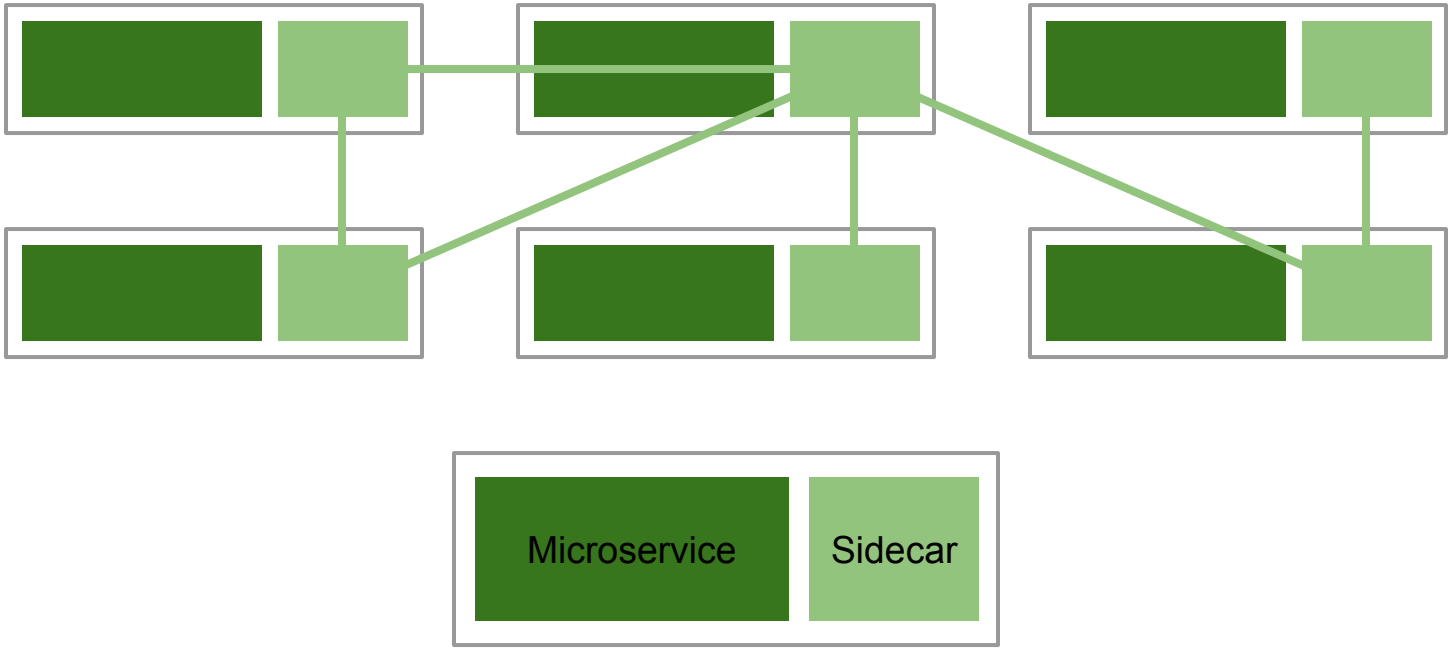
Securing Applications Running on Kubernetes With Istio

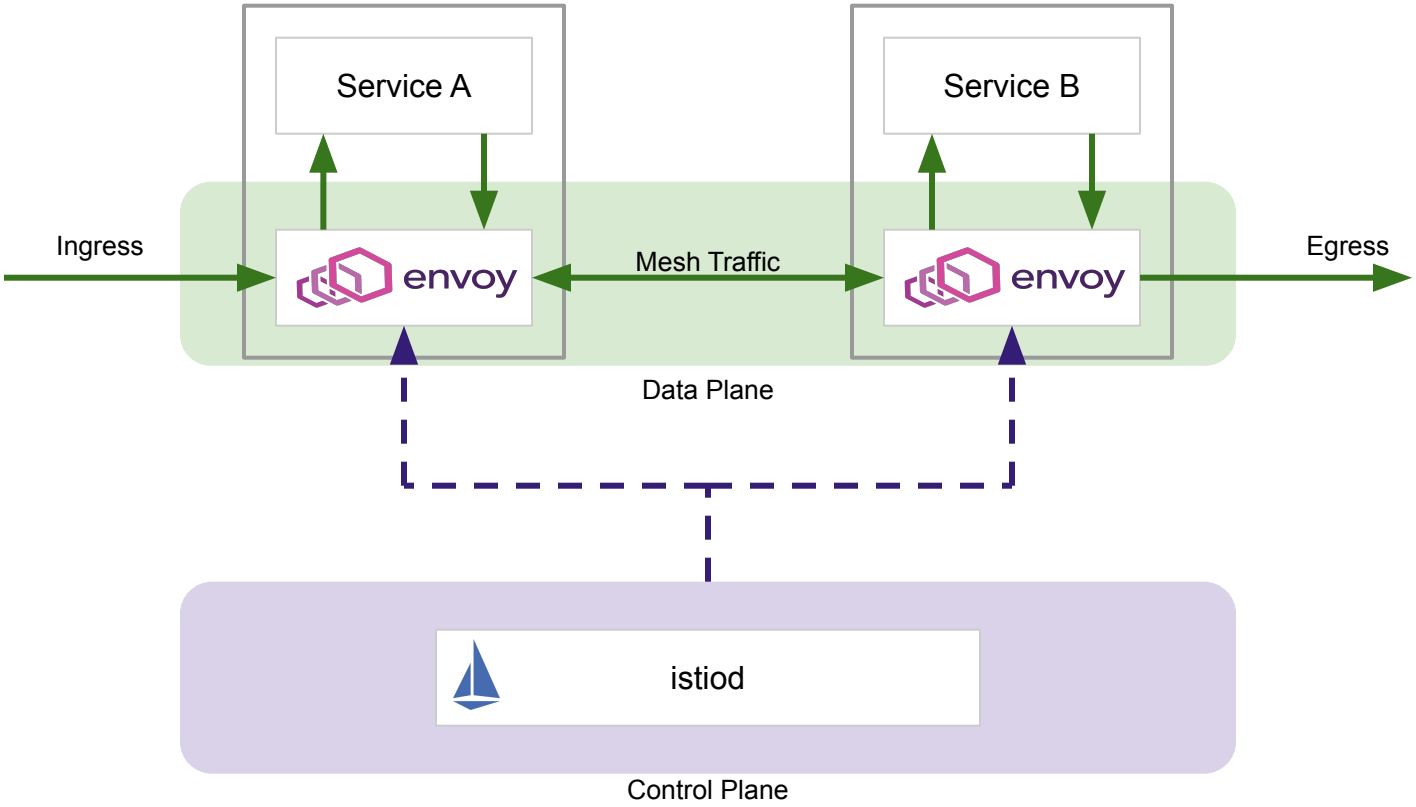
Agenda

- Service Mesh Introduction
- Istio
- Demo Application Architecture
- Demo

Service Mesh Introduction



Istio Architecture



Istio Features

- Traffic Management
- Observability
- Security

Istio Traffic Management

- Request Routing
- Request Timeouts
- Circuit Breakers
- Traffic Mirroring
- Traffic Shifting
- Fault/Latency Injection
- Blue/Green & Canary

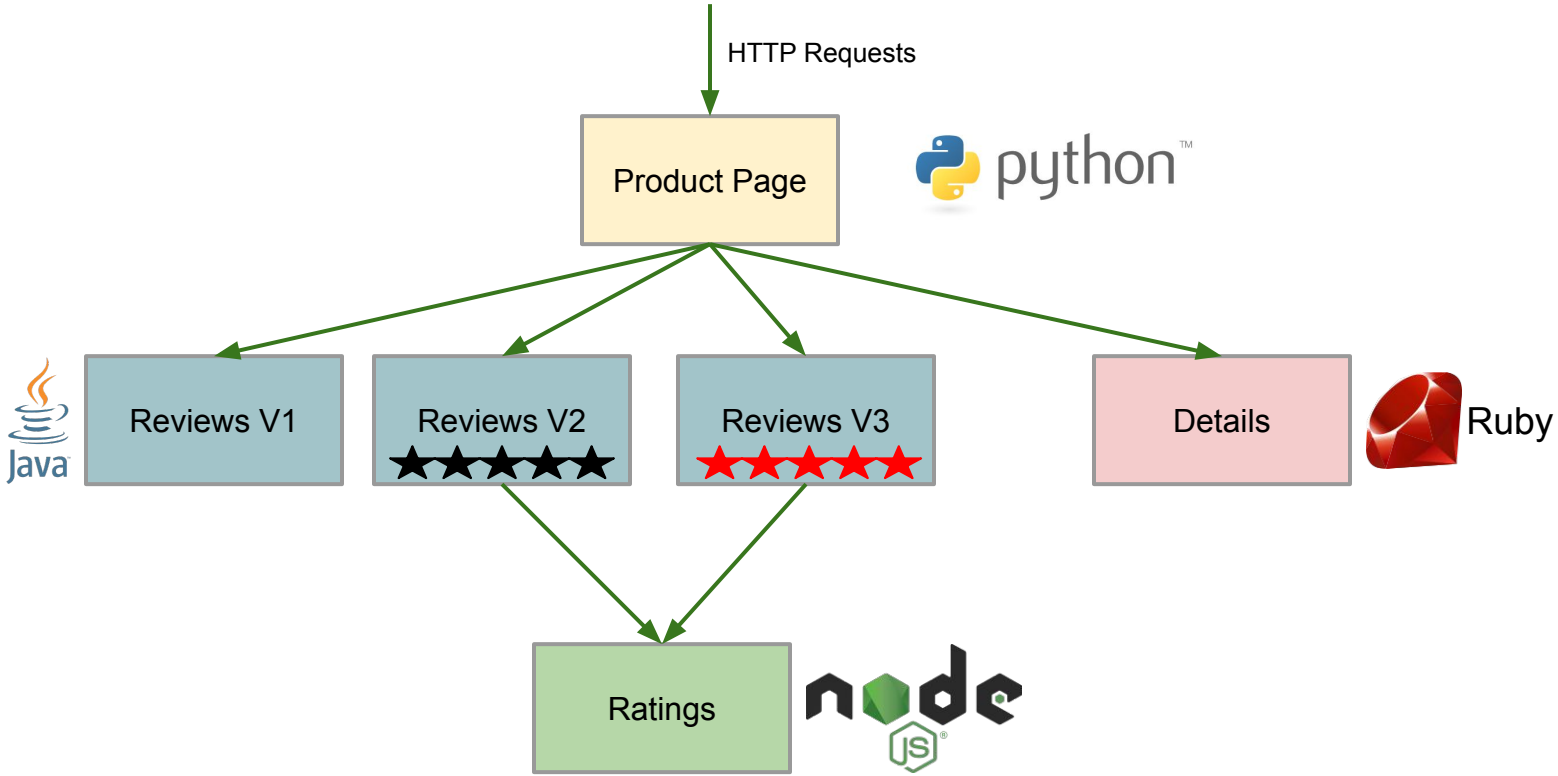
Istio Observability

- Metrics
- Tracing
- Access Logging

Istio Security

- Securing Ingress with Istio Gateway’s and TLS
- Securing Traffic Between Services With MTLS
- Authentication
- Authorization
- Securing Egress

Istio Bookinfo Demo Architecture



Additional Setup Information

- Docker for MacOS (Edge)
- Kubernetes 1.16.5
- Istio 1.5.2

Initial Setup (Step0)

- Verify that the Cluster is in the default state
- Verify that the default namespace has no pods running in it

Initial Setup (Step1)

- Install istio using istioctl
- Enable automatic sidecar injection in the default namespace
- Install the bookinfo application
- Serve the bookinfo site over HTTP

Securing Ingress with TLS (Step 2)

Add certificate for bookinfo application and configure gateway to service application over HTTPS

Securing Traffic Between Services With MTLS (Step3)

Enable STRICT mtls, which disallows mixed traffic. Verify that calls using MTLS pass, and calls not using MTLS fail

End User Authentication (Step 4)

Add Policy that requires all ingress traffic to contain a valid JSON Web Token (JWT)

Authorization (Step 5)

Add Authorization Policy that allows requests that have the correct request principal and group claim

Securing Egress (Step 6)

Setup an egress gateway, directing all egress traffic through it

Network Policy (Step 7)

- Requires a CNI that supports them

Thank You!

Aaron Mell
aaronamell@gmail.com
github.com/aaronmell
Kubernetes Slack: aaronmell