Solo.io presents

A Buyer's Guide to Service Mesh

A comparative analysis of cloud-native enterprise service mesh

Contents

IMPORTANT NOTICE:

This competitive analysis has been carried out on a best efforts basis. Please read our legal disclaimer.
If you wish to provide feedback on the document, please complete the contact form at https://www.solo.io/company/contact/

1. Executive Summary

The modern application development environment is undergoing an evolution to meet the needs of organizations today. Advancements in artificial intelligence (AI) and growing demands of customer experiences are just some of the drivers behind the exponential growth and consumption of data and infrastructure demands.

Development teams must adapt their applications workloads as they scale to ensure they address these demands in order to help their organizations modernize. These influences across global organizations are also creating massive shifts across the infrastructure landscape.

To support the new demands of technology teams, the landscape is creating new solutions, tools, frameworks, and platforms to help technology teams deliver their expectations faster and whilst ensuring their applications' performance. As a result, we've seen organizations move from on-premise to cloud-based data centers, shift t o microservices-based architecture from monolithic stacks, and rapidly adopt cloud-native technologies.

With the predicted growth of container adoption tipped to reach over 90% of organizations by 2027, the need for technologies like service mesh can help more organizations adapt to these rapidly changing requirements and provide technical teams with an efficient and reliable way to help them manage the growing web of communications across their network, infrastructure, and application layers.

In this Buyer's Guide we evaluate service mesh solutions using eight key criteria including:

- **Data Plane Options**
  Covering capabilities including compatibility with Envoy, virtualized workload support, extensibility, and deployment methods.
- **Control Plane Functions**
  Reviewing each solution's hosting, tenancy, scalability, availability, and resiliency capabilities.
- **Security and Governance**
  Comparing each solution's governance around security including authentications, integrations, and policy management features.
- **Network Integrations**
  Analyzing the support compatibility across gateways, multi-cluster environments, and federated set ups.

- **Performance and Reliability**
  Investigating each solution's benchmarking, resourcing, and latency requirements.

- **Observability and Operations**
  Evaluating how each solution delivers metrics and insights and improves the development lifecycle.

- **Enterprise Support**
  Comparing the different support offerings including health checks, documentation, and professional services.

- **Open Source Community**
  Review each vendor's contribution and engagement with open source communities and projects.

The service mesh market is still relatively new, experiencing continual changes and fluctuations as organizations modernize, while simultaneously adapting to the evolving cloud-native landscape. Some users may choose to build their own service mesh solution, which can be a complex and resource heavy initiative.

2. About the Vendors

2.1. Solo.io

Solo.io's portfolio encompasses Gloo Gateway and Gloo Mesh, all leveraging popular open source technologies such as Kubernetes, Istio, Envoy, Cilium, eBPF, GraphQL, and WebAssembly.

The product portfolio of Solo.io is built upon these technologies due to their vibrant, engaged communities and widespread adoption by industry leaders. Notably, Envoy, a cornerstone of Solo.io's offerings, has been endorsed by major cloud service providers for its reliability and versatility as a proxy solution. Solo.io empowers both customers and community users to scale their modern infrastructure networking effectively, facilitating the easy operation of cloud-native applications.

Solo.io has strong ties across the open source community, holds influential positions in the Cloud Native Computing Foundation (CNCF), and is the leading contributor to CNCF open source projects worldwide including Istio, Envoy, and Cilium. Solo.io simultaneously supports their growing enterprise customers with their application networking demands alongside their commitment to the open source community.

2.2. Buoyant

Buoyant was founded in 2015 by two infrastructure engineers from Twitter. In 2017 they released Linkerd, a service mesh that became a CNCF incubating project in 2018 and graduated in 2021. Linkerd is built on the programming languages Rust and Go and positions itself as a solution for lightweight mesh for workloads.

2.3. Isovalent

Founded in 2017 by early eBPF contributors, Isovalent are the creators of open source project Cilium. As the main maintainer and developer of Cilium with over 85% of the code contributions, Isovalent has established a community of Cilium users looking to advance their application networking and network security. In December 2023, Cisco announced its intention to acquire the privately-held Isovalent, which was finalized in April 2024.

2.4. Hashicorp

Hashicorp was founded in 2012 and stemmed originally from the open source software Vagrant. It positions itself as a broad IT infrastructure company providing a suite of software tools and services designed to help organizations build infrastructure to improve the software development cycle and its supporting architecture. In December 2021, Hashicorp became a publicly traded company IPO'ing on the NASDAQ stock exchange. In April 2024, Hashicorp was acquired by IBM.

2.5. Google Cloud

Google Cloud Platform is the cloud computing business unit of Google, launched in 2008. It provides a wide range of computing services including cloud storage, analytics, machine learning, and infrastructure components. Known as one of the three major hyperscalers across the globe, Google Cloud holds an influential position in the cloud native and computing industry as it is responsible for originating the popular open source container orchestration system Kubernetes.

2.6. Red Hat

Red Hat is a well-established open source software provider founded in 1993. It established itself as a leader in enterprise Linux and eventually extended its influences across containerized workloads with its OpenShift family of solutions. In 2019, IBM completed its acquisition of Red Hat for $34 billion. Red Hat remains an independent subsidiary of IBM.

In this guide we provide readers with a comprehensive overview of the six service mesh solutions including: Gloo Mesh by Solo.io, Linkerd by Buoyant, Isovalent Enterprise for Cilium, Hashicorp Consul, Anthos Service Mesh by Google, and Red Hat OpenShift Service Mesh.

3. Market Alternatives

In this guide, we evaluate six of the leading service mesh solutions and score their functionality using the categories mentioned previously. All solutions will be benchmarked and compared based on their enterprise offering.

3.1. Gloo Mesh

Built by Solo.io, Gloo Mesh is a cloud-native service mesh that is designed to simplify the complexity and operational overhead of mesh technology across any environment, including multi-cluster and multi-cloud. Built on the popular open source project Istio, Gloo Mesh enables platform engineering teams to boost their productivity at scale with improved observability, performance, and security. Its open-core design is part of Solo's wider application networking platform Gloo, which also includes additional application networking tools including Gloo Gateway and Gloo Network for Cilium.

Gloo Mesh is built upon and tightly integrated with a collection of open source projects that Solo maintains and contributes to as part of its commitment to the Cloud Native Community Foundation (CNCF), including Istio. There are two iterations of Gloo Mesh, including Gloo Mesh Core, which provides optimal support for users working within smaller environments, and Gloo Mesh Enterprise, designed for large-scale workloads across multiple clusters and environments.

3.2. Buoyant Enterprise for Linkerd

Created by Buoyant, Linkerd is an open-source service mesh for Kubernetes, built on Rust, and is a CNCF graduated project since 2021. Linkerd markets itself as an alternative to Istio, another open source service mesh option.

Buoyant provides a production-ready enterprise edition of Linkerd called Buoyant Enterprise for Linkerd, which is positioned as a hardened distribution of Linkerd with additional tools and feature-sets. The enterprise edition also comes with support for users across various pricing tiers.

Starting from May 2024, Buoyant stopped providing stable builds of open source Linkerd, while charging companies to access stable builds of Linkerd through their enterprise product. By no longer providing upstream builds, Buoyant have positioned themselves as the only source of "stable" Linkerd, a break with the social contract of CNCF projects to date.
In this guide, we will refer to Buoyant Enterprise for Linkerd as 'Buoyant Enterprise.'

3.3. Isovalent Enterprise for Cilium

Isovalent was founded by the co-creator of the eBPF technology and is the dominant maintainer of CNCF project Cilium. Cilium incorporates multiple sub-projects, which are also maintained by Isovalent as part of its offering, including Hubble for networking observability, Tetragon for security and runtime tooling, and Cilium Service Mesh. Some functionalities of each sub-project are only available via Isovalent Enterprise for Cilium.

Isovalent Enterprise for Cilium is the enterprise support and packaging for project Cilium. Isovalent Enterprise for Cilium positions itself as a hardened, enterprise-grade, eBPF-powered networking solution that offers advanced networking and security features alongside enterprise-grade support for Cilium deployments.

In April 2024, the Cisco acquisition of Isovalent was finalized. The impact the acquisition has on the upstream Cilium project is undetermined.

In this guide we will refer to Isovalent Enterprise for Cilium as 'Isovalent Enterprise'.

3.4. Hashicorp Consul

Hashicorp Consul is one of the eight main commercial product lines from Hashicorp and was first released in April 2014. It positions itself as an identity- and security-based service mesh that helps connect applications across on-premise and multi-cloud environments. Hashicorp operates a "source available" model and provides access to the source code for all of its infrastructure portfolio including Consul, with additional enterprise offerings available through its paid subscription.

Hashicorp Consul can be consumed through a managed or self managed service as part of the Hashicorp Cloud Platform. Hashicorp recommends enterprises running workloads in complex environments to utilize the self-managed service version of Hashicorp Consul.
In April 2024, IBM announced its acquisition of HashiCorp. The impact of this acquisition on HashiCorp customers going forward remains uncertain.

In this guide, we will refer to the commercial offering of Hashicorp Consul as 'Hashicorp Consul.'

3.5. Anthos Service Mesh

Anthos Service Mesh is part of leading 'hyperscaler' Google Cloud Platform's (GCP) portfolio of cloud solution offerings. It is designed to be a fully managed service mesh for GCP users that is built on open source project Istio. Google launched Istio with Lyft and IBM in 2017 and remains a contributor and maintainer of the project alongside Solo.io.

Anthos Service Mesh is a suite of tools that helps users operate and monitor a service mesh on-premise or on GCP workloads. It also integrates with other GCP tools including Cloud Logging and Cloud Monitoring and Google Kubernetes Engine or Anthos clusters. Anthos Service Mesh capabilities center around authentication, insights, and traffic controlling.

Anthos Service Mesh is delivered as a fully managed solution on GCP or as a tested and supported Istio distribution as part of GKE Enterprise to be installed on-premise. In addition to this, GCP users get easy access and integration into the ecosystem of Google-products. In April 2024, Google pre-announced 'Google Cloud Service Mesh', a converged service mesh offering that merges Anthos Service Mesh with another offering called Traffic Director. It will provide the same service mesh features when launched.

In this Buyers Guide, we will refer to Anthos Service Mesh and the forthcoming Google Cloud Service Mesh in the comparison sections under 'Anthos Service Mesh.'

3.6. Red Hat OpenShift Service Mesh

Red Hat OpenShift Service Mesh (OSSM) was released in August 2019 as part of the Red Hat OpenShift Kubernetes commercial portfolio. It is based on open source project Istio and integrates other open source projects including Kiali for dashboarding and observability alongside Jaeger for distributed tracing.

It has been designed to integrate with the extended Red Hat portfolio of products. OSSM's main differentiation from Istio is that it takes a multitenant approach allowing multiple independent control planes within clusters.

In this Buyers Guide, Red Hat OpenShift Service Mesh will be referred to as 'Red Hat OSSM.'

4. Capabilities Overview

A Buyer's Guide to Service Mesh provides a thorough examination of the strengths and weaknesses of each vendor, delivering a detailed comparative analysis. To facilitate a quick overview in this executive summary, we have visually depicted the relative performance of each vendor below.

In this overview, and throughout this document, we will summarize the scoring using the following ranking in each category:

- The full ball (●): best-in-class platform
- The three-quarters ball (◕): runner-up
- The half ball (◗): acceptable capability
- The quarter ball (◔): weak capability
- The empty ball (○): no capability

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Data Plane Options	◕	◔	◔	◕	◗	◕
Control Plane Functions	●	◕	◔	◕	◗	◕
Security & Compliance	●	◕	◔	◕	◕	◕
Network Integrations	●	◗	◗	◗	◕	◕
Performance & Reliability	◕	◕	◗	◕	◗	◕
Observability & Operations	●	◕	◗	◕	◗	◕
Enterprise Support	◕	◗	◗	●	◕	●
Open Source Community	●	◔	◔	◔	◕	◕

4.1. Capabilities

When evaluating service mesh solutions, organizations often consider various criteria to ensure that the chosen solution aligns with their specific requirements and objectives. These categories have been selected to focus this competitive evaluation with a handful of criteria.

4.1.1. Data Plane Options

In modern serverless and microservices-based architecture, the data plane is responsible for handling the actual network traffic between services, intercepting all communications between services, and ensuring secure and reliable communications.

Data planes consist of a set of lightweight network proxies known as sidecars deployed alongside each service instance. These proxies intercept all incoming and outgoing traffic to and from the service, allowing for centralized control and management of communication within the mesh. Innovators in the service mesh space such as Solo now provide a sidecarless deployment method as an alternative. Modern service mesh offerings now often integrate with popular open source proxies like Envoy, leveraging its reliable and performant capabilities in complex architectures.

4.1.2. Control Plane Functions

A service mesh control plane is responsible for managing and configuring the data plane proxies deployed alongside each service instance. It provides a centralized platform for defining and enforcing policies, monitoring service-to-service communication, and dynamically controlling traffic behavior and deployments within the mesh environment.

When assessing which service mesh fits your environment, review components like what host options are available to help you manage services within the mesh, if multi-tenancy is required, the scalability of the solution if traffic or number of services increase, and its availability and resiliency.

Control planes play a critical role in orchestrating and managing the complex interactions between services within a service mesh, providing centralized control, visibility, and governance over service-to-service communication.

4.1.3. Security and Compliance

Ensuring the security and governance of workloads remains paramount for organizations, especially when managing high volumes of network traffic and communication across applications and environments. Solutions must offer robust capabilities for policy and risk management to address these challenges effectively.

A reliable and secure service mesh provider should prioritize key features such as mutual authentication, encryption, and data integrity across networks and services. Integration with frameworks like SPIFFE/SPIRE to issue and manage identities for services is also essential. Incorporating such functionality helps prevent unauthorized access, mitigates threats, and reduces the risk of potential data breaches. Moreover, it enables organizations to maintain accessibility and control over their workloads, networks, and environments.

Building a reliable and secure application begins at the network level. Leveraging a modern service mesh allows users to meet the stringent security requirements of their services, networks, and communication needs. By providing granular control, robust access management, and centralized service management, a service mesh empowers organizations to ensure the integrity and confidentiality of their data while facilitating seamless communication across distributed environments.

4.1.4. Network Integrations

Network integrations serve as the backbone of a service mesh, orchestrating communication across distributed environments. They empower service meshes to efficiently connect with both internal and external systems, ensuring reliable data exchange within the mesh architecture.

Service meshes equipped with robust network integrations excel in delivering essential functionalities including load-balancing traffic between services, optimizing resource utilization and minimizing latency. Modern service meshes often require support for multi-cluster integrations, allowing them to span across multiple Kubernetes clusters or environments. Network integrations facilitate seamless communication between services deployed in different clusters, promoting scalability, fault tolerance, and geographical distribution of services.

These integrations can also be federated in more complex topologies where independent service meshes or clusters can communicate and collaborate seamlessly. This fosters cross-domain communication and integration between disparate environments, enhancing flexibility and scalability for complex architectures.

4.1.5. Performance and Reliability

Performance of a service mesh can be measured against different performance characteristics including latency, throughput, and resource utilization. This data generally helps users determine which service mesh best suits their unique environments requirements.

Reviewing the latency of a mesh is important as service meshes introduce additional networking and processing overhead as they manage communications between services. This increase in processing requirement can introduce latency across an environment’s

system and network. A good modern mesh should have features built in to minimize latency, including connection pooling, load balancing, and timeouts.

Having a reference point to benchmark helps users make better resourcing decisions and ensure their network and service mesh performs at an optimal and reliable level. Measuring conditions like loads and traffic patterns of different configurations can help assess the scalability of a mesh or its reliability under certain circumstances, allowing users to identify roadblocks, improve performance, and optimize infrastructure spend as they scale.

4.1.6. Observability and Operations

Each application's network and infrastructure is unique, and as organizations scale teams must implement fundamental monitoring and troubleshooting processes to minimize any potential downtime.

Modern service meshes should have operational functions integrated, including basic observability capabilities like metrics, logs, and insights. This analytical data provides visibility into the performance of a mesh across a network and its distributed systems and environments enables users to better manage the lifecycle of their application.
This information is valuable for organizations and teams who are scaling and want visibility into how their services communicate, detect any abnormalities, or troubleshoot issues as they scale. Observability functions help organizations make better decisions and generate better performance and reliability outcomes across their service mesh and ultimately improve their application's performance.

4.1.7. Enterprise Support

Implementing modern solutions like service meshes is getting more complex by the day with new infrastructure platforms, tools, and services available to support teams. Finding a vendor with good enterprise support options is invaluable to any organizations looking to rapidly scale and achieve reliable and optimized performance fast.

Enterprise support helps technical teams attain stability faster through timely assistance and knowledge from certified expertise within the field. This helps organize minimize downtime, ramps down costs, and ultimately improves systems and network stability through fast bug fixes, customized solutions, and efficient user training.

With the infrastructure market rapidly changing, no two environments are the same, and the technical expertise generally offered by enterprise service mesh providers including health checks, professional services, and onboarding can be invaluable. It can minimize any security vulnerabilities, resolve any governance issues, and enable organizations to implement best practices tried and tested in the field.

Enterprise support provides teams within organizations with the confidence to deploy service mesh across their environment and helps improve the user satisfaction of the mesh solution.

4.1.8. Open Source Community

The open source community has played an instrumental role in the service mesh industry; it has single-handedly shaped what expectations of modern service meshes should encompass through several key projects across the open source community including projects like Istio, Linkerd, and Envoy.

The enterprise market for service mesh is competitive, each vendor carving functions and features to differentiate from each other. Some vendors prefer to create their own lock-in ecosystem whilst others build their solutions based on open source projects.
The value of an open source-based solution stems from the diversity of community contributors to projects, enabling better innovative features released and rapid iteration across different use cases and requirements because of its open feedback and contribution-based development cycle. This leads to an improved quality project and, by extension, enterprise product because of its active development usually leading to consistent release cycles.

An open source-based solution also provides the added advantage of a free knowledge base from the community, including documentation, events, workshops, and training. Enterprise service mesh users should investigate if utilizing an open source-based solution is a better fit for their environment.

5. Feature Analysis

5.1. Data Plane Options

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Envoy Compatibility	●	○	◗	●	●	●
VM Environments	●	◗	○	◕	○	◕
Extensibility	◕	◔	○	◕	○	◗
Deployment Modes	●	◔	◗	◔	◔	◔

5.1.1. Envoy Compatibility

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ○
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ●
- Anthos Service Mesh: ●
- Red Hat OSSM: ●

Gloo Mesh: Envoy is popularly adopted by the service mesh community and Solo utilizes Envoy Proxy as its data plane across its solutions. In Gloo Mesh, the Envoy-based data plane is used to handle inter-service communication and traffic management across the service mesh it manages.

It serves as an L3 or L4 application, sidecar proxy or L7 proxy in Gloo Mesh, and has the same lifecycle as the proxy's parent application to extend applications across multiple technology stacks including legacy applications that may not offer the extensibility. It also provides functionality to Gloo Mesh's traffic routing, load balancing, fault injection, and access control across microservices.

Buoyant Enterprise for Linkerd: Buoyant has not built Linkerd based on Envoy and instead built its service mesh solution based on its own dedicated proxy called 'Linkerd2-proxy'. Buoyant claims that this proxy is purpose-built for users looking to deploy in microservices and Kubernetes-based environments. The usability of this proxy and by extension the functionality of Buoyant Enterprise for Linkerd service mesh is limited as it is reliant on Buoyant for updates and improvements and is not extensible by design.

Isovalent Enterprise for Cilium: Isovalent Enterprise utilizes the Envoy proxy specifically for Layer 7 processing requirements including: ingress, gateway API, network policies with L7 functionality, and L7 protocol visibility. The Cilium agent starts an Envoy proxy as a separate process within the Cilium agent pod. It is deployed in a shared-node method where the Envoy is a separate process within the Cilium agent. This may cause difficulty for users looking to deploy and configure as the unit of scale is the entire node, as Envoy was not designed to be multi-tenant.

Hashicorp Consul: Hashicorp Consul supports Envoy as a proxy and configures by optionally exposing a gRPC service on a local agent. Hashicorp Consul can also configure Envoy sidecars to proxy traffic for HTTP 1.1, HTTP2, gRPC, and TCP-based protocols. For older editions, Envoy proxies can only proxy TCP traffic at Layer 4. Users can configure Layer 7 features in some configuration entries and provide users with the ability to customize Envoy configurations to their proxy services.

Anthos Service Mesh: Anthos Service Mesh has limited support for versions of Envoy as it depends on the instance of Anthos Service Mesh installed. Anthos GKE deployed on-premises includes a Layer 7 load balancer with an Envoy-based ingress controller.

Users utilizing Google's new control plane 'Traffic Director' for Google Cloud Service Mesh can deploy Envoy proxies with it as long as it meets the technical prerequisites to run alongside their applications. This is locked into the GCP subscription and is a separately billed component. You can also utilize Envoy and Traffic Director for observability metrics into GCP Cloud Trace and Cloud Logging.

Red Hat OSSM: Red Hat OSSM uses Istio for its data plane to run Envoy containers as a proxy controlling all network communications in and out of pods. All ingress and egress network traffic between services flows through the Envoy proxies. The Envoy proxies are the only Istio components that interact with data plane traffic.

5.1.2. Virtual Machine Environments

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◗
- Isovalent Enterprise for Cilium: ○
- Hashicorp Consul: ◕
- Anthos Service Mesh: ○
- Red Hat OSSM: ◕

Gloo Mesh: Users can onboard external workloads including virtual machines to an Istio service mesh with Gloo Mesh. This is done via a deployment of three agents to the virtual machine: an Istio sidecar agent, SPIRE agent, and an OpenTelemetry collector agent. By adding the Istio sidecar agent to your virtual machine workload, users can achieve full bidirectional communication between applications on the cluster's service mesh. The ability to run external workloads with Gloo Mesh extends the reach and functionality of a service mesh, delivering consistent outcomes across any instance including external (VMs and bare metal) workloads, hyperscalers, or on-premises.

Buoyant Enterprise for Linkerd: Linkerd supports external workloads outside of Kubernetes, including virtual machines, physical machines, and other non-Kubernetes locations into the Linkerd mesh. Buoyant Enterprise claims that this will provide a uniform layer of security, observability, and reliability across workloads utilizing the Linkerd mesh. This feature was recently released in February 2024 as part of the Linkerd 2.15, simultaneously with Buoyant's controversial new model for stable releases on Linkerd.

Isovalent Enterprise for Cilium: Incorporates Cilium Mesh which connects Kubernetes workloads with external workloads, including virtual machines and physical servers. Setting up support for external workloads on Kubernetes clusters on Cilium is still a beta feature with limitations for support of encryption of traffic to/from external workloads.

Hashicorp Consul: Supports virtualized environments to deliver consistent service mesh experiences across Kubernetes and virtual machine environments. Users configure, deploy, and bootstrap a Hashicorp Consul service onto their virtual machine, which then interacts with the Hashicorp Consul UI, CLI, and API.

Anthos Service Mesh: Anthos Service Mesh does not appear to support virtual machine workloads. There is limited information available for Anthos users to deploy Anthos Service Mesh on virtual machine workloads.

Red Hat OSSM: Managing virtualized workloads is relatively straight-forward as part of the Red Hat OpenShift Container Platform subscription, as OpenShift Virtualization is now integrated with Red Hat OSSM. Users can monitor and control traffic between pods that run VMs on the default pod network. Adding VMs to Red Hat OSSM is done via a sidecar injection into the VM configuration file to expose your VM as a service and view your application in the service mesh.

5.1.3. Extensibility

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ○
- Hashicorp Consul: ◕
- Anthos Service Mesh: ○
- Red Hat OSSM: ◗

Gloo Mesh: Gloo Mesh also enables users to create a WebAssembly filter and store it in an OCI-compliant registry. Users can then use the Gloo WasmDeploymentPolicy custom resource to apply the Wasm filter to your service mesh.

Users can customize the settings of Gloo Mesh Enterprise to meet their needs using Kubernetes CRDs and Helm charts. Helm installations allow for extensive customization of Gloo Mesh settings across product and sandbox setups.

Buoyant Enterprise for Linkerd: Three Buoyant Enterprise for Linkerd extensions can be added to enable additional functionality, including viz (metrics and visibility), jaeger (distributed tracing), and multicluster for cross cluster routing. Third party extensions can be utilized and require additional configuration through each extension's own CLI.

Isovalent Enterprise for Cilium: Envoy proxy is shipped with Isovalent Enterprise for Cilium, but there are minimal Envoy extensions and custom policy filters. Isovalent Enterprise uses this minimal distribution as its host proxy for enforcing HTTP and Layer 7 policies.

Hashicorp Consul: The service mesh deployments provide Envoy extensions available for users to modify their Hashicorp Consul-generated Envoy resources and add additional functionality. There are two methods to modify Envoy behaviors, either through escape hatches or using the EnvoyExtension parameter.

The Envoy extensions enable additional service mesh functionality in Hashicorp Consul by changing how the sidecar proxies behave. Hashicorp Consul currently supports six extensions, including external authorization, Lua, Lambda, OpenTelemetry Access Logging, property override, and WebAssembly.

Anthos Service Mesh: Anthos Service Mesh supports WebAssembly with limited information found on other extensions available.

Red Hat OSSM: Red Hat OpenShift enables its users to use WebAssembly extensions to add features into the OpenShift Service Mesh proxies. The Red Hat OpenShift Service Mesh extensions are Envoy HTTP Filters. Users write extensions using an SDK that exposes the proxy-wasm API and compile it to a WebAssembly module and then package that module into a container.

5.1.4. Deployment Modes (Sidecar and Sidecarless Design)

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◔
- Anthos Service Mesh: ◔
- Red Hat OSSM: ◔

Gloo Mesh: Built on Istio, Gloo Mesh offers both sidecar and sidecarless deployment modes providing service mesh with the best deployment options for their workloads. Solo led the creation of 'ambient mode' in Istio – the new sidecarless design – and is the first to commercially support it in the market. Gloo Mesh has integrations (in beta) with Istio's new ambient mode, which removes the requirement of running sidecars along applications in your mesh. With sidecarless deployments, users instead can use node-level ztunnels to route Layer 4 traffic between apps, and waypoint proxies to enforce Layer 7 traffic policies.

Some Gloo Mesh components including the Gloo Agent, Gloo insights engine, and Gloo analyzer can be deployed as sidecar containers to other pods to help reduce the amount of compute resourcing required to run Gloo Mesh.

Buoyant Enterprise for Linkerd: Buoyant Enterprise Linkerd v2.0 is a sidecar-based service mesh, which migrates away from the single proxy per node service mesh provided in v1.0. Linkerd is not built nor integrates with Istio and therefore cannot leverage ambient mode's sidecarless deployment option for service mesh.

Isovalent Enterprise for Cilium: Isovalent developed project Cilium to do networking with eBPF technology and subsequently added service mesh features like sidecar proxy deployment to its offering. Isovalent Enterprise for Cilium runs a per-node proxy model in lieu of a sidecar deployment and claims to be more performative as it moves parts of the application networking requirements directly onto the kernel. But there are limitations around security and navigating multi-tenancy workloads users should be aware of.

Hashicorp Consul: You can deploy sidecar proxies services in Hashicorp Consul, which run on the same node as the single service instance they handle traffic for. They can also be on the same virtual machine or running in a separate container in the same network namespace. Currently there is no sidecarless deployment method available on Hashicorp Consul. However, there is an open ticket in Hashicorp Consul's GitHub repository requesting the need for a sidecarless deployment model.

Anthos Service Mesh: There are two sidecar deployment methods as part of Anthos Service Mesh. As part of 'Traffic Director' users can set up an Envoy sidecar service mesh. This feature has limited support as it is considered 'pre-GA.'

GKE Enterprise users can inject sidecar proxies with Anthos Service Mesh. Automatic sidecar proxy injection occurs when Anthos Service Mesh detects a namespace label that users configure to the workload's Pod. The proxy intercepts all inbound and outbound traffic to the workloads and communicates with Anthos Service Mesh.Istio's ambient mode is not currently a supported feature in Red Hat OpenShift's Service Mesh.

Google does not currently support or indicate plans to release and/or integrate a sidecarless deployment method across their service mesh offerings.

Red Hat OSSM: The Red Hat OSSM data plane is a set of intelligent proxies deployed as sidecars. These proxies intercept and control all inbound and outbound network communication between microservices in the service mesh. Envoy proxy intercepts all inbound and outbound traffic for all services in the service mesh. Envoy is deployed as a sidecar to the relevant service in the same pod.

Istio's ambient mode is not currently a supported feature in Red Hat OpenShift's Service Mesh. However, documentation from Red Hat states as the feature matures in upstream Istio it will be supported by Red Hat at a later date.

5.2. Control Plane Functions

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Host Options	●	◗	◔	●	◕	◕
Tenancy	●	◔	◔	◗	◗	◕
Scalability	●	◔	◔	●	◗	◕
High Availability	●	●	◗	◕	◕	◗
Resiliency	◕	◕	◔	◕	◗	◕

5.2.1. Host Options

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◗
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ●
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh can be deployed across multiple host options via CLI and Helm. Users can unify the configuration, operations, and visibility on service connectivity across their distributed applications with Gloo Mesh. Applications can run in different hosted environments including: virtual machines (VMs), Kubernetes clusters across different distributions on premises or via various cloud providers including major hyperscalers, and even in different service meshes.

Buoyant Enterprise for Linkerd: The control plane can be installed in two ways, either via the CLI or with Helm. Buoyant Enterprise for Linkerd can be deployed on any Kubernetes distribution and enables users to mix and match their deployment across different host options, including on premise and in the cloud.

Isovalent Enterprise for Cilium: Positions themselves as cloud agnostic and Cilium can be installed on any Kubernetes cluster via a Helm repository. Designed deployment pathways for major hyperscalers' distributions and popular community and enterprise distributions and environments, including Red Hat, Rancher, and Alibaba.

Hashicorp Consul: Hashicorp Consul can be installed via CLI or Helm. For Kubernetes workloads that are multi-cluster installations and involve cross-partition or cross datacenter communication, it is recommended that deployment is via the Hashicorp Consul Helm chart. For heterogeneous workloads, Hashicorp Consul agents can join a server running inside or outside of Kubernetes. For single cluster deployments, Hashicorp Consul K8s CLI tool enables users to quickly install and interact with Hashicorp Consul on Kubernetes.

Anthos Service Mesh: For GKE Enterprise users they can utilize a managed service mesh where their workloads and clusters are enrolled based on their GKE release channel.

For users with GKE clusters outside of Google Cloud ecosystem, there is an option to utilize Anthos Service Mesh with an in-cluster control plane where Google supports the current and previous two minor versions of Anthos Service Mesh. However, there are differences in supportability and functionality between the two Google service mesh offerings

.

Red Hat OSSM: Various host options are available to users and are part of the OpenShift Container Platform offering, including major hyperscalers, on premise, and VM providers. Users must subscribe to the OpenShift Container Platform and configure it to their infrastructure requirements. Installation is via the Red Hat OpenShift CLI.

5.2.2. Tenancy

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Users can set up multi-tenancy for their service mesh with Gloo Mesh. Gloo introduces a new concept for Kubernetes-based multitenancy, the Workspace custom resource. A workspace consists of one or more Kubernetes namespaces that are in one or more clusters. Users can create a workspace for teams and each team can start building their applications in a few Kubernetes namespaces within a single cluster. As teams scale across namespaces and clusters the workspaces scale alongside.

This functionality in Gloo Mesh helps provide boundaries for Gloo Mesh, Istio, and Kubernetes resources to be accessed across other workspaces. The boundaries enable teams to simplify managing services, security, and shared settings across teams within a single organization.

Buoyant Enterprise for Linkerd: Buoyant has not released a dedicated feature for multi-tenancy functionality within Linkerd, but positions the solution to be performant in multi-tenant environments. There are third party tools that Linkerd can integrate with to configure its service mesh solution to deliver multi-tenancy services in Kubernetes.

Isovalent Enterprise for Cilium: Isovalent's Enterprise edition of Cilium has a functionality called 'Multi-Network,' which provides the ability to connect Kubernetes Pods to multiple network interfaces. This feature can be used alongside Cilium Network Policies to isolate network traffic between tenants by assigning different interfaces to different tenants or namespaces.
Hashicorp Consul: Hashicorp Consul provides the feature called 'Administrative Partitions' that addresses the common resource inefficiencies faced by users who run multiple service meshes across different teams. The feature consolidates Hashicorp Consul deployments into a single Hashicorp Consul control plane and provides tenant separation via partition groups.

Anthos Service Mesh: For GKE Enterprise users of Anthos Service Mesh, they can utilize Kubernetes namespaces for administrative control. Anthos Service Mesh builds on Kubernetes namespaces by using them as a unit of tenancy within a service mesh as per Istio documentation.

Traffic Director users can implement 'Fleet' on Google Cloud to organize clusters and resources. Only Google Cloud project resources can be grouped logically and managed as fleets and only Kubernetes clusters can be fleet members.

Red Hat OSSM: Red Hat OSSM comes with ServiceMeshControlPlane pre-configured for multi-tenancy by default. It employs a multitenant operator to oversee the lifecycle of the service mesh control plane. Within a mesh, namespaces serve as the means for tenancy.

In its support for "soft multi-tenancy," Red Hat OSSM facilitates one control plane and one mesh per tenant, allowing for the existence of multiple independent control planes within the cluster. Multitenant deployments delineate the projects with access to the service mesh while ensuring its isolation from other control plane instances.

5.2.3. Scalability

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ●
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Solo runs internal scalability tests for every release to confirm that the translation and user experience times remain within expected boundaries, and to measure scalability and performance improvements between releases. Gloo Mesh runs Istiod as the control plane and is considered performant when Gloo resources that the user creates are translated into Envoy and Istio resources, and are applied in the user's environment in a reasonable amount of time. Gloo Mesh also utilizes the Trim proxy configuration to improve performance.

Solo's internal scalability tests show that the Gloo Mesh using workspaces efficiently and clearly defining which services to import/export can minimize the configuration translation load and improve scalability.

Buoyant Enterprise for Linkerd: Buoyant states that it has designed Linkerd to consume the least amount of memory and CPU as possible with regards to data plane proxies. For the control plane, they state that Linkerd can "scale gracefully." There are no scalability tests available to showcase the effectiveness of these claims.

Isovalent Enterprise for Cilium: Isovalent runs scalability tests on Cilium based on users planning to run Cilium on clusters with more than 200 nodes in CRD mode.

There are some performance limitations users can expect. More work from the Isovalent and Cilium project team has to be done in improving the optimization of the memory footprint of eBPF maps and reducing the memory footprint of the Cilium agents. More tests are also required to verify Cilium's behavior when it loses connectivity with the kube-apiserver during upgrades to the control plane.

Hashicorp Consul: Is able to run clusters with 10,000+ nodes, but it takes longer to bring deployments back online after an outage, impacting time to recovery. The recommendation for large service deployments is to limit it to 5,000 Hashicorp Consul client agents per Hashicorp Consul datacenter.

This is to reduce the blast radius of Hashicorp Consul clients or dataplanes that may have been impacted in an outage. The 5,000 agent limit also helps manage agent gossip. Hashicorp Consul agents use gossip protocol to share information and whenever an agent joins or leaves the gossip pool the other agents propagate that event through the pool, so if there is churn with agents within the pool, performance can be affected.

In Kubernetes environments, it is possible to deploy Hashicorp Consul agents inside pods alongside services running in the same pod, however this unsupported deployment pattern has known performance issues at scale. At large volumes, pod registration and deregistration in Kubernetes causes gossip instability that can lead to cascading failures as services are marked unhealthy, resulting in further cluster churn.

Anthos Service Mesh: Istiod scales well vertically with large requests and horizontally for replicas. Users can ensure that CPU limits are not restrictive and if Istiod reaches the CPU limit, throttling may occur, which will negatively affect configuration distribution.

Users should also be aware that there are some limitations when large changes in cluster sizes can cause unbalanced loads due to long-lived connections. Users can mitigate this issue by having multiple replicas of Istiod and pre-scaling if they expect extreme scale up of clusters.

Red Hat OSSM: The Red Hat OSSM control plane is also powered by Istiod. The control plane supports thousands of services, spread across thousands of pods with a similar number of user-authored virtual services and other configuration objects. Istiod's CPU and memory requirements scale with the number of configurations and possible system states. The CPU consumption scales with the rate of deployment changes, configuration changes, and the number of proxies connecting to Istiod.

5.2.4. High Availability

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ●
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◗

Gloo Mesh: Users can improve the resiliency of their Gloo management plane through horizontal replica scaling and deploying multiple management clusters. Using these methods together they can build high availability and resilient architecture.

Horizontal replica scaling enables both resilience and distributed scaling by increasing the replica count of the management server deployment within one Kubernetes cluster. Each replica handles a subset of connected agents in workload clusters. If a replica pod fails, the agents that were connected to that management server replica automatically connect to a different replica.

The management server deployment can span multiple availability zones, which can provide resilience. Users can enable high availability at the level of availability zones by ensuring that replicas are deployed to worker nodes that run in different zones. For multiple management clusters you can add redundancy by deploying a Gloo management server across multiple Kubernetes clusters.

Buoyant Enterprise for Linkerd: Can run production workloads in high availability mode with its enterprise feature 'high availability zonal load balancing'. This requires running three replicas of critical control plane components and setting production-ready CPU and memory resource requests on control plane components.

It also requires users to set production-ready CPU and memory resource requests on data plane proxies, requires the proxy auto-injector be functional for any pods to be scheduled, and sets anti-affinity policies on critical control plane components to ensure that they are scheduled on separate nodes in separate zones by default.

Isovalent Enterprise for Cilium: Isovalent Enterprise incorporates Cluster Mesh, their multi-cluster implementation, to enhance services' high availability and fault toleration. If resources become temporarily unavailable or are misconfigured in one cluster or go offline for upgrades, it enables failover to other clusters, ensuring your services remain accessible at all times.

Hashicorp Consul: Hashicorp Consul should be deployed with 5 nodes within the Hashicorp Consul cluster, distributed between three availability zones, as this architecture can withstand the loss of two nodes from within the cluster or the loss of an entire availability zone. For Hashicorp Consul Enterprise customers, additional resiliency is possible by implementing a multi-cluster architecture, which allows for additional performance and disaster recovery options.

Users can also utilize Hashicorp Vault with Hashicorp Consul to run high availability mode to protect against outages by running multiple Vault servers. When running in high availability mode, Vault servers have two additional states: standby and active.

Anthos Service Mesh: Anthos Service Mesh supports high availability configurations in primary clusters for GKE Enterprise users. A single mesh can have more than one primary cluster for high availability or to reduce latency.

For multiple primary deployments, control planes can be replicated as per Istio documentations, stating that for isolation, performance, and high availability requirements, users can confine clusters to availability zones and regions.

Red Hat OSSM: Provides a multimesh or federated deployment model to let users share services and workloads between separate meshes managed in a distinct administrative domain. Red Hat OSSM utilizes the Istio multi-cluster model that requires trust between meshes and remote access to all Kubernetes API servers where individual meshes reside.

A federated mesh is a group of meshes behaving as a single mesh. The services in each mesh can be unique services, for example a mesh adding services by importing them from another mesh can provide additional workloads for the same services across the meshes, providing high availability, or a combination of both. In Red Hat OSSM all meshes that are joined into a federated mesh remain managed individually, and you must explicitly configure which services are exported to and imported from other meshes in the federation.

5.2.5. Resiliency

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh uses policies to control traffic within the service mesh environment to ensure network resiliency and security of microservices. Policies supporting Gloo Mesh's resilience capabilities include: adaptive request concurrency, failover, fault injection, outlier detection, retries, timeouts, TCP connection, and Trim proxy configuration.

Buoyant Enterprise for Linkerd: Buoyant Enterprise for Linkerd has built some resiliency features including fault injections, TCP proxying and protocol detections, retries, and timeouts.
Isovalent Enterprise for Cilium: Isovalent Enterprise positions itself as being designed at the networking and application protocol layer that is inclusive of resiliency requirements, including network processing protocols such as IP and TCP. Detailed resiliency capabilities are limited in documentation.

Hashicorp Consul: Hashicorp Consul helps organizations build their applications' resiliency through proactive measures across their enterprise subscription of Hashicorp Consul including: automated upgrades, enhanced read scalability, advanced federation for data centers, and network segments to restrict LAN gossip between client and server agents. Hashicorp Consul also suggests hardened recovery processes by implementing redundancy zones and automating backups.

Anthos Service Mesh: For GKE Enterprise users of Anthos Service Mesh there are built in resiliency features available including traffic controls, out of the box recovery, fault injection tools, retries, timeouts, and redirects.

Red Hat OSSM: Service entries to the service registry are maintained by Red Hat OpenShift Service Mesh internally. Once the service entry is added, the Envoy proxies send traffic to the service as if it is a service in your mesh allowing you to manage traffic for services, redirect and forward traffic for external destinations or services to legacy infrastructure, define resilience policies including retries, timeouts and fault injections, or run a mesh service in a VM.

5.3. Security and Compliance

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
mTLS/Mutual Authentication	●	●	◔	●	◕	◕
SPIFFE Integration	●	●	◗	◕	◗	◗
Attribute-Based Authentication	●	◕	◔	◗	◕	◕
Policy Management	●	◕	◔	◗	◕	●

5.3.1. mTLS/Mutual Authentication

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ●
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ●
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◕

Gloo Mesh: Mutual TLS is used to secure both the relay connection between Gloo management service and agent to ensure both identities are valid and between workloads across the service mesh.

There are three main approaches to mTLS and certificate management required with Gloo Mesh including: self-signed server certificate with managed client certificates, bring-your-own server certificate with managed client certificate, or bring-your-own server and client certificate with OpenSSL, AWS, or Vault.

It is important to note that the server TLS certificates are not automatically rotated, and across all three certification variants it is recommended that users utilize tools like cert-manager or use a preferred PKI provider to manage the certificate's lifecycle.

Buoyant Enterprise for Linkerd: Automatically enables mutually-authenticated Transport Layer Security (mTLS) for all TCP traffic between meshed pods, and adding authenticated, encrypted communication to users' applications with additional configuration.

For production use or cross-cluster traffic, additional preparation is required. This includes the trust anchor generated by the default Linkerd install CLI command that expires after 365 days. After that, it must be manually rotated. Similarly, the default cluster issuer certificate and key expire after a year. These must be rotated before they expire.

Isovalent Enterprise for Cilium: mTLS support in Isovalent Enterprise is through Cilium Service Mesh and is currently a beta feature. Several core security features are yet to be implemented and there is no current option to build a single trust domain across multiple clusters for combining clusters for across Cilium cluster mesh and service mesh. The current support of mutual authentication also only works within a Cilium-managed cluster and is not compatible with an external mTLS solution.

The design of this mutual authentication is eventually consistent, as the security implementation's data path can lead to failures in the security properties and cause traffic to proceed through services though disallowed.

Hashicorp Consul: Core functionality of Hashicorp Consul is based on mTLS where Hashicorp Consul service mesh provides each service with an identity encoded as a TLS certificate. This certificate is used to establish and accept connections to and from other services. The identity is encoded in the TLS certificate in compliance with the X.509 SPIFFE Verifiable Identity Document (SVID). This enables Hashicorp Consul service mesh services to establish and accept connections with other SPIFFE-compliant systems.

Anthos Service Mesh: Anthos Service Mesh GKE Enterprise users can enable auto mTLS by default, where a client sidecar proxy automatically detects if the server has a sidecar. The client sidecar proxy automatically detects if the server has a sidecar.

The client sidecar sends mTLS to workloads with sidecars and sends plaintext to workloads without sidecars. Services accept both plaintext and mTLS traffic, and users, as they inject sidecar proxies to Pods, are recommended to also configure services to only accept mTLS traffic.

Red Hat OSSM: By default, mTLS in Red Hat OpenShift Service Mesh is enabled and set to permissive mode, where the sidecars in service mesh accept both plain-text traffic and connections that are encrypted using mTLS.

If a service in your mesh is communicating with a service outside the mesh, strict mTLS could break communication between those services. Using permissive mode while migrating workloads to service mesh, users can then enable strict mTLS across the mesh, namespace, or application.

5.3.2. SPIFFE integration

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ●
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◗

Gloo Mesh: Gloo Mesh uses Istio, implements the SPIFFE specifications, and also supports replacing the implementation with a SPIRE server and agents. Users are required to deploy three agents into their virtual machine including an Istio sidecar, SPIRE agent, and OpenTelemetry collector agent.

Gloo Mesh supports SPIFFE as the workload identity mechanism and allows users to write security policies in terms of these identities to account for regulatory and compliance restrictions.

Buoyant Enterprise for Linkerd: Utilizes SPIFFE to provide a consistent layer of cryptographic identity and authentication to off-cluster workloads. Buoyant Enterprise for Linkerd generates SPIFFE IDs for non-Kubernetes workloads using SPIRE, and the IDs can be used alongside Linkerd's existing ServiceAccount-based IDs.

Users can encrypt all traffic to VM workloads by default, and add zero-trust controls over all access right down to the level of individual HTTP routes and gRPC methods for specific clients.

Isovalent Enterprise for Cilium: Through Cilium Service Mesh's current mutual authentication support, identity management is through SPIFFE. Isovalent claims that users can now deploy Cilium with a SPIRE server and enable Cilium mutual authentication flag. Workloads will then have their identities created by the SPIRE component (or server) to automatically manage and rotate certificates.

Hashicorp Consul: The TLS certificate that Hashicorp Consul provides for each service is in compliance with SPIFFE X.509 Identity Document, which enables Hashicorp Consul services to establish and accept other SPIFFE-compliant systems.

Anthos Service Mesh: Users can configure secure naming with Traffic Director, enabling them to define a list of allowed names, or SPIFFE identities, for a particular service that a client is attempting to connect to. During the TLS exchange, the service's backend returns an X.509 certificate to the client. The client then inspects the certificate to confirm that the X.509 certificate matches one of the names or SPIFFE identities.

Red Hat OSSM: Since Red Hat OSSM is based on Istio, by extension it integrates with SPIFFE or SPIRE protocols.

5.3.3. Attribute-Based Authentication

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh utilizes Open Policy Agent (OPA) policies to enable users with more fine-grained access control across their service mesh as part of an attribute-based access control function.

There are three main ways to implement OPA with Gloo's external authentication service including: loading the Rego rules from a Kubernetes configuration map as an OPA module, running an OPA server as a sidecar to the external authentication service, or bringing your own OPA server. You can use a mix of implementations across the same cluster, but they require separate external authentication policies for each implementation.

Buoyant Enterprise for Linkerd: There are two mechanisms to configure Buoyant Enterprise for Linkerd policy including: a set of default policies set at the cluster, namespace, workload, and pod level, and a set of CRDs that specify fine-grained policy for specific ports, routes, and workloads. For fine-grained policies CRDs can be changed dynamically and policy behavior will be updated on the fly. The general pattern for Linkerd's dynamic, fine-grained policy is to define the traffic target that must be protected, define the types of authentication that are required before access to that traffic is permitted, and then define the policy that maps authentication to the target.

Isovalent Enterprise for Cilium: In Isovalent Enterprise's current mutual authentication support, identity management is provided through the use of SPIFFE. There is little additional information available in the future roadmap for attribute-based authentication.

Hashicorp Consul: Hashicorp Consul utilizes Hashicorp Vault as a system to store credentials and add authentication capabilities, where administrators can create Vault credentials for each user and assign per role or attribute to give them access to keys.

Hashicorp Consul also uses access control lists to help manage authenticate requests and access to resources. Hashicorp Consul's authentication method is a component that performs authentication against trusted external parties to authorize the creation of ACL tokens.

Anthos Service Mesh: Anthos Service Mesh for GKE Enterprise utilizes Identity-Aware Proxy (IAP) to help users safely access services based off of Google's BeyondCorp principles. IAP verifies user identity and context of the request to determine if a user should be allowed to access an application or resource.

IAP integration allows complete context-awareness control to workloads running on Anthos Service Mesh, where users get fine-grained access policies based on attributes of the original request, and is scalable, secure, and highly available across applications through the Google Cloud load balancer.

Red Hat OSSM: Red Hat OSSM implements fine-grained policies to facilitate zero-trust networking through the OpenShift Container Platform, where administrators can use groups to manage users and change their permissions to improve collaboration. Specifically for Red Hat OSSM, they have made external authorization generally available, which utilizes Istio's authorization policy to delegate access control to external authorization systems and enables users to integrate with providers like OPA or oauth2proxy.

5.3.4. Policy Management

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◕
- Red Hat OSSM: ●

Gloo Mesh: Gloo Mesh allows users to manipulate and control network traffic to the applications in your cluster via policy management. Policy behavior depends on several factors that users set up when creating the policy, including which policy they apply, if there are multiple policies to the same source and the filter order for applied policies.

Users have greater control over how and where to apply policies, as they can be applied to different resources including destinations, listeners, routes, and workloads. Policies can also be applied using Kubernetes labels and selections that match the route table, virtual destination, or workload as long as they are in the same workspace for the policy to apply the resource.

Buoyant Enterprise for Linkerd: Buoyant Enterprise for Linkerd has its own authorization policy method, which allows users to control which types of traffic are allowed to meshed pods. By default, Linkerd allows all traffic to transit the mesh, and uses a variety of mechanisms, including retries and load balancing, to ensure that requests are delivered successfully. Its policy features allow you to deny access to resources unless certain conditions are met, including the TLS identity of the client.

Buoyant Enterprise for Linkerd: Buoyant Enterprise for Linkerd has its own authorization policy method, which allows users to control which types of traffic are allowed to meshed pods. By default, Linkerd allows all traffic to transit the mesh, and uses a variety of mechanisms, including retries and load balancing, to ensure that requests are delivered successfully. Its policy features allow you to deny access to resources unless certain conditions are met, including the TLS identity of the client.

Isovalent Enterprise for Cilium: Part of Isovalent Enterprise is Cilium Service Mesh which is reliant on the network policy configurations of Cilium. The configuration of the Cilium agent and the Cilium Network Policy determines whether an endpoint accepts traffic from a source or not.

All policy rules are based upon a whitelist model and each rule in the policy allows traffic that matches the rule. If any traffic does not match any rules it is dropped pursuant to the Cilium policy enforcement modes.

There are three policy enforcement modes that agents can be put into including default – the default behavior for policy enforcement without any specified value, always where policy is enabled on all endpoints and never where policy enforcement is disabled on all endpoints.

Hashicorp Consul: Within Hashicorp Consul, policies are a group of one or more access control list (ACL) rules linked to ACL tokens. Rules are one of several attributes that form a policy. Rules are building blocks that define access to resources. Users need to assemble rules into policies based off of the ACL.

The `acl policy` command is used to manage Hashicorp Consul ACL policies. It exposes commands for creating, updating, reading, deleting, and listing policies.

In addition to directly-linked policies, roles, and service identities, Hashicorp Consul Enterprise enables ACL policies and roles to be defined in the Namespaces definition.

Anthos Service Mesh: Anthos Service Mesh security policies including authentication and authorization policies should be enforced on all traffic in and out of the mesh, and its authorization policy provides mesh, namespace, and workload-level access control for your workloads in the mesh. Users can test Anthos Service Mesh authorization policy support in dry-run mode without any real production traffic and without enforcing it to help better understand the effect of a policy. Users can also integrate GKE's Policy Controller with Anthos Service Mesh to ensure security best practices and application and enforcement of programmable policies across Kubernetes clusters. Policy Controller is based on the OPA Gatekeeper project and is integrated with the Google Cloud console and includes a built-in dashboard, pre-built policies, and compliance controls. It is only available via the GKE enterprise edition license.

Red Hat OSSM: Red Hat OpenShift Service Mesh automatically creates and manages a number of network policies' resources in the service mesh control plane and application namespaces. This is to ensure that applications and the control plane can communicate with each other.
Different administration profiles are available to manage policies within a service mesh.
Administrators are able to delegate permissions to add projects to a service mesh and project administrators automatically are given permission to create `ServiceMeshMember` resources in projects, but cannot link it to a `ServiceMeshControlPlane` without administrator approval.

5.4. Network Integrations

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Gateway Support	●	◔	◔	◕	●	●
Multi-cluster Support	●	◕	◔	◗	◕	◕
Flat Network	◕	◗	◕	◔	●	◕
Federated Set Up	●	○	○	◕	◔	●

5.4.1. Gateway API Support

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◕
- Anthos Service Mesh: ●
- Red Hat OSSM: ●

Gloo Mesh: Users can install Istio ingress, egress, and east-west gateways into their clusters to help with Istio lifecycle management. In each `GatewayLifecycleManager` resource, users provide gateway settings in an `IstioOperator` configuration.

When they create the `GatewayLifecycleManager` in the management cluster, Gloo translates the configuration into gateways in your registered workload clusters for you. As part of the Solo portfolio, Gloo Mesh also integrates with Gloo Gateway, which utilizes Envoy and also supports the Kubernetes Gateway API.

Buoyant Enterprise for Linkerd: Utilizes gateways in 'hierarchical mode' for multi-cluster communications alongside its multi-cluster functionality, which allows pods to connect to Kubernetes services across cluster boundaries in a secure and fully transparent way.

Hierarchical mode requires that the gateway IP of the destination cluster be reachable by pods on the source cluster. It can be mixed and matched with the 'flat mode' multi-cluster communication, which requires pods on the source cluster to be able to directly connect to pods on the destination cluster.

Isovalent Enterprise for Cilium: Cilium Service Mesh supports Gateway API, and a deployment generally has two parts that handle Gateway API resources: the Cilium agent and the Cilium operator. The Cilium operator watches all Gateway API resources and verifies whether the resources are valid. If resources are valid and accepted the process of translation into Cilium Envoy Configuration resources starts. The Cilium agent picks up the Cilium Envoy Configuration resources and uses the resources to supply the configuration to the built-in Envoy or the Envoy DaemonSet.

Hashicorp Consul: Multiple types of gateways are shipped with Hashicorp Consul, designed to provide connectivity into, out of, and between Hashicorp Consul service meshes. There are three types of gateways Hashicorp Consul users can configure including:

1. Mesh gateways that enable service-to-service traffic between Hashicorp Consul datacenters and admin partitions.

2. Ingress gateways that enable connectivity within organizational networks from services outside the Hashicorp Consul service mesh to services inside the mesh.

3. Terminating gateways that enable connectivity within organizational networks from services in the Hashicorp Consul service mesh to outside the mesh.

Anthos Service Mesh: For GKE Enterprise users, Anthos Service Mesh gives you the option to deploy and manage gateways as part of your service mesh. Gateways are Envoy proxies that provide you with fine-grained control over traffic entering and leaving the mesh. Gateways are primarily used to manage ingress traffic, but you can also configure gateways to manage other types of traffic, including egress gateways and east-west gateways.

Configuring gateway support with Traffic Director is more complex and requires users to utilize Google Cloud Load Balancing to get traffic into your mesh and provide a managed ingress experience.

Red Hat OSSM: As part of Red Hat OSSM, users can use a gateway to manage inbound and outbound traffic for your mesh to specify which traffic you want to enter or leave the mesh. Gateway configurations are applied to standalone Envoy proxies that are running at the edge of the mesh, rather than sidecar Envoy proxies running alongside your service workloads. Red Hat OSSM gateway resource can layer 4-6 load balancing properties, such as ports, to expose and configure Red Hat OSSM TLS settings.

5.4.2. Multi-Cluster Support

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh has a multicluster and multimesh management plane that is based on hardened open source projects like Envoy and Istio. Users can unify the configuration, operation, and visibility of service-to-service connectivity across their distributed applications. These applications can run in different virtual machines (VMs) or Kubernetes clusters on premises or in various cloud providers, and even in different service meshes.

Buoyant Enterprise for Linkerd: Can connect Kubernetes services across cluster boundaries securely and fully transparent to the application, and independent of network topology. This multi-cluster function has been designed to provide a unified trust domain, separate cluster failures with domains, support any network type, and provide a unified model alongside in-cluster communication.

Buoyant Enterprise for Linkerd's cross-cluster connections are transparent to the application code and will establish a connection between clusters that is reliably encrypted and authenticated on both sides with mTLS.

Isovalent Enterprise for Cilium: Multi-cluster networking and support is available through Isovalent Enterprise for Cilium with Cluster Mesh. Cluster Mesh extends the networking datapath across multiple clusters. It allows endpoints in all connected clusters to communicate while providing full policy enforcement and allows multiple clusters to join into a large unified network, regardless of the Kubernetes distribution or location each of them is running.

Hashicorp Consul: Users can deploy multiple Hashicorp Consul clusters across multiple datacenters with basic or advanced federation topologies. There are two approaches to

federating multiple Hashicorp Consul clusters together, including a basic WAN federation that allows for discovery in all datacenter locations or cloud regions to communicate with each other, or the advanced WAN federation with network areas, which allows for service discovery and service mesh within a defined data center location. Its advanced WAN federation approach requires an Hashicorp enterprise license.

Anthos Service Mesh: GKE Enterprise users of Anthos Service Mesh can implement 'fleets' to help manage multi-cluster deployments. Fleets are a Google Cloud concept for logically organizing clusters and other resources, letting you use and manage multi-cluster capabilities and apply consistent policies across your systems. Fleets form a crucial part of how enterprise multi-cluster functionality works in Google Cloud.

GKE Enterprise Anthos Service Mesh users can also enable cross-cluster load balancing to join two clusters in a single mesh using Mesh CA or Istio CA. Setting up multi-cluster mesh outside of the Google environment is also possible for GKE workloads on VMware, Azure, Bare Metal, and AWS.

Red Hat OSSM: Users can set up Istio Multi Cluster on Red Hat OSSM as part of their multimesh or federated deployment model. The Istio multi-cluster model requires a high level of trust between meshes and remote access to all Kubernetes API servers on which the individual meshes reside. Red Hat OpenShift Service Mesh federation takes an opinionated approach to a multi-cluster implementation of Service Mesh that assumes minimal trust between meshes.

5.4.3. Flat Network

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◗
- Isovalent Enterprise for Cilium: ◕
- Hashicorp Consul: ◔
- Anthos Service Mesh: ●
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh supports flat network topology as part of Gloo Mesh's multi-mesh and multi-cluster capabilities. Users utilizing a flat network to reduce overhead of ingress and allow for direct service-to-service connectivity across clusters can find support with Gloo Mesh, in addition to its multi-network capabilities, from Gloo's centralized management plane that can be used to manage services and policies across workloads spanning clusters.

Buoyant Enterprise for Linkerd: Supports pod-to-pod communication for clusters that share a flat network, where pods can establish TCP connections and send traffic directly to each other

across cluster boundaries. In these environments, Buoyant Enterprise for Linkerd does not use a gateway intermediary for data plane traffic.

Isovalent Enterprise for Cilium: Leverages Hubble, a distributed networking and security observability platform to deliver simple flat Layer 3 networking that provides its users with the ability to span multiple clusters connected to all application containers. IP allocation is kept simple by using host scope allocators, enabling each host to allocate IPs without any coordination between hosts.

Hashicorp Consul: Users can utilize Hashicorp Consul to adopt a service mesh that enables flat and fully encrypted networks across multiple environments. For users with multiple Kubernetes environments that run multi-cluster service discovery and request routing, they can utilize Hashicorp Consul to secure traffic routing across the different environments based on service level identity rather than the IP address, and flatten the network.

Anthos Service Mesh: As part of GKE, users can implement a fully integrated flat network model that offers ease of communications with applications outside Kubernetes and other Kubernetes clusters. With GKE, it also ensures compatibility with service meshes including Istio and Anthos Service Mesh, which can communicate across clusters because Pods communicate directly with each other.

Red Hat OSSM: Red Hat OpenShift Container Platform cluster uses a virtualized network for pod and service networks called OpenShift Virtualization. It supports the flat layer 2 topology and connects workloads by a cluster-wide logical switch.

The Open Virtual Network (OVN)-Kubernetes network plugin is the default network provider for OpenShift Container Platform and replaces the previous OpenShift SDN network plug-in. OVN-Kubernetes is an open-source, fully-featured Kubernetes CNI plugin that uses Open Virtual Network (OVN) to manage network traffic flows. It implements Kubernetes network policy support including ingress and egress rules and utilizes Geneve protocol to create an overlay network between nodes.

5.4.4. Federated Set Up

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ○
- Isovalent Enterprise for Cilium: ○
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◔
- Red Hat OSSM: ●

Gloo Mesh: Gloo Mesh federates Gloo and Kubernetes resources so that services can communicate with each other across clusters within the workspace. Gloo can federate ungrouped services for an entire workspace, or groupings of services that you define in select Gloo custom resources. You can also use both types of federation together, and behavior is determined by whether the services that back the host that you call are ungrouped or grouped.

Buoyant Enterprise for Linkerd: There is no explicit federated set up methodology from Buoyant Enterprise for Linkerd. It is reliant on its multi-cluster feature that enables Linkerd to connect Kubernetes services across cluster boundaries, independent of network topology. This provides a unified trust domain, separates failures, and supports heterogeneous networks and a unified model for in-cluster communications.

Isovalent Enterprise for Cilium: Isovalent Enterprise does not have an explicit federated pathway for its service mesh. The architecture of the Cilium control plane offers some similarities to typical federated meshes, including networks between clusters allowing inter-cluster communications, policies, and rules across clusters and cluster failure guard rails.

Hashicorp Consul: In Hashicorp Consul, federation is the act of joining two or more Hashicorp Consul datacenters. When datacenters are joined, Hashicorp Consul servers in each datacenter can communicate with one another. Hashicorp Consul's federated set up enables service on all clusters to make calls to each other through the service mesh, intentions can be used to enforce rules around service communication to clusters, layer 7 routing rules can enable multi-cluster failover and traffic splitting.

Anthos Service Mesh: Anthos users need to rely on GKE Enterprise and Traffic Director multi-cluster management to deploy services across multiple clusters. There is no inherent Anthos Service Mesh federated set up available.

Integration with Anthos Service Mesh can be done via Google Fleet, which lets users manage capabilities and policies consistently across multiple clusters. Fleet uses workload identity federation to provide each application with a distinct federated identity that can be used to authenticate services to and from Google Cloud.

Red Hat OSSM: Red Hat OpenShift Service Mesh federation takes an opinionated approach to a multi-cluster implementation of service mesh that assumes minimal trust between meshes. It assumes that each mesh is managed individually and retains its own administrator. The default behavior is that no communication is permitted and no information is shared. The sharing of information between meshes is on an explicit opt-in basis. Support functions such as certificate generation, metrics, and trace collection remain local in their respective meshes.

5.5. Performance Benchmarking

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Latency Analysis	●	◕	◗	◗	◗	●
Benchmarking	◕	◕	◔	◕	◔	◔
Resource Utilization	◕	◗	◔	◕	◗	◕
Health Checks	●	●	●	●	●	●

5.5.1. Latency Analysis

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◗
- Red Hat OSSM: ●

Gloo Mesh: Performance and latency of requests are managed by the Gloo telemetry pipeline powered by open source project OpenTelemetry. The Gloo telemetry pipeline integrates with Jaeger as the tracing platform. Jaeger is an open source tool that follows the path of a request as it is forwarded between microservices.

The events and interactions are captured by the Gloo telemetry pipeline and visualized in the Jaeger UI within Gloo Mesh UI. Gloo Mesh users can utilize Jaeger to identify any service latency through the mapped data events. For Istio enabled workloads, users need to instrument their workloads to generate traces and send them to the Gloo telemetry pipeline and Jaeger instance.

Buoyant Enterprise for Linkerd: Utilizes Jaeger to help users trace, debug, and identify any bottlenecks and understand the latency cost of each component across systems. It can be configured to emit trace spans from proxies for administrators to see what time requests and responses spend inside systems.

Buoyant Enterprise for Linkerd provides additional features to help analysis, including line service topology and dependency graphs, aggregated service health, aggravated path/route health, latencies, and request values.

Isovalent Enterprise for Cilium: Isovalent Enterprise positions its 'bandwidth manager' as an alternative to tracing and performance analysis. Bandwidth manager is used to optimize TCP and UDP workloads and efficiently rate limit individual Pods to reduce latency across networks. However, there is currently limited visibility available to users looking for information on their system performance and service resources including performance and latency.

Isovalent Enterprise relies on project Hubble integration for a majority of its observability requirements. Cilium states that some other integrations are available with tracing systems like Jaeger and OpenTelemetry, but these are in beta with limited documentation available.

Hashicorp Consul: Observability features available to Hashicorp Consul users are available through Hashicorp Cloud Platform (HCP). For server metrics, Hashicorp Consul servers send a variety of agent telemetry metrics that can provide observability into network operations directly to Hashicorp Consul Central.

For service meshes, Envoy proxies emit metrics for connections and requests to services as well as metrics for their own resource usage. This is collected with the Hashicorp Consul telemetry collector. Envoy proxies cannot push metrics directly to HCP Consul Central, therefore Hashicorp Consul telemetry collector receives metrics from the proxies and then exports the data to HCP. Users can analyze latency of their systems with the data extracted via HCP observability dashboard, featuring information including leader status, heartbeats (time tracking of services), and leader transactions plus more.

Anthos Service Mesh: Has built-in observability functions to provide a full view of performance and health of services across environments. Telemetry data is available and Anthos Service Mesh for GKE Enterprise relies on sidecar proxies to collate the proxy telemetry data.

Anthos Service Mesh provides several preconfigured service dashboards in the Google Cloud console so users don't have to manually set up dashboards and can immediately get an overview of services in your mesh, including latency results, traffic, and errors. Anthos Service Mesh GKE Enterprise users, for an additional cost, can implement Cloud Trace to collect latency data from applications in near real-time and follow sample requests through systems end to end.

Red Hat OSSM: Red Hat provides its own tracing platform to gather insights into the service architecture, including collecting telemetry materials on distributed transactions to help users optimize performance and latency and identify root cause analysis.

Red Hat OSSM is built on Istio, which injects a sidecar proxy on the data path, and latency is an important consideration. Istio adds an authentication filter, a telemetry filter, and a metadata exchange filter to the proxy. Every additional filter adds to the path length inside the proxy and affects latency.

5.5.2. Vendor Benchmarking Results

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◔
- Red Hat OSSM: ◔

Gloo Mesh: Gloo Mesh is considered performant when Gloo resources that the user creates are translated into Envoy and Istio resources, and are applied in the user's environment in a reasonable amount of time. Internal scalability tests show that the Gloo Mesh scalability threshold is reached more quickly if the snapshot size is greater than 20 MB.

Additional benchmarking thresholds include: translation times are considered too high when greater than 60 seconds, user experience time is too high when configuration is greater than 120 seconds, and downtime of the Gloo management server.

Solo runs internal scalability tests with every release of Gloo Mesh to verify translations and user experience times and measure performance improvements between releases. These are accessible to Gloo Mesh customers via their support platform.

Buoyant Enterprise for Linkerd: Markets itself as a solution that consumes a minimal amount of memory and CPU at the data plane layer. The metrics published by Buoyant for Linkerd are that a single linkerd-proxy instance can proxy many thousands of requests per second in under 10mb of memory and a quarter of a core, all with a p99 tail latency of under 1ms.

Isovalent Enterprise for Cilium: Provides performance benchmarking across a variety of scenarios based on the follow metrics:

1. **Throughput:** Maximum transfer rate via a single TCP connection and the total transfer rate of 32 accumulated connections.

2. **Request/Response Rate:** The number of request/response messages per second that can be transmitted over a single TCP connection and over 32 parallel TCP connections.

3. **Connections Rate:** The number of connections per second that can be established in sequence with a single request/response payload message transmitted for each new connection. A single process and 32 parallel processes are tested.

Hashicorp Consul: To test the control plane scalability of Hashicorp Consul, they have constructed a large-scale benchmark. It is based on 5 Hashicorp Consul servers of the Service Mesh Control Plane that were able to deliver control plane updates to 172,000+ service instances (containers) in under 1 second in a Hashicorp Consul cluster of 10,000 virtual machines/nodes and Hashicorp Consul clients.

Anthos Service Mesh: The company recommends setting up a specific test environment for Anthos Service Mesh. It also provides observability into health and performance of services by obtaining telemetry data via sidecar proxies injected as separate containers into pods. Anthos Service Mesh integrates with other Google solutions including Cloud Monitoring and Cloud Logging through the Google Cloud project. Performance benchmarking of the service mesh and workloads can be based on service-level metrics including latency, traffic, and errors rates.

Red Hat OSSM: For control plane performance, benchmarks can be measured by comparing the rate of deployment and configuration changes and proxies connected to Istiod. Similarly, data plane performance can be measured on factors including number of connections, request and response sizes, CPU cores and number of proxy filters, and worker threads. Latency, throughput, and the proxies' CPU and memory consumption are measured as a function of these factors.

5.5.3. Resource Utilization

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◗
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh defines its production small environments with 1 management cluster, 2 workload clusters, and less than 1000 services. This setup uses the following resources: 8 vCPU and 32 GB of memory for management cluster instances, Gloo management service to utilize 4 vCPUs and 16GB of memory for resource requests, and 8 vCPU and 32GB of memory for resource limits.

The Gloo agent will use 1 vCPU and 2 GB of memory for resource requests and 2 vCPU and 4 GB of memory for resource requests, and Redis in the management plane will consume 2 vCPU and 8GB of memory.

Buoyant Enterprise for Linkerd: Linkerd's CPU usage was 212ms of control plane CPU time. For the data plane the maximum memory consumed by a Linkerd proxy was, on average, 26.3mb and the maximum proxy CPU time recorded was 36ms.

Isovalent Enterprise for Cilium: Conducted a testing scenario to run Cilium in CRD mode with over 200 nodes. It consisted of 3 controller nodes utilizing 32 vCPU, 120GB of memory and 100 worker nodes consuming 2 vCPU, 4GB of memory. It also consisted of 1 metric node utilizing 32 vCPU and 120GB of memory. Cilium can scale to this range, but requires more testing to verify some behaviors around connectivity during control plane upgrades.

Hashicorp Consul: The resource requirements for Hashicorp Consul services in production environments are 8-16 CPU and 32-64 GB RAM of memory depending on size of services. Hashicorp Consul recommends starting from the following instances (or similar) and scaling up as needed and limiting deployments to a maximum of 5,000 Hashicorp Consul client agents per Hashicorp Consul datacenter.

Anthos Service Mesh: Resource requirements of Anthos Service Mesh requests standard GKE clusters with machine type that has at least 4 vCPUs. The minimum number of nodes depends on your machine type. Anthos Service Mesh requires at least 8 vCPUs. If the machine type has 4 vCPUs, your cluster must have at least 2 nodes.Google Fleet and Cloud Monitoring can then abstract performance metrics from standard deployments to identify problems based on total CPU, memory and utilization over a specific period of time across disks, clusters, name spaces and fleet groups.

Red Hat OSSM: Red Hat conducts tests across different performance outcomes including the control and data plane. General load tests show that running Istio 1.12.3 on Red Hat OSSM requires that the Envoy proxy uses 0.35 vCPU and 40 MB memory per 1000 requests per second through the proxy and Istiod consumes 1 vCPU and 1.5 GB of memory.

5.5.4. Health Checks

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ●
- Isovalent Enterprise for Cilium: ●
- Hashicorp Consul: ●
- Anthos Service Mesh: ●
- Red Hat OSSM: ●

Gloo Mesh: Gloo Mesh users can use the ingress gateway to periodically check the health of an upstream service in your cluster. If an upstream service is unavailable, the service is removed from the load balancing pool until health is re-established. Active health check policies are applied at the Destination level.

Buoyant Enterprise for Linkerd: Provides a check command that will perform a series of checks to validate that the Linkerd CLI and control plane are configured correctly. If the command encounters a failure it will print additional information about the failure and exit with a non-zero exit code.

Isovalent Enterprise for Cilium: Provides the cilium-health tool that gives visibility and instant snapshot into the overall health of the cluster's networking connectivity and helps with troubleshooting connectivity issues. Insights provided can help troubleshoot network issues and identify misconfigurations of node settings.

Hashicorp Consul: Hashicorp Consul health checks are configurations that verifies the health of a service or node. Health checks configurations are nested in the service block. Users can create multiple types of different checks including: script, HTTP, TCP, UDP, Time-to-live, Docker, gRPC, and more. For Kubernetes environments, users can sync service health information with Kubernetes health checks.

Anthos Service Mesh: Health checks are available through the Google Cloud load balancer backends, Traffic Director backends, and application-based autohealing for managed instance groups. Google Cloud records the success or failure of each probe.

For GKE Enterprise users, Anthos Service Mesh has load balancing for health checks as well. However it does not integrate with the Google Cloud load balancer forcing users to pick between the two different health check feature sets avaliable within the Google ecosystem.

Red Hat OSSM: Health checks are available within OpenShift platform as part of the container runtime. Multiple probes can be specified for the pod and container including readiness, liveness health check, and startup probe. Users can utilize two views - Developer and Topology. Developer view ensures your application is healthy within the container with the probe metrics and the Topology view allows you to edit the health checks added to applications in containers.

5.6. Observability and Operations

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Lifecycle Management	●	●	◔	◕	◗	◕
Insights & Analytics	◕	◔	◗	◗	◔	◕
Metrics and Logs	●	◕	◗	◕	◗	◕

5.6.1. Lifecycle Management

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ●
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh incorporates Gloo's Istio Lifecycle Manager to install and manage Solo distributions of Istio service meshes in clusters for you. Using the Gloo-managed service meshes, they no longer need to manually install and manage the `istiod` control plane. Instead, users provide the Istio configuration in a Gloo custom resource and Gloo translates this configuration into managed `istiod` control plane, and gateways in each workload cluster.

Any changes made after the deployment are automatically prograted to Istio installations and, depending on the type of change, users can apply updates to installs with a revisioned canary upgrade or in-place upgrade.

Buoyant Enterprise for Linkerd: Provides lifecycle automation functionality, including fully automated installs, upgrades, and rollbacks, of both control plane and data plane. It is fully GitOps compatible and integrates with open source tools like Argo and Flux.

For customers of Buoyant Enterprise for Linkerd, the lifecycle automation feature eliminates the toil of keeping things up to date for control plane components and for proxies across every cluster. Lifecycle automation is implemented by a Kubernetes operator that runs on each cluster and reads configuration from a set of custom resources. These resources allow you to customize not just the behavior of the operator itself, but also the version of Linkerd installed on the clusters.

Isovalent Enterprise for Cilium: Lifecycle management is disjointed, with multiple manual processes and restrictions placed on users dependent on the version of Cilium being used in Isovalent Enterprise. Upgrades and rollbacks are tested in specific circumstances between consecutive minor releases only and can only be done at one minor release at a time.

Isovalent Enterprise users are advised to run pre-flight checks with any upgrades with Kubernetes, any upgrades to Isovalent Enterprise users must ensure all components are running the same version.

Hashicorp Consul: Hashicorp Consul Enterprise enables the capability of automatically upgrading a cluster of Hashicorp Consul servers to a new version as updated server nodes join

the cluster. This automated upgrade will spawn a process which monitors the amount of voting members currently in a cluster. Demotion of legacy server nodes will not occur until the voting members on the new version match. Once this demotion occurs, the previous versioned servers can be removed from the cluster safely.

Enterprise Hashicorp Consul also incorporates the Autopilot feature, which allows for automatic, operator-friendly management of Hashicorp Consul servers. It includes cleanup of dead servers, monitoring the state of the Raft cluster, and stable server introduction.

Anthos Service Mesh: For all GKE-based clusters, including on-premises and public clouds, GKE Enterprise provides tools for cluster management and lifecycle (create, update, delete, and upgrade), including command line utilities. GKE Enterprise users can also leverage GKE Autopilot, but only with managed Anthos Service Mesh version 1.21.3 and above.

Red Hat OSSM: To keep your Red Hat OSSMh patched with the latest security fixes, bug fixes, and software updates, you must keep your Operators updated. However, Red Hat OpenShift Service Mesh Operator supports multiple versions of the Service Mesh control plane, therefore updating the Red Hat OpenShift Service Mesh Operator does not update the specific version of your deployed control plane.

For clusters, Red Hat OpenShift Container Platform has the Operator Lifecycle Manager (OLM), which controls the installation, upgrade, and role-based access control (RBAC) of Operators in a cluster. The OLM runs by default in OpenShift Container Platform and queries for available Operators as well as upgrades for installed Operators.

5.6.2. Insights and Analytics

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◗
- Anthos Service Mesh: ◔
- Red Hat OSSM: ◕

Gloo Mesh: Comes with the Gloo UI to help digest and evaluate the metrics identifying the health and efficiency of your service mesh. Gloo Mesh comes with an insights engine that automatically analyzes your Istio and Cilium setups for health issues and provides a checklist to address issues that may be hard to detect across environments. The engine provides data onto the Gloo UI Insights page, where users can view recommendations to harden your Istio and Cilium setups, and steps to implement them in your environment.

Users can import the operations dashboard in Grafana to monitor the health of Gloo Mesh components. The operations dashboard lets you monitor the health of your Gloo Mesh environment, such as the average translation and reconciliation time for the Gloo management server, or translation errors that occurred.

Buoyant Enterprise for Linkerd: Provides a full on-cluster metrics stack, including CLI tools and a web dashboard. Insights can be pulled from the golden metrics of Success Rate, Traffic Requests per Second, and Latencies. To access Linkerd's observability features users need to install the Viz extension, which includes recording of golden metrics, TCP-level metrics, reporting of metrics per service, generating topology graphics, and request samplings.

To visualize and consume the insights, the Linkerd dashboard provides a high level view of what is happening with your services in real time. It can be used to view "golden metrics," visualize service dependencies. and understand the health of specific service routes.

Isovalent Enterprise for Cilium: Isovalent Enterprise integrates with Hubble as its primary observability platform, which provides visibility into network flows and supports Prometheus and OpenTelementry metrics. Hubble is able to provide visibility at the node level, cluster level, or even across clusters in a multi-cluster. Cilium and Hubble metrics can be enabled independently of each other, by which insights can be derived. Integrations with Prometheus and Grafana are available to ingest and visualize the insights and analysis.

Hashicorp Consul: Hashicorp Consul observability features are available to enhance their service mesh capabilities with data including metrics, logs, and distributed traces to generate insights and analytics to improve performance debug services.

For enterprise users there is HCP Consul Central, the hosted management plane service available through HashiCorp Cloud Platform. It centralizes global management operations across all Hashicorp Consul clusters and provides global visibility and control for both HashiCorp-managed and self-managed Hashicorp Consul clusters for services in multiple cloud environments and regions, with features including an observability dashboard, access to service metrics, and cluster management.

Anthos Service Mesh: GKE enterprise users can obtain telemetry data via the sidecar proxies injected into containers of the pods in workloads. The proxies intercept all inbound and outbound HTTP traffic to the workloads and report the data to Anthos Service Mesh. The telemetry data is automatically uploaded to the Anthos Service Mesh pages in the Google Cloud console.

Anthos Service Mesh provides several preconfigured service dashboards in the Google Cloud console so users don't have to manually set up dashboards and charts. This telemetry enables

operators to observe service behavior to help troubleshoot, maintain, and optimize their applications. Operators can also implement service level objectives (SLOs) in Anthos Service Mesh for services and monitor or alert your services based on determined SLOs.

Red Hat OSSM: Red Hat OpenShift Service Mesh provides multiple consoles to view service mesh data including Kiali, Jaeger, Grafana, and Prometheus. The Kiali console is utilized to view the applications topology, general health, and overall metrics. The Kiali integration enables users to view insights of the mesh components across different levels, including the application service and workloads. Dashboards and graph views of namespaces are available from Kiali. Users can utilize them to help visualize the analysis. There are four graphs to choose from: application, service, versioned application, or workload.

5.6.3. Metrics and Logs

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◕
- Anthos Service Mesh: ◗
- Red Hat OSSM: ◕

Gloo Mesh: Gloo Mesh comes with a telemetry pipeline automatically set up, collecting over 20 unique types records metrics for Istio proxy, Istiod, and Cilium. These metrics are collected by the pipeline with Prometheus and can be ingested by the Gloo observability tools and insights displayed via the Gloo UI Graph.

The Gloo management server also incorporates Prometheus, which automatically scrapes its metrics. These logs can be viewed by accessing the Prometheus expression browser or the Gloo Dashboard in Grafana. Users can also configure the Gloo telemetry pipeline to scale the Gloo management service metrics and make the metrics available to the Gloo telemetry gateway.

Buoyant Enterprise for Linkerd: Provides an extensive set of metrics for all traffic that passes through its data plane. The primary metrics monitored are the golden metrics of success rate, traffic requests per second, and latencies. These metrics are collected at the proxy level and reported on the proxy's metrics endpoint.

A separate tool is used to collect metrics from all proxies and aggregate them together for consumption like the linkerd-viz extension. This extension creates an on-cluster Prometheus instance for a short time window of approximately 6 hours and does not persist data across restarts. Buoyant Enterprise for Linkerd is not designed as a long-term historical metrics store and recommends exporting these metrics into an external metrics store.

Isovalent Enterprise for Cilium: Cilium metrics provide insights into the state of Cilium itself, namely of the agent, envoy, and operator processes whereas Hubble provides metrics on the network behavior of Cilium-managed Kubernetes pods with respect to connectivity and security. Cilium integrates with Prometheus and Grafana and can automatically scale both Cilium and Hubble metrics, which can be visualized with the Cilium Dashboard.

Hashicorp Consul: Hashicorp Consul proxy metrics also give you detailed health and performance information about your service mesh applications, including upstream/downstream network traffic metrics, ingress/egress request details, error rates, and additional performance information of your applications. Users can enable proxy metrics in Hashicorp Consul and do not need to configure or instrument your applications in the service mesh to leverage proxy metrics. Hashicorp Consul can also emit access logs to record application connections and requests that pass through proxies in a service mesh, including sidecar proxies and gateways.

Integrations with dashboards from Grafana can help visualize the information from the insights collected or users can utilize Hashicorp Consul's built-in UI, which includes a topology visualization to show a service's immediate connectivity performance at a glance.

Anthos Service Mesh: Anthos Service Mesh creates audit logs as part of Google Cloud recording the administrative and access activities within Google Cloud resources. Types of metrics and logs collected available for Anthos Service Mesh include: administrator activity log, data access audit logs, access/envoy logs, and traffic logs.

Red Hat OSSM: By default, Red Hat OSSM installs the Service Mesh control plane (SMCP) with a dedicated instance of Prometheus for collecting metrics from a mesh. Users can use the Kiali console to view metrics and logs for both the application and proxy and control how often the data is refreshed.

Red Hat OSSM users can view their inbound and outbound metrics across their application workloads with Kiali console and the metrics can be displayed in predefined dashboards. The application and workload detail views show request and response metrics such as volume, duration, size, or TCP traffic. The service detail view shows request and response metrics for inbound traffic only.

For logs users can view the 'Workload Detail' page in the Kiali console, which displays a unified logs view that displays both application and proxy logs.

5.7. Enterprise Support

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Product Documentation	●	◔	◗	●	◗	●
Enterprise Support	◕	◕	◕	●	●	●

5.7.1. Product Documentation

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ●
- Anthos Service Mesh: ◗
- Red Hat OSSM: ●

Gloo Mesh: Gloo Mesh has clear documentation that includes instructional examples and guides. Users may face minor confusion from the two variations of Gloo Mesh (Gloo Mesh Core and Gloo Mesh Enterprise) documentation available. Searchability is reliable and indexing of topics is logical and simple.

Good additional resources including Solo Academy and other learning resources are also available online. Slack offers a direct communication channel if community users have questions.

Buoyant Enterprise for Linkerd: Has limited use case and configuration examples. Documentation is primarily at the fundamental level, and difficult to search for key terms, and indexing of topics is limited, creating a hard to navigate documentation experience. Additional resources including self-paced courses and service mesh academy are available.

Isovalent Enterprise for Cilium: Isovalent Enterprise has limited use case and configuration examples due to lacking product areas. Documentation is primarily written at the fundamental

level and lacks depth in certain areas, but has good integration with product release cycles andgood indexing and key terms searchability. Isovalent Enterprise is accessible via support portal only.

Hashicorp Consul: Hashicorp Consul has clear online documentation between open source and commercial portfolios. There's an integrated documentation experience between community and enterprise versions of Hashicorp Consul that provides simple context and easy to follow examples and instructions across the entire portfolio suite. Documentation is grouped logically and is easily searchable via key terms. Additional technical examples are available through tutorials and blog and community resources.

Anthos Service Mesh: There is complex product differentiation between Anthos Service Mesh for GKE Enterprise and Traffic Director, and the problem is compounded by difficult to search and navigate documentation with key terms. Some components of documentation lack technical depth; there are some basic tutorials and examples provided.

Red Hat OSSM: Red Hat has clear online documentation that guides users between the open source integration and enterprise add-ons of Red Hat OSSM with Red Hat Openshift Container Platform. However, indexing of pages needs clarity and it can be hard to search for common key terms. There are good examples and deeper technical examples from blogs linked from documentation, as well as good integration with product release cycles.

5.7.2. Service and Support

- Gloo Mesh: ◕
- Buoyant Enterprise for Linkerd: ◕
- Isovalent Enterprise for Cilium: ◕
- Hashicorp Consul: ●
- Anthos Service Mesh: ●
- Red Hat OSSM: ●

Gloo Mesh: Gloo Mesh's published SLAs provide assurances that issues are responded to in a timely manner. There is expert and community support available on Slack. It does not offer 24x7 worldwide coverage for enterprise support.

Buoyant Enterprise for Linkerd: Provides 24x7 on-call support, training, architectural review, and production runbooks for enterprise edition customers. Community support is available from online forums with mediocre participation.

Isovalent Enterprise for Cilium: There is access to enterprise-hardened Cilium with 24x7 SLA support available for Isovalent Enterprise. Additional expert support including architect, custom integrations, and training available. Community support is available through Slack.

Hashicorp Consul: There are three tiers of technical support available including bronze, silver, and gold, with public defined SLAs. Access to support is based on severity and differs between support levels. There's an active community board.

Anthos Service Mesh: Three tiers of technical support are available: standard, enhanced premium, and comprehensive GCP support. Target response times vary depending on support level – there are not defined SLAs. GCP support is broad and primarily around the GCP platform, specific service mesh only support is unavailable.

Red Hat OSSM: There are three tiers of technical support: self-support, standard, and premium. Access to 24x7 support is available with Premium. There are publicly available defined SLAs. Additional expert support packages are available to subscribe to. There's a rich customer-only access knowledge base.

5.8. Open Source Community

	Gloo Mesh	Buoyant Enterprise Linkerd	Isovalent Enterprise Cilium	Hashicorp Consul	Anthos Service Mesh	Red Hat OSSM
Multi vs Single Vendor Project	●	◔	◔	◔	●	◕
Release Cadence	●	○	◗	◕	●	●
Community Engagement	●	◗	◕	◔	◕	◗

5.8.1. Multi vs Single Vendor Project

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◔
- Isovalent Enterprise for Cilium: ◔
- Hashicorp Consul: ◔
- Anthos Service Mesh: ●
- Red Hat OSSM: ◕

Gloo Mesh: Built by Solo and is based off of the popular open source project Istio. The Istio project was started by teams from Google and IBM in partnership with the Envoy team from Lyft and is fully open source and based on Envoy. Solo is the top contributor to the Istio project alongside Google, IBM, Huawei, and Red Hat.

Gloo Mesh adds additional features to boost the resiliency, observability, and security of Istio service mesh across any environment and workloads.

Buoyant Enterprise for Linkerd: Linkerd is an open source service mesh built by Buoyant. It is built on its own proxy Linkerd2-proxy and Rust language. The Linkerd project has over 150 contributors with the majority of the project's contributions sponsored by Buoyant. Buoyant Enterprise adds security and governance features on top of Linkerd as part of its commercial offering.

Isovalent Enterprise for Cilium: Isovalent is the creator and main contributor to open source project Cilium, which was donated to the CNCF in 2021 and is now a graduated project. It is built on eBPF network technology. Isovalent Enterprise for Cilium adds 24x7 support to Cilium, additional training, and professional support services, on top of access to the enterprise-hardened Cilium version.

Hashicorp Consul: There are two versions of Hashicorp Consul, an open source and enterprise edition. Community users can submit pull requests or issue tickets on the GitHub repository for open source Consul, however it is primarily maintained and built by Hashicorp. Hashicorp Consul has been designed to help enterprises with multiple data centers connect their services with additional security, operational, and governance features plus support.

Anthos Service Mesh: Anthos Service Mesh for GKE Enterprise is powered by Istio. Anthos Service Mesh is deployed as a uniform layer across users' infrastructure. Service developers and operators can use its rich feature set without making changes to application code.
Traffic Director is Google Cloud's fully managed application networking platform and service mesh. Traffic Director uses the same control plane (xDS) APIs that popular open source projects such as Envoy and Istio use. It is separate from Anthos Service Mesh. Both options are designed for Google Cloud users and are an additional subscription cost to the Google Cloud platform.

Red Hat OSSM: Red Hat OpenShift Service Mesh is based on Istio and is available as part of the Red Hat OpenShift subscription. As a commercial offering it adds a layer of additional features on top of Istio for Red Hat OpenShift users including security, traffic, telemetry, and governance. Strong integration with other open source projects like Kiali, the Istio console, and Jaeger for distributed tracing. Red Hat is a contributor to the Istio project.

5.8.2. Release Cadence

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ○
- Isovalent Enterprise for Cilium: ◗
- Hashicorp Consul: ◕
- Anthos Service Mesh: ●
- Red Hat OSSM: ●

Gloo Mesh: Solo supports n-3 for Gloo Mesh Enterprise and n-4 for Solo distributions of Istio and Cilium. Gloo Mesh Enterprise releases a new minor version, n, each quarter. When the new minor version is released, the previous n-4 for Gloo Mesh Enterprise or n-5 for Solo distributions of Istio and Cilium become unsupported.

Buoyant Enterprise for Linkerd: There are two forms of releases and packages of Linkerd – stable and edge releases. As of February 2024, the Linkerd project is no longer producing open source stable release artifacts. Instead, the vendor community around Linkerd is responsible for supported, stable release artifacts. Users looking for a stable version of open source Linkerd will need to purchase Buoyant Enterprise for Linkerd.

Linkerd Edge releases are still available weekly but may contain partial features modifications and breaking changes.

Isovalent Enterprise for Cilium: New versions of Cilium are released based on completion of feature work that has been scheduled for that release. Minor releases are frequent as three stable branches of the project are maintained at a time.

Hashicorp Consul: Hashicorp Consul major releases are made approximately every 6 months with patch releases weekly. Generally available releases of Hashicorp Consul are supported for up to two years. Hashicorp Consul Enterprise offers annual Long Term Support (LTS) releases starting with version v1.15.

Anthos Service Mesh: Anthos Service Mesh release channels are tied to Google Kubernetes Engine (GKE) release channels. Google automatically manages the version and upgrade cadence for each release channel. Google aims to do stable releases every 2 to 3 months for GKE.

Red Hat OSSM: Red Hat OSSM provides consistent releases in line with updates on open source projects upgrades. Upgrades of versions are managed by Red Hat OpenShift Service Mesh Operator. Timing around releases is unidentifiable due to closed source code, however Red Hat aims to forecast releases at a 4 month cadence in its life cycle policy.

5.8.3. Community Engagement

- Gloo Mesh: ●
- Buoyant Enterprise for Linkerd: ◗
- Isovalent Enterprise for Cilium: ◕
- Hashicorp Consul: ◔
- Anthos Service Mesh: ◕
- Red Hat OSSM: ◗

Gloo Mesh: Solo is active across the open source and cloud native community. They are contributors to multiple open source projects including Istio. In 2024, Solo led the development of the latest feature release in Istio – ambient mode – and remains an active member in the cloud native computing foundation (CNCF) community.

The Solo portfolio, including Gloo Mesh, is open source and cloud native based, built on open source technologies including Istio, Envoy, Kubernetes, Backstage, and Cilium.

Buoyant Enterprise for Linkerd: Buoyant created Linkerd and was an incubating-level CNCF project in 2018 with the project graduating in 2021. Linkerd is an open source project with the source code available in GitHub, however in 2024 they released a new model for stable releases which will only be available to Buoyant Enterprise for Linkerd customers. Buoyant has multiple contributors to Linkerd who are sponsored by Buoyant.

Isovalent Enterprise for Cilium: Cilium is open source and became an incubating project with the CNCF in October 2021 and graduated in October 2023. The project has seven companies contributing to the project with over 75% of code contributors of the project from Isovalent. In April 2024, Cisco's acquisition of Isovalent was completed and the downstream impacts of the acquisition on project Cilium remains unknown.

Hashicorp Consul: Hashicorp has built a strong community through its product suite that offers both an open source and enterprise offering. Eight products in their portfolio including Hashicorp Consul have made their source code broadly available and open.

In 2020, Hashicorp joined the CNCF to further integrations with CNCF projects and work closer with the cloud native community. Hashicorp Consul has added integrations to deploy service mesh natively onto Kubernetes clusters via a Helm chart and also provides a native integration into Envoy to be used as a sidecar proxy for a service mesh, a traditional API gateway or ingress layer, or mesh gateway to enable multi-data center networking.

Hashicorp Consul is available free via an open source license for practitioners to utilize. Hashicorp Consul also provides a free-trial of their managed service offering.

Anthos Service Mesh: Google is a maintainer of the Istio project, originally leading the project with IBM in partnership with the Envoy team at Lyft, extending their open source legacy as founders of Kubernetes.

Anthos Service Mesh was originally designed to be based on Istio for users to manage service mesh on-premise or on Google Cloud with full Google support. Today Anthos Service Mesh gives users GKE Enterprise tested and supported distribution of Istio, letting them create and deploy a service mesh on GKE on Google Cloud and other platforms with full Google support.

In April 2024 at Google Next, Google introduced Traffic Director – Google Cloud's fully managed application networking platform and service mesh alongside its existing Anthos Service Mesh offering.

Red Hat OSSM: Red Hat OpenShift Service Mesh is a commercial solution based on the popular open source project Istio. Red Hat remains a consistent maintainer on the Istio project and regularly provides patch updates to its service mesh offering to match upgrades from Istio. Red Hat's OpenShift portfolio is dependent on open source cloud native technologies and remains a member of the CNCF contributing to multiple projects including Istio and Knative.

Additional add-ons to Red Hat OSSM are also open source driven with plugins like OpenShift Service Mesh Console being built on open source project Kiali and Red Hat distributed tracing components built with Jaeger. Red Hat also integrates with multiple popular third party open source tools including OpenTelemetry for telemetry data processing, Prometheus for metrics, and Grafana for observability dashboards.

6. About the Author

Established in 2017 in Cambridge, MA, Solo.io is a prominent leader in cloud-native application networking. Our expertise lies in seamlessly connecting, securing, and monitoring Kubernetes applications and APIs on a global scale. Our renowned products – Gloo Gateway and Gloo Mesh – are the preferred choice for Fortune 2000 industry leaders and cloud-native innovators alike, representing market leadership across industries and geographies.

In the rapidly evolving cloud-native landscape, Solo.io leads with solutions that redefine application development's speed and agility. We are dedicated to streamlining and securing cloud-native services and empowering next-generation digital experiences, particularly in generative AI (genAI).

With Solo.io, you're equipped to seamlessly and securely expand in a multi-cloud world, ensuring your applications not only keep pace with technological advancements, but lead the way in innovation.

7. About This Guide

The guide was created by a team at Solo.io who reviewed publicly available documentation, vendor information, blogs, videos, and technology details. This comparative analysis dives into the strengths and weaknesses of each platform using selected categories and features to offer valuable insights to potential buyers seeking a detailed understanding of the service mesh alternatives.

8. Legal Disclaimer

This Buyer's Guide ("Guide") has been created to provide enterprise buyers with valuable information when considering the purchase of products or services from various vendors. Before utilizing the information contained within this Guide, it is important to read and understand the following disclaimer:

1. General Information: This Guide is intended for general informational purposes only. It does not constitute legal, financial, or professional advice. Users are advised to seek professional advice and conduct their own research before making any purchasing decisions.

2. Accuracy of Information: While we have made every effort to ensure the accuracy and reliability of the information provided in this Guide, we do not guarantee its completeness, accuracy, or timeliness. Information may become outdated or inaccurate over time, and providers' offerings may change without notice.

3. Third-Party Providers: This Guide may contain information about products or services offered by third-party providers. We do not endorse or recommend any specific provider, product, or service mentioned in this Guide. Users are encouraged to independently verify the information provided and make their own assessments regarding suitability and compatibility with their needs.

4. Product and Service Comparisons: The comparisons made in this Guide are based on the information available at the time of publication and may not reflect the current market conditions or the most recent updates from providers. Users are responsible for verifying the accuracy of any product or service comparisons and should consider their individual preferences and requirements.

5. No Warranty or Guarantee: We make no warranties or representations regarding the products or services mentioned in this Guide, including but not limited to their quality, performance, suitability, or fitness for a particular purpose. Users are advised to review the terms and conditions, warranties, and guarantees provided by the relevant providers before making any purchases.

6. Legal Compliance: It is the responsibility of users to ensure that any products or services they purchase comply with local, state, and federal laws and regulations. This Guide does not constitute legal advice or a guarantee of legal compliance.

7. No Liability: We shall not be liable for any loss, damage, or inconvenience arising from the use of the information provided in this Guide. Users use this Guide at their own risk and agree to release us from any liability or claims.

8. Updates and Revisions: We may update or revise this Guide at any time without prior notice. Users should check for the most recent version of this Guide before relying on its information.

By accessing and using this Guide, you acknowledge that you have read and understood this disclaimer and agree to its terms and conditions. If you do not agree with any part of this disclaimer, please refrain from using this Guide.

This disclaimer is subject to change without notice. Please check for updates periodically.

9. Glossary

ACL (Access Control List):
A list of permissions attached to an object that specifies which users or system processes are granted access to resources, and what operations they are allowed to perform.

Adaptive Request Concurrency:
Capability to adjust the number of concurrent requests based on workload and resource availability.

Administrative Partitions:
Mechanism for managing resources across different teams or tenants within a service mesh environment.

Ambient Mode:
A deployment mode in Istio that removes the requirement of running sidecars alongside applications in a service mesh.

API (Application Programming Interface):
A set of rules and protocols that allows different software applications to communicate with each other.

API Gateway:
A server that acts as an API front-end, receiving API requests, enforcing throttling and security policies, passing requests to the back-end service, and then passing the response back to the requester.

Attribute-based Authentication:
A method of authentication where access to resources is granted based on the attributes or characteristics of the entity requesting access.

Authentication:
The process of verifying the identity of a user, system, or application to ensure that they have the necessary permissions to access an API.

Authorization:
The process of granting or denying access to specific resources or actions based on the authenticated user's permissions.

Blast Radius:
The potential impact of a failure or outage within a system, measured by the number of affected components or services.

CLI (Command Line Interface):
A text-based interface for interacting with computer systems and executing commands.

CRD (Custom Resource Definition):
In Kubernetes, an extension mechanism that allows users to define custom resources and controllers.

Data Plane:
The component responsible for handling the actual network traffic between services in a service mesh.

DevOps:
A set of practices that combines software development (Dev) and IT operations (Ops) to improve collaboration and productivity by automating infrastructure, workflows, and continuously measuring application performance.

eBPF (Extended Berkeley Packet Filter):
A technology enabling programmable packet processing within the Linux kernel, used by Cilium to optimize network communication.

Endpoint:
A specific URL or URI where an API can be accessed.

Fault Injection:
Intentional introduction of faults or errors into a system to test its resilience and error handling capabilities.

Federated Deployment:
Model allowing multiple service meshes to behave as a single entity while maintaining separate administrative domains.

Fleet:
Logical grouping of Google Cloud project resources for management purposes within Anthos Service Mesh.

GraphQL:
A query language for APIs and a runtime for executing those queries with existing data. It allows clients to request only the data they need.

Hardened Recovery Processes:
Procedures and measures designed to ensure data integrity and system stability during recovery from failures or outages.

Helm Charts:
Packages used to define, install, and manage Kubernetes applications.

Horizontal Replica Scaling:
Increasing the number of identical instances of a service or component to distribute workload and improve resilience.

Identity-Aware Proxy (IAP):
A Google Cloud service that provides secure access to applications, allowing administrators to control and manage access based on user identity and context.

Istiod:
The control plane component of Istio, responsible for managing configuration, policy enforcement, and telemetry across the service mesh.

Istio Proxy Injection:
The process of automatically injecting sidecar proxies into Kubernetes pods to enable them to participate in a service mesh.

JWT (JSON Web Token):
A compact, URL-safe means of representing claims to be transferred between two parties, commonly used for authentication and authorization.

Latency Analysis:
Evaluates the latency of requests and traffic within the service mesh.

Layer 7 Load Balancer:
A load balancer operating at the application layer of the OSI model, capable of distributing traffic based on various criteria such as URL paths or HTTP headers.

Middleware:
Software that acts as a bridge between different applications, allowing them to communicate or share data.

Microservices:
An architectural style that structures an application as a collection of small, independent services that communicate through APIs.

mTLS (Mutual Transport Layer Security):
A security protocol that ensures encrypted communication between client and server, with both parties verifying each other's identities using certificates.

Multi-tenancy:
The capability of a system to serve multiple tenants (users or applications) while maintaining isolation and security.

Namespaces:
Logical partitions within a Kubernetes cluster, used to organize and isolate resources, including pods, services, and storage volumes.

NetworkPolicies:
Kubernetes resources used to define rules for inbound and outbound traffic to pods, providing fine-grained control over network communication within a cluster.

Observability:
The ability to measure the internal state of a system and infer its behavior based on external outputs.

OAuth (Open Authorization):
An open standard for access delegation commonly used for enabling secure authorization to APIs.

OPA (Open Policy Agent):
An open-source policy engine that enables fine-grained, context-aware access control policies to be defined and enforced across various systems and services.

Outlier Detection:
Process of identifying and isolating instances or services exhibiting abnormal behavior within a service mesh.

Pod:
The smallest deployable unit in Kubernetes, representing a single instance of a running process in a cluster.

Protocol Detection:
Ability to identify and handle different network protocols within a service mesh.

Proxy:
An intermediary server that forwards requests from clients to other servers, often used in API gateways for routing and load balancing.

RBAC (Role-Based Access Control):
A security paradigm that restricts system access to authorized users based on their roles and permissions.

Rate Limiting:
Restricting the number of API requests a user or client can make within a specified time period.

REST (Representational State Transfer):
A set of architectural principles for designing networked applications, often used for building APIs.

Service Mesh:
A dedicated infrastructure layer for handling service-to-service communication, providing features like load balancing, service discovery, and security.

Sidecar Proxy:
A helper component deployed alongside a main application container, responsible for handling communication, security, and other cross-cutting concerns.

SPIFFE (Secure Production Identity Framework For Everyone):
A set of standards for securely identifying and authenticating communication between application services, often used in service mesh environments.

TCP Proxying:
Mechanism for intercepting and forwarding TCP traffic between services within a service mesh.

WebAssembly Filter:
A filter implemented using WebAssembly that can be applied within a service mesh to modify or intercept network traffic.

Webhook:
An HTTP callback that allows third-party applications to receive real-time information from another application when a specific event occurs.

WebSockets:
A communication protocol that provides full-duplex communication channels over a single, long-lived connection.

Zero Trust Networking:
A security model that assumes no trust between entities within a network, requiring verification of every request for access regardless of the source.

solo.io

The Gateway to AI Innovation