

Service mesh or API management?

Deploy a comprehensive service architecture for your organization

See what's inside

About this e-book

Chapter 1

Understand APIs and microservices

Chapter 2

Identify traffic direction and domain boundaries

Chapter 3

Deploy API management and service mesh solutions

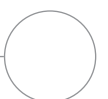
Chapter 4

Build a complete service architecture with Red Hat

Chapter 5

Discover success across industries

Ready to get started?



About this e-book

Our increasingly digital world requires organizations to adapt their business and IT strategies, including application development and deployment. Application programming interfaces (APIs) and microservices can help your organization modernize your applications to improve speed, agility, and efficiency. In fact, 58% of organizations are adding a layer of APIs to modernize their applications.¹ Meanwhile, 61% of organizations have used microservices for more than one year, and 28% have used them for more than three years.²

Even so, building a service management architecture that optimizes use of APIs and microservices can be confusing. Many organizations consider either API management or service mesh solutions but may not realize that these two technologies work best together.

This e-book provides essential knowledge and expert guidance for building an effective service management architecture that encompasses both API management and service mesh solutions.

Building a comprehensive service management architecture can help your organization:



Accelerate new service and API development by 2x.³



Speed application launch and update cycles.⁴



Deliver real-time, data-based services to users.⁵

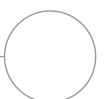
¹ F5 Networks. "State of Application Strategy Report 2021," March 2021.

² Loukides, Mike and Steve Swayer. "Microservices Adoption in 2020." O'Reilly, 15 July 2020.

³ Red Hat case study. "Amsterdam Airport Schiphol builds agile cloud with Red Hat OpenShift," 3 August 2017.

⁴ Red Hat case study. "Leading UAE bank builds competitive private cloud with Red Hat," 1 May 2019.

⁵ Red Hat case study. "Uruguayan city builds apps with APIs and Red Hat OpenShift," 22 January 2020.

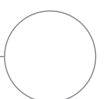
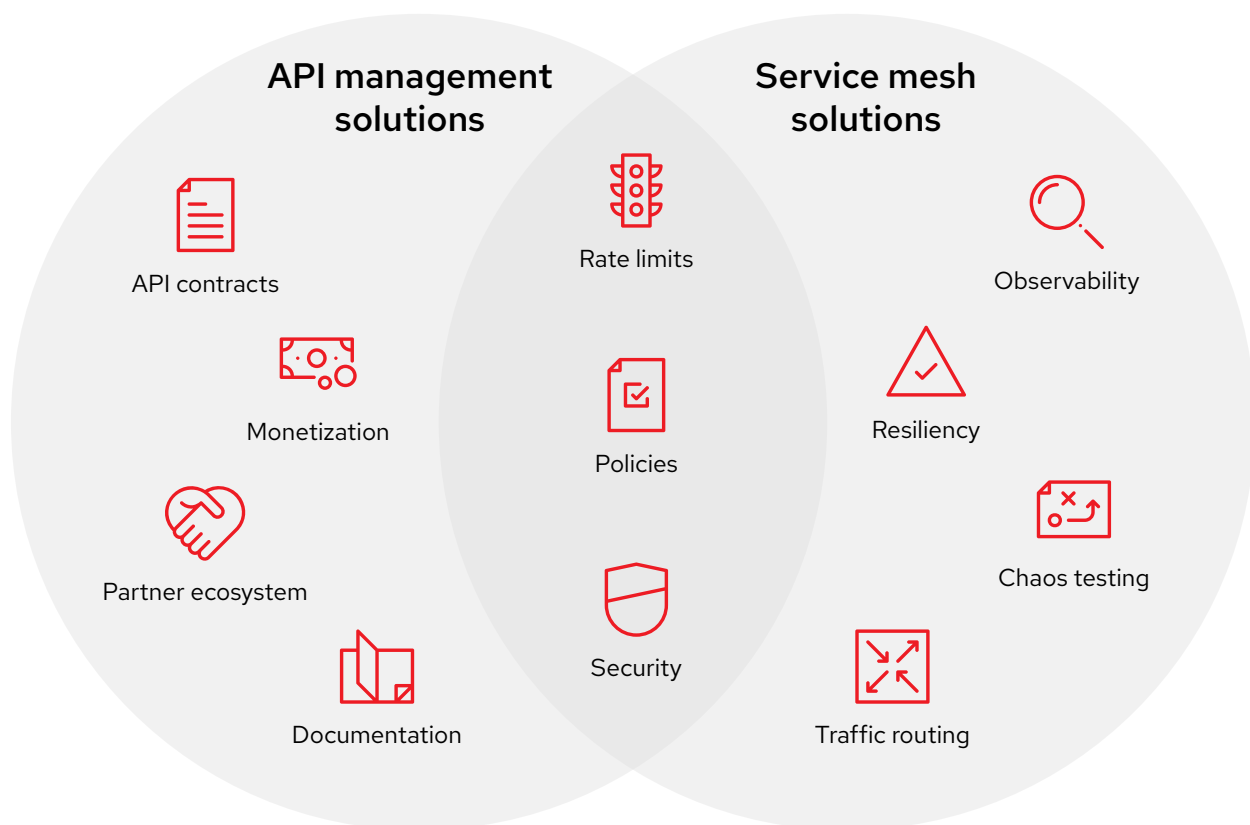


Understand APIs and microservices

Many organizations want to move to API-centric and microservices-based IT architectures to improve speed, agility, and efficiency. In fact, many organizations already use APIs to interact with their customers, partners, and other external users. Consistent, effective management of these interfaces and services is critical for successful deployment and use.

However, determining which approaches and tools to use within your organization can be confusing. Most organizations look for either an API management or a service mesh solution to accomplish this. These solutions have many things in common, but also many differences. While many organizations approach this as an either-or decision, these two technologies are actually complementary and work together to deliver a complete service management architecture.

The following chapters discuss the similarities and differences between API management and service mesh approaches, when to choose one over the other, and how to set up a comprehensive service management architecture using both solutions together.



What are APIs and microservices?

Microservices and APIs are at the core of **cloud-native application development** – an approach for rapidly developing innovative application components to meet new and changing business needs.

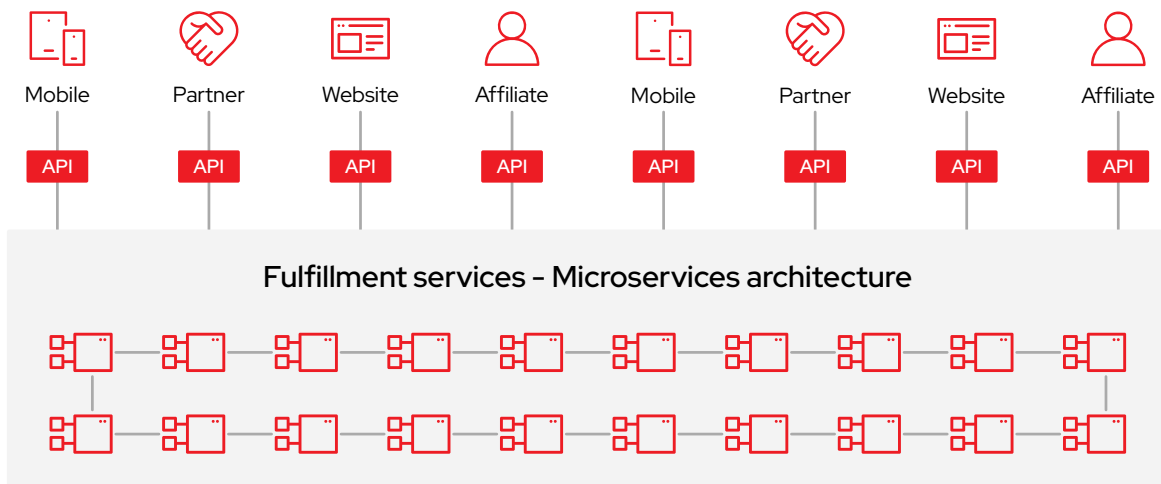
APIs

APIs provide an interface into your business capabilities, allowing consumers to access services in a controlled manner. They consist of tools, definitions, and protocols for integrating application software and services. APIs let your products and services communicate with other products and services without needing to build new connectivity infrastructure. They can be created for internal, partner, or even public use, allowing others to develop applications that incorporate your services and providing additional revenue streams. As a business-focused construct, APIs serve as contracts between service providers and service consumers.

Microservices

Microservices deliver specific sets of functions and responsibilities as part of a larger application or business process. They are an architectural approach to building applications. As a deployment pattern, microservices can be combined and orchestrated to create complete applications and processes. They are distributed and loosely coupled, so changes to one microservice will not affect other microservices or break the entire application. Additionally, each microservice has its own life cycle and can be managed and scaled independently. Finally, microservices can be delivered either through a service mesh or via an API.

Business services - API connections

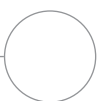
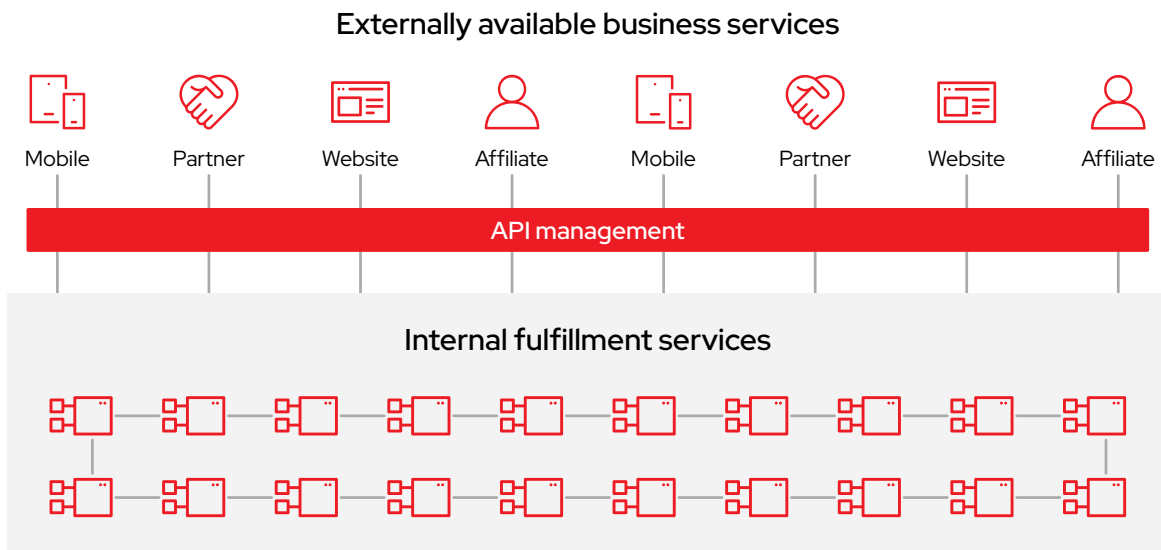


What is API management?

API management refers to the processes for exposing, controlling, and analyzing the interfaces that connect applications and data across your enterprise and across clouds. It allows organizations that create APIs or use others' APIs to monitor activity and ensure that both developer and application needs are met. Managing APIs also helps you make sure your interfaces are used in compliance with the appropriate corporate, regulatory, and security policies.

API management encompasses everything needed to use APIs in a business strategy:

- ▶ Monetization
- ▶ API consumer management
- ▶ Ecosystem creation and administration
- ▶ Product development
- ▶ Partner outreach and onboarding



How does API management work?

API management centralizes control of your API program from analytics and access control to monetization and developer workflows. At a minimum, API management solutions should provide:

- ▶ **Authentication.** Create and check user keys and login credentials.
- ▶ **Authorization.** Authorize access and use via your existing ID management systems.
- ▶ **Traffic management.** Throttle and rate limit API use to protect your interfaces and services from abuse by consumers and some denial of service (DoS) attacks.
- ▶ **Security.** Encrypt communications and manage security certificates.

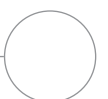
Many API management solutions also provide:

- ▶ Developer portals.
- ▶ API life-cycle management.
- ▶ Analytics.
- ▶ Support for API monetization.

Comprehensive API management solutions simplify discovery and documentation of your interfaces and assets, provide flexible access controls for different users, and deliver straightforward consumption billing and invoicing.

Who uses API management?

API management is ideal for teams that want to deliver services beyond their domain to other internal groups or external users.



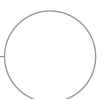
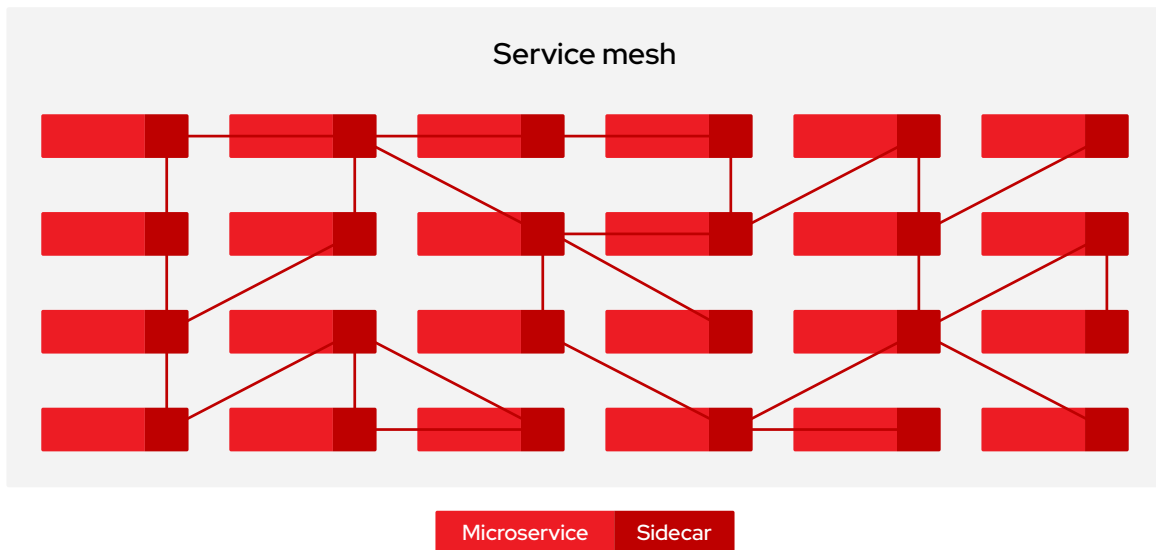
What is a service mesh?

Service meshes provide connectivity between application services and add capabilities like resiliency, security, observability, routing control, and insights. They control how different parts of an application share data with one another. Unlike API management, a service mesh is a dedicated infrastructure layer that runs alongside applications. This visible infrastructure layer can document how different parts of applications interact, making it easier to improve communication and avoid application downtime.

Service meshes route requests from one service to the next, optimizing how all the moving parts work together. Within cloud-native application development approaches, they allow you to assemble large numbers of discrete services into functional applications. Service meshes also simplify development to increase productivity, speed, and quality:

- ▶ Distributed request tracing provides visibility into service connections, streamlining issue discovery and troubleshooting.
- ▶ Automatic request rerouting can avoid failed services, making applications more resilient.
- ▶ Performance metrics help you optimize communication in your runtime environments and improve your user experience.

Service mesh implementation is transparent to developers, allowing them to focus on creating valuable applications and features rather than rebuilding or connecting standard underlying services like security and routing.



How do service meshes work?

Service meshes take the logic and policies governing service-to-service communication out of individual services and abstract it to a layer of infrastructure. To do this, service meshes are built into applications as an array of network proxies.

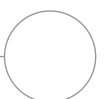
Requests are routed between microservices through these proxies. For this reason, the individual proxies that make up a service mesh are sometimes called *sidecars*, since they run alongside each service rather than within them. Taken together, these proxies – decoupled from each service – form a mesh network. As a result, they can be updated and changed independently of the services they support and connect. Common proxy policies and functions include:

- ▶ Authentication.
- ▶ Authorization.
- ▶ Encryption.
- ▶ Security.
- ▶ Traffic management.

Service mesh infrastructure layers use control planes and data planes to manage services. The control plane allows you to define rules and policies to govern your services. The data plane contains the actual proxies and applies the appropriate rules and policies to your services at runtime.

Who uses service meshes?

Service meshes are ideal for development teams that need to access services through network interfaces and protocols.



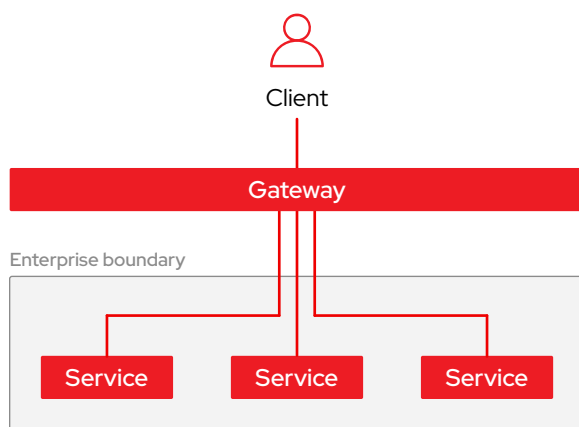
Identify traffic direction and domain boundaries

Identifying the direction of traffic between your services and interfaces is the first step in deciding whether to deploy an API management or service mesh solution.

North-south traffic

North-south traffic patterns connect services and applications with external clients. This traffic typically:

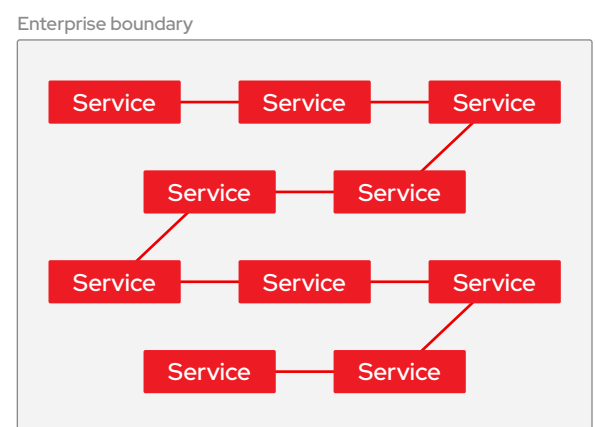
- ▶ Involves connections to service consumers beyond your enterprise boundary.
- ▶ Requires security, policy enforcement, access control, and analytics capabilities.
- ▶ Is governed by formal contracts between service providers and service consumers.
- ▶ Scales to tens of service connection points.



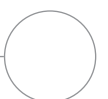
East-west traffic

East-west traffic patterns link individual microservices to form complete applications. This traffic usually:

- ▶ Involves multistage routing to orchestrate services within your enterprise boundary.
- ▶ Requires mutual security and authorization controls and tracing and observability functions.
- ▶ Is governed by informal contracts between service providers and service consumers, if needed.
- ▶ Scales to thousands of service interfaces.



Based on these key characteristics, north-south traffic patterns align well with the capabilities of API management solutions, while east-west traffic patterns correspond more closely with service mesh solutions.

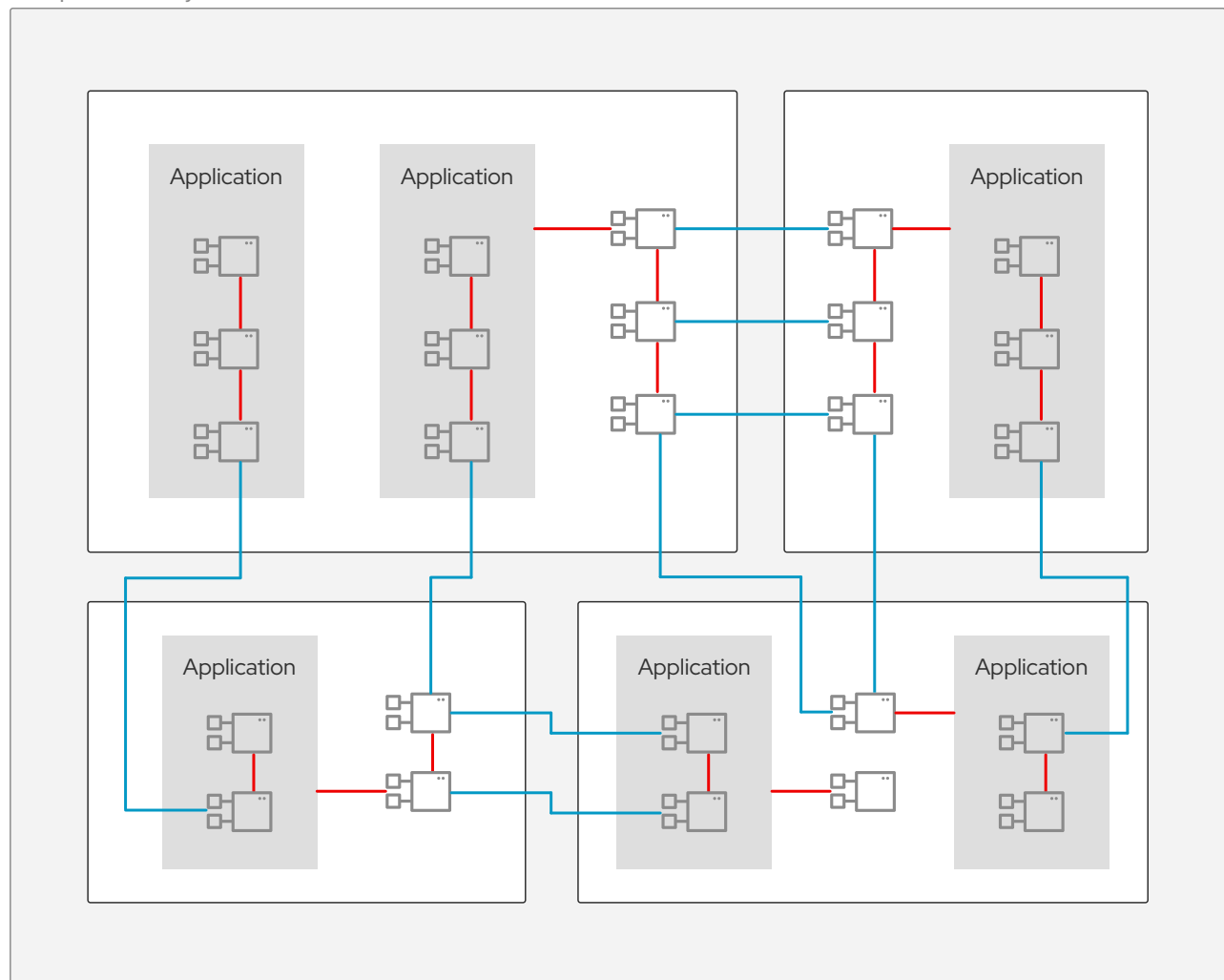


Introducing domain boundaries

Traffic direction provides a straightforward guide for when to choose API management or service mesh solutions. However, most organizations are not so simple. Typical organizations contain multiple groups that create and manage their own services and interact with other teams and external parties.

Domain boundaries can help you divide your organization into smaller, more manageable areas. Much as your enterprise boundary denotes the perimeter of your overall organization, domain boundaries designate the perimeters of groups within your organization.

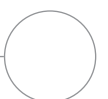
Enterprise boundary



— Domain boundary

— Intradomain connection

— Interdomain connection



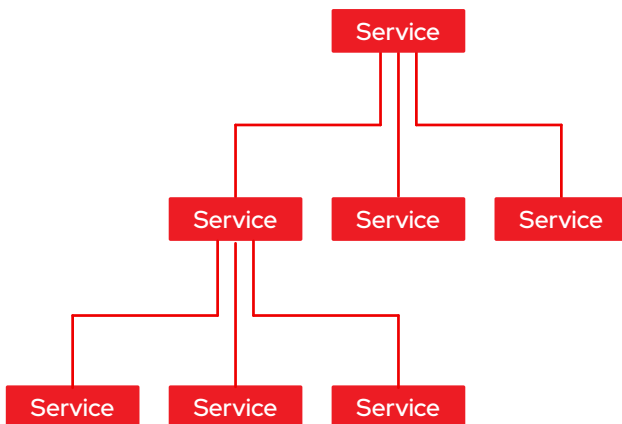
Interdomain versus intradomain traffic

The same north-south and east-west traffic patterns that occur in relation to your enterprise boundary also apply to domain boundaries within your organization. As a result, you should generally choose API management for interdomain traffic and service meshes for intradomain traffic.

Interdomain traffic

Interdomain traffic crosses domain or enterprise boundaries to connect services with consumers beyond your group or team. Interdomain traffic follows north-south traffic patterns:

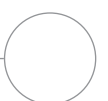
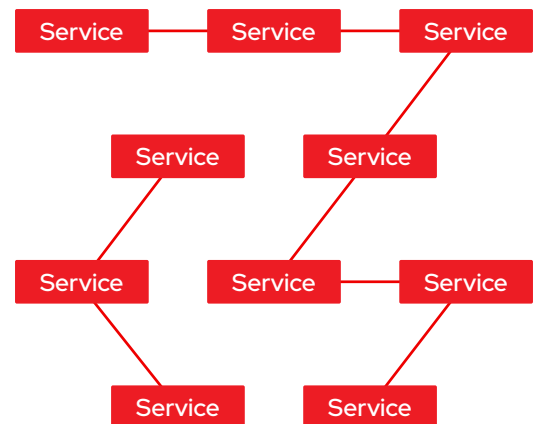
- ▶ Hierarchical 1:N connection structures
- ▶ Separate service providers and service consumers
- ▶ Authorization and authentication required
- ▶ Formal use contracts needed
- ▶ Guided service discovery, accessible developer portal, and formal documentation



Intradomain traffic

Intradomain traffic stays within domain and enterprise boundaries to link individual microservices. Intradomain traffic typically conforms to east-west traffic patterns:

- ▶ Non-hierarchical 1:1 connection structures
- ▶ Service providers and consumers within same team
- ▶ Authentication required
- ▶ Implicit or informal contracts, if any
- ▶ Internal documentation within code



Deploy API management and service mesh solutions

API management and service meshes can be used together to create a comprehensive service management architecture within your organization.

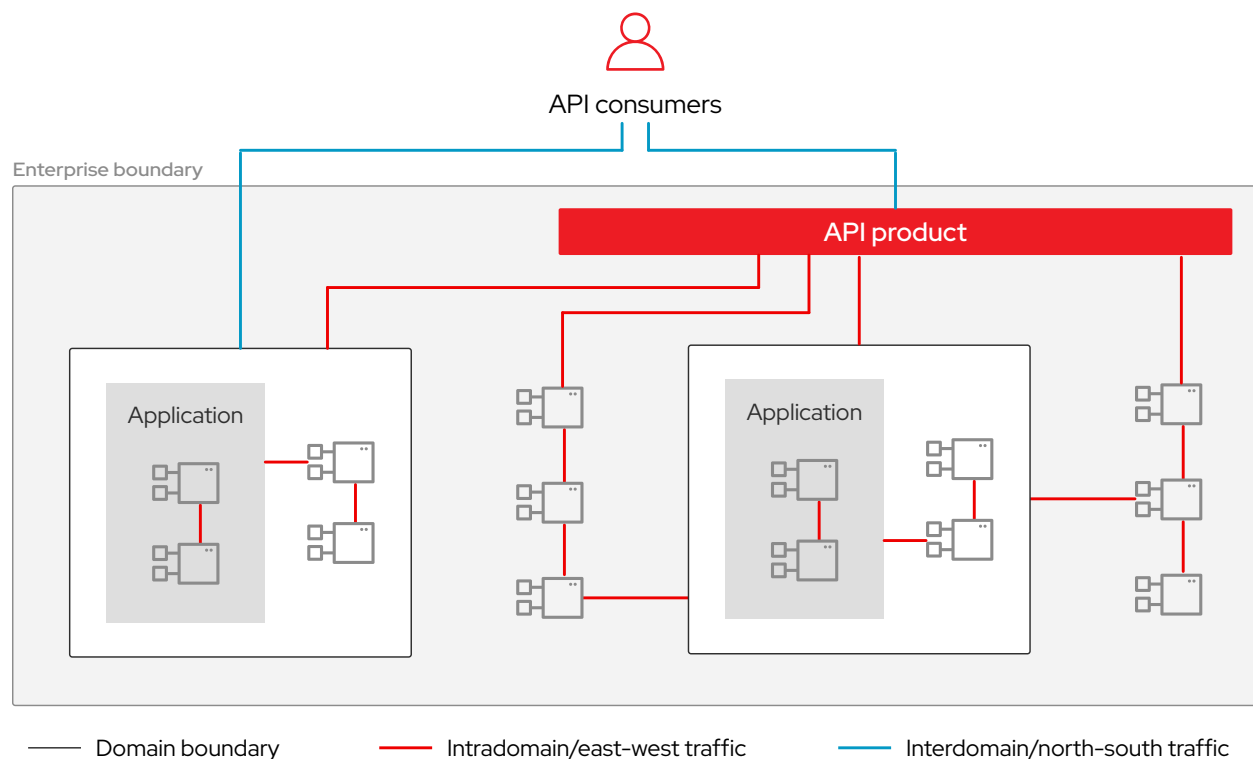
Use API management for north-south traffic patterns that cross domain or enterprise boundaries and service meshes for east-west traffic patterns that stay within those boundaries.

This advice can help you choose the most efficient path forward in most circumstances. Even so, there are exceptions to this guidance. For example, your organization may already use platforms, technologies, or runtimes that are incompatible with service mesh solutions, making it significantly easier to deploy API management for some east-west traffic patterns.

Be sure to consider your existing infrastructure, business goals, and available time frame when planning your API management and service mesh solution strategy.

Get help from Red Hat

Red Hat can help you assess your infrastructure and create a plan for moving forward. Our experts work with your organization to achieve your goals using an integrated approach that brings together people, process, and technology.



Maximize your value through integration

The most effective service management architectures combine API management and service meshes to deliver the right capabilities for the right traffic at the right time. When building your service management architecture, deploy API management and service meshes according to these recommendations.

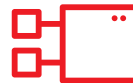


API management

Use API management to administer relationships between exposed services and consumers. API management is best for:

- ▶ North-south traffic patterns.
- ▶ Interdomain traffic.
- ▶ API-led development situations in which not all platforms can access a service mesh.

In general, API management is not appropriate for connecting services or microservices that are part of the same business domain.

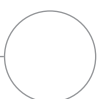


Service meshes

Deploy service meshes when you need advanced traffic control, security, resilience, and observability for microservices and cloud-native applications. Service meshes are best for:

- ▶ East-west traffic patterns.
- ▶ Intradomain traffic.

Avoid using service meshes in situations in which service consumers require formal documentation, billing, application use plans, and third-party authorization.



Find the right solutions for your organization

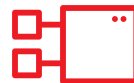
There are many API management and service mesh solutions available. Choosing the right solutions for your organization can significantly impact the efficiency and value of your service management architecture.



API management solutions

Look for an API management solution that goes beyond a simple gateway and access controls to provide the functionality needed to implement a complete API strategy:

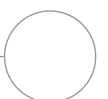
- ▶ **Monetization.** **Monetization capabilities** help you turn your services into revenue. Create metric-based use contracts for accessing your APIs and offer multiple access and service level tiers for your customers.
- ▶ **Developer portal options.** Developer portals typically provide API documentation and onboarding processes. Help developers find and use your APIs more easily.
- ▶ **API life-cycle management.** Like all IT assets, APIs undergo multiple life-cycle stages. Simplify management of your APIs from initial design, through implementation, to retirement.
- ▶ **Analytics.** Analytics help you understand what's going on with your APIs. Learn which consumers or apps are calling which APIs and how often, and gain insight into how many APIs have failed and why.
- ▶ **Policies.** Policies help you ensure your APIs are used and secured appropriately. Define and enforce policies for traffic management, security, and use to keep your API program operating in compliance.



Service mesh solutions

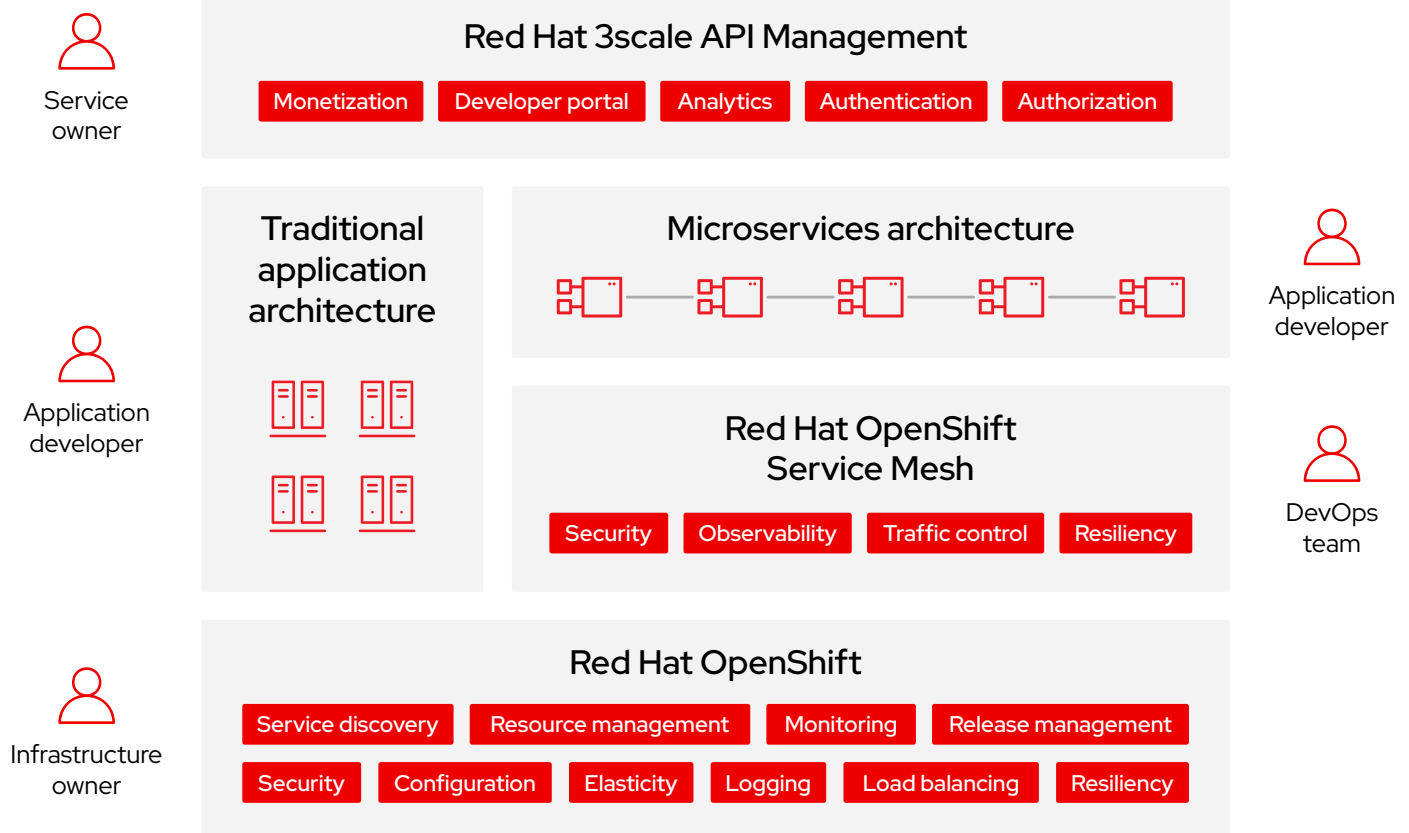
Look for a service mesh solution that provides everything you need to build and operate a microservices architecture:

- ▶ **DevOps tools.** Microservices are a key element in DevOps approaches. Adopt DevOps more easily with solutions that include development kits, continuous integration/continuous deployment (CI/CD) tools, and flexible automation capabilities.
- ▶ **Developer support.** Developers are most productive when they have the right tools for their job. Offer multiple development languages, runtimes, and databases to help your developers innovate faster.
- ▶ **Cross-infrastructure support.** Hybrid infrastructures are becoming the norm for innovative organizations. Deploy your microservices across physical, virtualized, and all types of cloud infrastructure to optimize microservice performance, security, and costs.
- ▶ **Microservice life-cycle management.** Microservice management is critical to ensure cloud-native applications run predictably and reliably. Independently scale microservices, control configurations, enforce routing and security rules, and understand how traffic flows between your services in near-real time.



Build a complete service architecture with Red Hat

Red Hat offers integrated API management, service mesh, and infrastructure platform products to help you build a comprehensive service management architecture. This architecture is divided into functional areas, providing targeted capabilities for multiple stakeholders. We build each product in the architecture with these stakeholders in mind to deliver an efficient, user-friendly experience.



Integrated products for service management architectures

Infrastructure foundation

Red Hat can help you address the inherent difficulties in managing and supporting varied applications, services, APIs, and microservices through the use of a container platform.

Red Hat® OpenShift® is a hybrid cloud, enterprise **Kubernetes** platform that helps IT development and operations teams work together to deliver and manage microservices-based applications. It supports containerized, legacy, and cloud-native applications and provides everything you need to build a microservices foundation:

- ▶ Built-in infrastructure security and workload isolation
- ▶ Microservices configuration, discovery, load balancing, and management
- ▶ Monitoring and logging
- ▶ Infrastructure resource management, resiliency, and elasticity

API management

Red Hat offers modular, lightweight, and comprehensive API management solutions that are open source, built on open standards, and available on-site or in the cloud.

Red Hat 3scale API Management is an API management platform that makes it easy to share, secure, distribute, control, and monetize your interfaces. It includes everything you need to implement an effective API program:

- ▶ API traffic control
- ▶ API program management
- ▶ Red Hat OpenShift integration
- ▶ Hybrid cloud support across all components
- ▶ Comprehensive security
- ▶ High availability and performance
- ▶ Customizable developer portal

Service mesh

Red Hat includes service mesh capabilities within Red Hat OpenShift, installed via a Red Hat OpenShift operator for simpler deployment.

Based on a set of open source projects, **Red Hat OpenShift Service Mesh** provides a uniform way to connect, manage, and observe microservices-based applications. It includes:

- ▶ **Istio**, an open source project for integrating and managing traffic between services.
- ▶ **Jaeger**, an open, distributed tracing system that track requests as they move between services.
- ▶ **Kiali**, an open source project for viewing configurations, monitoring traffic, and analyzing traces.
- ▶ Multiple networking interfaces.
- ▶ The Red Hat 3scale Istio plugin for integration with Red Hat 3scale API Management.

Try Red Hat OpenShift Service Mesh at learn.openshift.com/servicemesh.

Gain flexibility through integration

Our platforms give you the flexibility needed to build an effective, efficient services architecture that meets your organization's needs today and adapts easily to future change.



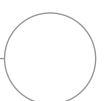
Deploy a consistent foundation for all of your microservices.



Continue using your preferred tools, languages, and runtimes.



Start where you are today and expand at your own pace.



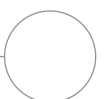
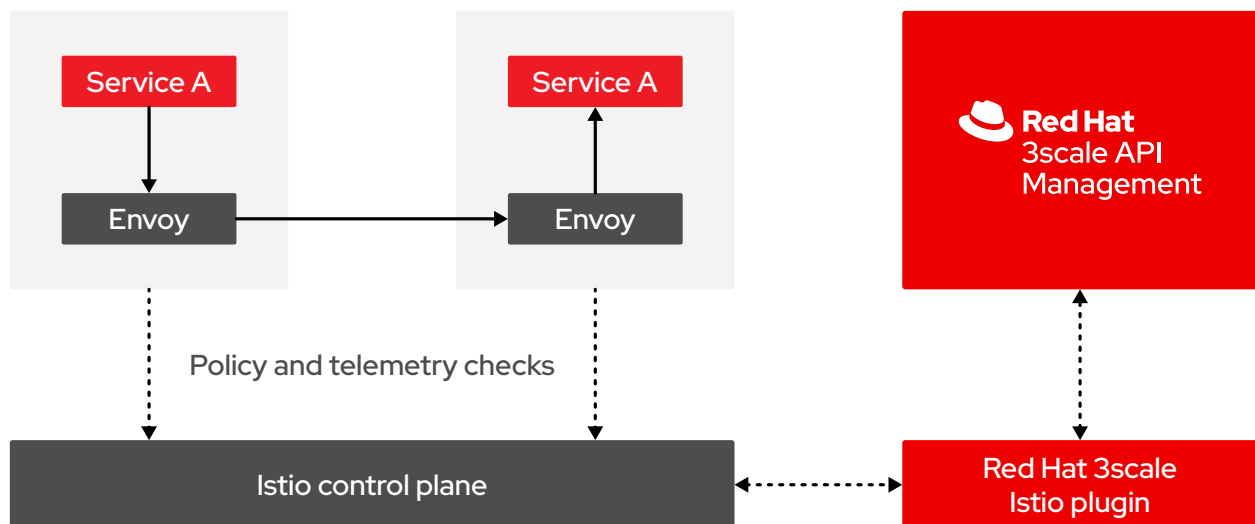
Deploy API management and service mesh solutions together

Red Hat API management and service mesh products integrate to allow you to use them together more easily and effectively.

How does it work?

Red Hat OpenShift Service Mesh implements a service control plane using Istio and a service data plane using **Envoy**, an open source edge and service proxy technology. The control plane is used to configure rules and policies that are applied at the microservice runtime. The data plane controls communication between microservices according to these rules and policies.

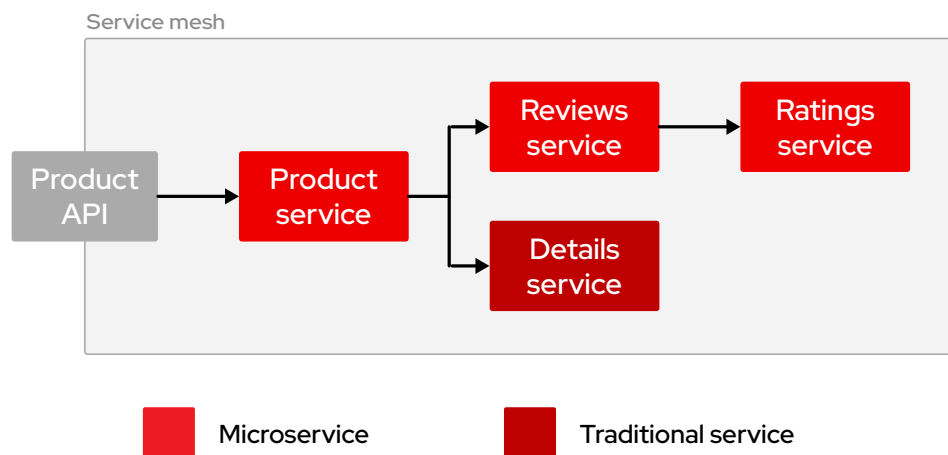
The Red Hat 3scale Istio plugin, included with Red Hat OpenShift Service Mesh, connects the control plane to Red Hat 3scale API Management. During operation, policy and rule checks run from the data plane, through the control plane and plugin, to Red Hat 3scale API Management to authorize API requests and report use. API policies are configured through Red Hat 3scale API Management and applied via the plugin and service mesh control plane. This integration allows you to efficiently set up an API with a service mesh backend.



Deliver a complete service using APIs and microservices

This section describes how to set up a complete service offering using Red Hat 3scale API Management and Red Hat OpenShift Service Mesh. In this example, we construct a simple products rating API. Our goal is to create controlled access to a set of microservices and traditional application services that provide information about a product. Users, developers, and applications will be able to access details, reviews, and ratings for a particular product.

To accomplish these objectives, we will set up an API that provides access to a core product microservice, which can connect to other services to provide product details, reviews, and ratings information.

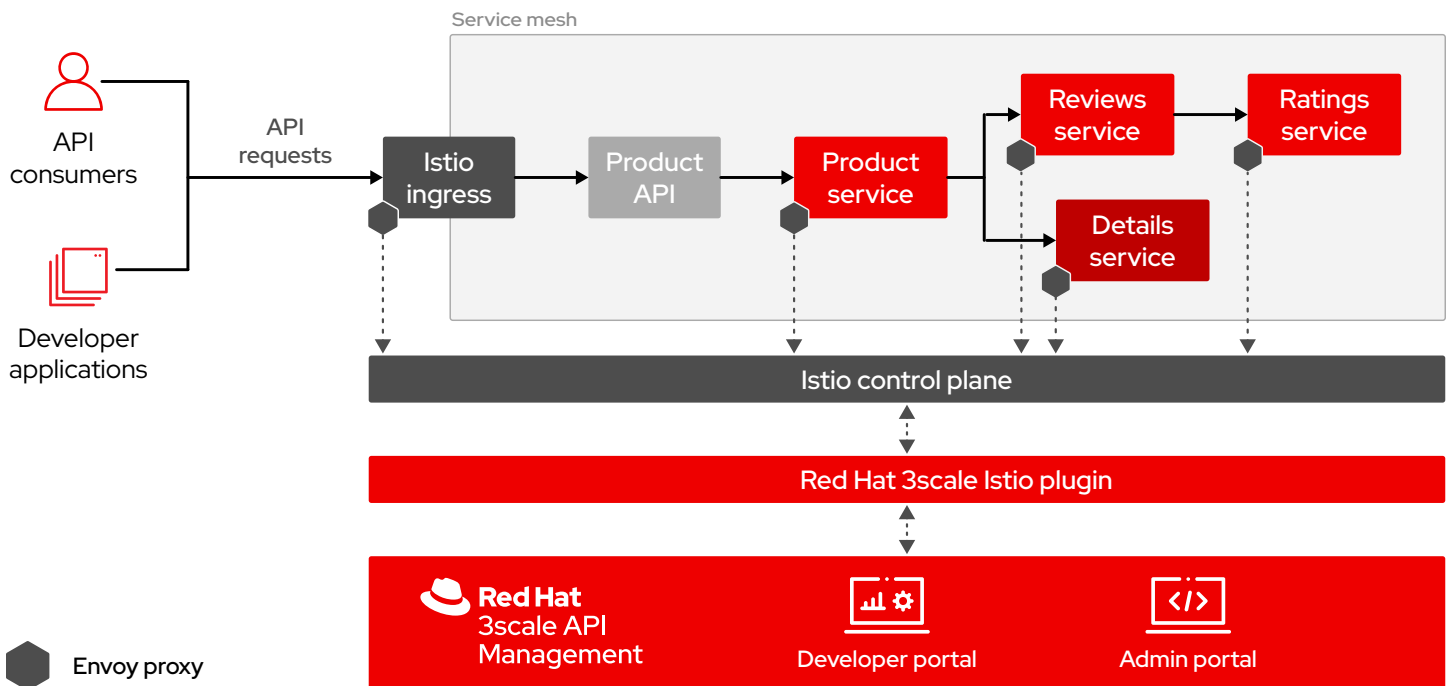


This page shows a more detailed configuration for this service example. The product, review, and rating services are implemented as microservices within Red Hat OpenShift Service Mesh. These services interact with each other within the same domain boundary. The details service uses existing, traditional application architecture.

All of these services sit behind the product API, which is independent of the service implementation. The API is set up via Red Hat 3scale API Management, allowing formal rules and policies to be applied to its use.

Requests enter the service mesh through standard Istio ingress processes. At each ingress point, the request is checked against the predefined policies via Envoy proxies and the Red Hat 3scale Istio plugin. If the request complies with all policies, it is routed forward to the product service. The product service can then interact with the details, reviews, and ratings services. These interactions are also governed by predefined policies applied via Envoy proxies. Finally outbound traffic is routed back to the requester via Envoy proxies and the Red Hat 3scale Istio plugin.

This configuration provides managed access to product information for internal and external developers while preventing direct access to the services themselves and maintaining detailed control over the back-end service implementation.



Take advantage of flexible deployment options

Red Hat makes it easier to build an integrated API management and service mesh architecture. You can start with either an API management or a service mesh solution and expand to incorporate the other at your own pace and with less effort.

Red Hat 3scale API Management

Starting point: Red Hat 3scale API Management

If you already have Red Hat 3scale API Management, you can add Red Hat OpenShift Service Mesh to your environment using the Red Hat 3scale Istio plugin.

1. Enable the Red Hat 3scale Istio plugin.
2. Activate Red Hat 3scale API Management authorization for the services you wish to move to your service mesh.
3. Once the desired services are added to your service mesh, deactivate authorization for them through Red Hat 3scale API Management APIcast gateways.

Your existing API-based access controls operate without change. You can continue to use your current rules and policies via Red Hat 3scale API Management while moving the back-end service implementation to a service mesh. This configuration ensures traffic control gateways are not duplicated.

Red Hat OpenShift

Starting point: Red Hat OpenShift Service Mesh

If you already have Red Hat OpenShift Service Mesh, you can add Red Hat 3scale API Management to your environment using the Red Hat 3scale Istio plugin.

1. Enable the Red Hat 3scale Istio plugin.
2. Activate Red Hat 3scale API Management authorization for the services you wish to administer as APIs.
3. Configure API management policies and access controls through Red Hat 3scale API Management for the nodes and services you wish to expose as APIs.

Your existing service mesh operates without change. You can continue to develop, connect, and manage microservices using Red Hat OpenShift Service Mesh while implementing certain microservices as APIs without no duplication in traffic control gateways.

Read the [Red Hat OpenShift Container Platform documentation](#) to learn more about using the Red Hat 3scale Istio plugin.



Discover success across industries



Amsterdam Airport Schiphol, Europe's fourth-busiest airport, wanted to improve its passenger experience and become the best digital airport. This goal includes providing seamless journeys for passengers – for example, by minimizing the time spent on the travel booking process – improving the cost-efficiency of its operations, and using the latest technology to collaborate with airlines and other stakeholders

To support this shift, Schiphol decided to migrate several of its IT systems to the cloud to become more flexible, secure, and efficient. The airport deployed **Red Hat OpenShift** as the foundation for its hybrid cloud environment, supported by **Red Hat 3scale API Management**, Red Hat Middleware, and other Red Hat products to accelerate development and deployment and improve API management. With help from **Red Hat Consulting**, Schiphol deployed **Red Hat OpenShift Dedicated** in just 10 days.

With the new platform, Schiphol can more quickly add new services. Connecting its on-premise environment to Red Hat OpenShift lets the airport easily exchange data between its main systems and cloud APIs. In addition, by using its Airport Service Bus platform to infuse data into its API services in Red Hat OpenShift, the airport's IT teams can develop new APIs 50% faster. Overall, the new Red Hat environment supports Schiphol's vision of agile, self-service processes and vendor flexibility that help its IT teams quickly and efficiently develop and deploy new customer-facing services.



Launched a production cloud platform in 10 days



Cut new service and API development time by 50%

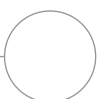


Deployed self-service capabilities for increased agility



"With [Red Hat] 3scale API Management, the learning curve is small, and you can deploy APIs very quickly."

Mechiel Aalbers
Senior Technical Application Coordinator,
Amsterdam Airport Schiphol





Emirates NBD provides retail, wholesale, and corporate financial services across the UAE and international markets. To increase the speed and flexibility of IT delivery while controlling costs, it began an ambitious multimillion-dollar initiative to transform its end-to-end technology infrastructure and practices. With a centralized private cloud platform built with Red Hat container, integration, and API technologies, Emirates NBD simplified collaboration with third-party partners, cut software development cycles by months, and launched innovative digital engagement capabilities to compete with cloud-native companies. Now, it can take advantage of emerging technology to offer the services modern consumers in key markets demand.



Simplified collaboration between teams and with partners using APIs



Reduced app launch and update cycles from 6-18 months to hours



Established a flexible, stable foundation for private and hybrid cloud



Lufthansa Technik

Lufthansa Technik, the world's largest independent provider of airline maintenance, repair, and overhaul (MRO) services, sought to create a digital platform for the aviation industry. The solution, AVIATAR, would help material planners, engineers, and other end users better predict events and, as a result, save time and money. To support its development and operation, the company created a hybrid cloud infrastructure based on enterprise open source software from Red Hat. The AVIATAR team can now use agile DevOps approaches, automation, internal and third-party integration, and self-service capabilities to quickly iterate based on data and feedback. As a result, Lufthansa Technik provides an innovative digital platform that helps the world's airlines optimize their operations.



Sped application workflows with self-service, automation, and DevOps



Gained flexibility through improved infrastructure and solution integration

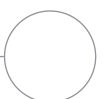


Expects to save airlines millions of euros on MRO services



"To completely change our infrastructure and evolve toward microservices and APIs, we had to create a new organizational structure. We now have agile teams with product owners and business owners working together."

Miguel Rio Tinto
Group CIO, Emirates NBD





Intendencia de Montevideo

To support a population of 1.5 million people, the municipal government of the **City of Montevideo**, Uruguay, employs 8,000 public administration officers. The city sought to develop an IT platform that would integrate the municipality's infrastructure, services, and data to improve citizen satisfaction. It used Red Hat OpenShift to create a microservices- and Kubernetes container-based platform to integrate internal and third-party services and data. It also built a city service API and state web portal, MontevideoAPI, using Red Hat 3scale API Management to deliver valuable information – based on real-time transportation data – to citizens and partner organizations. The city can now deliver improved traffic services to citizens.



weathernews

Weathernews Inc., one of the world's largest private weather information service companies, wanted to speed application and service development to meet an increase in demand for weather information across industries and markets. The company sought to integrate various internal and external systems using standardized, centrally managed APIs. Weathernews also needed to ensure its API environment was secure and reliable. By building its API management platform with Red Hat 3scale API Management, Weathernews has not only expanded internal use of APIs for more efficient development and faster service delivery to its customers but also gained new, robust provisioning security capabilities.



Developed a bus tracking app used by more than 150,000 citizens per month



Created a standard API platform to support essential business functions



Integrated real-time data on private and public transportation



Improved API provisioning speed, doubling development productivity



Improved state decision-making and traffic services for citizens

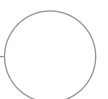


Gained robust access controls to protect the platform and company data



“It was a great benefit of Red Hat 3scale API Management that we could readily start on a small scale. The license provisioning, which had been the biggest challenge, is now done quickly and safely with access control.”

Masaki Fukuda
Service Menu Development Group Leader, Weathernews



Ready to get started?

A comprehensive service management architecture can help you increase speed, agility, and efficiency. Red Hat offers integrated API management and service mesh solutions that help you get the most from your APIs and microservices.

Learn more about deploying a service management architecture with Red Hat:
red.ht/service-mesh-api-mgt

Get started faster with Red Hat Services.

Work with Red Hat experts to deploy an effective service management architecture. Red Hat offers consulting and training services to help your organization adopt API management and service mesh solutions faster.

Schedule a complimentary discovery session:
redhat.com/consulting