tetrate

Simplify Kubernetes and Multi-Cloud Complexity with the Service Mesh

Table of Contents

Executive Summary

As today's enterprises shift to the cloud, Kubernetes has emerged as the de facto platform for running containerized microservices. And while Kubernetes operates as a single cluster in many deployments, enterprises and federal agencies inevitably run their applications on a complex, often confusing, architecture of multiple clusters deployed to a hybrid of multiple cloud providers and private data centers. This approach creates a new set of challenges. How do your services find each other? How do they communicate securely? How do you enforce access and communication policies? How do you troubleshoot and monitor health? Even on a single cluster, these are not trivial concerns. In a multi or hybrid-cloud environment, the complexity can be overwhelming.

For federal agencies and enterprise organizations seeking the flexibility to deploy applications on cloud providers that align with their cost, compliance or strategic consideration, the service mesh is an essential technology that addresses the above challenges. It empowers the consistent and streamlined deployment of applications across diverse Kubernetes clusters spanning different clouds. At a high level, the service mesh simplifies Kubernetes complexity by:

1. Decoupling traffic management from Kubernetes by running proxies
2. Centalizing and standardizing the management of networking concerns
3. Improving overall security posture using mTLS to encrypt traffic for secure communication and enabling zero trust security operations across any environment
4. Ensuring your system remains performant and efficient as it scales

As an additional benefit, the service mesh collects a trove of valuable data from logs, traces and metrics related to your network traffic. This data can help you create a more robust and reliable system.

When used together, Kubernetes and the service mesh provide a powerful platform for building and operating complex, distributed applications efficiently and securely across multi-cloud environments with less specialized expertise in each cloud and less manual toil. Developers, platform engineers, network and security professionals are able to perform their jobs better individually while collectively innovating faster. These benefits, especially developer productivity, are inordinately impactful given today's reliance on digital technologies. Increasing developer productivity, removing complexity and reducing toil provide a faster path to production and reduce time to market, which reduces time to value.

The Network Requirements of Enterprise Systems

Kubernetes provides a robust and clean networking model with many of the fundamental building blocks of networking supported. However, as a federal agency or an enterprise organization, you probably need much more. For example:

- Sophisticated routing
- Strong security
- Observability
- End-user and inter-service authentication and authorization
- Load balancing
- Canary testing
- Health checks
- Timeouts and retries
- Fault injection
- Bulkhead
- Rate limiting

Before the modern, cloud-native era, the landscape for enterprise organizations was proprietary. Organizations ran their systems in private data centers. Infrastructure was mostly static, with separate IT teams responsible for capacity planning. Software was typically a large monolith with long release cycles.

To handle the enterprise networking requirements mentioned above, the common practice to ensure adherence to policies and interconnectivity between subsystems was to have standard client libraries used by all software teams. This, of course, led to a lack of flexibility, over-budget and past-deadline project failures and slow decay, as there was no way for complex software systems to stay up to date with modern innovation.

Networking in the Modern, Multi-Cloud Era

In the modern era, software systems are deployed in the cloud, on multiple clouds, private data centers and even edge locations. The infrastructure is dynamic. The software comprises hundreds and thousands of microservices that may be implemented in multiple programming languages. The infrastructure and application development follow DevOps practices for continuous delivery. Security is integrated into the process following DevSecOps practices. Different components of the system are released constantly.

This was a boon for productivity and flexibility - but brought on new problems of management, control and policy enforcement. All these microservices implemented in multiple languages somehow need to interact. Developers and administrators need to understand the flow of information, be able to detect and mitigate problems and secure the data and the infrastructure.
Enter the service mesh, built on open source Istio.

A service mesh is a dedicated infrastructure layer that decouples some of the critical operational tasks of a distributed application from its business logic. Large-scale Kubernetes-hosted microservices applications are natural candidates for service meshes due to their complex requirements of inter-service communication (e.g. retries, timeouts, traffic splitting), observability, (e.g. metrics, logs, traces), and security features (e.g. authentication, authorization, encryption, zero trust). Service meshes can offload many operational concerns of the Kubernetes cluster, leaving developers to focus on business logic.

Service Mesh Benefits

A service mesh has many benefits in a modern, distributed and dynamic networking environment, such as Kubernetes-based systems, where new workloads are deployed constantly, pods come and go and instances scale up or down.

- **Offload Networking Concerns:** The service mesh externalizes all the networking concerns from the applications. Now they can be managed and updated centrally. By offloading all networking concerns to the service mesh, service developers can focus their efforts solely on their application and business logic.
- **Transparent Library Upgrades:** With a service mesh, you can upgrade your service mesh and everyone immediately enjoys the latest and greatest transparently. Traditionally, to introduce a change or upgrade to a client library, you would need to negotiate with each team individually, supporting multiple versions of libraries and across multiple programming languages.
- **Simpler Management of Cross-Cutting Concerns:** As a central component that touches all of your services, the service mesh can handle cross-cutting concerns - such as observability, health checks and access policy enforcement - across all services in your Kubernetes-based system.
- **Zero Trust Security Out of the Box:** The service mesh can add a layer of security to an enterprise's inter-service communication by employing a zero trust approach to access and using mTLS to encrypt traffic for secure communication. Additionally, limiting access from application to application helps to ensure that a malicious attacker who exploits one service cannot move laterally through your network to exploit other services.
- **Benefit of Community:** You benefit from the efforts of experts that keep evolving, improving and optimizing the service mesh built on open source Istio. The service mesh is also used and battle-tested by many organizations. This means that problems that might impact you may have been discovered and reported by other users.

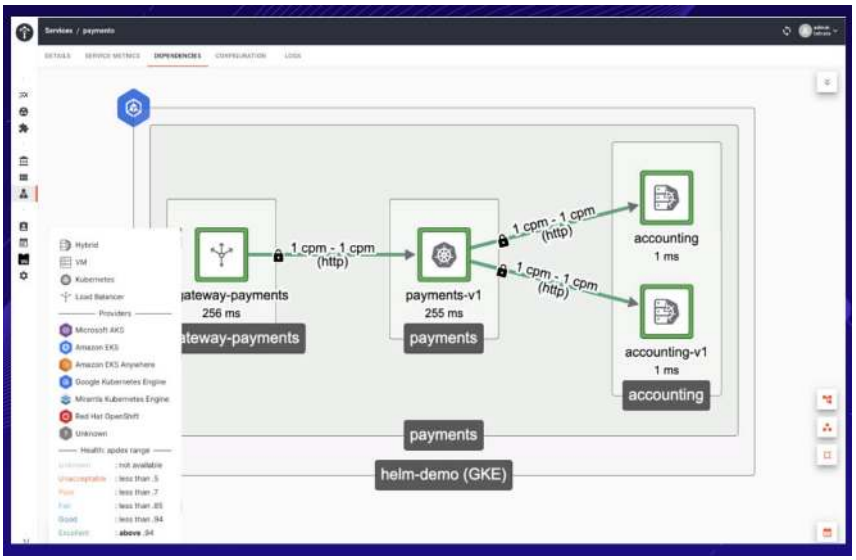
Unlocking the Synergy Between Service Mesh and Kubernetes

The synergy between Kubernetes and service meshes is powerful as the service mesh builds on top of the basic Kubernetes networking model. Kubernetes offers a flexible and extensible framework, making integration with service meshes seamless. While Kubernetes is a widely deployed platform for large-scale distributed systems, out of the box it doesn't address all the needs of complex enterprise systems deployed across multiple clouds and private data centers. This is where an Istio-based service mesh fills that gap, providing vital connectivity, management and observability across Kubernetes clusters while seamlessly connecting multiple clusters across diverse cloud infrastructures.

For large systems - in particular, systems composed of multiple Kubernetes clusters - the service mesh becomes a standard add-on. Once enterprises and federal agencies begin working with multiple clusters, which might spread across different clouds, the service mesh becomes essential. Service meshes are designed to work seamlessly with various Kubernetes distributions and container orchestration platforms including AKS, EKS, GKE and OKE ensuring that you can adopt and manage multi-cloud environments without being locked into a specific vendor. Furthermore, you can define and enforce consistent network policies, security policies and access control policies using a service mesh. This simplifies governance and compliance efforts in a multi-cloud setup, helps eliminate inefficiencies and reduces friction between networking, security and development teams.

TSB: Manage Multi-Cloud Complexity via a Single Pane of Glass

Tetrate Service Bridge (TSB) is Tetrate's implementation of the powerful Istio open-source project, allowing you to manage, observe and secure your services without having to change your application code. TSB unburdens your operations and development teams by simplifying service delivery across the board, from traffic management and mesh telemetry to securing communications between services. Additionally, TSB provides a single pane of glass for observability, visualization, health checks, policy management and policy enforcement. Offloading these concerns - such as authentication, authorization, metrics collections, and health checks - to a central component is a game changer for DevOps, application developer productivity and multi-cloud adoption.



TSB's management guardrails offer safety and agility, transforming the platform and security teams from a bottleneck into an agility accelerator for the organization.

TSB allows organizations to deploy their services to cloud providers that make the most sense for them based on cost, compliance or other strategic factors. Instead of being locked into a single vendor, organizations can use TSB to connect all of their clusters and services together regardless of the cloud that each runs on. As a global management and control plane, TSB is able to provide the following capabilities to applications:

Uniform Communication: In a multi-cloud setup, applications may be deployed across various cloud providers, each with its own networking and communication mechanisms. TSB abstracts these differences, providing a uniform way for services to communicate regardless of their location. This simplifies service-to-service communication and eliminates cloud-specific configurations.

Service Discovery: TSB offers centralized service discovery, making it easier for applications to locate and communicate with services, whether they are hosted in a single cloud or distributed across multiple clouds. This simplifies cross-cloud service discovery, reducing the complexity of maintaining service registries.

Traffic Routing and Load Balancing: TSB can intelligently route traffic between services, regardless of their cloud location, based on policies and rules. This facilitates load balancing and ensures efficient resource utilization across multiple clouds.

Security: Multi-cloud environments introduce additional security challenges. TSB can enforce consistent security policies, such as FIPS validated mutual TLS (mTLS) encryption, authentication and access control, regardless of where services are deployed. This ensures that security measures are uniformly applied across clouds.

Observability: TSB provides centralized observability features like distributed tracing, metrics collection and logging, which enable teams to monitor and troubleshoot applications consistently across different clouds. This unified observability simplifies root cause analysis and performance optimization.

Resilience: Multi-cloud environments may experience disruptions or outages in specific regions or providers. TSB can implement automated resilience mechanisms like retries, circuit breaking and failover to mitigate the impact of such disruptions, ensuring uninterrupted service availability.

Traffic Control: When managing workloads across multiple clouds, it's essential to have fine-grained control over traffic routing and shifting. TSB offers traffic management capabilities, enabling controlled rollouts, canary deployments and blue-green deployments across clouds.

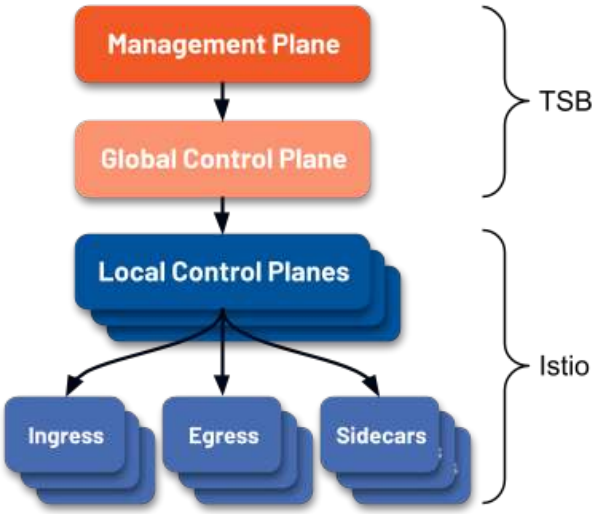
Policy Management: TSB allows for consistent policy enforcement across clouds, including network policies, security policies, zero trust and access control policies. This simplifies governance and compliance efforts.

Hybrid Cloud Support: TSB can extend beyond cloud environments to include on-premises or data center deployments, creating a seamless hybrid cloud setup. This is valuable for organizations with hybrid infrastructure.

Interoperability: TSB is cloud-agnostic and can work with various Kubernetes distributions, container orchestration platforms, and even legacy bare metal systems. This interoperability simplifies multi-cloud adoption without vendor lock-in concerns.

While a service mesh supports your enterprise with cross-cloud connectivity, gaining a clear picture of everything you're running across your clouds is still challenging. As your systems grow in scale and spread across more clusters and clouds, the ability to see it all through a single pane of glass is critical. Tetrate Service Bridge gives you the tools you need for monitoring, visualization, optimizing and troubleshooting your network - whether you run multi-cloud or hybrid-cloud and whether you use Kubernetes or non-Kubernetes deployments.

TSB enforces the above policies at runtime in a reliable, available way by implementing a tree-shaped architecture composed of three layers:



- **Management Plane:** TSB's management plane is the root of the tree, where your stakeholders - application developers, platform, security, networking and operations owners - configure and observe the mesh.
- **Control Plane:** TSB delivers both a global control plane - which is responsible for cross-cluster service discovery and configuration distribution - and a local control plane, which is responsible for programming an upstream, OSS Istio control plane.
- **Data Plane:** Deployed as sidecars and gateways, Envoy acts as the mesh's data plane, handling the bits-and-bytes of your application traffic. Envoy can be deployed in many modes: as an ingress or egress proxy, traditional reverse load balancer, or as a sidecar.

Tetrate Service Bridge also enables management of the mesh's capabilities: it lets you manage who in your organization can change which mesh configurations affecting what physical infrastructure. In other words, it allows safe multi-tenant usage of the service mesh: your central teams can mandate policy while delegating configuration to application teams - who can't configure the system outside of the bounds they've been given. These management guardrails offer safety and agility, transforming the platform and security teams from a bottleneck into an agility accelerator for the organization. On top of the runtime policies that the mesh can enforce for your applications, TSB provides a variety of management capabilities to make the mesh useful in a complex enterprise:

- **Hierarchical Identity and Access Management:** bring your existing identity provider, assign your physical infrastructure into a hierarchy, then collaboratively enforce controls across the entire infrastructure with policy defaults, bounds and delegation.
- **Default Configurations and Controls:** set configuration defaults for all types of configuration on your hierarchy, with the ability for teams lower in the hierarchy to replace with only stricter policy. Enforce permissive policies in test environments but strict policies in production.
- **Visibility and Operability:** view your application topology and understand how traffic flows - at the application level, across your entire infrastructure. Assess application health with mesh operational metrics and bootstrap availability measures in your organization with objective measures like ApDex.
- **Audit and Traceability:** see what changed in the system when and visualize the system's runtime behavior before and after. Every event in the system is recorded and this audit log can be exported to other systems. TSB's visibility closes the loop on policy change to runtime effect.

TSB: Enable Zero Trust Operations at Runtime

Achieving a zero trust architecture (ZTA) is a key goal for many enterprises and government entities. These organizations have existing infrastructure, network and security investments and most of these organizations have multi-cloud and hybrid cloud infrastructure where each application stack runs on different compute resources and is also geographically dispersed for resiliency. Strengthening security postures in the face of evolving cloud-native environments requires the adoption of new tools, higher levels of automation and adaptive policy frameworks.

The objective of recently published NIST SP 800-207A, co-authored by Tetrate's founding engineer Zack Butcher and NIST's Sr. Computer Scientist Ramaswamy Chandramouli, is to provide guidance for realizing an architecture that can enforce granular application-level policies while meeting the runtime requirements of ZTA for multi-cloud and hybrid environments.

Fundamentally, a zero trust mindset means that "the attacker is already inside": traditional perimeter based security is ineffective because a motivated attacker can always get inside your network. Instead, we need to focus on bounding attacks in both space and time. To achieve this at runtime, you need to be performing at minimum five checks on every hop between components in your infrastructure:

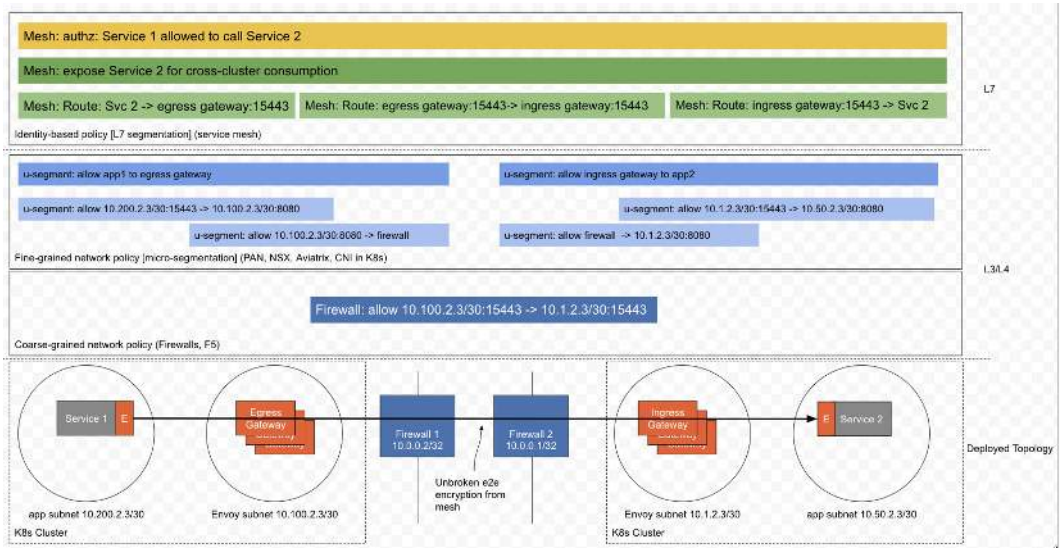
1. **Encryption in Transit:** mTLS in the mesh, (m)TLS to external services
2. **Service Identification & Authentication:** SPIFFE identifies for workloads in the mesh
3. **Service to Service Authorization:** Built-in policies good starting point, mature implementations should leverage dedicated authz infra for richer policy decisions - e.g. Next Gen Access Control (NGAC)
4. **End-user Authentication:** Defer to trusted identity provider or IDaaS
5. **End-user to Resource Authorization:** Integrate with existing systems via OIDC or leverage dedicated authz infra - e.g. NGAC

In addition to these runtime checks, we need the ability to monitor the system continuously to ensure policy is being enforced and to respond to changes on demand by updating policy - all of which can be handled by the service mesh.

TSB: Universal Policy Enforcement

As a dedicated infrastructure layer, TSB is an invaluable security tool for modern applications. The service mesh's sidecar intercepts all traffic in and out of your applications, where it acts as a universal policy enforcement point. This allows the service mesh - which centrally manages a fleet of your applications' sidecars - to become the modern cloud native security kernel (NIST SP 800-204B).

The sidecar is able to enforce security and traffic policies, as well as generate telemetry to allow operators to close the loop on policy changes: they can author a change, observe its effect on the runtime and make additional changes as needed - all in a real time feedback control loop. In other words, the mesh provides the capabilities to implement the runtime controls needed to achieve a zero trust posture.



Kubernetes and the Service Mesh - Better Together

The adoption of Kubernetes in federal agencies and enterprise organizations is revolutionizing how organizations manage their IT infrastructures. Automating deployment, scaling and management of containerized applications allows organizations to embrace a cloud-native paradigm at scale and more easily employ best practices such as microservices and DevSecOps. However, as powerful as Kubernetes is, not utilizing a service mesh can lead to several drawbacks in networking and management of your containerized applications.

Kubernetes and the service mesh complement each other's strengths and address different aspects of deploying and managing containerized, microservices-based applications. Together, they provide a powerful platform for building and operating complex, distributed applications efficiently, securely and in a way that accelerates the productivity of developers, platform engineers, network and security professionals.

Here is a summary of how a service mesh and Kubernetes are better together:

	Kubernetes	Service Mesh
Consistency Across Multi-Cloud Environments	Kubernetes abstracts cloud-specific details, allowing you to deploy applications across multiple cloud providers consistently.	Service meshes extend this consistency to service communication and security policies, making it simpler and easier to deploy and manage microservices across diverse cloud environments.
Orchestration and Management	Kubernetes excels at container orchestration, automating the deployment, scaling and management of containers and services. It provides the infrastructure and abstractions for running containers efficiently.	Service meshes enhance Kubernetes by providing a dedicated layer for managing service-to-service communication, including advanced traffic routing, load balancing and security features. This simplifies the operational aspects of managing communication within Kubernetes clusters.
Traffic Control	Kubernetes offers basic load balancing and traffic routing capabilities, but they may not be sufficient for complex microservices architectures.	Service meshes provide advanced traffic control, allowing fine-grained routing, load balancing, circuit breaking and traffic shifting. This helps optimize application performance and resilience.
Security	Kubernetes offers security features like RBAC (Role-Based Access Control) and network policies to secure the infrastructure and access to the Kubernetes API.	Service meshes enhance security by providing mutual TLS (mTLS) authentication between services, access control and encryption for service-to-service communication. They also enforce security policies consistently across microservices, improving the overall security posture.
Observability	Kubernetes offers basic monitoring through the use of tools like Prometheus and Grafana.	Service meshes offer comprehensive observability with distributed tracing, metrics collection and logging. This enables better visibility into service behavior and simplifies troubleshooting and performance optimization.
Traffic Management	Kubernetes has some traffic management features, but service discovery and routing can be challenging in complex microservices environments.	Service meshes provide advanced service discovery and routing, simplifying inter-service communication, making it easier to manage application traffic.
Resilience	Kubernetes supports auto-scaling, which is beneficial for maintaining service availability.	Service meshes enhance resilience through features like retries, circuit breaking and failover. This helps mitigate the impact of service disruptions.

About Tetrate

Rooted in open source, Tetrate was founded to solve the application networking and security challenges created by modern computing so enterprises can innovate with speed and safety in hybrid and multi-cloud environments. As applications evolve into collections of decentralized microservices, monitoring and managing the network communications and security among those myriad services becomes challenging. This is why some of the largest financial institutions, governments and other enterprises rely on Tetrate to deliver modern application networking and security on a foundation of Zero Trust. http://www.tetrate.io.

Tetrate Service Bridge (TSB)

Tetrate Service Bridge (TSB) builds upon the advantages of 100% upstream Istio by adding powerful tooling and support to manage complex multi-cluster applications and services across any hybrid or multi-cloud environment. As the first FIPS-certified Istio Distro, Tetrate offers FIPS-compliant Istio builds that have been vetted and tested for security. Tetrate also provides lifecycle management and enterprise support for Istio and Tetrate Service Bridge, ensuring that government agencies have access to a secure and reliable platform that meets their unique security and compliance needs. https://tetrate.io/tetrate-service-bridge

Tetrate Academy

Accelerate your service mesh journey with expertly curated, hands-on training courses from the co-creators of open source Istio and Envoy. Private training for enterprise customers available upon request. https://academy.tetrate.io