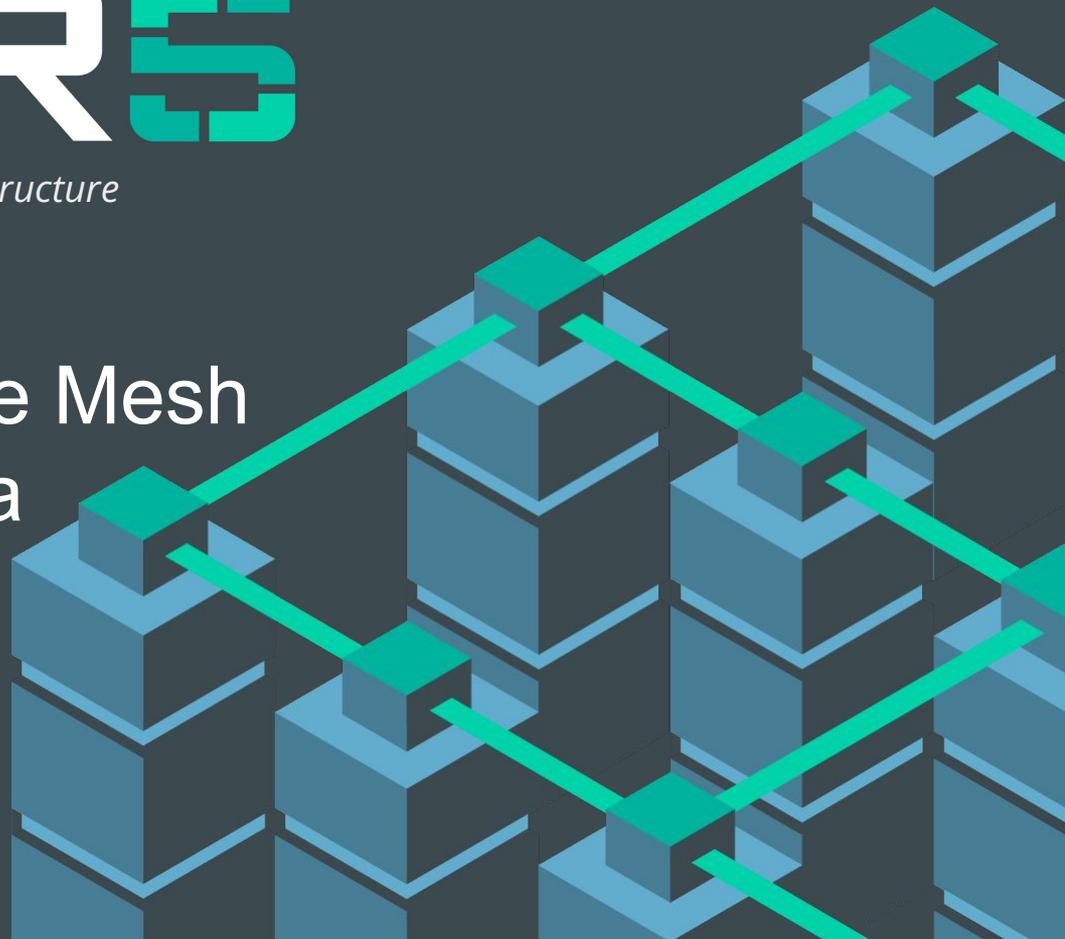


LAYER5

Expect more from your infrastructure

Solving the Service Mesh Adopter's Dilemma



Content



- Introduction
- Getting started with Service Meshes
- Different service meshes
- Service Mesh Architecture
- Functionality
- Roadmap
- Meshery
- Which Mesh should I adopt

Distributed systems management is hard

Particularly without openly governed, enterprise-grade management software



Physical networking is difficult.



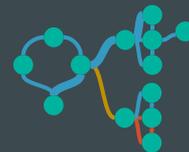
Cloud native networking is even more challenging.



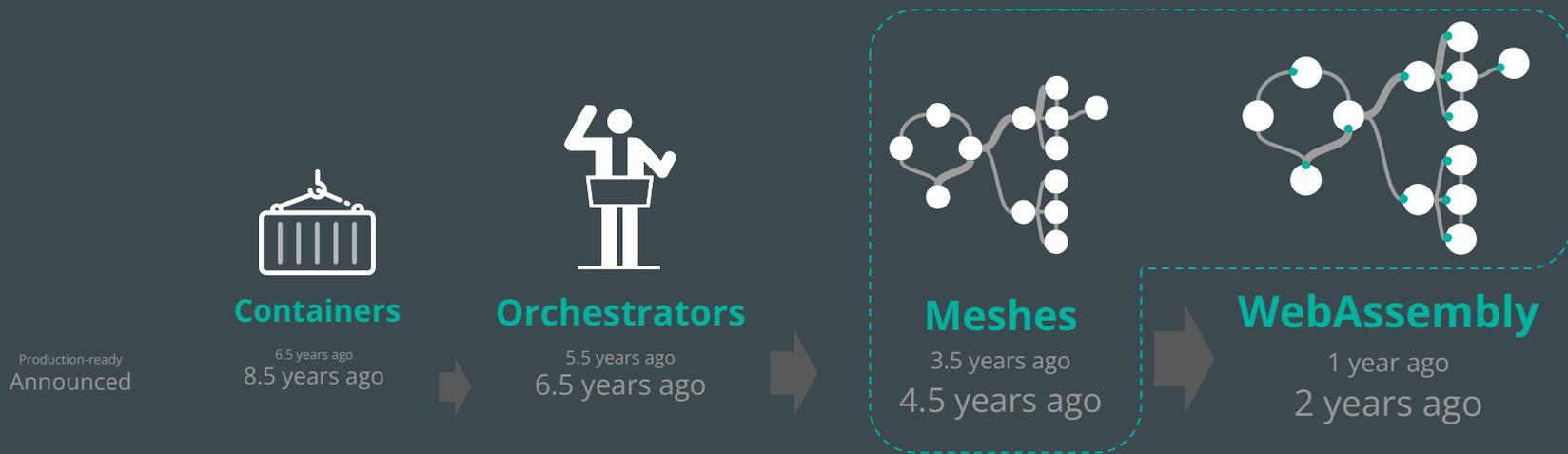
Networking is unfamiliar to developers and operators.



Networking has never been more significant to developers and operators.

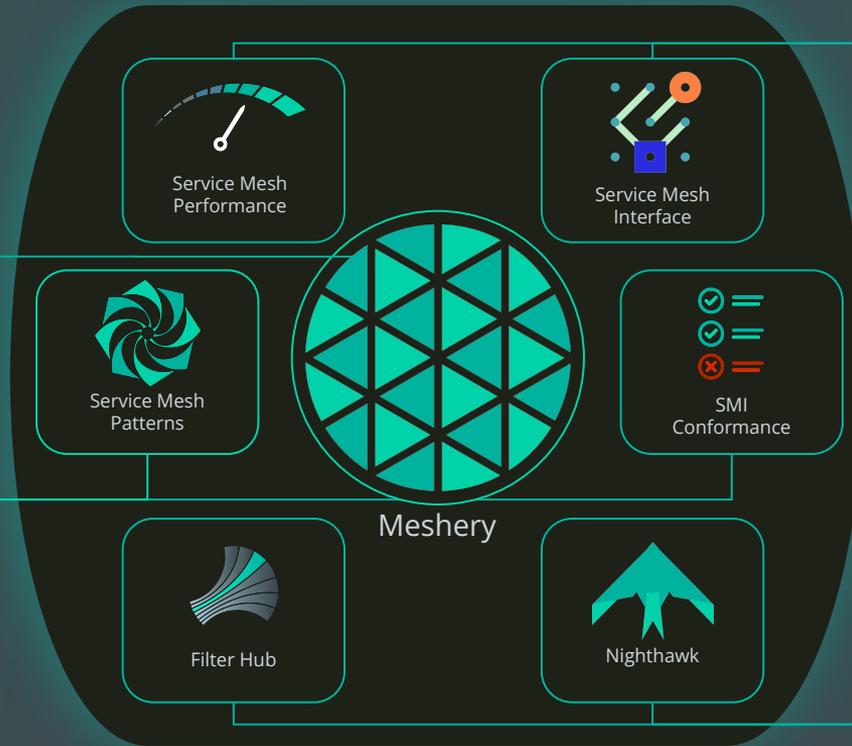


And it's only getting more complex between cloud **and** edge infrastructure



LAYERS

Expect more from your infrastructure



The Only Openly Governed
Service Mesh Manager



Defining Service
Mesh Best Practices

Define and Enforce
Service Mesh Standards



Advanced Analysis and
Service Mesh Intelligence



Community-first

Sustainable open governance, not just open source



Layer5 500+ contributors

all projects

Meshery 300+ contributors

15 maintainers across different organizations:

Layer5, Red Hat, Rackspace, Intel, Quantex, Lumina Networks, VMware, Citrix, Octarine, HashiCorp, Independent, Microsoft, Google



#1 Most Popular Project
in Linux Foundation
Mentorship Program

It's meshy out there

Infrastructure diversity is reality for enterprises



In a multi-mesh world with a landscape of 20 service meshes... let's find your best fit.

<https://layer5.io/landscape>

These factors drive service mesh diversity:

1. Open source governance dictates a world of multiple meshes.
2. Huge range of microservice patterns drives service mesh opportunity.
 - a. Open source projects and vendors create features to serve microservice patterns (they splinter the landscape and function differently).
3. Different organizations need different scopes of service mesh functionality.
4. Hybrid drives infrastructure diversity.
 - a. Accommodate hybrid workloads - non-containerized workloads need to integrate and benefit from your service mesh as well.

Service Mesh Landscape





A Multi-Mesh World



Forrester: Layer5 and Meshery Help Developers Focus On The Business

“ Diverse microservices patterns and technologies, together with the requirements of given microservice applications, provide myriad opportunities for service mesh differentiation and specialization — including meshes native to specific cloud platforms. This will lead to a world where many enterprises use multiple service mesh products, whether separately or together. ”

Source: Forrester, Oct. 2019

Strengths of Service Mesh Implementations



Linkerd

Time to Value,
Performance



Istio

Powerful Feature Set,
Extensibility



Consul

Support for
Non-Kubernetes
Workloads



NGINX

Interoperability with
Existing Ingresses



Network Service Mesh

Layer 2 and Layer 3
Functions

Service mesh standards to the rescue



Meshery implements and advances these standards

Service Mesh Interface (SMI)

A standard **interface** for service meshes on Kubernetes.



Microsoft

Meshery
[the SMI Conformance Tool](#)

Service Mesh Performance (SMP)

A format for describing and capturing service mesh **performance**.



Layer5

Meshery
[an implementation of SMP](#)

Multi-Vendor Service Mesh Interoperation

A set of API standards for enabling service mesh **federation**.

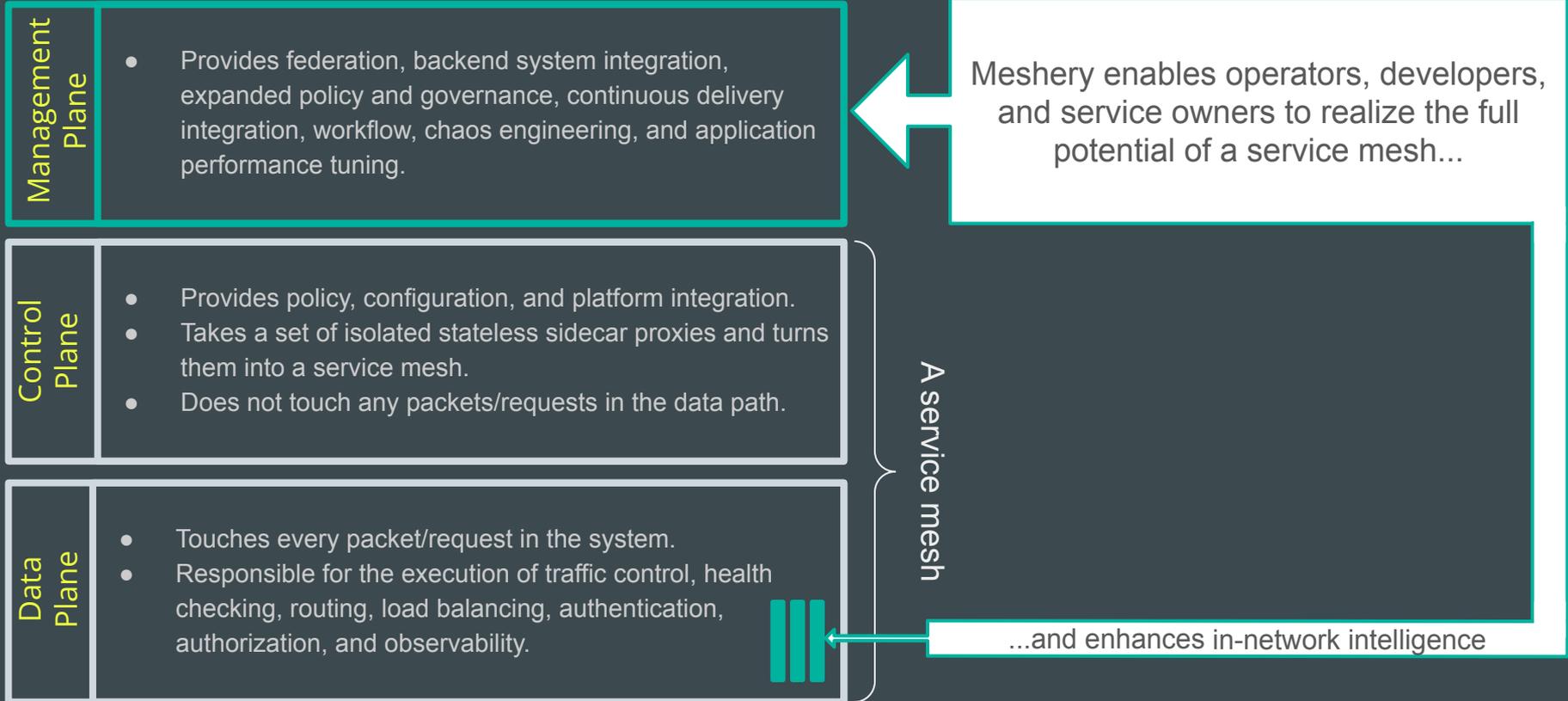


VMware

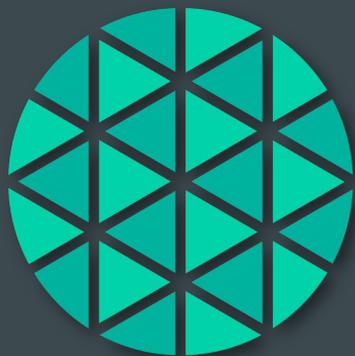
Service Mesh Management



The service mesh management plane



We are the makers of



MESHERY

THE MULTI-MESH MANAGER



SMP



Core Infrastructure
Initiative



CLOUD NATIVE
COMPUTING FOUNDATION



Google
Summer of Code



Service Mesh
Interface (SMI)



COMMUNITYBRIDGE



Google
Season of Docs

The service mesh management plane platform



Multi-Mesh Management

- ✓ Lifecycle
- ✓ Workload
- ✓ Performance
- ✓ Configuration
- ✓ Patterns and Practices
- ✓ Chaos and Filters

Working with each service mesh project to incorporate Meshery into their release process as the measure of their adherence to service mesh standards.

Supports:



The screenshot displays the Meshery Dashboard interface. The left sidebar contains navigation options: Dashboard, Lifecycle, Citrix Service Mesh, Consul, Istio, Kuma, Linkerd, Network Service Mesh, Octarine, Open Service Mesh, Traefik Mesh, Configuration, Applications, Filters, Patterns, Performance, Profiles, Conformance, and Service Mesh Interface. The main content area is divided into three sections:

- Service Mesh:** A table showing the status of various service mesh components across different systems.
- Connection Status:** A section showing the status of various adapters and metrics.
- Metrics:** A section showing the status of various metrics.

Control Plane Pods	Component	Version
istiod-6bf8dd5718-lz8v9	istiod	1.10.3
grafana-56585969cd-pb24g	grafana	7.5.5
istio-ingressgateway-67c99c69bd-z59lc	istio	1.10.3
prometheus-55db9cf9b6-bq4gn	prometheus	v0.5.0

Control Plane Pods	Component	Version
kuma-control-plane-5475f9df6d-ncdv6	kuma	1.2.2

Control Plane Pods	Component	Version
osm-controller-568f88d8f-hz9nw	osm	v0.9.1
osm-injector-5d65b7d67-x6qvh	osm	v0.9.1

Service Mesh Interface Conformance

Meshery, the service mesh compliance tool



The screenshot shows the Meshery web interface with a table titled "SMI Conformance Result". The table has columns for Test, SMI Version, Service Mesh, Service Mesh Version, SMI Specification, Capability, and Test Status. The table contains 8 rows of data, with 4 rows highlighted in light blue. The table is displayed in a modal window over a blurred background of the Meshery interface.

<input type="checkbox"/>	Test	SMI Version	Service Mesh	Service Mesh Version	SMI Specification	Capability	Test Status
<input type="checkbox"/>	TA-01	v1alpha3	Linkerd	edge-20.7.5	Traffic Access	Full	Passed
<input type="checkbox"/>	TA-02	v1alpha3	Linkerd	edge-20.7.5	Traffic Access	Full	Failed
<input type="checkbox"/>	TM-01	v1alpha3	Linkerd	edge-20.7.5	Traffic Metrics	Half	Passed
<input type="checkbox"/>	TM-02	v1alpha3	Linkerd	edge-20.7.5	Traffic Metrics	None	Passed
<input type="checkbox"/>	TM-03	v1alpha3	Maesh	v1.3.2	Traffic Metrics	None	Failed
<input type="checkbox"/>	TM-04	v1alpha3	Maesh	v1.3.2	Traffic Metrics	Full	Passed



Operate and upgrade
with confirmation of
SMI compatibility.

Meshery Functionality

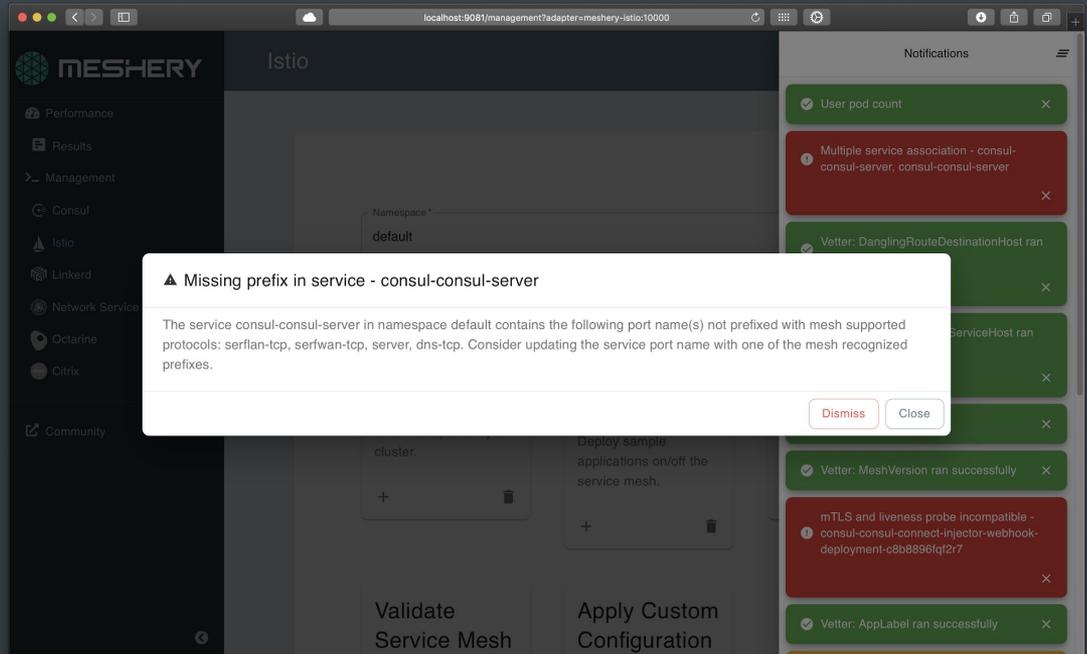
- ✓ Defines compliant behavior.
- ✓ Produces compatibility matrix.
- ✓ Ensures provenance of results.
- ✓ Runs a set of conformance tests.
- ✓ Securely ensures integrity of results.
- ✓ Manages all SMI compatible service meshes.
- ✓ Built into participating service mesh's release pipeline.
- ✓ Common [sample application](#) for validating test assertions.

Configuration Best Practices

Operate with confidence



Assess your service mesh configuration against deployment and operational best practices with Meshery's configuration validator.



Performance Management

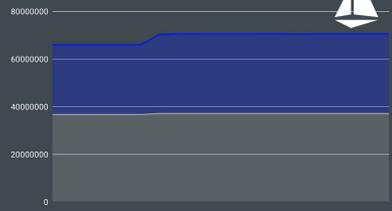
Understand value vs Overhead



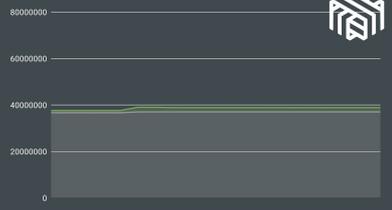
Consul sidecar + app memory usage



Istio sidecar + app memory usage



Linkerd sidecar + app memory usage



MESHERY

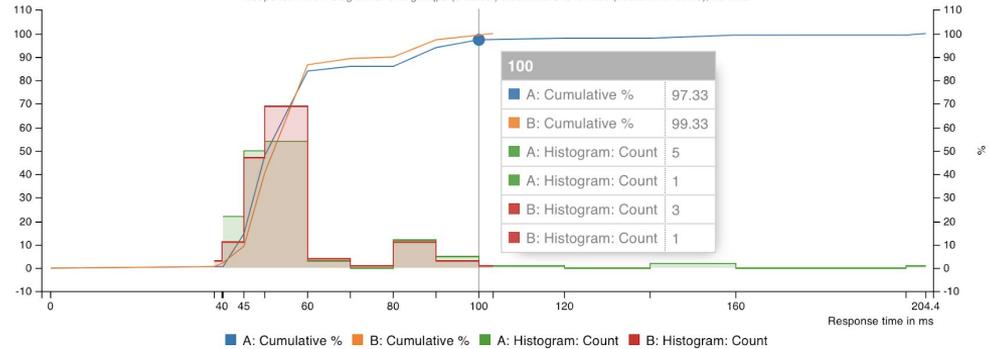
View & Compare Results

- Performance
- Results
- Management
- Consul
 - meshery-consul:1000
- Istio
 - meshery-istio:10000
- Linkerd
 - meshery-linkerd:10000
- Network Service Mesh
 - meshery-nsm:10000
- Octarine
 - meshery-octarine:10000
- Community

Comparison

A: Book Info Product Page performance test 2 - http://10.199.75.64:31380/productpage - 2019-10-11 13:27:28
Response time histogram at 5 target qps (5 actual) 2 connections for 30s (actual time 30.1s), no error

B: Book Info Product Page performance test - http://10.199.75.64:31380/productpage - 2019-10-11 13:16:23
Response time histogram at 5 target qps (5 actual) 2 connections for 30s (actual time 30.1s), no error



Close

> Book Info Product Page performance test Istio Friday, October 11, 2019 1:16 PM 5.0 30.1 0.052 0.103 ...



WebAssembly Filter Management

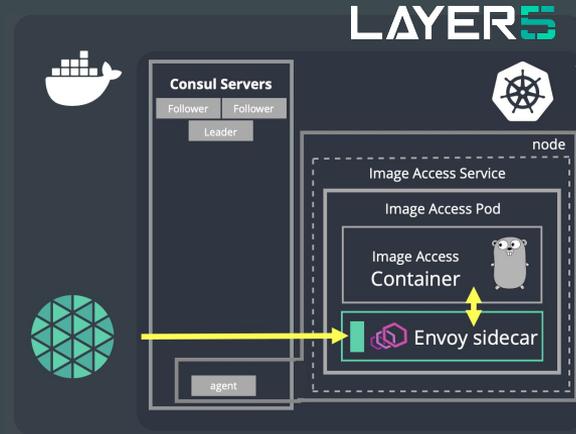
The only WASM manager for any Envoy data plane



INTELLIGENT PERFORMANCE MANAGEMENT

*POLICY-BASED EMBEDDING OF SERVICE
OPTIMIZERS*

- Get your MeshMark and use MeshMark suggestions to make your services and data plane faster.
- Deploy filters modules that optimize your services and data plane automatically.



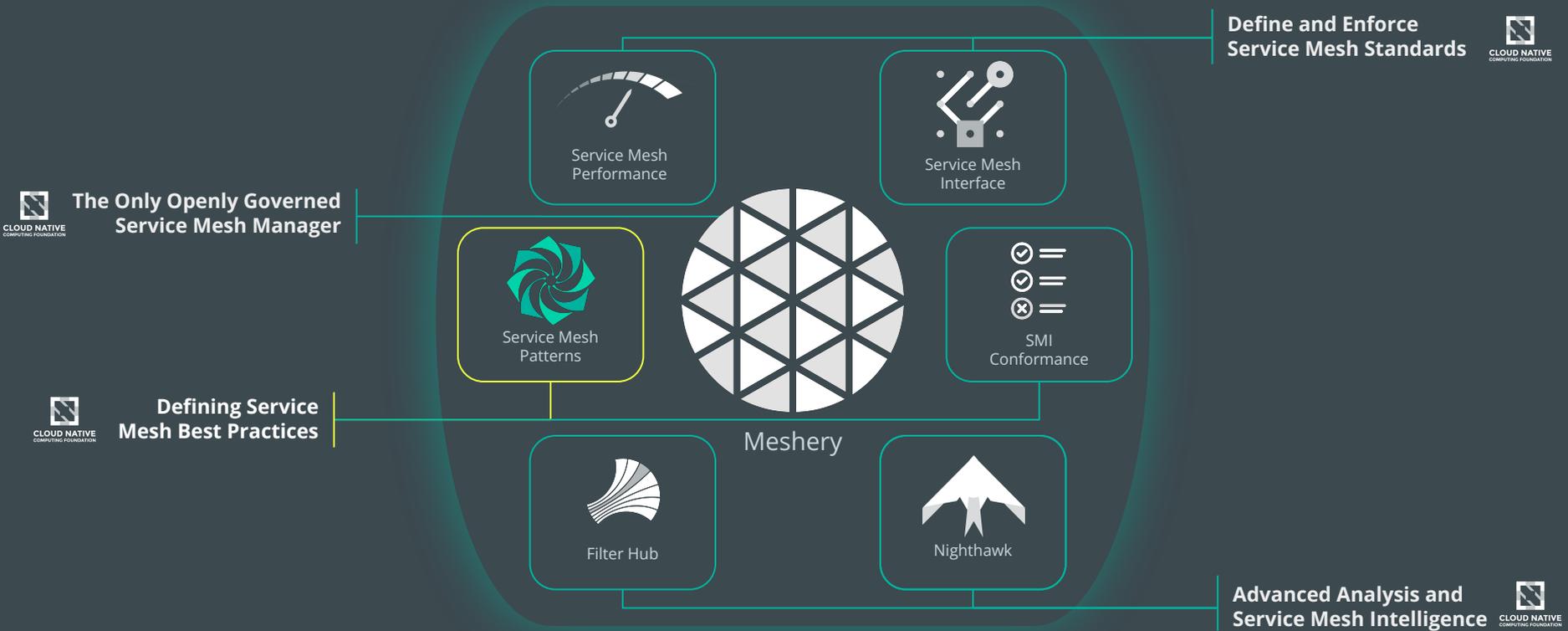
Embedding in-network intelligence to deliver business performance management and automated application performance optimization.



“Layer5 Offers Promising Solution for Cloud Native Networking”

LAYERS

Cloud Native for the rest of us





Service Mesh Patterns

Enabling use of repeatable architectural patterns



Service Mesh Patterns enable the business function in simple language.

- Patterns capture service mesh behavior in a single file and an end-user centric way.

Service Mesh Patterns are service mesh agnostic.

- But, still allow users access service mesh-specific features and differentiation.

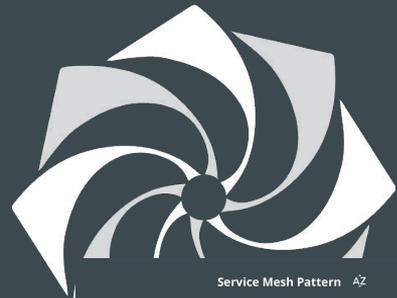
Service Mesh Patterns are reusable.

- Not only are patterns idempotent, but you can easily copy a pattern and modify to suit.

```
name: IstioSM
version: 1.0.1
services:
  istio:
    type: IstioMesh
    namespace: istio-system
    settings:
      version: 1.8.2
    traits:
      mTLS:
        policy: mutual
      namespaces:
        - istio-test
    automaticSidecarInjection:
      namespaces:
        - default
        - istio-test

  grafana:
    type: GrafanaIstioAddon
    namespace: istio-system
    dependsOn:
      - istio
      - prometheus

  prometheus:
    type: PrometheusIstioAddon
    namespace: istio-system
    dependsOn:
      - istio
```



Meshery delivers *Service Mesh Patterns*



Service Mesh Pattern		Category
Area I: It's a Mesh Out There		
A world of multiple service meshes		Foundational
Pattern: How a service mesh empowers an Operator: Retry Budgets		Foundational
Pattern: How a service mesh empowers a service owner		Foundational
Pattern: How a service mesh empowers a Developer		Foundational
Pattern: Employing planes of a service mesh		Foundational
Area II: Patterns of Initialization and Deployment		
Pattern: How to get started with any service mesh; Local Deployment		Deployment
Pattern: Sidecar Proxies		Deployment
Pattern: Node Agents		Deployment
Pattern: Proxyless Service Mesh		Deployment
Pattern: Passive and Active Health Checking		Deployment
Pattern: Workload Onboarding and Service Mesh Adoption		workloads
Pattern: Expanding the Mesh to Brownfield Environments		workloads
Pattern: Segmenting the Monolith (Strangler)		workloads
Area III: Patterns of Configuration		
Pattern: Data plane extensibility		Observability
Pattern: Transparently Proxying TLS		Traffic Management
Pattern: Foundational Traffic Routing		Traffic Management
Pattern: Local and Global Rate Limiting		Traffic Management
Pattern: Timeouts		Traffic Management
Pattern: Retries		Traffic Management
Pattern: Circuit Breaking		Traffic Management
Pattern: Bulkheading with Resiliency		Resiliency

The screenshot shows the Meshery MeshMap interface. On the left is a sidebar with navigation options: Dashboard, Lifecycle, Configuration, Performance, Conformance, and MeshMap. The main area displays the MeshMap configuration for 'MeshMap' (Name: MeshMap, Saved to Meshery Cloud). It includes buttons for 'Save As', 'Verify', and 'Deploy'. The diagram shows a Gateway connected to a Virtual Service, which is connected to a Service. The configuration panels on the left show the following components and versions:

- Application: [Meshery icon]
- Service: [Meshery icon]
- Istio Components: [Dropdown]
- Kuma Components: [Dropdown]
- Versions: 1.3.0 (Versions dropdown) / Version 1.3.0 (dropdown)
- Open Service Mesh Components: [Dropdown]
- Traefik Mesh Components: [Dropdown]
- Linkerd Components: [Dropdown]
- Versions: stable-2.10.2 (Versions dropdown) / Version stable-2.10.2 (dropdown)
- Server: [Meshery icon]
- Server Authorization: [Meshery icon]

Orange arrows point from the 'Retries' and 'Circuit Breaking' patterns in the table to the corresponding components in the MeshMap configuration.

The world's authority on *Service Mesh Patterns*

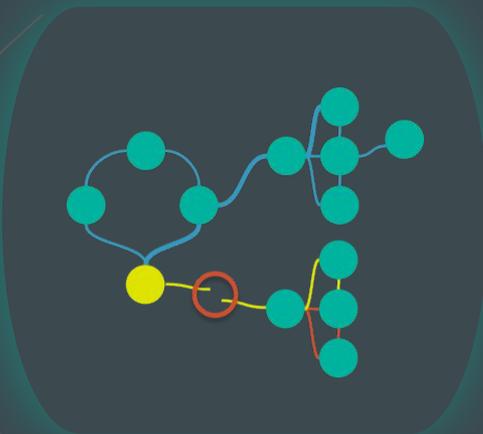


PATTERNS

github.com/service-mesh-patterns



layer5.io/books/service-mesh-patterns



SMP087: Circuit breaker pattern

LAYER 5

Cloud Native for the rest of us



The Only Openly Governed
Service Mesh Manager

CLOUD NATIVE
COMPUTING FOUNDATION



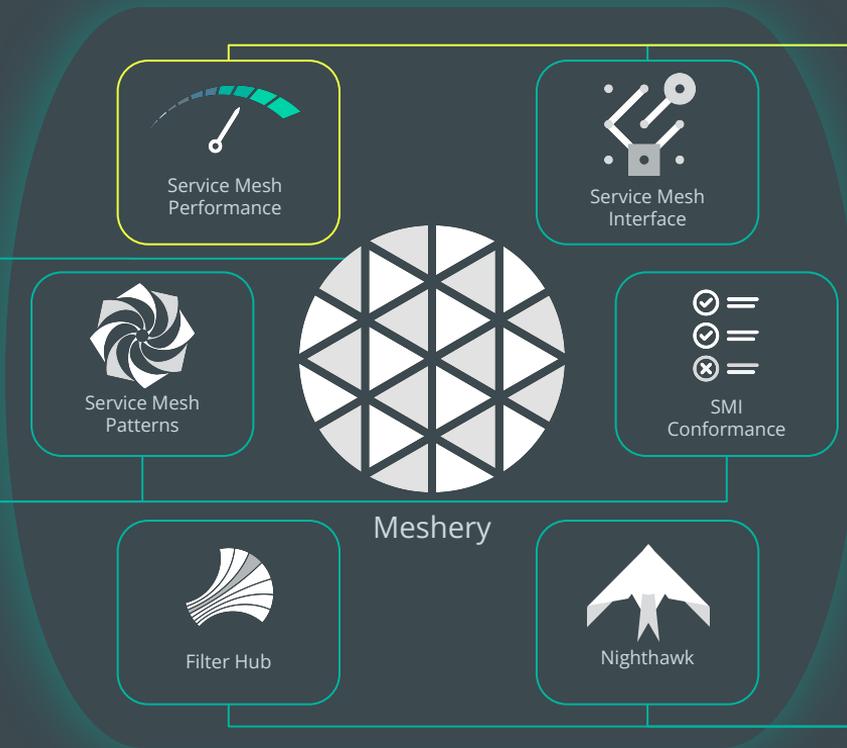
Defining Service
Mesh Best Practices

CLOUD NATIVE
COMPUTING FOUNDATION

Define and Enforce
Service Mesh Standards



CLOUD NATIVE
COMPUTING FOUNDATION



Advanced Analysis and
Service Mesh Intelligence



CLOUD NATIVE
COMPUTING FOUNDATION



Service Mesh Performance

*vendor neutral service mesh performance
measurement standard*



We are the makers of

Service Mesh Performance (SMP)



SMP

smp-spec.io



LAYERS



intel



A cloud native application networking standard.

A vendor neutral specification for capturing details of infrastructure capacity, service mesh configuration, and workload metadata.

Facilitates:

- a universal performance index to gauge a service mesh's efficiency against deployments in other organizations' environments.
- benchmarking of service mesh performance
- exchange of performance information from system-to-system / mesh-to-mesh
- apples-to-apples performance comparisons of service mesh deployments.

MeshMark

from the Service Mesh Performance Specification



An open standard for measuring performance of service meshes in context of the value they provide.

Its purpose is to convert measurements into insights about the value of functions a service mesh is providing.

It does so by specifying a uniform way to analyze and report on the degree to which measured performance provides user value.

Distilling a variety of overhead signals and key performance indicators into a simple scale. Measurement data may not provide a clear and simple picture of how well those applications are performing from a business point of view, a characteristic desired in metrics that are used as key performance indicators.

Reporting several different kinds of data can cause confusion. Reducing measurement data to a single well understood metric is a convenient way to track and report on quality of experience.

LAYERS

Cloud Native for the rest of us



The Only Openly Governed
Service Mesh Manager

CLOUD NATIVE
COMPUTING FOUNDATION



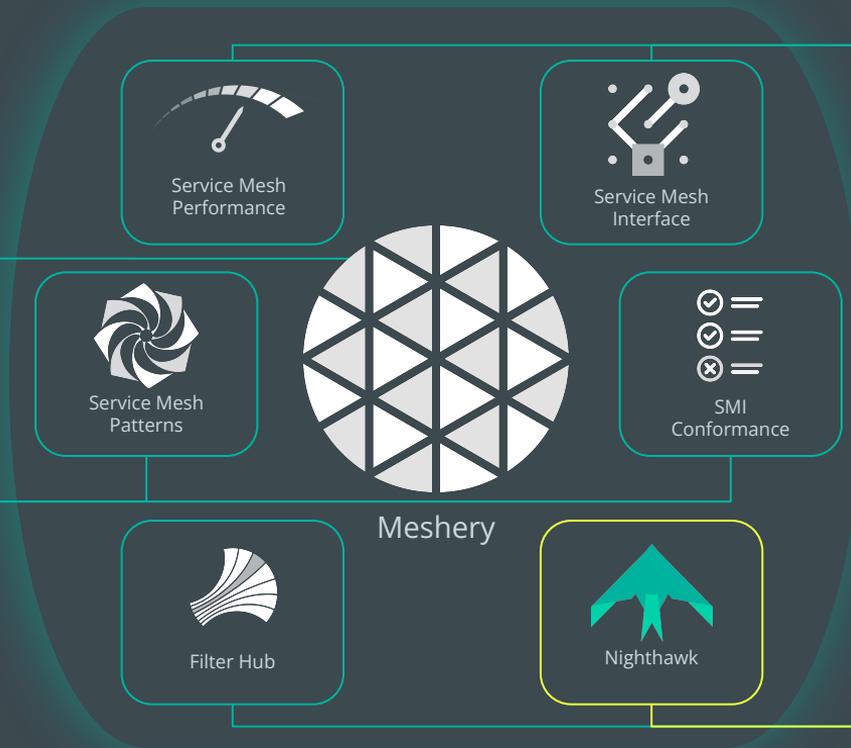
Defining Service
Mesh Best Practices

CLOUD NATIVE
COMPUTING FOUNDATION

Define and Enforce
Service Mesh Standards



CLOUD NATIVE
COMPUTING FOUNDATION



Advanced Analysis and
Service Mesh Intelligence



CLOUD NATIVE
COMPUTING FOUNDATION

We are the makers of *Nighthawk*

Distributed systems require distributed analysis



NIGHTHAWK

getnighthawk.dev

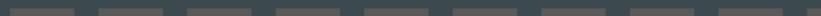
Nighthawk

- a Layer 7 performance characterization tool created by Envoy project.
- a load generator custom-built for data plane proxy testing.



Meshery

- the service mesh management plane
- supports wrk2, fortio, and Nighthawk as single instance load generators.



- **Recursively evaluate optimization algorithms** using adaptive load controllers in Meshery for ongoing insight and automatic tuning.
- Parallelize distributed performance testing with high precision for insight into high tail percentiles. **Unlock distributed systems behavioral analysis.**
- **Model your service mesh topology** and optimize for your ideal configuration in context of how much you value properties of resiliency, performance, throughput, latency, and so on before you deploy to production.



Community Partners

RESEARCH PARTNERS



TECHNOLOGY PARTNERS



Our Service Mesh Training

delivered to 5,000+ students



Day 0 Workshop

What you'll learn -

Kubernetes

- Container orchestration concepts

Kubernetes architecture

- Control plane components

Kubernetes constructs

- Pods, Namespaces, Deployments, StatefulSets, DaemonSets, Services, ConfigMaps, Volumes

Cluster Management

- Monitoring strategies, Best practices
- Upgrades

Workload Management

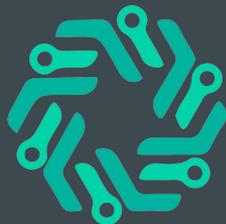
- Rolling Updates
- Continuous Delivery, GitOps

Multi-cluster deployment models

- Identify the best-suited deployment model for your requirements.

Day 1 Workshop

What you'll learn -



MESHMASTER

**Service Mesh Expert
Certification Program**

applications on different service meshes.

Day 2 Workshop

What you'll learn -

Observability

- Methods for managing telemetry, monitoring, and reporting

Traffic Management

- How to manage traffic through load balancing and resilient communication
 - a. Request Routing and Canary Testing
 - b. Fault Injection and Circuit Breaking
- How to enforce policies and rate limiting

Security

- Identity - securing communication with the mesh with identity and mTLS.
- Policy - using traffic policies to operate securely.

Service mesh performance

- Examine and understand the tradeoffs of value delivered vs overhead incurred.

Operational Best Practices

- Running workloads on service mesh
- Troubleshooting the service mesh

Go to Market

LAYER5

THE SERVICE MESH COMPANY



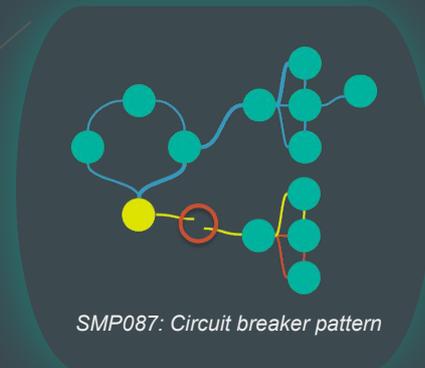
Defining Service Mesh Patterns



CLOUD NATIVE
COMPUTING FOUNDATION

CNCF Service Mesh Working Group

- [Meet](#) on 1st and 3rd Thursday of every month at 11am Pacific.
- Connect: Slack Channel ([#tag-network](#)).
- Join: [Service Mesh WG](#) mailing lists at [lists.cncf.io](#)



layer5.io/books/service-mesh-patterns



MESHERY
THE MULTI-MESH MANAGER

#1 Most Popular Project

in Linux Foundation
Mentorship Program

The screenshot shows the Linux Foundation Mentorship program website. The URL is mentorship.lfx.linuxfoundation.org. The page features a navigation bar with 'OLFX | Mentorship' and a 'Sign In' link. Below the navigation, there are links for 'Enroll a Program', 'Become a Mentor', and 'Become a Mentee'. The main content area is divided into several sections:

- Mentorship Leaderboard:** A large graphic showing 7.3K Applications, 306 Accepted, and 171 Graduated. A central circle indicates \$708.5K Stipends Paid.
- Most Popular Mentors:** A list of mentors with their names and the number of mentees they have:
 - Shuah Khan: 300
 - Greg Kroah-hartman: 115
 - Sridhar Rao: 51
 - Grzegorz Bazior: 48
- Most Popular Programs:** A list of programs with their respective application counts and graduation counts:
 - Meshery: 714 applications, 3 graduates
 - Linux Kernel: Evaluate And Improve...: 352 applications, 1 graduate
 - Linux Kernel Bug Fixing Summer 20...: 241 applications, 0 graduates
 - Linux Kernel Bug Fixing Spring 2021: 184 applications, 6 graduates
- Top 10 Organizational Supporters:** A list of organizations and their support amounts:
 - CLOUD NATIVE COMPUTING FOUNDATION: \$309.6K
 - HYPERLEDGER: \$193.6K
 - OLFNWORKING: \$190K
 - THE LINUX FOUNDATION: \$143K

A yellow arrow points from the '#1 Most Popular Project' text to the 'Meshery' program entry in the 'Most Popular Programs' section.