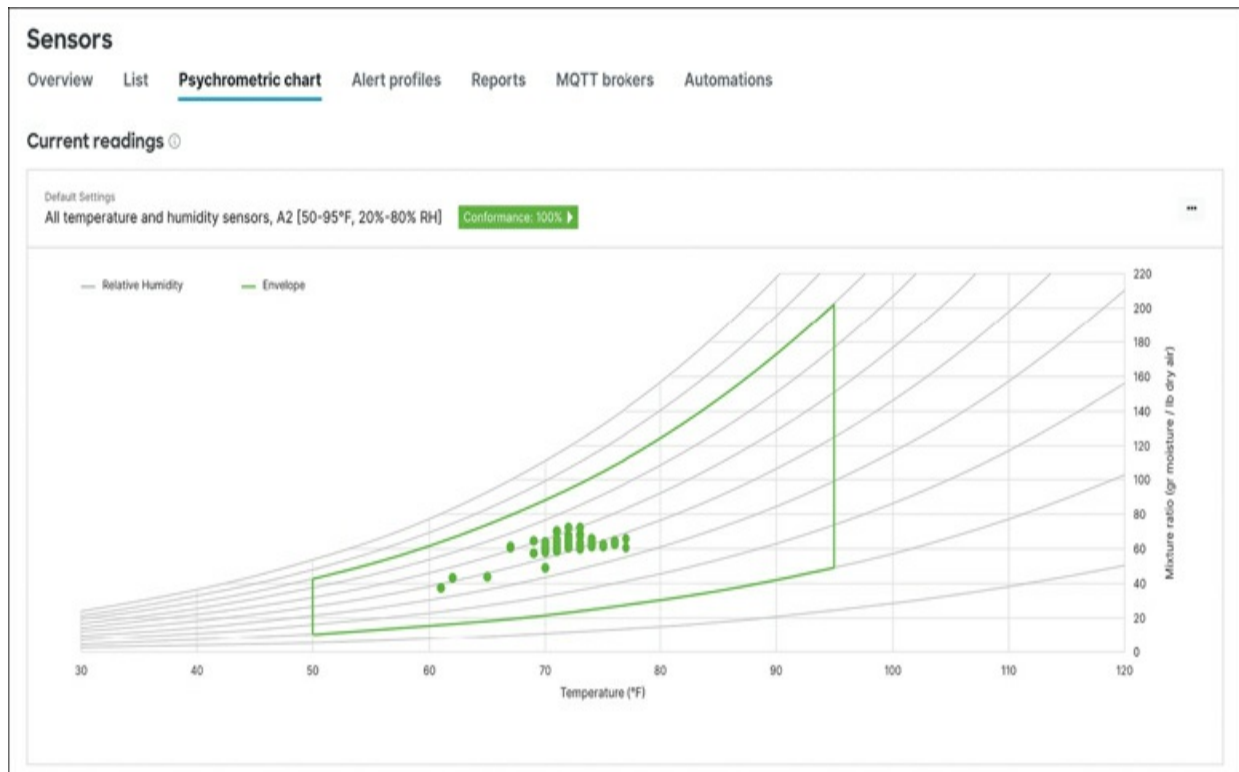


Figure 13-8 Psychrometric Chart of Cisco Meraki MT Sensor Data

The MT platform offers several integration and automation capabilities:

- **Sensor Sight:** As previously mentioned, this native integration links MT sensor events with MV camera footage, providing visual context for alerts.
- **Automation Builder:** This tool allows administrators to create simple automation workflows triggered specifically by the smart button sensor. A button press can trigger actions such as sending custom notifications, triggering MV camera snapshots, enabling/disabling a wireless SSID, toggling switch ports on/off, or sending a custom webhook payload.
- **Webhooks:** Alerts from any MT sensor can trigger webhooks, sending data to external URLs. This enables integration with IT service management (ITSM) platforms (e.g., creating tickets in ServiceNow), messaging platforms (e.g., Webex), building management systems, or custom applications.
- **MQTT:** For real-time data streaming, MT sensors can publish detailed telemetry data to an MQTT broker located on the customer's network or

in the cloud. MQTT is a lightweight publish/subscribe protocol ideal for IoT applications. Subscribing clients can receive sensor readings along with timestamps almost instantaneously. This enables use cases like real-time dashboards, integration with industrial control systems, or data feeds into custom analytics platforms.

- **API:** The Meraki Dashboard API can also be used to retrieve sensor readings and configure settings programmatically.

Device Management (Meraki Systems Manager)

Beyond managing the network infrastructure itself, the Meraki platform extends its cloud management capabilities to the endpoints connecting to the network through Meraki Systems Manager (SM). Systems Manager is a cloud-based unified endpoint management (UEM) solution that provides centralized visibility, configuration, and security for a wide range of devices, including smartphones, tablets, laptops, and desktops, all managed from the same Cisco Meraki Dashboard used for network devices. This integration offers the potential for simplified operations, particularly for organizations already invested in the Meraki ecosystem, by providing a single console for managing both the network and the devices using it.

Systems Manager is the exception to other capabilities in the Cisco Meraki platform in that it is a pure SaaS offer with no dependency on Cisco Meraki hardware. Even so, instead of managing Cisco Meraki devices, SM manages third-party endpoints that typically belong to individual users.

Systems Manager provides core UEM capabilities across several functional areas:

- **Mobile Device Management (MDM):** This forms the foundation, enabling administrators to provision device settings and enforce restrictions, maintain an inventory of managed devices, track device location, and perform remote actions like locking or wiping devices (either fully or selectively removing only corporate data and applications). For iOS devices, supervision provides deeper control over restrictions and configurations. Remote troubleshooting capabilities, including remote desktop viewing for certain platforms, are also included.

- **Mobile Application Management (MAM):** SM facilitates the deployment and control of applications. Administrators can push public applications from app stores (Apple App Store, Google Play) as well as privately developed enterprise applications. It provides an enterprise app store for users; supports managed app configuration to preconfigure settings within apps; integrates with volume purchasing programs like Apple VPP; and allows application inventory, blacklisting, or whitelisting. Simple containerization options help separate work and personal data within apps.
- **Mobile Content Management (MCM):** This capability focuses on secure distribution and access to corporate documents and files. Features include the Backpack capability for delivering files to Android devices with synchronization, integration with enterprise file sync and share (EFSS) services, configuration of Per-App or Always-On VPN settings for secure resource access, and policies to control data leakage via copy/paste restrictions or email attachment controls.
- **Mobile Identity (MI):** This capability uses device context to dynamically apply configurations and policies. Based on factors like geolocation, time of day, user group membership, assigned tags, device type, or security posture, SM can automatically add or remove applications, adjust settings, or trigger network policy changes via Sentry integration.

Systems Manager boasts broad cross-platform support, capable of managing devices running iOS, iPadOS, macOS, Windows, Android, ChromeOS, and even tvOS. Various enrollment methods are available to suit different deployment scenarios, including manual enrollment via agent installation or profile download, automated enrollment methods like Apple's Automated Device Enrollment (ADE, formerly DEP) and Android Zero-Touch Enrollment for corporate-owned devices, Apple Configurator for bulk iOS enrollment and supervision, Active Directory GPO for Windows agent deployment, and user self-service portals.

A primary function of any UEM solution is endpoint security policy enforcement. Systems Manager allows administrators to define and deploy granular security policies across managed devices. Common policies include enforcing strong passcodes, requiring device-level data encryption, setting

screen lock timeouts, restricting device features (like camera or app store access), and checking for minimum OS versions or detecting compromised devices (jailbroken/rooted). SM continuously monitors devices for compliance with these policies and can automatically apply tags or trigger remediation actions if a device falls out of compliance. For Windows endpoints, it can manage BitLocker encryption, and for iOS, it integrates with the Cisco Security Connector (CSC) app for enhanced security visibility and control.

Systems Manager provides tools to securely support bring-your-own-device (BYOD) initiatives. Administrators can enforce essential security settings like passcodes and data encryption on personal devices accessing corporate resources. The selective wipe capability is particularly important for BYOD, allowing IT to remove only corporate applications and data if an employee leaves or a device is lost, without impacting personal files and photos. Integration with Meraki networking devices allows for automatic identification (fingerprinting) and differentiated network access policies for BYOD devices versus corporate-owned ones.

The most powerful aspect of Systems Manager, particularly within the context of the Cisco Meraki platform, is the Sentry feature. Sentry deeply integrates endpoint posture information from SM with network access policies enforced by Cisco Meraki MR access points and MX security appliances. This feature enables dynamic, context-aware security that goes beyond static firewall rules or VLAN assignments. Key Sentry components include

- **Sentry Enrollment:** Streamlines onboarding by redirecting unmanaged devices connecting to a specific "enrollment" SSID to the SM enrollment portal.
- **Sentry Wi-Fi:** Automatically provisions secure WPA2/WPA3-Enterprise Wi-Fi profiles, including unique EAP-TLS client certificates, to SM-enrolled devices connecting to designated SSIDs on MR access points. This component provides strong, certificate-based authentication without manual configuration or reliance on less secure pre-shared keys.
- **Sentry VPN:** Similarly automates the configuration of client VPN settings (for connecting to an MX appliance) on enrolled devices,

simplifying secure remote access.

- **Sentry Policies:** Allows administrators to link SM tags (which can be assigned manually or dynamically based on security posture, location, schedule, and so on) to specific network group policies on MR or MX devices. If a device's tag changes (e.g., it becomes noncompliant and gets a Quarantined tag), the network infrastructure automatically enforces the corresponding policy (e.g., assigns the device to a restricted VLAN, applies specific firewall rules or traffic shaping). This creates a dynamic access control system where network privileges are continuously adjusted based on the real-time trustworthiness and compliance of the endpoint.

The Sentry feature set converges endpoint management and network security to enable more sophisticated, automated, and context-aware security postures, aligning well with Zero Trust security principles. While Systems Manager provides a broad range of UEM capabilities, its primary strength lies in this tight integration with the Meraki network infrastructure, offering unified visibility and powerful cross-platform automation possibilities managed from a single SaaS dashboard.

The Cisco Meraki Cloud

Cisco Meraki delivers all the capabilities we have discussed up to this point using a scalable, resilient cloud platform that has evolved over the years. As with the other products and platforms we have discussed, we will discuss how the various components of the Cisco Meraki cloud fit into the overall SaaS architectural model described in [Chapter 2](#), “[SaaS Architecture](#),” and shown in [Figure 13-9](#).

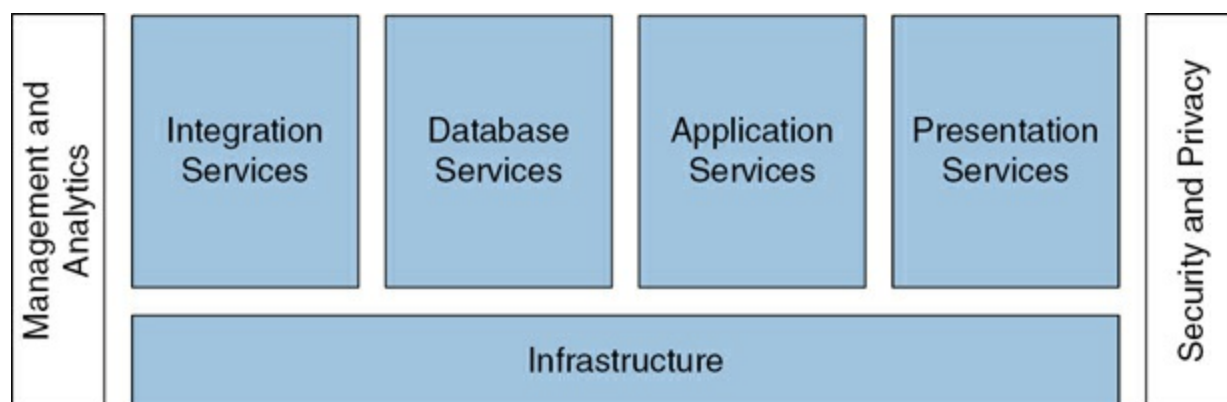


Figure 13-9 SaaS Architectural Model

Infrastructure

The Cisco Meraki platform is hosted in a variety of global data centers and public clouds. The platform is a fully multitenant cloud platform that ensures segregation of customer data while allowing workloads to run on shared compute resources. Meraki provides a 99.99% uptime service-level agreement (SLA) for the cloud platform. Remember, though, that Meraki devices can generally run autonomously if they lose connectivity to the Meraki cloud, so any downtime of the platform primarily affects the ability to provision and monitor the network. User traffic is typically not impacted in any way.

To achieve these levels of availability, data is replicated across three independent data centers to ensure no loss of customer data in the event of an outage. Additional backups for disaster recovery scenarios are also maintained to two independent third-party cloud storage services to ensure diversity of the backups. Cisco Meraki is constantly evolving their data center footprint to ensure they comply with a variety of data sovereignty and other regulations that might dictate where customer data is allowed to be stored. [Figure 13-10](#) shows the locations of the various global Cisco Meraki data centers at the time of this writing.

cloud providers, taking a multicloud approach that leverages several public cloud providers alongside Cisco's private data centers. This approach allows the development and operations teams to select the optimal environment for each specific workload, a decision driven by a clear set of business and technical requirements.

The pull toward public cloud is driven by several factors. First is the need to meet customers where they are, particularly concerning the increasingly strict landscape of data sovereignty and compliance laws, which often require customer data to reside within a specific country or region. Public cloud providers offer a global footprint that would be prohibitively expensive and time-consuming to replicate with private data centers. Second, latency can be a critical factor for some use cases, and deploying services in public cloud regions closer to end users can significantly improve performance. Third, the agility of the public cloud allows Meraki to spin up new infrastructure for development, testing, and production workloads in minutes, in contrast to the long lead times and supply chain dependencies associated with building out physical data centers.

Although the use of public cloud has increased, the private data centers remain strategically important. They provide a necessary hosting option for customers who, for legal, contractual, or policy reasons, cannot or will not have their data hosted in a public cloud. Furthermore, they provide an alternative in regions where major public cloud providers do not yet have a presence, ensuring the Meraki cloud can serve a global customer base.

In the early days, the Meraki platform was built as a largely monolithic application running on bare metal servers, but over time, this configuration has evolved into a more modern microservices-based approach, as we will discuss in the next section. This shift has imposed the need to modernize the infrastructure layer to support these new workloads. The modern Meraki platform makes heavy use of containerized workloads that are deployed using Kubernetes. Kubernetes allows for workloads to be easily moved between clouds (either public or private) as required.

The Cisco Meraki platform employs Infrastructure as Code (IaC) principles to manage its complex environment. Terraform is used extensively to automate the provisioning of infrastructure across public clouds and its Kubernetes clusters. This methodology ensures that environments are built in

a repeatable and documented manner. Automation platforms like Ansible are also used to automate configuration management using a declarative model.

To seamlessly and securely connect these disparate environments, the Cisco Meraki cloud leverages its own networking technologies, creating a single, manageable network that spans on-premises customer sites, private data centers, and multiple public clouds.

Application Services

Many applications services are responsible for various aspects of managing and monitoring Cisco Meraki products. As we have discussed previously, the core of the Meraki platform is the Meraki Dashboard, so the application services that provide Dashboard services comprise most of the application services.

While the Meraki platform started with monolithic application services, the platform has evolved over time to a microservices-based architecture, facilitating better agility in creating new features while sustaining existing services.

Meraki's application architecture provides a textbook example of the evolution from a monolithic application to a hybrid architecture that strategically incorporates microservices, so we will discuss how this evolution has occurred to highlight how other SaaS providers might evolve their cloud platforms over time.

The original Meraki Dashboard platform was built as a monolithic application, with its core written in Ruby on Rails and supported by various other applications written in C++ and Scala, all running on Linux servers. In the early years of the company, this was an effective choice given the state of the industry at the time.

As the scale of the platform grew, the monolithic architecture began to present challenges with the complexity of the single codebase slowing down productivity and agility. The response to these challenges was to begin a thoughtful transition toward a microservices architecture encompassing a multiyear journey. [Figure 13-11](#) provides a high-level depiction of the services that comprised the Cisco Meraki platform before the transition to

containerization and microservices.

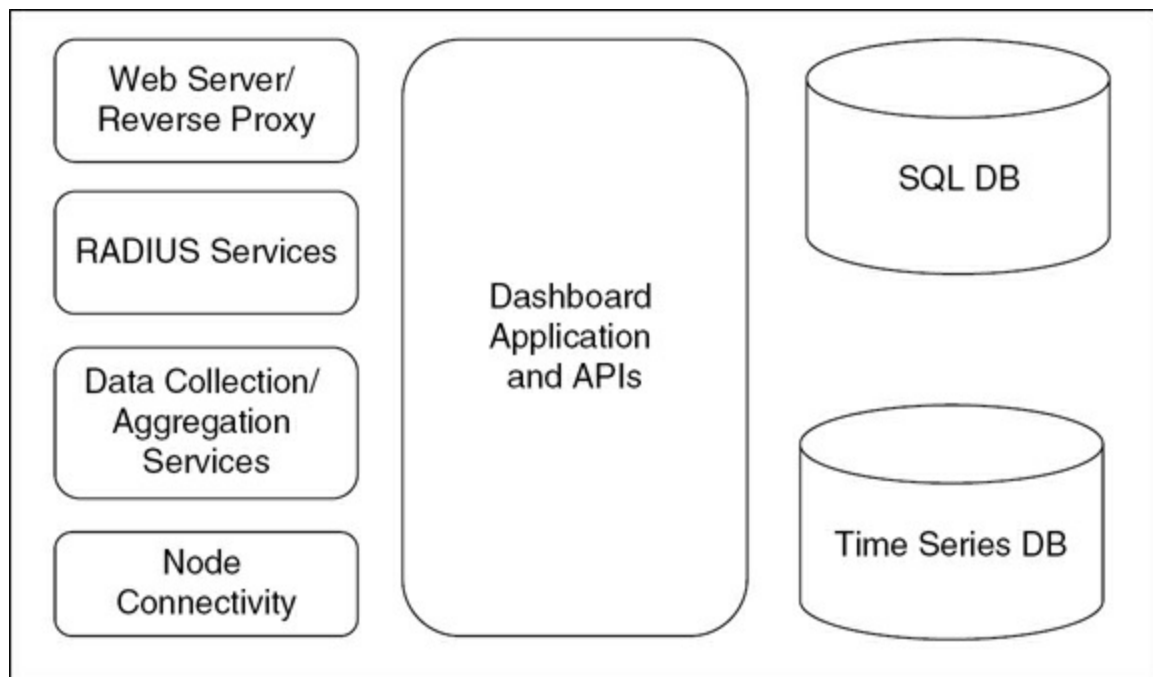


Figure 13-11 High-Level Depiction of the Services in the Cisco Meraki SaaS Platform

If you focus on the application services in [Figure 13-11](#), you can see that most of the functionality provided by the Meraki Dashboard are provided by a single, monolithic application.

The first step to evolve this architecture was to take the existing monolith application and break it down logically by containerizing different components. While still part of the same application, these services could be run in their own isolated containers. This approach provided immediate and significant benefits without requiring a full rewrite. It allowed teams to experiment with and deploy updates to one part of the application without fear of breaking another, and it limited the "blast radius" of any potential failure, thus improving reliability. This step also made security patching faster and more targeted.

The team then took an approach where new functionality was built from the ground up as independent microservices instead of being added to the existing monolith. For example, when a new service to handle secure connectivity between Cisco Meraki devices and the cloud was being built, it

was built as a separate microservice.

One of the inherent advantages to leveraging a microservice architecture is that different microservices can be written using different technologies. For example, this connectivity service was written in Go (Golang), a language selected specifically for its high performance and concurrency features, which were ideal for this network-intensive application, something that would not have been possible if built as part of the original monolith.

Over time, the application started to look more like what is depicted in [Figure 13-12](#). (These representative service names are not intended to be a precise description of the services in the Cisco Meraki platform.)

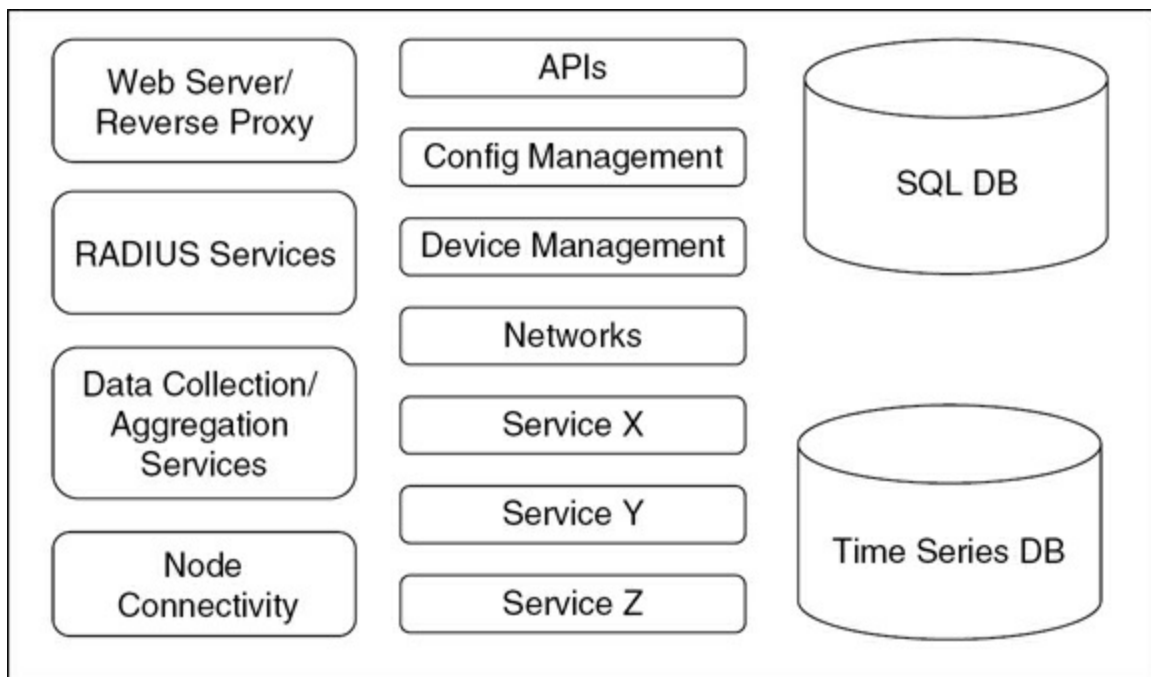


Figure 13-12 Depiction of Meraki Dashboard Application Broken Up into Several Services

Once the application services were broken up into separate containerized microservices, this led to the next evolution to push these microservices into a Kubernetes environment. This approach allows the Cisco Meraki platform to easily scale out and provides additional levels of resiliency and availability that were not possible when services were running directly on physical servers. While in the past all the services shown in [Figure 13-12](#) may have been located on the same server, as the platform evolves, those services run

in a Kubernetes environment that allows for flexibility in how the services are deployed.

Presentation Services

The Cisco Meraki platform provides presentation services to two distinct “users.” Meraki Dashboard is a web-based front-end user interface that customers use to interact with the platform, but the services running in the cloud also must interact with the millions of devices that are relying on the cloud for management and monitoring. These are two very distinct services that the platform must provide, and therefore, two different sets of presentation services are required to facilitate these connections.

For the Meraki Dashboard web-based GUI, the Cisco Meraki platform leverages industry standard web services to provide the interface for user browsers to connect to the platform. Because of the sharded architecture of the Cisco Meraki platform, there are services responsible for authenticating users, then directing them to the application services that manage the organization and network that the administrator is attempting to manage. This architecture is all largely transparent to the end user other than the fact that the browser URL can change as an administrator navigates from one organization to another.

The more interesting component that sits at the presentation layer is those services that provide connectivity for the customer devices being managed by the Cisco Meraki platform. To achieve this connectivity, Cisco Meraki built a proprietary, lightweight, and highly secure communication channel that connects every Meraki device back to the cloud. It is not a standard HTTPS connection but a purpose-built tunnel with several key characteristics that make it suitable for securely managing devices at scale.

All management data transiting the tunnel is encrypted using strong AES-256 encryption. The protocol is extremely lightweight, consuming only about 1 kilobit per second (kbps) of bandwidth per device during normal operation when no active configuration changes are being made. It uses efficient data serialization formats like Google Protocol Buffers to minimize overhead. Also, this interface functions like a WebSocket connection in that devices establish a TCP connection to the Cisco Meraki cloud and that persistent

connection allows the cloud to send commands or requests to the network device, even if the device is using a private IP address behind a NAT.

When an administrator makes a configuration change in the Meraki Dashboard UI, this action updates the device's configuration data stored in the back-end database. The Meraki cloud then sends a command through the secure tunnel to push the new configuration down to the physical device, where it is applied. In the reverse direction, the device sends a constant stream of telemetry and analytics data up to the cloud via the tunnel, which is then processed and visualized in the Dashboard's graphs and charts.

Figure 13-13 shows how an event-driven remote procedure call (RPC) engine brokers transactions between the customer device and the various services that might need to send or receive data from the device. This architecture provides an abstraction layer where services do not need to concern themselves with how to talk to the device.

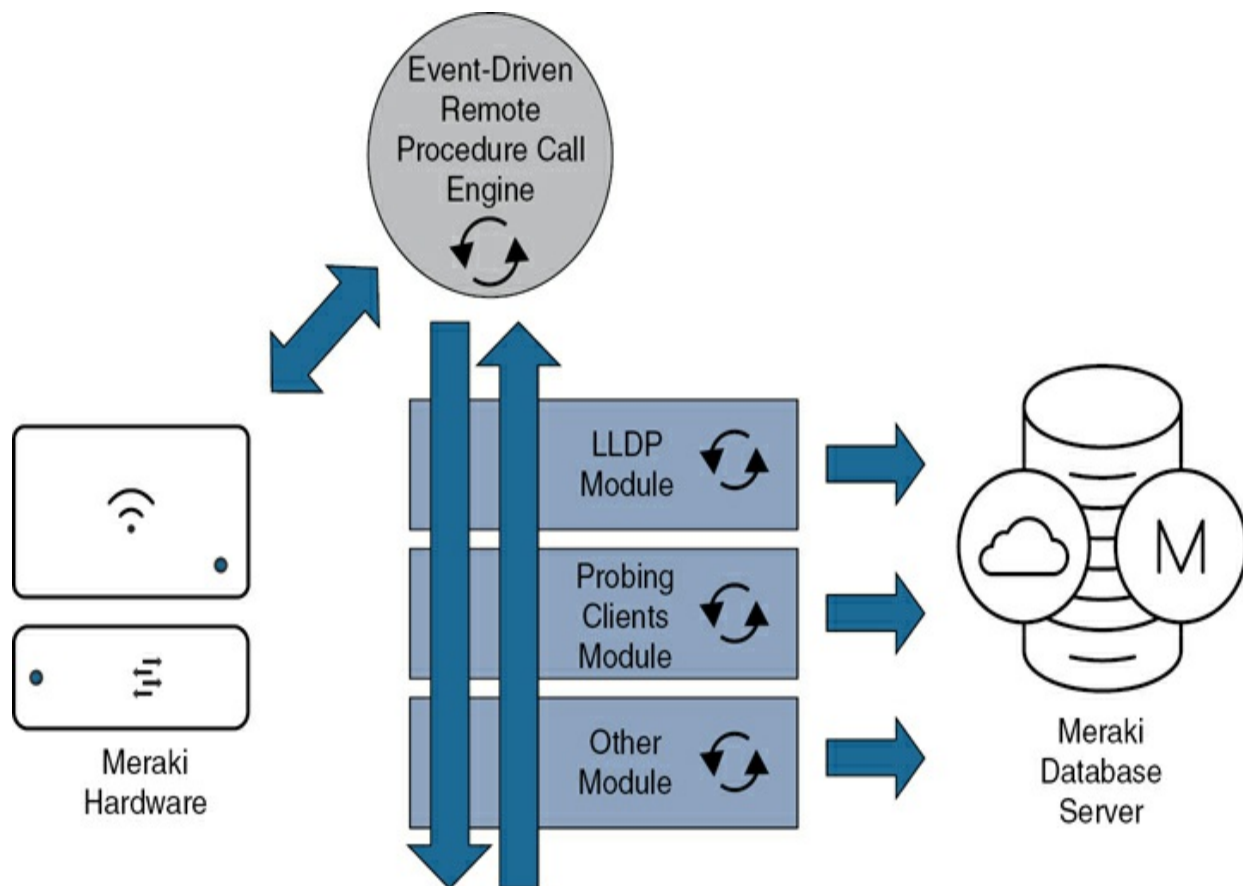


Figure 13-13 Event-Driven Remote Procedure Call Architecture

If a Meraki device were to lose its connection to the cloud, local network traffic, Internet access, and existing security policies would continue to function uninterrupted. From a scalability perspective, this design means Meraki only needs to scale its infrastructure to handle the relatively small management and analytics packets, not the potentially massive volume of data flowing through its customers' networks. That said, with the increasing need to send and process more telemetry from customer networks, the demands on these services continue to increase to meet this need.

Database Services

The Database Services layer is responsible for the persistent storage and retrieval of all application and user data. The Meraki platform employs a multitiered data storage strategy. It utilizes a combination of standard relational databases, a proprietary time-series database, and modern public cloud data services. This purpose-built fabric is designed for extreme scale, reliability, and performance, and it forms the foundation for the platform's intelligence capabilities.

Meraki's data architecture is segregated based on the type of data and its specific access patterns, ensuring that each component is optimized for its task.

- **Configuration Data:** For core network and device configuration data, Meraki relies on a highly available, replicated SQL relational database. This type of data requires the strong transactional consistency and structured schema that a relational database provides.
- **Time-Series and Analytics Data:** Cisco Meraki devices continuously send runtime data such as device performance, client connection changes, and other real-time metrics. This data is stored in a distributed metrics database optimized for high-throughput, append-only writes of time-series data. It achieves its performance by clustering data by both timestamp and a hierarchical key, allowing for extremely fast retrieval of recent data for dashboard visualizations.
- **Public Cloud Data Services:** As the need for more data analytics has grown, the Cisco Meraki platform has increasingly relied on public cloud data services to complement its existing systems. For new, large-

scale data science and machine learning initiatives, the platform leverages cloud storage services for building cost-effective data lakes and high-performance data warehousing.

While not specifically a database service, a critical component of the Cisco Meraki architecture is its data transport layer, which facilitates the movement of data between systems. Message bus technology such as Apache Kafka provides a highly reliable and scalable conduit to transport massive volumes of telemetry data from the core applications and devices (often originating in Meraki's private data centers) to the analytics platforms that may be running either in a private data center or in the public cloud.

This architecture provides several key benefits: It decouples the data producers from the data consumers, allows for data to be buffered and processed asynchronously, and enables data streams to be routed to multiple destinations simultaneously. This ingest pipeline is what allows Meraki to collect and analyze billions of data points to power its next generation of intelligent features.

Across all tiers, data durability and availability are paramount. All customer data, including network configurations in the SQL database and historical statistics in the time series database, is replicated across at least three independent, geographically dispersed data centers in near real time. This approach ensures that, in the event of a catastrophic failure of an entire data center, the platform can fail over to a replica with up-to-date data, a process that is transparent to the end user and is fundamental to delivering the platform's 99.99% uptime SLA.

Integration Services

Integration services are the building blocks that allow a SaaS platform to connect with other applications, automate workflows, and become part of a larger business ecosystem. Cisco Meraki provides a set of powerful application programming interfaces (APIs) that enable an open, extensible platform for automation and innovation. Many functions that are available from the Cisco Meraki Dashboard GUI can be performed through the Cisco Meraki APIs.

The Meraki Dashboard API is a modern, RESTful interface that uses standard HTTPS requests and returns data in the human-readable JSON format. The API is built on the OpenAPI Specification, a standard framework for defining RESTful APIs. The use of this framework ensures the API documentation is always accurate and up to date with the latest platform capabilities. It also allows developers to use a vast ecosystem of standard tools, such as generating a Postman collection for easy testing or automatically generating client-side SDKs in various programming languages.

Beyond the general-purpose Dashboard API, Meraki offers a suite of specialized APIs designed to unlock the unique value of its advanced hardware, particularly its wireless access points and smart cameras:

- **Location Scanning API:** This API provides a real-time stream of Wi-Fi location data from Meraki MR access points. It can deliver the latitude, longitude, and x/y coordinates of detected Wi-Fi devices, enabling a rich ecosystem of third-party applications for retail analytics, foot traffic analysis, visitor engagement, and loyalty programs.
- **MV Sense API:** Meraki's MV smart cameras perform machine learning-based object detection and classification at the edge (directly on the camera). The MV Sense API makes this data, such as people counts and movement tracking, accessible to external applications. In this way, businesses can integrate video intelligence into their operations, for use cases ranging from queue management to workspace optimization.
- **Captive Portal API (EXCAP):** This API allows developers to create highly customized and branded guest Wi-Fi splash pages. This capability moves beyond simple network access to create rich user experiences, integrate with social media platforms for marketing, and capture valuable user analytics.

To support real-time, event-driven workflows, the Meraki platform also supports webhooks. Instead of requiring an application to constantly poll the API to check for changes, webhooks allow an administrator to configure the Dashboard to send an automatic HTTP POST notification to a specified URL whenever a particular event occurs, such as a device going offline or a security event being detected.

While these integration services provide significant value to customers who make use of them, they also enable other Cisco products to integrate directly with the Cisco Meraki platform. For example, Cisco Webex provides an integration option where Webex administrators can obtain information about network devices when troubleshooting media quality issues in Webex Control Hub. This integration significantly reduces the amount of time needed to diagnose a problem by automatically correlating data between the two platforms and is enabled by the APIs exposed through the integration layer.

The Cisco Meraki platform has a similar integration with the Cisco ThousandEyes platform, allowing administrators to get a fuller picture of network issues by melding the data from telemetry provided by the Cisco Meraki devices with the application layer data provided by the ThousandEyes agents running in the network. All these integrations are made possible by the integration layer services in the SaaS architectural model.

Management and Analytics

For other SaaS platforms, management and analytics is an additional layer that enables the administrator to monitor and maintain the services provided by the cloud. In the case of the Cisco Meraki cloud, management and analytics is a core function of the service that the platform provides and in many ways *is* the service being provided. Being that this is a core function of the platform, we have already discussed many of the components that enable management and analytics in the Cisco Meraki platform when discussing other services like the application services handling device configuration and management or the services responsible for collecting real-time data and storing them in time series databases.

Here are a few examples of the types of analytics and diagnostics data that the Meraki Dashboard provides, enabled by the time-series data collected by the cloud. When administrators are troubleshooting wireless issues in an enterprise environment, understanding metrics like signal strength, signal-to-noise ratio, latency, and channel utilization are key to both diagnosing reactive issues as well as proactively detecting potential design or pervasive issues. [Figure 13-14](#) shows the roaming diagnostic information for a client in Cisco Meraki Dashboard.

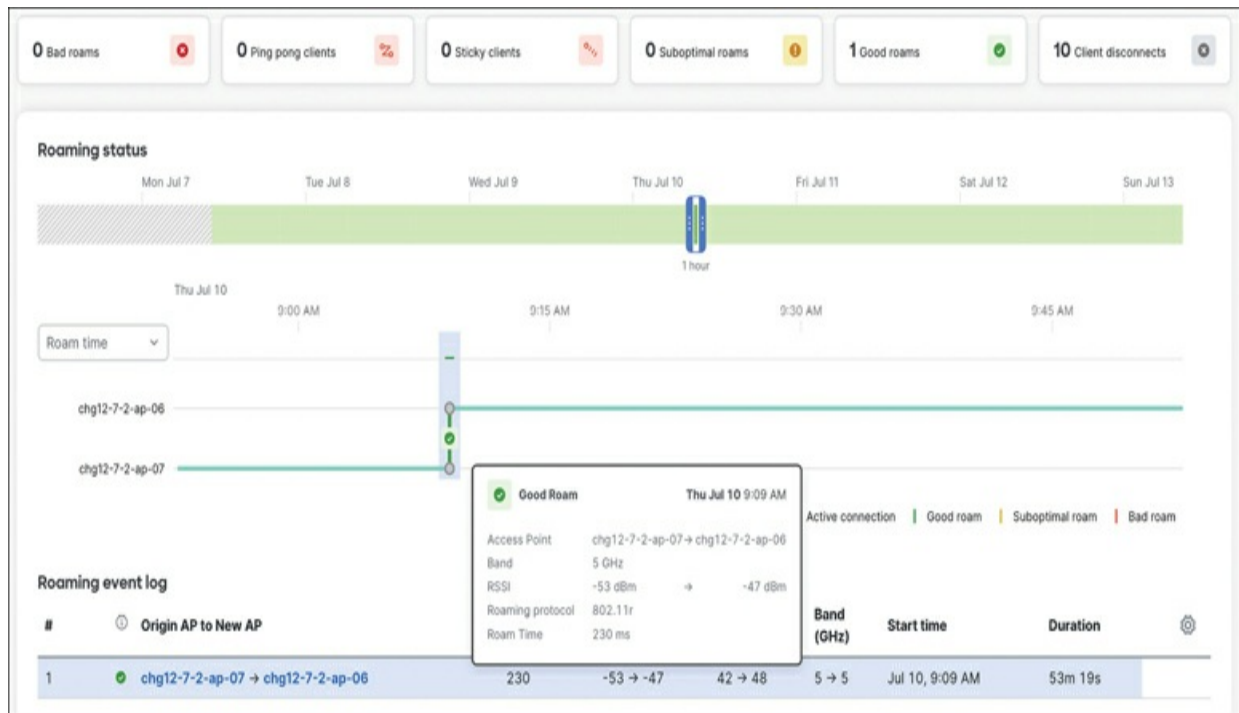


Figure 13-14 Roaming Information for a Client in Cisco Meraki Dashboard

Where in the past an administrator may have had to comb through log files on wireless LAN controllers or access points to figure out the parameters around a client roaming from one access point to another, the Meraki Dashboard is able to take the telemetry data and present it in a way that is easy to understand. To facilitate analytics like this, back-end services process and aggregate telemetry data, making it quick and easy to present to an administrator.

Another example of useful analytics is the client's view, as shown in [Figure 13-15](#).

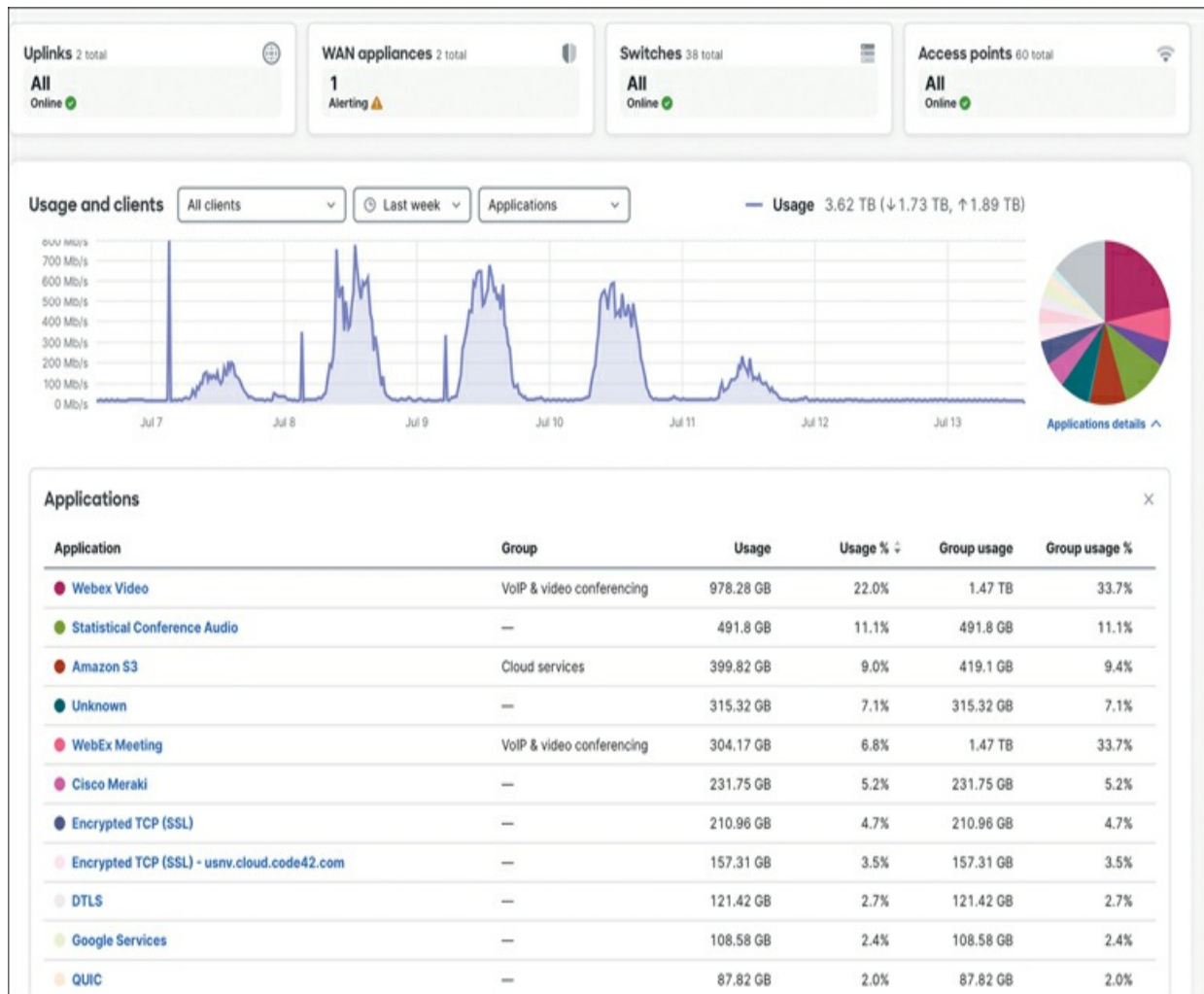


Figure 13-15 Client View in Cisco Meraki Dashboard

This view allows administrators to easily see overall network utilization over a specified period and then drill down into which workloads are consuming the most network bandwidth. As seen in Figure 13-15, 22% of traffic on this network is for Webex video. The client and usage view gives insights into overall trends and allows administrators to easily spot anomalies.

These are just a few of the examples of analytics data presented in Cisco Meraki Dashboard, all enabled by the persistent tunnel continuously providing telemetry and metrics to the cloud and the application services that consume that data and store it with the help of services at the database layer.

One area that is worth discussing is how the SaaS platform itself is monitored. As Meraki's architecture evolved to include more distributed

components like microservices and Kubernetes clusters, its internal monitoring and analytics capabilities had to evolve as well. The observability includes centralized logging, time-series data, and advanced analytics to detect faults and ensure uptime.

The platform is watched over by a 24x7 automated failure detection system that tests every server from multiple locations every five minutes. Any detected anomaly triggers a rapid escalation process across multiple operations teams to ensure swift resolution.

Security and Privacy

Security and privacy are core building blocks of the Cisco Meraki platform and manifest in two distinct ways. First, the Cisco Meraki platform helps protect customer networks by providing visibility into security-related events as well as protection and enforcement of security-related policies. For example, [Figure 13-16](#) shows how the Cisco Meraki Dashboard Security Center highlights attacks that have been detected and blocked in the network.

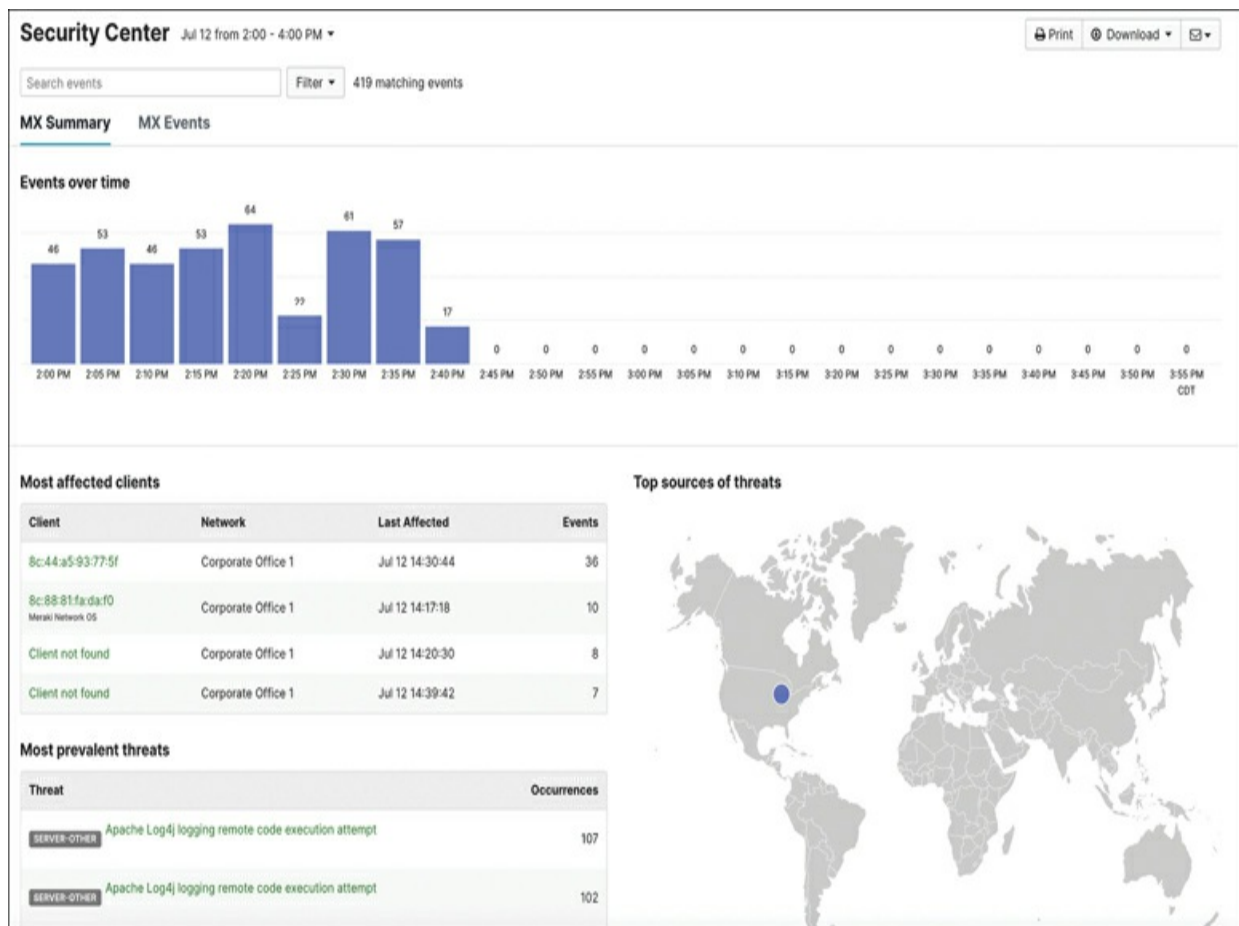


Figure 13-16 Security Center in Cisco Meraki Dashboard

Administrators can see what types of attacks are being attempted in their network through this dashboard. In this case, the Security Center shows evidence of attempts to exploit a well-documented Log4j vulnerability. Cisco Meraki security devices know about vulnerabilities like this through security intelligence provided by Cisco’s Talos organization (<https://talosintelligence.com/>). Cisco Talos has hundreds of security researchers whose role is to investigate vulnerabilities in both Cisco products as well as third parties. The security researchers then create “Snort rules” (documented publicly on [snort.org](https://www.snort.org)) and publish them through a live feed that is consumed by the Cisco Meraki network devices.

The Cisco Meraki platform provides administrators with a rich set of tools to secure their own organizations and enforce the principle of least privilege. These tools include configurable strong password policies; two-factor authentication; fine-grained role-based access control (RBAC) to limit what

different administrators can see and do; and a comprehensive, searchable audit log that records every configuration change, who made it, and when.

In addition to the security capabilities provided by the Cisco Meraki platform, the platform itself uses a multilayer approach to security across the platform architecture that extends from the physical security of its data centers to the processes used to write and deploy code.

In general, sensitive end-user data never flows through Meraki's cloud; therefore, the platform has limited exposure for eavesdropping on customer traffic. This architectural separation is critical for helping customers achieve compliance with stringent industry regulations like the Payment Card Industry Data Security Standard (PCI DSS) and the Health Insurance Portability and Accountability Act (HIPAA), which place strict controls on the handling of sensitive data.

The underlying infrastructure that runs the Cisco Meraki cloud is protected by rigorous physical and cybersecurity controls. These facilities maintain industry-standard certifications, including SOC 2, PCI, and ISO 27001.

The cloud services themselves are hardened against attack. The environment is protected by 24x7 automated intrusion detection systems and is segmented by IP and port-based firewalls. Access to network infrastructure leverages strong multifactor or passwordless authentication with role-based access, using the principle of least privilege.

As with all Cisco products, Cisco Meraki follows a strict secure development lifecycle (SDL) to ensure that all aspects of the development process are secured with appropriate checks and balances to ensure compliance. For example, security is considered from the very beginning of the development process, identifying potential vulnerabilities in the design phase before any code is written through threat modeling and secure design.

The development lifecycle further extends into the development and build pipelines where automated scanning and compliance checks protect against accidental or malicious introduction of security vulnerabilities either through poor development practices or supply chain-related vulnerabilities in third-party libraries.

Cisco's product security incident response team (PSIRT) actively engages

with the external security community through a public vulnerability reporting program and a "bug bounty" program that rewards researchers for finding and reporting security issues. This program is augmented by the threat intelligence provided by the Cisco Talos security research group.

You can find additional details on the security and privacy policies of the Cisco Meraki platform at <https://meraki.cisco.com/trust/>.

Summary

As we have illustrated in this chapter, the Cisco Meraki platform is a clear example of Software-as-a-Service (SaaS) principles applied to network hardware management. It shows how a SaaS model can deliver a complete service by hiding operational complexity behind a cloud-based application and accelerate feature and capability enablement across many large-scale networks.

The Cisco Meraki Dashboard is the core of the SaaS platform. As a multitenant, browser-based application, it provides a single interface to manage globally distributed infrastructure. As we have seen, the platform's design maps directly to the SaaS reference architecture discussed previously, incorporating a hybrid cloud architecture with a variety of services providing application, database, presentation, and integration services with security and management built in.

This architecture allows the platform to deliver on the core promises of SaaS, including continuous delivery. New features, security intelligence, and firmware updates are pushed from the cloud, ensuring the service constantly improves for all customers.

The vast amounts of telemetry collected by the platform are used to provide rich analytics in addition to the provisioning and management capabilities. The platform's extensive APIs and support for webhooks demonstrate its function as an extensible platform. The Cisco Meraki platform clearly demonstrates the primary value of the SaaS model: shifting the burden of complexity from the customer to the provider to deliver a powerful, easy-to-use service.

References

- Meraki Cloud Architecture: https://documentation.meraki.com/Platform_Management/Dashboard_Ac
- Meraki Dashboard: <https://meraki.cisco.com/en-au/products/meraki-dashboard/>
- Meraki Platform: <https://meraki.cisco.com/en-au/platform/>
- Meraki Wireless: <https://meraki.cisco.com/en-au/products/wi-fi/>
- Meraki Switches: <https://meraki.cisco.com/en-au/products/switches/>
- Meraki Security and SD-WAN: <https://meraki.cisco.com/en-au/products/security-sd-wan/>
- Meraki Cameras: <https://meraki.cisco.com/en-au/products/smart-cameras/>
- Meraki Sensors: <https://meraki.cisco.com/en-au/products/sensors/>

Chapter 14. Management: Cisco Intersight

Cisco Intersight is the culmination of a journey of innovation within the data center, particularly in the area of computing and Cisco's Unified Computing System (UCS). Many of the challenges that have presented themselves in the data center space have been around problems of scale. Over time, both Cisco's hardware and software have combined to rise to the challenge and solve those issues of scale. More than a decade ago, Cisco solved the challenge of bare-metal proliferation and server portability, where configuring each server individually was both inefficient and prone to inconsistency, with the concept of a *service profile* allowing hardware to be easily replaced and reprogrammed identically to before. The inheritance model of policies and pools within a service profile allowed UCS Manager (UCSM) domains to scale to support hundreds of physical servers within a single domain. From a hardware perspective, Cisco's virtual interface card (VIC) helped support the massive volume of virtual machines (VMs) that began running within the data center by providing virtual interfaces to combine network and storage traffic on the same physical adapter. The blade/chassis architecture also helped ease both the power/cooling requirements and physical cabling needs when adding compute capacity to the data center. UCS customers could stand up a new server within just a few minutes with zero new cabling needs in many instances.

In addition to scale, Cisco UCS strongly integrated with other components of the data center, whether through technologies like VM Fabric Extender (VM-FEX), which reduces latency to help support latency-sensitive applications such as high-performance computing (HPC), high-frequency trading (HFT), and so on; by extending the physical switch connectivity direct to the VM; or

via robust application programming interfaces (APIs) allowing for third-party integrations and monitoring systems.

Cisco Intersight has continued the trend in supporting challenges of *scale*, with the aim to reduce operational costs, simplify management, and support the web scale levels of compute and networking needed for tomorrow's data center. Because Intersight is a Software-as-a-Service offering, it scales to meet the demand of any customer size with your customers. Where UCS Manager could historically support hundreds of servers, Intersight boasts the ability to manage a virtually limitless number of servers. Intersight's API-first approach allows for countless integrations for both greenfield and brownfield deployments. With a single pane of glass to the entire compute infrastructure and AI insights into both the hardware, firmware, and software deployed in the data center, Intersight is poised to equip customers with the tooling and systems needed to tackle a scale of compute that had been previously untenable.

Intersight Overview

As a cutting-edge, cloud-based management platform, Cisco Intersight revolutionizes how IT infrastructure is monitored and managed. Designed to address the challenges of increasingly complex and distributed data center environments, Intersight brings together compute, storage, and networking management into a unified, centralized interface. By leveraging the power of a SaaS delivery model, Intersight ensures continuous access to new features, proactive security updates, and improved performance with minimal user effort. In this chapter, we will provide a comprehensive overview of Intersight, breaking down its key capabilities, deployment options, integration features, and unique advantages.

To begin, in the “[Cloud-Managed Compute](#)” section, we will explore the foundational concept behind Intersight. Here, we will explain how IT management has evolved over time, from single-server management with Cisco Integrated Management Controller (IMC) to domain-wide control with UCS Manager and UCS Central, culminating in the cloud-native architecture of Intersight. The benefits of SaaS, including always-up-to-date features, scalability, and operational simplicity, are highlighted. Additionally, we will address customer concerns about uptime, security, and control, outlining how

Intersight delivers high reliability and flexibility through its deployment modalities, including private cloud options.

Next, in “[SaaS vs. On-Prem Implementation](#),” we will delve deeper into the deployment models offered by Intersight. While the SaaS model is recommended for most customers due to its cost-effectiveness, scalability, and rapid vulnerability patching, we will also explain the use cases for on-premises options. We will introduce the connected virtual appliance (CVA) for customers who require some connectivity to Cisco and the private virtual appliance (PVA) for those needing an air-gapped solution. Comparisons between these models provide clarity on which deployment path best fits specific operational needs.

In “[Device Integration \(Cisco and Third Party\)](#),” we will focus on Intersight’s capability to unify management across a diverse range of Cisco and third-party hardware. From standalone C-series servers to UCS domains, HyperFlex clusters, and even Nexus switches, this feature ensures a single pane of glass for administrators. Additionally, Intersight Assist extends management to third-party products such as VMware vCenter, Pure Storage, and NetApp, while integration with popular platforms such as ServiceNow and Terraform showcases its flexibility for modern IT workflows.

Finally, in “[Key Features and Functions](#),” we will outline the rich feature set that makes Intersight a transformative platform. This feature set includes foundational capabilities like device inventory and compliance checks, as well as advanced automation features like firmware upgrades, compatibility analysis, and operating system installation. We will also emphasize Intersight’s API-first design, which allows for seamless programmability and extensibility. Through its subscription-based licensing model, Intersight ensures organizations can tailor their experience to meet their unique requirements while scaling effortlessly over time.

Cloud-Managed Compute

As the volume of managed servers increases, having a centralized management strategy can improve consistency, reduce operational costs, and improve visibility of issues across the entire install base. That is where Intersight and the SaaS model of cloud-managed compute enters the picture.

The SaaS journey can be an intimidating one for customers when they first embark on it. Many customers are concerned about uptime, change windows, and operational control. Although Intersight has multiple deployment modalities that include on-premises implementations that would allow full control, Cisco has taken many steps to ensure the SaaS delivery mechanism is a great choice for many customers.

The benefits of a cloud-delivered offering are plentiful. The SaaS delivery model allows customers to take advantage of a continuous integration/continuous delivery (CI/CD) strategy of deployment, resulting in always-up-to-date feature sets, no need to upgrade code, and features that are always available instantaneously with backward compatibility. That CI/CD model also means that any security concerns or potential vulnerabilities are addressed immediately and for the entire Intersight user base.

The SaaS microservices-based architecture scales infinitely horizontally. Intersight customers do not have to worry about operational load. To assuage concerns around uptime and control, Cisco publishes uptimes and any outages via <https://status.intersight.com>. Figure 14-1 shows a screenshot from this site.

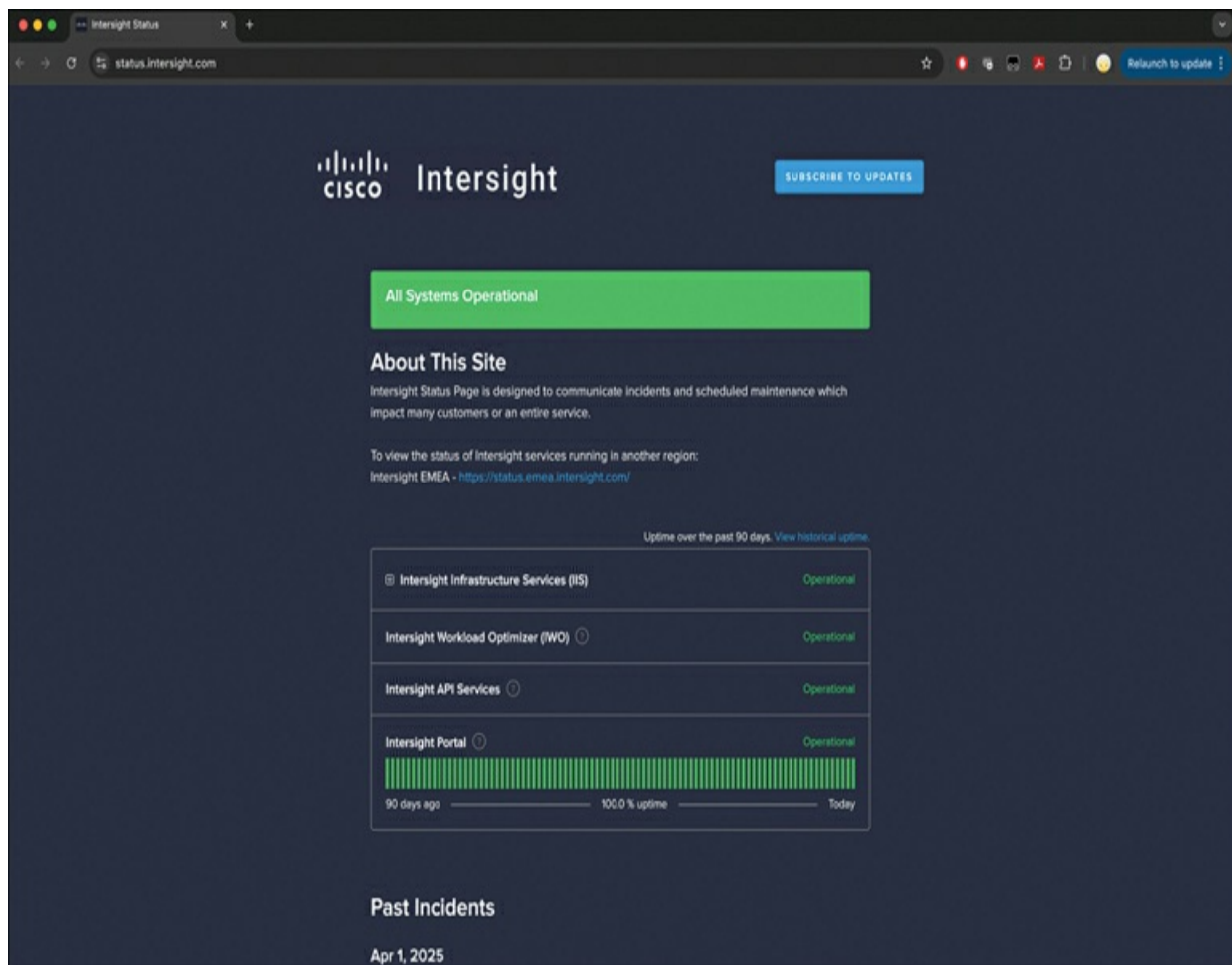


Figure 14-1 Using <https://status.intersight.com> to View Current Status and Incidents for Intersight

Should a SaaS outage occur, the Intersight DevOps team handles the situation without any need for customer involvement, and the endpoint devices (UCSM, Cisco Integrated Management Controller [CIMC], Hyperflex, and so on) are all unaffected from a functionality perspective. We will discuss private cloud delivery modalities further in this chapter, but customers should not discount the plethora of benefits in a SaaS delivery model. Unless a customer has a specific need, SaaS is by far the recommended delivery model.

As you would expect with a fully centralized system such as Intersight, visibility into the health, configuration, version-distribution/compliance across an entire install-base is an extremely valuable feature. Having a centralized place to view software uniformity, compliance, and AI Insights is

a game changer in the data center.

Aside from removing operational and scale concerns for customers, Intersight also provides a centralized feature set of capabilities that were not possible in the fragmented management architecture previously available to its customer base. One great example of this is Intersight's tunneled keyboard video and mouse (KVM) capability. From the Intersight SaaS portal, leveraging the always-on, secure, bidirectional connectivity between the on-premises devices and Intersight Cloud, you can launch a live KVM direct to any of the connected servers across all sites, in any geographical location. This capability eliminates the need to search for server domains or IP addresses or to use jump servers and VPNs.

The path to a full SaaS-managed compute environment can be a multistep journey. Intersight intends to meet customers where they are, allowing end customers to claim endpoints such as UCSM and CIMC but not fully managing them, providing the level of visibility and observability customers have come to expect with a SaaS offering but not requiring a full lift-and-shift of management into the cloud. For customers choosing this approach, Intersight offers a cross-launch capability that allows them to continue using Intersight as a single landing page but then cross-launch into element managers such as UCSM, CIMC, Hyperflex Connect, and UCS Director.

Endpoints that connect to Cisco Intersight do so using a piece of software called a *device connector*. This lightweight software module runs on the endpoint itself. It is responsible for establishing a durable WebSocket between the endpoint device running in the customer's network to Intersight SaaS or Intersight appliances. [Figure 14-2](#) illustrates the device connector's connection to Intersight. Intersight appliances will be covered in the "[SaaS vs. On-Prem Implementation](#)" section. The WebSocket provides connectivity that is

- **Secure:** The Transport Layer Security (TLS) secured connection uses Advanced Encryption Standard (AES) with a 256-bit randomly generated key.
- **Always On:** The WebSocket is durable and reestablishes connectivity to Intersight should its connection be severed for some reason.
- **Bidirectional:** Intersight can send control messages to the device

connector, and the device connector can send information back to Intersight.

- **Always Up to Date:** Intersight ensures the device connector is running the appropriate version.

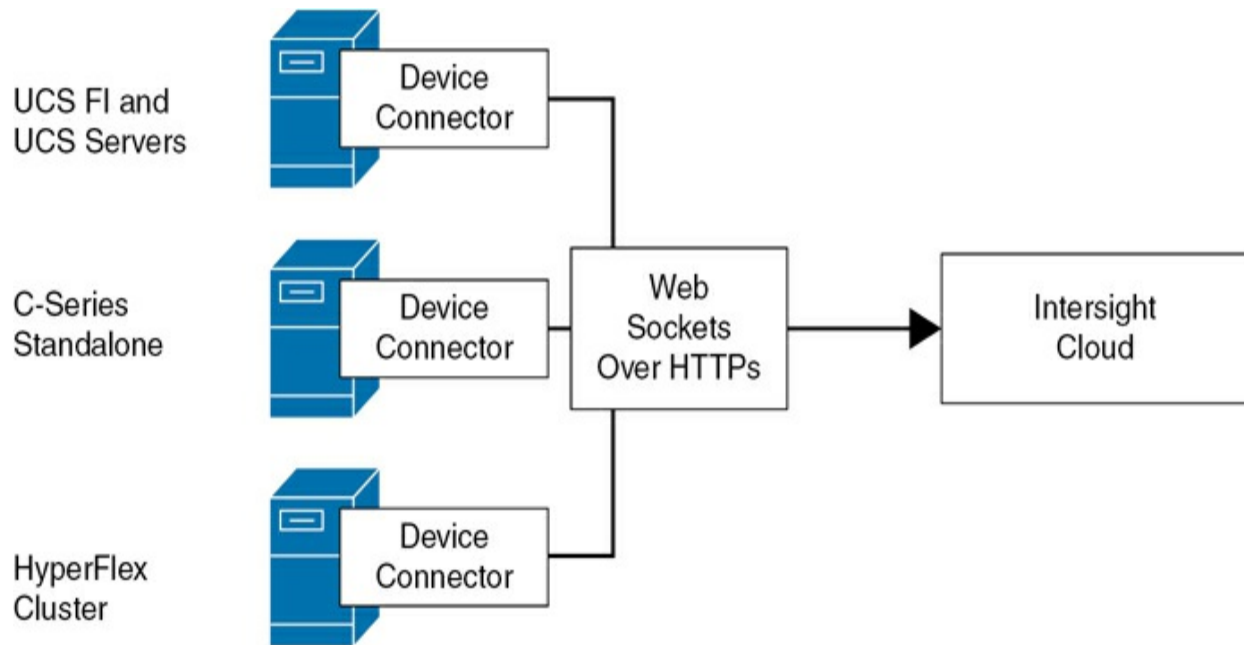


Figure 14-2 Device Connector Web Socket Connectivity to Intersight

The device connector itself initiates the connectivity to Intersight, which means customers do not have to configure any port forwarding or expose any services to the public Internet. It is embedded in the endpoint itself (bundled with the firmware); as such, there is very little for the end customer to do to enable connectivity. For more information on how the device connector can be configured to meet specific requirements, see the “[Common Deployment Scenarios](#)” section later in this chapter.

As we all know, there is a wide range of additional technologies in the data center beyond just compute resources. For information on how Intersight can communicate with devices that do not have a device connector embedded, see the “[Device Integration \(Cisco and Third Party\)](#)” section later. In addition to compute devices, several data center networking components run a device connector and can connect directly to Intersight:

- Application Policy Infrastructure Controller (APIC)

- Data Center Networking Manager (DCNM)
- Nexus 9000 Switches
- MDS 9000 Switches
- Nexus Dashboard

SaaS vs. On-Prem Implementation

The SaaS deployment modality is recommended for all customers unless there is a specific need to be on-premises, which is quite rare. Using SaaS is not only more cost-effective for customers but requires less management, no traditional operations costs (hosting, patching, scaling, and so on), and no maintenance. Additionally, the scale of SaaS is unmatched by any of the on-prem implementation options, yielding an option where customers no longer must even think of scale and volume considerations at the management plane.

Customers should avoid the misconception that air-gapped or on-prem is, by default, more secure than a SaaS offering. Because of the CI/CD nature of SaaS, vulnerability patching is done typically much quicker than it can be for on-prem implementations. Likewise, the concern around uptime is a misconception. Intersight DevOps engineers are available around the clock to ensure the platform is performant and available.

For those customers requiring an on-prem implementation, there are two options: a connected virtual appliance or a private virtual appliance. The CVA aims to provide the control of an on-prem management plane with some of the benefits of a connected experience to Cisco. The PVA loses all of the benefits of connectivity to Cisco but provides the end customer with the Intersight management experience and complete control over any telemetry because the deployment is air-gapped. For more information related to the connectivity of Intersight to Cisco, see “[TAC Integration](#)” later in this chapter.

Device Integration (Cisco and Third Party)

With the evolution of the various Cisco compute management software platforms, one of the key benefits of Intersight is providing a single pane of

glass to view and configure all your devices in one place. This capability allows administrators to integrate multiple different Cisco compute, storage, and networking platforms. Before Intersight, each one of these platforms would require its own unique management software that administrators would have to log in to and could manage only one device or a subset of devices that were a part of the same product family. Supported Cisco products and various management modes include

- Standalone C-series rack servers
- UCSM Managed (UMM) B-, C-, and X-series servers
- Intersight Managed Mode (IMM) B-, C-, and X-series servers
- Hyperflex storage clusters
- Nexus and MDS switches

Note

To see a full list of supported hardware, visit https://intersight.com/help/saas/supported_systems.

Intersight also supports a wide range of third-party products that are commonly deployed alongside UCS to complete the data center ecosystem. This integration allows administrators to not only manage the physical Cisco devices but also configure other parts of the solution from a single interface—for instance, provisioning remote storage to a UCS server or patching and updating an OS.

Intersight even can support some basic monitoring and management of third-party servers, such as powering them on or rebooting them by leveraging the industry standard Redfish capabilities that are also commonly utilized in other products, including Dell and HP servers. Because the device connector is a Cisco proprietary concept and Cisco is responsible for developing the software and integrating it into its products, Intersight has created a platform to integrate some of its third-party systems and partners: Intersight Assist. Essentially, Assist acts like a device in the middle of third-party products and Intersight; it ensures proper communication between them. Some examples of supported third-party products are

- VMware vCenter
- Pure Storage
- NetApp
- Hitachi
- Nutanix
- Dell/HP servers

Note

For a full list of third-party integrations, see https://intersight.com/help/saas/supported_systems#supported_hardware_party_targets.

Integration with Intersight does not stop with only targets that support being onboarded directly into Intersight. Many commonly utilized software platforms or tools also can leverage Intersight's APIs via software development kits and plug-ins.

For example, Intersight has a powerful plug-in developed for use with ServiceNow; many IT organizations use it to manage IT incidents and their resolution. In this way, ServiceNow can leverage Intersight's constant monitoring and reporting of health from all the various endpoints it manages. If an alert is raised for a device that needs attention, this plug-in can automatically and simultaneously raise an incident in the ServiceNow platform so that administrators are alerted to it and track its resolution. Some other examples of SDKs and plug-ins are

- Python SDK
- Ansible
- Terraform
- PowerShell

Note

For a full list of SDKs and plug-ins, see the "Downloads" section of Intersight's help documentation at

<https://intersight.com/apidocs/downloads/>.

To highlight the power of Intersight and the integration of multiple Cisco and third-party products, let's consider an example of a typical administrative maintenance task: keeping your UCS servers up to date with the latest supported server firmware, OS version, and necessary driver combinations to keep them operating as expected. Typically, as an administrator, you may be responsible for only maintaining the OS running on the server; in that case, you would have to collect the current running OS version and locate any driver versions that are installed to run the various components inside a server like a VIC adapter's Fibre Channel (FC) interfaces, Ethernet interfaces, RAID controllers, and so on.

After collecting that information, you must then find all the current server firmware running on that physical device—for example, the BIOS, CIMC, and VIC adapter firmware. You may even have to consult another department that is responsible for maintaining the physical hardware and software running on them if that is not maintained by your department.

When you have all that information, you would then consult a commonly used tool called a hardware compatibility list (HCL). After manually selecting all of the hardware and software combinations running on that server from the drop-down menus in the HCL, you would be presented with the recommended drivers and software versions that are supported. You would then have to repeat that same process for any given platform that may have differed from the previous combination. For example, you may have a different server model or a different OS version.

Note

You can access the HCL tool at
<https://ucshcltool.cloudapps.cisco.com/public>.

All that process can be simplified with Intersight by leveraging its built-in HCL tool and supported integrations. In Intersight, this tool will be displayed to the administrator all on one page. Intersight is aware of the server model in question and all of the components installed in that server. It is also aware of the firmware running on these components (VIC, BIOS, CIMC, and so on). With visibility into the OS leveraging some additional software plug-ins, it

will also be able to pull the running OS version and the drivers installed for those devices. It will then cross-reference the HCL database that is used to maintain the standard HCL tool and provide the user with the current status of compatibility and, if anything is incorrect, provide the necessary recommended versions to correct them. [Figure 14-3](#) details HCL functionality in Intersight.

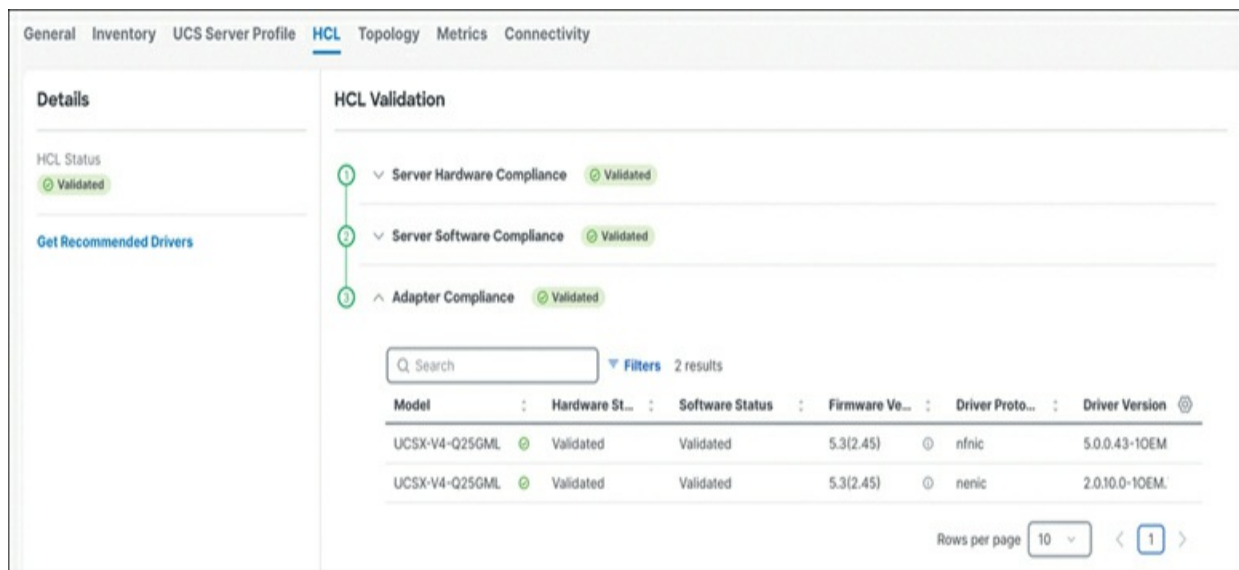


Figure 14-3 HCL in Intersight

Key Features and Functions

Intersight is a robust, feature-rich management platform enabled by the rapid pace of a SaaS delivery model. Moving on from the traditional monolithic UCS firmware development schedule, which would release new firmware typically on a three-month schedule, new features can now be delivered on a weekly basis.

Although some new features still may require an infrastructure firmware upgrade, consisting of Fabric Interconnect, Input Output Module (IOM), Intelligent Fabric Module (IFM), or the servers themselves, many new features or fixes can be delivered via a back-end microservice update to the cloud or an update to the device connector firmware running on the endpoints. These updates can be done without any downtime to the targets as well. In this way, administrators do not have to plan large-scale change windows with their companies to account for downtime to servers running

production applications. Major security fixes for any newly discovered vulnerabilities can also be deployed via these updates and allow for patching of critical security flaws without any user intervention or planning.

One of the key and foundational features that Intersight first provided is basic inventory management of all the supported targets. Once a device is onboarded into Intersight and is claimed, all the device’s components are inventoried. This feature allows for basic searching and filtering of devices, so administrators can quickly find assets or a subgroup of assets that may meet a particular resource requirement. [Figure 14-4](#) shows an inventory listing of targets in Intersight.

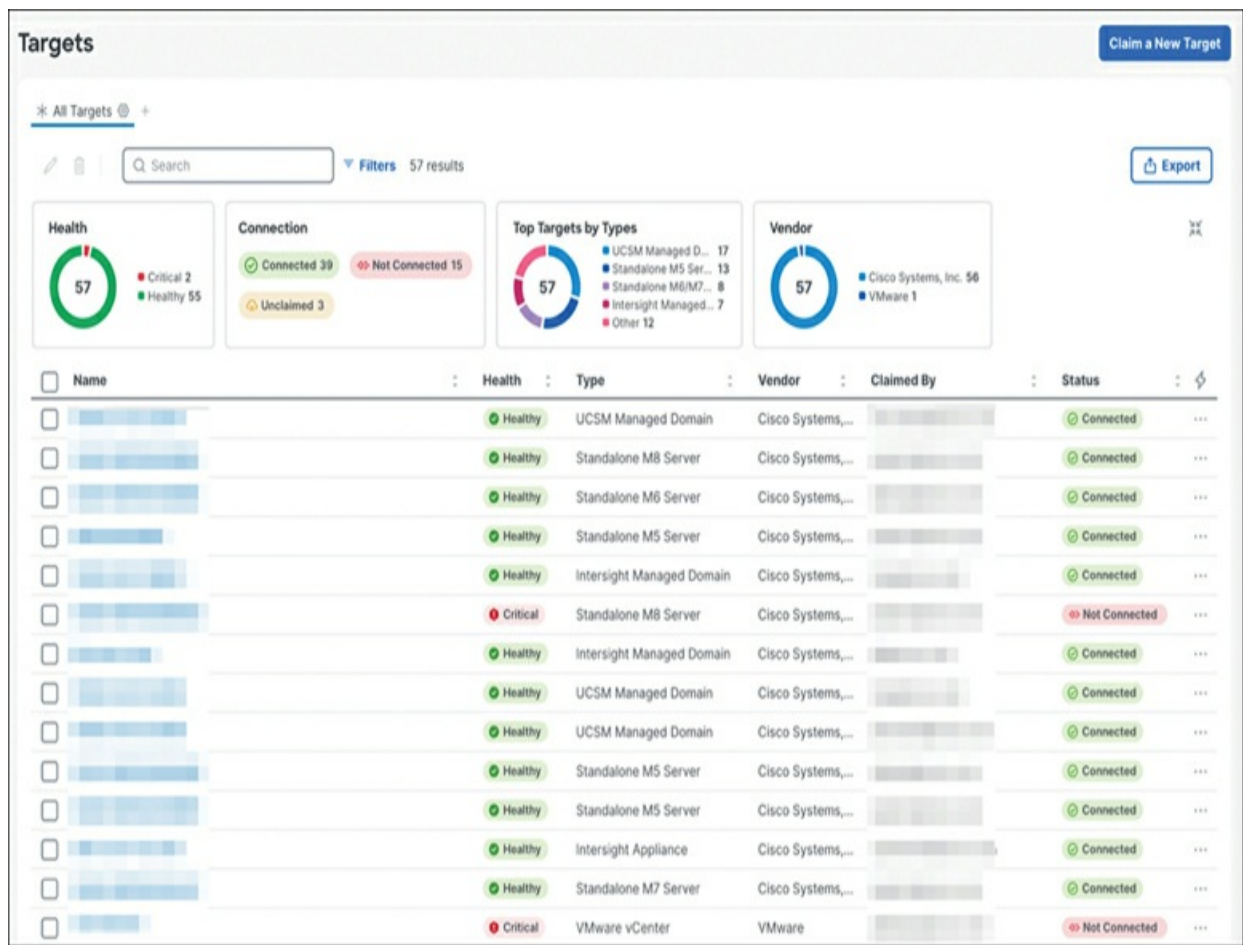


Figure 14-4 Intersight Target Inventory Listing

These different views can also be edited and customized to display only columns or widgets that are important to administrators, as illustrated in [Figure 14-5](#).

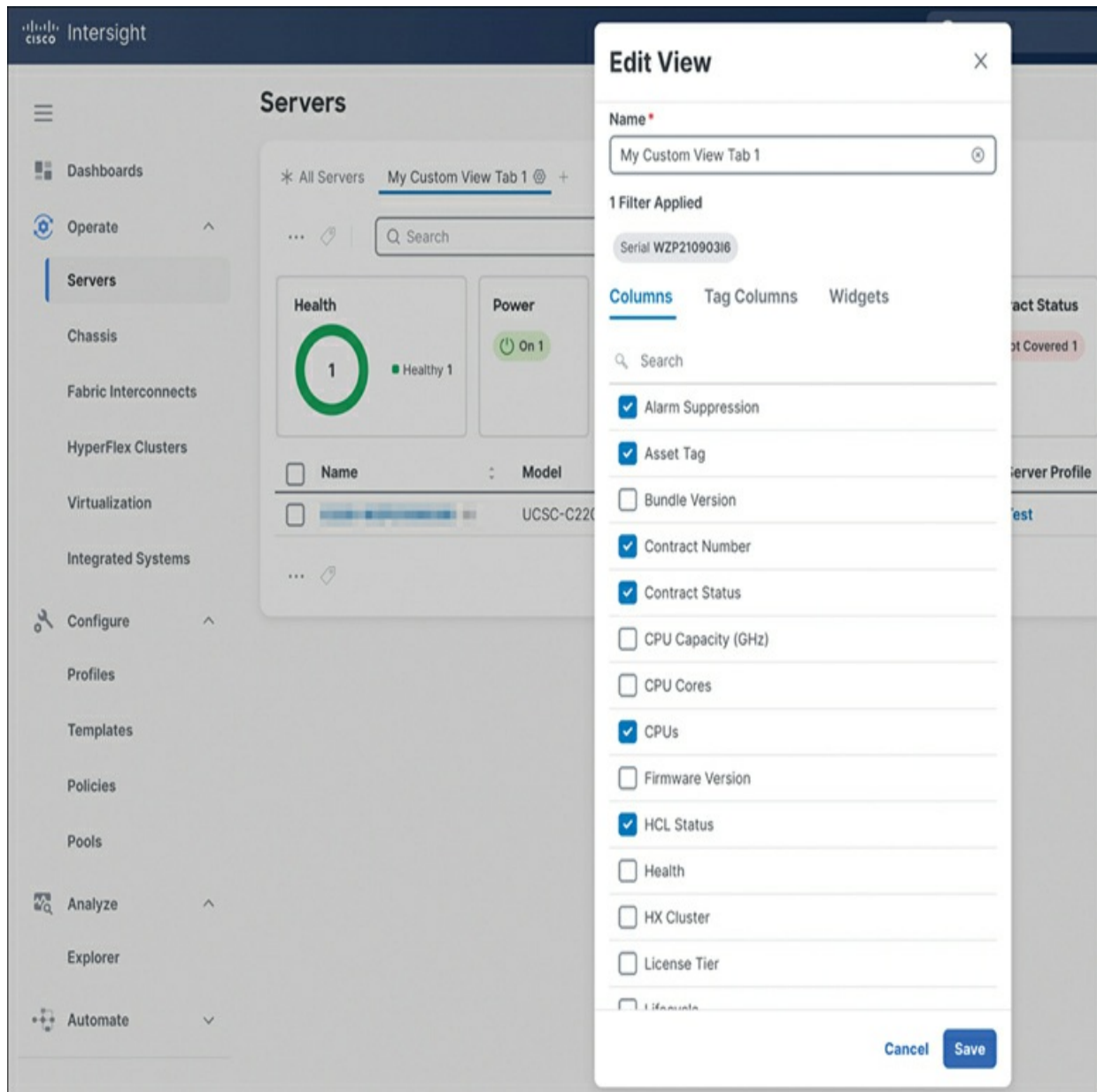


Figure 14-5 Customizing the Display for Intersight Inventory

Another foundational principle that developers kept in mind during Intersight's inception was to build the product with an API-first mentality. Intersight is built off robust REST APIs that are documented according to the OpenAPI standard, which enables Intersight to be programmable; any platform that can leverage a REST-based API framework can interact with it.

To have access to these features, Intersight is based off a subscription-based licensing model. There are currently two licensing tiers: Essentials and

Advantage. Customers wishing to claim endpoints that do not require licensing can leverage an Unlicensed tier but will not experience any of the benefits provided by Essentials or Advantage. The main differences between Essentials and Advantage licensing tiers are which features are enabled.

All the features that are enabled in the Essentials tier are also available in Advantage but with the addition of more advanced features. For example, the Advantage tier enables features such as OS installation, which will be discussed in more detail later in this section. New customers evaluating Intersight may use and take advantage of a free 90-day trial after creating their account. It is important to mention that starting with the UCS M7 generation and beyond, licenses are bundled with the purchase of a server.

To manage these licenses, Intersight also leverages Cisco Smart Licensing, like many other Cisco products. This feature provides an easy way to track, transfer, and activate your licenses in a consolidated portal. To enable the telemetry for Intersight to interact with your Smart Licensing account, you must register your account from within the Licensing section in Intersight and enter a Smart Licensing Instance Registration Token that you can obtain from your Cisco Smart Software Manager account. [Figure 14-6](#) shows how to handle license management in Intersight.

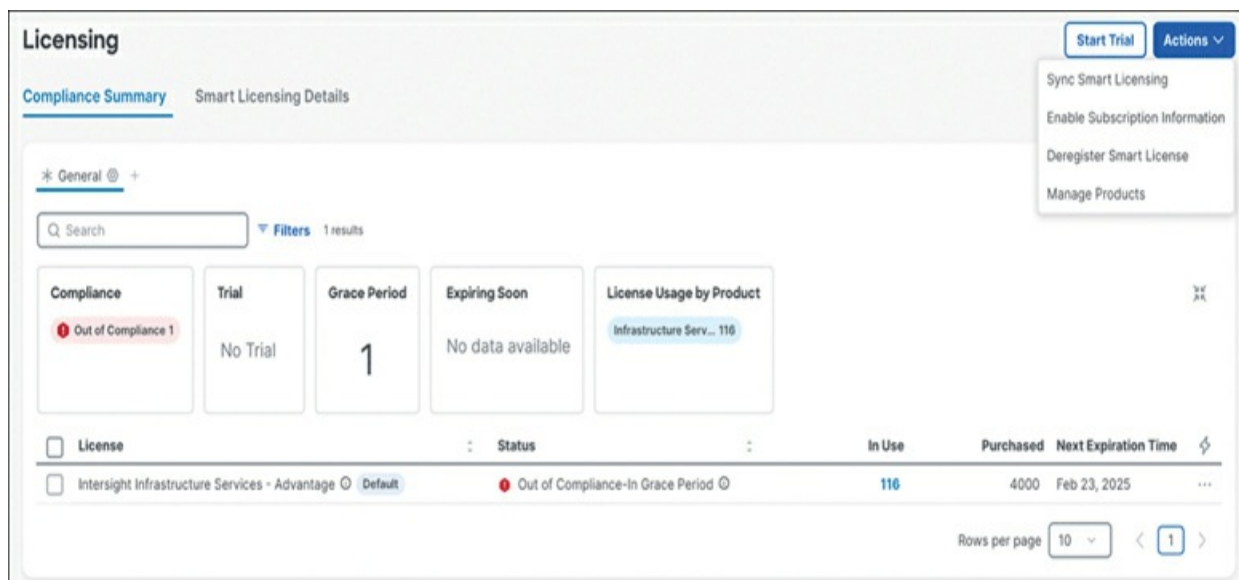


Figure 14-6 License Management in Cisco Intersight

Once registered, Intersight will now sync with the Smart Licensing Portal and

provide a pool of licenses that were purchased and available for use. This feature provides an easy way to track your usage inside Intersight and any resources that may be out of compliance and future considerations for purchasing more as environments may scale over time. You can also enable default tiers of licenses to be applied when new devices are onboarded.

A core concept of what makes UCS powerful and a differentiator from other server vendors is the ability to configure the devices with service profiles and templates that are made up of policies and can be reused. Intersight has now taken that core reuse concept one step further by allowing the use of a single policy across multiple management platforms. For example, you can now create a single SNMP policy for All Platforms, as shown in [Figure 14-7](#). These platforms include the following:

- UCS Server (Standalone)
- UCS Server (FI-Attached)
- UCS Domain
- UCS Chassis



Figure 14-7 Configuring an SMMP Policy for All Platforms

In a legacy UCSM Managed (UMM) domain not managed by Intersight, those policies could only be reused within the scope of the same UMM domain and could not be used in other UMM domains, for example, or by standalone C-series rack servers that may be deployed elsewhere.

Similarly, Intersight supports the sharing of the various ID pools. What once used to be a risk while managing multiple different environments and platforms was that one domain might overlap some form of ID that must be

unique—for example, a MAC address. That overlap can be avoided because, as Intersight is maintaining its assignment, it cannot be leased out more than once to one unique device.

As mentioned earlier, an important administrative subject for the basic maintenance and management of a data center environment is managing firmware and updates. Although the core framework of Intersight is maintained and updated through the SaaS delivery model or via virtual appliance upgrades, updating the endpoint device's firmware will always be required—for example, applying a server BIOS upgrade or Fabric Interconnect upgrade to apply a new feature or security fix. Before Intersight, this task required the use of multiple management platforms, such as logging in to UCSM or the CIMC user interface, depending on the devices that need to be upgraded. Administrators had to manually retrieve the necessary firmware from [Cisco.com](https://www.cisco.com), upload the firmware to the device's management platform, and apply all the detailed steps that are required for it to be upgraded. Intersight takes away that burden because all the devices can be upgraded from a single interface.

As a precursor, Intersight has access to all the firmware images supported for a given device. Firmware bundles are uploaded to Intersight's cloud provider for all the supported devices that are connected. Depending on the delivery model via SaaS or the on-prem appliance, a device will then reach out to the cloud and download the necessary images directly after an upgrade is triggered.

Because a private virtual appliance does not have this external connection to the cloud, firmware bundles are manually uploaded to the appliance via a local machine or remote share in which it has access. To download the firmware, Intersight leverages a content delivery network (CDN) for a geographically distributed group of servers that caches content closer to the end devices to attempt to speed up the transfer of these files. This process requires specific network requirements to access these download URLs. See the “[Device Onboarding and Security](#)” section for more information.

This process is streamlined and made easy through an Upgrade Firmware Wizard that walks you, as admin, through the process; the target is upgraded to the desired firmware of your choosing. After an upgrade is triggered or scheduled, you do not need to do much more than monitor and approve any

necessary reboots as they are to be scheduled.

The following are supported devices that can be upgraded via Intersight:

- Cisco UCS C-Series M4, M5, M6, M7, M8; and S-Series M4, M5 servers that are configured in Standalone mode
- Cisco Fabric Interconnect-attached UCS B-Series M3, M4, M5, M6 servers; C-Series M3, M4, M5, M6, M7; and UCS X-Series M7 servers in UCSM Managed mode
- Cisco Fabric Interconnect-attached UCS B-Series M5, M6 servers; UCS C-Series M5, M6, M7, M8 servers; and UCS X-Series M6, M7, M8 servers in Intersight Managed mode
- Cisco Fabric Interconnect-attached Cisco UCS S3260 M3, M4, and M5 servers in UCSM Managed mode
- Cisco Fabric Interconnect-attached Cisco UCS S3260 chassis in UCSM Managed mode
- Cisco UCS Fabric Interconnects Series 6200, 6300, 6400, and 6500 in UCSM Managed mode
- Cisco UCS Fabric Interconnects Series 6400, 6500; and Cisco UCS Fabric Interconnects 9108 100G in Intersight Managed mode

While this is not an exhaustive list of the features supported by Intersight, the last powerful key feature that is worth mentioning is another task that can be very daunting and tedious: installing an operating system. After you utilize Intersight to deploy your basic server configurations, it is time to get an OS up and running in its desired configuration. Intersight can fully automate this process.

The configuration wizard will walk you through some basic parameters needed for the intended configuration, such as the OS image to be installed, server configuration utility, and the operating system configuration file. The OS image will need to be hosted on a network share that is accessible via the server's management IP address and will be used to install the OS to disk.

Intersight will use the server configuration utility to prep the server for the OS installation, such as configuring the local storage drives as a viable boot

location for the OS. This image will also be hosted on a network share.

Lastly, the operating system configuration file has the basic parameters you would like configured at the OS level. For example, this includes basic networking information like IP address, gateway, and hostname. There are three ways to supply Intersight with this information for the installation:

- **Cisco Mode:** Users can use standardized templates provided by Cisco that are common settings for most supported operating systems that can be used.
- **Custom Mode:** Users are responsible for supplying their own custom kickstart files.
- **Embedded Mode:** Users can provide any OS settings that are preconfigured and stored inside the OS image.

After this prework is done and supplied to Intersight, it will take care of the rest of the steps needed to get an OS up and running and can be left unattended while it is deployed. This capability can drastically reduce the amount of time and effort required to get a fully configured and deployed OS into production from scratch. Additionally, this feature increases the consistency of configurations across the data center.

Note

Like most SaaS platforms, Intersight has features and functions released at a rapid rate. In fact, regular weekly updates are made to the product, so it can be hard to keep up. You are encouraged to visit the “What’s New” section at https://www.intersight.com/help/saas/whats_new to view the latest information.

So far, we have provided an overview of the Intersight platform. We introduced the concept of cloud-managed compute, highlighted the difference between SaaS and on-prem implementations, covered integrations with Cisco and third-party devices, and detailed key features and functions. The next section will build on this foundational knowledge so that you can see how Intersight provides automation and insights for the data center.

Automation and Insights

The real power and business value of Cisco Intersight is realized when you utilize the automation and insights that this SaaS platform provides. More specifically, the automation and insights provided by Intersight can be divided into two key areas: orchestration and programmability and Infrastructure as Code (IaC). Each will be covered in its own section.

In the “[Orchestration](#)” section, you will see how Intersight can simplify and automate complex workflows across diverse environments. This capability allows you to create and execute tasks that span compute, network, storage, and virtualization domains using prebuilt libraries and building custom workflows. Real-time monitoring and error handling in Intersight for these tasks ensure that execution happens efficiently and with minimal risk.

Next, the “[Programmability and Infrastructure as Code](#)” section will highlight Intersight’s API-first approach, which ensures that every feature available in the user interface is also accessible via APIs. This design enables seamless integration with third-party tools like Ansible and Terraform, allowing organizations to adopt IaC methodologies effectively. Additionally, Intersight’s webhook and event-driven architecture enhance automation by enabling real-time responses to system events, further reducing the need for manual intervention.

Orchestration

At its core, Intersight provides orchestration capabilities designed to simplify and combine common tasks. These tasks often require configuring multiple different devices and subsystems. Intersight accomplishes this goal by utilizing a concept called *workflows*.

In the upper-right corner of the Intersight window, you will find a button that will pull up all the active requests. Anytime an admin takes an action on a device within Intersight, its overall status is monitored through the Requests tab. A request can be a single task or a series of tasks that need to be completed that Intersight is automating and orchestrating for them. [Figure 14-8](#) illustrates a workflow for Blade Discovery.

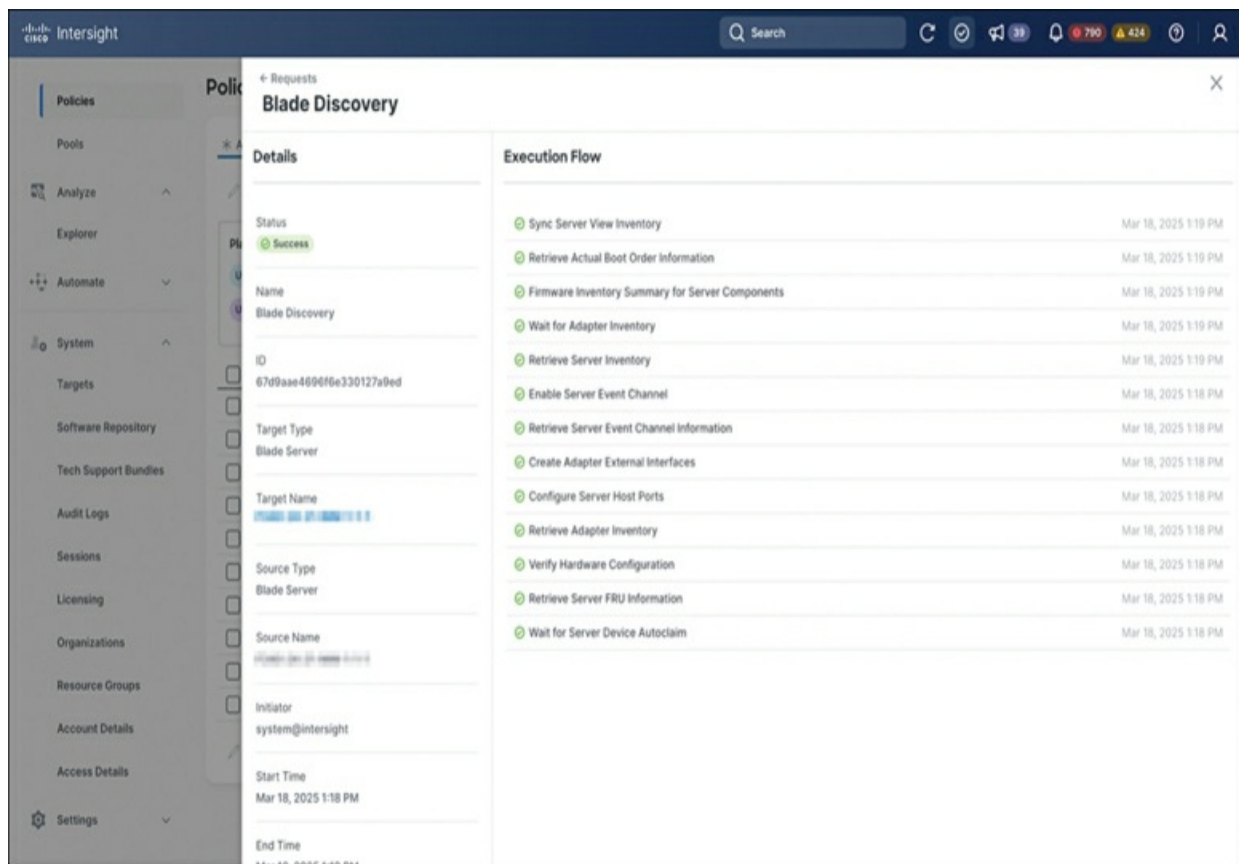


Figure 14-8 Intersight Workflow for Blade Discovery

This capability is incredibly powerful because it enables you to monitor the task's overall progress and see it completed on a step-by-step basis. If the task fails, this tab will list the current step it was attempting to complete. For help in debugging or troubleshooting, the issue should raise an error message to help diagnose and fix the problem. An example of a workflow that can be monitored in the Requests tab is a firmware upgrade on a Fabric Interconnect or an OS install of a server.

Another powerful example of Intersight's capability to orchestrate complex or tedious tasks is to build onto the automation it provides for one task and then multiplies it with the use of Bulk operations. Intersight allows you to select multiple devices on which you want to perform a particular action.

An easy example is to select a group of servers that need to be rebooted, powered on, or upgraded, and these tasks will all be done at once in parallel. There's no need to go to each individual device and perform the action. An even more powerful example is to perform an OS installation on multiple

servers. At this point, you are already leveraging the automation built into the OS installation feature and then combining and enhancing the feature by allowing Intersight to orchestrate that task over many devices at once.

Intersight also presents administrators with real-time insights into the status and health of the targets, plus the ability to take necessary actions quickly. It accomplishes this with the use of dashboards, widgets, and custom views and pages presenting summaries of an environment's status.

One key insight it provides is through alarms. Alarms are raised as an important notification that an endpoint might have encountered some type of problem. These alarms leverage device connectors' always-on and bidirectional communication flow, allowing them to trigger rapidly instead of waiting for a polling schedule. These alarms come in on a prioritized basis in the form of Critical, Warning, and Informational.

When the condition that triggered the alarm in the first place has cleared, the alert will be moved to a Cleared category and can be reviewed for historical reference as necessary. These alarms can also automate the task of alerting the appropriate people in charge of managing that device in the event an alarm is triggered. A user has the option of setting up email notifications and can customize where these notifications are sent to and at what severity levels.

Beyond collecting the health of targets in one location via the Alarms page, Intersight also enables you to create custom dashboards and views. On the Intersight home page, you will find yourself in the Dashboards section. This is a central location where you can get a high-level overview of your data center.

Examples of dashboards are Server, Hyperflex, and Fabric Interconnect health summaries showing their overall fault counts. Storage dashboards can show overall storage utilization for arrays and clusters. If storage is running low in a Hyperflex or Nutanix cluster, or SAN storage from a third-party integrated array, Intersight's orchestration can simplify the storage administrator's role by provisioning more space with just a click. Simply expand it with Intersight's orchestration of the storage administrators' role by provisioning more space with the click of a button.

A license status widget shows the number of devices out of license compliance and usage, helping organizations plan for future growth and scale. All of these dashboards and pages can be customized, added, and removed, providing users with the most relevant information for their daily tasks.

Intersight also provides a powerful feature to audit an environment for new security vulnerabilities, field notices, and devices approaching end of life. This information is all collected in one central location under the Advisory section. Before this feature existed, reviewing an environment was a laborious task, especially when done at scale. Often it is easy to miss that a new Common Vulnerabilities and Exposures (CVE) list has been released to the public. Not only does Intersight handle the notification of this list, but it also uses the CVE inventory to gather any device that may be running impacted firmware and provide the list with recommended actions that need to be taken. An example of an advisory related to a vulnerability, along with all affected devices, is shown in Figure 14-9.

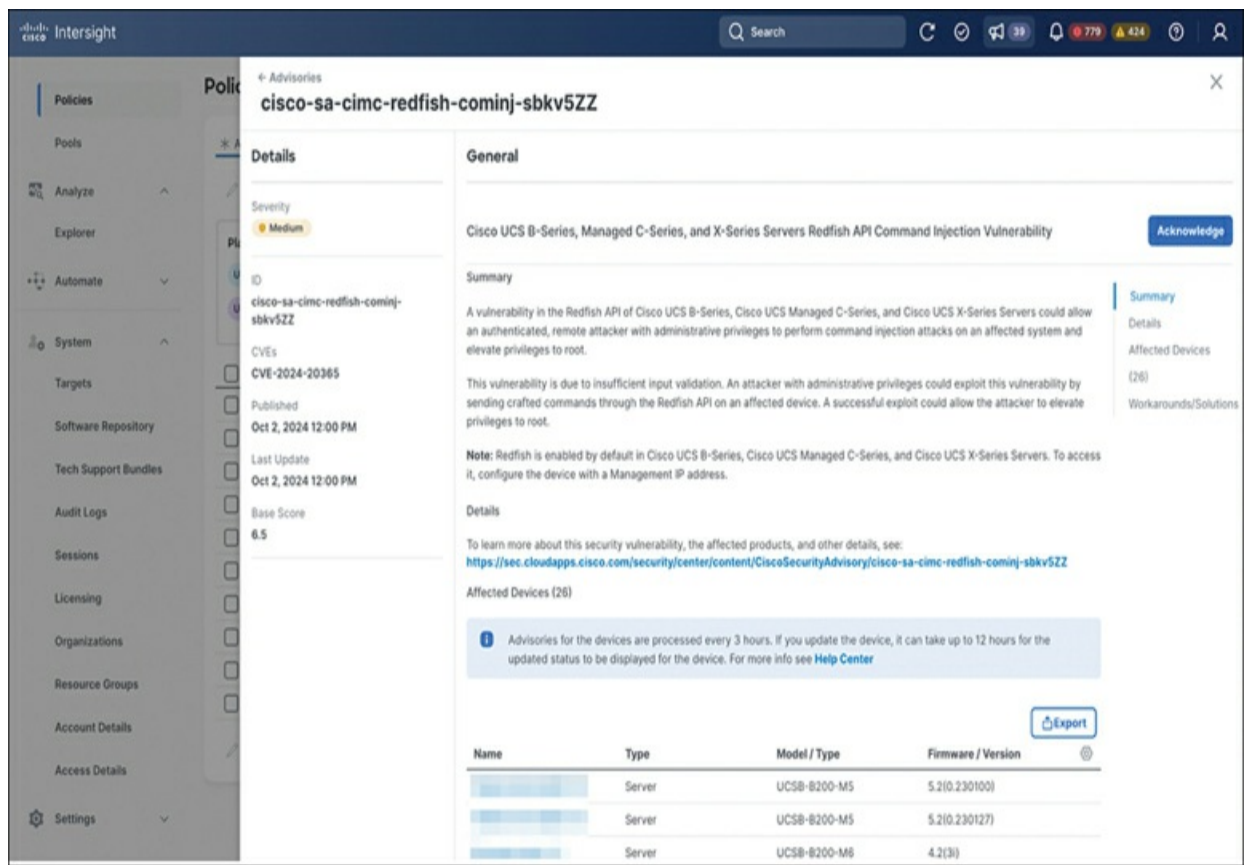


Figure 14-9 Advisory for a Vulnerability Showing Affected Devices

Field notices are handled in a similar manner. For any new field notices released, Intersight alerts you of any critical information about a product that is impacted and whether an action or workaround needs to be taken. End-of-life notifications are also communicated to you as devices begin to reach their end of support.

Missing one of these notifications can be absolutely devastating to an environment. This is especially true when it comes to something like a field notice, which can often affect a production device if not addressed. Discovering a known issue after it has caused an impact can be particularly frustrating. Luckily, with Intersight, it is always leveraging new information as it is released and proactively informing users of the impact to connected devices.

Although Intersight has quite a bit of automation and insights running out of the box, what if you want to run some relatively bespoke automation yourself? Intersight actually can support this capability as well with the Intersight Cloud Orchestrator (ISO) feature, which is referred to as *workflows* within Intersight. This feature uses the same underlying orchestration engine for workflows that was previously discussed in this section. Customers with Intersight Advantage licensing can author their own custom workflows that can do anything from powering servers on or off to running full custom PowerShell scripts.

Programmability and Infrastructure as Code

Intersight was developed with an API-first mentality, which doesn't just mean that everything you can do in the UI can be done in the API. It means the UI itself consumes the API and *only* the API. Prior to any UI feature being developed, the foundational API capabilities are first designed and built. As a result of this methodology, customers can benefit from a robust set of well-documented APIs that can support automation for any capability a human can do manually.

Intersight has chosen to use OpenAPI (formerly Swagger) as the documentation mechanism for all API endpoints that are available. The API spec is publicly available for both SaaS and the on-prem appliance at

<https://intersight.com/apidocs/apirefs/All/introduction/>.

Having the API spec available on the appliance itself as opposed to a public Internet doc is nice because the appliance itself knows what is and is not supported based on the current revision. Within the API documentation, customers can view available endpoints, requests, and response schemas and can test live API calls against any supported endpoint. Figure 14-10 shows an example from the Intersight API documentation and how an API request can be initiated.

The screenshot displays the Intersight API documentation interface. On the left, the 'API Reference' sidebar shows the hierarchy: 'compute/BladeIdentities'. The main content area shows the 'GET /api/v1/compute/BladeIdentities' endpoint. It lists several query parameters: '\$filter' (string), '\$orderby' (string), '\$top' (integer), '\$skip' (integer), '\$select' (string), '\$expand' (string), and '\$apply' (string). Each parameter has a description of its function. On the right, the 'REST Client' panel is open, showing the same endpoint and a 'Send' button. Below the 'Send' button, the 'Response Text' tab displays a JSON response. The response is a JavaScript Object Notation (JSON) object representing a collection of blade identities. The JSON structure includes an 'ObjectType' field set to 'compute.BladeIdentity.List', a 'Results' array containing a single object, and various fields for the blade identity such as 'AccountMoid', 'AdminAction', 'AdminActionState', 'Ancestors', 'ChassisId', 'ClassId', 'CreateTime', 'CurrentChassisId', 'CurrentSlotId', 'CustomPermissionResources', 'DiscoveredBladeIdInCurLocation', 'DomainGroupMoid', 'FirmwareSupportability', 'Identifier', 'LastDiscoveryTriggered', 'Lifecycle', 'LifecycleModTime', 'ManagedNodes', 'ManagerSlotId', 'ModTime', 'Moid', 'Name', 'NewBladeIdInDiscoveredLocation', and 'ObjectType'.

Figure 14-10 Intersight API Documentation and API Request

Intersight's APIs are standardized. All APIs provide JavaScript Object Notation (JSON) as output, they accept JSON as input for body payloads, and for GET API calls, they follow the OData standard (<https://www.odata.org/>). We will not cover the details of OData here, but you can find all query guidelines in the Intersight Query Syntax documentation at <https://intersight.com/apidocs/introduction/query/>. For an even deeper dive

with examples of API calls, consider reading *Cisco Intersight: A Handbook for Intelligent Cloud Operations* by Matthew Baker et al.

Although we will not delve into all the idiosyncrasies of the API, an overview of the GET API parameters will help you understand the general capabilities and what is possible:

- **\$filter:** Every object in Intersight is filterable, including items within nested JSON objects; this makes extraction of data incredibly surgical.
- **\$select:** You can select what top-level fields should be returned by the API, allowing you to reduce the payload sizes of large queries.
- **\$count:** This parameter instructs the query engine to return only the count of documents that would match the \$filter.
- **\$inlinecount:** This parameter allows API consumers to know the total volume of objects that would match the \$filter parameter without paginating all responses. This is different from \$count in that the actual results of the query are returned along with the count, hence the term *inline*.
- **\$stop:** This parameter returns the next *N* results from the given query.
- **\$skip:** This parameter allows for pagination (iterating over large responses of results and gathering all of them on the client side). By default, Intersight returns the first 100 results, and incrementing the \$skip parameter allows you to retrieve chunks of the response.
- **\$orderby:** You can choose which field or fields to sort the results and which direction (ascending or descending).
- **\$expand:** Intersight uses a NoSQL database backend, which typically does not have relational links like a SQL database, but this parameter allows the gathering of related documents. This can even be chained together multiple times to get multiple levels of related objects.
- **\$apply:** For those familiar with SQL, this parameter is similar to a GROUP BY clause; it is used to aggregate responses and perform transformations on the data. For example, this parameter could get you the count of servers summed up by the firmware they are running.

The choice of standardizing on OpenAPI as a documentation mechanism

comes with some added benefits. The spec itself is available on Intersight's download page (<https://intersight.com/apidocs/downloads/>) in both JSON and YAML (YAML Ain't Markup Language). OpenAPI also provides automated SDK generation, meaning the SDK package is always up to date with the latest endpoints and model validations.

Intersight publishes both a Python and a PowerShell SDK that are generated off the OpenAPI v3 spec. If you want to test the API directly, you can use the OpenAPI page to try out any API endpoint using your current authentication. If you want to try out API endpoints with API keys but are not yet ready to leverage an SDK, Cisco DevNet has published Postman JSON objects that can handle API authentication for you at <https://github.com/CiscoDevNet/intersight-postman>.

You can start consuming Intersight's APIs programmatically by generating API keys directly within Intersight itself. API keys adopt the persona and role-based access control (RBAC) of the creator. Although Intersight supports several authentication patterns, in practice the one you will use leverages *HTTP signature* authentication. You can find more information on API key generation and how to use them at <https://intersight.com/apidocs/introduction/security/#benefits-of-using-api-keys>.

Programmability is important and foundational, but Intersight goes a step further with Ansible and Terraform integrations available on the download page; they help organizations adopt an IaC methodology. See the “[Management and Analytics](#)” section in [Chapter 2](#), “[SaaS Architecture](#),” for more details on IaC, Terraform, and Ansible.

Customers can take advantage of open-source Terraform and Terraform Cloud and can claim Terraform Cloud as a target within Intersight. This allows for a SaaS-to-SaaS offering integration with enhanced authentication mechanisms, secret management, state management, and access to support. Both the Terraform Provider and Ansible modules are autogenerated as a result of the CI/CD deployment of Intersight and are therefore always up to date with the latest API spec.

Although the ability to interact with an API is critical to support automation, event-driven architectures are also an important component to help drive

automation reaction speed as well as efficiency. Automations that do not leverage an event-based system typically have to resort to polling for the data they want. The result is that those automations need to manage state (e.g., “what was the last time I made an API call?” or “what was the last object I retrieved?”). Automations leveraging a polling architecture also inherently introduce lag in the system where it is unnecessary because they must establish a polling interval.

Intersight helps support event-driven automations and notifications with two distinct features. For human consumption (or email-based automations), Intersight has an email notification system for new alarms that are created in Intersight. For machine consumption, Intersight has a webhook system that allows for a callback when objects are created or updated. As of the writing of this book, the following objects are supported:

- **cond.Alarm:** Issues that are occurring on endpoints.
- **workflow.WorkflowInfo:** New workflows executed either manually or in automation.
- **tam.SecurityAdvisory:** Security Advisory definition within the system. Notification on this would indicate Cisco has published a new security advisory but not whether it is applicable to your environment.
- **tam.AdvisoryInfo:** The state of an advisory. As of this writing, this object is created only when a user acknowledges an advisory.
- **tam.AdvisoryInstance:** An object describing which item in the account is affected by an advisory. This indicates there is an impact by an advisory in the system.
- **compute.Blade:** Blade servers within Intersight.
- **compute.RackUnit:** Rack servers within Intersight.

With its orchestration capabilities combined with programmability, Intersight enables powerful automation and insights for data center infrastructures. These capabilities enable you to perform actions on multiple devices simultaneously, significantly reducing manual effort and operational overhead. Intersight’s API-first approach prioritizes programmability, which facilitates the development of custom application scripts and seamless IaC integrations.

Providing the insights and automation functions we described in this section requires a robust architecture. In the next section, we will dive deeper into Intersight's SaaS architecture and discuss common deployment scenarios and use cases, along with other important capabilities, such as device onboarding, key management, and integration with Cisco TAC.

Architecture

An important piece of a cloud-based compute platform like Intersight is the overall architecture that supports it. Depending on the SaaS deployment model or one of the on-prem solutions, the architecture can be both vastly different but also share some of the same framework and concepts.

Let's begin with the SaaS cloud offering of Intersight. It is deployed in the Amazon Web Services (AWS) cloud service provider (CSP), but in theory it could be deployed in any provider capable of meeting its requirements. The cloud provider is responsible for maintaining all the physical infrastructure and resources to keep Intersight up and running.

Intersight also follows a common industry deployment model for a cloud product offering, and it is multisite. That means the cloud provider responsible for keeping Intersight up and running has multiple geographic locations. For example, currently, at least one instance of Intersight is running in the U.S., and one instance is running in the EU region. Region-specific instances of Intersight can help customers achieve General Data Protection Regulation (GDPR) and other data privacy compliance requirements, which is a common request for European customers to adhere to local laws and regulations.

The Cisco Intersight DevOps team, however, is responsible for maintaining all the development of the Intersight software itself. Intersight is developed from a standard containerized microservice architecture. Each microservice is responsible for its own function and interacts with other microservices if necessary to complete a task. These microservices get updated over time as new functionality or features become necessary. If just one or a few of these microservices need to be updated, only they will do so; there is no need to update the entire system. For a more detailed discussion of microservices, refer to the “[Microservices and Serverless Architectures](#)” section in [Chapter](#)

2.

When an update to this service is needed, it is scheduled to be updated in the quality and assurance (Q&A) cloud to ensure that no unforeseen issues are discovered. Once the update has been completed, it is pushed out to the production cloud and is accessible to all customers. An example would be a new UI update that includes a new section or feature that can be accessed. It may also require that the endpoints or targets require a device connector update.

The device connector is one of the most important pieces of software in this architecture. It is a lightweight but highly reliable connection to Intersight regardless of the deployment model (SaaS vs. appliance). This software runs on the target and is responsible for establishing a secure WebSocket connection with Intersight. When that connection is established, it will communicate with all the individual microservices running in the Intersight cloud and allow the device to be managed by it.

The device connector lives in the control plane of the server, completely decoupled from the hardware. If it restarts, crashes, or becomes disconnected from Intersight for any reason, the hardware will continue to run in the fashion in which it was configured before that connection was broken. The data plane will not be impacted.

Continuing with the theme of a SaaS deployment model, you should never have to concern yourself with maintaining or updating this firmware. Intersight is responsible for delivering the necessary updates. It will receive an automatic upgrade from the cloud whenever one is required for a new feature or fix.

In the appliance deployment scenario, however, it is bundled with the appliance version. For it to update in this deployment model, it would require an appliance upgrade, and then the appliance would push the device connector update if it included a version higher than the one running on the device. A key benefit to any update to the device connector is that it should be completely without impact to any data plane traffic. Once the device connector receives the update, it will restart and establish a new connection to Intersight.

Next, let's look at the on-prem architecture and the similarities with SaaS—but more importantly the differences. As far as similarities go, both models utilize the same microservices to deliver the same features and functionality of Intersight. These microservices are bundled into the VM itself. They also both rely on the device connector on the target device (or assist) to securely communicate with Intersight. The only difference is that the device connector will not communicate with [Intersight.com](https://intersight.com) in the cloud but with the virtual appliance itself.

The virtual appliance for both the CVA and PVA will be deployed with a fully qualified domain name (FDQN) and IP address that the device connector uses to establish its WebSocket. With this appliance, unlike the cloud, the customer is responsible for the maintenance and management or operational responsibility. The customer must ensure that the appliance is receiving regular updates for new features and fixes, ensuring that it's properly sized for the scale of the environment, its overall uptime, and availability—like any other VM that may be maintained in a data center.

The updates to an appliance are important because they indicate that the microservices running on the appliance are often behind the version that is running in the cloud. That means some features may not be available until a new appliance release is available to contain these updated versions. In contrast, in a SaaS delivery model, the updates are deployed in production as soon as they have been tested and are ready for customer use. It's also important to note that a feature might not be available for an appliance because the scale of the feature might require resources that are possible only in a limitless cloud provider's infrastructure and not capable via a single customer on-prem solution.

By far, the key difference between the SaaS, CVA, and PVA models is their connectivity to various endpoints. With SaaS, every device connector will have secure bidirectional connectivity in some form or fashion out to the public Intersight cloud.

The CVA, as its name implies, is “connected.” This means it still has a connection to the public Intersight cloud, but the data it is sharing and using that connection for is much more limited than with SaaS. See [Figure 14-11](#) for a topological view of a CVA deployment.

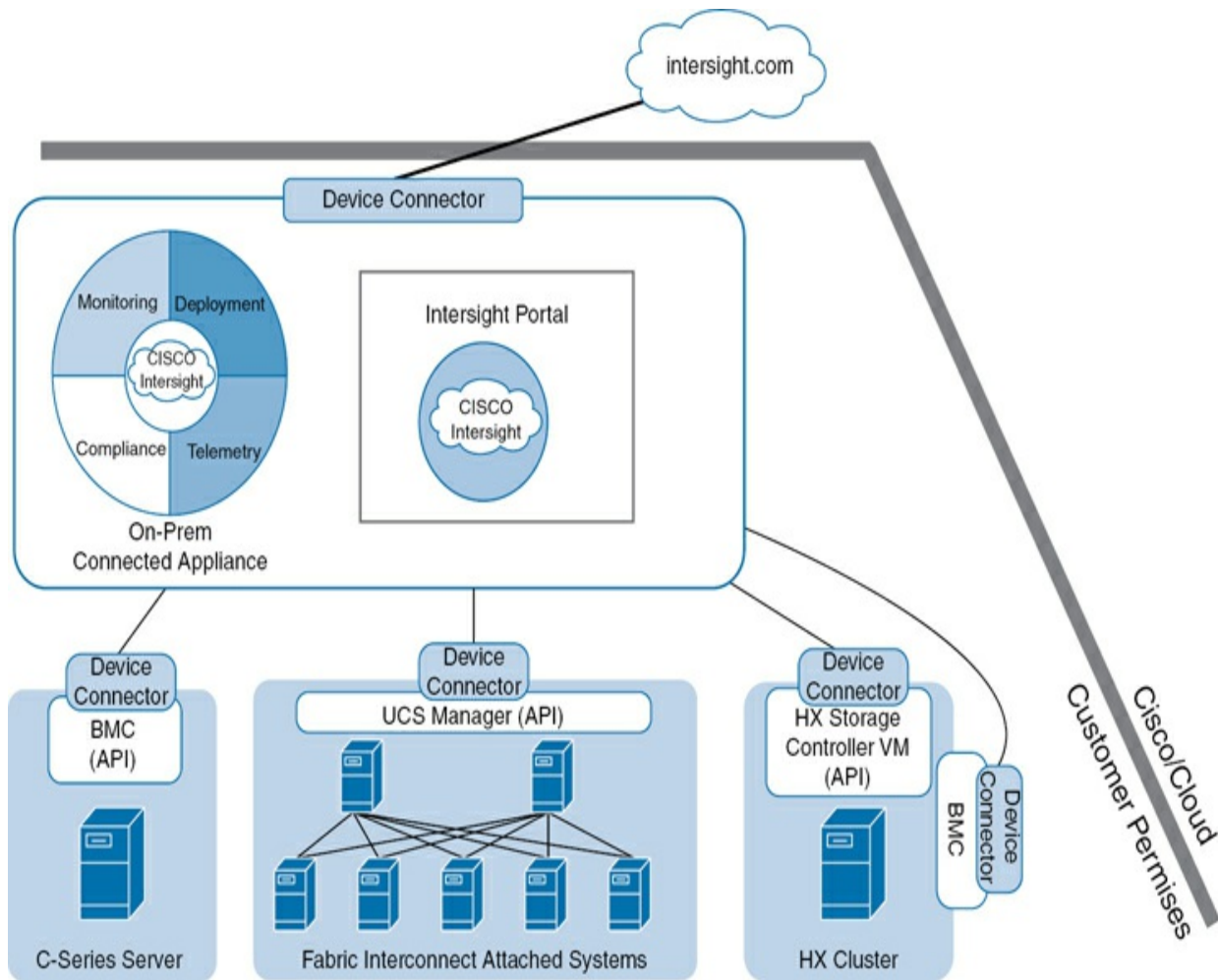


Figure 14-11 Intersight CVA Deployment

In a CVA integration, the overall management, configuration, and monitoring of the devices behind an appliance rely on that communication between the target and the appliance. However, outside connectivity is leveraged for telemetry data, which is required for real-time visibility and updates. Some examples of those features requiring outside connectivity are HCL, advisories, and TAC integration. We will cover TAC integration in an upcoming section of this chapter.

In the case of a PVA deployment, the name also implies its nature—being “private.” Often referred to as “air-gapped,” this on-prem model has zero connectivity to the outside world, as shown in [Figure 14-12](#). All the connected targets are visible only to the PVA, and the public instances of Intersight has no awareness of those devices or even the appliance itself. This model keeps all your data on-premises and managed by the PVA, at the

expense of losing all connected features/benefits.

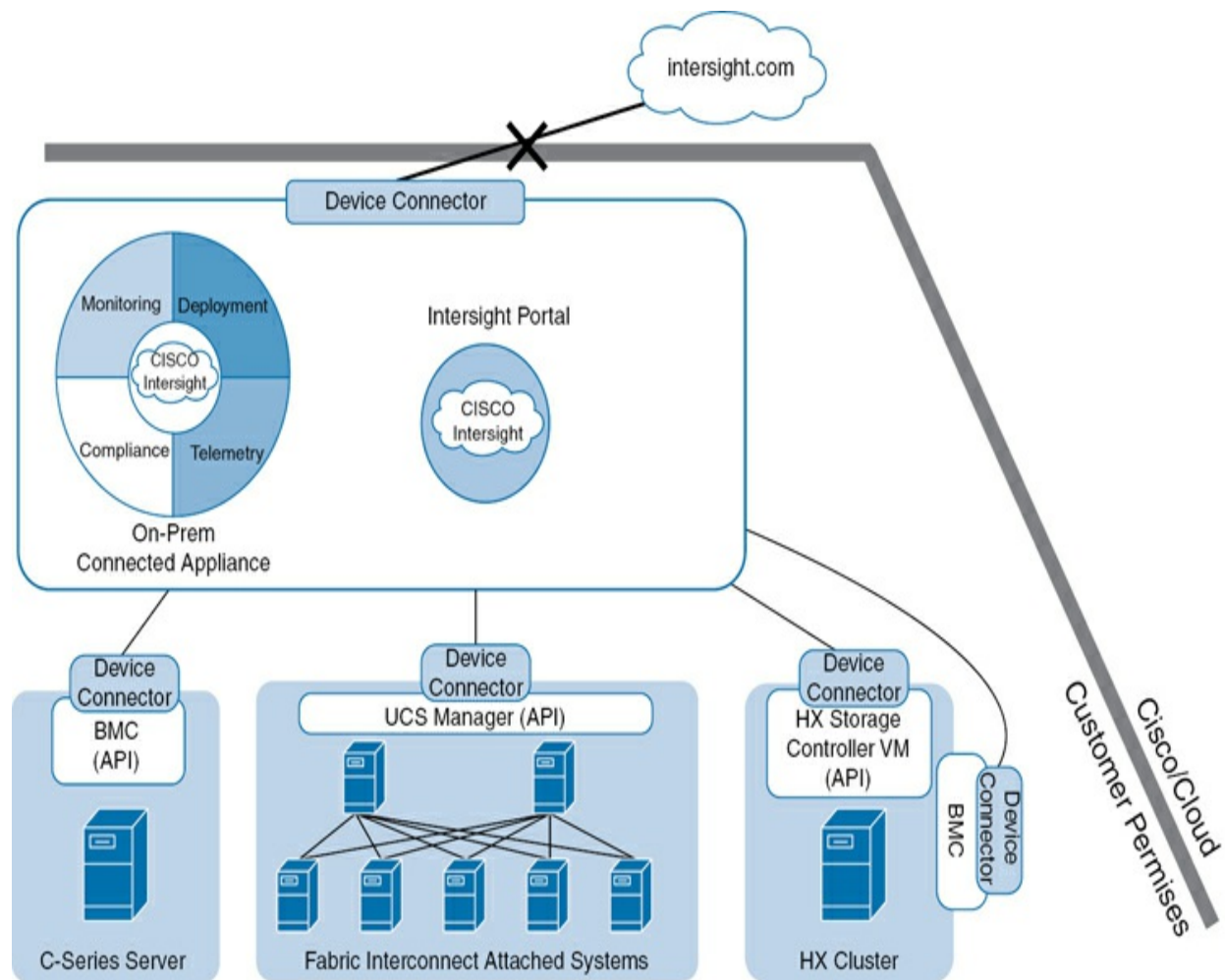


Figure 14-12 Intersight PVA Deployment

The last piece of architecture to discuss is Intersight Assist. If any of the third-party products mentioned in the “[Device Integration \(Cisco and Third Party\)](#)” section are used in this solution, an Assist that is accessible to either Intersight SaaS or one of the appliances must be deployed. In the case of SaaS, Intersight Assist is deployed as a separate VM in a customer’s data center. When an appliance is in use, this Assist is coupled with the appliance itself, and there is no need for a separate VM, as illustrated in [Figure 14-13](#). The Assist will act like a proxy device connector for third-party solutions such as Pure, Hitachi, and vCenter. Just like a normal device connector on a Cisco target, the Assist will set up a secure, always-on, bidirectional WebSocket that allows for any function supported on the device and

Intersight to happen via the standard APIs.

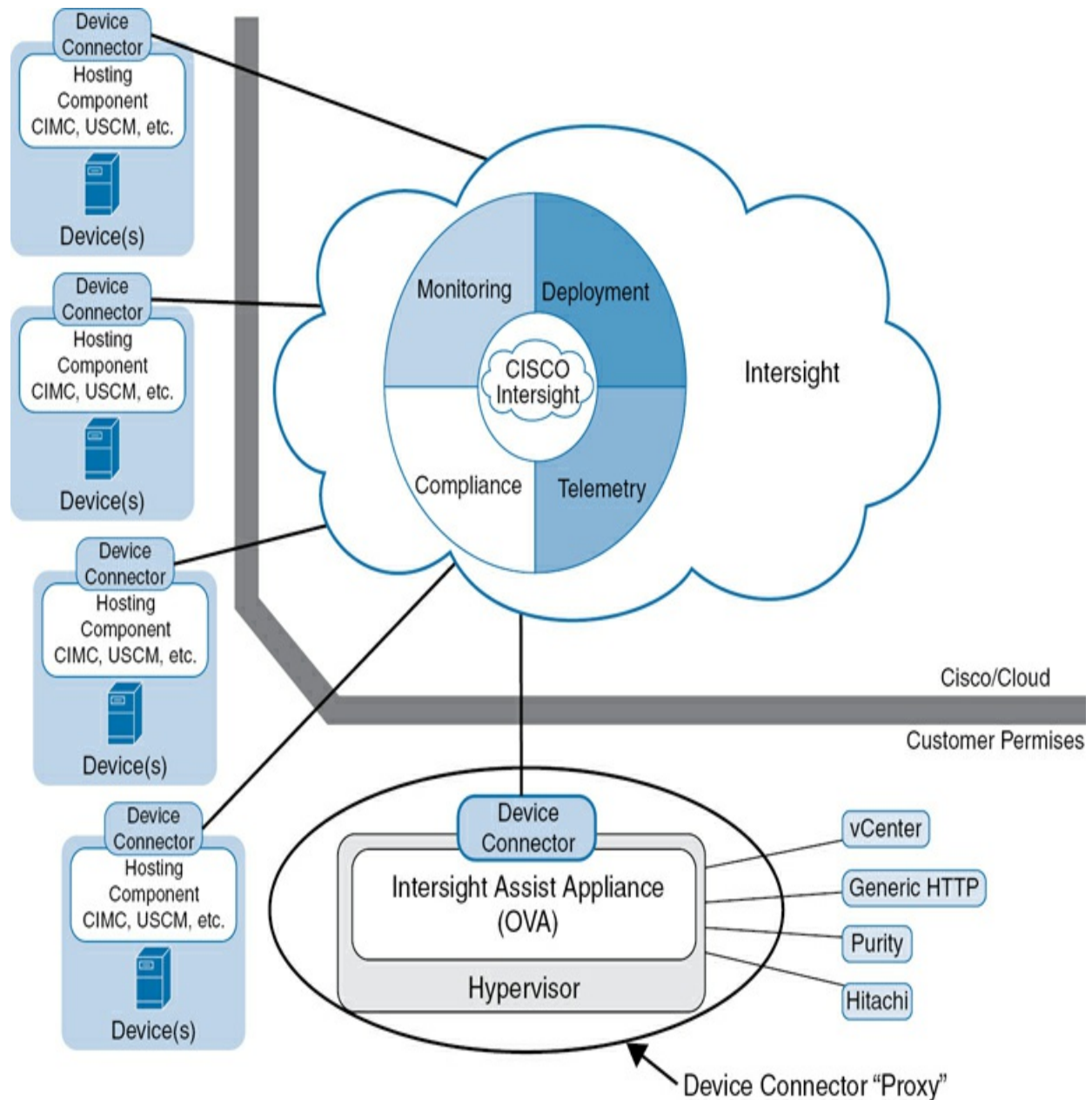


Figure 14-13 Intersight Assist for Integrating Third-Party Products

Common Deployment Scenarios

For customers considering Intersight, the first question they often find themselves asking is “Why?” What benefits would they gain by moving their IT infrastructure to be managed by Intersight. Hopefully, by now, we have

explained a lot of those benefits through the features it provides, but a few key considerations may be top of mind.

First, if you're an existing Cisco UCS customer, you may own multiple different platforms—for example, both standalone C-series servers and B- or X-series in UMM. Managing those platforms requires two separate interfaces. To manage these platforms before Intersight, you could use UCS Central, which could at least manage multiple B-series domains, but that would still leave the standalone C-series servers out of the picture.

This problem only continues to multiply as the complexity and scale of the data center continues to increase and we add other solutions like Hyperflex; third-party OS vendors like VMware, Red Hat, and Windows; or storage vendors like NetApp and Pure. Bringing all these vendors together in one place makes sense and ties back to the theme of a single pane of glass for management and visibility.

Beyond the obvious scale benefits, you may be considering the future of Cisco compute and its data center products in general. Cisco has invested heavily into Intersight as its preferred and future-proof management platform. It will continue to be prioritized and deliver the most up-to-date features.

Arguably one of the most important decisions you will need to make besides utilizing Intersight in the first place is which deployment model to choose. After you look at the architecture behind the SaaS and on-prem offerings, it's time to choose between SaaS, CVA, or PVA.

For most deployments, the general recommendation is to go with SaaS because it will be, by far, the easiest form to deploy. Also, it lets Cisco manage and maintain all the infrastructure needed to keep it up and running, while pushing out new features and fixes without any of the burden on you and no additional cost. This choice will, however, require an external connection to the cloud and can be one of the major considerations for you when deploying SaaS.

Note

Each device connector will need to ensure it has network connectivity to a few different endpoints in the cloud. All those specific URLs are listed in the “Network Connectivity

Requirements” section of the Intersight help page located at https://intersight.com/help/saas/getting_started/system_requirements#cis

These URLs will need to be allowed on any firewall communication between them and the device connector. Most customers and their security departments will also simply not allow direct unfettered network access from a device in their data center to the outside world. Fortunately, the device connector supports the use of a proxy and can provide a single management point for external connectivity needs, enforcing the requirements of a security department. The device connector will also require access to NTP and DNS servers, which typically is not a problem for any normal data center ecosystem. This requirement ensures it has proper time synchronization with Intersight as well as the capability to resolve the various Intersight URLs.

If SaaS is not a viable option, the next best recommendation would be to leverage an appliance and, preferably, the CVA. The CVA will still leverage some of the key features and benefits by allowing some telemetry and visibility to the Cisco cloud. For example, the CVA can pull firmware images from the cloud for updates to servers, Fabric Interconnects, and even the appliance itself. It also enables log collection and proactive RMAs through Cisco support, which will be covered in detail in the “[TAC Integration](#)” section later in this chapter.

The CVA option is usually chosen for a deployment where the same security considerations for an outside connection to the cloud in SaaS are not an issue, but they may have strict change window controls. For example, any form of firmware update must be approved and scheduled during certain periods of time, such as outside a change freeze. Updates could be upgrades to the appliance itself, which can be scheduled, and any device connector updates that would come along with an appliance upgrade. Some customers also choose this option due to data sharing concerns because they want to have some level of control over what data is shared with Cisco.

The final common deployment for Intersight would come in the form of a PVA. This choice is for customers who may have the strictest security requirements and must have a completely air-gapped connection for the devices in their data center. Common examples of customers with this

requirement are the financial industry and the federal government.

You can also find some interesting examples of an outside connection simply not being possible due to extreme bandwidth restrictions or overall lack of connectivity. It should be noted that Intersight's bandwidth and latency requirements are a minimum of 1.5 Mbps and no more than 500 ms of latency, which is extremely reasonable for most customers.

Depending on the types of products in the environment, the next consideration is how the C-series, B-series, and X-series will be deployed. They can be deployed either in UMM or IMM. There are several benefits to leveraging the latest deployment model of IMM that we won't discuss here but two main considerations we will discuss. The first is supported hardware because IMM only starts supporting the more recent hardware generations—for example, M5 servers and above and the fourth-generation Fabric Interconnects and newer. If you have any gear that isn't supported in IMM, that makes the decision pretty easy.

The next consideration is if there are any existing and deployed UMM domains, often referred to as brownfield deployments, or if these domains will be set up from scratch, referred to as greenfield deployments. For greenfield deployments, the simplest recommendation is to deploy them as IMM. This way, you will future-proof these domains and allow for the maximum number features to help automate and speed up deployment. These features assist with OS installation and prep work that can be done in advance, such as preconfiguring domain and service profiles. We will discuss the preconfiguration of domain and service profiles in more detail in the next section, [“Common Workflows and Use Cases.”](#)

For existing environments, deciding to leave them in UMM but connecting them to Intersight still provides a lot of the benefits of visibility that are enabled by the telemetry between targets and Intersight. Even in existing environments, they still can be migrated to IMM if preferred.

The IMM Transition Tool has been developed to help streamline the process of migrating from UMM to IMM. In short, this tool allows all the existing configurations in a UMM domain, such as service profiles, templates, and pools, to be re-created in the customer's Intersight account. It can even migrate configurations found in UCS Central and will leave them in their

account to be reassigned after a domain has been reconfigured to run in IMM. For additional information and details, refer to the IMM Transition Tool Configuration Guide at

https://www.cisco.com/c/en/us/td/docs/unified_computing/Intersight/IMM-Transition-Tool/User-Guide-4-0/b_imm_transition_tool_user_guide_4_0/m_overview_imm_tt_4_0.html.

Once you have decided on a deployment scenario, it's time to start building your chosen solution and getting devices onboarded into Intersight. In the next section, we will take you through some common workflows and use cases that are helpful in better understanding and maximizing Intersight as a SaaS management platform.

Common Workflows and Use Cases

Although you can deploy Intersight in numerous ways, including methods that leverage Intersight's highly programmatic automation, some common workflows and use cases can help you understand what this journey would look like.

Let's look now at a simple example of deploying newly purchased X-series gear in IMM. With X-series being the most recent, innovative, and technically advanced generation of servers, it's an easy decision to couple and deploy them with IMM. Often you may experience downtime as you wait for the physical gear to ship after purchase, or the gear is onsite but is in the process of getting racked and stacked. You can take advantage of this timeframe with Intersight and IMM.

You can take this time to set up your Intersight account so that it is ready for use. Because all the profiles, templates, and pools will be created in Intersight, they can be created ahead of time—even before the IMM domain is set up and claimed in Intersight.

Previously, with UMM, you did not have any access to the UCSM user interface until the Fabric Interconnects were powered on and went through initial setup. When you have created your desired configuration via your service profiles and templates, they can be copied, cloned, modified indefinitely, and reused as the environment scales.

A new UCS concept is domain profiles. They allow you to standardize a common configuration and copy and apply it to multiple domains as you did with service profiles for a server. In the domain profile, you can preconfigure all the settings that a Fabric Interconnect needs, such as creating VLANs, VSANs, DNS settings, and port roles. Port roles specify what a particular port configuration should be and how it will be used.

One good example would be a server role that allows Intersight to discover and inventory the X-series chassis. When this role is configured ahead of time, once the hardware is physically installed and cabled, as per the intended design, discovering those new servers is as easy as assigning the domain profile to the Fabric Interconnects.

Once that prep work has been done, it's time to configure the Fabric Interconnects so that they are integrated into Intersight. After the Fabric Interconnects have been powered on and set to IMM mode, you will have basic access to a new user interface called the device console. It is accessed by the Fabric Interconnects' IP addresses.

The device console is a useful tool. Although IMM devices are intended to be configured entirely in Intersight even without that initial connection, in the event you lose the connection—for example, an ISP outage—the domain will continue to run as it was configured and can still have basic access. You can perform tasks such as powering servers on or off and launching KVM and the API Explorer, among other things.

In the device console, the most important tab is Device Connector. On this tab, you can configure the device connector settings, such as DNS, NTP, or Proxy, to reach Intersight. If the settings are correctly configured and the DC can reach Intersight, these devices can be claimed from the Intersight user interface, which will be covered in more detail in the next section, “[Device Onboarding and Security](#).”

Once the domain is claimed in Intersight and visible, all you have to do is assign the previously created domain profile and server profiles to their designated servers. After that, you could also leverage the OS install feature to rapidly deploy the OS and have the underlying infrastructure deployed.

Device Onboarding and Security

Intersight takes a new and innovative approach to connectivity between on-prem devices and Intersight itself (cloud or appliance). As discussed previously in the “[Cloud-Managed Compute](#)” and “[Architecture](#)” sections, a piece of software named the device connector runs on all endpoints that can talk to Intersight natively. This WebSocket is initiated from the end device and is secured with TLS using AES with a 256-bit randomly generated key.

The WebSocket is initiated from the endpoint itself, so there is no need to make ports or IP addresses available publicly on the Internet. If you need to communicate through a proxy, firewall, or SSL decryptor that changes the SSL certificates provided, you may upload the applicable certs to the device connector itself to enable secure connectivity. By default, the device connector only trusts the certificates presented by <https://svc.intersight.com>.

Device connector connectivity to Intersight as well as all other endpoints is encrypted and established over port 443, except for port 80, which is purely used for access to a certificate revocation list (CRL). CRLs must be loaded over port 80 because they are used to validate the certificate presented by Intersight’s endpoints on port 443. If the CRLs were presented over HTTPS port 443, the server providing the CRL would also need to present a valid certificate, which would, in turn, require a CRL to be available, leading to an infinite loop situation. While access to the CRL over port 80 is not explicitly required, it improves the security posture of the connectivity between the device connector and Intersight because the device connector can validate whether any of the certificates have been revoked by the certificate authority (CA) before the scheduled expiration date.

By default, device connectors attempt to reach out to <https://svc.intersight.com> on port 443 for connectivity. If you want to claim your devices in the EU instance of Intersight, you will communicate with <https://svc.eu-central-1.intersight.com> after claiming the device. It is important to note that those devices will initially need connectivity to svc.intersight.com but will no longer communicate with that endpoint after it is claimed in an account residing in the EMEA instance of Intersight.

When leveraging an Intersight appliance (either CVA or PVA), you do not need any external connectivity from the devices themselves. However, when

you're leveraging the CVA, it will need all the same connectivity as any other device communicating with Intersight SaaS. For more information on the networking requirements for the device connector, see https://intersight.com/help/saas/getting_started/system_requirements.

Although the Device Connector is clearly secure in its connection to Intersight, it's important that the association to the customer account also be trustworthy. Intersight accomplishes this through a process referred to as *claiming an endpoint*. When a device is claimed, it becomes visible in a customer account. Intersight uses a unique claim code coupled with a device ID to ensure a device should be associated with a customer's account.

You can almost think of claiming like you do two-factor authentication. For hardware devices, the device ID is based on the underlying serial numbers. The claim code is an autogenerated unique value that rotates. The claim code is available on the endpoint only when it is connected to Intersight, and both Intersight and the end device know about the claim code. This ensures that customers claiming a device have access and can authenticate to the device itself.

One benefit of having the device connector code separate from the firmware/software running on the endpoint itself is that Intersight can automatically upgrade or update the code on the device connector with no impact to any of endpoints themselves. This auto-upgrade feature helps improve the security posture and endpoint compliance or uniformity.

The first action a device connector attempts to perform when connected to Intersight is check whether or not it should upgrade itself. Intersight will instruct device connectors to download and upgrade their components as needed when new versions are available. For devices connected to a CVA or PVA, the Intersight appliance itself will be the one to instruct the endpoint to upgrade its device connector and will do so only when the appliance itself is upgraded.

Not every piece of hardware needs to be claimed into Intersight. Management controllers are often used as an integration point. As of this writing, devices that run a device connector and can be claimed in Intersight are

- **UCS Manager:** This device allows telemetry for all hardware within the

domain.

- **C-Series Standalone:** Each server talks to Intersight itself.
- **Hyperflex Cluster:** All nodes in the cluster talk to Intersight but talk as a unit.
- **IMM Domain:** Although each endpoint within the domain runs a device connector, it uses a parent device connector on the Fabric Interconnect, and that parent is the only one needed to be claimed.
- **Application Policy Infrastructure Controller (APIC):** The controller itself is claimed and brings with it the rest of controllers, leaves, and spines.
- **Data Center Network Manager (DCNM):** This device reports on all leaves and spines it knows about.
- **UCS Director:** No underlying hardware is pulled into Intersight.
- **Intersight Appliance/Intersight Assist:** Limited telemetry of on-prem devices is shared for connected TAC use cases, but most of the telemetry stays on the appliance.
- **Nexus Switches:** Nexus 9000 switches can communicate directly with Intersight and can be claimed directly.
- **Multilayer Distributed Switch (MDS):** MDS devices can be claimed similarly to Nexus switches.
- **Nexus Dashboard (ND):** All switches Nexus Dashboard knows about will have telemetry in Intersight.
- **Other Devices:** Other devices are supported in Intersight but may not run a device connector itself and may be supported through Intersight Assist. Refer to the “[Device Integration \(Cisco and Third Party\)](#)” section earlier in this chapter for more information.

Note

New integrations are periodically created in Intersight. See the Supported Systems documentation for more information and the latest device support at

https://intersight.com/help/appliance/supported_systems.

Clearly, claiming devices at a controller level reduces the onboarding burden for Intersight customers, but some customers may have a significant amount of hardware to claim in Intersight to have full install-base visibility. Because of Intersight's API-first approach, anything a user can do in the UI, so can automation. In fact, a tool has been created to help claim devices in bulk in Intersight. For more information on this tool, consult <https://community.cisco.com/t5/data-center-and-cloud-knowledge-base/automated-intersight-target-claim/ta-p/3652214>.

The claim process in a SaaS deployment is slightly different than on an appliance (CVA/PVA). Devices automatically attempt to connect to SaaS. However, for a device to direct its communication toward an appliance, it must be reprogrammed. To support the reprogramming of the device connector, instead of entering a claim code and device ID, you enter an IP address or hostname of the endpoint and a username/password. These credentials are used only at device claim time. After claiming, all connectivity is handled across the durable WebSocket created by the device connector.

For customers wishing to utilize the Intersight EU instance, the claim process is identical: Devices are migrated to communicate with the EU instance instead of the U.S. instance. The only requirement is that the Intersight account must be created in the EU instance. As you can see in [Figure 14-14](#), you can view which region (instance) you are connected to by clicking the user icon in the top-right corner of the screen.

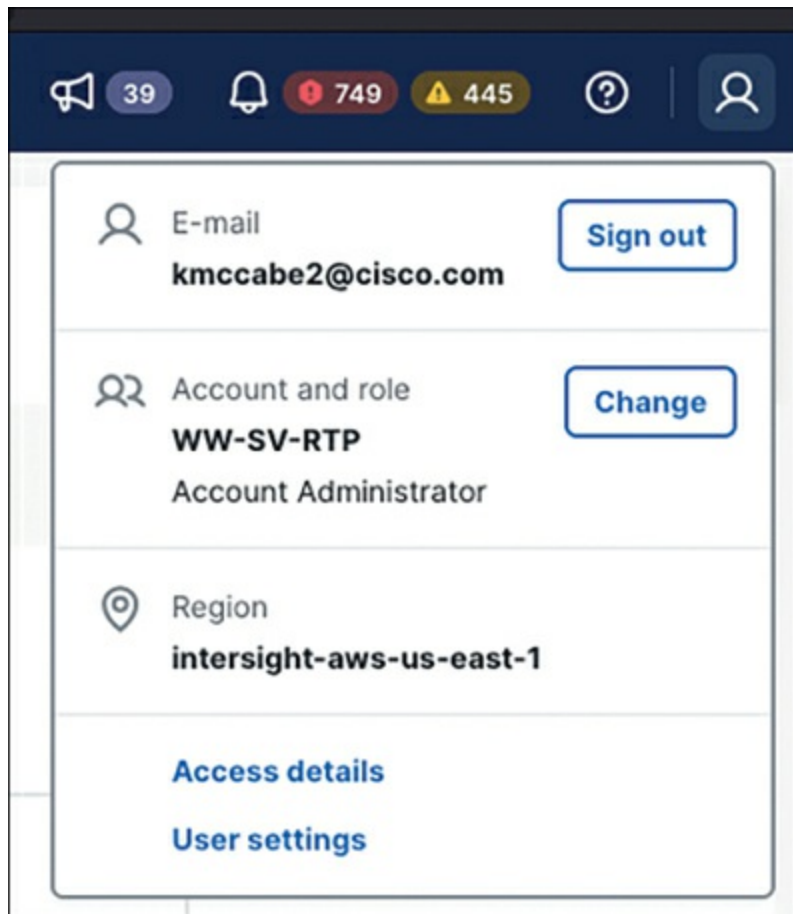


Figure 14-14 Viewing Your Region in Intersight

The claiming process discussed previously is related to devices running a device connector natively. The claim process is slightly different and has different requirements for third-party devices or other devices not running a device connector natively, such as vCenter, Pure, and Hitachi endpoints. Refer to the “[Architecture](#)” section earlier in the chapter for more details on supported devices.

For customers leveraging Intersight SaaS, Intersight Assist must be deployed on the customers’ premises to be able to talk back and forth to the devices without a device connector. The Assist is a lightweight version of the Intersight appliance that runs its own device connector but uses more traditional communication modalities to talk to the third-party devices (such as REST APIs).

When attempting to claim an endpoint that requires an Intersight Assist, you must already have an Intersight Assist claimed to your account. The first

required input to the Target Claim UI is what Intersight Assist will be utilized. You will also have to specify credentials to connect to the endpoint. [Figure 14-15](#) illustrates how a third-party device is claimed using Intersight Assist.

The screenshot shows the Cisco Intersight web interface. On the left is a navigation sidebar with options like Dashboards, Operate, Configure, Analyze, Optimize, System, Targets, Software Repository, Tech Support Bundles, Audit Logs, Sessions, Licensing, Organizations, Resource Groups, Account Details, Access Details, and Settings. The main content area is titled 'Claim a New Target' and 'Claim Pure Storage FlashArray Target'. It includes a warning banner about license compliance. The form contains the following fields: 'Cisco Assist' (a dropdown menu with 'Cisco Assist' selected and highlighted by a red rectangle), 'Hostname/IP Address' (a text input field), 'Port' (a dropdown menu with '443' selected), 'Username' (a text input field), and 'Password' (a text input field with a 'Show' button). There is also a 'Secure Connection' toggle switch which is turned on, and a 'Certificate' section with a 'Select Certificate' link. At the bottom of the form are 'Back' and 'Cancel' buttons, and a 'Claim' button in the bottom right corner.

Figure 14-15 Claiming a Third-Party Device Using Intersight Assist

Note

Customers running an Intersight appliance will not need to install an additional Intersight Assist. All of the functionality of Assist is bundled with the appliance.

When claiming any device within Intersight, you can select a set of resource groups to make the device available within. Resource groups are simply a collection of devices that are used in the broader RBAC strategy within

Intersight. For further details on how RBAC works in Intersight, see the following section.

User Management

Any user with a valid Cisco account can create a new Intersight account, and users can belong to multiple Intersight accounts. The user creating the account automatically becomes an account administrator for that account (although that role can be changed later). Additional users can be added to Intersight manually or configured via a group. For group access, the identity provider (IdP) must return the user in the memberOf portion of the Security Assertion Markup Language (SAML) response during the login flow. For more information on the login flow expectations for SSO providers, see https://intersight.com/help/saas/resources/sso_in_intersight_overview.

By default, the only IdP available on Intersight is the Cisco IdP. It's not listed when viewing the single sign-on (SSO) IdPs in Intersight and cannot be removed from the Intersight account. Customers can, however, add additional IdPs to their Intersight account to support a more seamless SSO experience for users in Intersight who would have already authenticated to their SSO provider. While Intersight can support any IdP complying with SAML 2.0 standards, other sites that Intersight may link to (for example, Support Case Manager to open TAC cases) will currently still require a login using the Cisco IdP.

Now that we have discussed users accessing Intersight itself, let's explore how users can be allowed or restricted from seeing end devices. Intersight has a robust RBAC strategy that can support many combinations and permutations of permissions and capabilities.

Intersight has a hierarchical approach that allows multiple levels of groupings with sharing of resources possible. An Intersight account is made up of organizations, and within organizations are one or more resource groups. As illustrated in [Figure 14-16](#), resource groups can be in multiple organizations if configured. When claiming a new device, you choose which resource group or groups a device belongs to, and by default, a device is added to "All" type resource groups. Resource groups can be configured to cover all targets (end devices) or a selected set of targets. For IMM domains, users can

choose to include only some of the devices within the domain in the resource group.

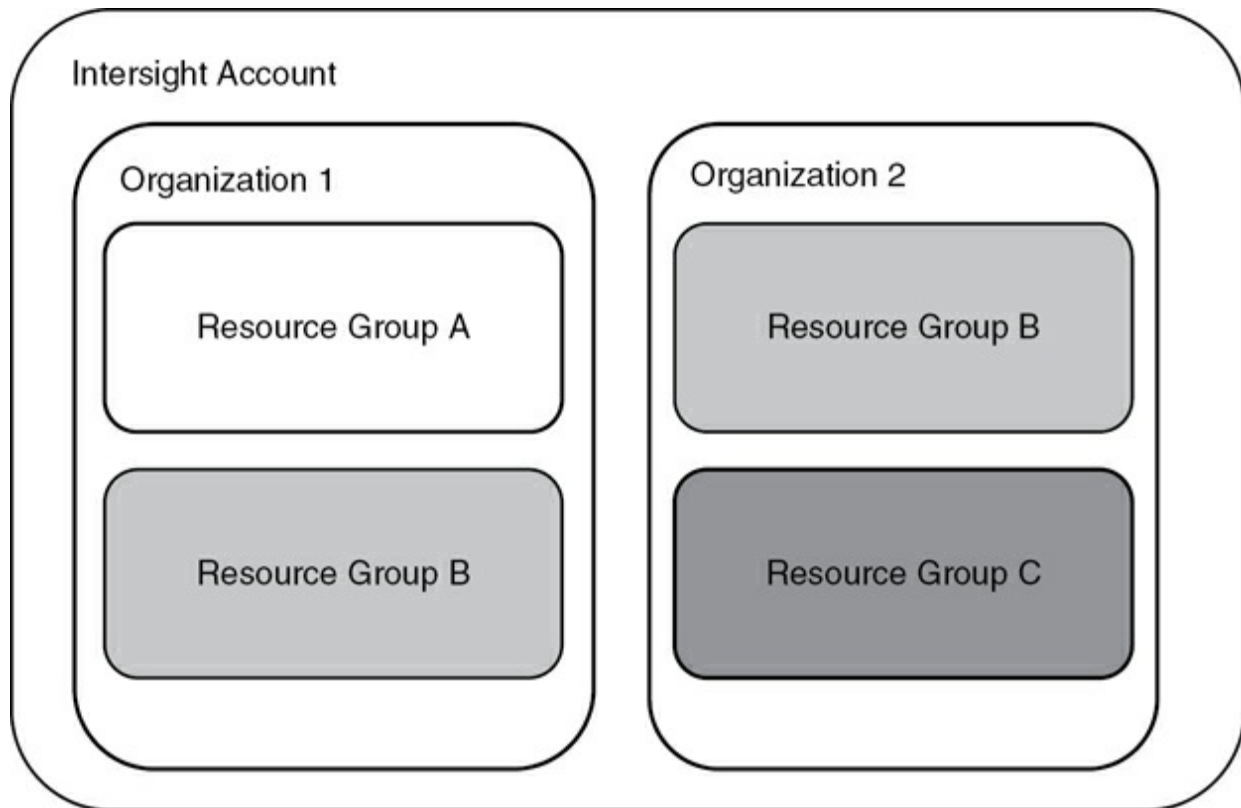


Figure 14-16 Relationship Between Resource Groups and Organizations in Intersight

Just as devices have a hierarchy, so do users. Users are associated with one or more roles, and each role has a set of privileges that are allowed in a particular organization. This tie into organizations is key here because that's how users are associated with end devices. [Figure 14-17](#) depicts the association between users, roles, privileges, and organizations.

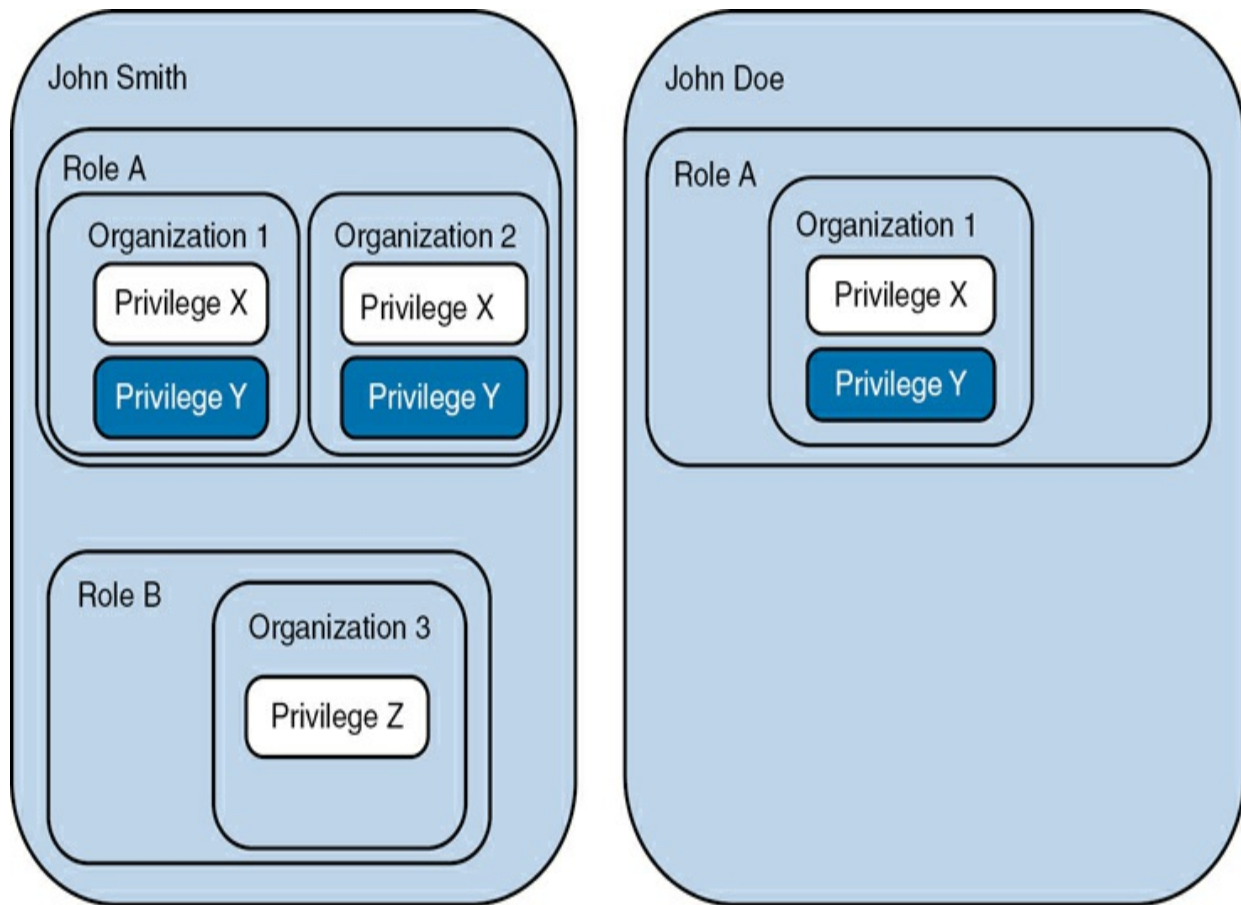


Figure 14-17 Associating Users, Roles, Privileges, and Organizations in Intersight

Although the underlying RBAC model involves significant complexity and inheritance, Intersight's design simplifies its administration for users. Intersight comes with several built-in roles, such as account administrator, server administrator, and read-only. For a full list of built-in roles, see https://intersight.com/help/saas/system/role_based.

In addition to built-in roles, a default organization and a default resource group are precreated. Initially, all devices will be available in the default resource group and, therefore, the default organization. So, only organizations that need detailed RBAC need to configure anything in Intersight, and many customers use the built-in defaults.

There are many security-based configurations within Intersight. Covering all of them is beyond the scope for this chapter, but here are some best practices that you should consider:

- Every single Intersight account should have more than one user with the role of account administrator.
- Every Intersight account utilizing additional IdPs should have at least one, preferably two users, created with the Cisco IdP that have the account administrator role.
- Using multifactor authentication (MFA) login requirements for users logging in with the Cisco IdP is a good security practice.

For more security configuration guidance, see the RBAC documentation for Intersight online

[https://intersight.com/help/saas/resources#role_based_access_control_\(rbac\)_](https://intersight.com/help/saas/resources#role_based_access_control_(rbac)_)

TAC Integration

One of the biggest benefits to being connected to a SaaS system is that when a problem occurs or if you have a question, the Technical Assistance Center (TAC) can access most of the information it needs without customer involvement. This means no more sitting in lengthy Webex meetings, no need to get files out of a virtual network computing (VNC) session, and no more trying to look up where that Trivial File Transfer Protocol (TFTP) server you use once a year resides.

TAC can not only retrieve the telemetry available on SaaS for your devices but also generate on-demand tech support bundles and other diagnostic data. Not only can humans generate data on demand, but automation running in TAC leverages Intersight's robust APIs to *automatically* generate tech support bundles for the device the TAC case was opened for. Automated tech support generation couples nicely with TAC's knowledge system that can leverage known signatures to automatically diagnose problems, sometimes within minutes of opening a case. This end-to-end flow is detailed in [Figure 14-18](#).

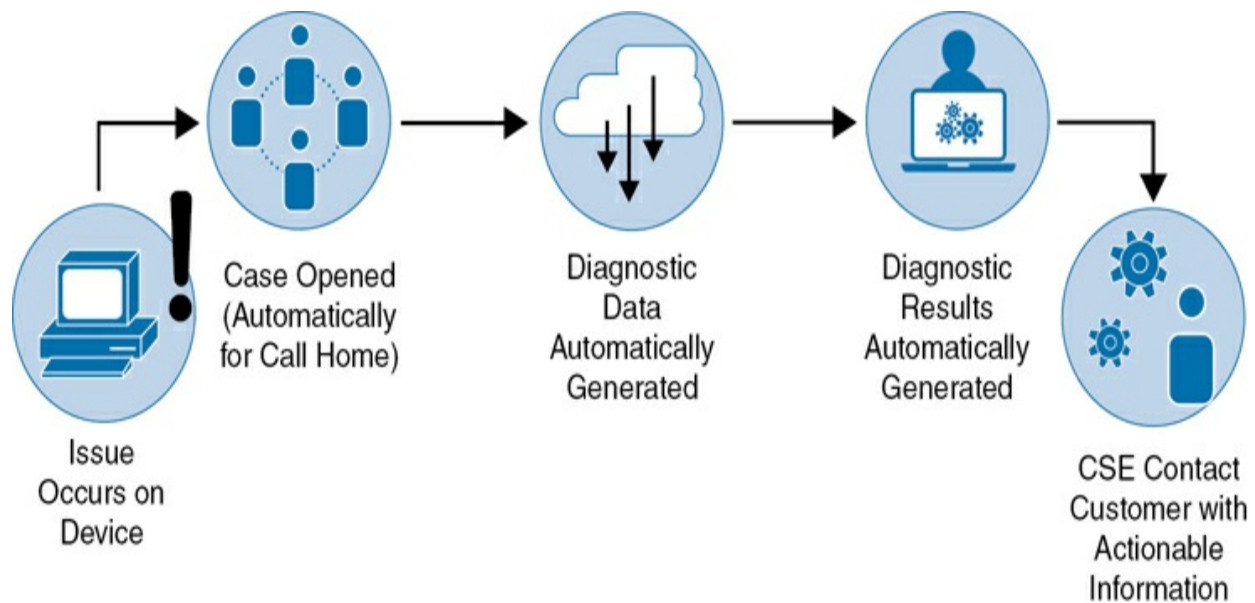


Figure 14-18 Intersight TAC Integration for Automated Case Open, Diagnostic Data Collection, and Analysis

Reducing the friction of getting help from Cisco is great, but wouldn't it be nice if Cisco *proactively* could check on the health of your devices because it has a wealth of telemetry? That's exactly where Proactive RMA comes in, as shown in [Figure 14-19](#). Cisco TAC integrates seamlessly with Intersight APIs and telemetry to monitor for hardware failures and issues within your network. Today Proactive RMA covers

- Memory failures
- UCS drive failures (drives connected via a RAID controller)
- Hyperflex drive failures (drives connected with a host bus adapter [HBA] or Just a Bunch Of Disks [JBOD] configuration)
- Nutanix drive failures (Note that Nutanix drives are detected when connected via Prism, but Intersight telemetry can assist with the case-open experience.)
- C-Series server fan failures
- Fabric Interconnect fan failures

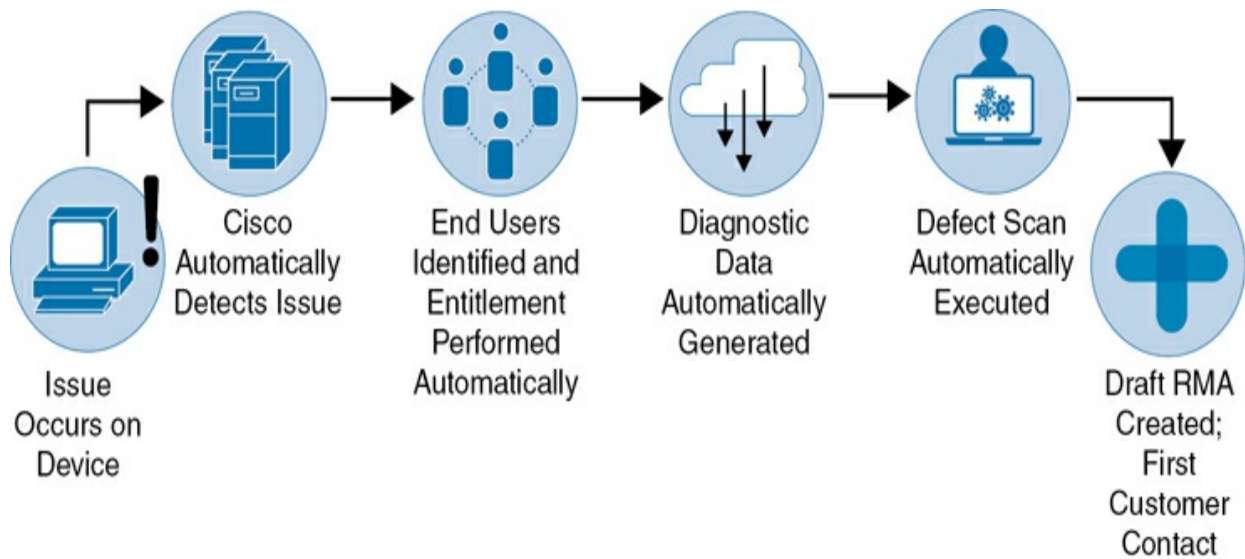


Figure 14-19 Intersight Proactive and Automated RMA Process

Both Proactive RMA and Connected TAC (automated tech support generation) are enabled by default. Customers can choose to opt out of either and can configure some parameters for Proactive RMA, such as the individual whom the case is entitled/opened with. For more information on Proactive RMA, see <https://www.cisco.com/c/en/us/support/docs/servers-unified-computing/intersight/215172-proactive-rma-for-intersight-connected-d.html>.

Additionally, both Proactive RMA and Connected TAC are supported on the connected virtual appliance. For the CVA use case, minimal telemetry is shared with Intersight SaaS to support routing the diagnostic data generation requests to the applicable CVA. The CVA handles requesting the endpoint device to generate the necessary data. The diagnostic data is transferred to the appliance and then back to Intersight SaaS, where it lives temporarily while being attached to the applicable TAC case.

If you have ACI Connected devices within your Intersight account, proactive cases may be opened on your behalf when critical issues are detected. Like other TAC integrations, this capability is available by default and requires no license. For more information on faults covered and configuration options, see <https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/217576-about-proactive-aci-engagements.html>.

Clearly, even with a wealth of telemetry and automation capabilities, not all issues can be handled in automation, but Intersight can help even in instances where you need to open a case manually. From the Intersight user interface, you can open a case from the context of a specific device or against Intersight in general (for non-device-specific questions). In all circumstances, creating a case from Intersight improves both your and TAC's experience. Within Intersight's user interface, you can open cases from

- Servers
- Chassis
- Fabric Interconnect
- Hyperflex clusters
- Top-right Help Icon in Intersight

Anytime a case is opened from Intersight, some specific telemetry is extracted, as well as the context where you are opening the case from. TAC then knows right away where the problem is occurring and with what device, so it can start investigating immediately.

Aside from contextual awareness, you can skip entering a product serial number or other entitlement information and proceed directly to describing the problem. This approach reduces the time it takes to open a TAC case and also ensures the entitled device is indeed the device having the problem. Opening TAC cases from Intersight is the recommended way for all Intersight customers to open a case. The main reason is that entitlement with an accurate serial number or subscription is key to TAC being able to pull the necessary data throughout the case lifecycle.

The absolute worst time to find out that a device is not covered by a TAC service contract is in your moment of need. A device is hard down, or an outage has occurred, and you need help immediately. Lack of contract coverage can lead to a delay when every moment matters. Intersight can help here as well. With an integration to TAC systems, Intersight can analyze all the hardware in your install base and ascertain what contract it's covered by, when the contract expiration is, and whether any assets are uncovered. Intersight can filter on contract status in applicable tables. Of course, the status can also be obtained via the API, and custom widgets can be added to

dashboards and table views, summarizing the health of coverage.

Along with contract status, customers can sometimes be surprised when devices reach their end of life. Within the Intersight Advisories framework, there are currently three types of advisories:

- Security advisories
- Field notices
- End-of-life advisories

While the end-of-life advisories are not necessarily a TAC integration, this framework aims to prevent any surprises when customers attempt to open a TAC case. These advisories are available to devices with a valid Essentials license and can help customers plan to lifecycle the hardware within their install base effectively.

All this TAC integration is great, but what about customers who obtain support from Cisco partners? Intersight has a curated privilege called Support Services. The intent behind this role is to give anyone needing to support a device the access needed to appropriately diagnose the issue. This service enables assigned users to

- View assets in Intersight
- Collect and download tech support bundles
- Cross-launch Cisco UCS Manager, Cisco IMC, and HyperFlex Connect as a read-only user
- Access all application APIs in a read-only mode
- Access IAM APIs like IdP, domain names, IP access management, users, groups, roles, organizations, resource groups, and sessions in read-only mode
- Create and use API keys and OAuth tokens, which is especially useful for partners wishing to enable automated analytics
- Access metrics explorations

Summary

Intersight represents a pivotal advancement in the realm of cloud-managed compute management software, offering a robust suite of features and capabilities designed to streamline and enhance data center operations. Throughout this chapter, we have explored the features and benefits of Intersight, including its capability to provide centralized management, automation, and insights across diverse environments.

As organizations increasingly adopt both Cisco and third-party vendors' data center products, the need for a single pane of glass becomes crucial. Intersight not only addresses these needs with its scalable architecture and intuitive interface but also empowers organizations to optimize performance, reduce operational costs, and respond swiftly to incidents.

By leveraging Intersight, businesses can achieve greater agility, resilience, and efficiency in managing their data center at scale. Its integration with multiple products and vendors with an API-first mentality allows businesses to future-proof managing their infrastructure as the rapid development and delivery of new features continues and Intersight evolves.

If you are interested in diving deeper into the topics discussed in this chapter and learning more about Intersight and all its capabilities, we encourage you to check out *Cisco Intersight: A Handbook for Intelligent Cloud Operations* by Matthew Baker et al.

References

- Baker, Matthew et al. *Cisco Intersight: A Handbook for Intelligent Cloud Operations*. Cisco Press, 2023.