KubeCon | CloudNativeCon

China 2021

Virtual

Solving the Service Mesh Adopter's Dilemma

Anita Ihuman, Layer5 MeshMate

Agenda

- Introduction
- Getting started with Service Meshes
- Functionality
- Why Adopt Service Meshes
- Service Mesh Architecture
- Service Mesh Abstrations
- The Adopters Dilemma
- Meshery

About Me

Anita Ihuman

Software developer,

MeshMate @ Layer5,

Developer Advocate @ Kyverno,

Technical Writer,

Open Source Advocate.

What are Service Meshes

Service Meshes are simply a way to control how different parts of an application share data with one another.

Partially considered as a microservice platform.

Functionality

Observability

- Metrics without instrumenting apps
- Consistent metrics across fleet
- Trace flow of requests across services
- Portable across metric back-end providers

Security

- Central to service mesh concept is Identity
- Every service gets a unique ID
- These ID are used to facilitate secure connections

Functionality

Traffic Control

(content-based traffic steering)

- Traffic steering
  - Look at the centers to request and route to a specific set of instances
- Traffic splitting
  - L7 tag base routing?
- Ingress and egress routing

Resiliency

(control over chaos)

- Timeouts and Retries with timeout Budget.
- Systematic fault injection
- Control connection pool size and request load
- Circuit breakers and Health checks

Why People Adopt Service Meshes

to avoid..

- Bloated service (application) code
- Duplicating work to make services production-ready
  - Load balancing, auto scaling, rate limiting traffic routing...
- Inconsistency across services
  - Retry, tls, failover, deadlines, cancellation, etc., for each language, framework
  - Siloed implementations lead to fragmented, non-uniform policy application and difficult debugging
- Diffusing responsibility of service management

Why People Adopt Service Meshes

Helps with Modernization

- Modernise your IT inventory without
- Rewriting your application
- Adopting microservices, regular services are fine
- Adopting new frameworks
- Moving to the cloud

Improves Developers' Speed

Service Mesh Architecture

Management Plane

- Provides federation, backend system integration, expanded policy and governance, continuous delivery integration, workflow, chaos engineering, and application performance tuning.

Control Plane

- Provides policy, configuration, and platform integration.
- Takes a set of isolated stateless sidecar proxies and turns them into a service mesh.
- Does not touch any packets/requests in the data path.

Data Plane

- Touches every packet/request in the system.
- Responsible for the execution of traffic control, health checking, routing, load balancing, authentication, authorization, and observability.

A service mesh

Meshery enables operators, developers, and service owners to realize the full potential of a service mesh...

...and enhances in-network intelligence

Service mesh standards to the rescue

Service Mesh Interface (SMI)

A standard interface for service meshes on Kubernetes.

Meshery
the SMI Conformance Tool

Service Mesh Performance (SMP)

A format for describing and capturing service mesh performance.

Meshery
an implementation of SMP

Multi-Vendor Service Mesh Interoperation (Hamlet)

A set of API standards for enabling service mesh federation.

Performance Management

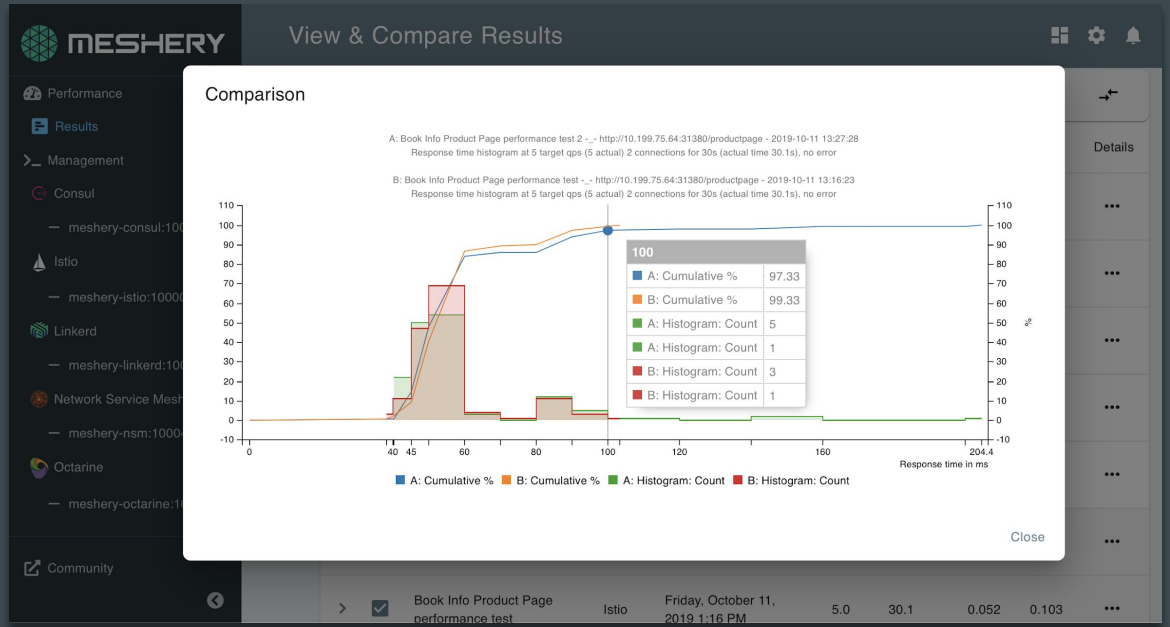
Understand value vs Overhead

SMP

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

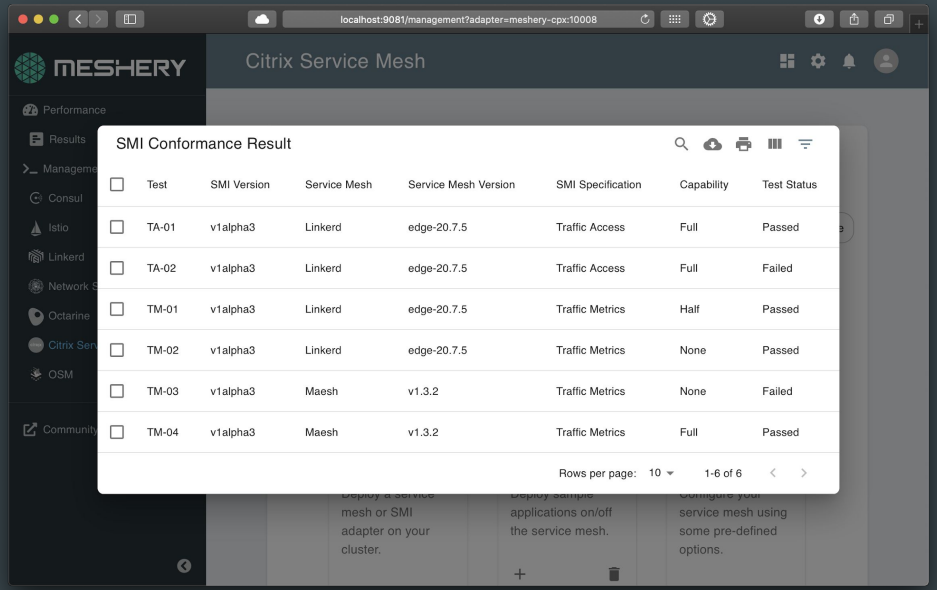
smp-spec.io

CLOUD NATIVE
COMPUTING FOUNDATION



Service Mesh Interface Conformance

Meshery, the service mesh compliance tool



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.

The Adopter’s Dilemma

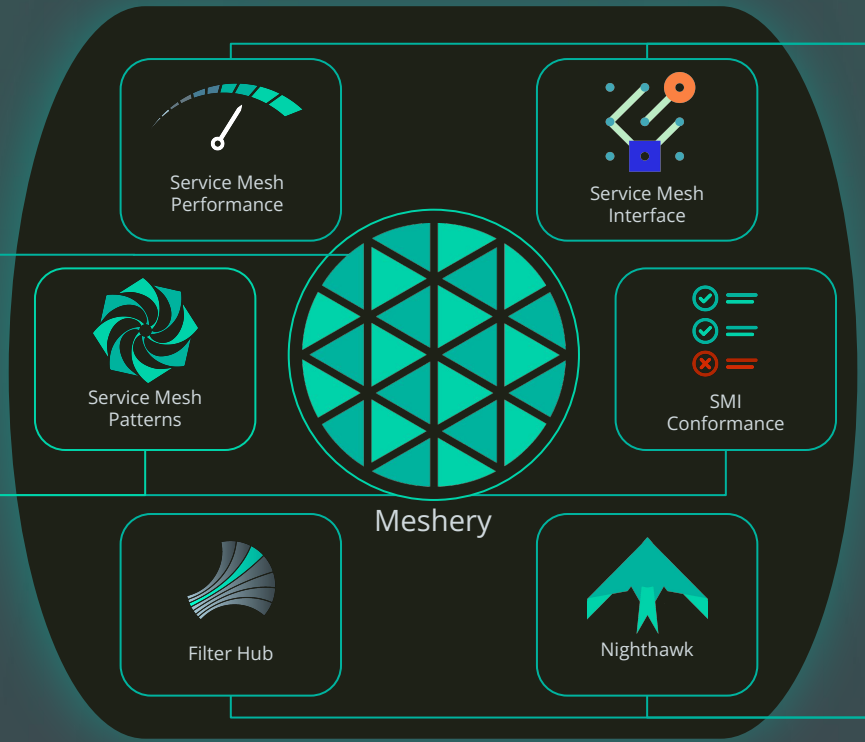
Questions frequently ask by adopters are:

- Which Service Mesh to use?
- How do I get started?
- What is the catch? Nothing is free.
- What overhead does being on the service mesh incur?

Service Mesh Management

LAYER5

Cloud Native Application Networking



Define and Enforce Service Mesh Standards

The Only Openly Governed Service Mesh Manager

Defining Service Mesh Best Practices

Advanced Analysis and Service Mesh Intelligence

MESHERY

THE MULTI-MESH MANAGER

Google
Season of Docs

Core Infrastructure
Initiative

CLOUD NATIVE
COMPUTING FOUNDATION

Google
Summer of Code

SMP

Service Mesh
Interface (SMI)

LFX
Mentorship

The service mesh management plane

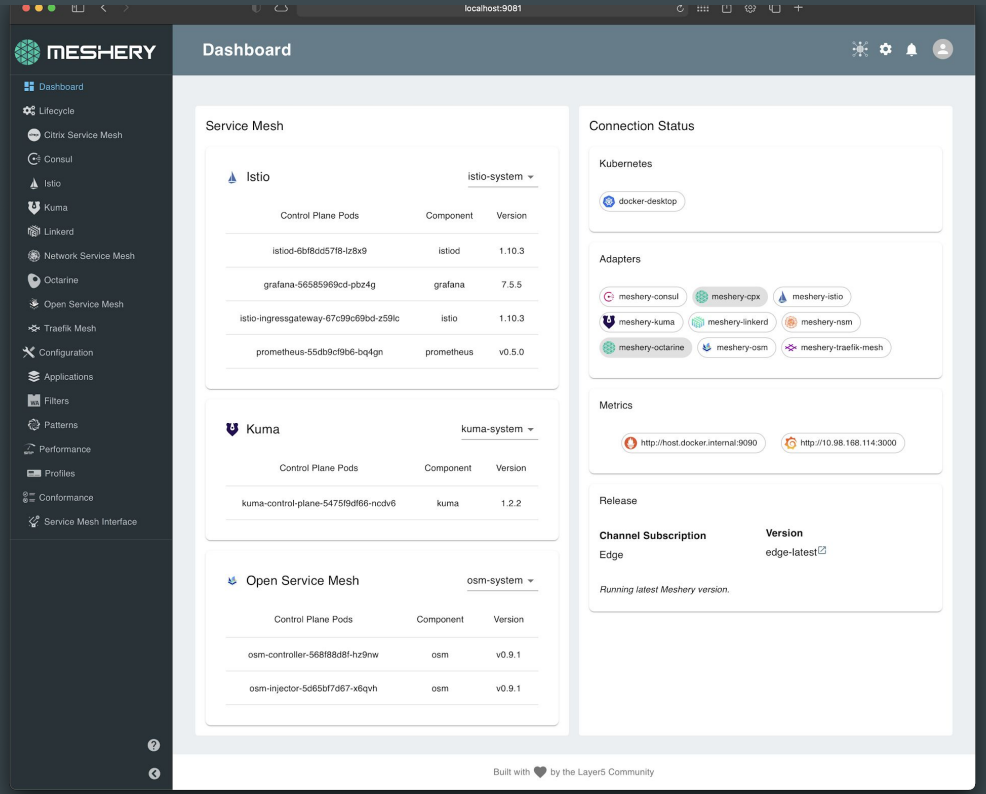
Multi-Mesh Management

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

Supports:

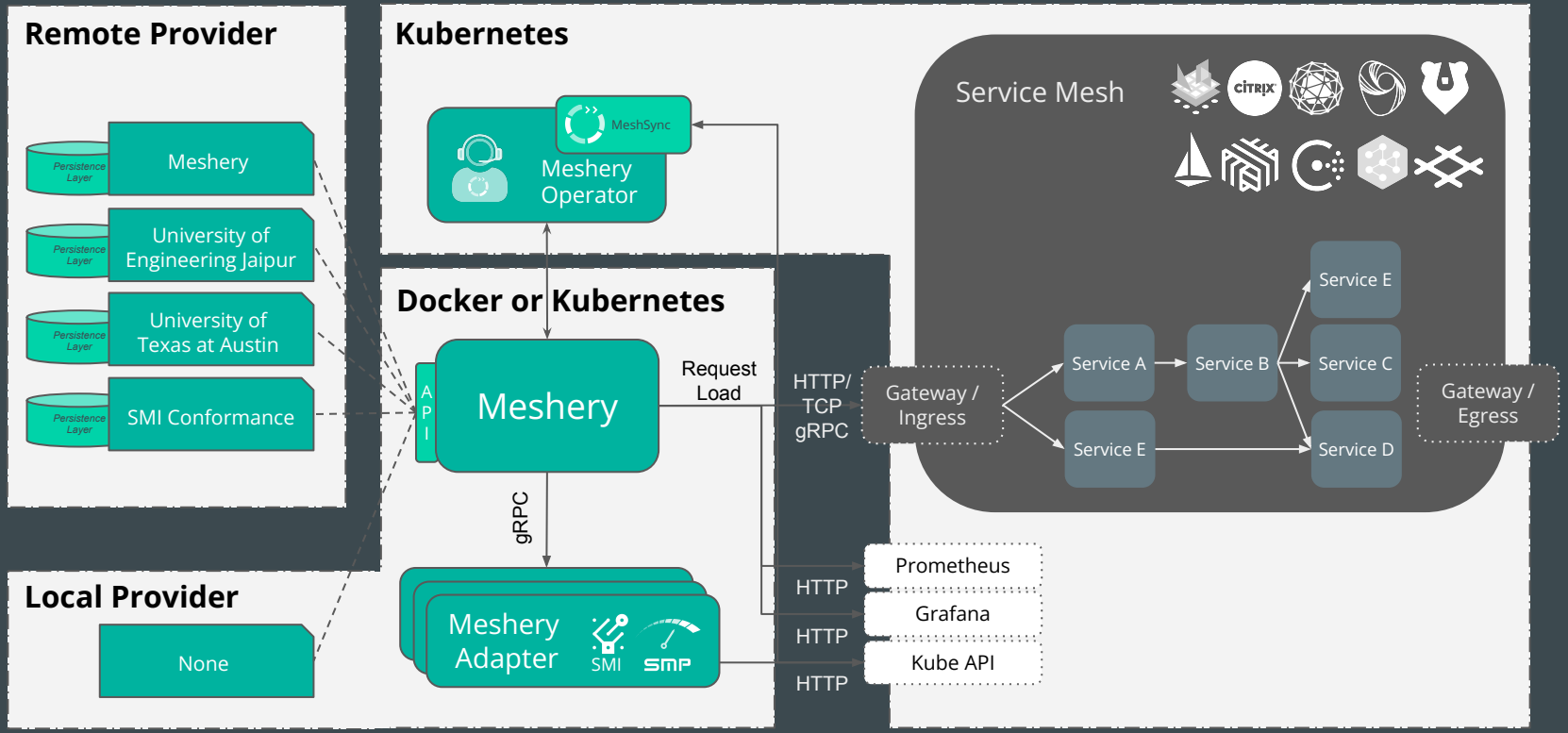
Working with each service mesh project to incorporate Meshery into their release process as the measure of their adherence to service mesh standards.

- Citrix Service Mesh
- Containous Maesh
- HashiCorp Consul
- Istio
- Linkerd*
- Octarine
- Network Service Mesh
- VMware NSX-SM
- AWS App Mesh
- Kong Kuma



Meshery Architecture

Providers



Join the Meshery Project!

A warm and welcoming community

300+ contributors

15 maintainers across different organizations:

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

Statistics

- 1,100+ Meshery users
- 1,050+ Twitter followers
- 1,000+ stars, 100+ releases
- 5,200+ performance tests collected
- 2,200+ Slack community

MESHMATE

Newcomer Onboarding Program

#1 Most Popular Project in Linux Foundation Mentorship Program

Thank You For Listening

You can connect via:

- GitHub: @Anita-ihuman
- Twitter: @Anita_ihuman
- LinkedIn: @anita-ihuman

MESHERY

SMP

CLOUD NATIVE
COMPUTING FOUNDATION

Google
Season of Docs

Core Infrastructure
Initiative

CLOUD NATIVE
Landscape

Google
Summer of Code

Service Mesh
Interface (SMI)

LFX
Mentorship