Google Cloud

Introduction to Istio Ambient Mesh

Abdel SGHIOUAR
Twitter: **@boredabdel**

Kubernetes Community Days
MUNICH 2023

SPONSORS

metalstack cloud
ELASTICBRAINS
APE FACTORY
QA|WARE
Google Cloud
Grafana Labs
AppsCode
mkdev
ISOVALENT
camp to camp
INNOVATIVE SOLUTIONS BY OPEN SOURCE EXPERTS
StormForge
okteto
lumigo
EXOSCALE
Red Hat
SysEleven
REPLY LIQUID
white duck
The Azure Cloud Experts
blueshoe

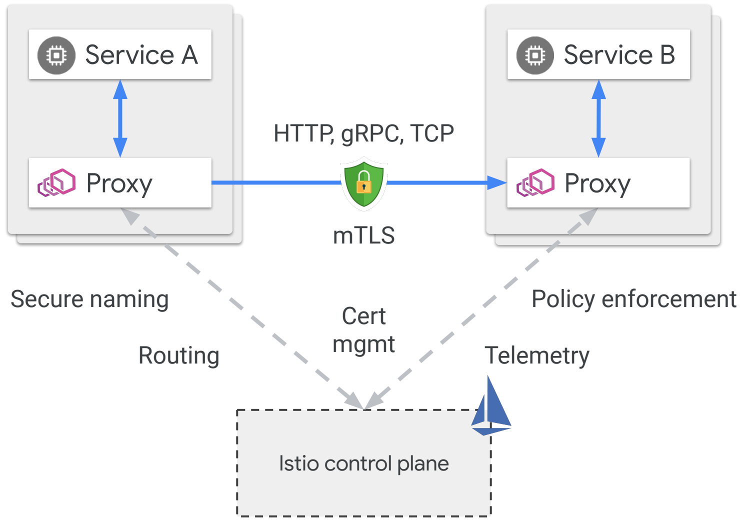
Speaker: Abdel SGHIOUAR(@boredabdel)
Company: Google Cloud

Intro to Istio Ambient Mesh

Google Cloud

Abdel SGHIOUAR
Senior Cloud Developer Advocate
Kubernetes Podcast co-host
CNCF Ambassador
Twitter: **boredabdel@**

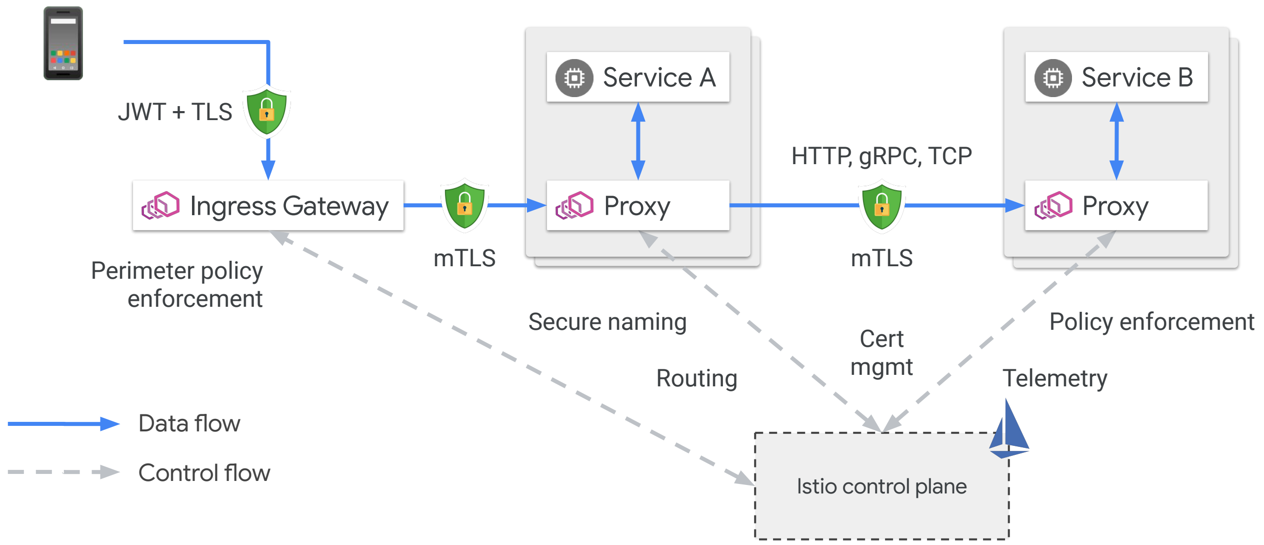
Istio current Architecture



→ Data flow

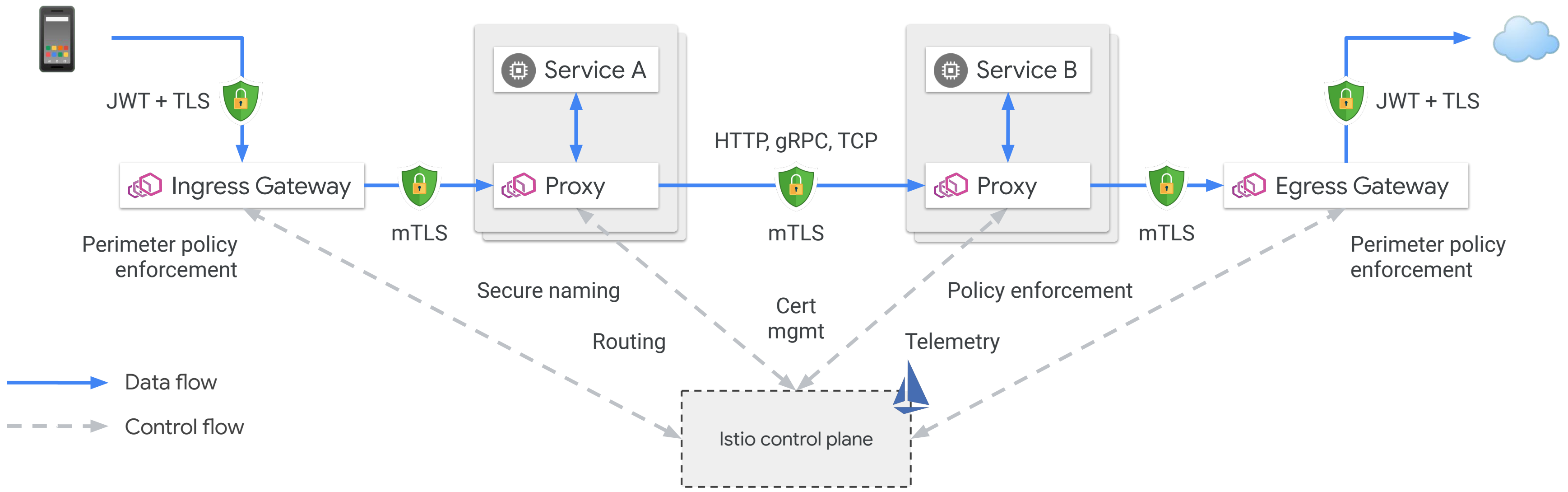
⇢ Control flow

Gateway for ingress into the mesh



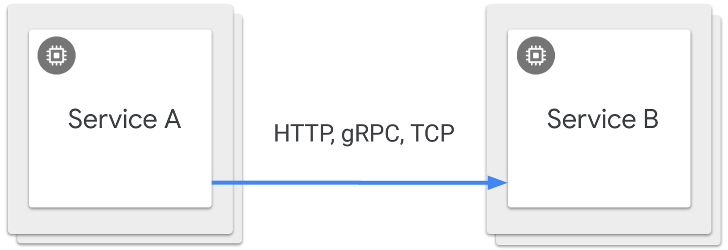
Google Cloud

...and for egress out from the mesh



Google Cloud

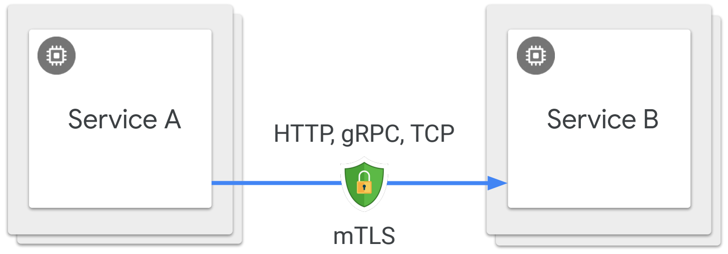
Let's take a step back



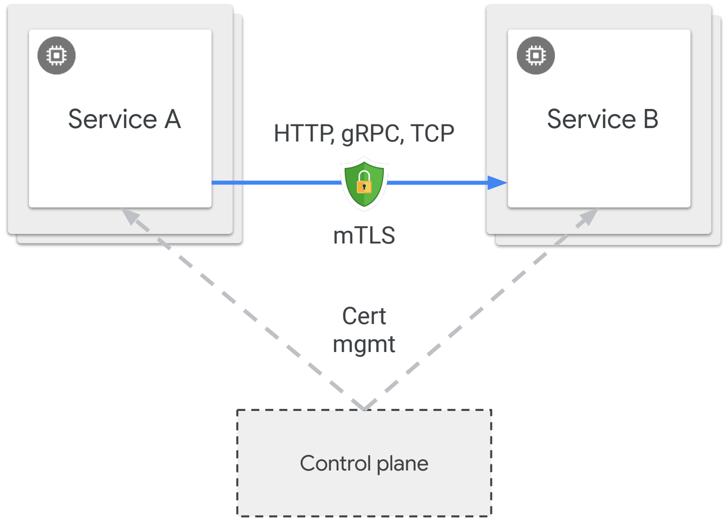
Data flow

Google Cloud

Let's add mTLS



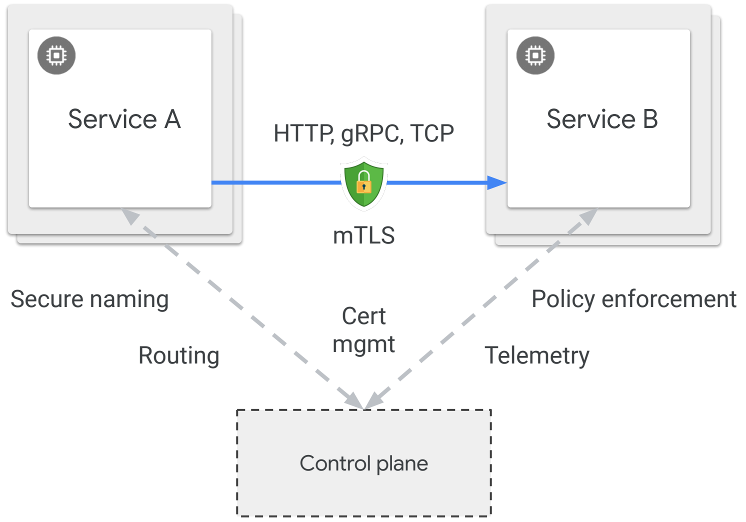
Manage the certificates, somehow



Data flow
Control flow

Google Cloud

We also want other application layer smarts



Data flow
Control flow

Google Cloud

All sorts of policies

Quality of Service

- Timeouts
- Retries
- Circuit breakers
- Traffic allocation

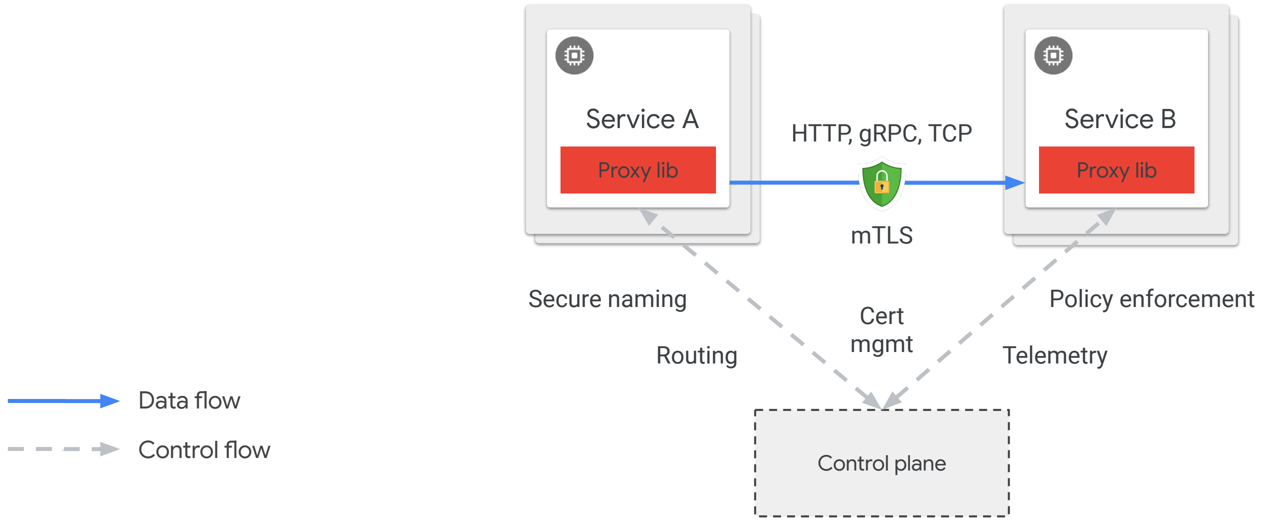
Authorization

- Local authorization
- 3rd party lookups
- Quotas and rate limiting

Traffic shaping

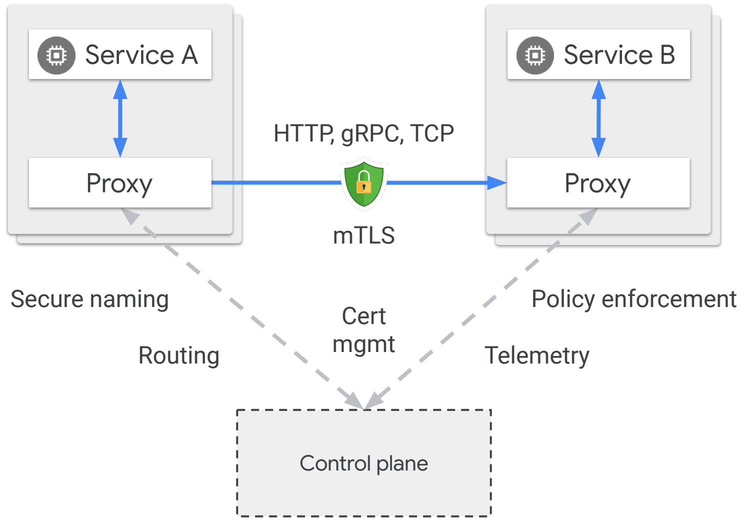
- Content-based routing
- Canaries
- A/B testing

All this logic needs to be enforced at the “edge”



Google Cloud

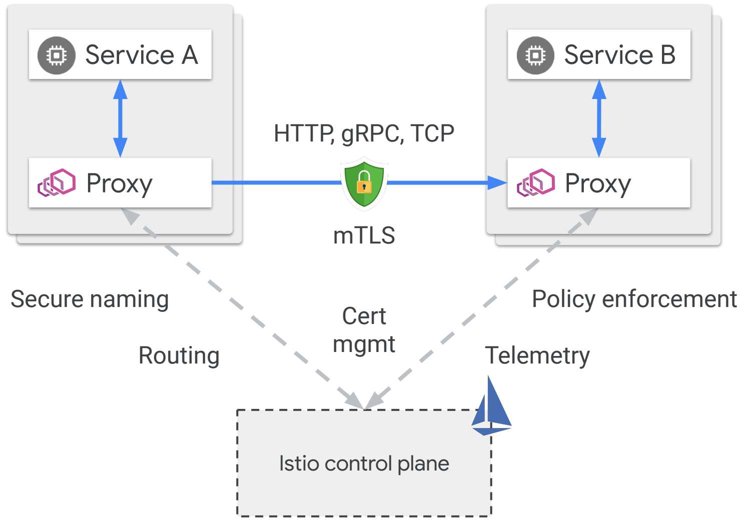
In the real world, we use “sidecar” proxies



Data flow
Control flow

Google Cloud

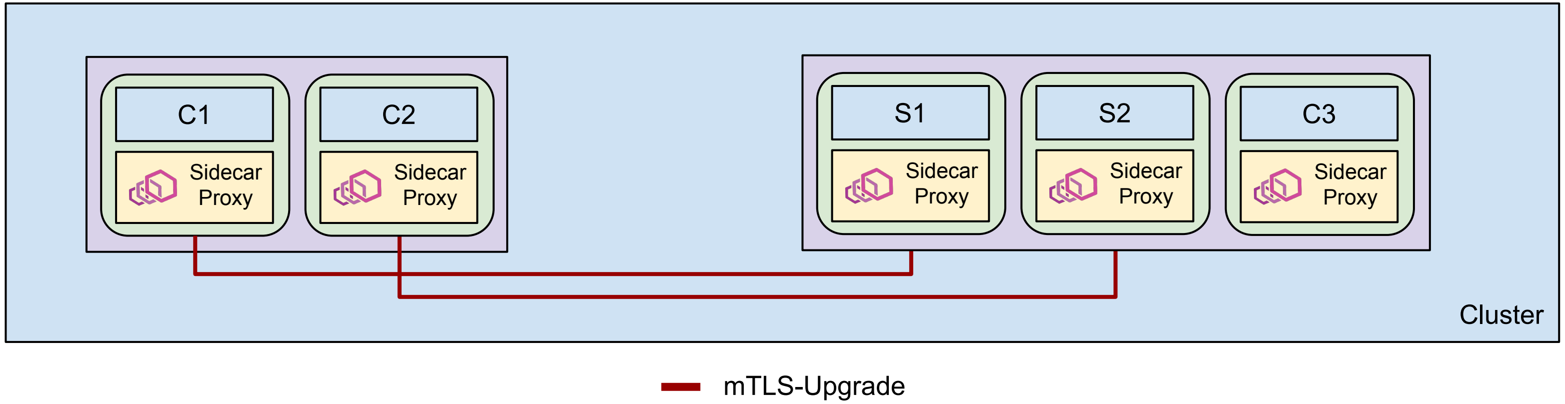
This is how we came here



Data flow
Control flow

Google Cloud

Istio’s Current Data Plane



Sidecars

- Improvement over previous alternatives to get benefits from mesh
- While still useful and important, sidecars have some complications:
  - Invasive
    - Requires modifying the workload–can’t be hot-inserted
    - Difficult install/uninstall/upgrade, requires restarts
  - Breaks some applications with broken HTTP implementations
  - Over-provisions resources for sidecars

Ambient Mesh Datapath Goals

- Non-disruptive to applications
  - Hot-insertion without modifying workload
  - Low risk of breaking traffic
  - Transparent, zero-downtime, upgrades
- Compatibility with sidecar-based Istio
  - Traffic interoperable with pods using the traditional sidecar
  - Smooth upgrade path from mTLS-only to full Istio
- Simple check-box enablement/disablement

Architecture

- Removes sidecar and splits proxy into two parts
- Treat mesh as two layers: Secure Overlay and L7 Processing
- Secure overlay implemented by a per-node shared ztunnel
  - ztunnel as a DaemonSet
  - Authentication and encryption to other ztunnels or waypoint proxies
  - L4 policies and telemetry
- Full L7 Istio implemented by a full L7 waypoint proxy
  - L7 policies and telemetry
- HBone provides authentication and encryption without breaking applications

Ambient Mesh Layers

L7 Processing Layer

All features of the Secure Overlay plus…

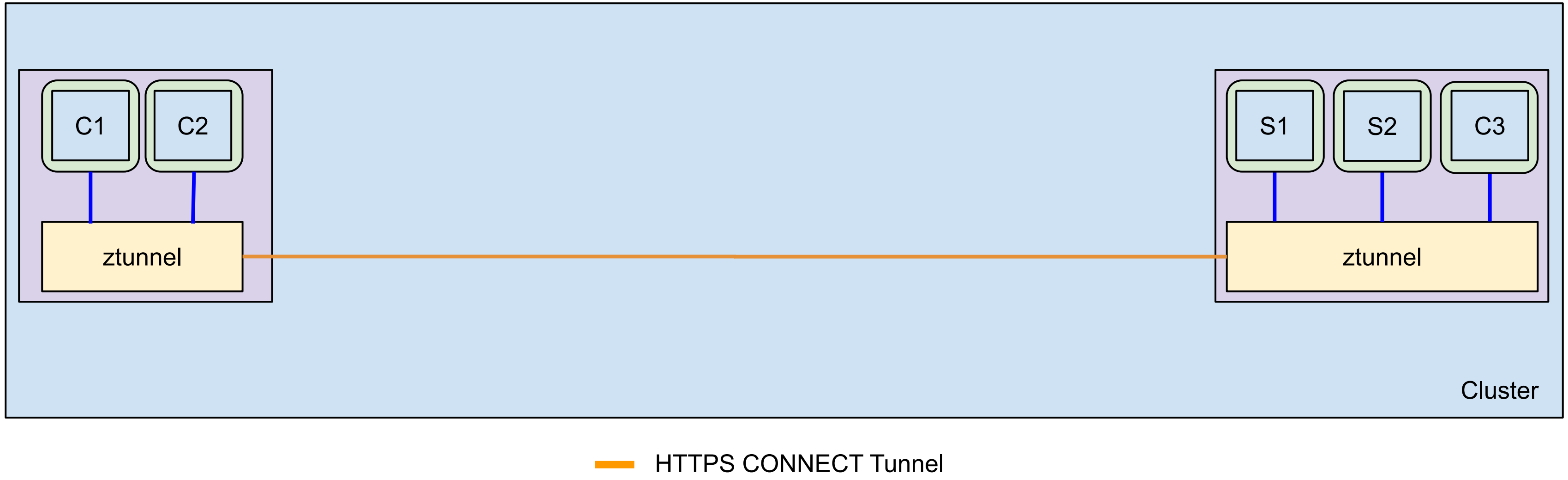
- **Traffic Mgmt:** HTTP routing & load balancing, Circuit breaking, Rate limiting, Fault injection, Retry, Timeouts, …
- **Security:** Rich authorization policies
- **Observability:** HTTP metrics, Access Logging, Tracing

Secure Overlay Layer

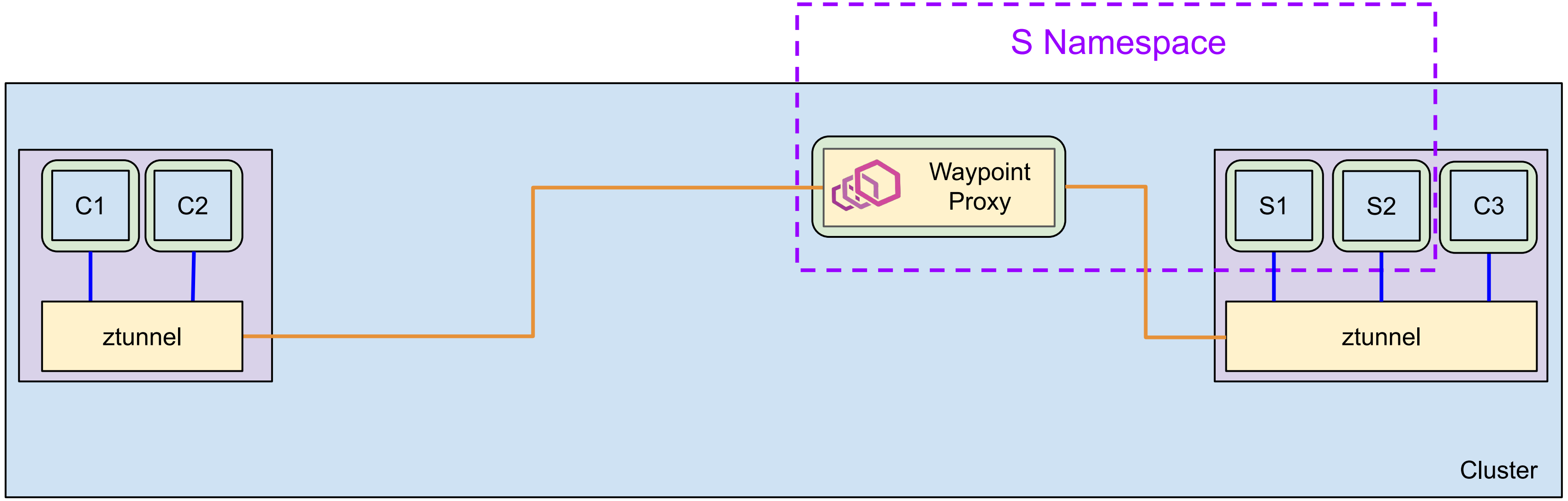
Streamlined, low resource, high performance with zero trust

- **Traffic Mgmt:** TCP Routing
- **Security:** mTLS tunneling, Simple authorization policies
- **Observability:** TCP metrics & logging

Secure Overlay



L7 Policies

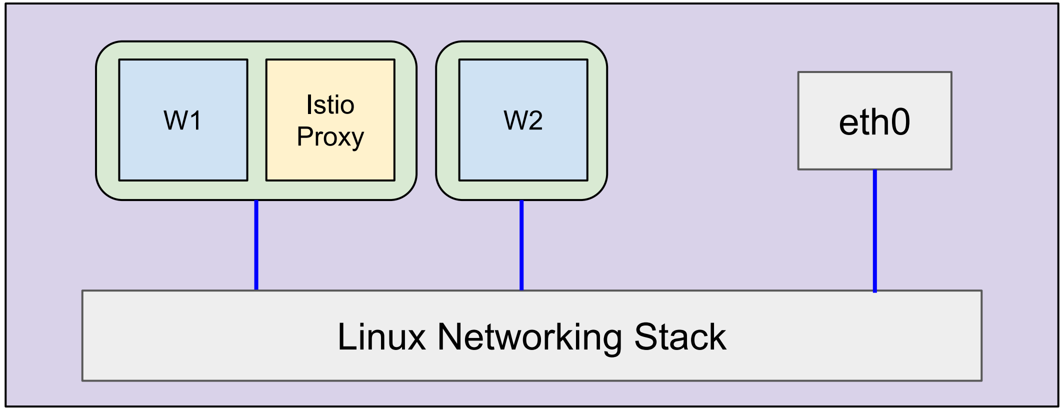


🟧 HTTPS CONNECT Tunnel

Deploying Ambient Mesh

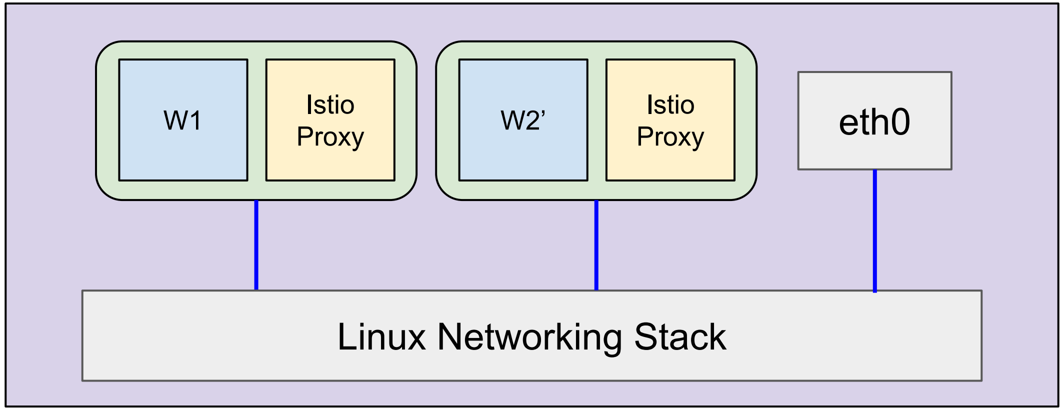
Traditional Istio Deployment

- Proxy loaded as sidecar with shared networking in pod
- iptables redirects the workload’s traffic in and out of the sidecar proxy
- Node networking stack unmodified



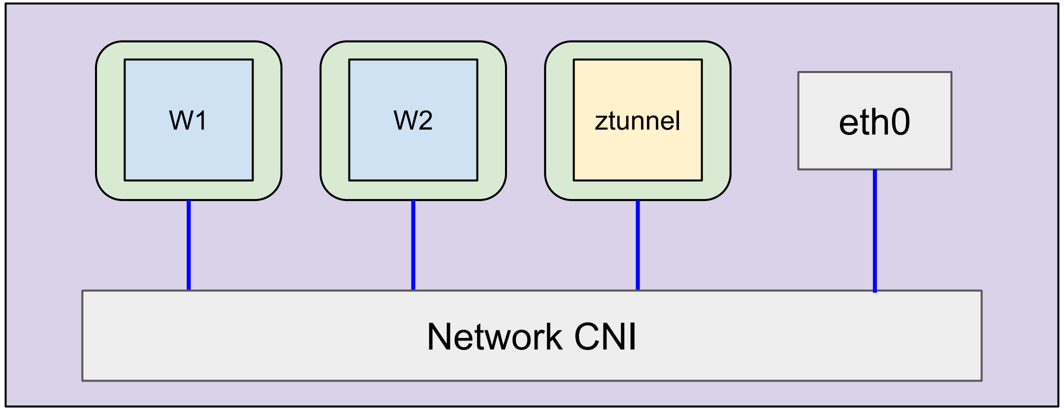
Traditional Istio Deployment

- Proxy loaded as sidecar with shared networking in pod
- iptables redirects the workload’s traffic in and out of the sidecar proxy
- Node networking stack unmodified
- Sidecar insertion makes modifications to workload pod that requires restart



Ambient Mesh Deployment

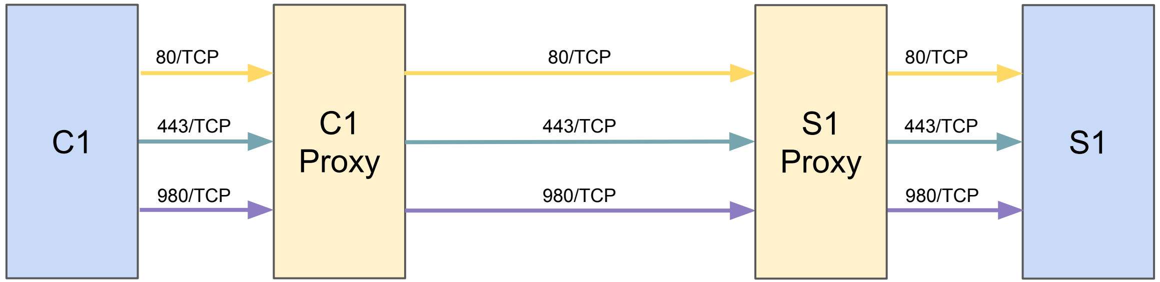
- CNI redirects traffic from the workload to the ztunnel to provide non-bypassability
- Allows hot-enablement of Istio through dynamic redirect



HBONE

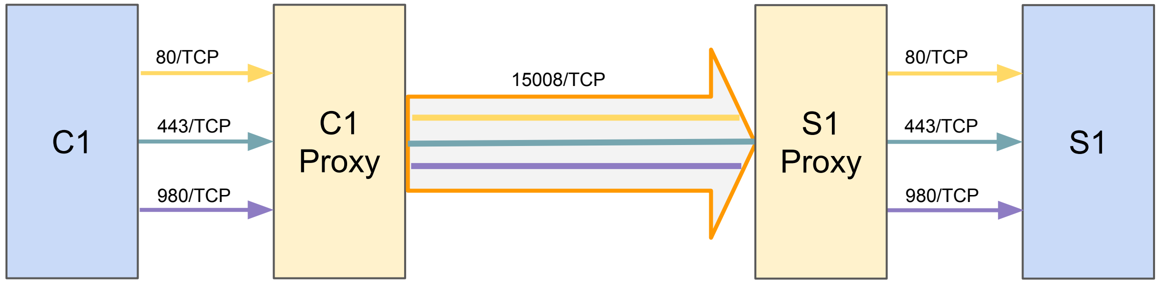
Traditional Istio Proxy Traffic

- Each connection from the client creates a new TCP connection between the proxies
- mTLS-tunneled traffic uses the same port numbers as the original
  - Sniffing code in Envoy determines whether traffic is encrypted or not
  - Breaks server-speaks-first protocols (e.g., MySQL) when using Permissive mTLS



HBone

- All traffic tunneled through a single mTLS connection using HTTP Connect
  - Fixes server-speaks-first protocols for Permissive mTLS
  - Amortizes cost of mTLS handshakes over multiple connections
  - Doesn’t require sniffing or metadata exchange hacks
  - Simplifies network policies, since Istio will use a single port
- Decouple mTLS encryption from the application

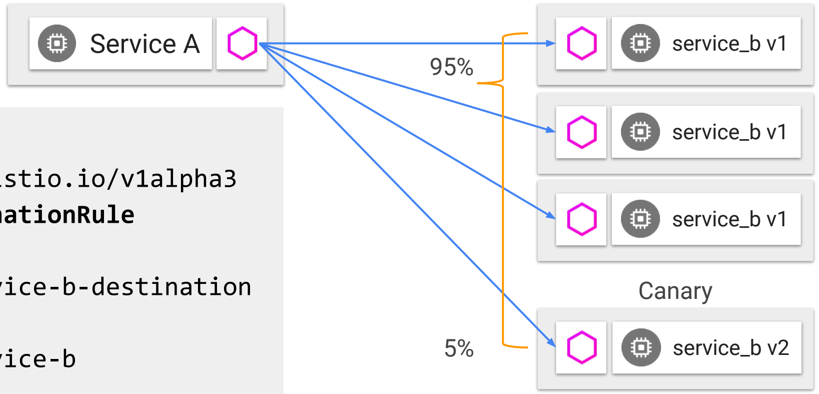


Demo

Istio Traffic Management

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name:service-b
spec:
  hosts:
  - service-b
  Http:
  - route:
    - destination:
        host: service-b
        subset: v2
      weight: 5
    - destination:
        host: service-b
        subset: v1
      weight: 95
```

```
apiVersion:
networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: service-b-destination
spec:
  host: service-b
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```



Gateway API OSS

A modern set of APIs for **L4 and L7 Load-Balancing and Mesh** in Kubernetes.

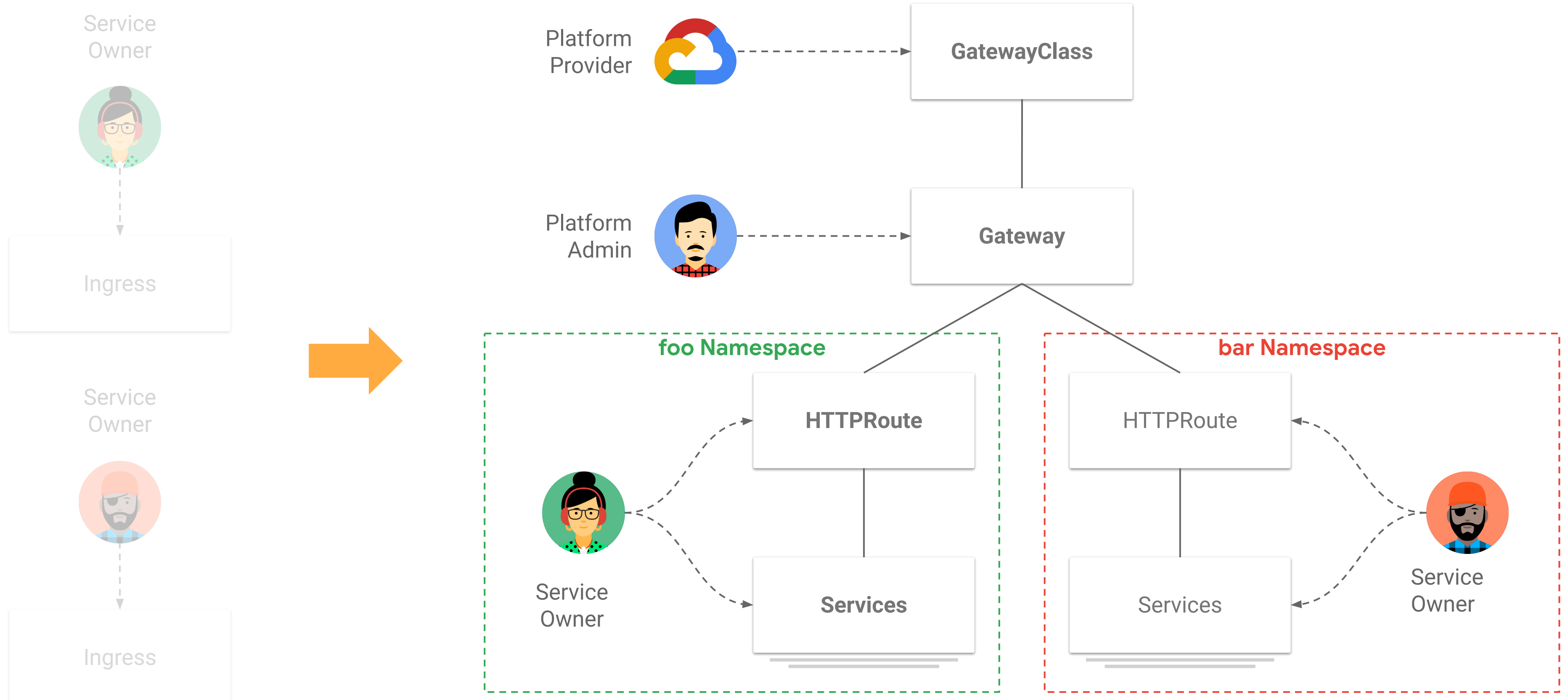
Evolution from Ingress and Istio, the Gateway API is designed to standardize how service networking is expressed.

8+ implementations (Google, Istio, +external vendors)

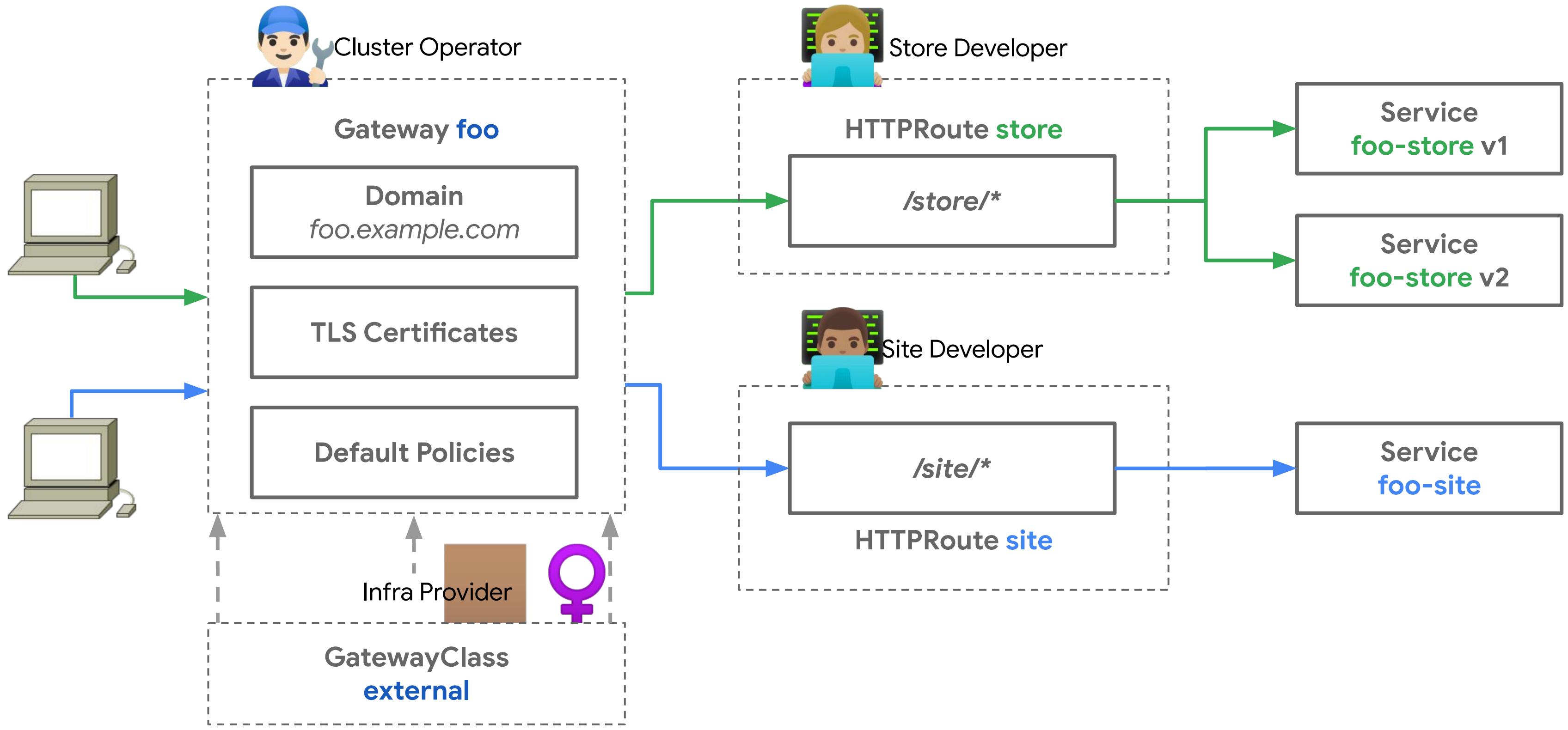
Istio Google Cloud Kong træfiklabs HAPROXY solo.io

NGINX vmware Red Hat HashiCorp AMBASSADOR LABS

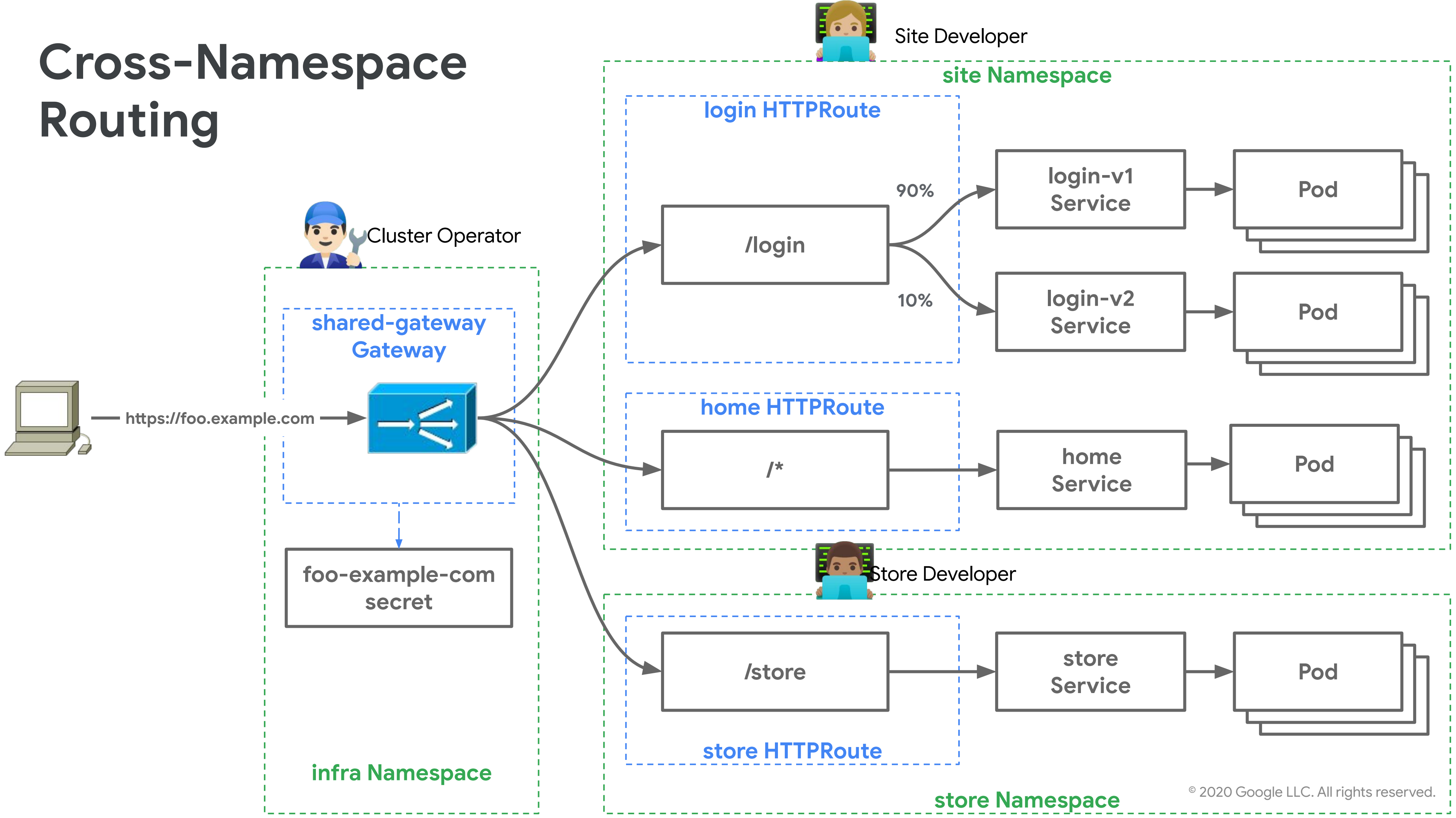
What is the Gateway API?



Role Oriented Resource Model

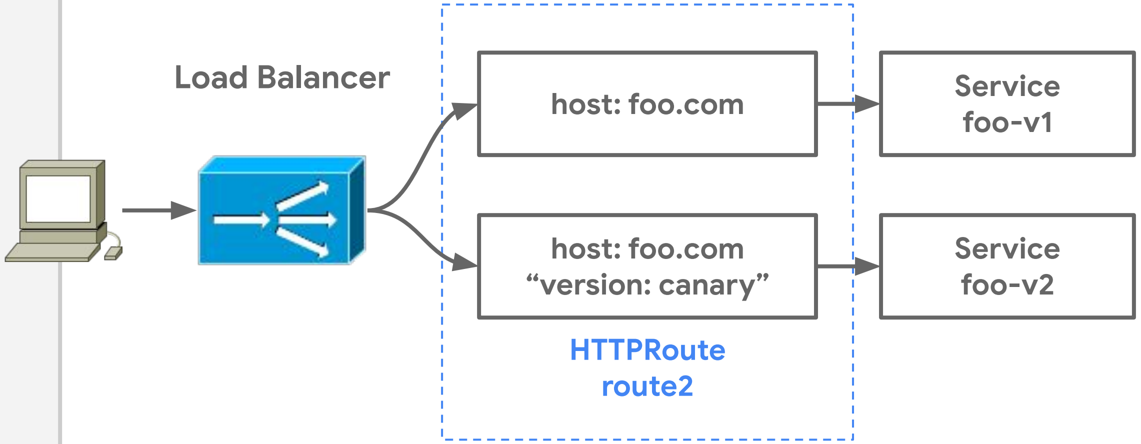


Cross-Namespace Routing



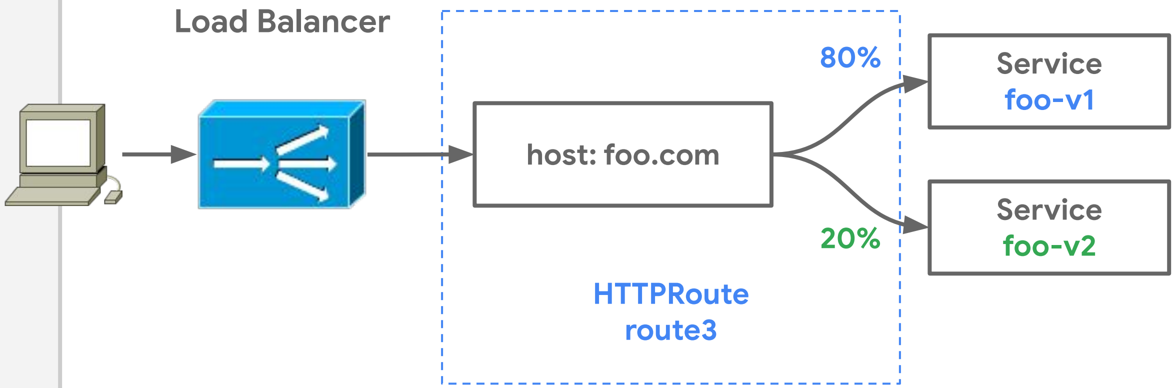
More expressive routing

```
kind: HTTPRoute
apiVersion: networking.x-k8s.io/v1alpha1
metadata:
  name: foo-route
  namespace: foo
  labels:
    gateway: internal-gw
spec:
  hostnames:
  - "foo.com"
  rules:
  - matches:
    - headers:
        values:
          version: canary
    forwardTo:
    - serviceName: foo-v2
      port: 8080
  - forwardTo:
    - serviceName: foo-v1
      port: 8080
```

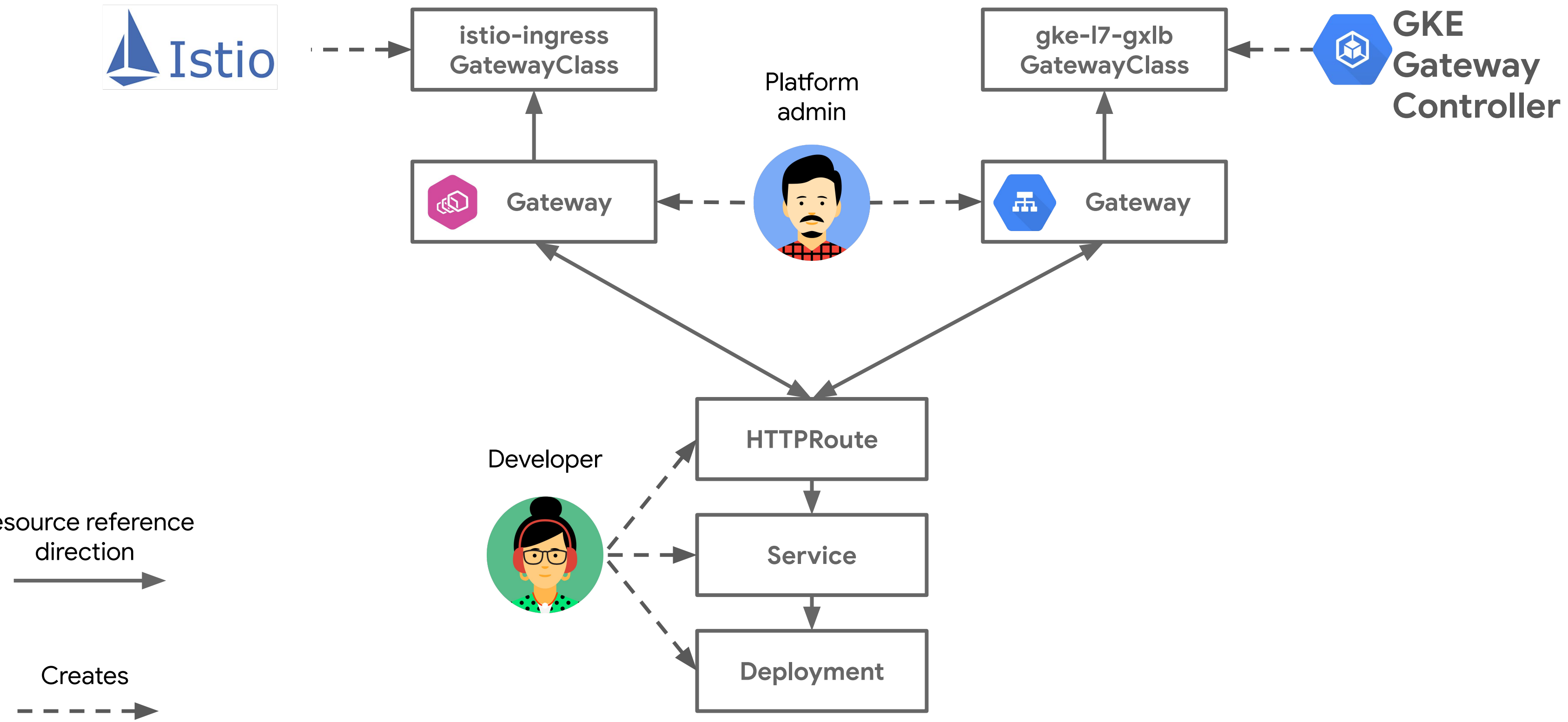


More expressive routing

```
kind: HTTPRoute
apiVersion: networking.x-k8s.io/v1alpha1
metadata:
  name: foo-route
  namespace: foo
  labels:
    gateway: internal-gw
spec:
  hostnames:
  - "foo.com"
  rules:
  - forwardTo:
    - serviceName: foo-v1
      port: 8080
      weight: 80
    - serviceName: foo-v2
      port: 8080
      weight: 20
```



Gateway <-> Route Relationships



Demo

Ambient Mesh Demo

- Basic Application with no Istio
- **Easily Install ambient mesh - secure overlay**
  - Customer enables ambient mesh to get mTLS
  - L4 authorization policies
  - Zero downtime, zero pod restarts
- **Easily Install Ambient Mesh - L7 policies**
  - Istio waypoint proxies are deployed and utilized
  - L7 Policies
  - Zero downtime, zero pod restarts
- **Easily uninstall Istio**
  - Zero downtime, zero pod restarts

Takeaways

- We expect ambient mesh to be the best fit for most users going forward
- Sidecars still have their place and will continue to be supported
  - Applications that require dedicated resources
  - Sites that need customization (e.g., EnvoyFilter)
  - Regulated environments that expect their deployment model
  - Users that just like sidecars and don’t want to change
- Ambient and sidecars can be deployed together and interoperate
- “Experimental” code and announcement today
- Plan to release in the coming months
- Please contribute!

Thank You!

Google Cloud

Abdel SGHIOUAR
Senior Cloud Developer Advocate
Kubernetes Podcast co-host
CNCF Ambassador
Twitter: **boredabdel@**