Service Mesh Introduction

Tetrate

Vikas Choudhary
https://www.linkedin.com/in/vikaschoudhary16/

Agenda

Tetrate

- Why Service Mesh? What is the Problem?
- What Service Mesh provides?
- What is Service Mesh
- Istio & Envoy Introduction

Why Service Mesh?

The Problem

IT’s shift to a modern distributed architecture has left enterprises unable to **connect, monitor, manage, or secure** their services in a consistent way.

Modern distributed architecture

↓

container based services
deployed into dynamic environments
composed via the network

Service Mesh is the Solution

The goal of a service mesh is to move the functionality required to connect, monitor, manage, and secure service communication out of the application, so application developers don’t need to worry about it.

- Consistency across the fleet
- Centralized control
- Fast to change (update config to affect change, not redeploy)

Connect

Get the network out of the application.

- Service Discovery(across multiple clusters, on BMs, on VMs)
- Resiliency
    - retry, outlier detection, circuit breaking, timeouts, etc.
- (Client Side) Load Balancing

Monitor

Understand what’s actually happening in your deployment.

- Metrics
- Logs
- Tracing
- Topology

Manage

Control which requests are allowed and how & where they flow.

- Fine-grained traffic control
  - L7, not L4!
  - Route by headers, destination or source ID, etc

Secure

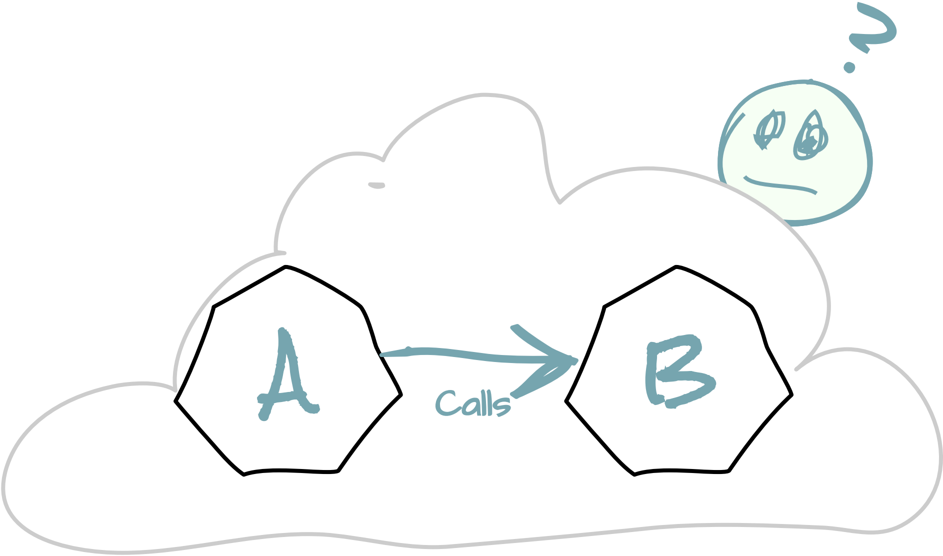
Elevate security out of the network.

- (L7) Workload Identity
  - IP:port is not an identity
  - Reachability != Authorization
- Service-to-Service Authentication and Authorization

Basics of Istio

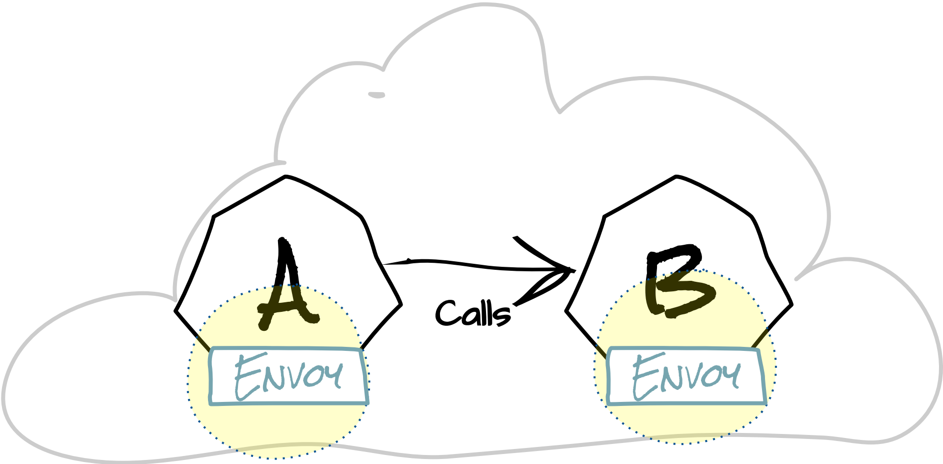
How Istio Works

A story as old as time: Service A meets Service B.



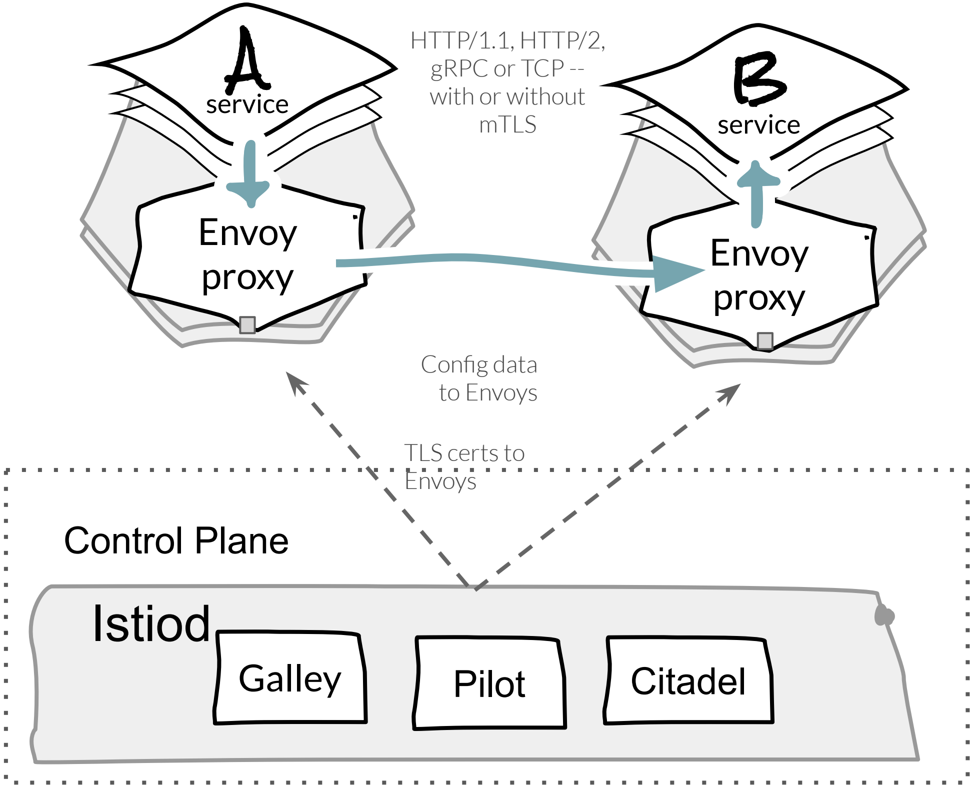
How Istio Works

Deploy a proxy (Envoy) beside your application (“sidecar deployment”).



Istio Architecture

Tetrate



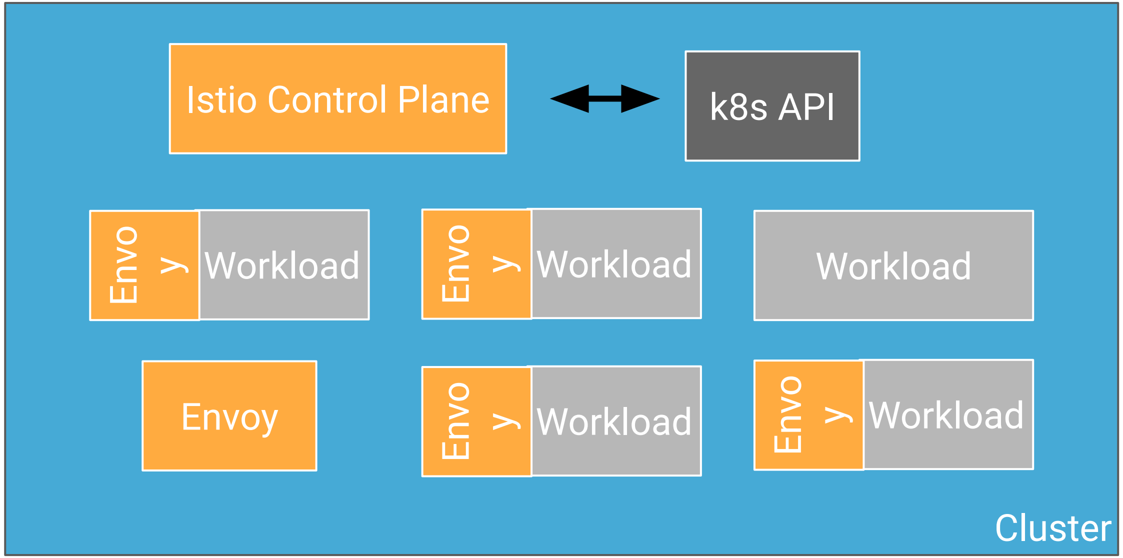
ISTIOD:

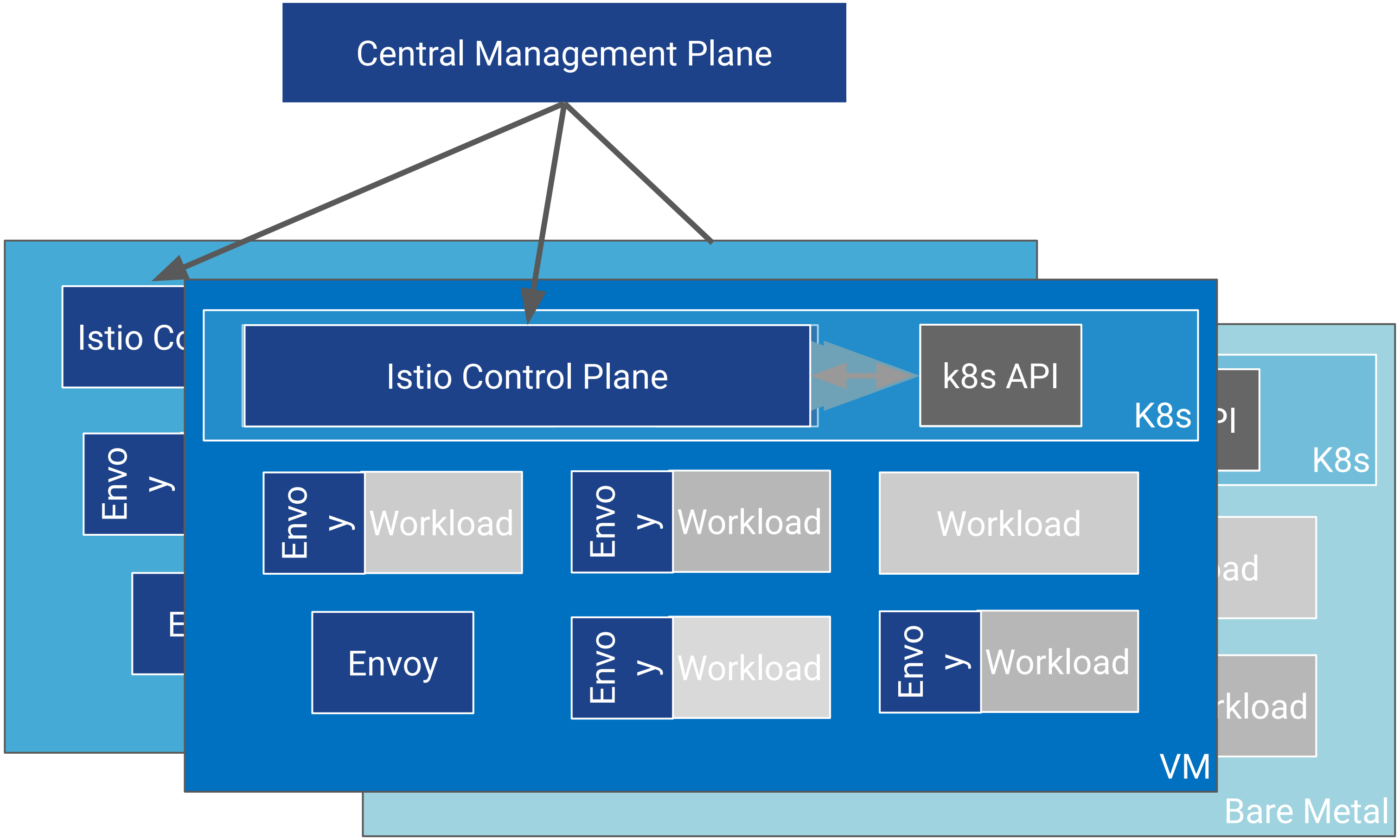
PILOT – Control plane to configure and push service communication policies.

CITADEL – Service-to-service auth[n,z] using mutual TLS, with built-in identity and credential management.

GALLEY – Configuration validation, distribution

ENVOY – Network proxy to intercept communication and apply policies.


Istio Control Plane
k8s API
Envoy
Workload
Envoy
Workload
Workload
Envoy
Envoy
Workload
Envoy
Workload
Cluster

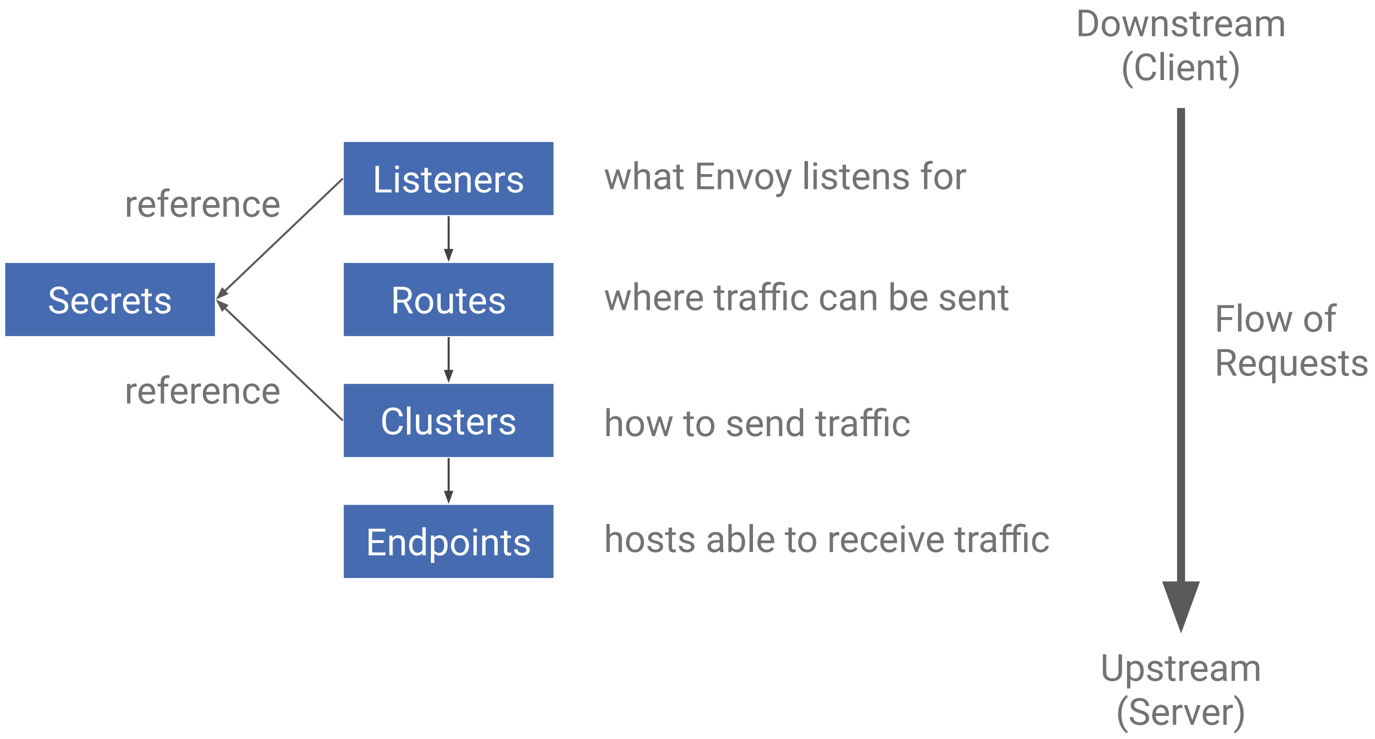

Central Management Plane
Istio Co
Envo y
E
Istio Control Plane
k8s API
K8s
PI
K8s
Envo y
Workload
Envo y
Workload
Workload
oad
Envoy
Envo y
Workload
Envo y
Workload
rkload
VM
Bare Metal

Envoy Proxy - Programmability

Tetrate

- Configured with Discovery Services
  - Collectively “xDS APIs”
    - Listener (LDS)
    - Route (RDS)
    - Cluster (CDS)
    - Endpoint (EDS)
    - Secret (SDS)
  - *Aggregated Discovery Service (ADS)* for delivering the data for each xDS API over a single pipe => ordered configuration delivery
- Push based model
  - As Pilot’s internal model changes, it computes affected Envoys and pushes updated config to them
  - Bootstrap config cannot be overridden by push

Envoy Proxy - Conceptual Configuration Model



Downstream
(Client)

Listeners — what Envoy listens for

Routes — where traffic can be sent

Clusters — how to send traffic

Endpoints — hosts able to receive traffic

Secrets

reference

reference

Flow of Requests

Upstream
(Server)

References:

- https://istio.io/
- https://www.envoyproxy.io/
- https://www.tetrate.io/resources/

Thank You!!!