STORAGE DEVELOPER CONFERENCE
SDC21
BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

Istio Service Mesh

A Primer

Presented by

Binitha MT
Senior Software Engineer
Dell Technologies

Deepika K
Senior Software Engineer
Dell Technologies

Agenda

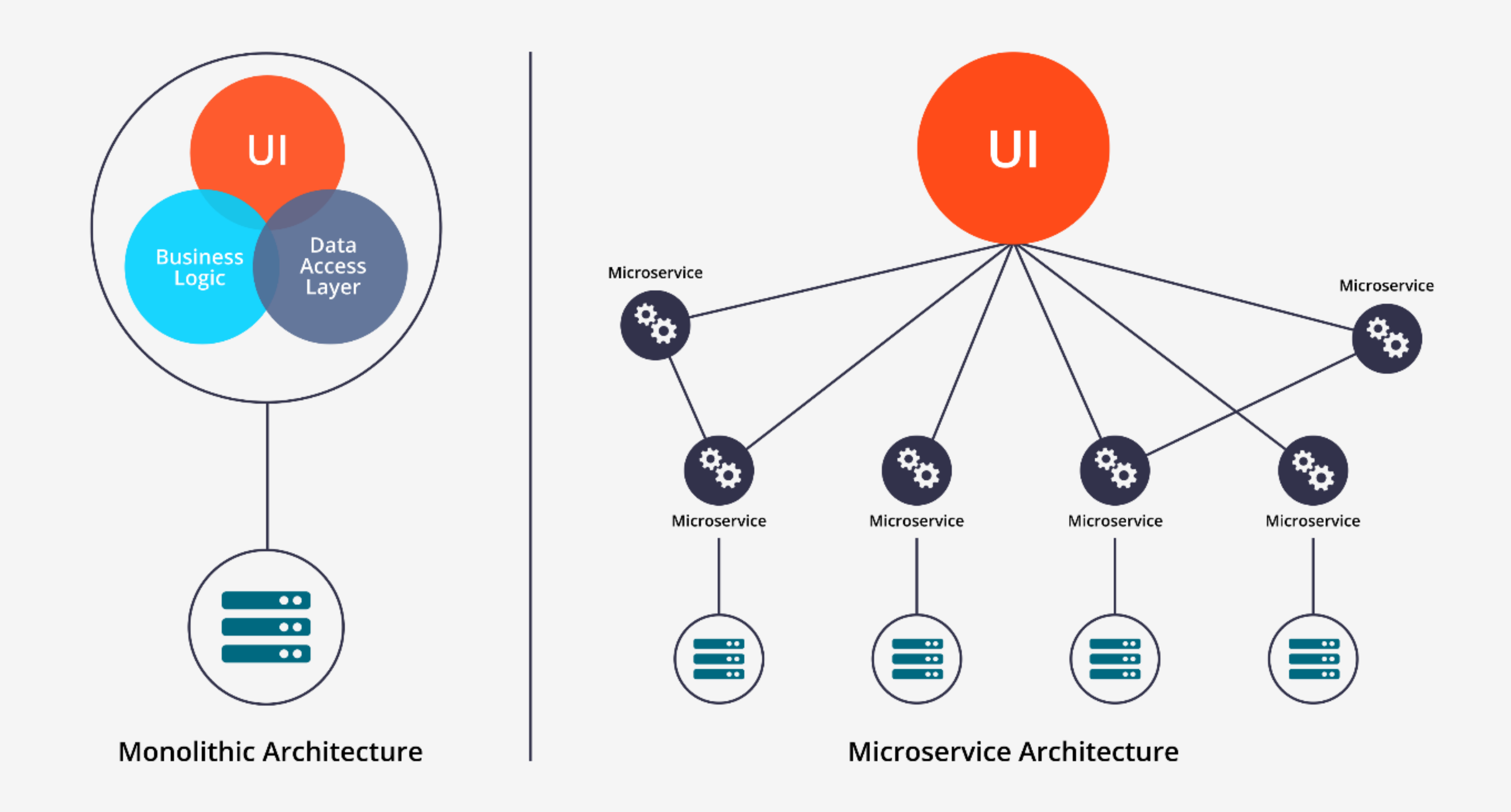
- Orientation
  - Evolution of Microservices
- Service Mesh
  - Requirements of Service Mesh
- Istio
  - W’s of Istio Service Mesh
  - Istio Architecture
  - Understanding Istio components
- Istio Core features
  - Traffic Management
  - Observability
  - Security
- Wrap Up/Questions

Speaker Photo Will Be Placed Here

Orientation

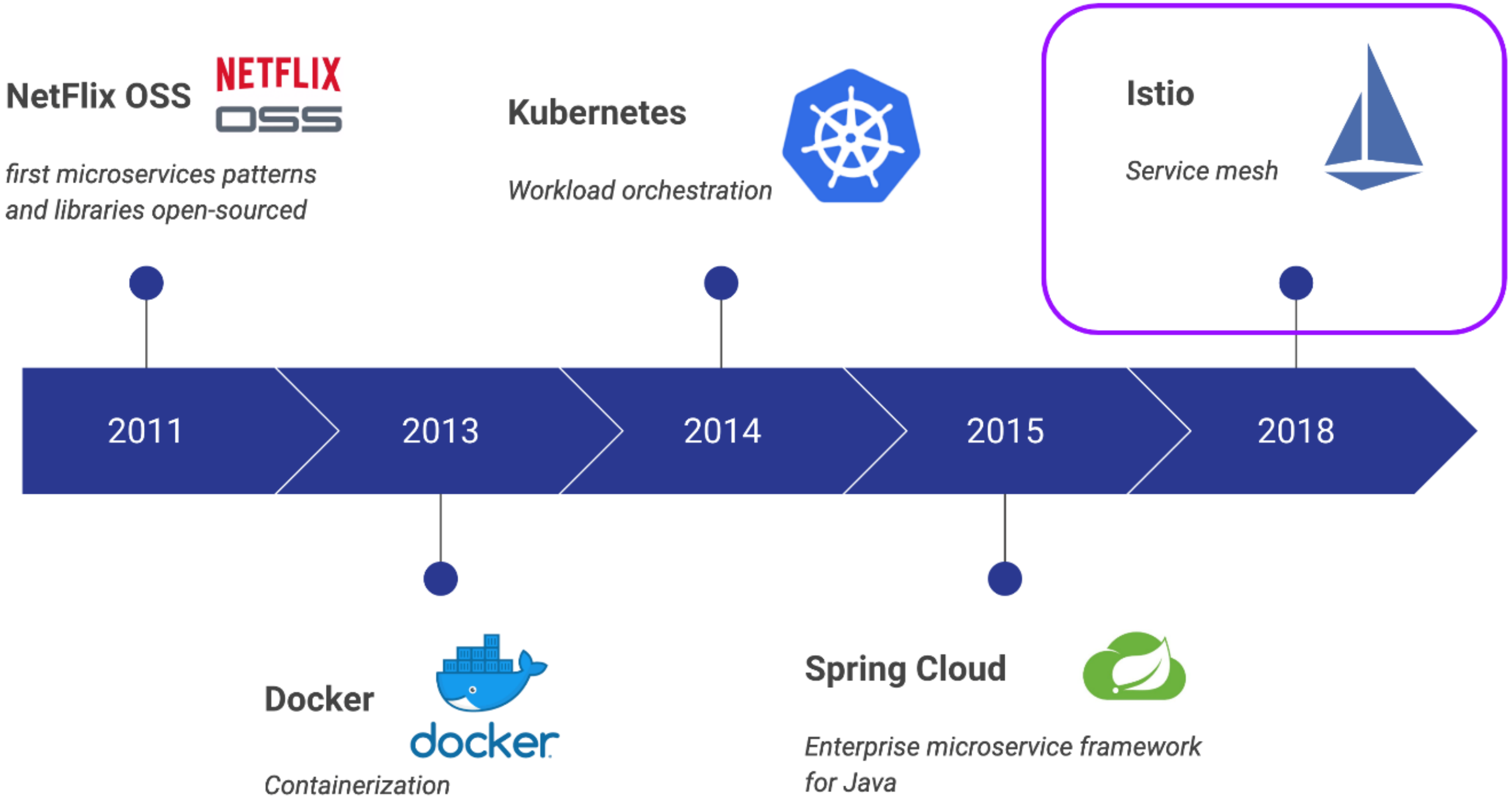
Monolithic vs Micro Service

Speaker Photo Will Be Placed Here



Evolution of microservices

Speaker Photo Will Be Placed Here



2011

NetFlix OSS

first microservices patterns and libraries open-sourced

2013

Docker

Containerization

2014

Kubernetes

Workload orchestration

2015

Spring Cloud

Enterprise microservice framework for Java

2018

Istio

Service mesh

Microservices Challenges

- ***Configuration Service***
- ***Service Registry / Discovery***
- ***Circuit Breaker / Retry***
- ***Rate Limiting***
- ***Event Driven Messaging (Async)***
- ***Audit***
- ***Load Balancing / Intelligent Routing***
- ***API Gateway***
- ***Authentication & Authorization***
- ***Monitoring***
- ***Distributed tracing***
- ***Log Aggregation***

Speaker Photo Will Be Placed Here

Service Mesh

Service Mesh

- A dedicated infrastructure layer to make service-to-service communication **fast, safe** and **reliable**
- A **transparent layer** on top of your services
- A way to make the **network** aware of application protocols like HTTP and gRPC
- An **observability** tool
- A **security** tool

Speaker Photo Will Be Placed Here

Requirements of Service Mesh

Speaker Photo Will Be Placed Here

Discovery

Rate limiting

Canary releases

A/B testing

Metrics

Monitoring

Failure recovery

Access control

End-to-end authentication

Istio

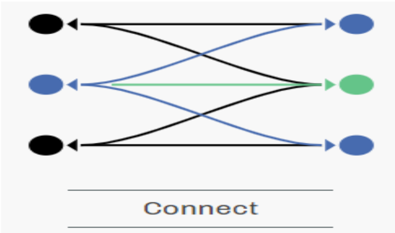
Istio

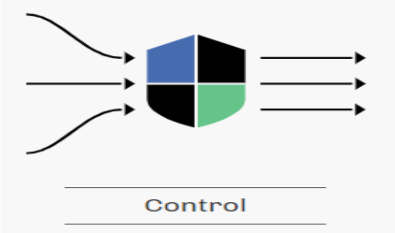
Speaker Photo Will Be Placed Here

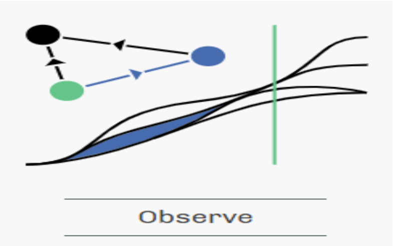
A service mesh designed to **connect, manage and secure** micro services

Open Source - Created by Google, IBM, and Lyft in 2017

Zero Code Changes







Why use Istio?

- Automatic **load balancing** for HTTP, gRPC, WebSocket and TCP traffic.
  - ** gRPC — a modern open-source high-performance RPC framework that can run in any environment

- Fine-grained control of **traffic behavior** with rich routing rules, retries, failovers and fault injection

- A pluggable policy layer and configuration API supporting **access controls**, rate limits and quotas.

- Automatic **metrics, logs and traces** for all traffic within a cluster, including cluster ingress and egress.
  - ** cluster ingress — a collection of rules that allow inbound connections to reach the cluster services.
  - ** cluster egress — a collection of rules that allow outbound connections.

- **Secure service-to-service communication** in a cluster with identity-based authentication and authorization.

Speaker Photo Will Be Placed Here

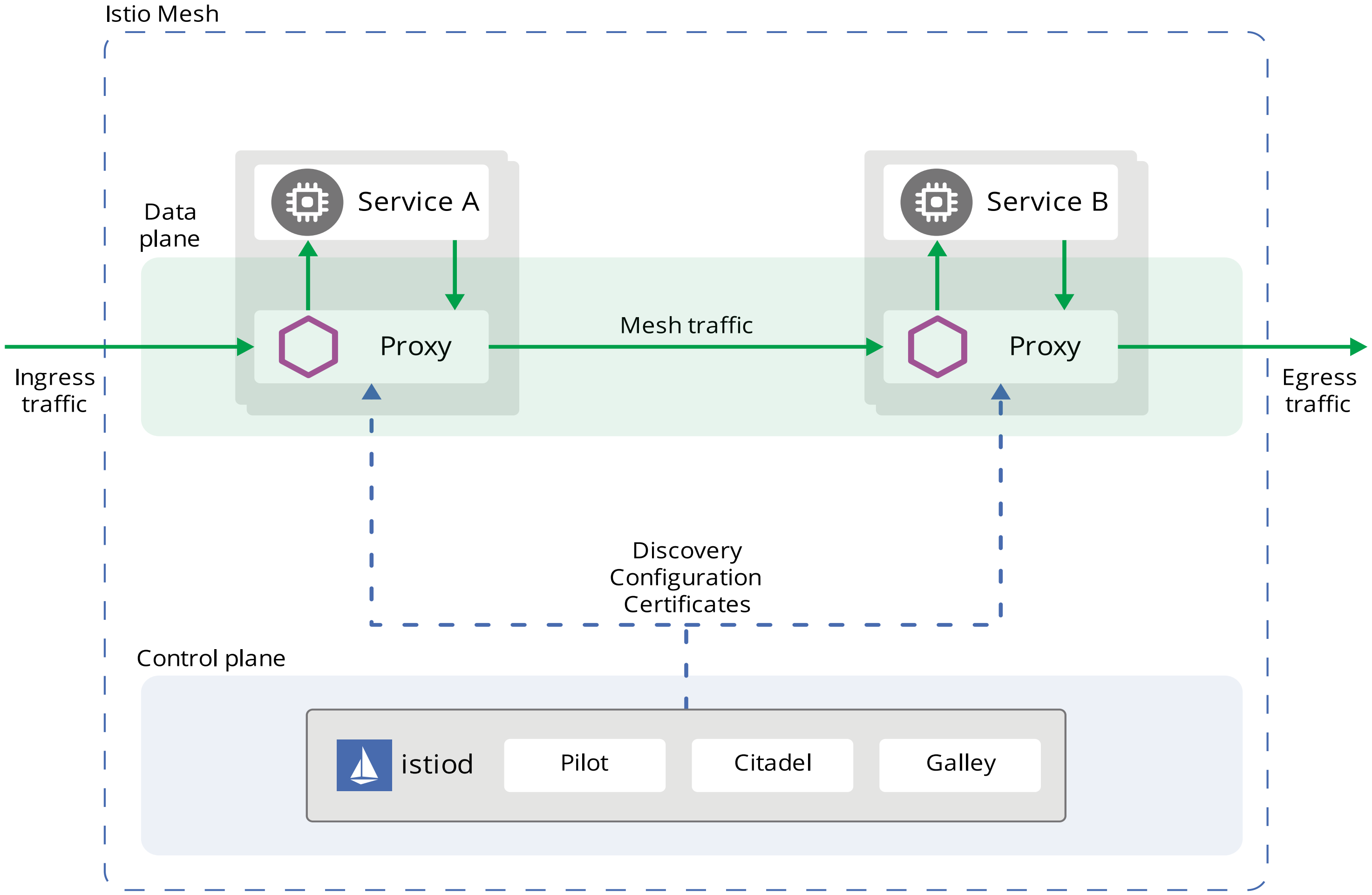
Who is Istio for?

Speaker Photo Will Be Placed Here

- **Infrastructure Operators:** Monitor traffic across clusters and regions, add failovers
- Platform Engineers: Build CI/CD tools for app developers, migrate legacy services
- **App Developers:** Investigate service metrics and behavior, debug during outages
- **Security Admins:** Enforce authentication and authorization policies
- **Quality Assurance:** Mirror production traffic to a test environment

Istio Architecture

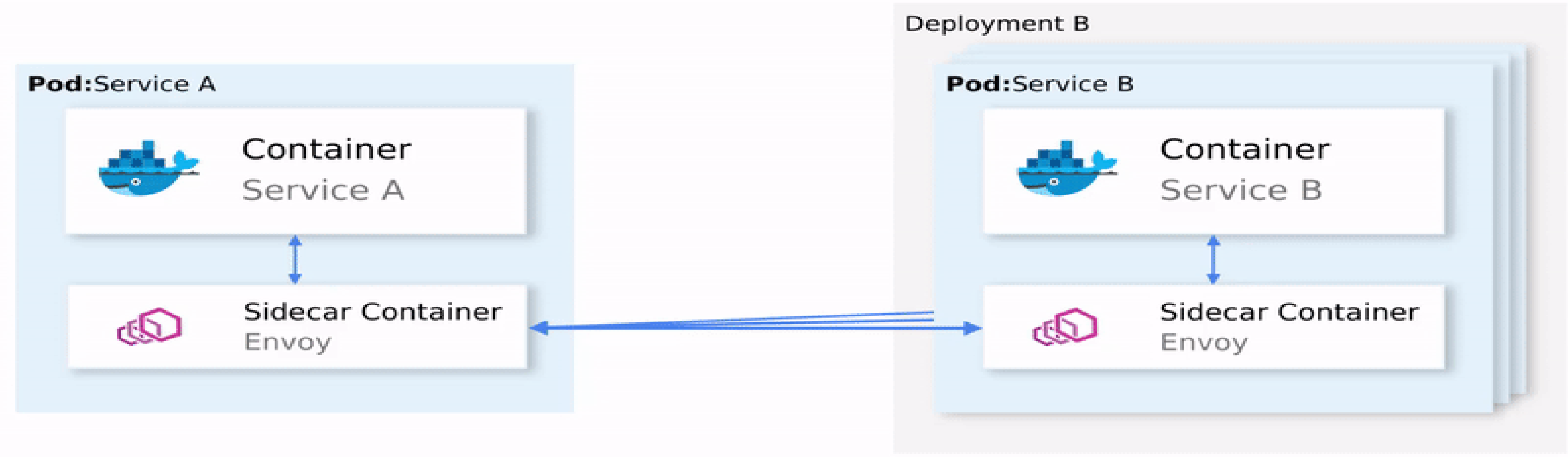
Speaker Photo Will Be Placed Here



Istio Architecture – The Data Plane

Speaker Photo Will Be Placed Here

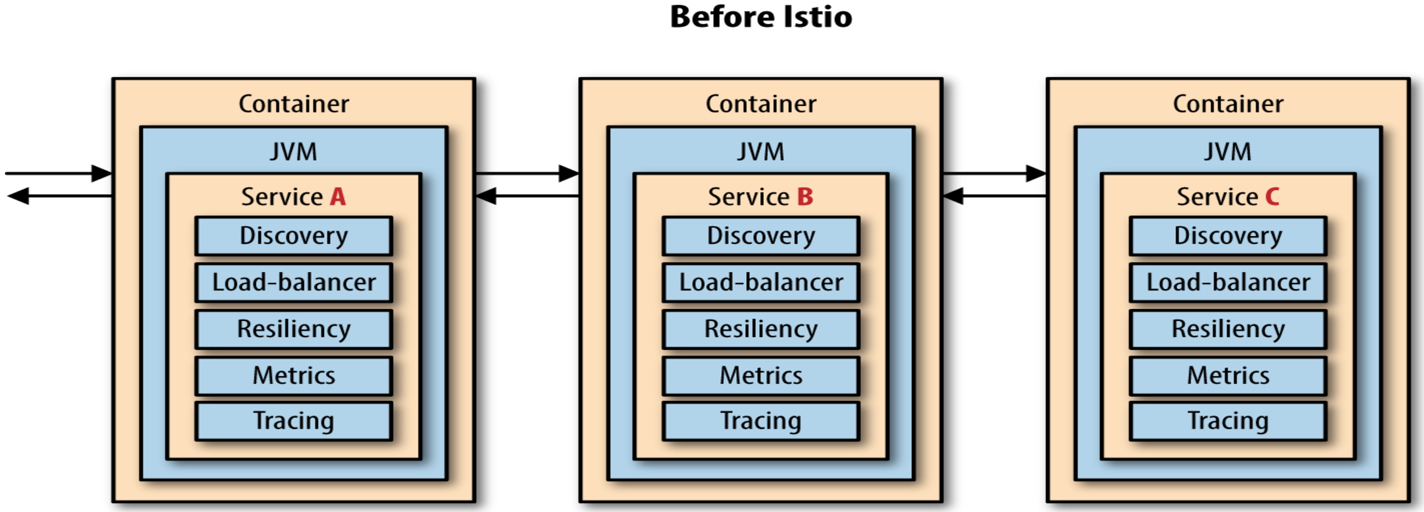
- Is composed of a set of intelligent proxies named envoy which is deployed as a sidecar.
- These proxies mediate and control all the network communication between micro-services along with Mixer (a general-purpose and telemetry hub)



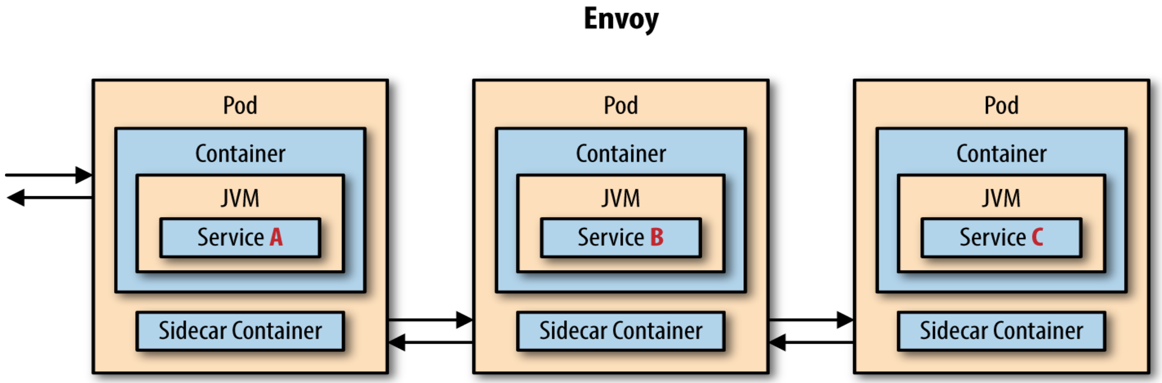
- ➢ C++ based L4/L7 proxy
- ➢ Low memory footprint
- ➢ Battle tested at Lyft
  - Runs with 100+ services
  - 10K VMs
  - 2M requests/sec

Istio Architecture – The Data Plane

Speaker Photo Will Be Placed Here



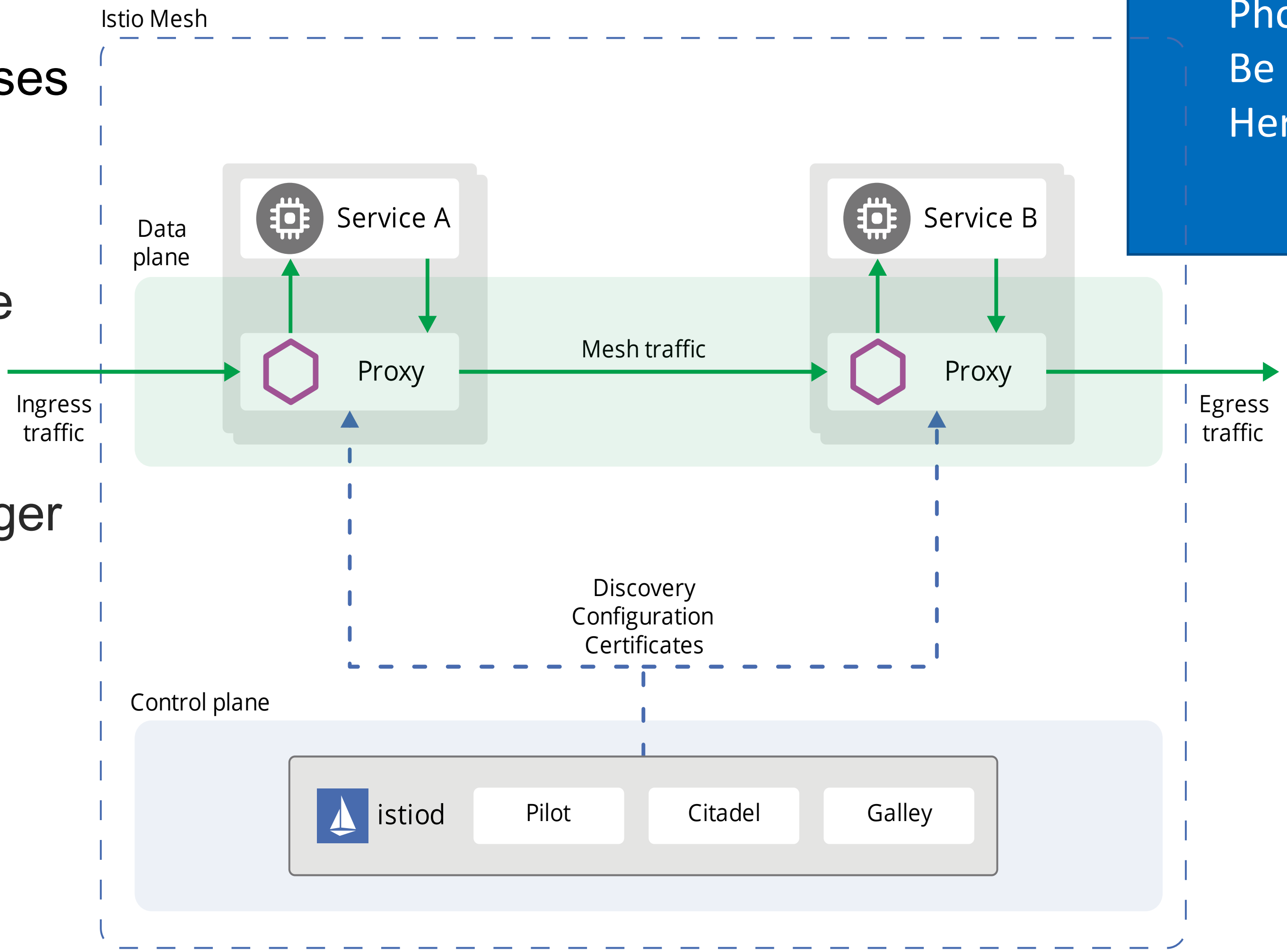
envoy



Istio Architecture – The Control Plane

Speaker Photo Will Be Placed Here

- Manages, Controls and Supervises the network of microservices.
- Istio Daemon composed of three components:
  - ➢Pilot – Driver of Istio
  - ➢Galley – Configuration Manager
  - ➢Citadel - Security Policy



Istio Core Features

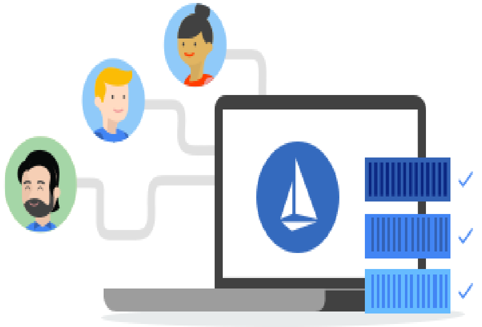
Istio Core Features

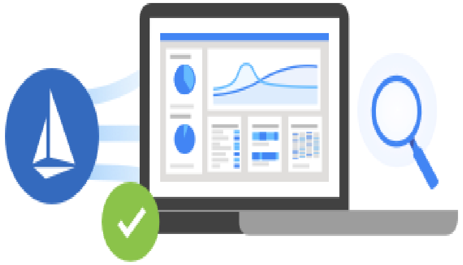
Speaker Photo Will Be Placed Here

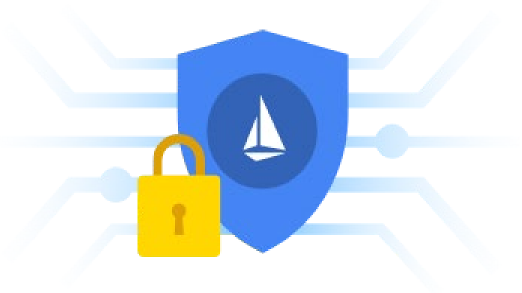
01 Traffic Management

02 Observability

03 Security



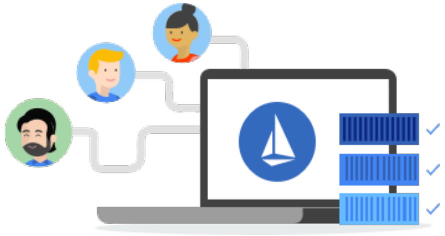




Istio Core Features

Speaker Photo Will Be Placed Here

01 Traffic Management



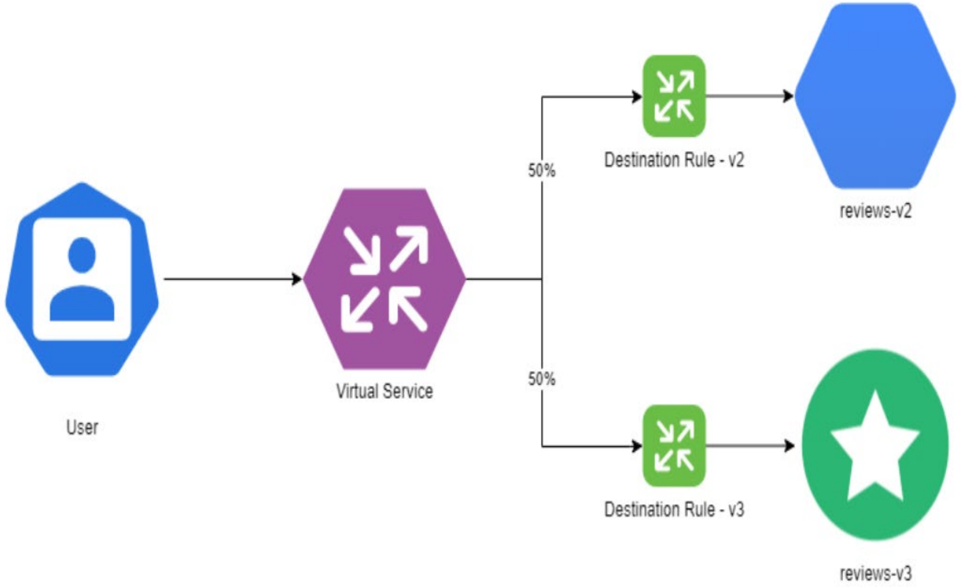
- Safe rollouts with traffic splitting
- Client-side load balancing
- Timeouts, retry, circuit-breaking

Virtual Service & Destination Rules

Speaker Photo Will Be Placed Here

Virtual Service: A Virtual Service defines a set of request routing rules that can be used to distribute traffic to different destinations in the service mesh.

Destination Rules: are rules applied to traffic after they have been routed to a destination by a virtual service.

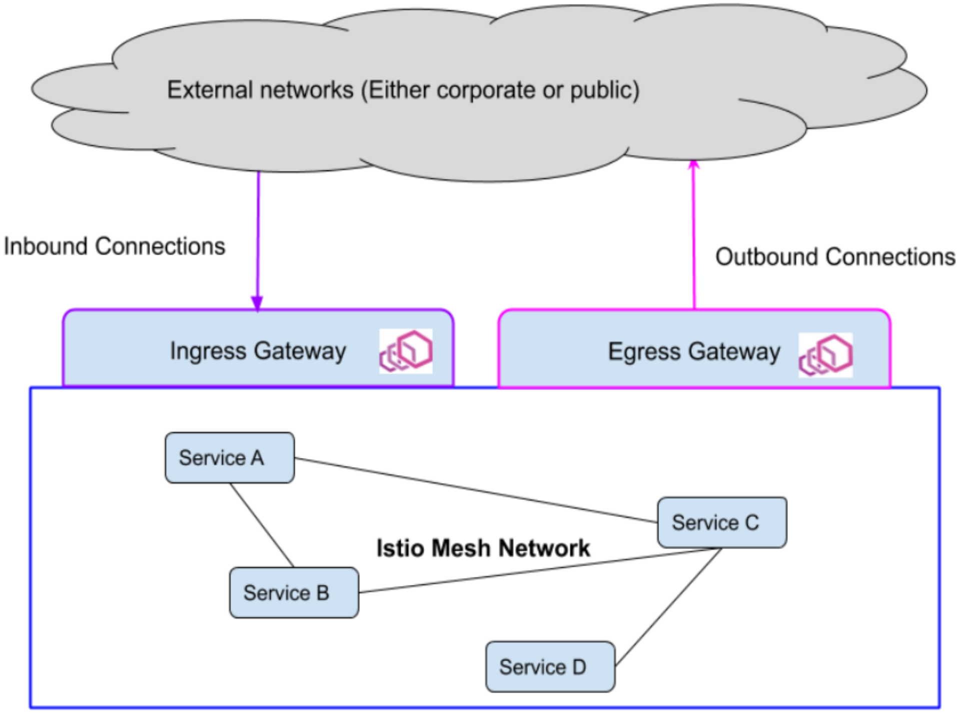


Gateway

Speaker Photo Will Be Placed Here

- Istio Gateway describes a load balancer operating at the edge of the mesh receiving incoming or outgoing HTTP/TCP connections. The specification describes a set of ports that should be exposed, the type of protocol to use, virtual host name to listen to, etc.

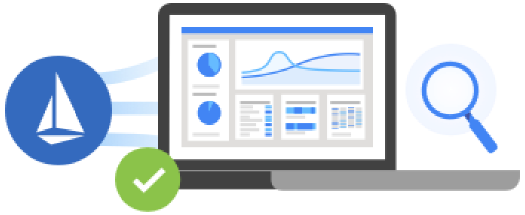
- Istio Gateway is based on envoy proxy.



Istio Core Features

Speaker Photo Will Be Placed Here

02 Observability



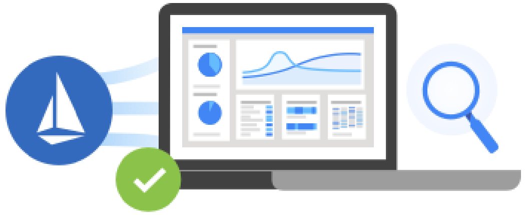
- Telemetry for every service
- Logs for all traffic
- Service graph

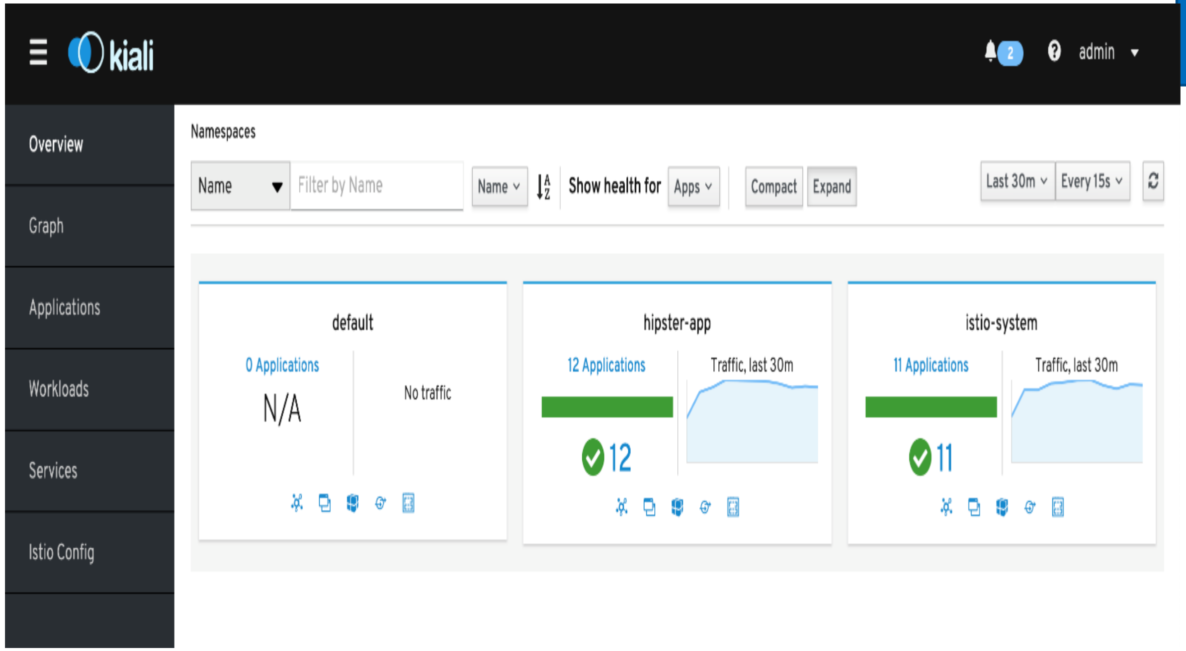
Istio Core Features

Speaker Photo Will Be Placed Here

Management Console

02 Observability





Namespaces

Name ▾ | Filter by Name | Name ⌄ | Show health for Apps ⌄ | Compact | Expand | Last 30m ⌄ | Every 15s ⌄

default
0 Applications
N/A
No traffic

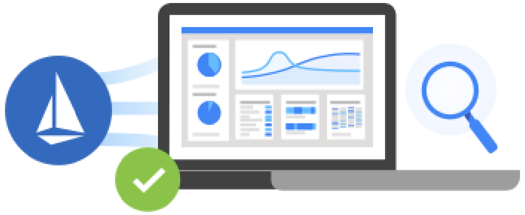
hipster-app
12 Applications
Traffic, last 30m
✓ 12

istio-system
11 Applications
Traffic, last 30m
✓ 11

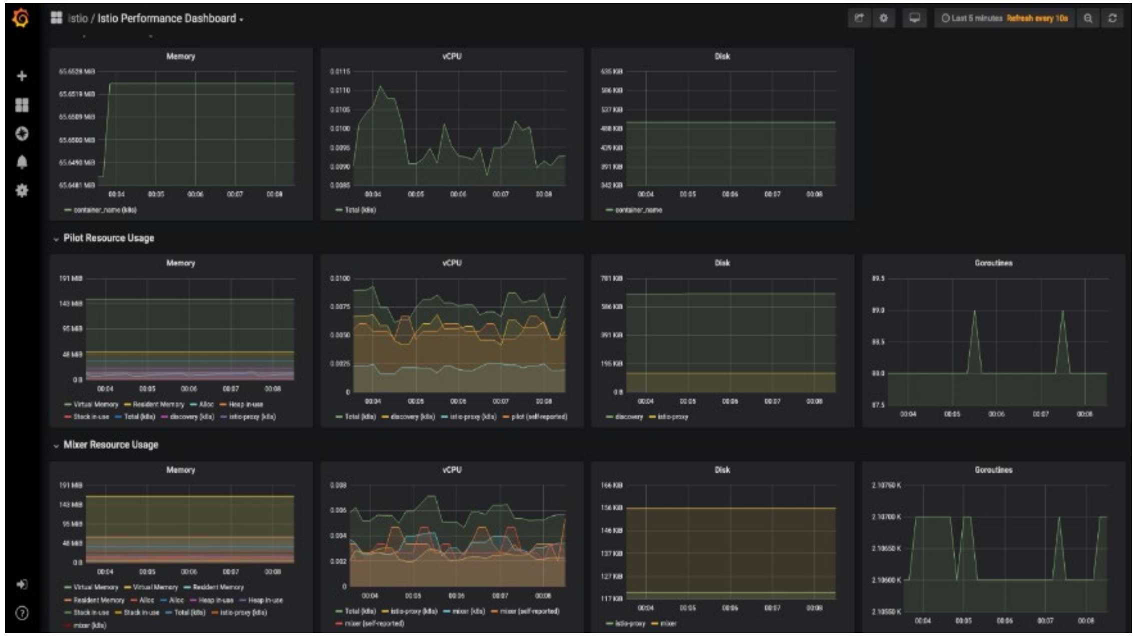
Istio Core Features

Speaker Photo Will Be Placed Here

02 Observability



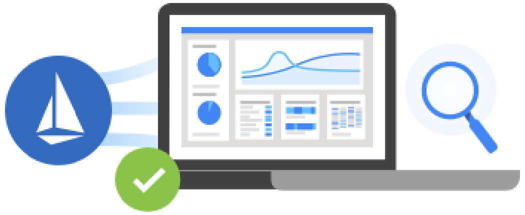
Network Traffic Topology



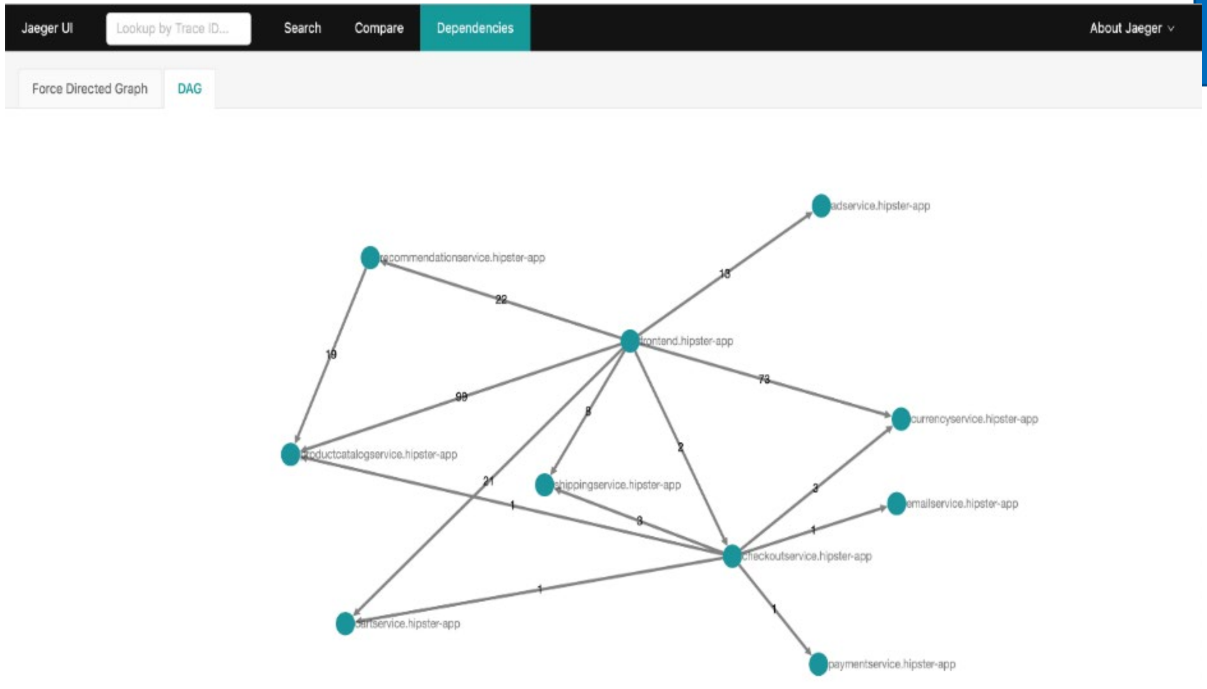
Istio Core Features

Speaker Photo Will Be Placed Here

02 Observability



Tracing & Troubleshooting



Istio Core Features

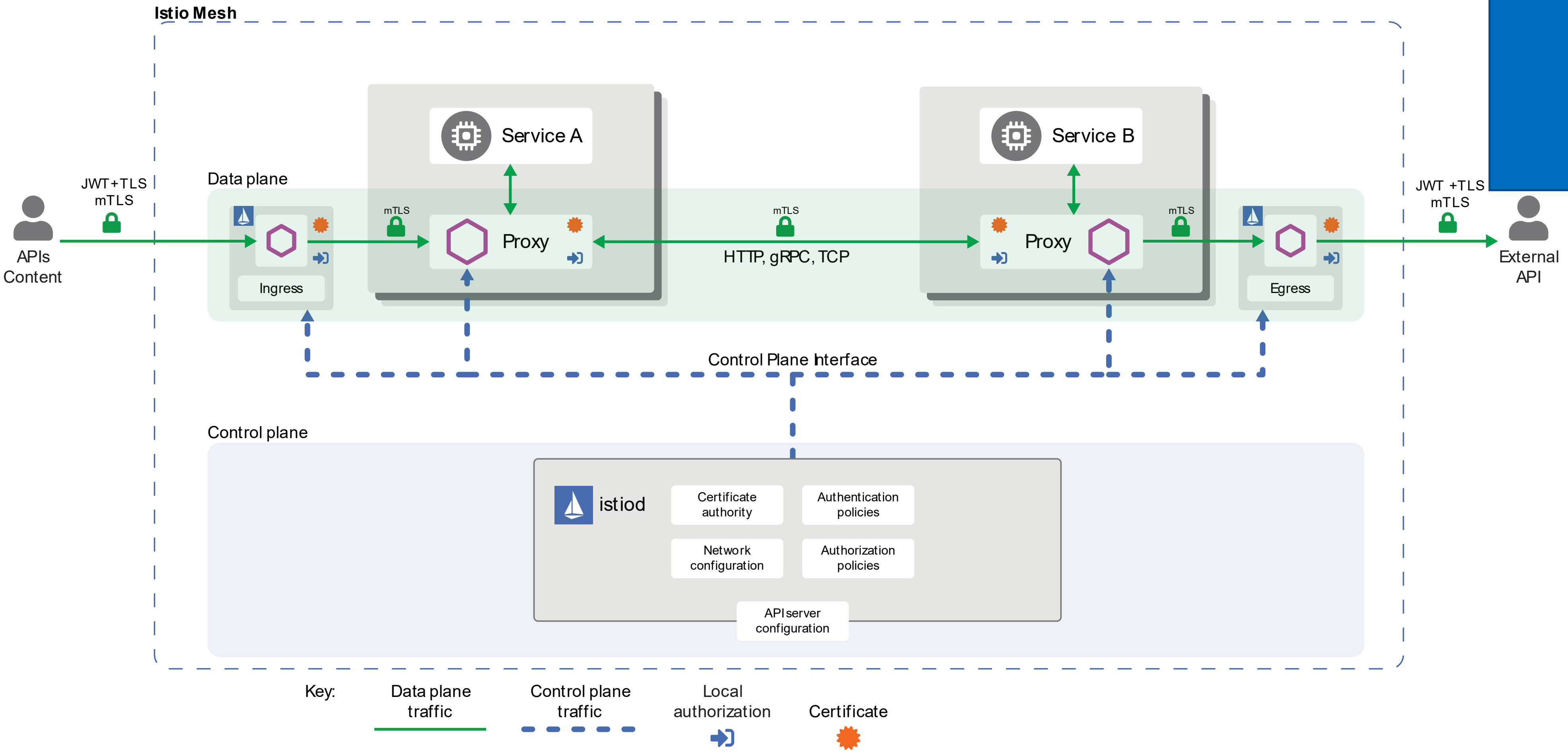
Speaker Photo Will Be Placed Here

03 Security

- Encryption in transit
- Service identity, authentication
- Authorization

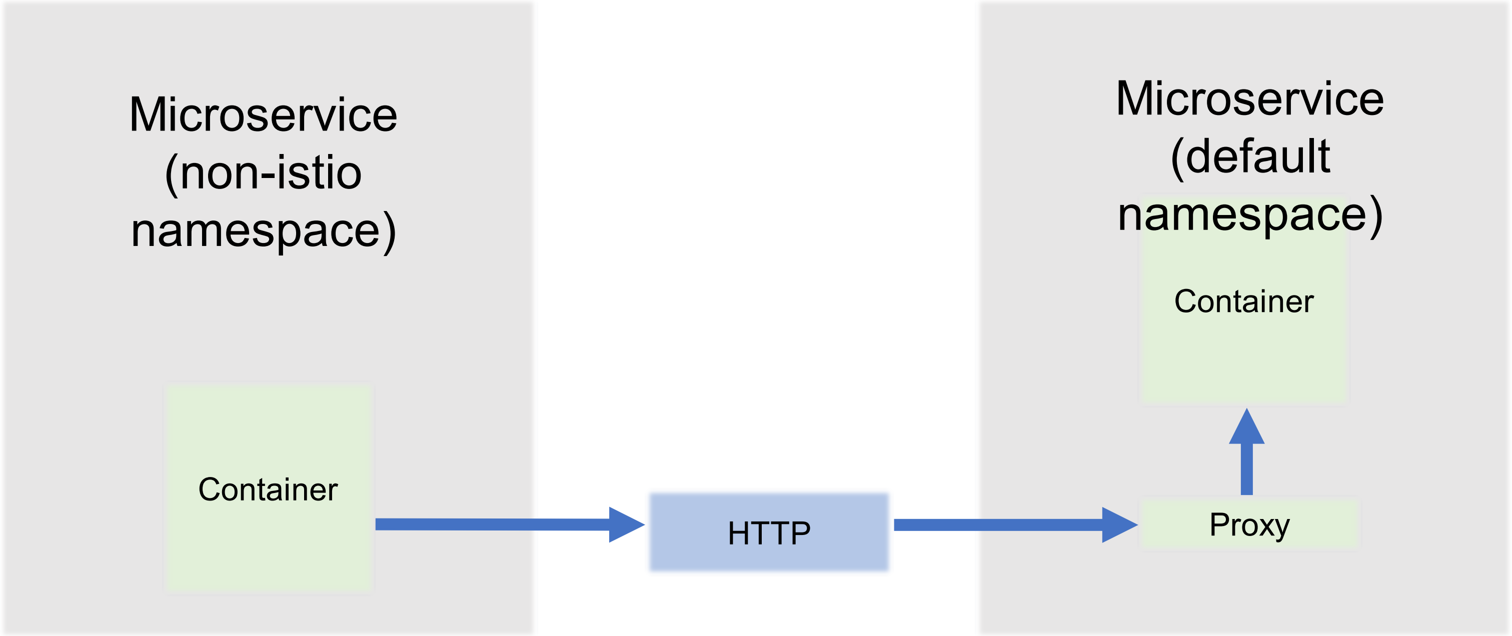
Istio Security

Speaker Photo Will Be Placed Here



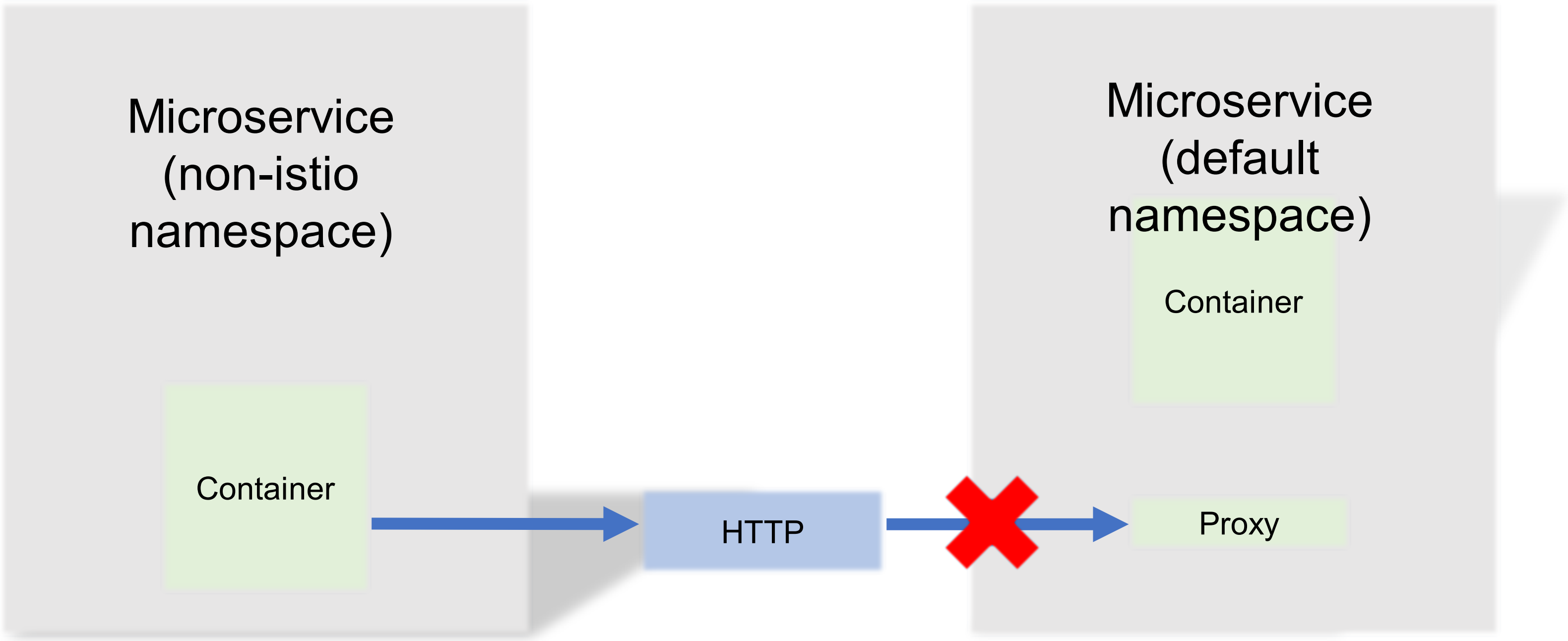
Permissive mTLS

Speaker Photo Will Be Placed Here


Microservice (non-istio namespace)
Container
HTTP
Microservice (default namespace)
Container
Proxy

Strict mTLS

Speaker Photo Will Be Placed Here


Microservice (non-istio namespace)
Container
HTTP
Microservice (default namespace)
Container
Proxy

Istio Key Take Away

Speaker Photo Will Be Placed Here

- Istio brings tools to DevOps and SRE to manage network concerns on behalf of Dev teams letting them focus on delivering business value.

- Istio tackles microservices concerns in a uniform and declarative way.

- This talk has enabled the participants with core features of the Istio technology, Traffic Management, Security, Monitoring and Observability.

echo $questions

Please take a moment to rate this session.

Your feedback is important to us.

STORAGE DEVELOPER CONFERENCE SDC 21