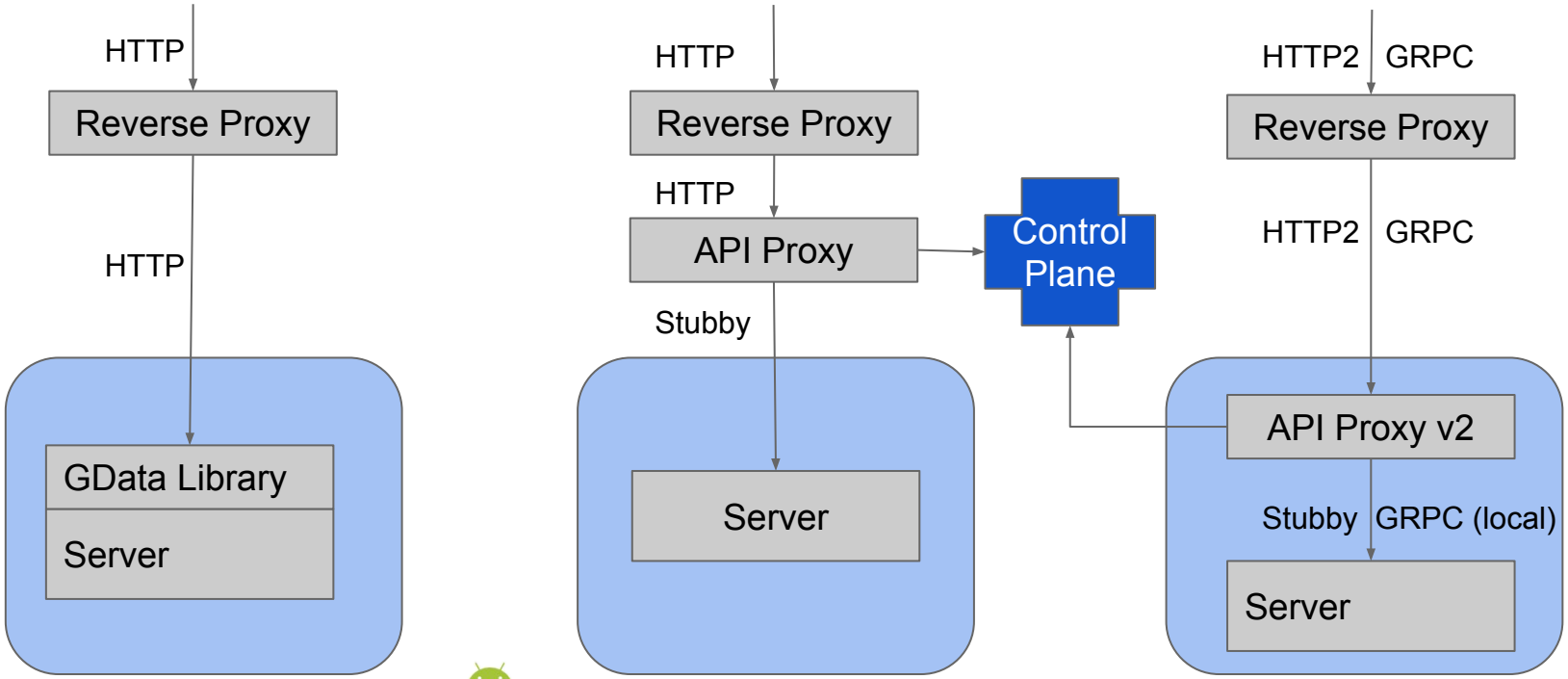
Istio

A modern service mesh

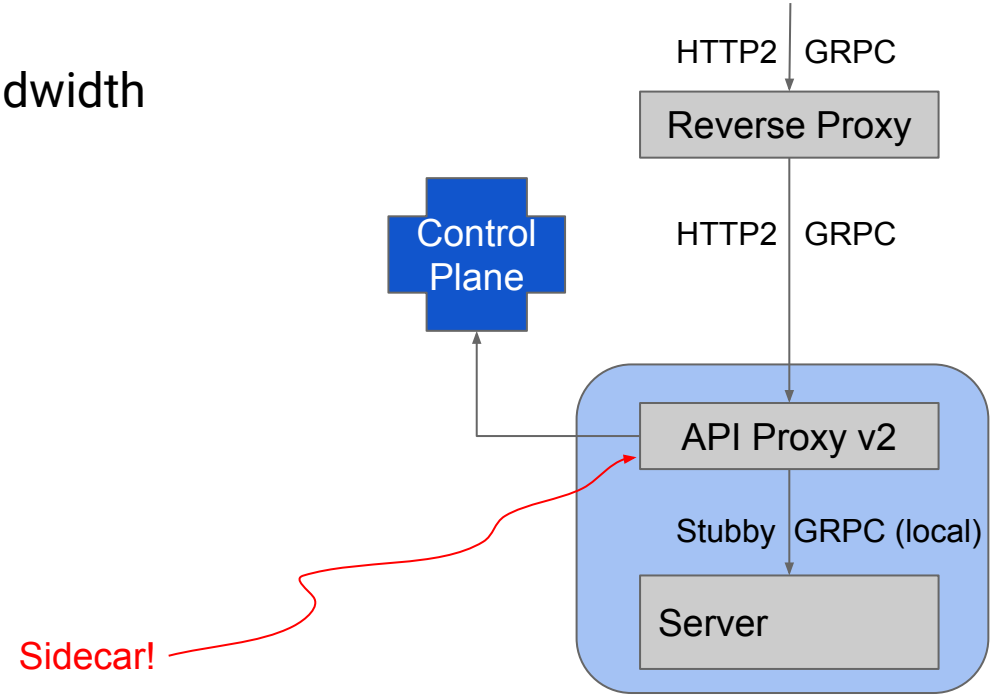
Louis Ryan
Principal Engineer @ Google
@louiscryan

My Google Career



Cloud → Internal & External Convergence

- Network distance & bandwidth
- Protocols
- Isolation & Reliability
- Security Concerns



Decoupling → Velocity

- Operators & Developers
- Code & Networking
- Network Topology & Security
- Modernization & Architecture

What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency
- Traffic Control
- Security
- Policy Enforcement
- Zero code change

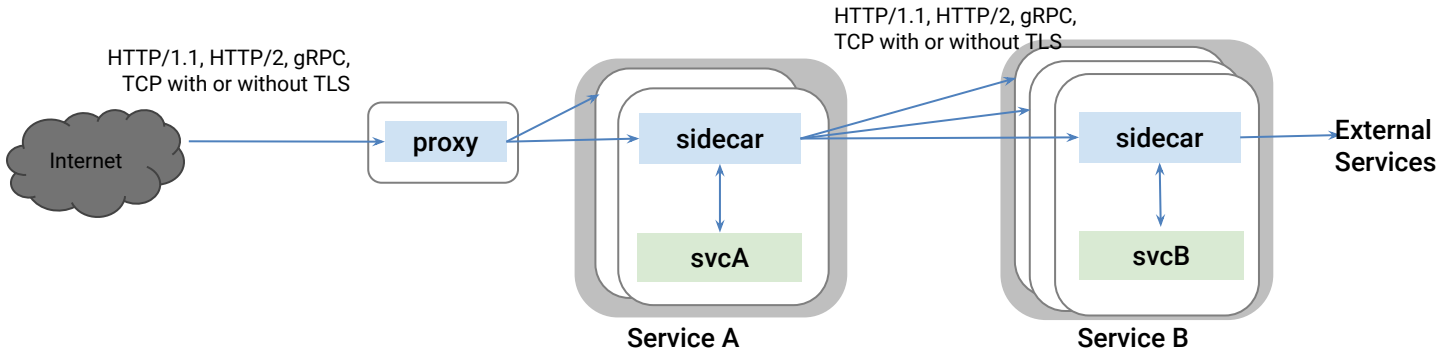
FREE!

What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency
- Traffic Control
- Security
- Policy Enforcement

Weaving the mesh - Sidecars



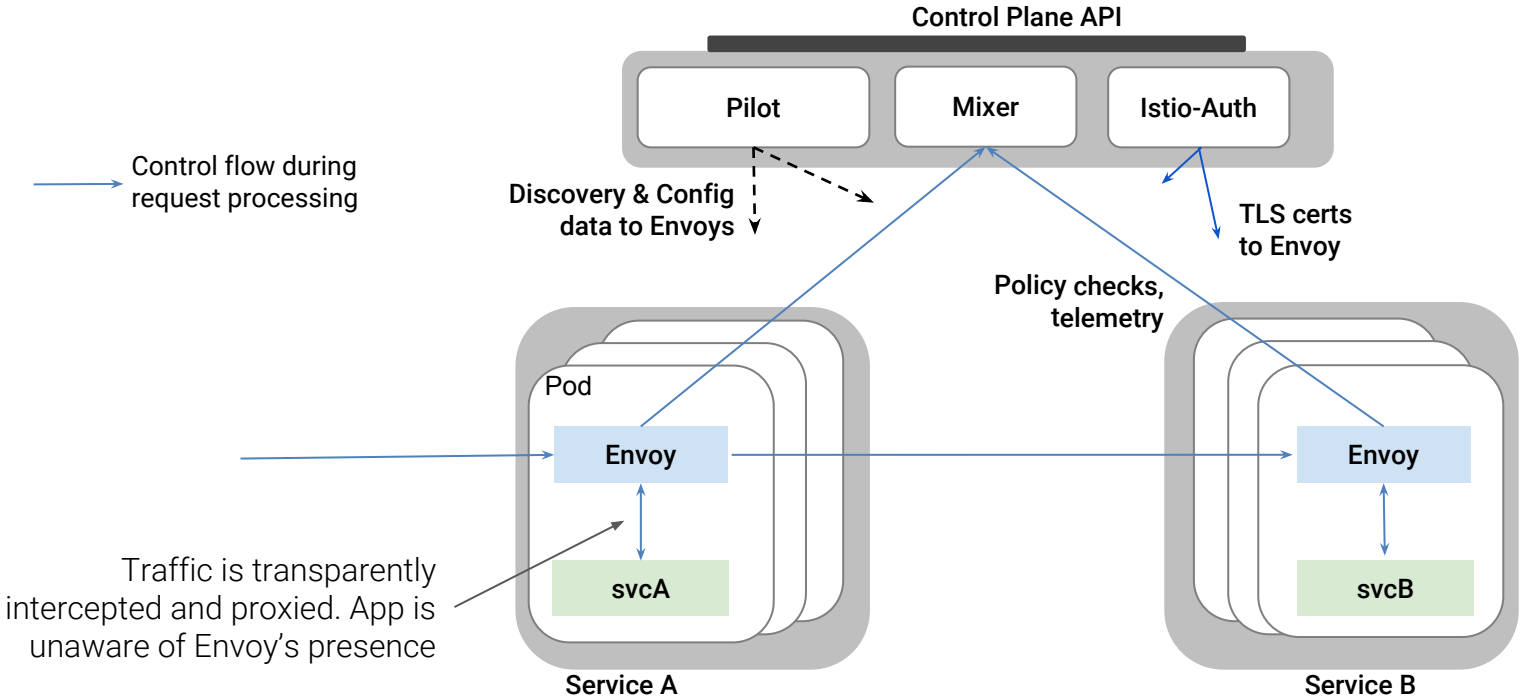
Outbound features:

- Service authentication
- Load balancing
- Retry and circuit breaker
- Fine-grained routing
- Telemetry
- Request Tracing
- Fault Injection

Inbound features:

- Service authentication
- Authorization
- Rate limits
- Load shedding
- Telemetry
- Request Tracing
- Fault Injection

Istio - Putting it all together



Our sidecar of choice - Envoy

- A C++ based L4/L7 proxy
- Low memory footprint
- Battle-tested @ Lyft
  - 100+ services
  - 10,000+ VMs
  - 2M req/s

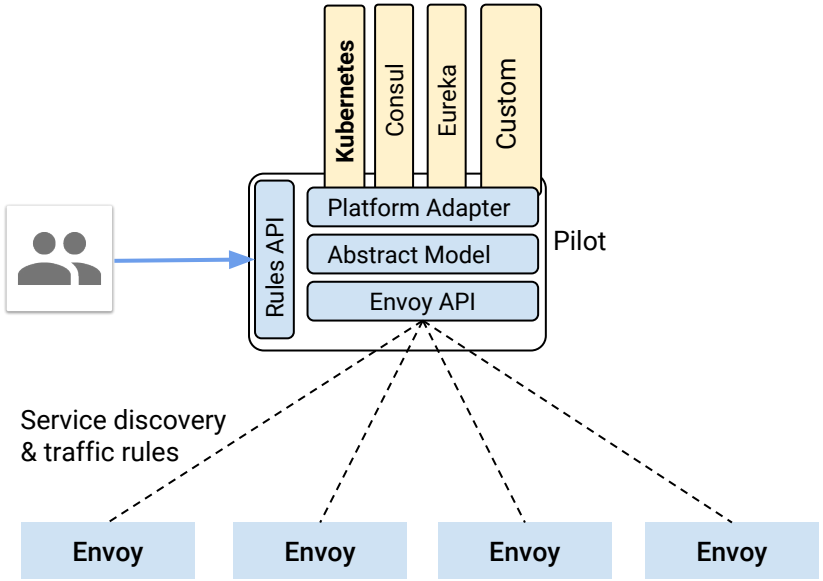
Plus an awesome team willing to work with the community!

Goodies:

- API driven config updates → no reloads
- Zone-aware load balancing w/ failover
- Traffic routing and splitting
- Health checks, circuit breakers, timeouts, retry budgets, fault injection, …
- HTTP/2 & gRPC
- Transparent proxying
- Designed for observability

Modeling the Service Mesh

1. Environment-specific topology extraction
2. Topology is mapped to a platform-agnostic model.
3. Additional rules are layered onto the model. E.g. retries, traffic splits etc.
4. Configuration is pushed to Envoy and applied without restarts



What is a ‘Service Mesh’ ?

A network for services, not bytes

- **Observability**
- Resiliency
- Traffic Control
- Security
- Policy Enforcement

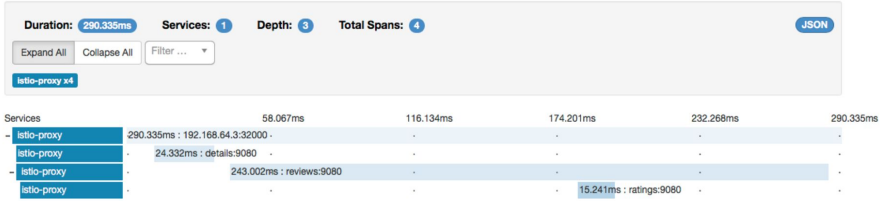
Visibility

Monitoring & tracing should not be an afterthought in the infrastructure

Goals

- Metrics without instrumenting apps
- Consistent metrics across fleet
- Trace flow of requests across services
- Portable across metric backend providers

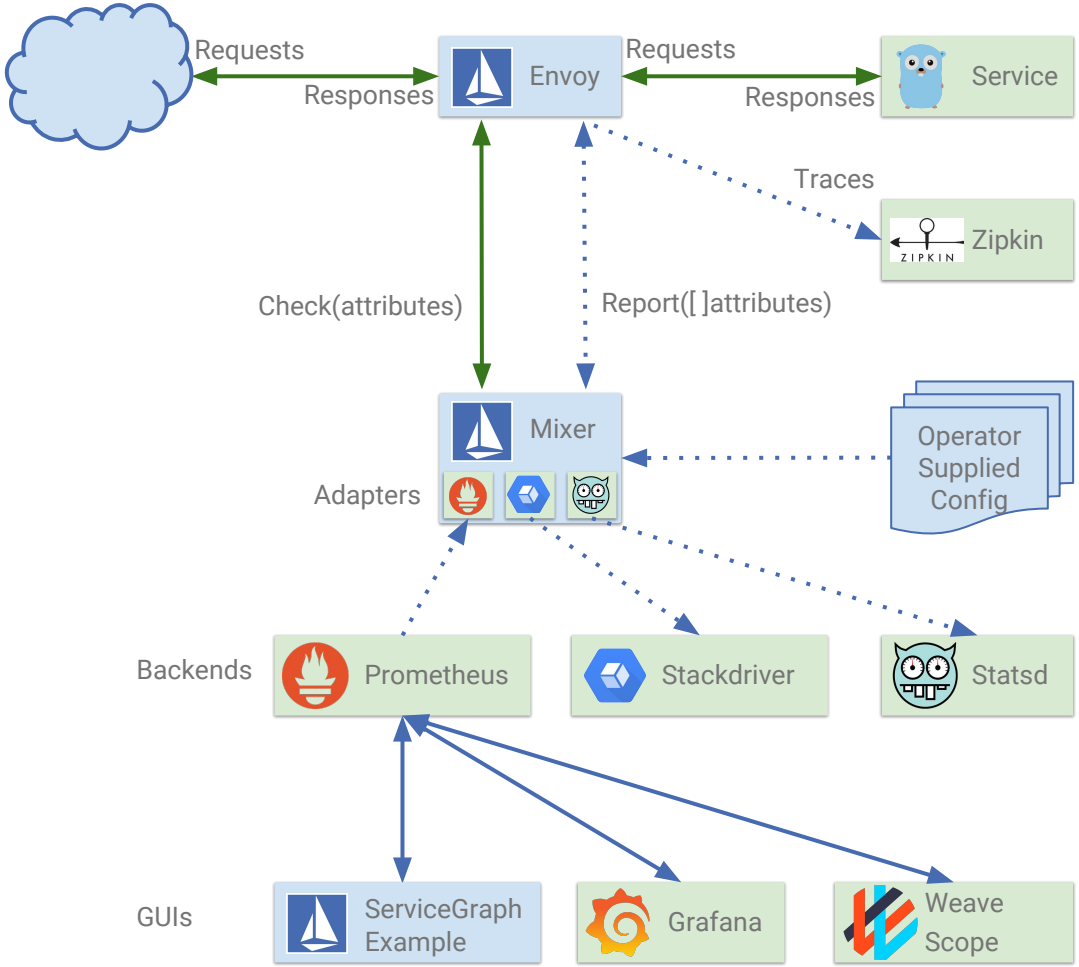
Istio - Grafana dashboard w/ Prometheus backend



Istio Zipkin tracing dashboard

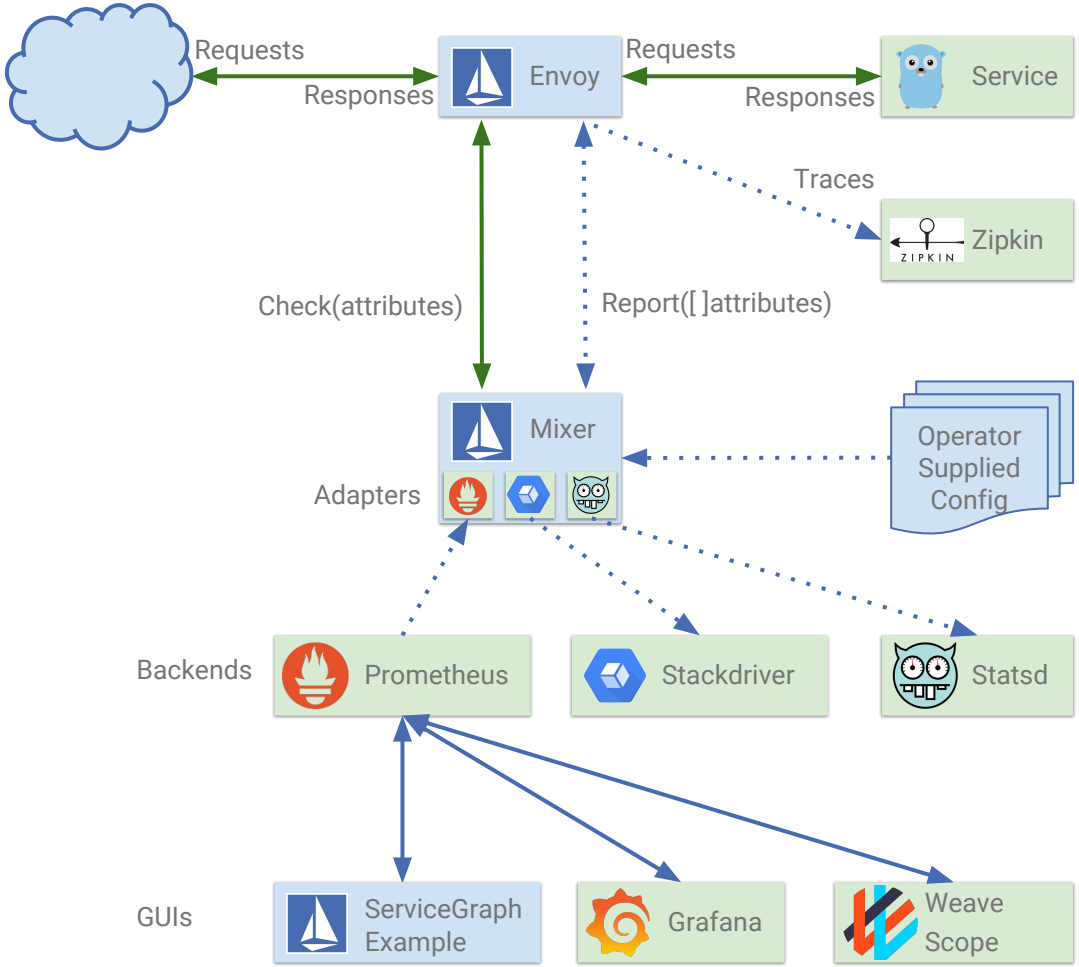
Visibility: Metrics

- Mixer collects metrics emitted by Envoys
- Adapters in the Mixer normalize and forward to monitoring backends
- Metrics backend can be swapped at runtime



Visibility: Tracing

- Applications do not have to deal with generating spans or correlating causality
- Envoys generate spans
  - Applications need to forward context headers on outbound calls
- Envoy sends traces to Mixer
- Adapters at Mixer send traces to respective backends



What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- **Resiliency**
- Traffic Control
- Security
- Control

Resiliency

Istio adds fault tolerance to your application without any changes to code

```
// Circuit breakers

destination: serviceB.example.cluster.local
policy:
- tags:
    version: v1
  circuitBreaker:
    simpleCb:
      maxConnections: 100
      httpMaxRequests: 1000
      httpMaxRequestsPerConnection: 10
      httpConsecutiveErrors: 7
      sleepWindow: 15m
      httpDetectionInterval: 5m
```

Resilience features

- Timeouts
- Retries with timeout budget
- Circuit breakers
- Health checks
- AZ-aware load balancing w/ automatic failover
- Control connection pool size and request load
- Systematic fault injection

What is a ‘Service Mesh’ ?

A network for services, not bytes

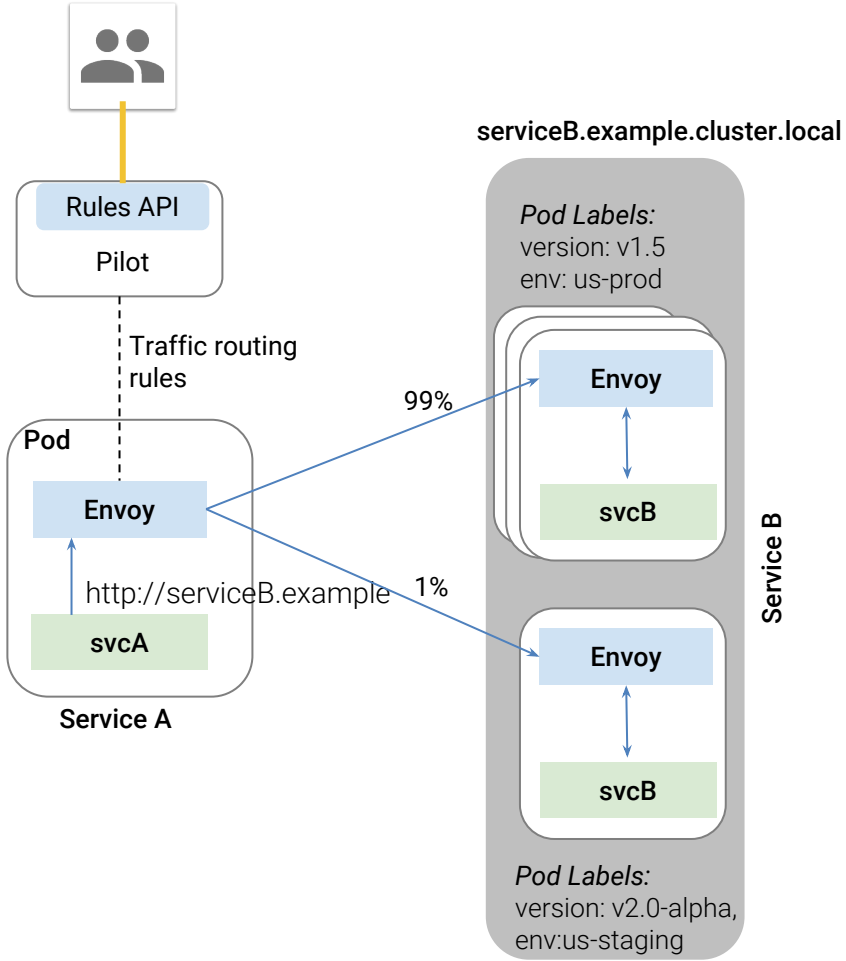
- Observability
- Resiliency & Efficiency
- **Traffic Control**
- Security
- Policy Enforcement

Traffic Splitting

```
// A simple traffic splitting rule

destination: serviceB.example.cluster.local
match:
  source: serviceA.example.cluster.local
route:
- tags:
    version: v1.5
    env: us-prod
  weight: 99
- tags:
    version: v2.0-alpha
    env: us-staging
  weight: 1
```

Traffic control is decoupled from infrastructure scaling

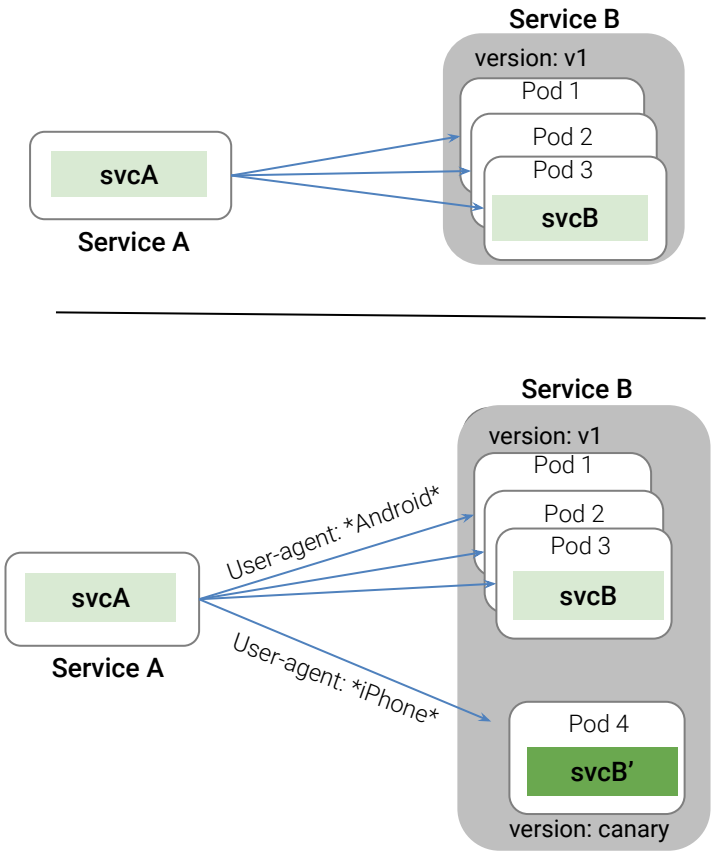


Traffic Steering

```
// Content-based traffic steering rule

destination: serviceB.example.cluster.local
match:
  httpHeaders:
    user-agent:
      regex: ^(.*?;)?(iPhone)(;.*)?$
precedence: 2
route:
- tags:
    version: canary
```

Content-based traffic steering



What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency & Efficiency
- Traffic Control
- **Security**
- Policy Enforcement

Securing Services

- Encryption by default
- Verifiable identity
- Secure naming / addressing
- Revocation

Problem: Strong Service Security at Scale

Concerns

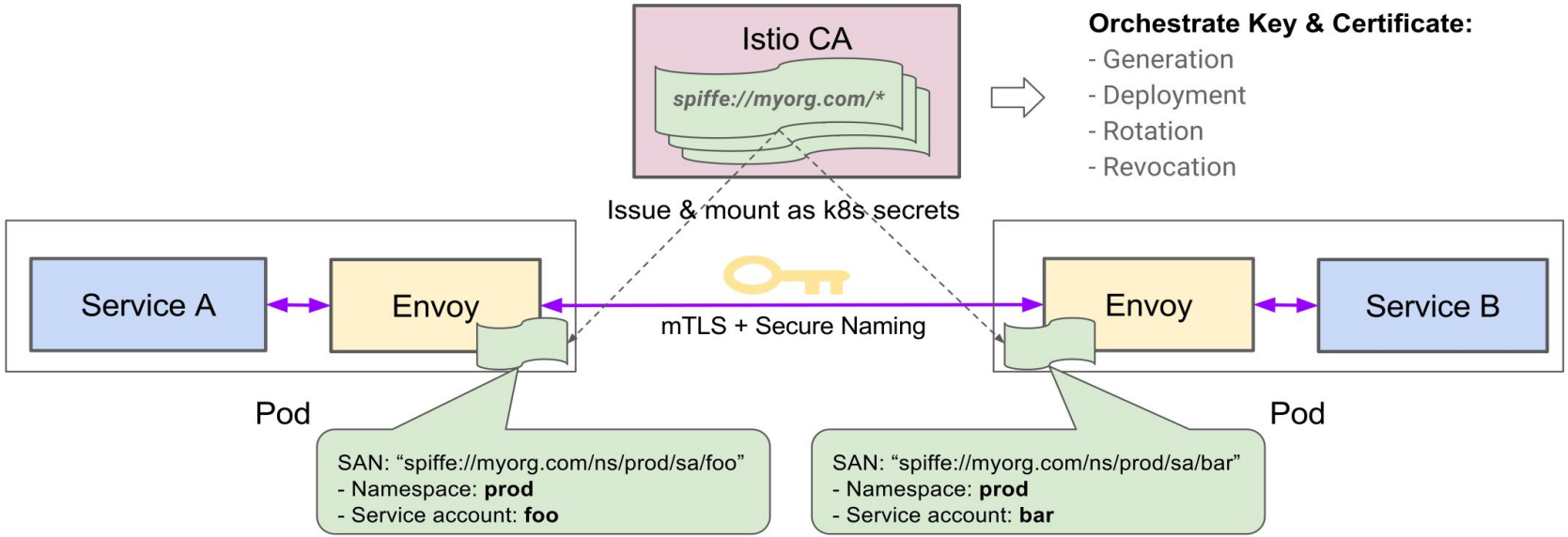
- Insiders
- Hijacked services
- Microservice attack surface
- Workload mobility
- Brittle fine-grained models
- Securing resources not just endpoints
- Audit & Compliance

Wants

- Workload mobility
- Remote admin & development
- Shared & 3rd party services
- User & Service identity
- Lower costs

Traditional perimeter security models are insufficient

Istio - Security at Scale



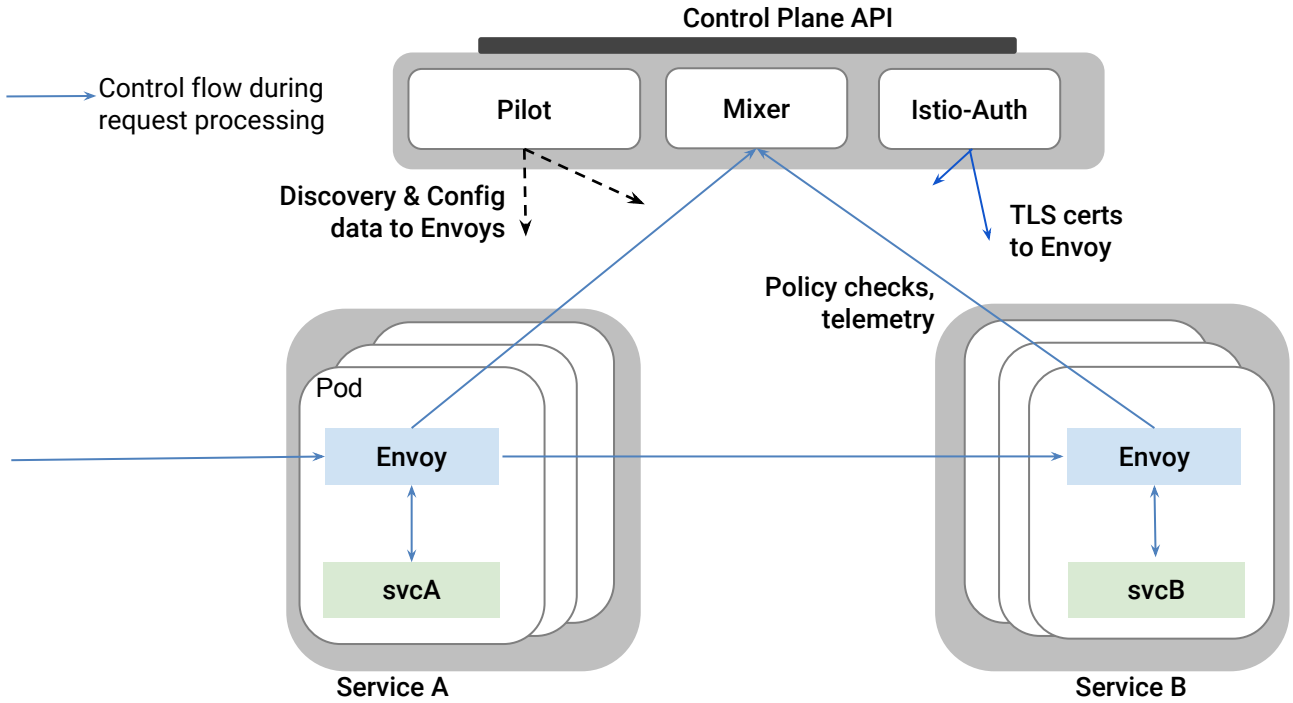
spiffe.io

What is a ‘Service Mesh’ ?

A network for services, not bytes

- Observability
- Resiliency & Efficiency
- Traffic Control
- Security
- **Policy Enforcement**

Putting it all together



What’s Mixer For?

- Nexus for policy evaluation and telemetry reporting
  - Precondition checking
  - Quotas & Rate Limiting
- Primary point of extensibility
- Enabler for platform mobility
- Operator-focused configuration model

Attributes - The behavioral vocabulary

```
target.service = “playlist.svc.cluster.local”
request.size   = 345
request.time   = 2017-04-12T12:34:56Z
source.ip      = 192.168.10.1
source.name    = “music-fe.serving.cluster.local”
source.user    = “admin@musicstore.cluster.local”
api.operation  = “GetPlaylist”
```

Roadmap

- Production Readiness
- Multi-Cloud & Multi-Environment
- Networking - Extension models, UDP, QUIC, performance, ...
- Moar integrations - ACLs, Telemetry, Audit, Policy, ...
- Security - HSM, Cert & Key stores, federation, ...
- API Management

Thanks! Phew