Service Mesh -

from technical selection to best practice

CLOUD NATIVE
COMPUTING FOUNDATION

About me

- Ruofei Ma – Principal software engineer, FreeWheel
- The author of the book ***Istio in practice***
- The columnist of ***Service mesh in practice*** in time.geekbang.org
- A consultant of the IT book expert committee of Posts and Telecom Press
- A member of the committee of the largest service mesh technology community, servicemesher.com in China
- Istio.io contributor

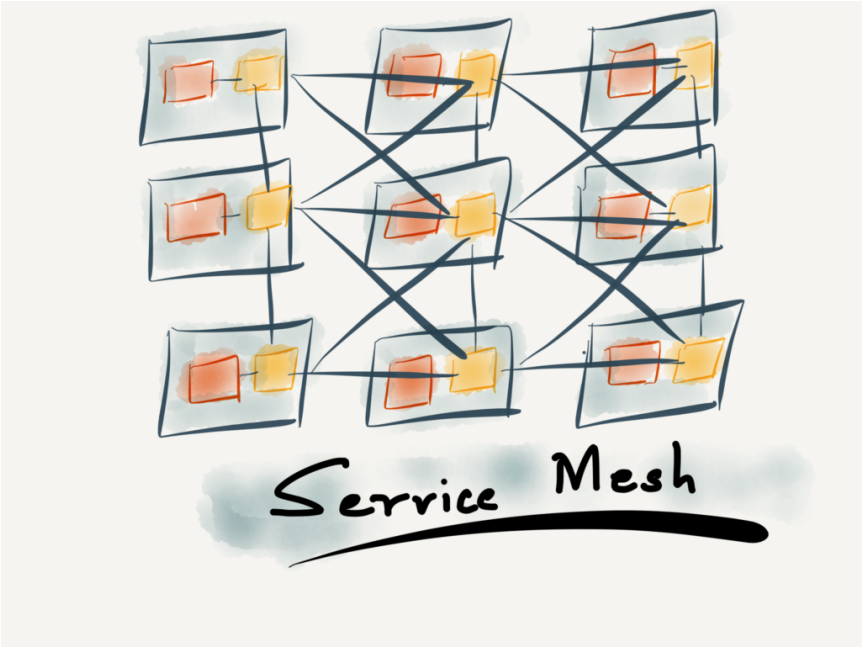
Agenda

- Why service mesh
- Market of service mesh
- Istio vs AWS App Mesh
- Best practice
- In the future

Why service mesh

Concept of service mesh

- Key points
  - Infrastructure
  - Delivery requests
  - Sidecar proxy
  - Transparency
- Service governance



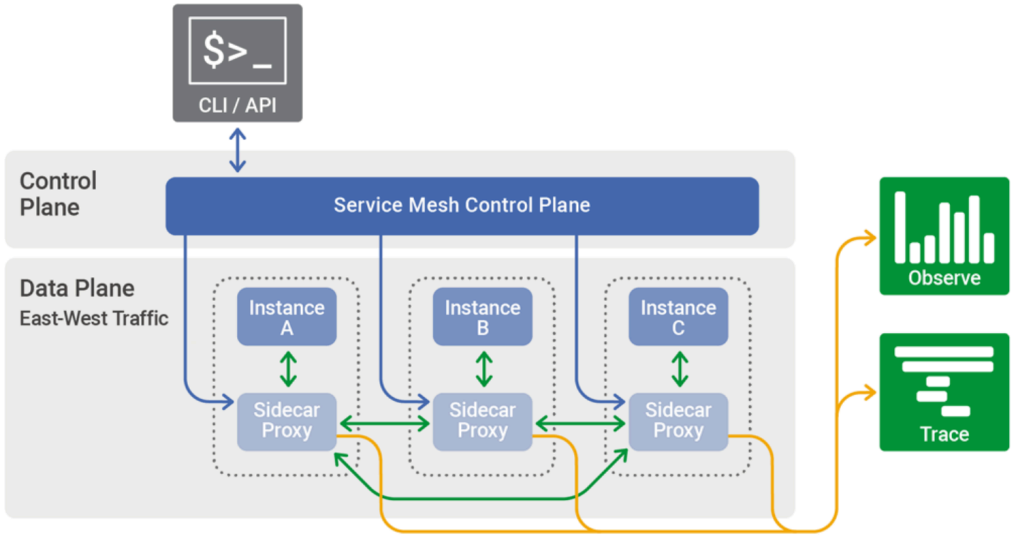
picture is from https://softwareengineeringdaily.com/2020/01/07/service-meshes/

Disadvantages of traditional service governance

- Complexity
- Costs
  - Human resource
  - Operations
- Coupling with application
- Language binding

Advantages of service mesh

- Transparency
- Lower costs
  - Development
  - Operation & maintenance
  - Resources



picture is from https://www.nginx.com/blog/what-is-a-service-mesh/

Market of service mesh

Products

Open-source

Hosted

AWS App Mesh

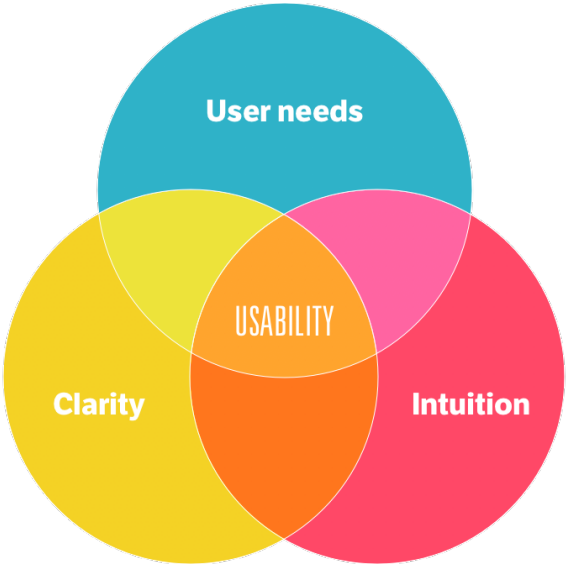
Easily monitor and contr

Traffic Director

Service Fabric Mesh

Trends

- Mixed-cloud support
- Usability
- Adoption costs
- Performance
- Standardization
- Ecosystem



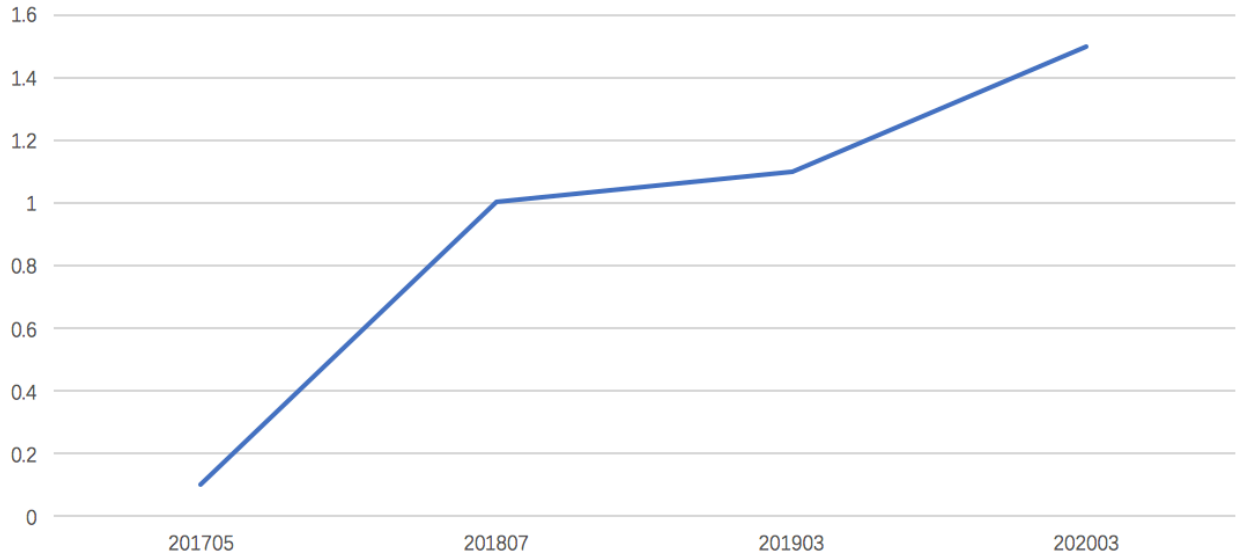
Istio vs AWS App Mesh

Why Istio & AWS App Mesh

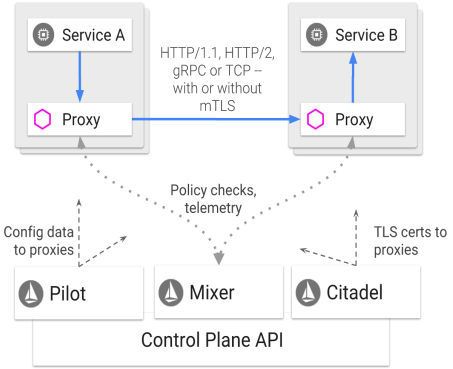
- First-tier companies (Google & Amazon)
- Open source vs hosted product
- Business scenarios: heavy user of AWS

Superstar: Istio

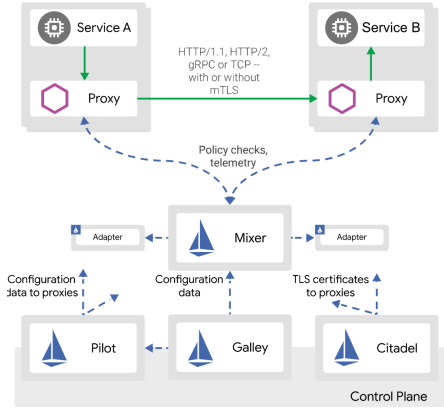
v0.1 v1.0 v1.1 v1.2 v1.3 v1.4 v1.5 v1.6



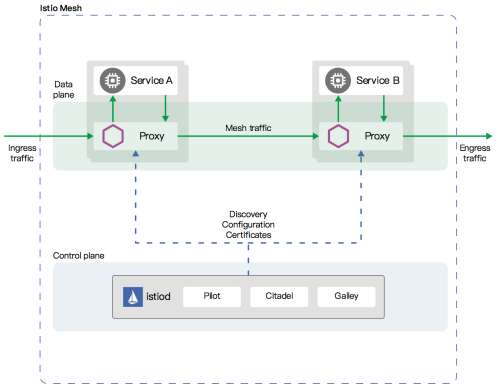
Istio's architecture revolution



v1.0

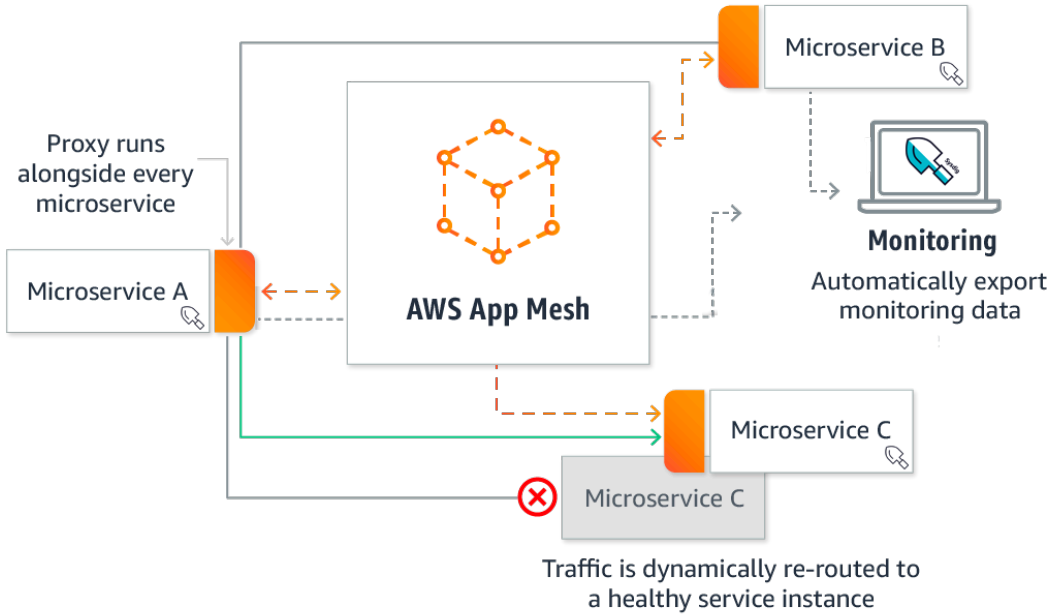


v1.1

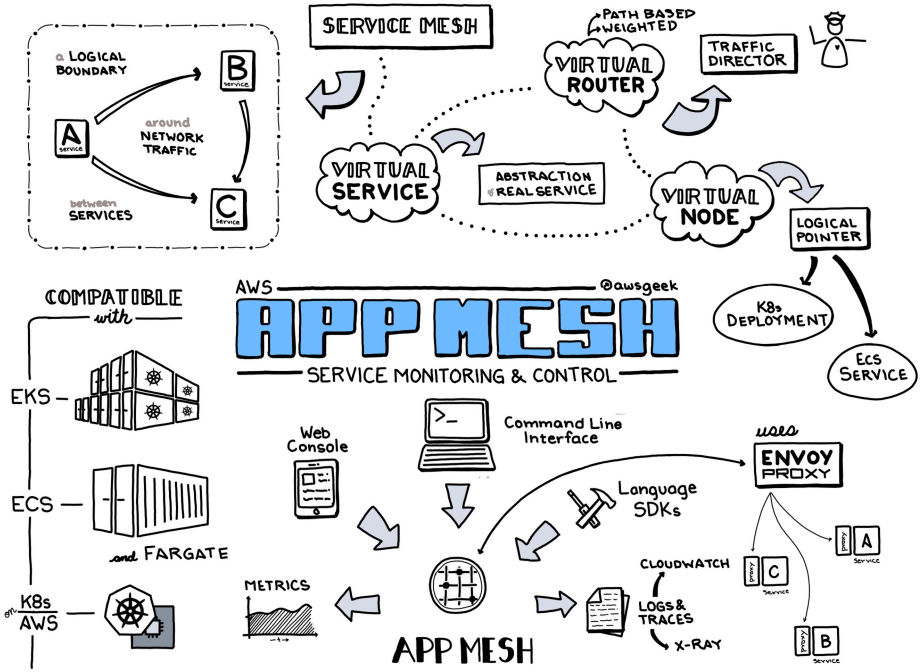


v1.5

Challenger: AWS App Mesh



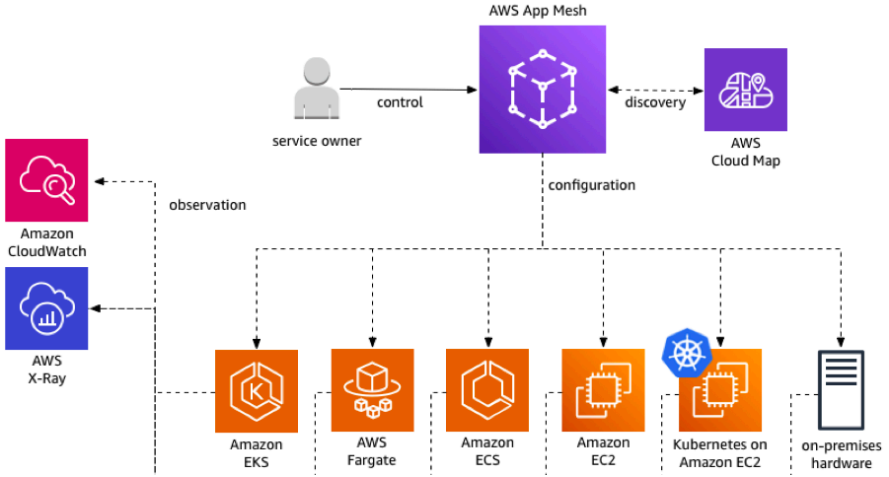
Overview of App Mesh



The picture from https://www.awsgeek.com/AWS-App-Mesh/

Characteristics of App Mesh

- Multiple workloads support
- Unify user experience
- Integration for AWS services



Comparison

CLOUD NATIVE
COMPUTING FOUNDATION

Comparison – product vision

Istio

- It is a completely open source service mesh that layers transparently onto existing distributed applications.
- It is also a platform, including APIs that let it integrate into any logging platform, or telemetry or policy system.
- Istio's diverse feature set lets you successfully, and efficiently, run a distributed microservice architecture, and provides a uniform way to secure, connect, and monitor microservices.

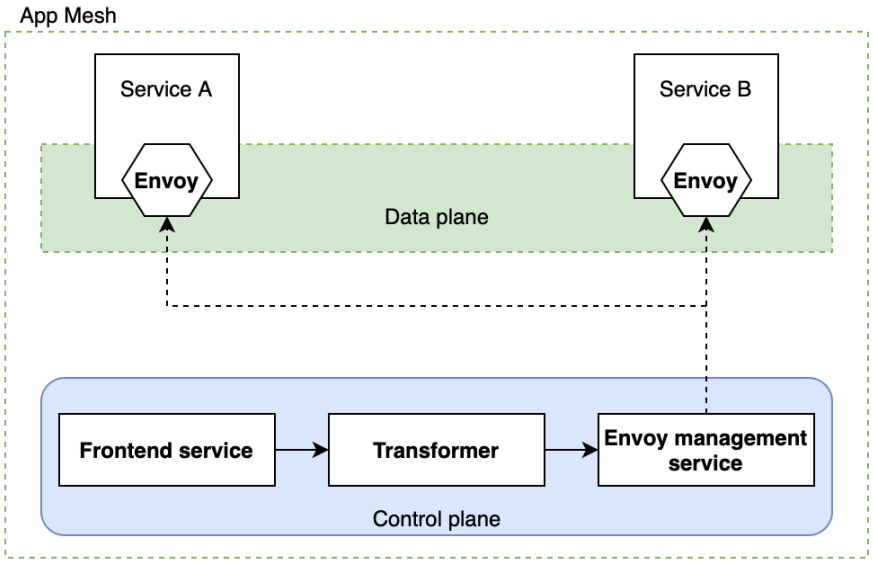
AWS App Mesh

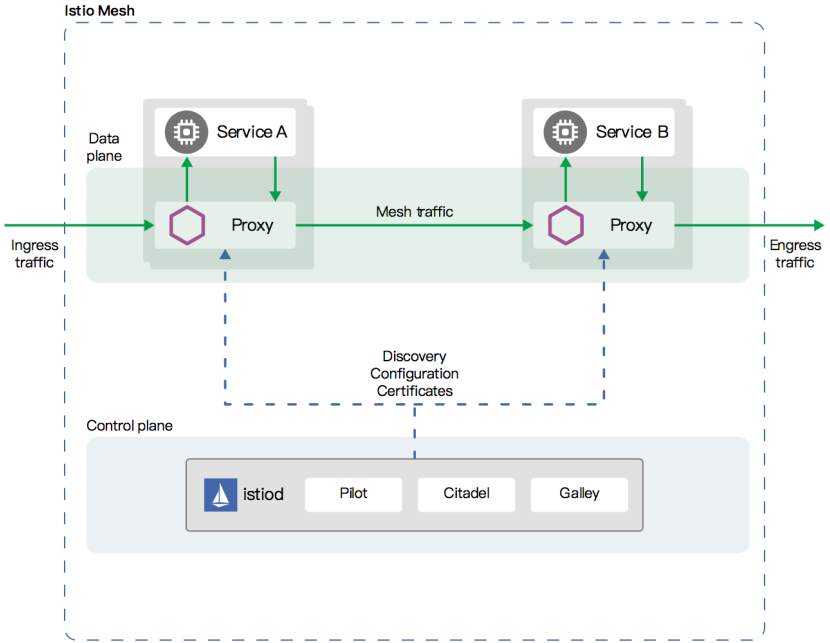
- App Mesh gives you a simple, declarative approach to model service communication. You can define rules for service-to-service communication and everything else is handled automatically.
- It provides consistent metrics, logs, and traces, and gives end-to-end visibility across an application to help you quickly identify and debug issues.
- Our vision for App Mesh is an AWS-native service mesh that integrates equally well with AWS primitives and advanced services.

Comparison – overview

	Istio	AWS App Mesh
Platform	Kubernetes, Consul, GCP	AWS (EKS, ECS, EC2, Fargate …)
Sidecar	Envoy	Envoy
Automatic injection	Yes	Yes
Protocol	TCP, HTTP, gRPC	TCP, HTTP, gRPC
Cross cluster mesh	Yes	Yes

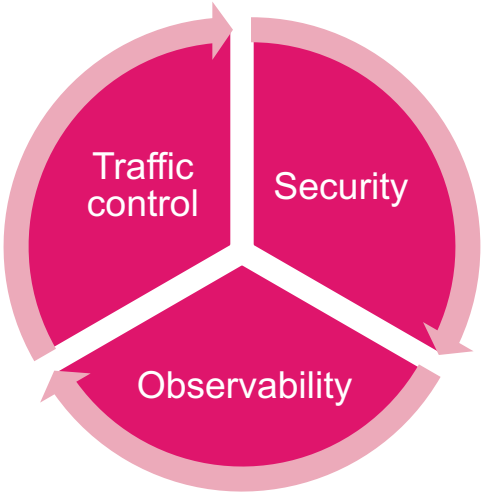
Comparison – architecture



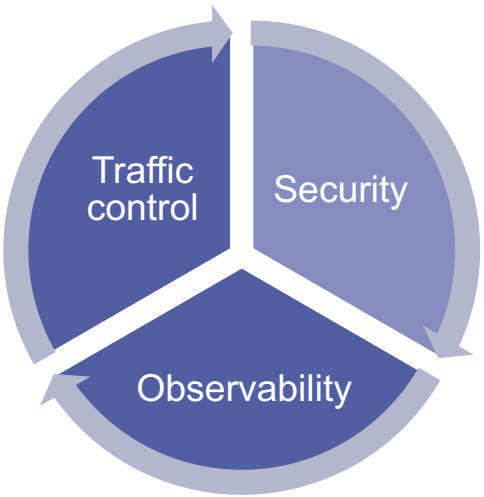


Comparison – features

Istio



AWS App Mesh



Comparison – traffic control

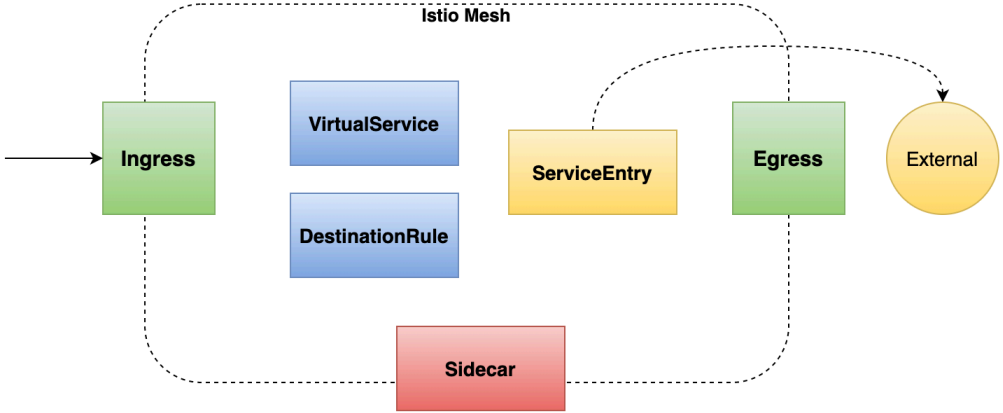
Istio

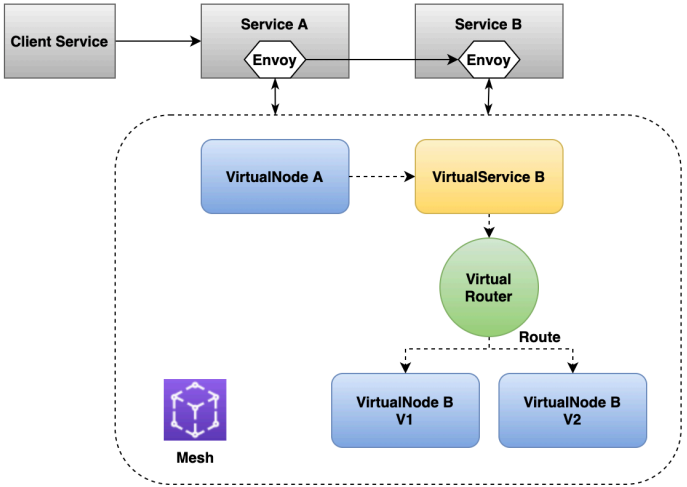
- Routing & traffic shifting
  - Traffic policy (Load balancing, connection pool)
  - Percentage-base traffic splits
  - Header- & path-based traffic splits
- Resilience
  - Timeout, retry, circuit breaking
- Debugging & testing
  - Mirror, fault injection

AWS App Mesh

- Routing
  - Percentage-base traffic splits
  - Header- & path-based traffic splits
  - Service discovery: DNS, Cloudmap
- Resilience
  - Timeout, retry
  - Path-based retry

Comparison – CRDs for traffic control





Comparison – security

Istio

- CA certificate
- Authentication
  - Peer (mTLS)
  - Request (JWT)
- Authorization policy

AWS App Mesh

- CA certificate
- mTLS
- AWS IAM

Comparison – observability

Istio

- Logging
- Metrics
  - Prometheus, Grafana
- Tracing
  - Zipkin, Jaeger, Datadog …
- Kiali

AWS App Mesh

- Logging
- Metrics
  - Prometheus, Grafana
- Tracing
  - Jaeger, X-Ray, Datadog …
- AWS Cloud Watch

Comparison - maintainability

	Istio	AWS App Mesh
Deployment	Hard -> Easy	Easy
Technical support	None	Official support center
Troubleshooting	More	Less
User guide & reference	Good	Medium

Comparison – troubleshooting

Istio

- CLI (istioctl)
- Envoy admin API & Log
- controlZ & pilot debug API
- Kiali

AWS App Mesh

- Envoy admin API & Log
- Controller & Injector log
- X-ray/ CloudWatch

Comparison - costs

	Istio	AWS App Mesh
Pricing	Free	Free
Human resource	Higher	Lower
Learning costs	Higher	Lower

How to choose?

- Identify the important problems solved by service mesh.
- Consider your requirements (features, usability, performance, etc.)
- Consider your company and system situations.
- Follow the tutorial to install, implement features with mesh.
- Performance test

Our practice

CLOUD NATIVE
COMPUTING FOUNDATION

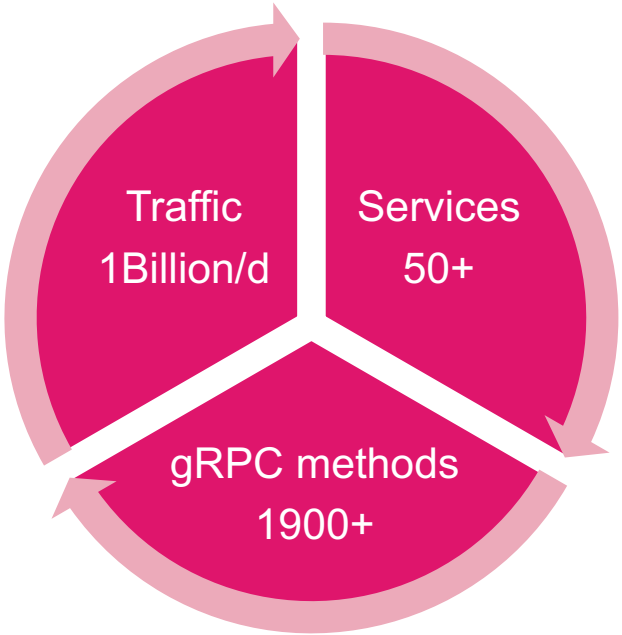
Business scenario

COMPANY	• FreeWheel – a Comcast company • we're pioneering a new TV ecosystem, transforming the way buyers & sellers transact, manage, & optimize their advertising
BUSINESS	• Unify linear & digital TV • Automate planning • Buying & selling
PRODUCT	• Digital media advertising system • Marketplace

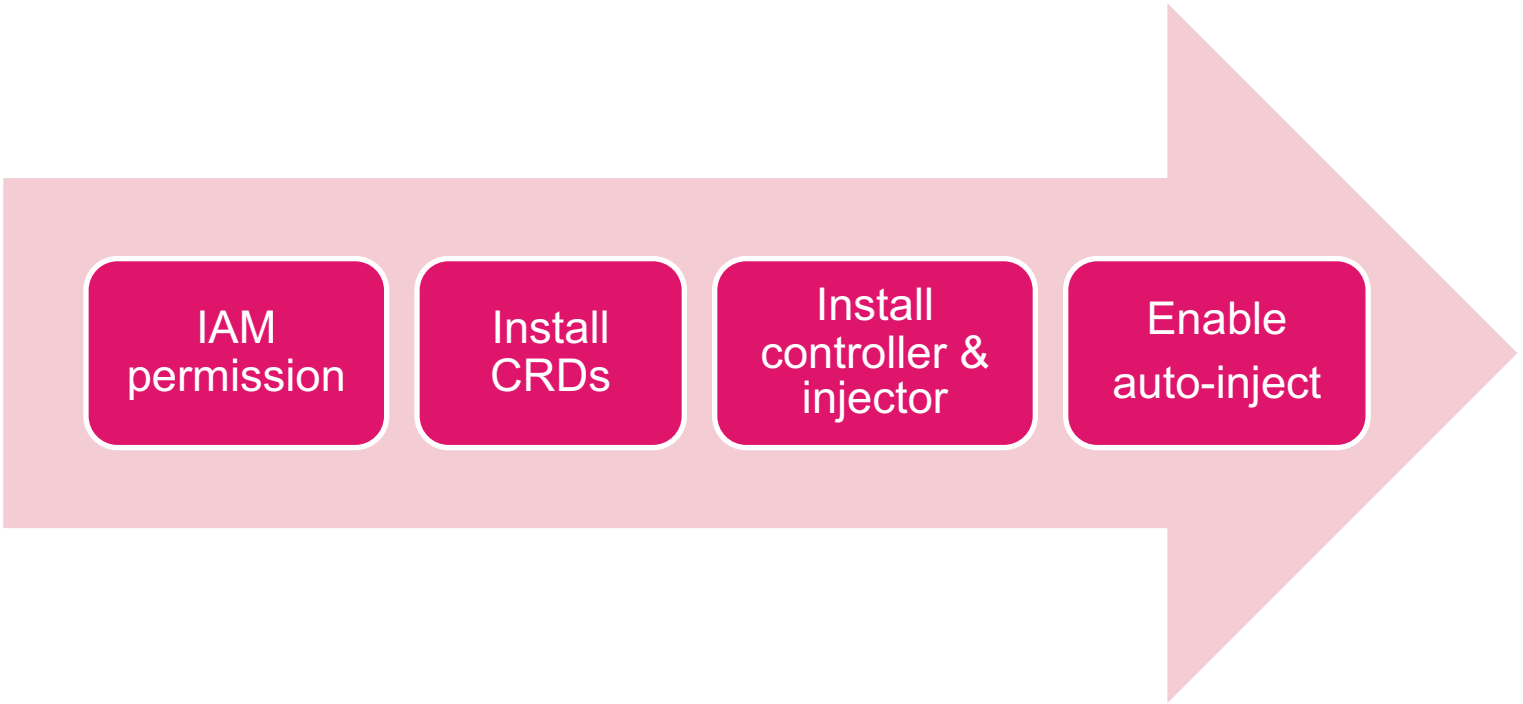
To cloud-native

Pain points

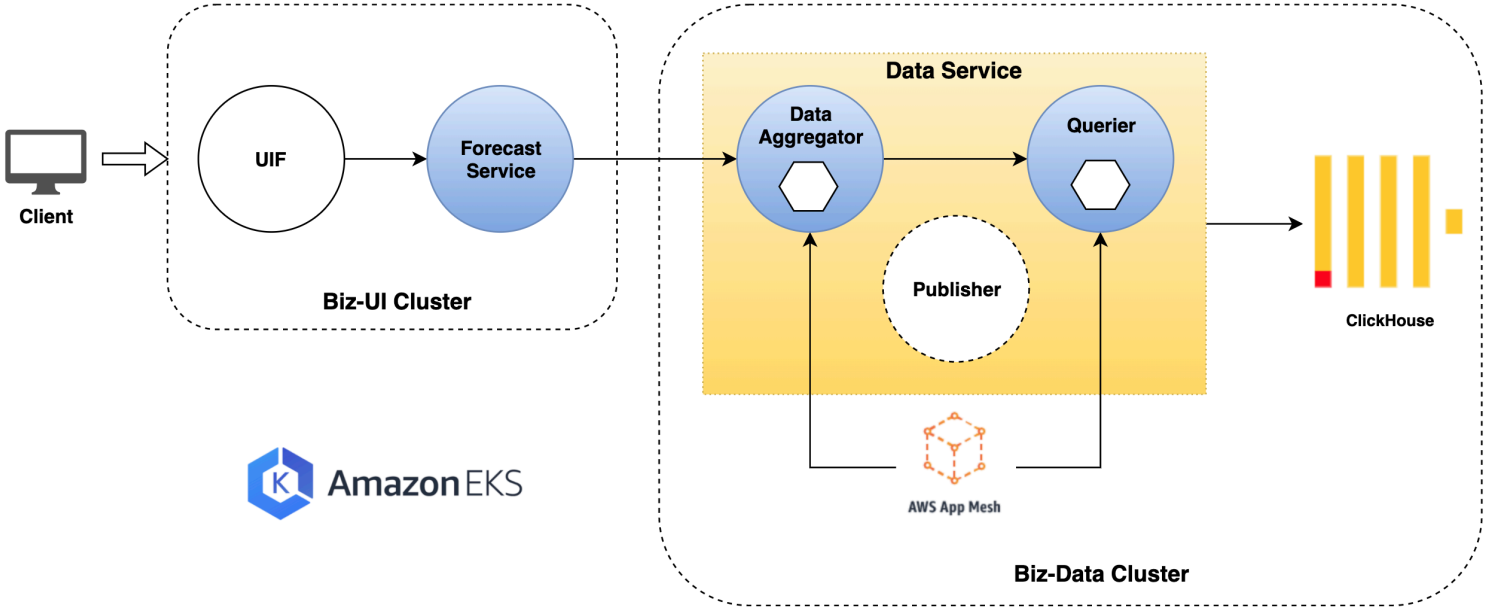
- Mixed deployment environments
- Complex business scenario
- Traffic control
- Lack of observability for services
- Troubleshooting



App Mesh adoption – Steps



App Mesh adoption – pilot project



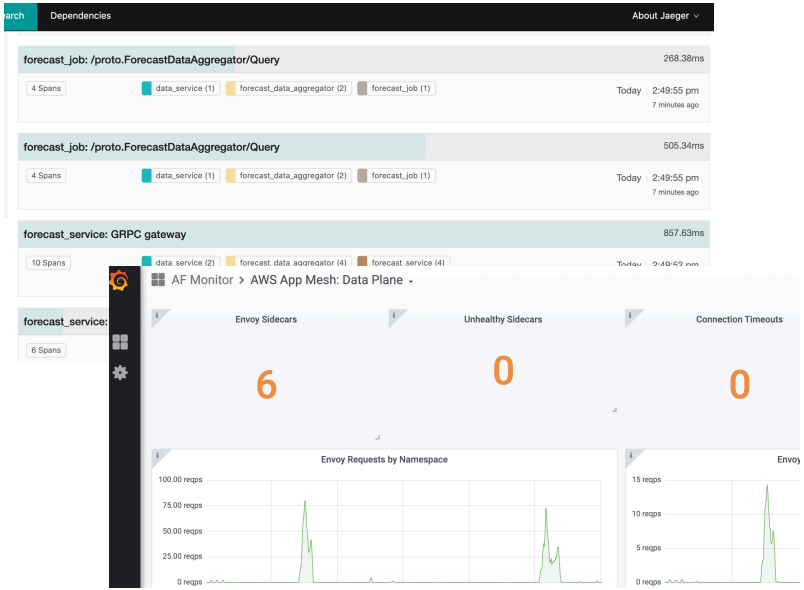
App Mesh adoption – configuration

```yaml
apiVersion: appmesh.k8s.aws/v1beta1
kind: Mesh
metadata:
  name: uiquery-mesh
---
apiVersion: appmesh.k8s.aws/v1beta1
kind: VirtualNode
metadata:
  name: forecast-data-aggregator #svc name
  namespace: uiquery #ns name
spec:
  meshName: uiquery-mesh
  listeners:
    - portMapping:
        port: 3370
        protocol: grpc
  logging:
    accessLog:
      file:
        path: /dev/stdout
  backends:
    - virtualService:
        virtualServiceName: query.uiquery.svc.cluster.local #svc na
  serviceDiscovery:
    dns:
      hostName: forecast-data-aggregator.uiquery.svc.cluster.local
```

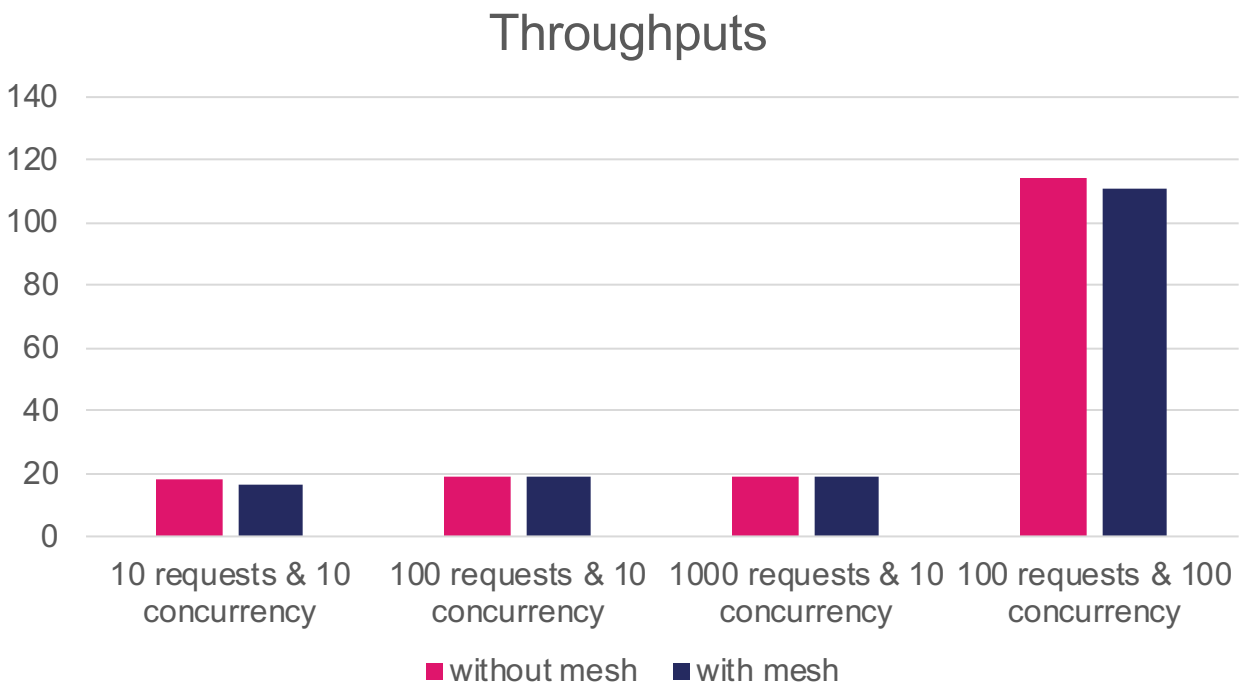
```yaml
apiVersion: appmesh.k8s.aws/v1beta1
kind: VirtualService
metadata:
  name: query.uiquery.svc.cluster.local #svc name
  namespace: uiquery
spec:
  meshName: uiquery-mesh
  virtualRouter:
    name: router
    listeners:
      - portMapping:
          port: 3360
          protocol: grpc
  routes:
    - name: route
      grpc:
        match:
          serviceName: proto.DataService
        action:
          weightedTargets:
            - virtualNodeName: query-service-uiquery
              weight: 100
```

App Mesh adoption – features

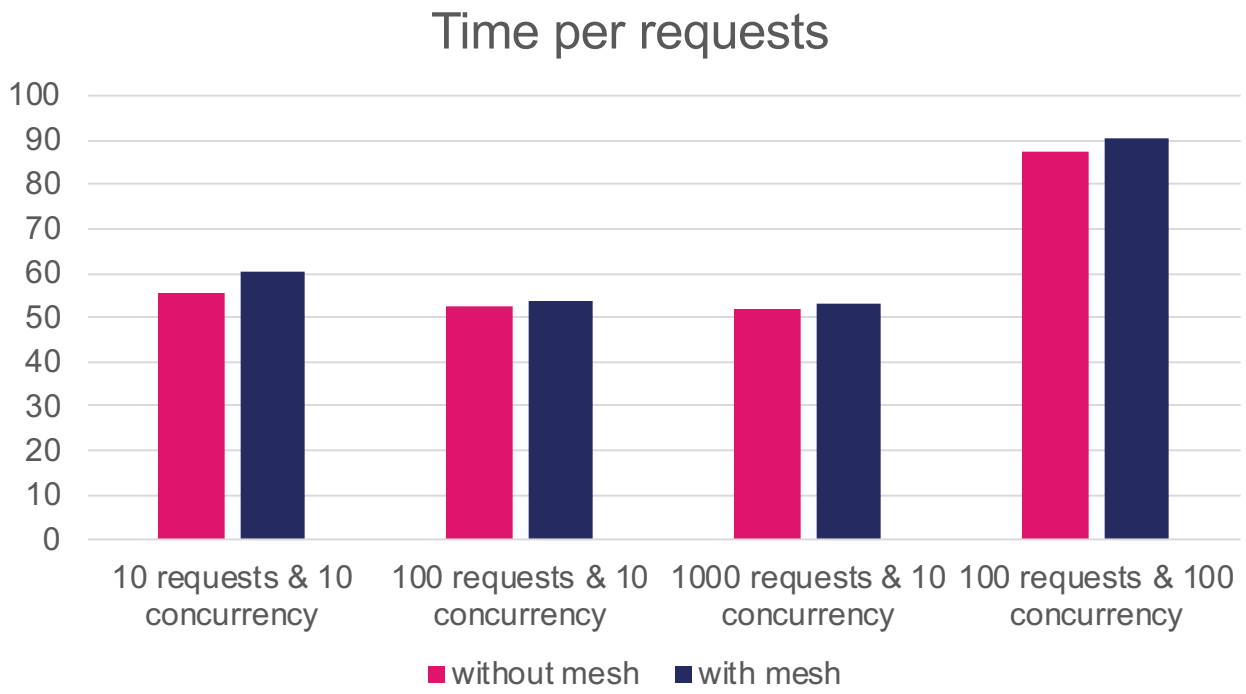
- Routing - Done
- Retry & Timeout - Done
- Observability – Done
- Canary release – In progress



App Mesh adoption - performance benchmark



App Mesh adoption - performance benchmark



Next steps

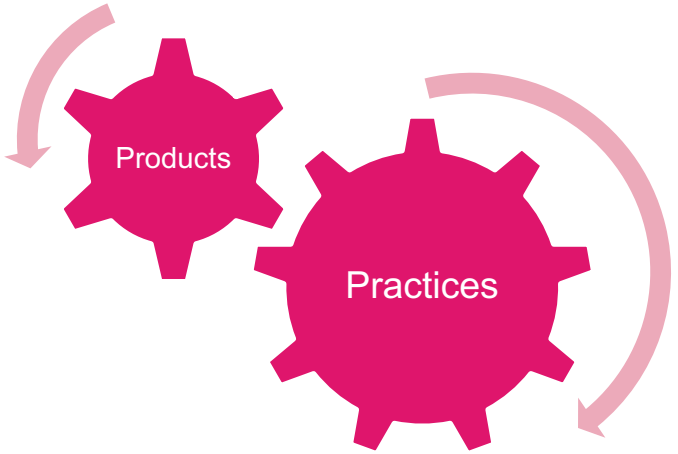
- Optimizing the chain of service communication
- Mesh by GitOps
- Deployment automatically (Flagger, Jenkins X)

In the future

CLOUD NATIVE
COMPUTING FOUNDATION

Currently

- Users: various deployment environments
- Products: maturity



Standardization

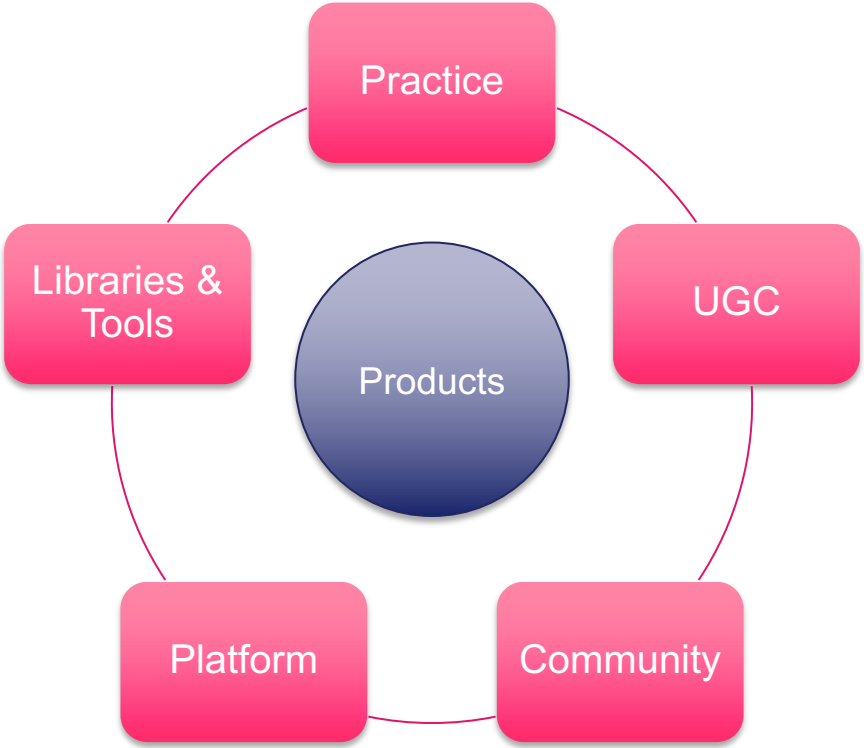
UDPA – Universal data plane API

The objective of the Universal Data Plane API Working Group (UDPA-WG) is to bring together parties across the industry interested in a common control and configuration API for data plane proxies and load balancers.

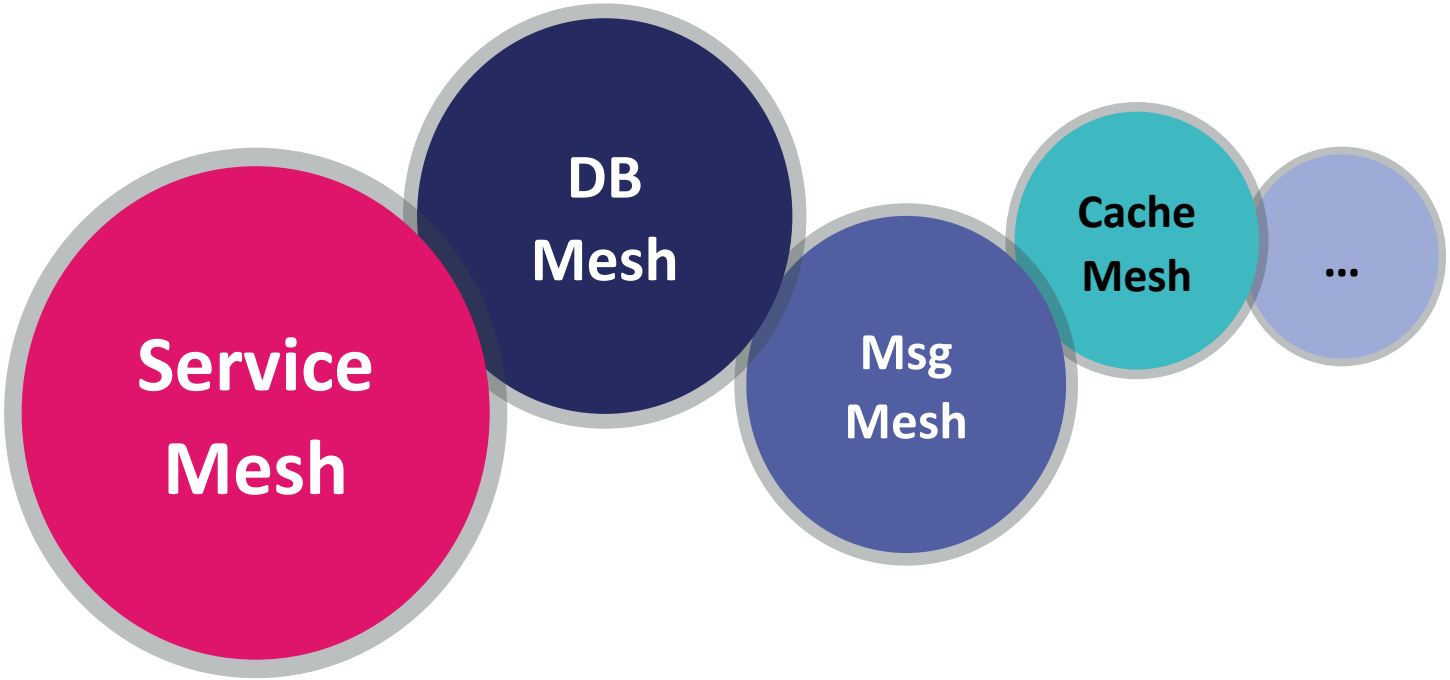
SMI – Service mesh interface

A specification for service meshes that run on Kubernetes. It defines a common standard that can be implemented by a variety of providers. This allows for both standardization for end-users and innovation by providers of Service Mesh Technology. SMI enables flexibility and interoperability and covers the most common service mesh capabilities.

Ecosystem of service mesh



Service mesh pattern



Thank You!

Contact me:
malphi@gmail.com
Dingtalk: malphi

ServiceMesher Community

Cloud Native Community

CLOUD NATIVE
COMPUTING FOUNDATION