Microsoft

# Open Service Mesh

**Maheshkumar R**
Cloud Solution Architect – Cloud Native

# Contents

- Service Mesh Overview and Architecture
- Service Mesh Interface (SMI)
- Open Service Mesh (OSM)
- Demos
- Roadmap

# Service Mesh Overview and Architecture

# What is a Service Mesh?

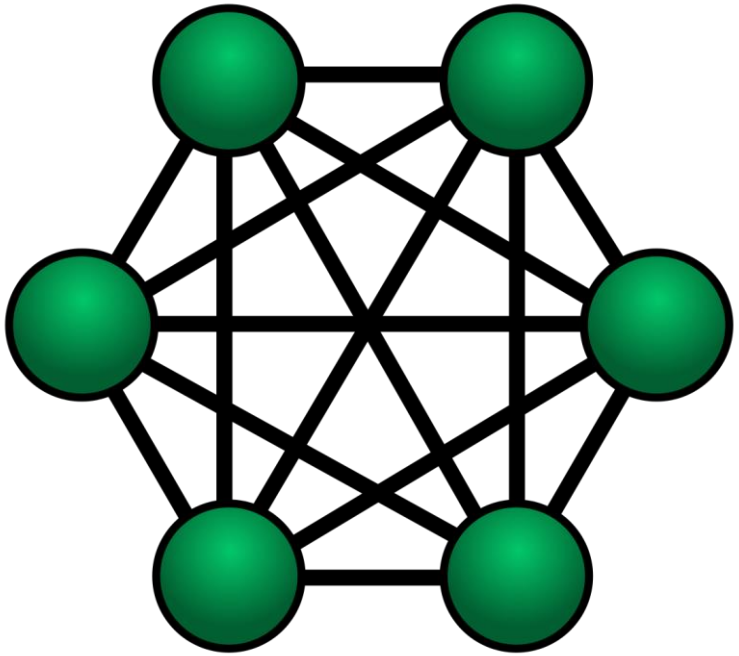
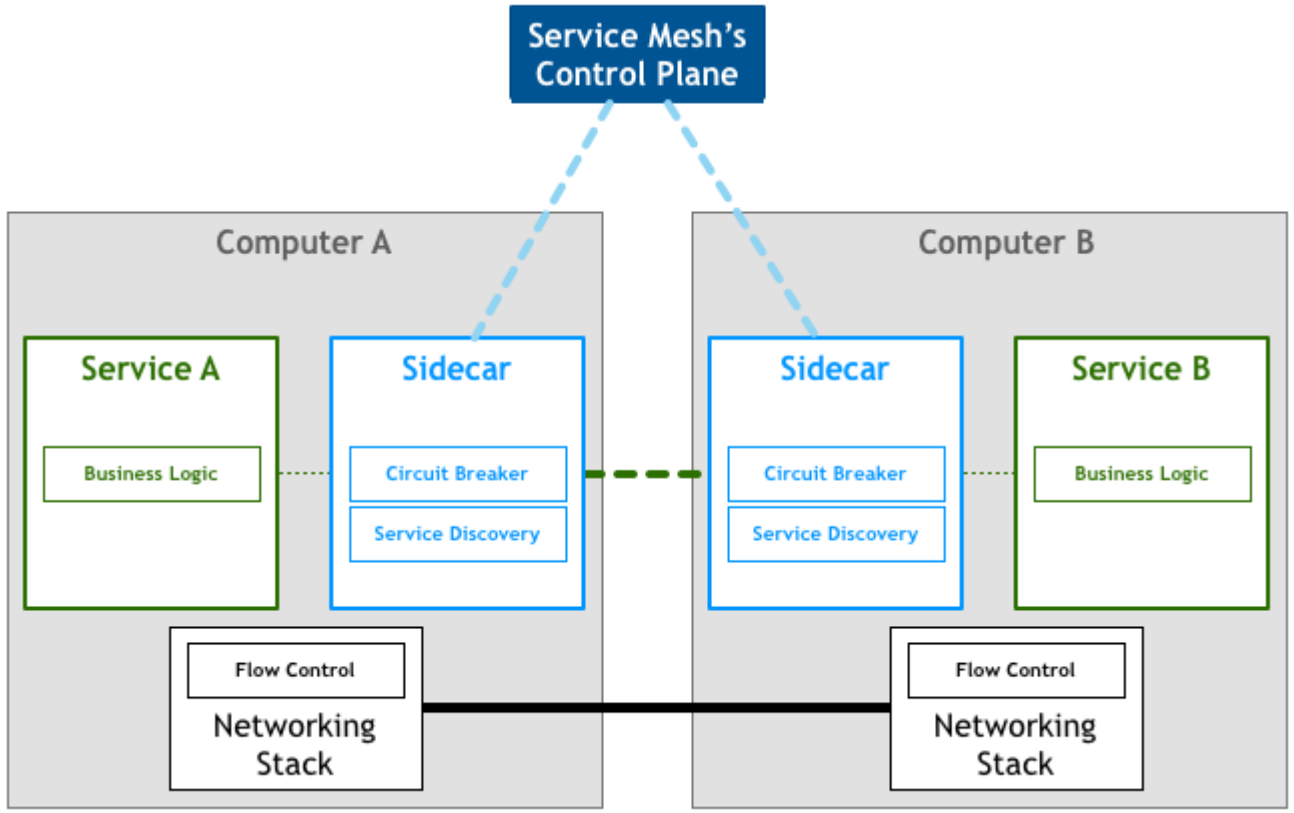
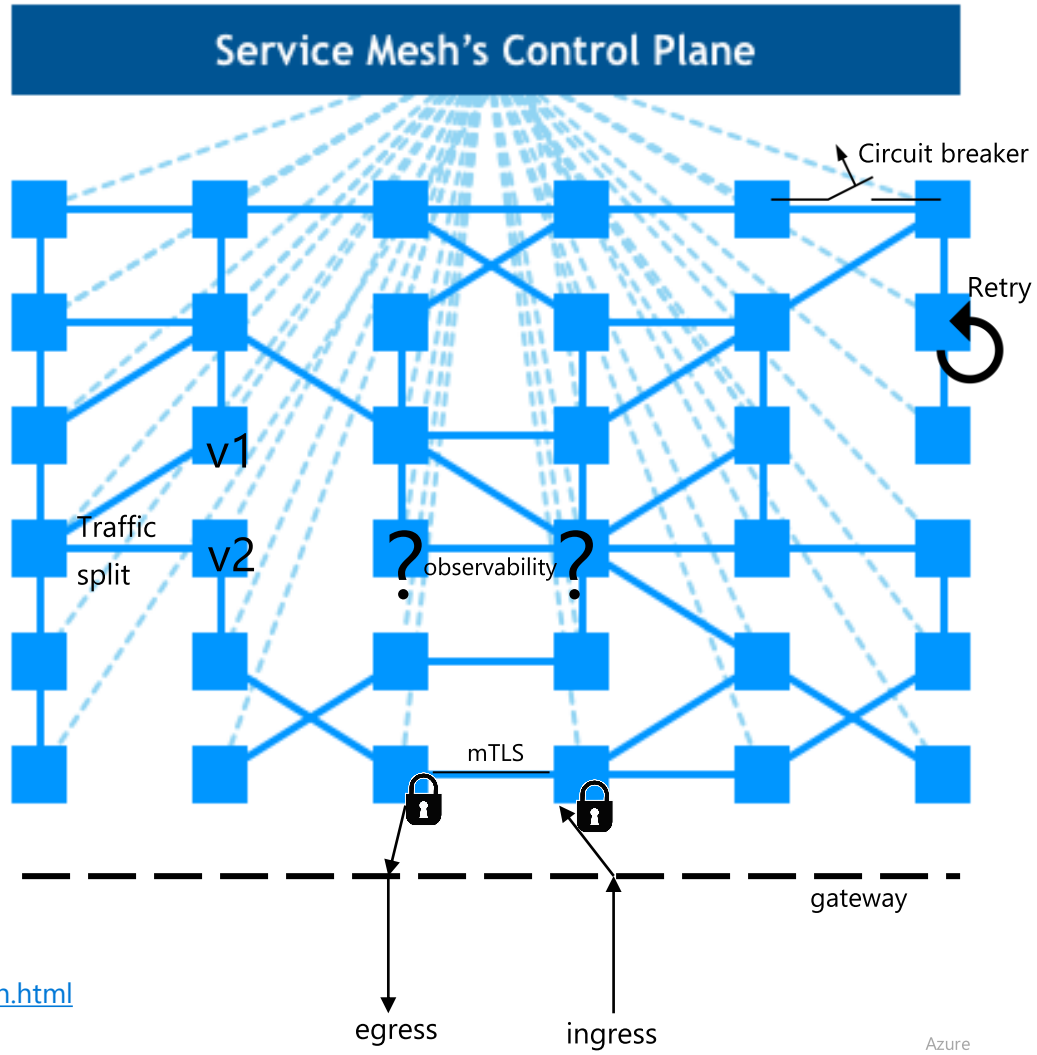


Image credit: https://en.wikipedia.org/wiki/File:Fully-connected_mesh_network.svg

- Emerged as an architectural pattern to support microservices
- Network infrastructure component that allows applications to offload **cross-cutting application concerns**
- Moves away from application-level libraries
- **Top customer requirements addressed:**
  - Zero-trust networks using encryption and access control
  - Traffic management and routing
  - Observability
  - Zero-touch management

# Service Mesh Architecture




Original Image credits:
https://philcalcado.com/2017/08/03/pattern_service_mesh.html

# Service Mesh Considerations

- Do I *really* need a Service Mesh?
- Do I need a mesh that spans clusters?
- Do I need a mesh that spans VMs and Kubernetes?
- Do I need Windows container support?
- Do I need commercial support?
- What are the overheads of operating a Service Mesh?
- Do I want operational policies enforced transparently or do I want developers to adopt the mesh capabilities as needed?

# Service Mesh Landscape

Microsoft 1st Party / OSS

Istio

LINKERD

Consul Connect

Open Service Mesh

Kuma

maesh

...and more

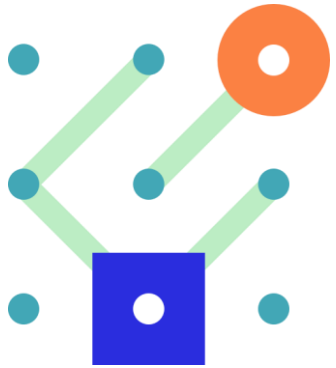
# Service Mesh Interface (SMI)

# Service Mesh Interface (SMI)


- A specification for a standard interface for service meshes on Kubernetes
- Provides a basic feature set for **most common use cases**
- **Kubernetes native** – CRDs, Extension API Servers
- **Provider agnostic** – avoid tight binding to a specific mesh provider
- **Extensible** – SMI APIs will evolve over time
- CNCF sandbox project

https://smi-spec.io/
https://github.com/servicemeshinterface/smi-spec

# SMI Features



**Service Mesh Interface provides:**

- A standard interface for meshes on Kubernetes
- A basic feature set for the most common mesh use cases
- Flexibility to support new mesh capabilities over time
- Space for the ecosystem to innovate with mesh technology

**SMI Covers:**

- Traffic policy and access controls– apply policies like identity and transport encryption across services and restrict which pods or routes are accessible (Layer 7 – HTTP, gRPC, etc.)
- Traffic telemetry – capture key metrics like error rate and latency between services (e.g. golden metrics)
- Traffic management – shift and weight traffic between different services (e.g. canaries, A/B, blue-green)

# Service Mesh Interface (SMI) for Kubernetes

A Kubernetes interface that provides traffic routing, traffic telemetry, and traffic policy

## Standardized

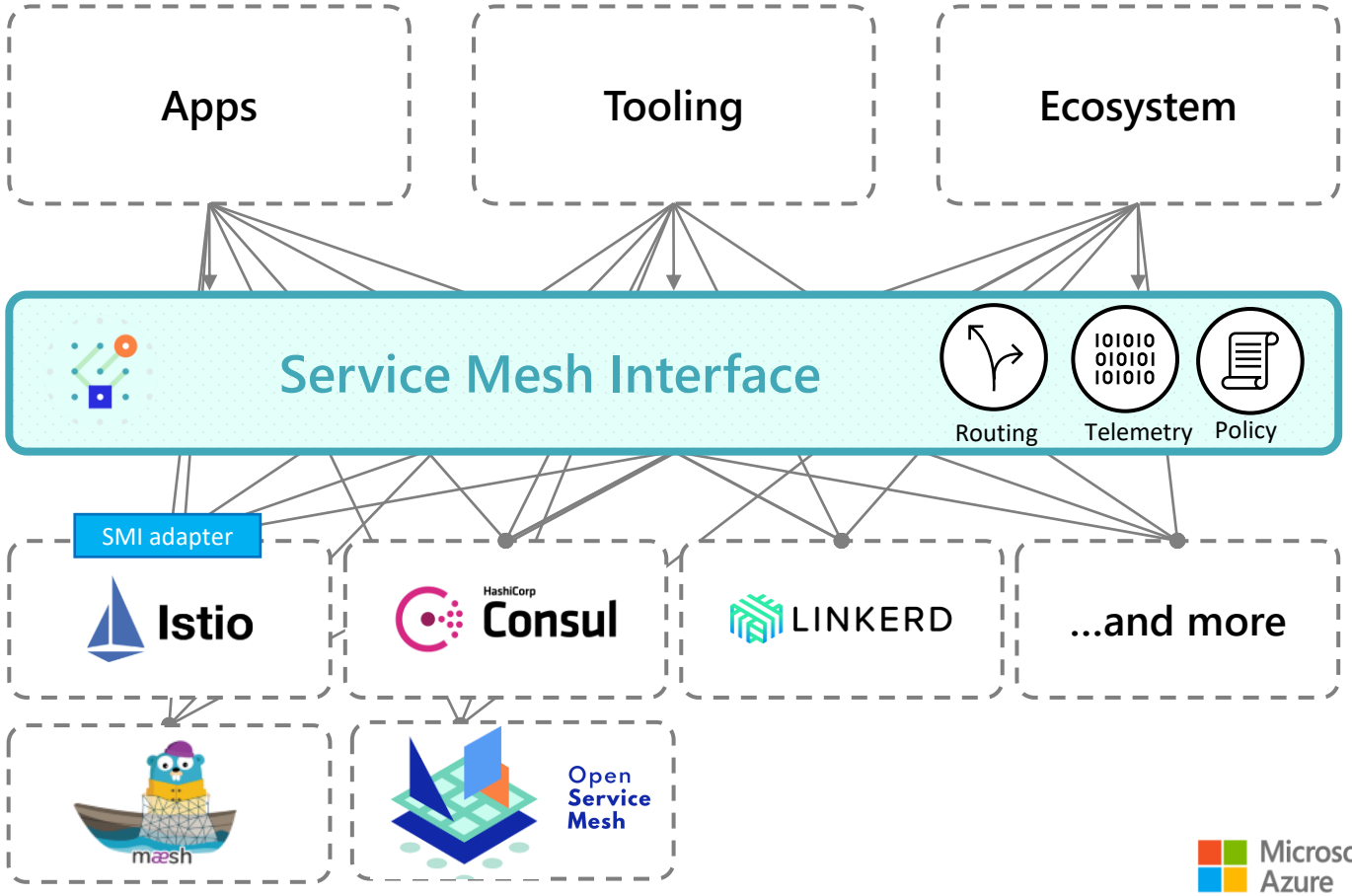
Standard interface for service mesh on Kubernetes

## Simplified

Basic feature set to address most common scenarios

## Extensible

Support for new features as they become widely available




Microsoft Azure

# Service Mesh Interface (SMI) for Kubernetes

In partnership with

Microsoft

LINKERD

HashiCorp

solo.io

Red Hat

RANCHER

docker

weaveworks

ASPEN MESH

Pivotal

vmware

LAYER5

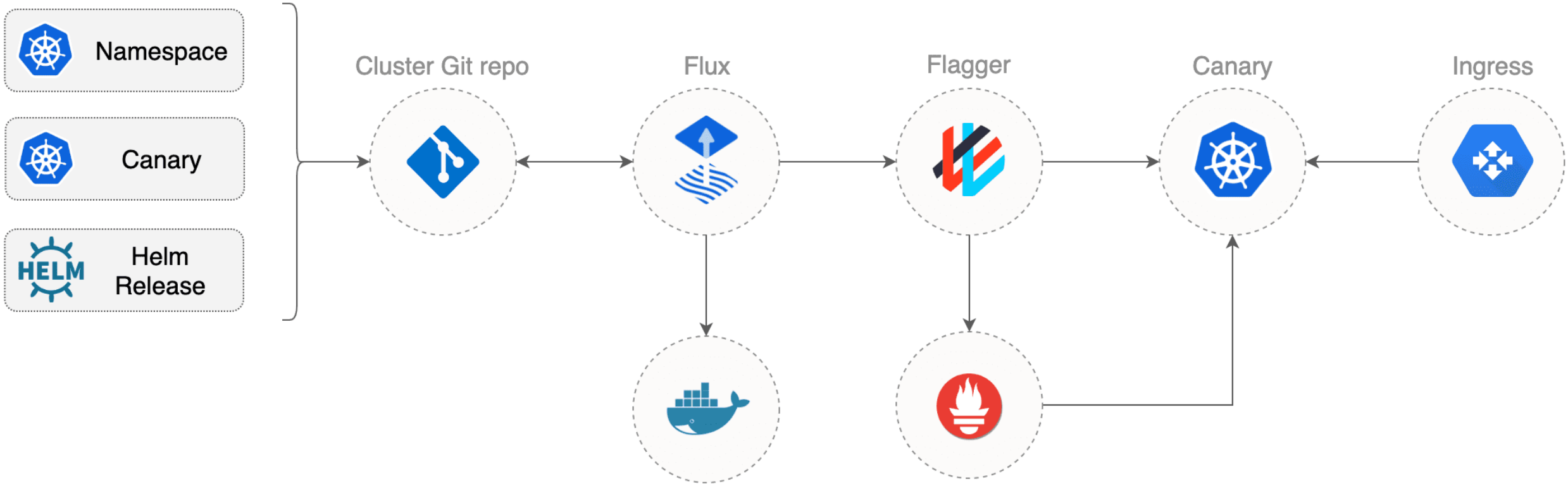
CANONICAL

kubecost

...

# SMI ecosystem example - Flagger

Progressive Delivery Operator for Kubernetes




**Canary** (progressive traffic shifting)

**A/B Testing** (HTTP headers and cookies traffic routing)

**Blue/Green** (traffic switching and mirroring)

https://flagger.app/

# Open Service Mesh (OSM)

# Open Service Mesh

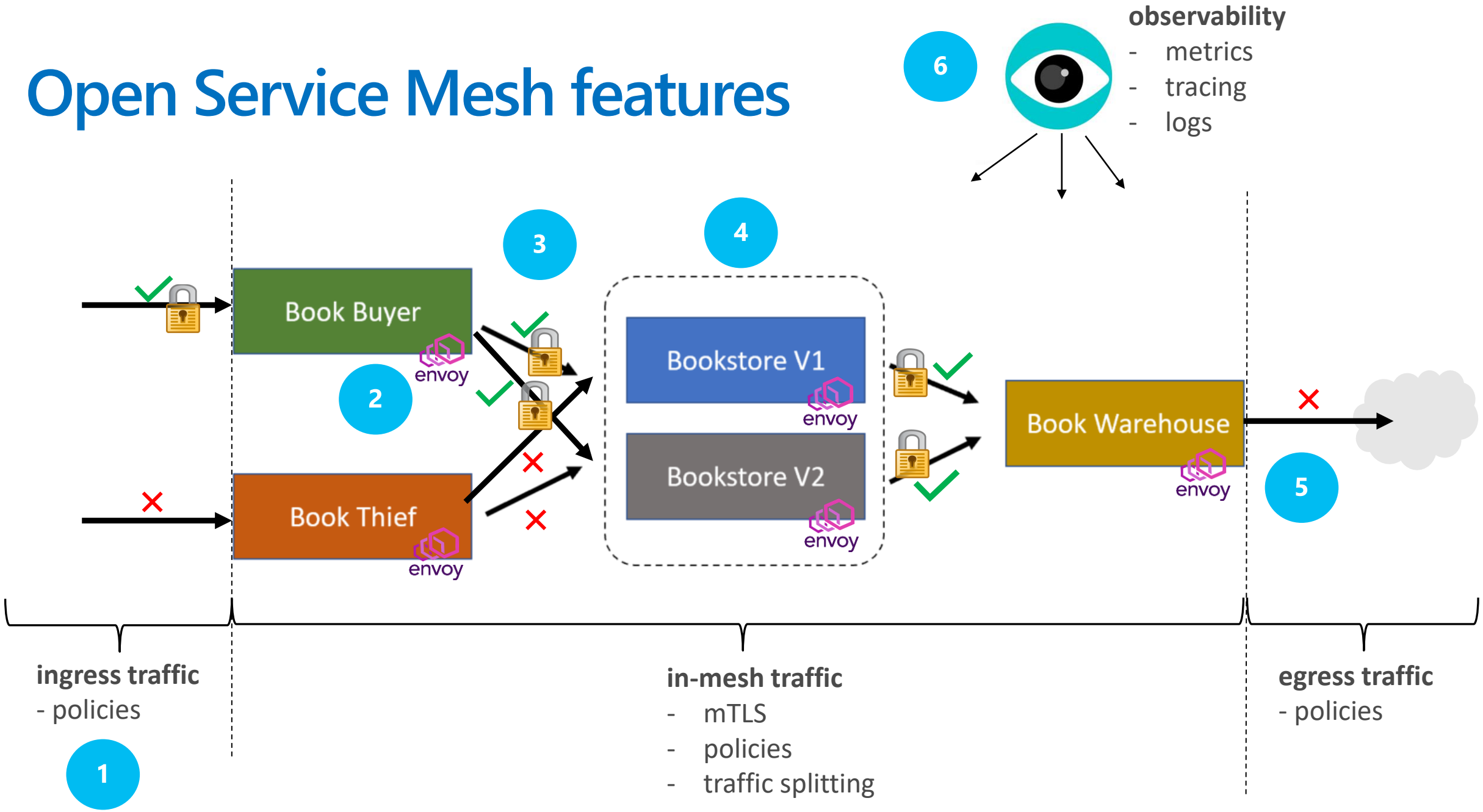
CLOUD NATIVE
COMPUTING FOUNDATION

Open Service Mesh is a Cloud Native Computing Foundation sandbox project.

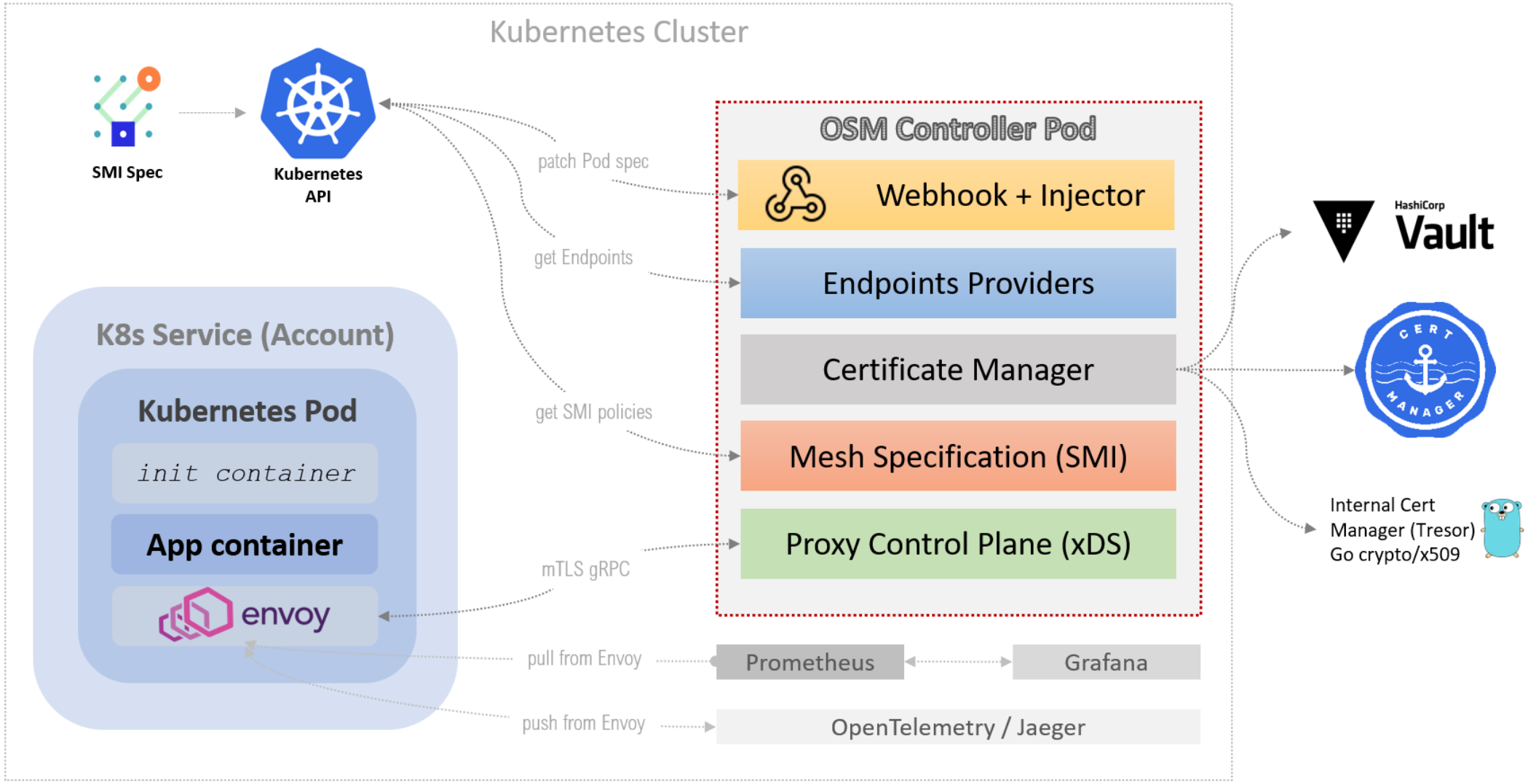
https://openservicemesh.io/

- Open Service Mesh (OSM) is a lightweight and extensible cloud native service mesh
- Uses the CNCF Envoy proxy
- Implements Service Mesh Interface (SMI)
- Created by Microsoft and donated to CNCF
- Features
  - Traffic shifting
  - mTLS
  - Supports external certificate management solutions
  - Observability via application metrics for debug/monitoring
  - Access control policies
  - Automatic sidecar injection

# Open Service Mesh features



# Open Service Mesh – Components and Interactions




https://docs.openservicemesh.io/docs/design_concepts/osm_components_interactions/

# Open Service Mesh – mTLS Support

- **Mutual TLS**: Yes (pod-to-pod encryption)
- **Status**: v1.0.0 – OSS upstream (self-installed)
  v1.0.0 – AKS add-on
- **Mechanism:** Sidecar proxy (Envoy)
- **OSI stack**: Layer 7
- **Supporting features:** Access control policies
- **Installation:**
  - Helm Chart (self-managed), AKS add-on (managed)
- **Support:**
  - Microsoft (when used in AKS or Azure Arc), CNCF and GitHub community

# Envoy resource requirements and latency

- OSM sidecars use Envoy proxy
  - Envoy proxy uses 0.35 vCPU and 40 MB memory per 1000 requests per second going through the proxy [1]
  - Envoy proxy adds 2.65 ms to the 90th percentile latency [1]

Source: Envoy proxy resource overheads [1]

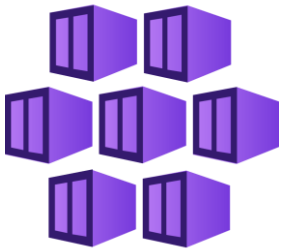
# Open Service Mesh v1.0.0 – resource requests and limits

| Key | Type | Default | Description |
|---|---|---|---|
| osm.injector.resource | object | { "limits":{"cpu":"0.5","memory":"64M"}, "requests":{"cpu":"0.3","memory":"64M"} } | Sidecar injector's container resource parameters |
| osm.osmBootstrap.resource | object | { "limits":{"cpu":"0.5","memory":"128M"}, "requests":{"cpu":"0.3","memory":"128M"} } | OSM bootstrap's container resource parameters |
| osm.osmController.resource | object | { "limits":{"cpu":"1.5","memory":"1G"}, "requests":{"cpu":"0.5","memory":"128M"} } | OSM controller's container resource parameters. See https://docs.openservicemesh.io/docs/guides/ha_scale/scale/ for more details. |
| osm.prometheus.resources | object | { "limits":{"cpu":"1","memory":"2G"}, "requests":{"cpu":"0.5","memory":"512M"} } | Prometheus's container resource parameters |

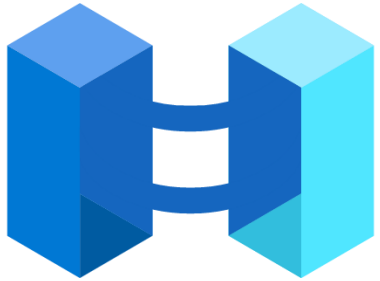
Source: https://release-v1-0.docs.openservicemesh.io/docs/overview/osm_resource_limits/

# Managed Open Service Mesh (OSM)

- Fully Managed and Supported by Microsoft
- Future support for Windows nodes
- **Azure Kubernetes Service**
  - Open Service Mesh Add-on (GA)
- **Azure Arc enabled Kubernetes**
  - Open Service Mesh Extension (public preview)



Azure Kubernetes Service



Azure Arc enabled Kubernetes

# Open Service Mesh (OSM) with AKS Add-on

Azure / AKS

Filter by title

- Configure data volumes
- Monitoring and logging
- Use Windows Server containers
- Develop and run applications
- Deploy the Open Service Mesh AKS add-on
  - About Open Service Mesh
  - Use the Azure CLI
  - Use Bicep template
  - Install the OSM CLI
  - Manage a new application
  - Onboard existing applications
  - Configure IP address and port range exclusion
  - Configure Azure Monitor
  - Integrate with Azure Application Gateway Ingress
  - Integrate with NGINX Ingress Controller
  - Troubleshoot Open Service Mesh
  - Uninstall the Open Service Mesh AKS add-on
- Use cluster extensions (preview)
- DevOps
- Troubleshoot
- Reference
- Resources

Download PDF Retiring

## Open Service Mesh AKS add-on

11/20/2021 • 2 minutes to read • Is this page helpful?

Open Service Mesh (OSM) is a lightweight, extensible, Cloud Native service mesh that allows users to uniformly manage, secure, and get out-of-the-box observability features for highly dynamic microservice environments.

OSM runs an Envoy-based control plane on Kubernetes, can be configured with SMI APIs, and works by injecting an Envoy proxy as a sidecar container next to each instance of your application. The Envoy proxy contains and executes rules around access control policies, implements routing configuration, and captures metrics. The control plane continually configures proxies to ensure policies and routing rules are up to date and ensures proxies are healthy.

The OSM project was originated by Microsoft and has since been donated and is governed by the Cloud Native Computing Foundation (CNCF). The OSM open source project, will continue to be a community led collaboration around features and functionality and contributions to the project are welcomed and encouraged. Please see our Contributor Ladder guide on how you can get involved.

### Capabilities and features

OSM provides the following set of capabilities and features to provide a cloud native service mesh for your Azure Kubernetes Service (AKS) clusters:

- OSM has been integrated into the AKS service to provide a fully supported and managed service mesh experience with the convenience of the AKS feature add-on
- Secure service to service communication by enabling mTLS
- Easily onboard applications onto the mesh by enabling automatic sidecar injection of Envoy proxy
- Easily and transparent configurations for traffic shifting on deployments



https://docs.microsoft.com/en-us/azure/aks/open-service-mesh-about

# Open Service Mesh via Azure Arc extension

Save Fe

## Azure Arc-enabled Open Service Mesh (Preview)

07/23/2021 • 12 minutes to read •

Open Service Mesh (OSM) is a lightweight, extensible, Cloud Native service mesh that allows users to uniformly manage, secure, and get out-of-the-box observability features for highly dynamic microservice environments.

OSM runs an Envoy-based control plane on Kubernetes, can be configured with SMI APIs, and works by injecting an Envoy proxy as a sidecar container next to each instance of your application. Read more on the service mesh scenarios enabled by Open Service Mesh.

### Support limitations for Arc enabled Open Service Mesh

- Only one instance of Open Service Mesh can be deployed on an Arc connected Kubernetes cluster
- Public preview is available for Open Service Mesh version v0.8.4 and above. Find out the latest version of the release here. The supported release versions are appended with notes. Ignore the tags associated with intermediate releases.
- Following Kubernetes distributions are currently supported
  - AKS Engine
  - AKS on HCI
  - Cluster API Azure
  - Google Kubernetes Engine
  - Canonical Kubernetes Distribution
  - Rancher Kubernetes Engine
  - OpenShift Kubernetes Distribution
  - Amazon Elastic Kubernetes Service

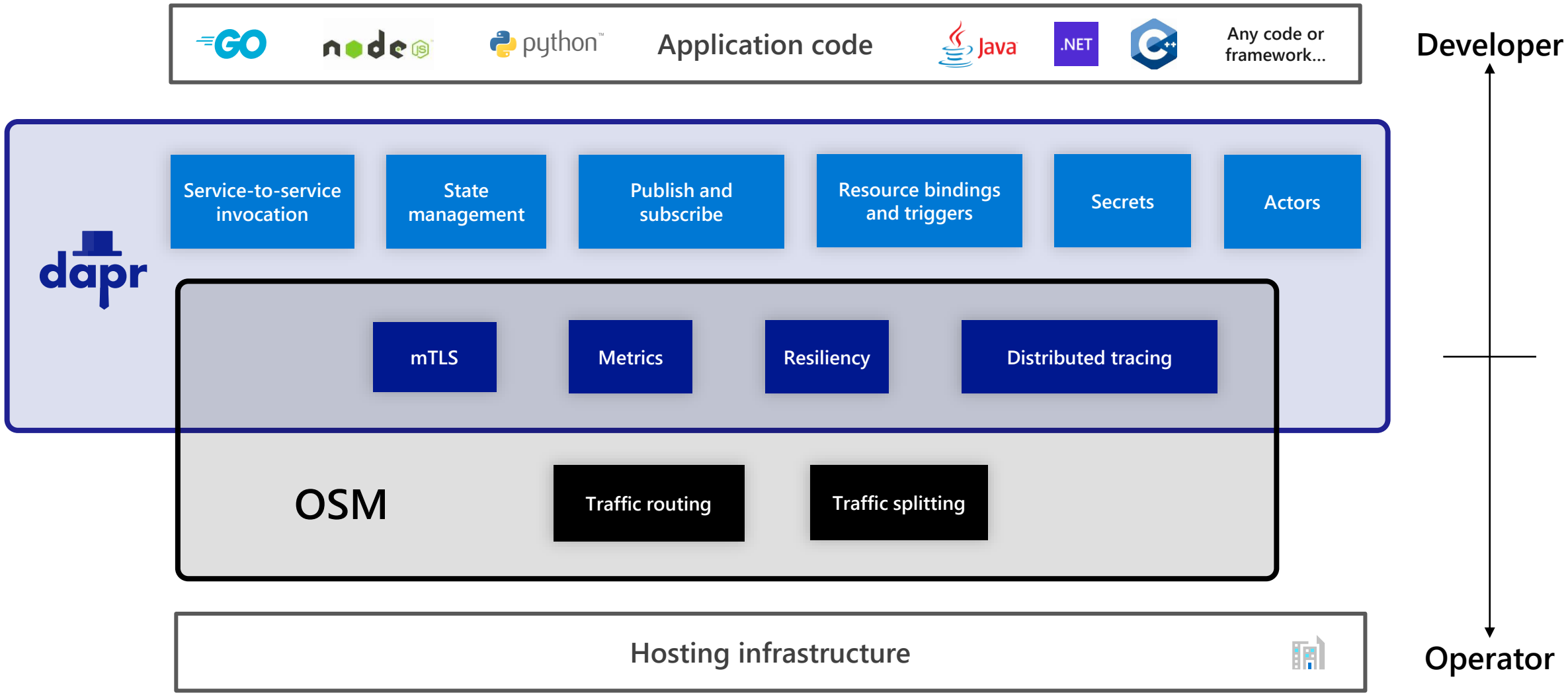


https://docs.microsoft.com/en-us/azure/azure-arc/kubernetes/tutorial-arc-enabled-open-service-mesh

# Open Service Mesh (OSM) comparison

| Feature | AKS/Arc-enabled Kubernetes add-on | OSM self-installed OSS |
|---|---|---|
| Installation | **AKS add-on**<br>**Arc-enabled Kubernetes extension**<br>Installs to `kube-system` namespace | **Helm chart**<br>**OSM CLI** (`osm install`)<br>Installs to `osm-system` namespace |
| Support | **Microsoft supported**<br>Raise Azure Support ticket | **Community supported**<br>GitHub issues |
| OSM Dashboard | **No** | **Yes** |
| Features | **All core features:** SMI, mTLS, traffic policy, egress policy, traffic split, certificate rotation, observability (metrics, tracing, logs), Global IP address/port exclusions | **All core features:** SMI, mTLS, traffic policy, egress policy, traffic split, certificate rotation, observability (metrics, tracing, logs), Global IP address/port exclusions |
| Certificate Authority | **Tresor** (self-signed CA) | **Tresor** (self-signed CA)<br>**Hashicorp Vault**<br>**Cert-Manager** (Let's Encrypt, HashiCorp Vault, Venafi and private PKI) |

# OSM vs Dapr


GO
node
python
Application code
Java
.NET
Any code or framework...
Developer
dapr
Service-to-service invocation
State management
Publish and subscribe
Resource bindings and triggers
Secrets
Actors
mTLS
Metrics
Resiliency
Distributed tracing
OSM
Traffic routing
Traffic splitting
Hosting infrastructure
Operator

# Demo 1:

# Open Service Mesh – Bookstore

OSM AKS add-on version

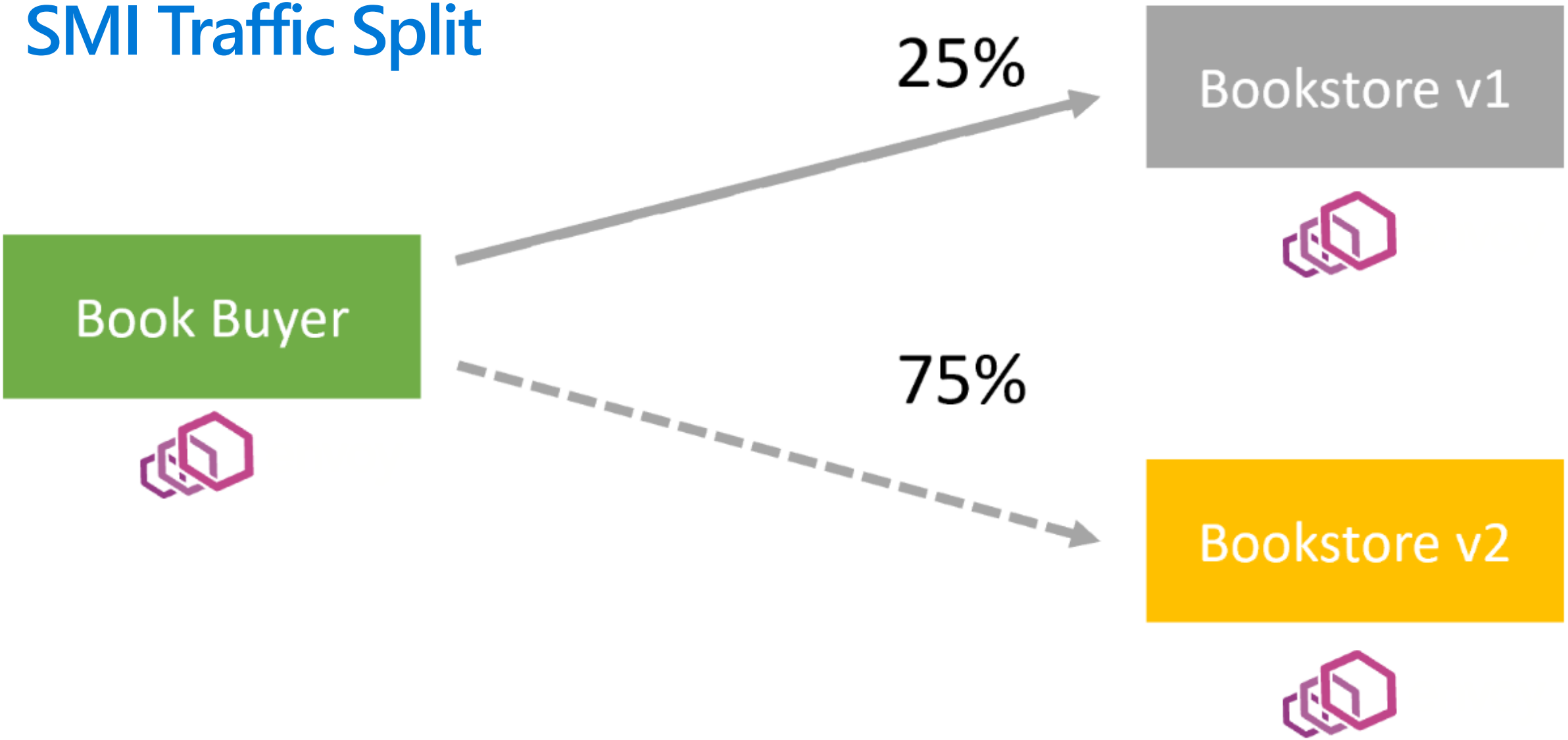
OSM upstream version

# Bookstore - Initial

# Service Mesh Layer 7 traffic encryption (mTLS)

# Bookstore V2

# SMI Traffic Split



# Azure Monitor integration (preview)

**Preview Docs:**
https://aka.ms/azmon/osmpreview

**Preview AzMon report:**
https://aka.ms/azmon/osmux


Home >

## OSM monitoring

aks-cni

Workbooks | Edit | Help | Auto refresh: Off

Time range: Last hour | Filter Internals : <Hide Internals> | Namespace: <All>

Requests | Connections

*Select a row below to start.*

Search

| Request source | Http requests total | Outbound bytes sent | Outbound bytes received |
|---|---|---|---|
| bookstore-v2 | 5471 | 1.38MB | 5.79MB |
| bookstore | 1908 | 493.87KB | 2.03MB |
| bookthief | 1229 | 959.71KB | 1.96MB |
| bookwarehouse | 2460 | 634.22KB | 384.38KB |
| bookbuyer | 6148 | 2.18MB | 9.54MB |

# Azure Monitor metrics

New Query 1*

Feedback | Queries | Query explorer

aks-demos | Select scope | Run | Time range : Last 24 hours | Save | Share | New alert rule | Export

Tables | Queries | Functions

Search

Filter | Group by: Resource t...

Collapse all

**Favorites**

You can add favorites by clicking on the ☆ icon

**Kubernetes Services**

- AzureActivity
- AzureDiagnostics
- AzureMetrics
- ContainerInventory
- ContainerLog
- ContainerNodeInventory
- Heartbeat
- InsightsMetrics
- KubeEvents
- KubeHealth
- KubeMonAgentEvents

```
InsightsMetrics
| where Name contains "envoy"
| extend t=parse_json(Tags)
| where t.app == "bookbuyer"
| order by TimeGenerated
```

Results | Chart | Columns | Display time (UTC+00:00) | Group columns

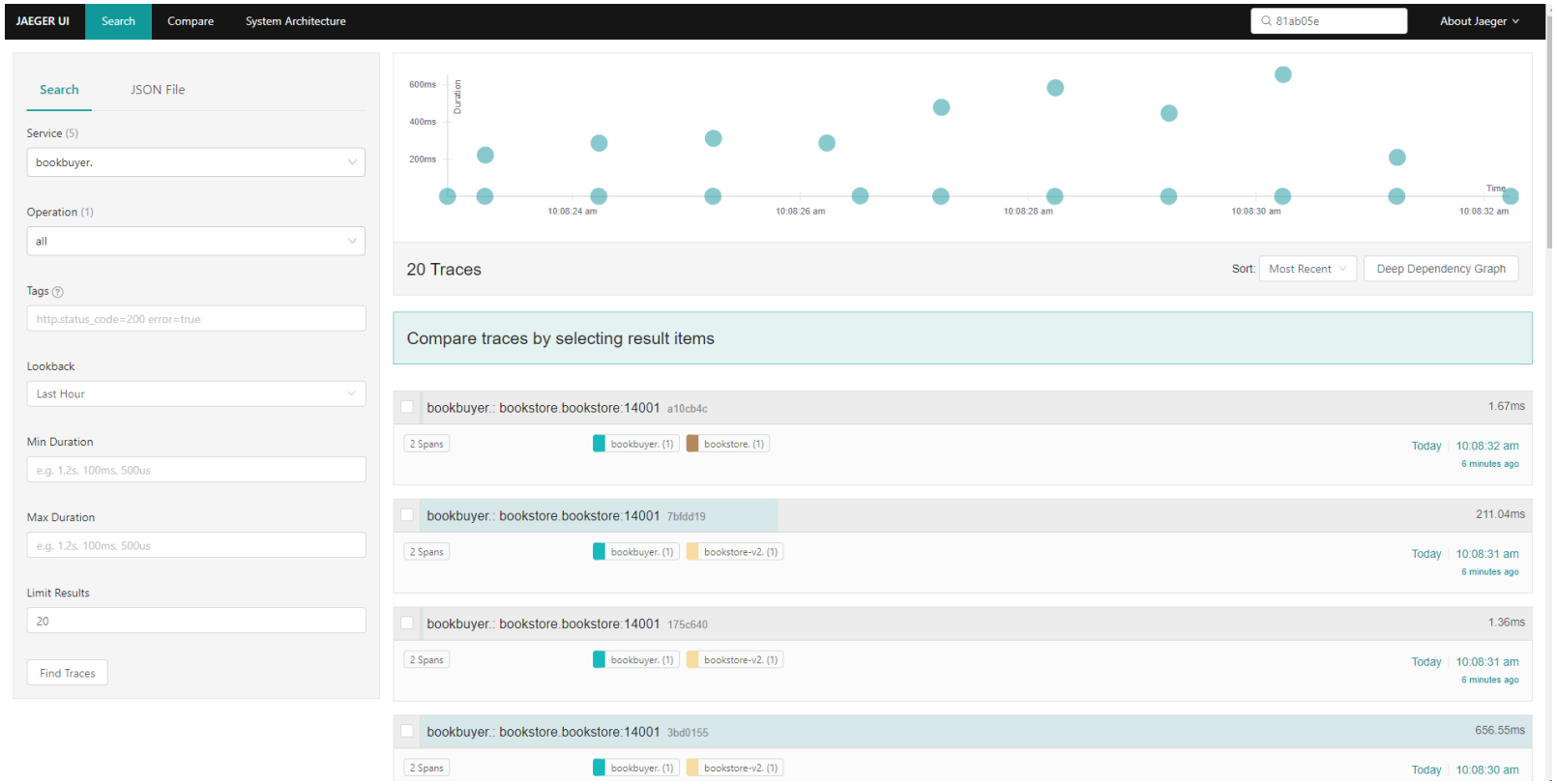
**Completed**. Showing results from the last 24 hours. 00:01.1 | 1,284 records

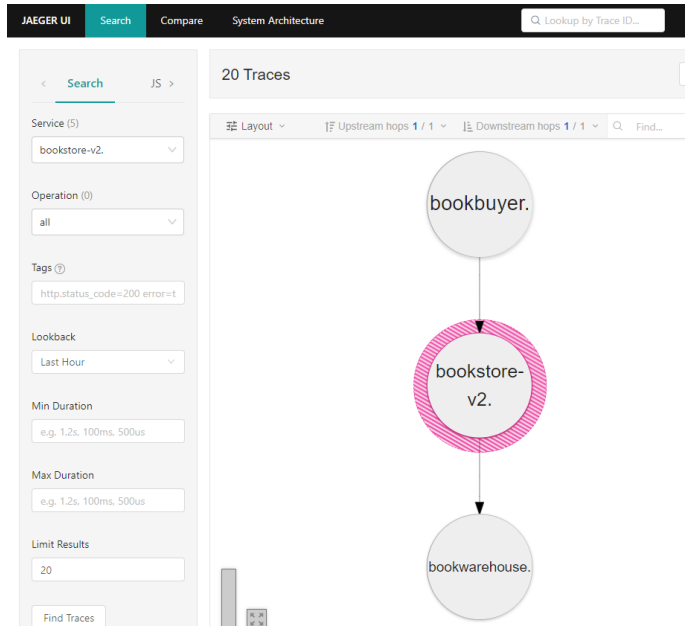
| TimeGenerated [UTC] | t | Computer |
|---|---|---|

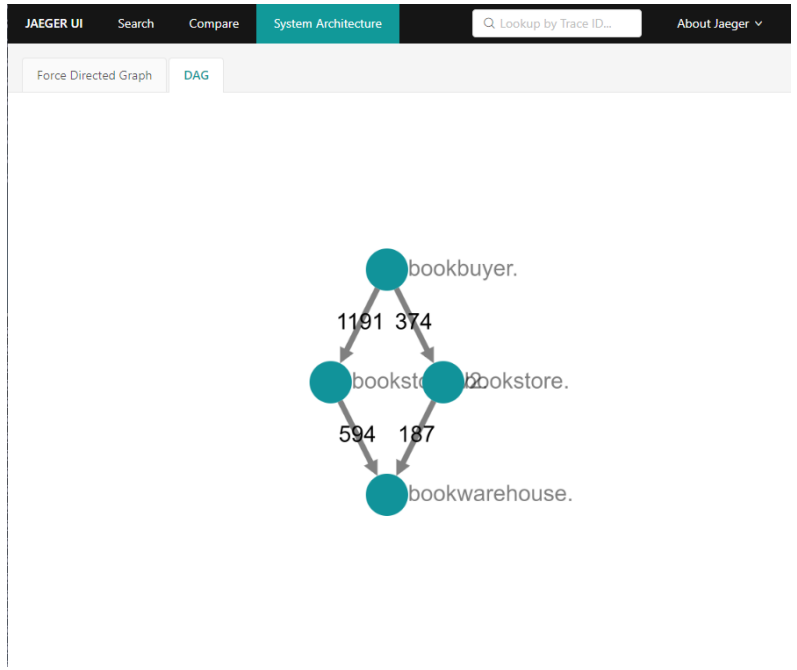
| | |
|---|---|
| SourceSystem | Insights |
| TimeGenerated [UTC] | 2022-02-03T11:24:00Z |
| Computer | aks-nodepool1-38579899-vmss000001 |
| Origin | container.azm.ms/telegraf |
| Namespace | container.azm.ms.osm/prometheus |
| Name | envoy_cluster_upstream_rq_time_bucket |

Page 1 of 26 | 50 items per page | 1 - 50 of 1284 items

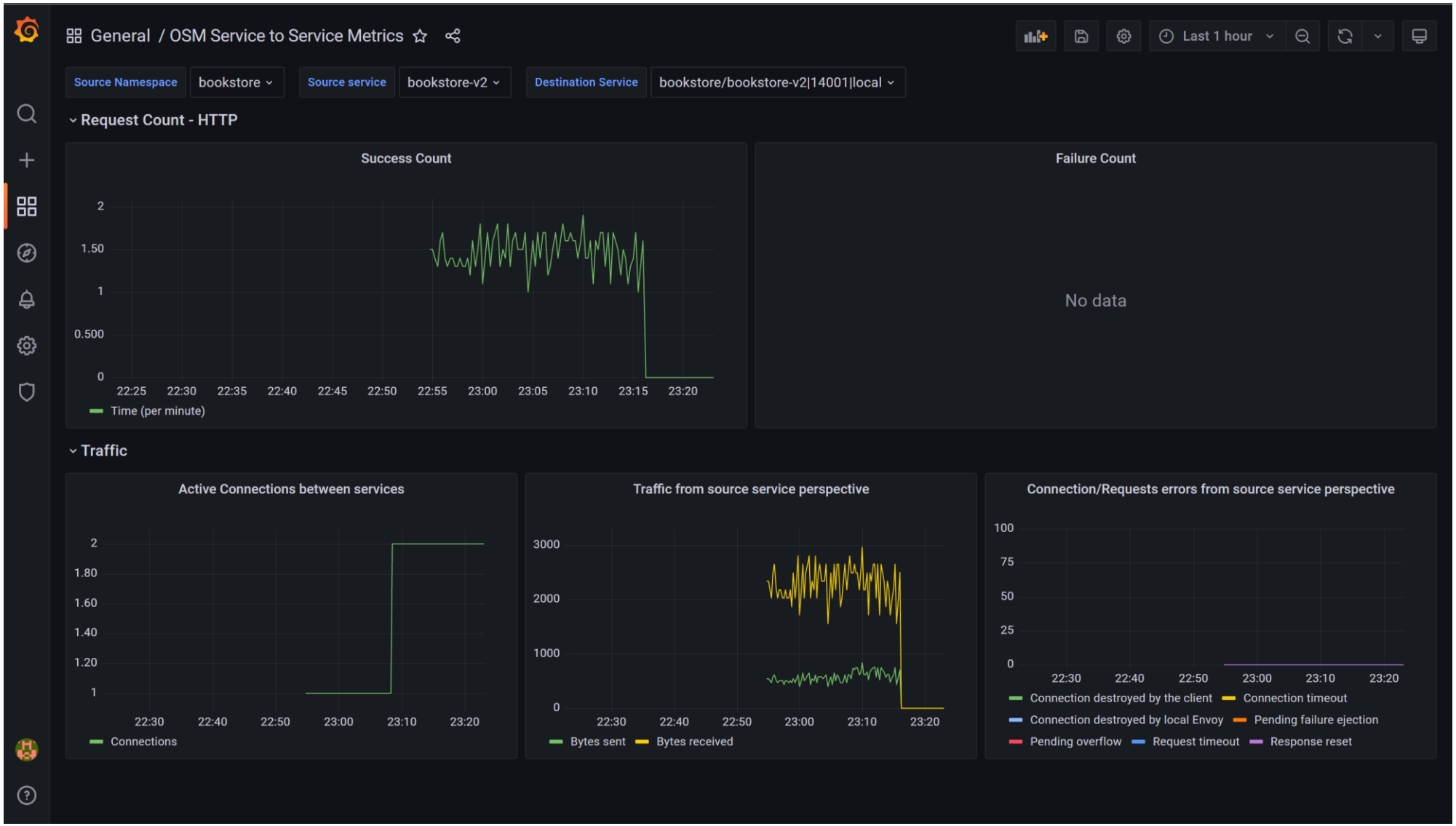
# Tracing with Jaegar



# Prometheus and Grafana monitoring



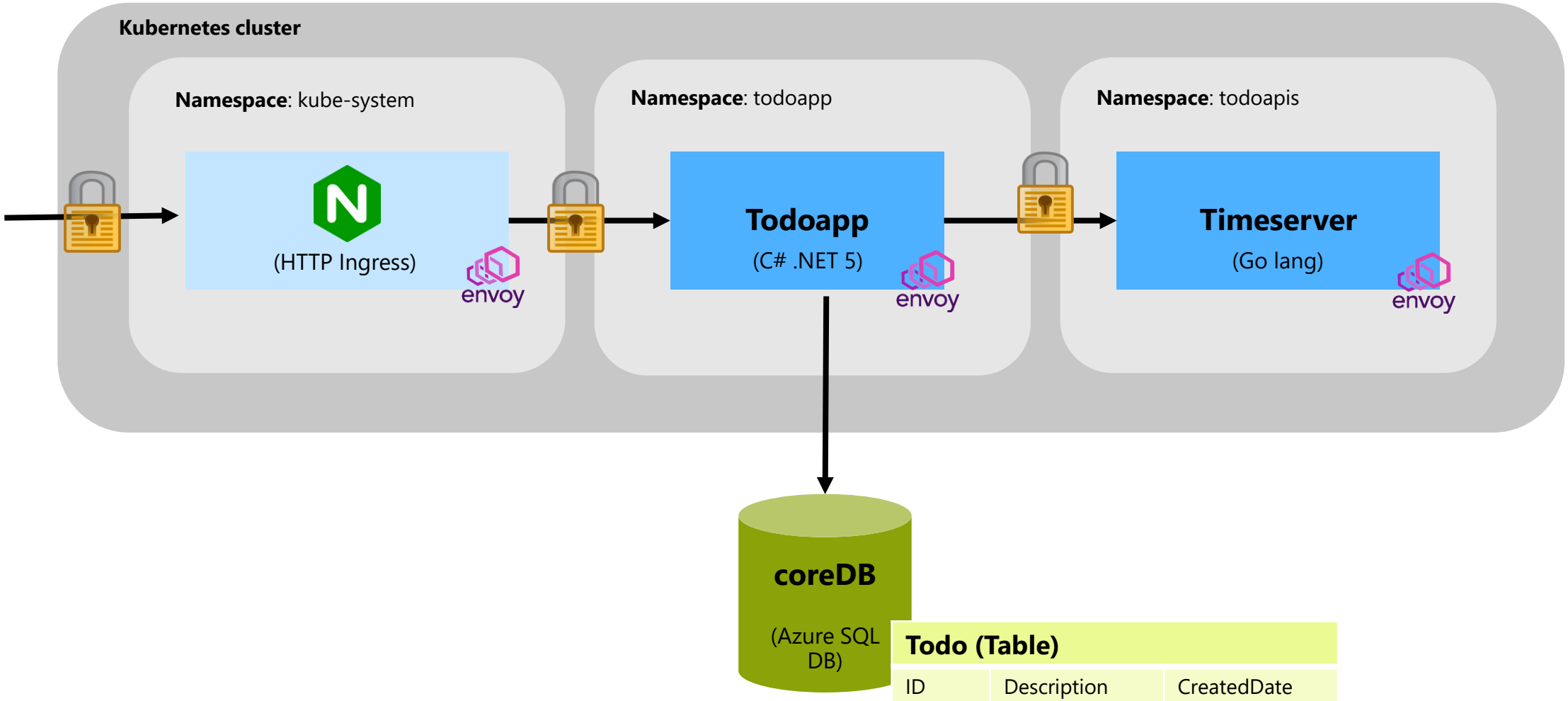
# Demo 2:

# ASP.NET Core and Azure SQL Database app with OSM on AKS

https://github.com/clarenceb/dotnetcore-sqldb-osm/blob/anz-developer/Demo.md

(note - use branch: `anz-developer`)

# Todo App overview




| Todo (Table) | | |
|---|---|---|
| ID | Description | CreatedDate |
| PK, int | nvarchar | datetime2 |

# Demo 3:

# OSM mTLS check

https://github.com/clarenceb/osm-mtls-check

# OSM mTLS check

*- [encrypted/httpbin-869b78c659-fzhsv/httpbin]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Start Stop Restart Options Open Save Close Reload Find Packet... Previous Packet Next Packet Go to Packet... First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In Zoom Out

Apply a display filter ... <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 62 | 10.025176 | 10.0.0.10 | 10.240.0.49 | DNS | 166 | Standard query response 0x2cee A osm-controller.kube-system.svc.cluster.local A 10.0.81.77 |
| 63 | 10.724055 | 10.240.0.77 | 10.240.0.49 | TCP | 76 | 56542 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1418 SACK_PERM=1 TSval=712180902 TSecr=0 WS=128 |
| 64 | 10.724114 | 10.240.0.49 | 10.240.0.77 | TCP | 76 | http(80) → 56542 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2131319556 TSecr= |
| 65 | 10.724668 | 10.240.0.77 | 10.240.0.49 | TCP | 68 | 56542 → http(80) [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=712180903 TSecr=2131319556 |
| 66 | 10.724736 | 10.240.0.77 | 10.240.0.49 | TLSv1.3 | 585 | Client Hello |
| 67 | 10.724752 | 10.240.0.49 | 10.240.0.77 | TCP | 68 | http(80) → 56542 [ACK] Seq=1 Ack=518 Win=64768 Len=0 TSval=2131319557 TSecr=712180903 |
| 68 | 10.726514 | 10.240.0.49 | 10.240.0.77 | TLSv1.3 | 1629 | Server Hello, Change Cipher Spec, Application Data |
| 69 | 10.726630 | 10.240.0.77 | 10.240.0.49 | TCP | 68 | 56542 → http(80) [ACK] Seq=518 Ack=1562 Win=64128 Len=0 TSval=712180905 TSecr=2131319558 |
| 70 | 10.728558 | 10.240.0.77 | 10.240.0.49 | TLSv1.3 | 1360 | Change Cipher Spec, Application Data |
| 71 | 10.728575 | 10.240.0.49 | 10.240.0.77 | TCP | 68 | http(80) → 56542 [ACK] Seq=1562 Ack=1810 Win=64128 Len=0 TSval=2131319560 TSecr=712180907 |
| 72 | 10.728608 | 10.240.0.77 | 10.240.0.49 | TLSv1.3 | 242 | Application Data |
| 73 | 10.728615 | 10.240.0.49 | 10.240.0.77 | TCP | 68 | http(80) → 56542 [ACK] Seq=1562 Ack=1984 Win=64128 Len=0 TSval=2131319560 TSecr=712180907 |
| 74 | 10.729263 | localhost | localhost | TCP | 76 | 52714 → http(80) [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2643075590 TSecr=0 WS=128 |
| 75 | 10.729276 | localhost | localhost | TCP | 76 | http(80) → 52714 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2643075590 TSec |
| 76 | 10.729288 | localhost | localhost | TCP | 68 | 52714 → http(80) [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2643075590 TSecr=2643075590 |
| 77 | 10.729362 | localhost | localhost | HTTP | 346 | GET / HTTP/1.1 |
| 78 | 10.729367 | localhost | localhost | TCP | 68 | http(80) → 52714 [ACK] Seq=1 Ack=279 Win=65280 Len=0 TSval=2643075590 TSecr=2643075590 |
| 79 | 10.731318 | localhost | localhost | TCP | 307 | http(80) → 52714 [PSH, ACK] Seq=1 Ack=279 Win=65536 Len=239 TSval=2643075592 TSecr=2643075590 [TCP |
| 80 | 10.731338 | localhost | localhost | TCP | 68 | 52714 → http(80) [ACK] Seq=279 Ack=240 Win=65408 Len=0 TSval=2643075592 TSecr=2643075592 |

```
Transmission Control Protocol, Src Port: 56542 (56542), Dst Port: http (80), Seq: 1, Ack: 1, Len: 517
Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 508
      Version: TLS 1.2 (0x0303)
      Random: 1ae17d26d3b600aef8ece04c25c9978a8b69390ae34e03518bb7e7e221be9fcf
      Session ID Length: 32
      Session ID: cd16ee44cc6a8b815ee53699eeda5db4c9efba55a67c913efde9a0096ddbe82b
      Cipher Suites Length: 18
      Cipher Suites (9 suites)
        Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
        Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
        Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
        Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
        Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 417
```

```
0000  00 00 00 01 00 06 c6 b5  0a 57 ae c9 00 00 08 00   ........ .W......
0010  45 00 02 39 39 db 40 00  3e 06 ea 86 0a f0 00 4d   E..99.@. >......M
0020  0a f0 00 31 dc de 00 50  a8 e9 84 c1 78 0e d6 89   ...1...P ....x...
0030  80 18 01 f6 16 0e 00 00  01 01 08 0a 2a 73 04 a7   ........ ....*s..
0040  7f 09 5b 04 16 03 01 02  00 01 00 01 fc 03 03 1a   ..[..... ........
0050  e1 7d 26 d3 b6 00 ae f8  ec e0 4c 25 c9 97 8a 8b   .}&..... ..L%....
0060  69 39 0a e3 4e 03 51 8b  b7 e7 e2 21 be 9f cf 20   i9..N.Q. ...!...
0070  cd 16 ee 44 cc 6a 8b 81  5e e5 36 99 ee da 5d b4   ...D.j.. ^.6...].
0080  c9 ef ba 55 a6 7c 91 3e  fd e9 a0 09 6d db e8 2b   ...U.|.> ....m..+
```

wireshark_-3U9RF1.pcapng  Packets: 109 · Displayed: 109 (100.0%) · Dropped: 0 (0.0%)  Profile: Default

# Roadmap

BLOG

# Announcing OSM v1.0.0

Phillip Gibson & Jon Huhn

@openservicemesh

Feb 01, 2022

OSM's contributors have been hard at work over the past several months preparing for the v1.0.0 release. Today, the OSM team proudly announces the release of version 1.0.0. Thanks to the community for helping drive the features they want to see in a service mesh and heightening the expectation for OSM to provide enterprise features and functionality.

[Read more]

# Roadmap

- Public Roadmap - https://github.com/openservicemesh/osm/projects/3
- CNCF Community Calls (YouTube) - https://www.youtube.com/playlist?list=PLMy1gNP8pzh53-ScOG8i1BQW5cvAGkF0X
- Upcoming features:
  - Windows container support
  - Multi-cluster support
  - Azure-specific integrations – Azure Monitor (OSM stats and health), App Insights (via OpenTelemetry)
  - External CA management
  - AGIC integration for a Microsoft support Ingress controller