GIGAOM

Image credit: nevarpp

Ivan McPhee, Chris Grundemann
Aug 20, 2021

GigaOm Radar Report for Evaluating Service Mesh v 1.0

GigaOm Radar Report for Evaluating Service Mesh

Table of Contents

1. Summary

Historically, developers independently implemented error handling, observability, and security within each application or microservice to ensure the success of inbound and outbound communication requests. However, as different teams repeated the process and coded similar functionality into each application (often using different programming languages), complexity, fragmentation, and security vulnerabilities were introduced into the environment.

A service mesh addresses this problem by "outsourcing" the management of service-to-service communication requests to an out-of-process application. Typically implemented alongside the workload as a "sidecar" proxy, a service mesh simplifies and streamlines runtime operations. Comprising a "data plane" of interconnected network proxies and a "control plane" for configuring the proxies and collecting metrics, it provides a shared infrastructure layer to manage intra-service runtime communications within a distributed, microservice-based software architecture.

Application agnostic and fully portable, the service mesh can be adopted by an organization to support any service written in any language or framework. Adding uniform capabilities across the environment, a service mesh provides authentication, authorization, discovery, encryption, load balancing, logging, observability, routing, and tracing.

While implementing a service mesh has zero impact on application code (other than "desired changes" such as the removal of redundant functionality handled by the mesh, propagating mesh headers to enable tracing, or other changes to maximize the benefits of the mesh), it does affect operational procedures and requires the familiarization of DevOps personnel with new concepts and technologies. Additionally, as an emerging technology, taking the time to choose the right service mesh for your organization is essential due to the additional complexity, latency, and resource consumption involved.

Although service mesh patterns can be applied to both monolithic and microservice-based applications, this study focuses on the latter running on various platforms, including containers/ Kubernetes and virtual machines (VMs). Also known as K8s, Kubernetes is an open source orchestration platform automating the deployment, management, and scaling of containers.

This report provides an overview of the service mesh landscape based on the following table stakes, which are mature, stable solution features common across all service meshes:

- **Dedicated Infrastructure Layer:** Delivering fast, reliable, and secure service-to-service communications, a service mesh is a dedicated infrastructure layer fully integrated within the distributed application to control the delivery of service requests. The infrastructure layer provides several functions, including service discovery, authentication and authorization, health checks, failure recovery, load balancing, and observability via the "data plane."
- **Sidecar Implementation:** Like a sidecar attached to a motorcycle, a sidecar implementation

provides third-party functionality alongside the actual workload within the container. A service proxy—such as Envoy—is attached to a workload during deployment to manage service-to-service communications within a service mesh. All management capabilities required by the workload (monitoring, control, and security) are implemented without changing a single line of application code.

- **Control Plane Configuration:** Comprising a set of APIs and tools used to control proxy behavior across the mesh, the control plane automatically configures data plane service proxies. Transforming a collection of isolated, stateless sidecar proxies into a distributed system, the control plane implements policies across all data planes running within the mesh.
- **Control Plane Telemetry:** In addition to configuring and managing proxies used to route traffic and enforce policies, the control plane collects telemetry data for each request. The detailed statistics, logging, and distributed tracing data collected provide observability into service behavior for troubleshooting, maintenance, and service optimization.

With many different service meshes and options available—and the landscape evolving—choosing the best service mesh for your organization depends on your use cases, existing software stack, architectural choices, and in-house capabilities. Your internal resources and skillsets most likely will influence your decision as to whether you adopt a lightweight, developer-friendly service mesh such as Linkerd or NGINX, or an Istio-based solution.

We recommend using this report to explore the different service meshes and delivery models available on the market, while identifying those matching your business requirements, use cases, and capabilities. Then, contact the relevant open source community or commercial vendor for additional information on features, deployment models, and cost.

For additional information about how to evaluate a service mesh, please read the report, *Key Criteria for Evaluating Service Mesh: An Evaluation Guide for Technology Decision Makers*, published by GigaOm.

HOW TO READ THIS REPORT

This GigaOm report is one of a series of documents that helps IT organizations assess competing solutions in the context of well-defined features and criteria. For a fuller understanding consider reviewing the following reports:

Key Criteria report: A detailed market sector analysis that assesses the impact that key product features and criteria have on top-line solution characteristics—such as scalability, performance, and TCO—that drive purchase decisions.

GigaOm Radar report: A forward-looking analysis that plots the relative value and progression of vendor solutions along multiple axes based on strategy and execution. The Radar report includes a breakdown of each vendor's offering in the sector.

Solution Profile: An in-depth vendor analysis that builds on the framework developed in the Key Criteria and Radar reports to assess a company's engagement within a technology sector. This analysis includes forward-looking guidance around both strategy and product.

2. Market Categories and Deployment Types

For a better understanding of the market and solution positioning (**Table 1**), we categorized solutions for service mesh by the target market segment:

- **Cloud/Managed Service Providers (CSPs/MSPs):** Service providers delivering pay-per-use, on-demand services to customers over the internet or delivering application, IT infrastructure, network, and security services and support for businesses on customer premises, in the MSP's data center (hosting), or in a third-party data center.
- **Network Service Providers (NSPs):** Service providers selling network services—such as network access and bandwidth—provide access to backbone infrastructure or network access points (NAP). In this report, NSPs include data carriers, ISPs, telcos, and wireless providers.
- **Large Enterprises:** Enterprises of 1,000 or more employees with dedicated IT teams responsible for planning, building, deploying, and managing their applications, IT infrastructure, networks, and security in either an on-premises data center or a colocation facility.
- **SMBs:** Small (<100 employees) to medium-sized (100-1,000 employees) businesses with limited budgets and constrained in-house resources for planning, building, deploying, and managing their applications, IT infrastructure, networks, and security in either an on-premises data center or a colocation facility.

Also, we recognize four deployment models for service meshes in this report: Single or Multiple Cluster, Single or Multiple Network, Single or Multiple Control Plane, and Single or Multiple Mesh. Choosing the most suitable deployment model depends on your use case's availability, isolation, and performance requirements.

- **Single or Multiple Cluster:** Service meshes can be configured as either a single cluster or as a single mesh including multiple clusters. A single cluster deployment may offer simplicity, but it lacks features such as fault isolation, failover, and project isolation available in a multi-cluster deployment.
- **Single or Multiple Network:** Workload instances directly connected without using a gateway reside in a single network, enabling the uniform configuration of service consumers across the mesh. A multi-network approach allows a service mesh to span various network topologies or subnets, providing compliance, isolation, high availability, and scalability.
- **Single or Multiple Control Plane:** The control plane configures all communication between workload instances within the mesh. Deploying multiple control planes across clusters, regions, or zones provides configuration isolation, fine-grained control over configuration rollouts, and service-level isolation. Moreover, in the event one control plane becomes unavailable, the impact of the outage is limited to the workloads managed by that control plane.
- **Single or Multiple Mesh:** While a single mesh can span one or more clusters or more networks,

service names are unique within the mesh. Since namespaces are used for tenancy, a federated mesh is required to discover services and enable communication across mesh boundaries. Each mesh reveals services that can be consumed by other services, providing line of business boundaries and isolation between test and production workloads.

Table 1. Vendor Positioning

	MARKET SEGMENT				DEPLOYMENT MODELS			
	Cloud/ Managed Service Provider	Network Service Provider	Large Enterprises	SMBs	Single or Multiple Cluster	Single or Multiple Network	Single or Multiple Control Plane	Single or Multiple Mesh
CNCF - Kuma	–	–	++	++	++	++	+++	+++
CNCF - Linkerd	++	++	+++	+++	++	++	++	–
Grey Matter	++	++	+++	+	++	+++	+++	++
F5 - Aspen Mesh	–	+++	++	+	++	++	++	++
F5 - NGINX	–	–	++	–	++	++	++	++
Hashicorp	+++	++	++	++	++	++	++	++
Istio	–	–	++	++	++	++	++	++
Kong	–	–	++	++	++	++	+++	+++
Solo.io	+	++	++	++	++	++	++	++
Traefik Labs	++	++	++	++	+	+	+	+
VMware	++	++	++	–	++	++	++	++

+++ Exceptional: Outstanding focus and execution
++ Capable: Good but with room for improvement
+ Limited: Lacking in execution and use cases
– Not applicable or absent

Source: GigaOm 2021

3. Key Criteria Comparison

Following the general indications introduced with the "*Key Criteria for Evaluating Service Mesh*," **Tables 2, 3, and 4** summarize how each service mesh included in this research performs in the areas we consider differentiating and critical in this sector. The objective is to give the reader a snapshot of different solutions' technical capabilities and define the market landscape's perimeter.

Indicating each service mesh's strengths and weaknesses, the tables provide the basis upon which organizations can create a shortlist, engage with the provider, and make informed decisions on which solution to adopt for their particular needs. Attributes and capabilities will vary from one service mesh to another and should be carefully evaluated based on each organization's needs and use cases.

Table 2. Key Criteria Metrics Comparison

	KEY CRITERIA							
	Platform	Service Proxy	Configurability	Extensibility	Monitoring & Observability	Routing	Resilience	Security
CNCF - Kuma	+++	++	+++	++	++	+++	++	+++
CNCF - Linkerd	++	+++	+++	++	+++	+	++	++
Grey Matter	+++	++	+++	+++	+++	++	++	+++
F5 - Aspen Mesh	++	+++	+++	++	+++	++	++	+++
F5 - NGINX	++	++	++	+++	++	+++	++	+++
Hashicorp	++	+++	++	+++	+++	++	++	++
Istio	+++	++	+	++	++	++	++	++
Kong	+++	++	+++	++	++	+++	++	+++
Solo.io	+++	++	+++	+++	++	++	++	++
Traefik Labs	++	+++	+++	++	+++	++	+	+
VMware	+++	++	+++	+++	++	++	+++	++

+++ Exceptional: Outstanding focus and execution
++ Capable: Good but with room for improvement
+ Limited: Lacking in execution and use cases
– Not applicable or absent

Source: GigaOm 2021

Key criteria differentiate one solution from another *based on features and capabilities*, outlining the primary criteria to be considered when evaluating Service Mesh solutions.

Table 3. Evaluation Metrics Comparison

	EVALUATION METRICS								
	Features	Openness	Performance	Resource Consumption	Management	Pricing & TCO	Ecosystem	Support	Roadmap & Vision
CNCF - Kuma	+++	+++	++	+++	+++	+++	++	++	+
CNCF - Linkerd	++	+++	+++	+++	+++	+++	++	+++	+++
Grey Matter	+++	++	++	++	+++	++	+++	+++	++
F5 - Aspen Mesh	+++	+++	++	++	+++	++	++	+++	+++
F5 - NGINX	+++	++	++	+++	++	+++	+	+	++
Hashicorp	++	+++	++	++	++	++	+++	+++	++
Istio	+++	+++	+	+	+	++	+++	++	+++
Kong	+++	+++	++	++	+++	+	++	+++	++
Solo.io	+++	++	++	++	+++	++	+++	+++	+++
Traefik Labs	++	+++	++	+++	+++	+++	++	++	+++
VMware	+++	++	++	++	+++	+	++	+++	+++

+++ Exceptional: Outstanding focus and execution
++ Capable: Good but with room for improvement
+ Limited: Lacking in execution and use cases
– Not applicable or absent

Source: GigaOm 2021

Evaluation metrics differentiate one solution from another *based on the impact that the solution may have on an organization*, reflecting fundamental aspects like flexibility, ease of use, and total cost of ownership.

Table 4. Specific Service Mesh Capabilities

	OTHER SCORING						
	Service Discovery	Load Balancing	Encryption	Circuit Breaker	Distributed Tracing	Fault Injection	Advanced Routing
CNCF - Kuma	+++	++	+++	++	++	++	+++
CNCF - Linkerd	++	+++	+++	+	++	++	+
Grey Matter	+++	++	+++	++	++	++	++
F5 - Aspen Mesh	++	++	++	++	+++	++	++
F5 - NGINX	++	+++	++	+++	+++	+	++
Hashicorp	+++	++	++	++	++	++	+++
Istio	++	++	++	+++	++	+++	+++
Kong	+++	++	+++	++	++	++	+++
Solo.io	+++	+++	+++	++	++	++	+++
Traefik Labs	++	+++	++	++	++	+	++
VMware	+++	++	+++	++	++	++	+++

+++ Exceptional: Outstanding focus and execution
++ Capable: Good but with room for improvement
+ Limited: Lacking in execution and use cases
– Not applicable or absent

Source: GigaOm 2021

4. GigaOm Radar

This report synthesizes the analysis of key criteria and their impact on evaluation metrics to create the GigaOm Radar graphic in **Figure 1**. The chart is a forward-looking perspective of all the solutions in this report, based on each product's technical capabilities and feature sets.

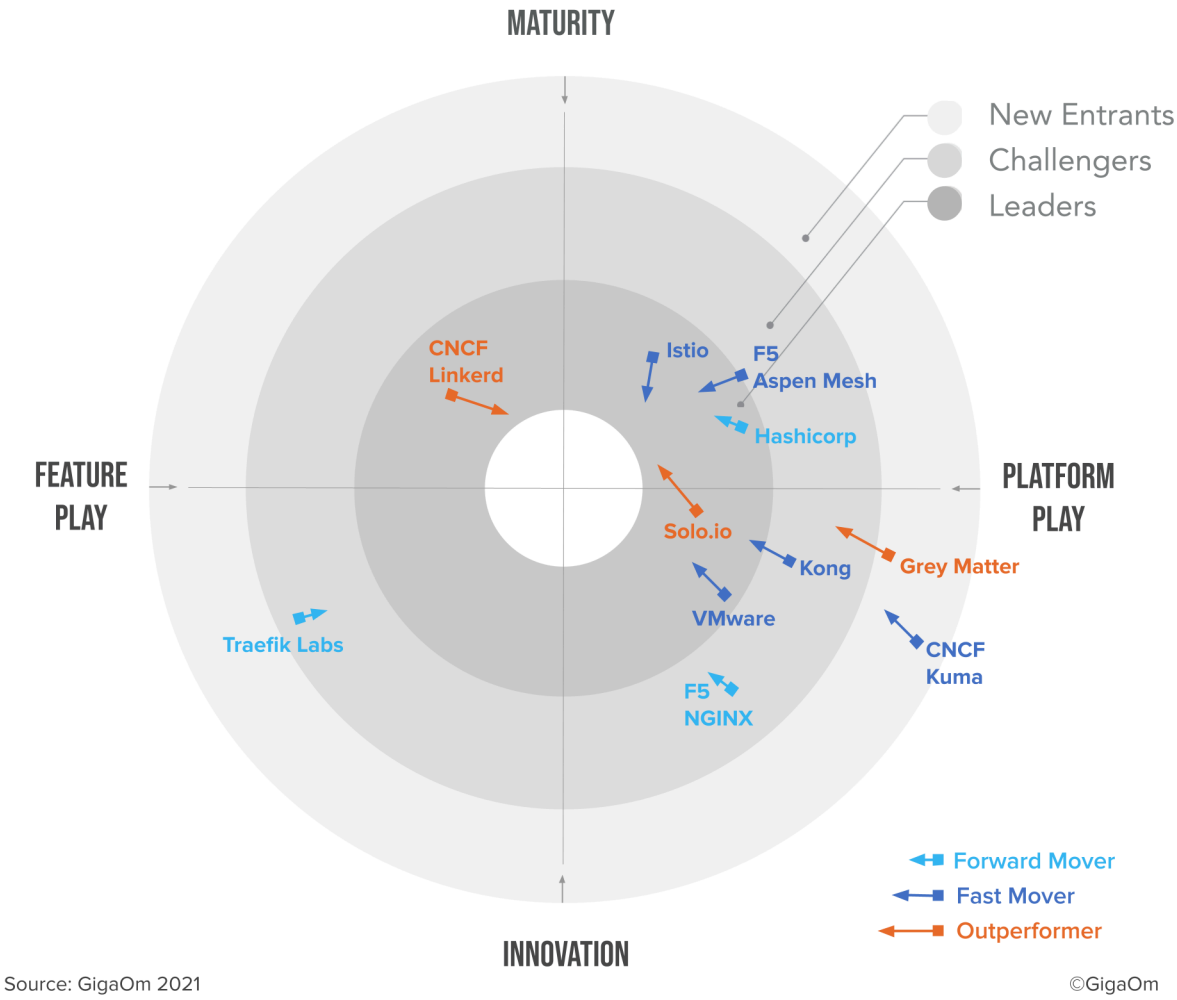

Figure 1: GigaOm Radar for Service Mesh

The GigaOm Radar plots solutions across a series of concentric rings, with those set closer to the center judged to be of higher overall value. The chart characterizes each solution on two axes—Maturity versus Innovation and Feature Play versus Platform Play—while providing an arrow that projects each solution's evolution over the coming 12 to 18 months.

As you can see in the Radar chart in **Figure 1**, there are five service meshes (Hashicorp Consul, Istio, Linkerd, Solo.io Gloo Mesh, and VMware Tanzu Service Mesh) in the leaders' ring and four challengers (Aspen Mesh, Kong, NGINX Service Mesh, and Traefik Mesh). There are also two new entrants (Grey

Matter and Kuma).

Only two service meshes, Linkerd and Traefik Mesh, focus on delivering critical features to meet the needs of specific use cases as opposed to pursuing a platform play. In addition, of all the service meshes included in this report, only Traefik Mesh lacks multi-cluster capabilities, concentrating on delivering a high-efficiency, low-resource service mesh on a per-node basis.

It's important to note that while Istio is positioned as a leader, some of the Istio-based service meshes are positioned as challengers, depending on how recently the hardened Istio distribution was released. For example, Aspen Mesh launched its Istio-based service mesh in December 2017 and it is positioned as a challenger. The Istio-based distributions positioned as leaders are Solo.io's Gloo Mesh and VMware's Tanzu Service Mesh, based on extensive investment and innovation.

One exception you'll notice is Grey Matter. Despite being released in February 2019, Grey Matter's narrow industry focus and lack of an open source solution places it as a new entrant bordering on becoming a challenger. However, we expect this to change over the next 12-18 months based on Grey Matter's efforts to engage with the open source community and its proven ability to incorporate emerging technologies into its platform. Likewise, we anticipate Kuma to follow a similar trajectory based on its built-in automation, security capabilities, and ease of use.

Of the five leaders, Istio, Linkerd, and Hashicorp's Consul each offer the use of different proxies but are established players with extensive market penetration, while Istio-based Solo.io and VMware are making significant investments to deliver both enterprise support and innovation in the service mesh arena. Two of these service meshes, Linkerd and Solo.io's Gloo Mesh, are identified as outperformers as well, together with Grey Matter, based on their rate of innovation and contribution to the service mesh market sector.

At the same time, we expect F5, Hashicorp, and VMware to focus on cementing their position within their installed bases by embedding their service mesh capabilities firmly within their respective portfolios before leveraging it to attract new customers. F5's incubation of Aspen Mesh—providing an open source complement to NGINX Service Mesh—highlights the importance of service mesh as a way to protect one's installed base. As new entrants increase adoption and innovation around enterprise-grade capabilities, we expect them to become challengers over the next 12-18 months.

Areas to keep a close eye on are Service Mesh-as-a-Service (SMaaS) and managed service mesh solutions. Hashicorp, Kong, and Solo.io all offer SMaaS, while Buoyant, the creator of Linkerd, has released Buoyant Cloud to proactively assess and validate the health of a Linkerd service backed up by specialist support from a team of Linkerd experts. Look for other vendors providing similar services over the next few months.

INSIDE THE GIGAOM RADAR

The GigaOm Radar weighs each vendor's execution, roadmap, and ability to innovate to plot solutions along two axes, each set as opposing pairs. On the Y axis, **Maturity** recognizes solution stability, strength of ecosystem, and a conservative stance, while **Innovation** highlights technical innovation and a more aggressive approach. On the X axis, **Feature Play** connotes a narrow focus on niche or cutting-edge functionality, while **Platform Play** displays a broader platform focus and commitment to a comprehensive feature set.

The closer to center a solution sits, the better its execution and value, with top performers occupying the inner Leaders circle. The centermost circle is almost always empty, reserved for highly mature and consolidated markets that lack space for further innovation.

The GigaOm Radar offers a forward-looking assessment, plotting the current and projected position of each solution over a 12- to 18-month window. Arrows indicate travel based on strategy and pace of innovation, with vendors designated as Forward Movers, Fast Movers, or Outperformers based on their rate of progression.

Note that the Radar excludes vendor market share as a metric. The focus is on forward-looking analysis that emphasizes the value of innovation and differentiation over incumbent market position.

5. Vendor Insights

Cloud Native Computing Foundation (CNCF): Kuma

Created by Kong and donated to CNCF as a sandbox project in June 2020, Kuma is an open source service mesh using Envoy as the data plane proxy and a control plane developed by Kong. Built to support both greenfield and legacy enterprise applications, Kuma offers scalable, multi-zone connectivity across multiple clusters and clouds using either Kubernetes (K8s) or virtual machines (VMs). It also supports a hybrid K8s/VM implementation offering simplified migration between environments.

Kuma At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Kuma	Envoy	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane		Control Plane	
C++	Go	Go	
Primary Use Cases			
PCI and GDPR compliance acceleration		Global traffic observability	
Zero-trust security		Intelligent traffic routing and reliability	
Pricing Model			
Freely available for download on GitHub			

Designed for the enterprise architect, Kuma ships with advanced multi-zone and multi-mesh support, enabling cross-zone communication across different clusters and clouds. With an out-of-the-box L4 and L7 policy architecture enabling discovery, observability, routing, traffic reliability, and zero-trust security, Kuma abstracts common use cases and automatically propagates service mesh policies across the infrastructure to support a multi-mesh, multitenant environment on the same control plane.

Unlike other service mesh solutions, Kuma provides native support for both Kubernetes and VMs on both control and data planes, with multi-mesh support spanning boundaries, including Kubernetes namespaces. Kuma's architecture includes control plane separation, with each zone allocated its own horizontally scalable remote control plane to minimize the possibility of one zone affecting other zones if it goes down. In addition, out-of-the-box multi-cloud, multi-cluster, and multi-zone support provide automatic policy synchronization and connectivity to support custom workload attributes for PCI (Payment Card Industry) and GDPR (General Data Protection Regulation) compliance.

Offering native service discovery across Kubernetes clusters, VMs, and bare-metal servers, Kuma supports global and remote deployment modes and native integration with API management solutions. Easy to use with no Envoy expertise required, Kuma packages Envoy with every installation, automatically injecting the sidecar proxy into workloads.

Strengths: While most service meshes prioritize Kubernetes/container-driven applications, Kuma also supports existing applications running on VMs or bare metal. Kuma is also the only open source, Envoy-based service mesh. (Note: While Istio and Istio-based service meshes may be open source, Istio's roadmap is actively managed by Google.) This makes it an attractive option for enterprises investing in an Envoy-based, open source tool free from the oversight of a cloud computing vendor.

Challenges: While claiming to address the limitations of first-generation service mesh technologies by enabling seamless management of any service on the network, Kuma is a relatively new entrant compared to other cloud-native service meshes such as Consul, Istio, and Linkerd. Kuma's success will depend largely on its adoption by the open-source community and its promotion by Kong as the underlying technology of Kong Mesh.

Cloud Native Computing Foundation (CNCF): Linkerd

The original "service mesh" released in 2016, Linkerd is an open source, CNCF-hosted security-first service mesh providing observability, reliability, and security for Kubernetes applications without adding complexity. It offers a lightweight and operationally simple approach to deploying a service mesh on any existing platform. Linkerd installs in minutes, requires zero configuration, and can be added incrementally to an application without disruption.

Linkerd At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Linkerd	Linkerd2-proxy	-	Buoyant Cloud
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	-	-	-
Languages			
Data Plane		Control Plane	
Rust		Go	
Primary Use Cases			
Minimizing complexity, optimizing performance, and providing mission-critical features for cloud-native applications using Kubernetes.			
Pricing Model			
Freely available for download on GitHub			

While other service meshes trend toward adding features supporting multiple use cases but requiring extensive configuration and tuning, Linkerd concentrates on a single use case with the goals of reducing its footprint, automating as much as possible, and minimizing the administrator's operational burden. Linkerd's simplicity also eliminates much of the complexity resulting in misconfiguration or the avoidance of security features due to the high cost of adoption.

Much of Linkerd's simplicity can be attributed to its data plane implementation using the internally developed Linkerd2-proxy—a lean, modern, scalable, and high-performance Rust-based network "micro-proxy"—rather than employing the commonly used Envoy Proxy. Since a fully deployed service mesh can entail running thousands—or tens of thousands—of micro-proxies, the impact on resource consumption and latency quickly compounds. Utilizing the Linkerd2-proxy allows Linkerd to maximize the speed and security of the data plane while optimizing resource consumption. Recent benchmarks conducted by Kinvolk GmbH (an open source engineering and technology company recently acquired by Microsoft) found Linkerd to be significantly faster than open source Istio while consuming an order of magnitude less data plane memory and CPU.

As the original creator of Linkerd, Buoyant recently launched Buoyant Cloud, an automated and unified service mesh dashboard built to monitor, assess, and validate the health of Linkerd clusters. Tracking data and control plane metrics, Buoyant Cloud identifies data plane inconsistencies, manages mesh lifecycles and versions, and proactively issues alerts.

Strengths: Designed from the ground up as a lightweight, security-first service mesh supporting mission-critical features for cloud-native applications using Kubernetes, Linkerd is the only service mesh committed to operational simplicity. If you're looking for an observable, reliable, and secure Kubernetes platform that won't break your budget, Linkerd should be on your shortlist.

Challenges: Linkerd's focus on a single use case may limit its application for particular enterprises and organizations. Moreover, Linkerd's data plane proxy supports only Kubernetes at this time. Customers requiring support for VMs or hybrid environments should explore Linkerd's roadmap for expanding support beyond Kubernetes clusters, including new ways to simplify and automate the implementation of policies between services.

Decipher Technology Studios: Grey Matter

Developed in-house from the ground up and released in February 2019, Grey Matter is an enterprise-proven, universal service mesh networking platform offering zero-trust security, exceptional Layer 3, 4, and 7 visibility, unmatched business intelligence, and automated performance optimization. Addressing many of the challenges introduced by a service-based architecture (SBA), Grey Matter is built on cloud-native principles and open source technologies, enabling granular service mesh-enabled observability, analytics, and automation to optimize traffic throughput across on-premises, multi-cloud, or hybrid environments.

Grey Matter At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Grey Matter	Consul/Envoy	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane			Control Plane
C++	Go	Rust	Go
Primary Use Cases			
Service-based architecture at scale		Anomaly detection	
Modernization, automation, and edge computing		Governance and compliance	
Pricing Model			
Pricing is based on an OPEX, per node subscription model			

The platform comprises an internally developed control plane for service-based architectures, and either an Envoy or Hashicorp Consul-based sidecar data plane with extended filters for east-west internal traffic routing. An API gateway controls north-south traffic flows. In addition, Grey Matter integrates with Open Policy Agent (OPA) for zero-trust, policy-based access control at every point on the mesh, and is flexible and open enough to interoperate with other service meshes.

Delivering a comprehensive audit-compliance engine and SPIFFE/SPIRE identity authorization out of the box, Grey Matter provides service audit compliance reporting without special instrumentation. (Note: SPIRE is a production-ready implementation of SPIFFE—the Secure Production Identity Framework for Everyone.) Real-time audit taps at Layers 3, 4, and 7 provide a single source of truth for every user and action on the mesh throughout the lifespan of each object.

Designed to treat proxy-based service mesh telemetry as a source of business intelligence, Grey Matter leverages AI and machine learning to analyze data, including Layers 3, 4, and 7 network insights, for automated performance optimization and resource control. Powered by recurrent neural autoencoders, Grey Matter's anomaly detection capabilities capture minute operational inconsistencies, predict potential issues, and alert users to inconsistencies against normal operational patterns via an intuitive contextual UI to take remedial action.

Grey Matter is designed to be platform-agnostic and polyglot. The platform wraps existing IT investments in a ubiquitous Layer 3, 4, and 7 network, securely connecting existing operations support system and business support system (OSS/BSS) layers to cloud-native technologies. Capable of operating in any public, private, hybrid, or multi-cloud or container orchestration platform, Grey Matter comes with built-in support for K8s, AWS EKS, Azure AKS, OpenShift OCP, OKD, Konvoy, and

bare metal. It is also container-agnostic, supporting Docker, CoreOS, OpenShift, Rancher, and other containers.

Strengths: In addition to providing a robust, enterprise-ready, container-agnostic, multi-environment platform, Grey Matter's mesh-enabled AI contextual awareness and analytics offer enhanced observability and optimization, improving resource management and availability with intelligent auto-scaling, routing, and load balancing. The automated OPA distribution helps implement governance and ensure compliance, with comprehensive audit trails for improved management and control.

Challenges: Despite using the Consul and Envoy proxies, Grey Matter does not yet provide an open source solution, which puts it at a disadvantage compared to its competitors. We expect this to change as the company increases engagement with the Envoy open source community over the next 12-18 months. In addition, Grey Matter is a small, bootstrapped company predominantly focused on U.S. government and Department of Defense clients, limiting both market awareness and the opportunity to promote customer successes.

F5 Networks: Aspen Mesh

A startup incubated within F5 Networks, Aspen Mesh was released in December 2017 as a fully supported enterprise-grade, secure-by-default Istio distribution for Kubernetes (see Istio in this document for additional information). The service mesh incorporates multi-cloud zero-trust security, compliance policy enforcement, protocol-level observability, and site reliability engineering (SRE)-based application optimization. Leveraging F5 Networks' global infrastructure, Aspen Mesh offers 24×7 white glove and concierge support for production environments with follow-the-sun options and on-demand, native-speaking support engineers.

Aspen At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Istio	Envoy	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	-	X
Languages			
Data Plane		Control Plane	
C++	WebAssembly (WASM)	Go	Python
Primary Use Cases			
Organizational standards for application-level services		Observability up and down the application stack	
5G traffic management, security, and compliance		Machine-assisted root cause analysis	
Pricing Model			
Pricing is based on an OPEX, per-node subscription model with optional paid services.			

The first of the open source Istio-based service meshes included in this report, Aspen Mesh reduces the complexity of Istio through lifecycle management, long-term support (LTS) releases, and additional services, adding advanced features to the open source distribution. These include simplified mutual Transport Layer Security (mTLS) management, fine-grained role-based access control (RBAC), Istio Vet (for discovering incompatible user application and Istio component configuration in a Kubernetes cluster), and single sign-on (SSO). In addition, objective-driven, AI/ML-powered insight recognition policy frameworks allow users to specify, measure, and enforce business goals, while the Aspen Mesh dashboard offers an intuitive user experience.

Aspen Mesh holds seats on the Istio Technical Oversight and Steering Committees and was the first non-founding vendor to release and manage a version of Istio. It's also one of the primary contributors to the Istio and Envoy communities. Moreover, Aspen Mesh's Packet Inspector provides protocol-level observability for microservices in production, while Rapid Resolve reduces mean-time-to-resolution (MTTR) with advanced troubleshooting and environment reporting capabilities.

A more robust version, Carrier-Grade Aspen Mesh works in conjunction with BIG-IP Service Proxy for Kubernetes (SPK), developed by F5 specifically for cloud-native 5G networks. SPK provides signaling protocol support unavailable in the standard Kubernetes platform, streamlining transitions to 5G while leveraging investments in 4G. The distribution offers authentication, encryption, observability, security, policy management, and packet capture of east-west traffic within each 5G core Kubernetes cluster. At the same time, a per-service secure proxy and firewall protects north-south traffic flowing into and out of containerized 5G services. Automated Kubernetes service discovery and policy configuration provide increased agility and engineering efficiency.

It should be noted that F5 Networks supports two service meshes—Aspen Mesh and NGINX Service Mesh—on the premise that customers often hold strong opinions on their choice of infrastructure stack based on existing investments in underlying technologies. While both support Kubernetes clusters, F5 provides options for either standardizing on NGINX infrastructure or adopting a service mesh based on open source Istio and Envoy.

Strengths: As an F5 Networks incubation, Aspen Mesh is uniquely positioned to help network service providers address the challenges that come with transitioning to 5G and cloud-native technologies. Carrier-Grade Aspen Mesh supports Cloud-native Network Functions (CNF), providing security, policy management, and observability of east-west traffic within each 5G core Kubernetes cluster. It also offers volumetric telemetry to isolate inter-function and intra-function issues rapidly.

Challenges: With several vendors providing enterprise-grade support for Istio, Aspen Mesh needs to find ways to differentiate itself for enterprise customers. While Carrier-Grade Aspen Mesh is a key differentiator for service providers, Aspen Mesh needs to focus on simplifying the Istio experience for enterprises and developing actionable, machine-assisted insights to help address customers' challenges.

F5 Networks: NGINX Service Mesh (NSM)

Released in May 2021, NGINX Service Mesh (NSM) is a developer-friendly, fully integrated, lightweight service mesh that leverages a data plane powered by NGINX Plus (a cloud-native, easy-to-use reverse proxy, load balancer, and API gateway) to manage container traffic in Kubernetes environments. NSM implements the Service Mesh Interface (SMI) specification, which defines a standard interface for service meshes on Kubernetes, and provides SMI extensions to update apps incrementally with minimal effort and interruption to production traffic.

NGINX ServiceMesh At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
NGINX	NGINX Plus	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane		Control Plane	
C++		Go	
Primary Use Cases			
Improve resilience		Reduce complexity	
Secure containerized apps		Deliver fine-grained traffic insights	
Pricing Model			
Free download with open source community support			

Integrating natively with NGINX Ingress Controller, NGINX Service Mesh creates a unified data plane to centralize and streamline configuration of ingress and egress (north-south) traffic management at the edge with service-to-service (east-west) reverse proxy sidecar traffic management. Unlike other service meshes, NSM does not automatically inject a sidecar into each workload, including NGINX Ingress Controller. Policies for manual or auto-injection depend on the deployment options chosen, allowing users to minimize latency and reduce complexity within Kubernetes environments, if desired.NSM offers a robust set of traffic distribution models, including rate shaping, quality of service (QoS), service throttling, blue-green deployments, canary releases, circuit breaker pattern, A/B testing, and API gateway features. Easy to use and infrastructure agnostic, the lightweight control plane manages NGINX Plus reverse proxy sidecars and data plane.

Extending mTLS encryption and Layer 7 protection down to the individual microservice, NGINX Service Mesh enables advanced security features, including configuration gating and governance, and zero-trust end-to-end encryption and service authorization. In addition, the NGINX Plus-based version of NGINX Ingress Controller provides default blocking of north-south traffic to internal

services and edge firewalling with NGINX App Protect.

NGINX Service Mesh is instrumented for metrics collection and analysis using OpenTracing and Prometheus, while the NGINX Plus API generates metrics from sidecars and NGINX Ingress Controller pods. The built-in Grafana dashboard can be used to visualize granular metrics, day-over-day overlays, and traffic spikes.

NSM is supported on several Kubernetes platforms, including Amazon Elastic Container Service for Kubernetes (EKS), the Azure Kubernetes Service (AKS), Google Kubernetes Engine (GKE), VMware vSphere, and stand-alone bare-metal clusters. It also integrates with several open source solutions, including Grafana, NATS, Kubernetes Ingress controllers, Open Tracing, Prometheus, and SPIRE.

Strengths: NGINX is an established open source leader in application delivery, with over 400 million websites worldwide relying on NGINX Open Source and NGINX Plus to deliver content quickly, reliably, and securely. For enterprises that want to avoid the complexity that comes with Kubernetes or Istio-based service mesh deployments, NGINX Service Mesh offers a simple, multi-cloud solution supporting scalability, security, reliability, and enterprise readiness.

Challenges: NGINX Service Mesh relies heavily on core NGINX components, generally limiting its application to customers committed to NGINX—and now F5—infrastructure. F5 addressed this limitation by acquiring and incubating Istio- and Envoy-based Aspen Mesh. While NGINX Service Mesh can be downloaded for free on F5's website, support is available only through the open source community, unless other F5 or NGINX products are purchased.

Hashicorp: Consul

Developed internally from the ground up and released as a service mesh in October 2018, Consul is actively maintained and supported by HashiCorp. Initially designed as a simple service discovery and key/value store before containers became mainstream, Consul has evolved to become a full-featured service mesh. The platform works out of the box with a simple built-in L4 proxy and supports third-party proxy integrations, including Envoy. In addition, HCP Consul is a fully managed Service Mesh-as-a-Service running on the HashiCorp Cloud Platform (HCP) offering push-button and self-service deployments.

Consul At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Consul	Consul/Envoy/HAProxy/NGINX	HCP Consul	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane		Control Plane	
C++		Go	
Primary Use Cases			
Service discovery and observability		Zero-trust networking	
Progressive delivery		Network infrastructure automation	
Pricing Model			
Pricing is based on an OPEX, per-hour subscription model. A free open source distribution is also available.			

Consul provides a full-featured control plane with service discovery, configuration, and segmentation functionality, and allows each feature to be used independently as needed. Closing the gap between applications and networking, Consul provides a step-by-step approach, allowing organizations to deploy service discovery and service registry before building out the service mesh implementation. It also offers networking infrastructure automation for dynamic IP environments.

Offered as either a self-hosted or managed solution—providing flexibility for enterprises of all sizes—Consul provides discovery and secure connectivity for any application running on any infrastructure or runtime. As a common control plane connecting modular applications everywhere, the service mesh provides security features such as ACLs, mTLS, and Certificate Authority (CA) distribution. It also offers progressive delivery capabilities—for canary deployments, Layers 4 and 7 traffic management, and advanced observability—for containers, VMs, and bare-metal environments. While not a typical service mesh feature, Consul can also automate L3 networking tasks, including dynamic firewalling, automated load balancing, and endpoint visibility.

Consul provides a consistent view of all services on the network, irrespective of different programming languages and frameworks, for real-time services like health and location monitoring. Correctly instrumented applications can send open tracing data through Envoy, and Envoy proxies can be configured to collect Layer 7 metrics and export them to monitoring tools such as Prometheus.

An extensible, multi-platform solution with flexible procurement options, Consul supports both on-premises (virtualized and bare metal) and cloud deployments, as well as multiple runtimes such as Kubernetes and Hashicorp Nomad. It also offers native capabilities and integrations for proxies (including Envoy, HAProxy, and NGINX), ingress solutions (including Ambassador and Nginx), and application performance monitoring (APM) solutions such as AppDynamics, Datadog, and Splunk.

Strengths: Consul is a simple, flexible service mesh offering multi-cluster support and integrations with external non-service-mesh workloads. While the free, open source version depends on community support, Consul is a great choice if you use other Hashicorp products and require production support. It's tightly integrated with the company's portfolio, and the Service Mesh-as-a-Service offering, Managed Consul, is an attractive option for Hashicorp customers looking for push-button and self-service deployments.

Challenges: With only a small open source community providing support, Consul's primary value is for existing Hashicorp users wishing to incorporate Kubernetes into their Hashicorp stack.

Istio

Released in May 2017, Istio is an ongoing collaboration among Google, IBM, Lyft, Red Hat, and other key contributors. One of the more mature—and complex—service meshes available, Istio offers a rich feature set, including dynamic service discovery, service-to-service authentication, load balancing, monitoring, policy creation, and traffic routing, based on the Envoy Proxy and a robust control plane tightly integrated with Kubernetes. In addition, Istio recently added full support for virtual machines, creating a unified control plane for managing both Kubernetes and VMs.

Istio At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Istio	Envoy	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	-	X
Languages			
Data Plane		Control Plane	
C++		Go	
Primary Use Cases			
Defense-in-depth for enterprise applications		Deployment and management efficiency	
Observability and SRE best practices		Secure cloud-native apps	
Pricing Model			
Free download with open source community support			

Istio has strong identity-based authentication and authorization, and offers fine-grained control of traffic behavior with advanced routing policies and management, including circuit breakers, failovers, fault injection, health checks, retries, and staged rollouts. In addition, its configuration API and policy layer support access, quota, and rate controls, while detailed logs, metrics, and traces provide in depth observability throughout the cluster.

However, as new features and functions are added, Istio is notoriously tricky to install, configure, and manage. The original idea of separating components based on operations and maintenance roles increased complexity and cost, especially for companies in which one person or team was responsible for the entire service mesh.

The project itself addressed Istio's complexity by abandoning its microservices *architecture* in favor of a monolithic approach, merging multiple, previously separate functions to simplify the service mesh and minimize the tradeoffs. However, Istio is retaining a microservices *approach* with strict boundaries between the code and what were formerly independent services. As a result, it is now a single process from the perspective of the cluster administrator.

Istio development is ongoing, with a centralized, multi-cluster controller, additional enhancements for supporting VMs, and stability improvements included in recent releases. While this approach may be good from an engineering perspective, Istio's quarterly release cycle may impact operational stability. In fact, Istio's complexity has resulted in a growing ecosystem with several vendors—including F5, Solo.io, and VMware—emerging to provide Istio-based service meshes underpinned by enterprise-grade services and support.

Although the open source and cloud-native communities wanted Istio to be donated to the Cloud Native Computing Foundation, Google decided to donate the trademark to the Open Usage Commons (OUC) comprising current and past employees, partners, and academicians from Google-funded institutions. This move allowed Google to retain control over the Istio project, influencing its roadmap and ensuring continued support for the company's portfolio.

After Kubernetes, Istio is a core building block for Google's technology stack. Anthos (Google's application modernization platform), Cloud Run (a commercial, proprietary Google Cloud service), and Knative (a serverless platform built on Kubernetes), are all dependent on Istio. An Istio-based service mesh enabling hybrid and multi-cloud capabilities, Anthos Service Mesh underpins Google's 5G mobile edge cloud and multi-cloud strategies. With so much riding on it—and despite the ambitions of other Istio partners—Google will most likely continue to be the driving force behind Istio's roadmap.

Strengths: Due to the marketing efforts of Google and IBM, "Istio" is often used interchangeably with "service mesh," positioning it as the go-to solution for adding observability, reliability, and security to the cloud-native stack. Compared to other service meshes, Istio's maturity, breadth of out-of-the-box features, and adoption by major industry players often ensure its inclusion on any service mesh shortlist.

Challenges: Due to its advanced features and complex configuration requirements, Istio is not as user- or developer-friendly as other service meshes. It often requires either a dedicated team or third-party professional services to assist in costly, resource-intensive, and time-consuming implementations. Moreover, with the Istio project supporting only the three latest releases (N-2), a quarterly release cycle can be overwhelming for teams with limited capacity and skills. This pressure can have a significant impact on development budgets and operational stability.

Kong: Kong Mesh

Released for GA in August 2020, Kong Mesh is a modern, enterprise-ready control plane for service mesh and microservices built on top of Envoy and Kuma, the open-source project authored by Kong and donated to the Cloud Native Computing Foundation. Kong Mesh extends Kuma's existing advanced feature set by including critical functionality for running enterprise workloads. Kong Mesh also provides additional service mesh features and integrations for the Kong Konnect platform, a full-stack connectivity platform delivered as-a-service for multi-cloud environments.

Kong Mesh At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Kuma	Envoy	Kong Konnect	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane		Control Plane	
C++	Go	Go	
Primary Use Cases			
Single cluster and multi-zone service connectivity		Intelligent, SLA-driven traffic routing	
Microservices adoption and transformation		Zero-trust security and GDPR compliance	
Pricing Model			
Pay-as-you-go model based on the number of data plane proxies connected to the control plane			

Deployed as a turnkey service mesh with a single command, Kong Mesh allows multiple service meshes to be managed as tenants of a single control plane to increase scale and reduce operational costs. Once installed, Kong Mesh improves service connectivity via policies that can be added to each mesh, service, or attribute that qualifies a traffic path, accelerating developer efficiency, cost reduction, GDPR compliance, and zero-trust security.

The latest release includes integrations with the open source, policy-as-code tool Open Policy Agent (OPA) for Layer 7 policy support, automatic configuration of Envoy for FIPS 140-2 compliance, and authentication between global and isolated control planes. Kong Mesh automates the distribution of those policies throughout multi-cluster and multi-region deployments, eliminating the need for manual configuration. It also extends the service mesh and OPA to include legacy infrastructure such as virtual machines.

Focused on ease of use, Kong Mesh leverages Kuma to deliver a supported, multi-mesh product that can scale across teams and lines of business, while at the same time providing cross-cluster and cross-cloud connectivity for modern architectures. Accelerating configuration and deployment, Kong

Mesh abstracts away the complexity of setting up a service mesh by encapsulating Envoy within its own processes. A native GUI provides quick visual feedback of what is happening in the system.

Supporting both Kubernetes and VM workloads, Kong's "run anywhere" philosophy allows Kong Mesh to be deployed across any environment, including multi-cluster, multi-cloud, and multi-platform. Organizations can use Kong Mesh's custom resources definitions (CRDs) to natively manage service meshes in Kubernetes, or start with a service mesh in VM environments and migrate to Kubernetes at their own pace.

Automating distributed service mesh policy propagation, Kong Mesh's universal mode provides advanced, multi-zone support with out-of-the-box discovery and connectivity of clouds, platforms, hybrid containers, and VMs, with automatic policy reconciliation across multiple zones. Kong Mesh also supports zones on non-Kubernetes containerized environments like AWS ECS and AWS Fargate.

Strengths: Kong Mesh's ease of use and built-in automation capabilities will resonate with users wishing to deploy a service mesh without encountering the configuration headaches typically associated with open source solutions. In addition, security-conscious organizations will be attracted by Kong Mesh's FIPS 140-2 compliance and consistent application of security policies across all environments.

Challenges: As a new entrant in the space, only time will tell if Kong's goal of becoming the "Switzerland" of connectivity by providing a full-stack platform running everywhere will become a reality. The company offers a "predictable and linear, pay-as-you-go" pricing model, but actually calculating the cost based on the number of data plane proxies connected to the control plane is a challenge. Kong needs to find a way to simplify its pricing model and make it easier for customers to calculate the potential total cost of ownership (TCO).

Solo.io: Gloo Mesh

Originally launched in early 2019, Gloo Mesh (previously known as Service Mesh Hub) is a modern Kubernetes-native control plane enabling the configuration and operational management of multiple heterogeneous service meshes across multiple clusters via a unified API. The Gloo Mesh API integrates with leading service meshes and abstracts away differences between their disparate APIs, streamlining the configuration, operation, and lifecycle management of multi-cloud, multi-mesh environments. Gloo Mesh comes in two editions: an open source version and the commercial, enterprise-ready Gloo Mesh Enterprise, sold as a standalone product. In March 2021, the company announced Gloo Cloud, the first and only Istio-based Service Mesh-as-a-Service (SMaaS).

Gloo Mesh At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Istio/Gloo Mesh	Envoy	Gloo Cloud	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	X	X
Languages			
Data Plane		Control Plane	
C++	WebAssembly (WASM)	Go	
Primary Use Cases			
Service mesh management		API gateway integration	
Multi-mesh and multi-cluster support		Production Istio support	
Pricing Model			
Pricing is based on an OPEX, topology-based subscription model. A free open source distribution is also available.			

Gloo Mesh can be run either in its own cluster or co-located with an existing mesh, enabling global traffic routing, load balancing, access control, and centralized observability of multi-cluster environments. It discovers meshes and workloads and establishes a federated identity, enabling the configuration of different service meshes through a single API. Gloo Mesh supports multi-platform service meshes spanning clouds and zones (including AWS App Mesh, open source Istio, and Microsoft Azure Open Service Mesh), locality-aware routing, and cross-cluster failover supporting zero trust networks.

Solo also provides Gloo Edge, a decoupled control plane for the Envoy Proxy. It allows customers to iteratively add service-mesh capabilities to their cluster ingress without investing in a full-blown service mesh. Moreover, it integrates with Flagger (a delivery tool that automates the release process for Kubernetes workloads) for canary automation, and plugs in natively to Consul, Istio, and Linkerd service-mesh implementations.

According to Solo.io, Gloo Mesh is a critical component of the Gloo API Infrastructure Platform, providing a modern approach to API management and accelerating innovation across distributed cloud workloads and environments. Additionally, Gloo Mesh integrates with the Gloo Edge API Gateway for end-to-end encryption, security, and traffic control, incorporating traffic management into both east-west and north-south data transfer flows.

An enhanced version of open source Istio (as opposed to a fork), Gloo Mesh Enterprise also includes an extended version of the Envoy Proxy. This enables the consistent configuration and orchestration of services across multiple VMs, clusters, clouds, and data centers from a single point of control. Focusing on ease of use, Gloo Mesh Enterprise validates upstream Istio software and incorporates

built-in best practices for extensibility and security, including role-based APIs.

Gloo Mesh is designed to simplify the operations and lifecycle management of multi-cloud, multi-mesh environments, providing both graphical and command-line UIs, multi-cluster observability, and debugging tools. Solo.io also created the WebAssembly Hub, a streamlined service for building, sharing, discovering, and deploying WebAssembly (WASM) extensions for managing traffic and delivering near-native performance of Envoy Proxy-based service meshes.

While the community supports the open source version of Gloo Mesh, Gloo Mesh Enterprise provides production and long-term support (LTS) with patches and hotfixes for the last four releases (N-3) of validated upstream Istio implementations with dedicated SLAs. In addition to traditional support channels, Solo.io also provides a Slack-based support channel for customers.

Strengths: Solo.io not only provides extended Istio support, but also an enhanced, production-ready version of open source Istio and Envoy, with additional innovation in Gloo Edge, Gloo Cloud, and WebAssembly Hub. The company has also been successful in recruiting recognized industry leaders in an attempt to wrestle the initiative away from Google and take the lead in influencing Istio's direction.

Challenges: Despite investing heavily in talent and innovation, Solo.io is still dependent on open source Envoy and Istio for its core offerings. Solo.io's ability to influence the direction of open source Istio is limited considering Google's dominance in the community's Steering and Technical Oversight Committees. While some customers will welcome Solo.io's innovation around Istio, others may balk at being tied to Google's strategy.

Traefik Labs: Traefik Mesh

Released in September 2019 and known previously as Maesh, Traefik Mesh is a simple, straightforward, and non-invasive service mesh utilizing Traefik Proxy, rather than Envoy, to manage service-to-service communications inside a Kubernetes cluster. Created and maintained primarily by Traefik Labs (previously known as Containous) and with over 2 billion downloads and more than 30,000 GitHub stars, Traefik Proxy is one of the most used cloud-native application proxies. Traefik Labs claims Traefik Mesh to be the simplest and easiest service mesh to deploy for enhanced control, security, and observability across all east-west traffic flows with minimal overhead.

Integrating natively with Kubernetes, Traefik Mesh is a lightweight service mesh supporting the latest Service Mesh Interface (SMI) specification. Traefik Mesh is the only mesh included in this report using a per node architecture instead of a sidecar proxy for simplicity and resource conservation. Since Traefik Mesh is opt-in by default, existing services are unaffected until explicitly added to the service mesh rather than being automatically injected into the application.

Traefik Mesh At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Traefik	Traefik	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	-	-	-
Languages			
Data Plane		Control Plane	
Go		Go	
Primary Use Cases			
Service discovery		Optimized communications	
Enhanced security		Accelerated delivery	
Pricing Model			
Open source community support or subscription-based enterprise support			

Designed for simplicity with a focus on efficiency and low-resource utilization, Traefik Mesh is easy to install and configure via a command-line interface (CLI). Its feature set includes traffic management capabilities such as circuit breakers, load balancing, retries and failovers, and rate-limiting. Traefik Mesh provides observability with out-of-the-box metrics preinstalled with Grafana and Prometheus, and is compatible with Datadog, InfluxData, and StatsD. Tracing is supplied through any OpenTracing solution, with full compatibility with Haystack, Instana, Jaeger, and Zipkin for resilient, scalable tracing and analysis.

In addition to basic security in the form of mTLS, Traefik Mesh is SMI-compliant and facilitates the fine-tuning of traffic permissions via access control. A specification for service meshes running on Kubernetes, SMI defines a common standard for service mesh providers, covering the most common capabilities and enabling flexibility and interoperability. Furthermore, since SMI is specified as a collection of Kubernetes APIs, users who know Kubernetes can use Traefik Mesh.

Built on top of open source Traefik Proxy and Traefik Mesh, Traefik Enterprise consolidates API management, ingress control, and service mesh within one simple control plane. A unified, cloud-native networking solution, Traefik Enterprise simplifies microservices networking complexity with distributed, highly available, and scalable features combined with premium, subscription-based bundled support for enterprise-grade deployments. In addition, Traefik Enterprise includes an enhanced dashboard with service mesh observability of internal east-west traffic.

Strengths: Utilizing the popular Traefik Proxy, Traefik Mesh offers lightweight SMI-compliant and non-invasive traffic management with good usability and performance. It's unique in this report because it doesn't use a sidecar proxy, but rather an opt-in, per node proxy connecting services for increased control and resource conservation. In addition, Traefik Mesh comprises core service mesh features, including circuit breakers, load balancing, rate limiting, retries and failovers, and security, as well as observability and out-of-the-box metrics.

Challenges: Traefik Mesh lacks multi-cluster capabilities, so if you're looking for a unified control plane spanning clusters, clouds, and meshes, you'll need to look elsewhere. And while it supports the SMI access control, it doesn't offer transparent, end-to-end encryption. In addition, Traefik Mesh does not support VMs.

VMware: Tanzu Service Mesh (TSM)

Initially released in December 2018 as NSX Service Mesh, Tanzu Service Mesh is an Istio-based, enterprise-class service mesh providing consistent connectivity and security for microservices across multi-cluster and multi-cloud Kubernetes environments. Tanzu Service Mesh integrates tightly with VMware's Kubernetes platform, Tanzu Kubernetes Grid, to provide standard service mesh capabilities via the Istio API. On top of this it layers unique, end-to-end use case support and integrated solutions that are challenging to achieve with service mesh technologies alone.

Developed in-house from the ground up, the TSM Global Controller is VMware's control plane and the primary means for supporting advanced use cases beyond those available with open source Istio and Envoy. These use cases include application continuity, resiliency, and security for single and multi-cluster environments running across Kubernetes clusters, clouds, and third-party service meshes.

Tanzu Service Mesh includes the TSM Global Controller—a control plane provided as a SaaS managed by VMware—and the TSM Data Plane running across customers' Kubernetes clusters. Based on open source Istio and Envoy, the TSM Data Plane delivers typical services such as authentication and authorization, circuit breaking, rate-limiting, timeouts and retries, traffic shifting, and other features. TSM Data Plane also includes the TSM Agent, providing a secure connection between the customers' clusters and the TSM Global Controller for managing the configuration and policies enforced in the TSM Data Plane.

Tanzu Service Mesh includes a unique application abstraction layer called Global Namespace (GNS), which acts as a logical grouping for microservices. Managed through an intuitive UI, GNSs provide modern applications with simplified configurability, API-driven automation, isolation, and operational consistency for DevOps and security, irrespective of the underlying platform or cloud. They also provide automated service discovery and naming (DNS), resiliency policies, security policies, service graphs, and traffic routing. Encompassing full automation of multi-cluster configuration, ingress and egress configuration, and seamless cross-cloud application portability, GNS supports microservices within a single cluster and microservices distributed across multiple clusters and clouds.

Tanzu Service Mesh At-a-Glance

Architecture		Management Options	
Control Plane	Data Plane Proxy	SMaaS	Managed
Istio	Envoy	-	-
Control Plane Support			
Kubernetes	Virtual Machines	Bare Metal	Hybrid
X	X	-	-
Languages			
Data Plane		Control Plane	
C++	Go	Go	TypeScript
Primary Use Cases			
Zero trust, secure multi-cloud connectivity		Multi-cloud application scalability	
Service bridging for application modernization		Performance and security monitoring and visibility	
Pricing Model			
Pricing is based on an OPEX, per-core subscription model with production support included.			

Providing Istio cluster onboarding and automated health monitoring and lifecycle management of the Istio/Envoy data plane, TSM is integrated with the full-stack capabilities of Tanzu—VMware's portfolio for modernizing applications and infrastructure. In addition, TSM works with VMware's NSX Advanced Load Balancer (formerly Avi Networks) to provide multi-cloud support, unified policies, load balancing, ingress, container networking, and observability across VMware and third-party Kubernetes environments.

VMware recently closed the acquisition of Mesh7 to accelerate the TSM roadmap, delivering support for Linux VMs on private and public clouds. Mesh7 provides contextual API observability and API security based on Envoy, delivering high-fidelity visibility and policy controls for APIs and API traffic. According to VMware, TSM's roadmap also includes integration with more of the Tanzu portfolio, including API Hub for VMware Tanzu, Spring Cloud Gateway, Tanzu Application Service (TAS), and vSphere with Kubernetes.

Strengths: Leveraging open source Istio, Tanzu Service Mesh provides robust enterprise services across multiple Kubernetes clusters, offering operational simplification and automation with advanced resiliency functions. In addition to supporting various application platforms, public clouds, and runtime environments, Tanzu Service Mesh supports federation across multiple clusters for end-to-end connectivity, resiliency, and security.

Challenges: Since Tanzu Service Mesh is deeply integrated with VMware's vertical stack, users adopting complementary VMware products will realize the full benefit. While this provides excellent value for VMware's installed base, non-VMware customers and those lured by the promise of a free, open source service mesh will find their choices somewhat limited.

6. Analyst's Take

As a relatively new sector with the "original" service mesh only introduced in 2016, it is still too early to predict which service mesh will emerge eventually as the winner. Today, Istio appears to be the one to beat based on customer adoption and vendor support. However, irrespective of its "home," Istio's roadmap is currently driven by Google to support its Istio-based portfolio. While this may provide a level of comfort for some enterprises, others may hesitate to adopt an "open source" service mesh controlled by one of the foremost cloud vendors rather than guided by the community.

Furthermore, F5's Aspen Mesh, Solo.io's Gloo Mesh, and VMware's Tanzu Service Mesh are all Istio-based distributions. While they provide much-needed enhancements and enterprise-grade support, they are still, at this point, inextricably linked to Google's strategy. Whether this will impact their eventual success or not remains to be seen. F5, Solo.io, and VMware are investing heavily in Istio to meet the needs of their respective customers. We anticipate Solo.io to be the disruptor in the space as it looks for ways to realize its goal of becoming the dominant player in the Istio ecosystem.

At the same time, there are several solid non-Istio service mesh offerings. For example, Linkerd and Traefik Mesh both offer lightweight service meshes that are easy to deploy and manage. Addressing some of the limitations of early service meshes, Kuma and especially Kong Mesh are rapidly emerging as serious contenders. Moreover, Grey Matter's mesh-enabled AI contextual awareness and analytics capabilities offer significant benefits over existing open source service meshes.

Our advice is to avoid adopting a service mesh based purely on industry hype. Do your homework. Understand the problem you're trying to solve, explore the potential tradeoffs, evaluate your resources and skills, and then choose a service mesh—and partner—that works best with your software stack.

While an Envoy or Istio-based service mesh or one with widespread support may be the "safe" choice, that should not always be the determining factor. Many use cases can be supported with an easy-to-use, lightweight, and infrastructure-agnostic service mesh incorporating essential functionality and supporting both east-west and north-south traffic.

7. About Ivan McPhee

Formerly an enterprise architect and management consultant focused on accelerating time-to-value by implementing emerging technologies and cost optimization strategies, Ivan has over 20 years' experience working with some of the world's leading Fortune 500 high-tech companies crafting strategy, positioning, messaging, and premium content. His client list includes 3D Systems, Accenture, Aruba, AWS, Bespin Global, Capgemini, CSC, Citrix, DXC Technology, Fujitsu, HP, HPE, Infosys, Innso, Intel, Intelligent Waves, Kalray, Microsoft, Oracle, Palette Software, Red Hat, Region Authority Corp, SafetyCulture, SAP, SentinelOne, SUSE, TE Connectivity, and VMware.

An avid researcher with a wide breadth of international expertise and experience, Ivan works closely with technology startups and enterprises across the world to help transform and position great ideas to drive engagement and increase revenue.

8. About Chris Grundemann

Chris Grundemann is a passionate, creative technologist and a strong believer in technology's power to aid in the betterment of humankind. He is currently expressing that passion by helping technology businesses grow and by helping any business grow with technology.

Chris has well over a decade of experience as both a network engineer and solution architect designing, building, securing, and operating large IP, Ethernet, and Wireless Ethernet networks. He has direct experience with service provider and enterprise environments, design and implementation projects, for-profit and not-for-profit organizations, big picture strategic thinking and detailed tactical execution, and standards and public policy development bodies. Chris frequently works with C-level executives and senior engineering staff at internet and cloud service providers, media and entertainment companies, financials, healthcare providers, retail businesses, and technology start-ups.

Chris holds eight patents in network technology and is the author of two books, an IETF RFC, a personal weblog, and a multitude of industry papers, articles, and posts. In addition to being the lead research analyst for all networking and security topics at GigaOm, he is the co-host of Utilizing AI, the Enterprise AI podcast. He is also a cofounder and Vice President of IX-Denver and Chair of the Open-IX Marketing committee. Chris has given presentations in 34 countries on 5 continents and is often sought out to speak at conferences, NOGs, and NOFs the world over.

Currently based in West Texas, Chris can be reached via Twitter.

9. About GigaOm

GigaOm provides technical, operational, and business advice for IT's strategic digital enterprise and business initiatives. Enterprise business leaders, CIOs, and technology organizations partner with GigaOm for practical, actionable, strategic, and visionary advice for modernizing and transforming their business. GigaOm's advice empowers enterprises to successfully compete in an increasingly complicated business atmosphere that requires a solid understanding of constantly changing customer demands.

GigaOm works directly with enterprises both inside and outside of the IT organization to apply proven research and methodologies designed to avoid pitfalls and roadblocks while balancing risk and innovation. Research methodologies include but are not limited to adoption and benchmarking surveys, use cases, interviews, ROI/TCO, market landscapes, strategic trends, and technical benchmarks. Our analysts possess 20+ years of experience advising a spectrum of clients from early adopters to mainstream enterprises.

GigaOm's perspective is that of the unbiased enterprise practitioner. Through this perspective, GigaOm connects with engaged and loyal subscribers on a deep and meaningful level.

10. Copyright