



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2021

Virtual

Intro and Deep Dive

CNCF TAG Network & CNCF Service Mesh WG

Lee Calcote, Layer5
Ken Owens, Fiserv

CNCF TAG Network Chairs



Lee Calcote

Layer5

@lcalcote



Ken Owens

Fiserv

@kenowens12



Ed Warnicke

Cisco

@edwarnicke

CNCF TAG Network

Introduction

Mission Statement

With an ever steady eye to the needs of workloads and developers who create them and operators who run them, TAG Network's mission is to enable widespread and successful development, deployment and operation of resilient and intelligent network systems in cloud native environments.

In this endeavor, we seek to:

1. Clarify and inform.
2. Collaborate and interrelate.
3. Assist and attract projects.
4. Afford impartial stewardship.

CNCF TAG Network Projects

KubeCon NA 2019

- CNI
- CoreDNS
- Envoy
- gRPC
- Linkerd
- NATS
- Network Service Mesh

KubeCon EU 2020

- BFE
- CNI-Genie
- Contour
- Kuma
- Service Mesh Interface

As of KubeCon NA 2020

- Chaos Mesh
- Open Service Mesh

As of KubeCon EU 2021

- Emissary Ingress
- k8gb

KubeCon NA 2021

- Service Mesh Performance
- Submariner
- Cilium
- Meshery

KubeCon China 2021

- *Proposed: FabEdge*

Working Groups and Papers

Working Groups

- [Universal Data Plane API](#)
- [Service Mesh Working Group](#)

Whitepapers

- [Cloud Native Networking Principles](#) proposed for incorporation into TAG Network.
- Service Mesh Patterns and Reference Implementation
- [Analyzing Service Mesh Performance](#)

Future:

- Techniques of Adaptive Service Mesh Optimization



Service Mesh Working Group

Deep-Dive

Service Mesh Specifications

It's a multi-mesh world

Service Mesh Interface (SMI)

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



Service Mesh Performance (SMP)

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



Multi-Vendor Service Mesh Interoperation (Hamlet)

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

to be determined

Cross-Project Initiatives

Collaborating across projects to achieve common goals

----- Service Mesh Interface Conformance -----



50 Tests

meshery.io/smi

----- Service Mesh Performance -----



5,000 Tests

smp-spec.io

----- Service Mesh Patterns -----



60 Patterns

meshery.io/service-mesh-patterns



Service Mesh Patterns

Enabling use of repeatable architectural patterns



PATTERNS

github.com/service-mesh-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**.
 - User ability to filter on service mesh compatibility..

Service Mesh Patterns are reusable.

- Not only are patterns idempotent, but you can easily copy a pattern and modify to suit.
- Imbued with **best practices**.
- Pattern components can be interchanged and used as building blocks, combining multiple components into a new, integrated pattern.



Service Mesh Pattern Catalog

Enabling use of repeatable architectural patterns



KubeCon



CloudNativeCon

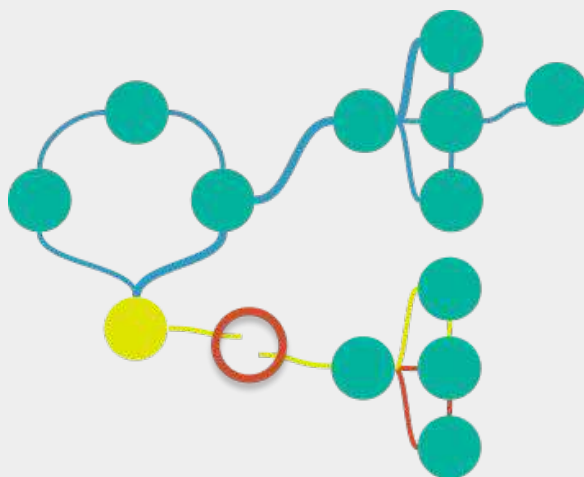


OPEN SOURCE SUMMIT

China 2021

TRAFFIC MGMT

Circuit Breaker



meshery.io/service-mesh-patterns

Authors: [Lee Calcote](#), [Nic Jackson](#)

CNCF Service Mesh WG

Area I: It's a Mesh Out There

Category

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

Service Mesh Interface Conformance

Validating compliance



meshery.io/smi

Purpose and Scope

- Provide an easy-to-use, service mesh and SMI-specific tool to give service mesh projects and users a suite of repeatable conformance tests.
- All service mesh projects participating in the Service Mesh Interface specification.

Project Goals

- Provide users with a compatibility matrix identifying the SMI features that are supported per service mesh.

Project Objectives

- Define a set of conformance tests and what behavior is expected of a conforming service mesh implementation.
- Built into each participating service mesh project's release tooling.

Validating Conformance

- Conformance to SMI specifications will be done through use of a service mesh's workload.
- A sample application is used as the workload to test: [Learn Layer5](#)

Service Mesh Performance

vendor neutral service mesh performance measurement standard



smp-spec.io

Directly enables:

- capturing details of infrastructure capacity, service mesh configuration, and workload metadata.

Facilitates:

- 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.
- a universal performance index to gauge a service mesh's efficiency against deployments in other organizations' environments.



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

Problem:

- 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.

MeshMark:

- Distills a variety of overhead signals and key performance indicators into a simple scale. Reducing measurement data to a single well understood metric is a convenient way to track and report on quality of experience.
- 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.

Distributed Performance Analysis

Distributed systems require distributed analysis



NIGHTHAWK

getnighthawk.dev

Problem:

- Many performance characterizing tools are limited to single instance load generator. This limits the amount of traffic and the variety of behavioral analysis.
- Distributed load testing in parallel poses a challenge when merging results without losing the precision we need to gain insight into the high tail percentiles.
- How to 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.

Nighthawk



Meshery

- a Layer 7 performance characterization tool created by Envoy project.
- a load generator custom-built for data plane proxy testing.
- the service mesh management plane
- supports wrk2, fortio, and Nighthawk as single instance load generators.



Distributed load testing offers insight into system behaviors that arguably more accurately represent real world behaviors of services under load as that load comes from any number of sources.

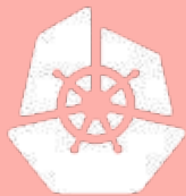
Explore how Nighthawk adaptive load controllers in the service mesh management plane, Meshery, offer ongoing insight and optimization.

Reference: [Design Spec](#), [Issue #72](#)

ENGAGE

CALL FOR PARTICIPATION

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



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2021

Virtual