Kubernetes Service Mesh

An Introduction to **Istio**

Joe Searcy
Platform Engineering

T · ·Mobile·

What is a “Service Mesh”?

- When two services need to communicate:
  - Figure out who to talk
  - Are they able to talk
  - Are we sure they are who they say they are
  - Are we allowed to talk to them
  - How long did it take to talk to them
  - Tell someone we talked to them

What is a “Service Mesh”?

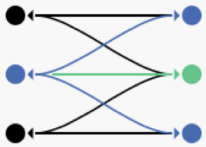
- When two services need to communicate:
  - Figure out who to talk to – **Service Discovery**
  - Are they able to talk – **Active/Passive Health Checking**
  - Are we sure they are who they say they are – **Authentication/Verification**
  - Are we allowed to talk to them - **Authorization**
  - How long did it take to talk to them – **Metrics/Tracing**
  - Tell someone we talked to them - **Logging**

What is Istio?

- An Open Source Service Mesh

Istio

Connect, secure, control, and observe services.

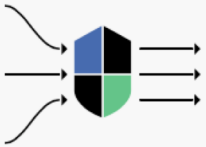


Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.

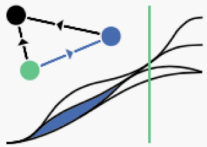
Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.



Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

What is Istio?

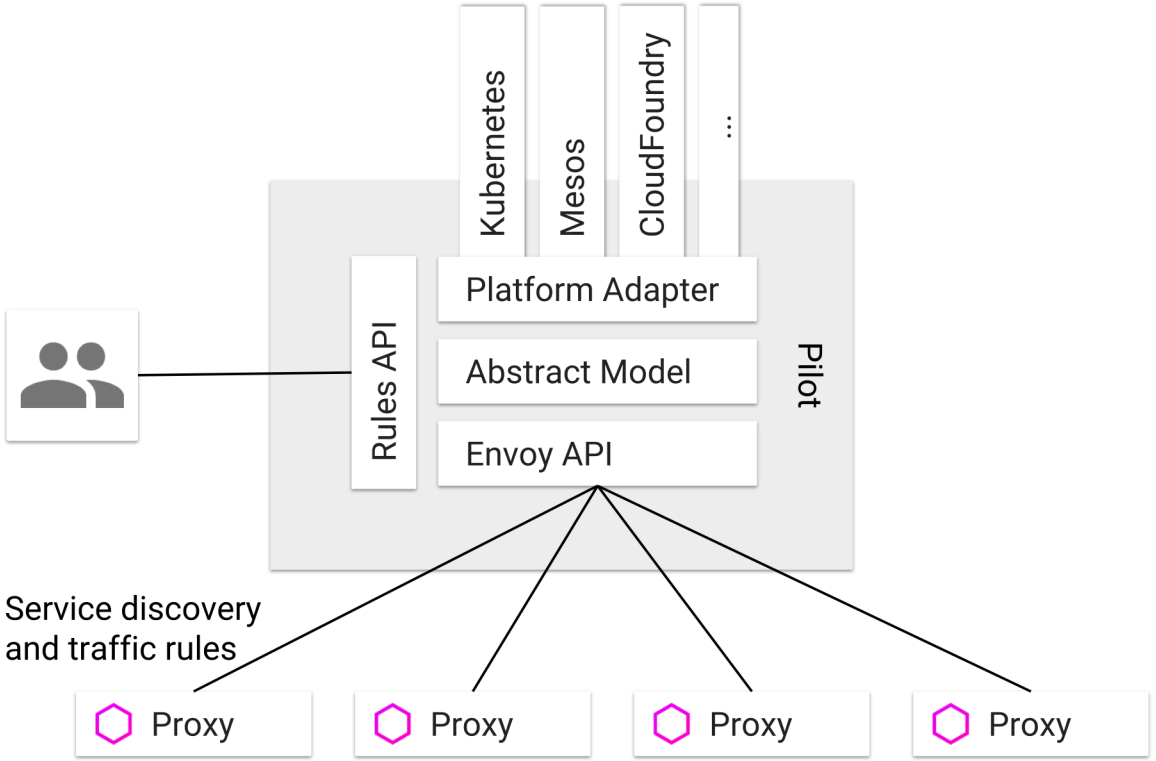
Architecture

- **Control Plane**
  - Pilot
  - Mixer
  - Citadel
  - Galley
- **Data Plane**
  - Envoy
  - Nginx *
  - Linkerd *

* Proxies that have been proven to integrate with Istio for data plane functionality. Still Experimental.

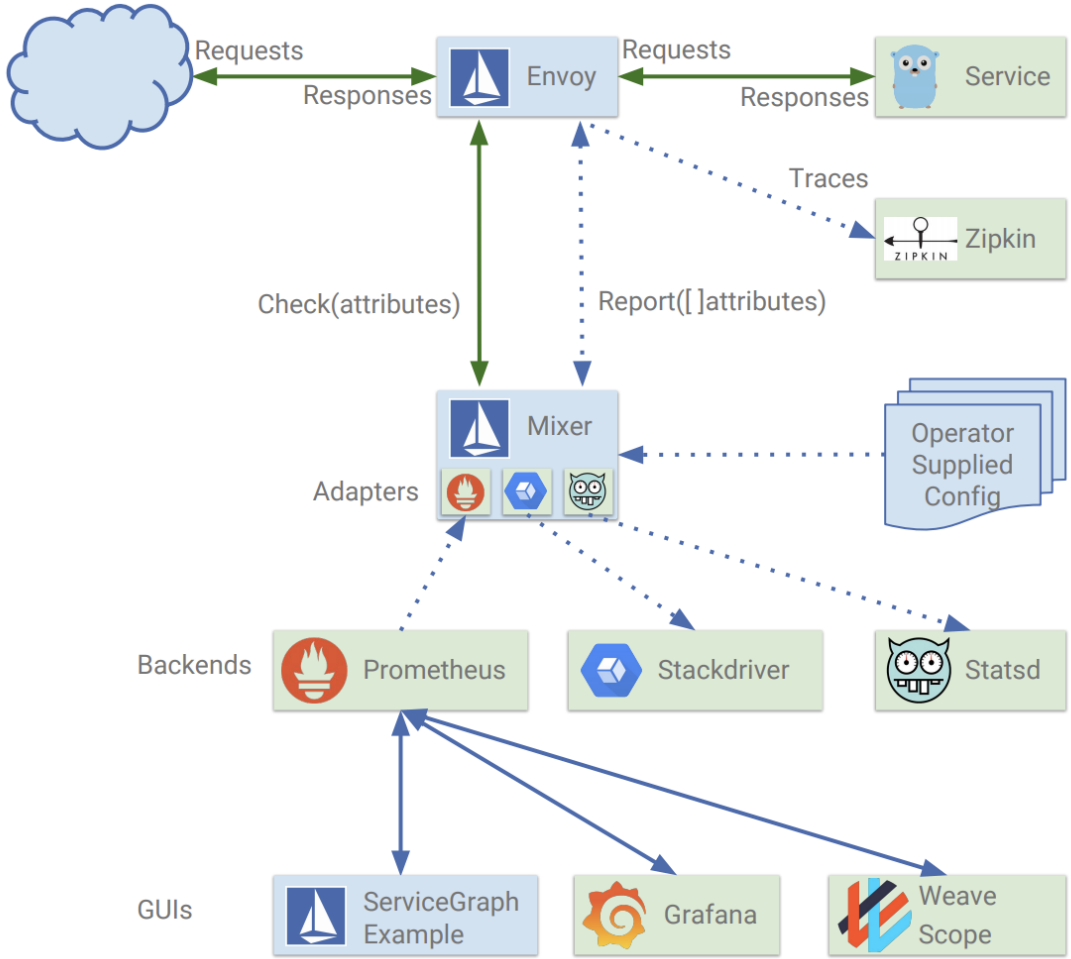
Istio Control Plane

Pilot provides service discovery for the Envoy sidecars, traffic management capabilities for intelligent routing



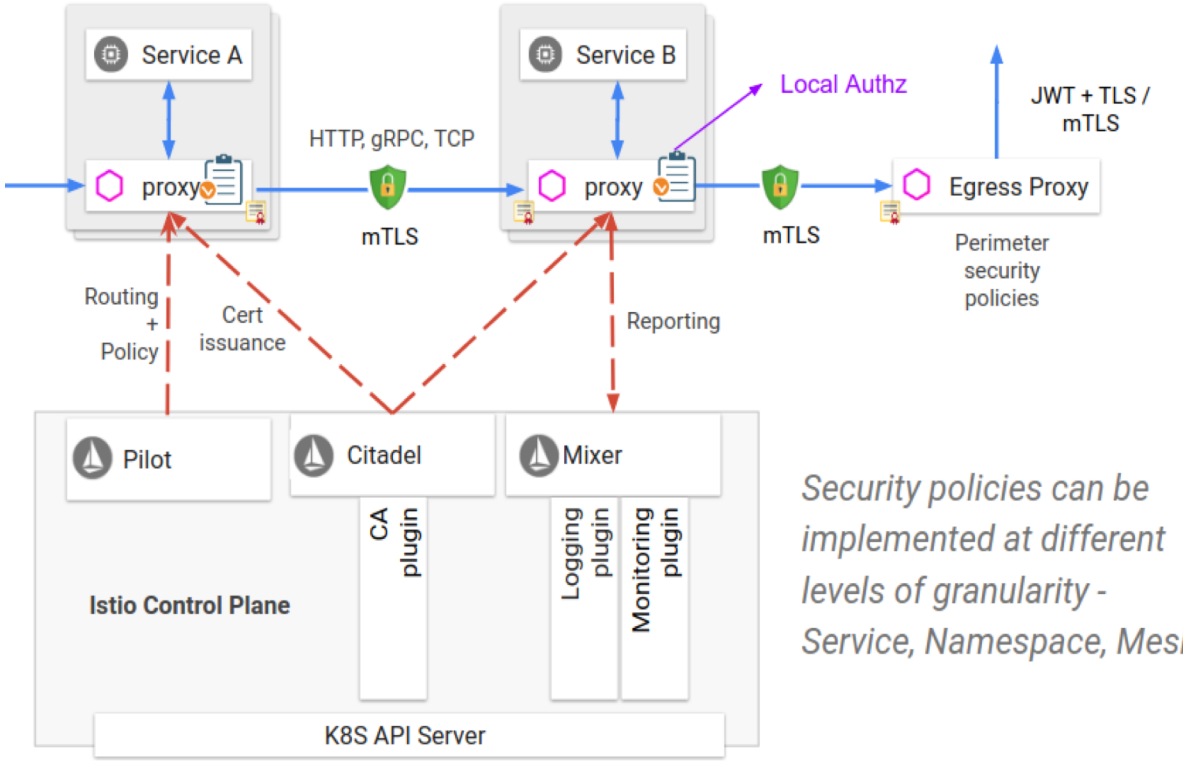
Istio Control Plane

Mixer enforces access control and usage policies across the service mesh, and collects telemetry data from the proxy and other services.



Istio Control Plane

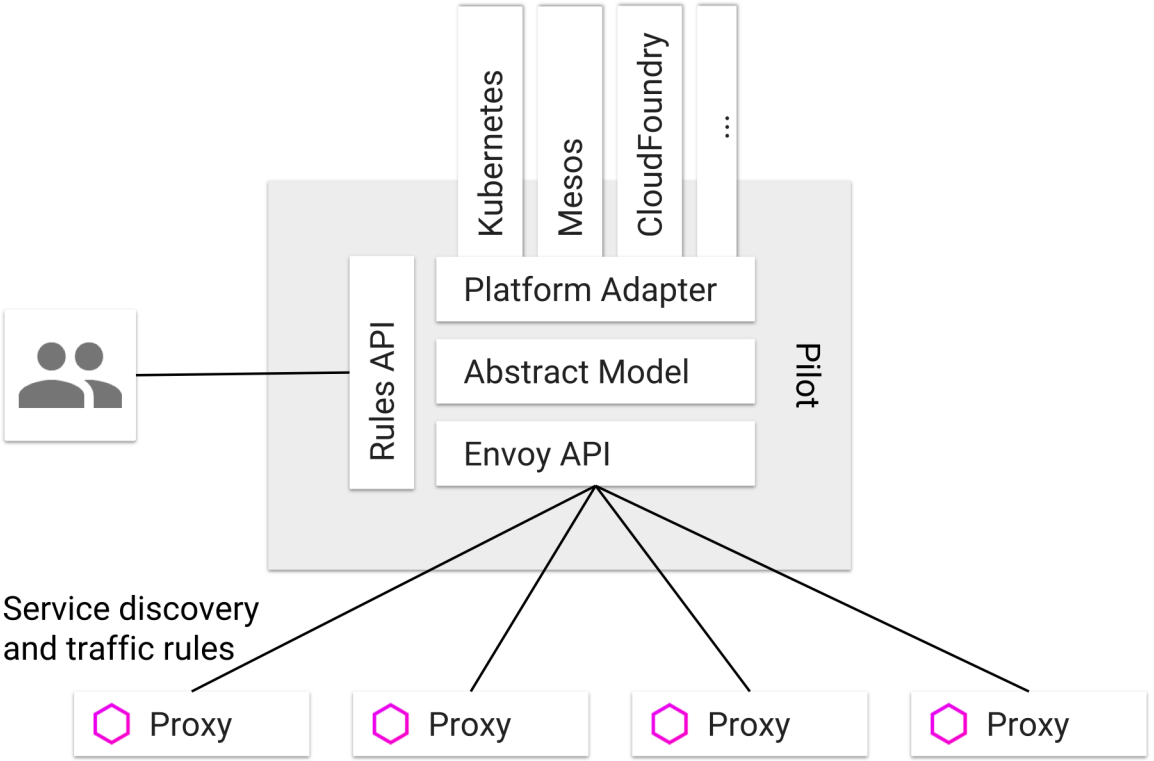
Citadel provides strong service-to-service and end-user authentication with built-in identity and credential management



Security policies can be implemented at different levels of granularity - Service, Namespace, Mesh.

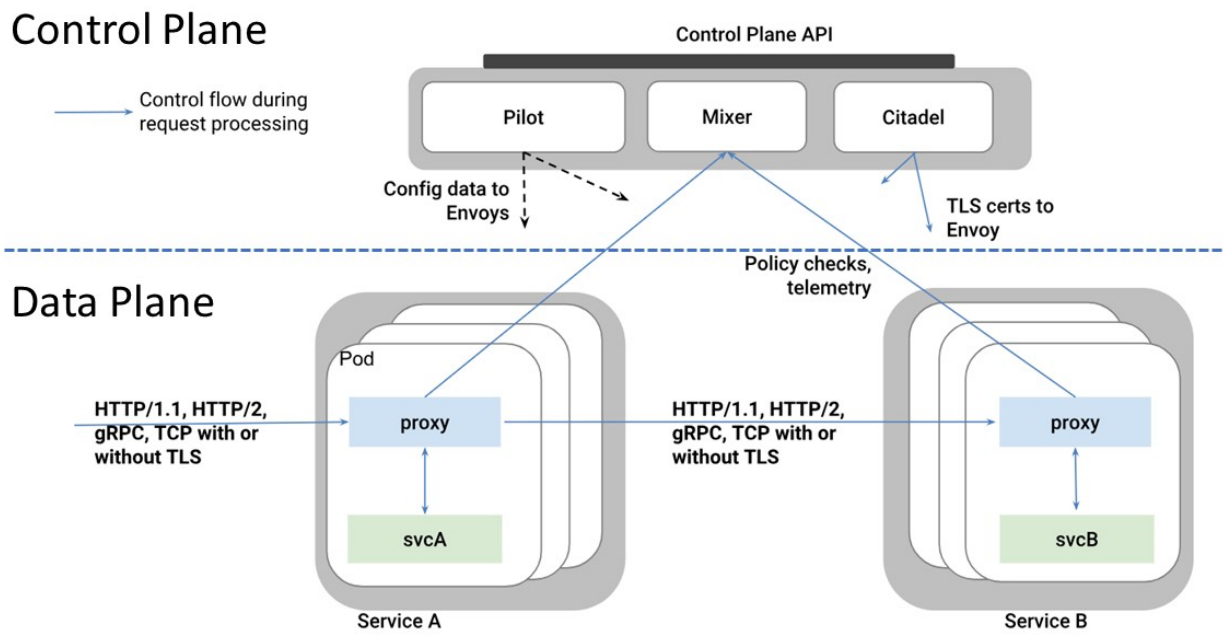
Istio Control Plane

Galley validates user authored Istio API configuration on behalf of the other Istio control plane components.



Istio Data Plane

ENVOY is an open source edge and service proxy designed for cloud-native applications



What is Istio?

TL;DR

Istio is a control plane for **Envoy**

Istio - Installation

```
GAATLMAC0509877:istio jsearcy3$ kubectl get pods -n istio-system
NAME                                      READY     STATUS      RESTARTS   AGE
istio-citadel-84fb7985bf-dcjc9            1/1       Running     0          21h
istio-cleanup-secrets-r25x5               0/1       Completed   0          21h
istio-egressgateway-bd9fb967d-6jpdj       1/1       Running     0          21h
istio-galley-655c4f9ccd-v9lcv             1/1       Running     0          21h
istio-ingress-56795fd96c-7pnl8            1/1       Running     0          21h
istio-ingressgateway-688865c5f7-q59fj     1/1       Running     0          21h
istio-pilot-6cd69dc444-mpbr9              2/2       Running     0          21h
istio-policy-6b9f4697d-tfjpn              2/2       Running     0          21h
istio-sidecar-injector-8975849b4-52ckc    1/1       Running     0          21h
istio-statsd-prom-bridge-7f44bb5ddb-qm68t 1/1       Running     0          21h
istio-telemetry-6b5579595f-jn2qz          2/2       Running     0          21h
prometheus-84bd4b9796-4q4cc               1/1       Running     0          21h
```

Istio - Installation

```
GAATLMAC0509877:istio jsearcy3$ kubectl get pods -n istio-system -l istio=pilot
NAME                            READY     STATUS    RESTARTS   AGE
istio-pilot-6cd69dc444-mpbr9    2/2       Running   0          21h
GAATLMAC0509877:istio jsearcy3$ kubectl get pods -n istio-system -l istio=mixer
NAME                               READY     STATUS    RESTARTS   AGE
istio-policy-6b9f4697d-tfjpn       2/2       Running   0          22h
istio-telemetry-6b5579595f-jn2qz   2/2       Running   0          22h
GAATLMAC0509877:istio jsearcy3$ kubectl get pods -n istio-system -l istio=citadel
NAME                              READY     STATUS    RESTARTS   AGE
istio-citadel-84fb7985bf-dcjc9    1/1       Running   0          22h
GAATLMAC0509877:istio jsearcy3$ kubectl get pods -n istio-system -l istio=galley
NAME                              READY     STATUS    RESTARTS   AGE
istio-galley-655c4f9ccd-v9lcv     1/1       Running   0          22h
```

PILOT

MIXER

CITADEL

GALLEY

Istio – Sidecar Proxy Injection

```
$ kubectl label namespace default istio-injection=enabled
```

```
$ kubectl describe pods -n default httpbin-77647f7b59-nrtgb
```

```
istio-proxy:
    Container ID:  docker://b4c8136766fc6818b979b49638786418b1a1676eeb4eda50a24cee37b7fb753f
    Image:         docker.io/istio/proxyv2:1.0.2
    Image ID:      docker-
pullable://istio/proxyv2@sha256:54e206530ba6ca9b3820254454e01b7592e9f986d27a5640b6c03704b3b68332
    Port:          <none>
    Host Port:     <none>
    Args:
      proxy
      sidecar
```

Lines omitted for brevity

```
Environment:
      POD_NAME:                      httpbin-77647f7b59-nrtgb (v1:metadata.name)
      POD_NAMESPACE:                 default (v1:metadata.namespace)
      INSTANCE_IP:                    (v1:status.podIP)
      ISTIO_META_POD_NAME:           httpbin-77647f7b59-nrtgb (v1:metadata.name)
      ISTIO_META_INTERCEPTION_MODE:  REDIRECT
    Mounts:
      /etc/certs/ from istio-certs (ro)
      /etc/istio/proxy from istio-envoy (rw)
```

Istio - Ingress

Ingress Controller

Native Kubernetes Support

Independent Component

Simple Install/Config

Well Documented

Combined L4-6 and L7

Ingress Gateway

CRD Add-On

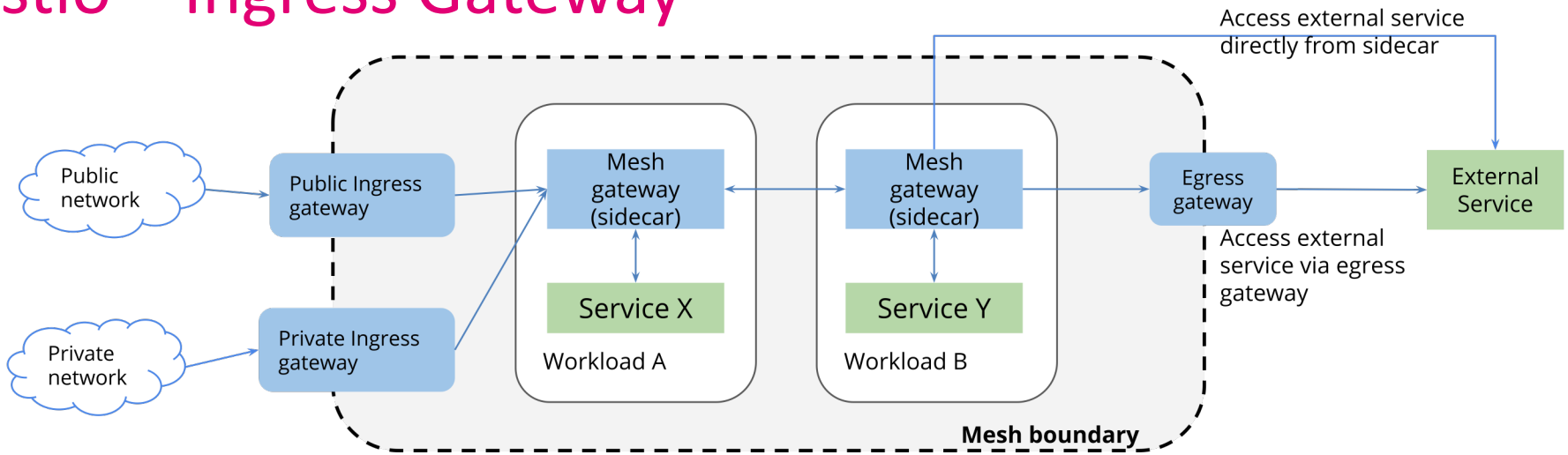
Only works within the Istio stack

Still New/Lacking Documentation

Separation of L4-6 and L7

Supports Complex Routing Rules

Istio – Ingress Gateway



Gateway | **VirtualService** | **DestinationRule** | **ServiceEntry**

Istio – Demo 1

- httpbin
  - Regular with ingress gateway
  - Edge Service rule (timeout)

Istio - Demo 2

- bookinfo
  - Regular – route to any version
  - Controlled - Route all to one version
  - Context – Routed to version based on identity
  - Traffic Splitting – 50/50 between 2 versions

Q&A