

Making Microservices Micro

with Istio



Google Cloud Platform

Ray Tsang

Developer Advocate

Google Cloud Platform

@saturnism | +RayTsang



Ray Tsang

Developer

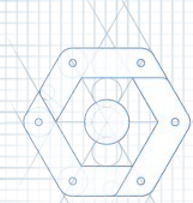
Architect

Traveler

Photographer

[flickr.com/saturnism](https://www.flickr.com/photos/saturnism/)

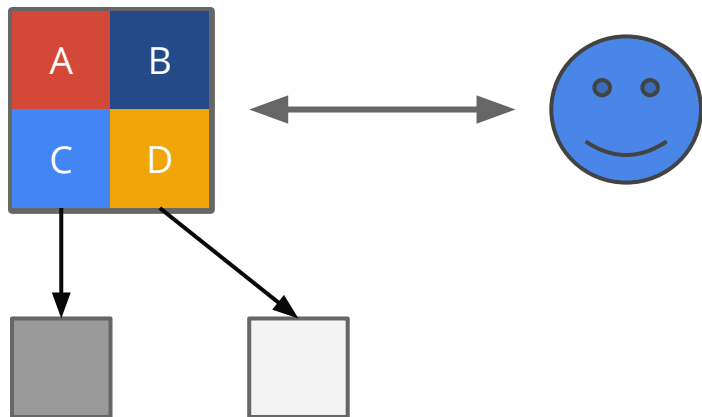


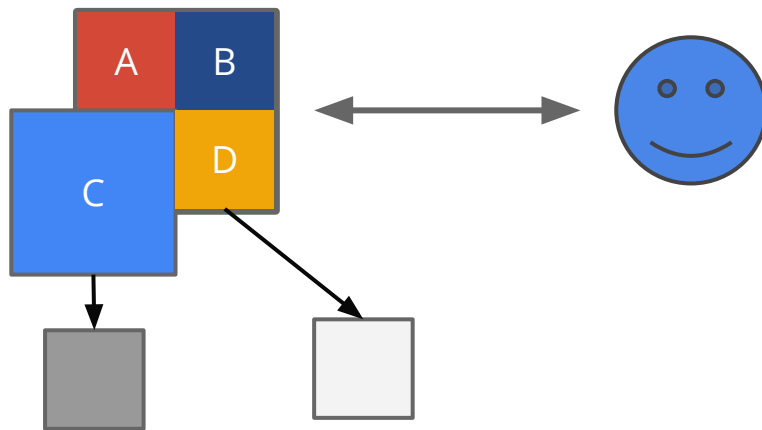


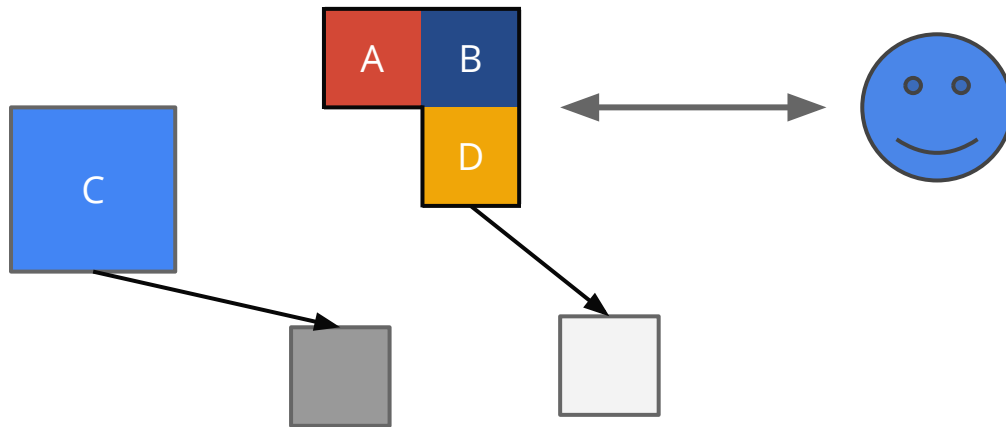
Microservices?

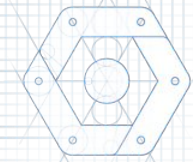
You probably heard a lot already!

No theories here - just a how to solve problems









So many instances...

Deployment

Resource Isolation & Utilization

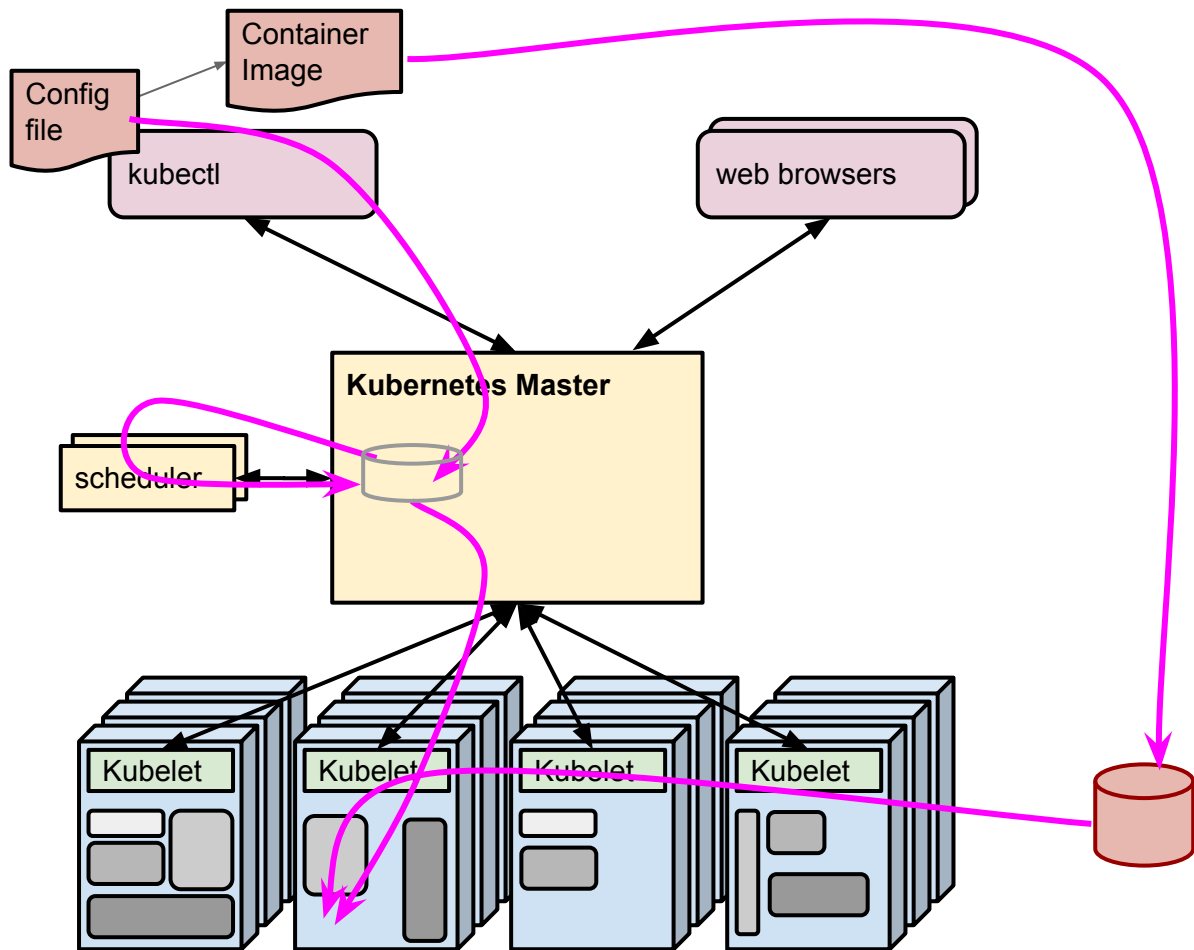
Resiliency

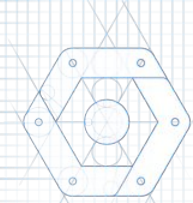
Networking



kubernetes

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: work-server-v1
  ...
spec:
  replicas: 2
  template:
    ...
    spec:
      containers:
        - name: work-server
          image: saturnism/work-server-istio:v1
```



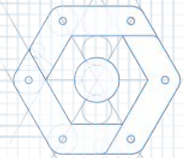


Control Plane

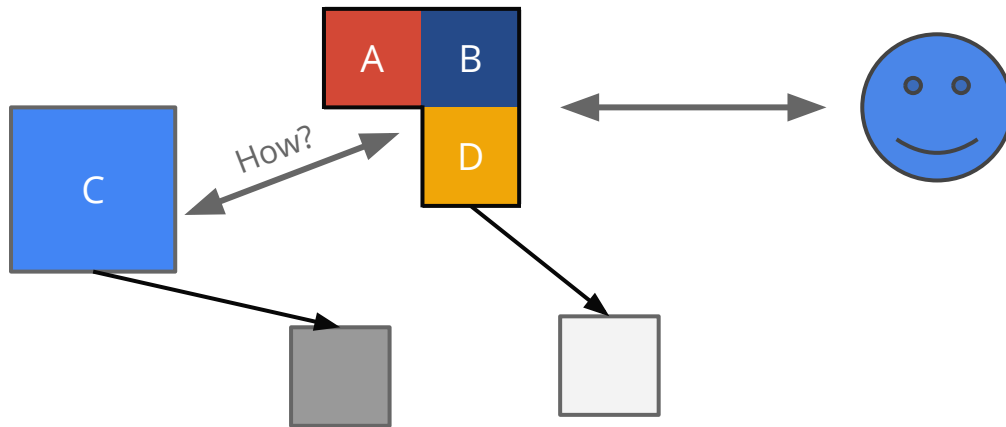
Cluster of machines as one

Well-defined API & types

Abstraction of infrastructure



Let's see it...



Beyond Deployment

Load Balancing

Fault Tolerance

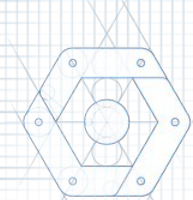
Observability & Insight

Monitoring & Tracing

Circuit Breaking



Popular Open Source Tools



Eureka - Service Registry

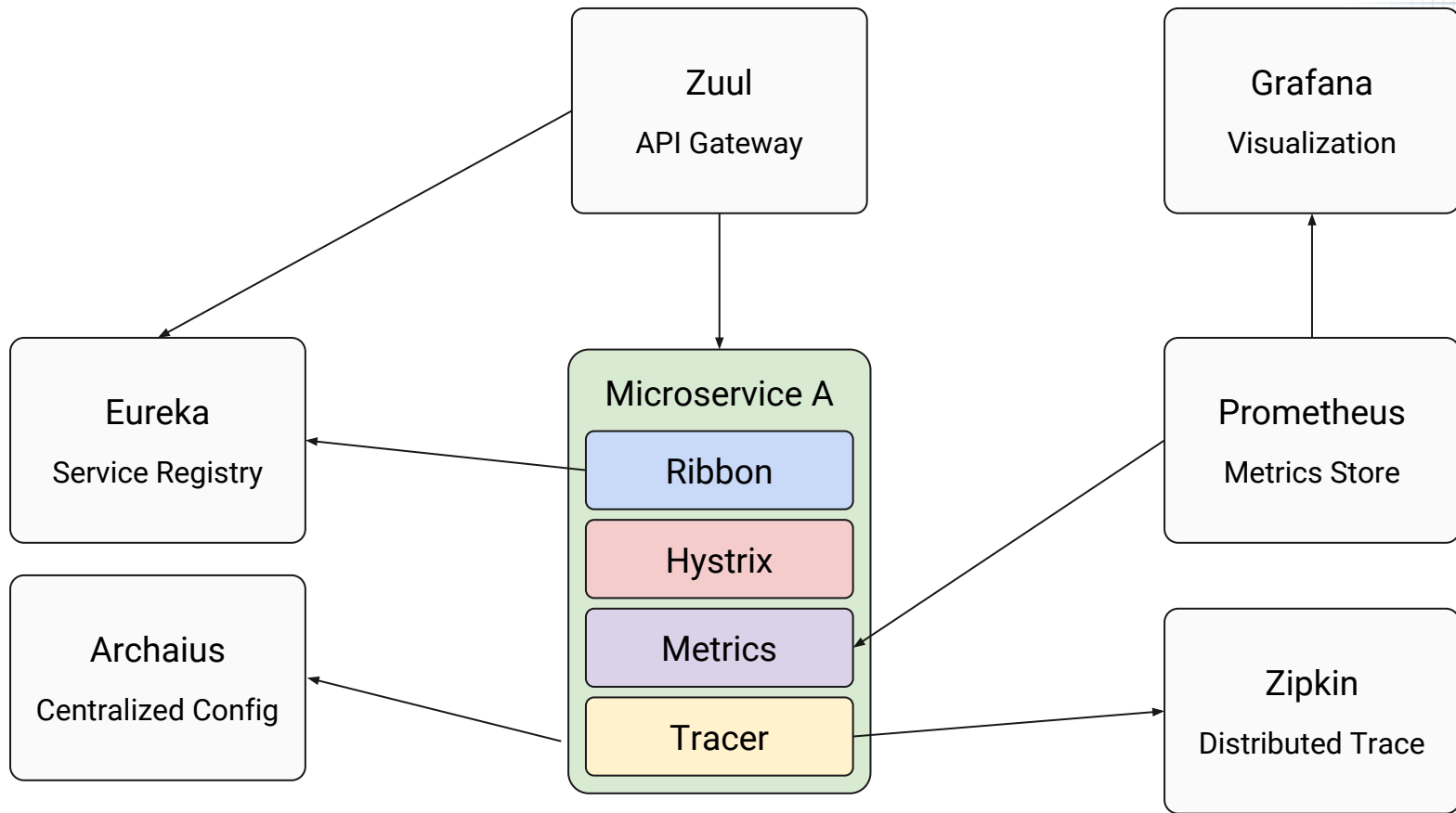
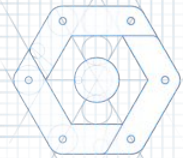
Ribbon - Client Side LB

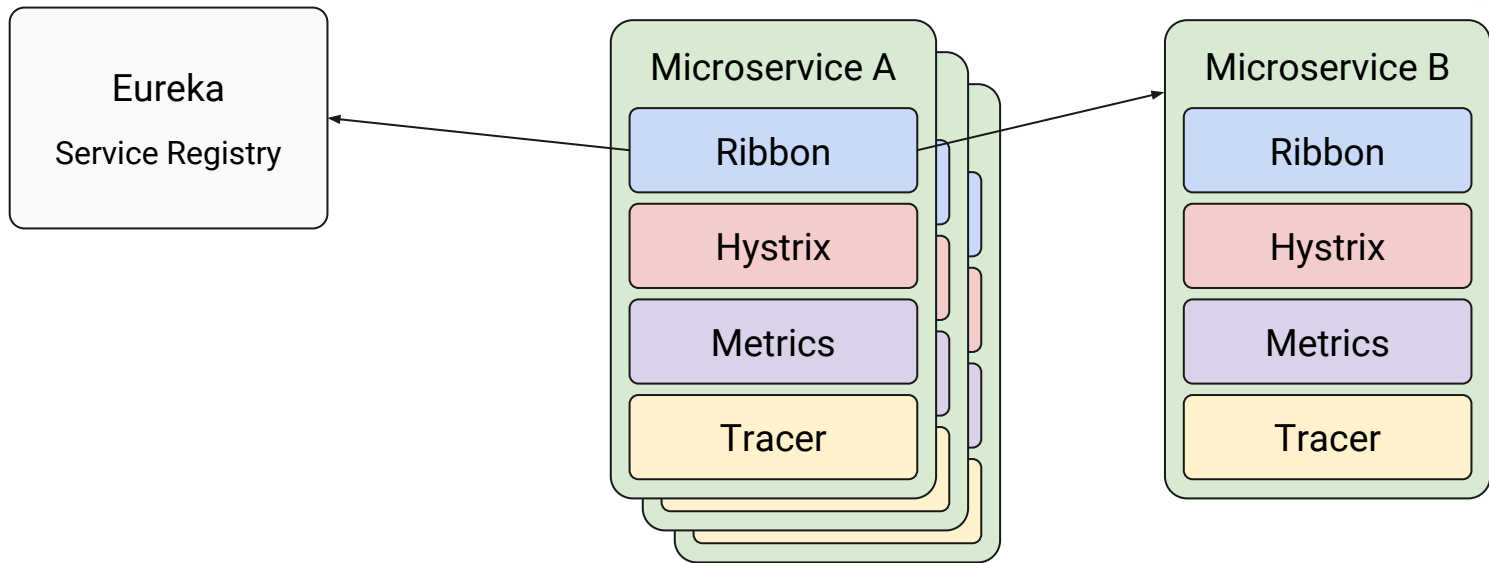
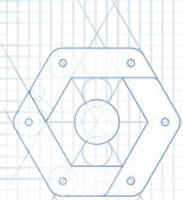
Hystrix - Circuit Breaker

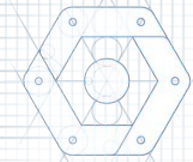
Zipkin - Distributed Tracing

Prometheus - Monitoring

Grafana - Data Visualization



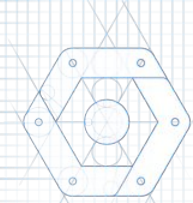




This is easy when...

Single stack

Framework w/ Spring Boot



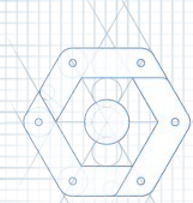
This becomes more difficult...

Multiple stack

Multiple frameworks

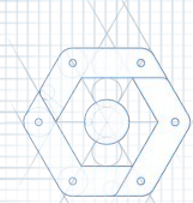
Polyglot

Legacy



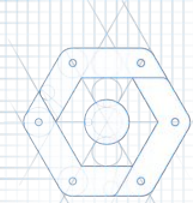
At the end of the day...

Let Microservice A talk to Microservice B!



As simple as...

Making a HTTP request?



Enter Istio, a Service Mesh!

A complete framework for **connecting, securing, managing and monitoring** services



Secure and monitor traffic for microservices **and** legacy services

An **open platform** with key contributions from Google, IBM, Lyft and others

Multi-environment and multi-platform, but Kubernetes first

What Where When How

Control Plane

Service to Service Communication

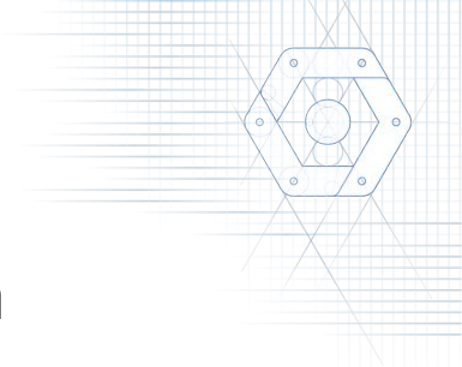
Routing Rules

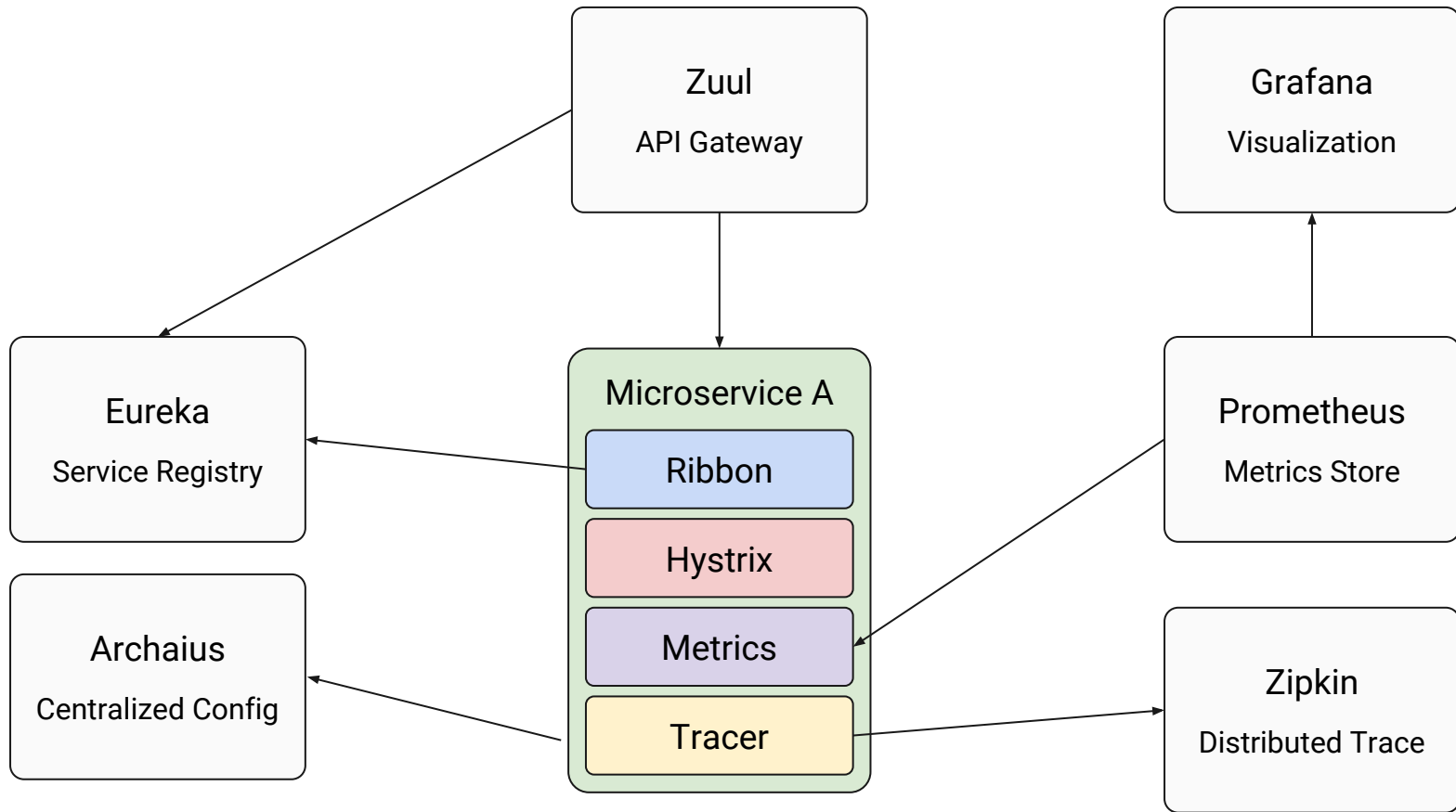
Retries

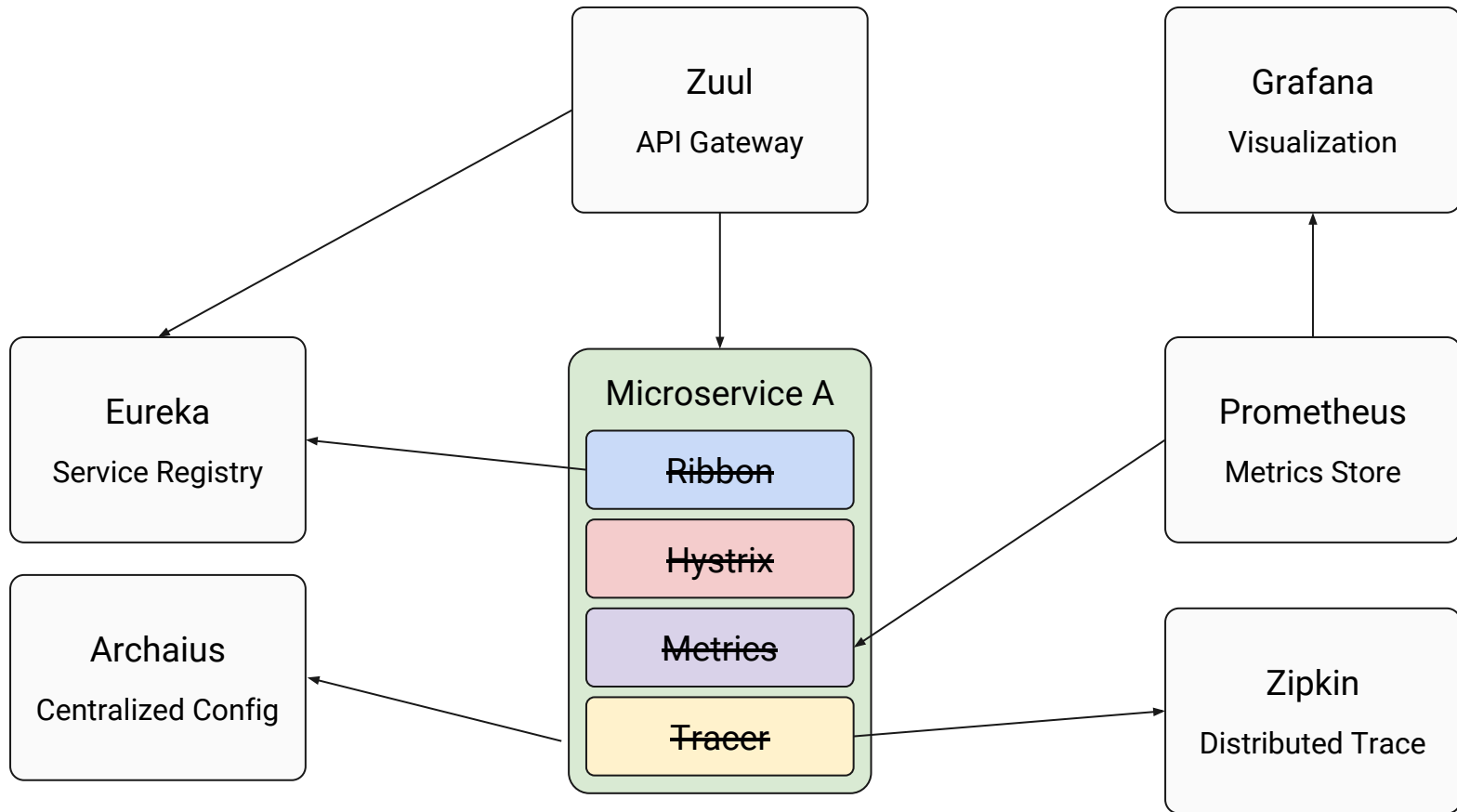
Circuit Breaker

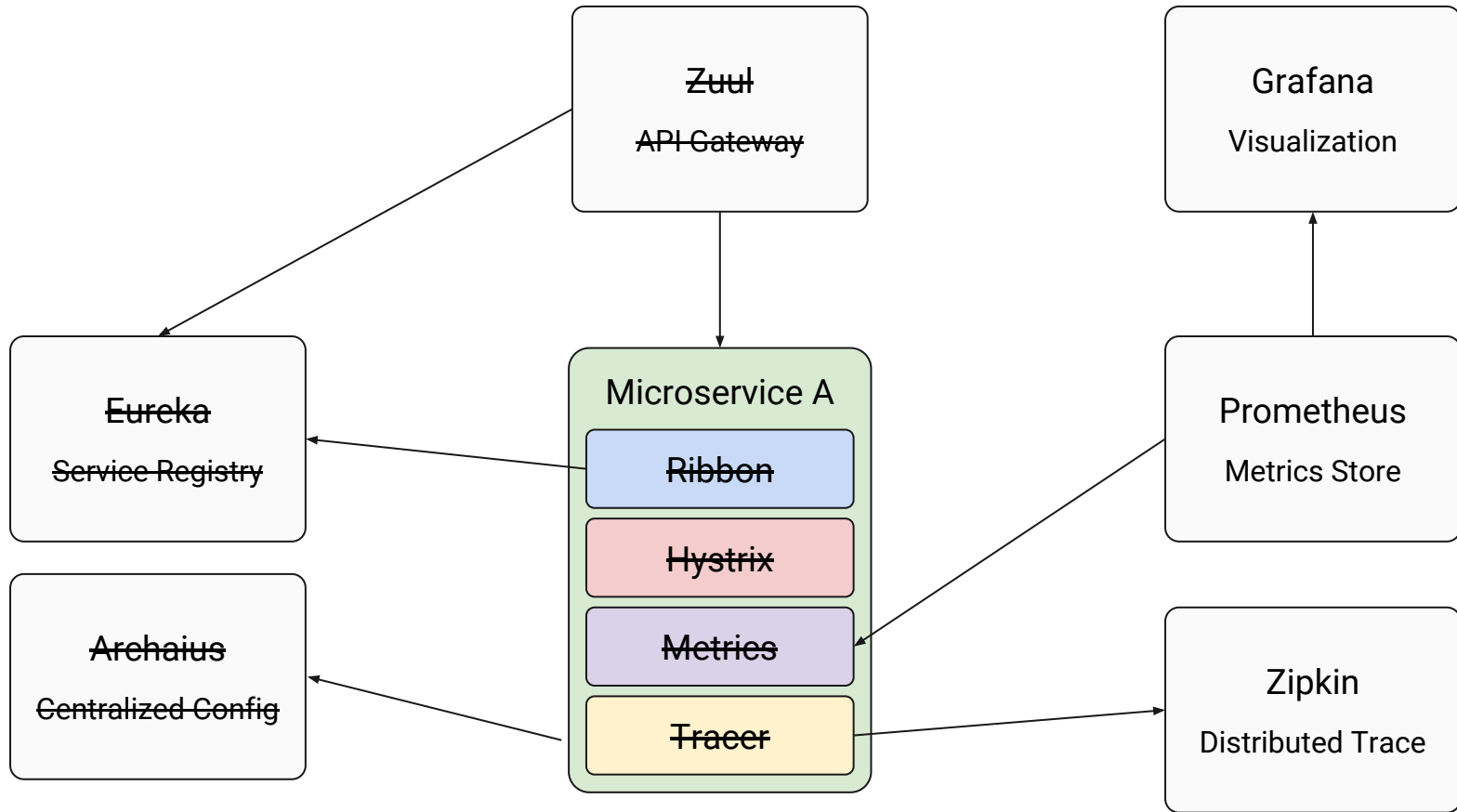
Performance Monitoring

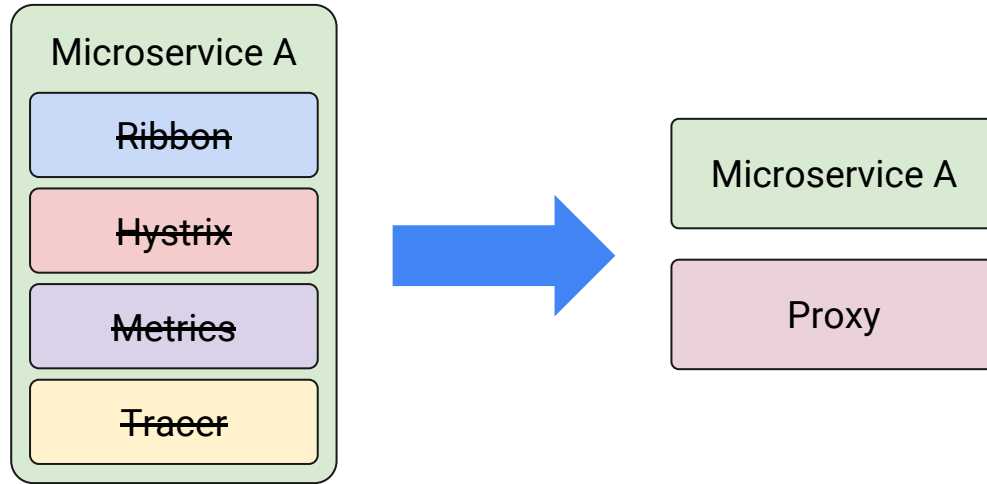
Tracing













A C++ based L4/L7 proxy

Low memory footprint

Battle-tested @ Lyft

100+ services

10,000+ VMs

2M req/s

An awesome team willing to work with the community!



Dynamic service discovery

Load balancing

TLS termination

HTTP/2

gRPC proxying

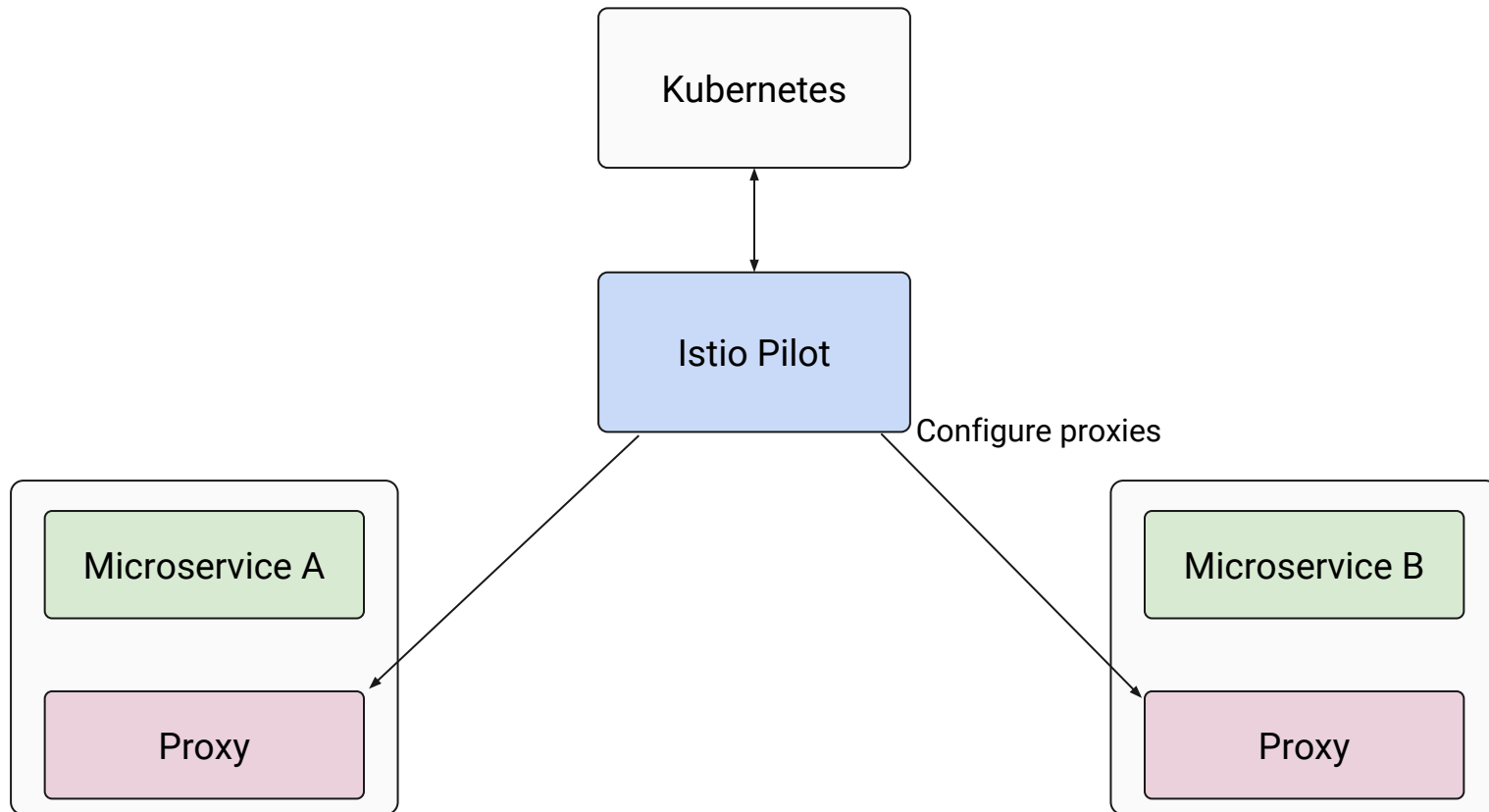
Circuit breakers

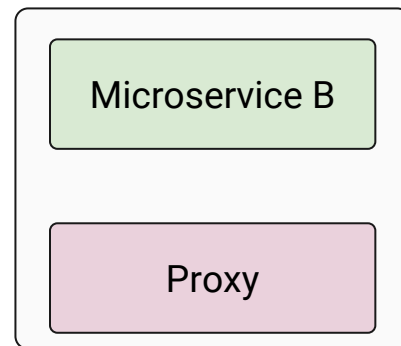
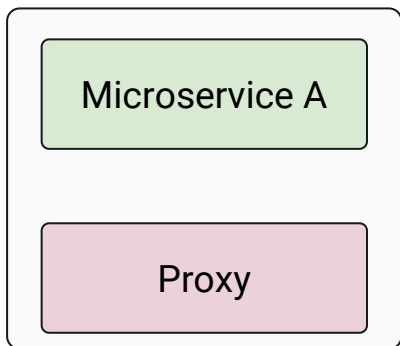
Health checks

Traffic split

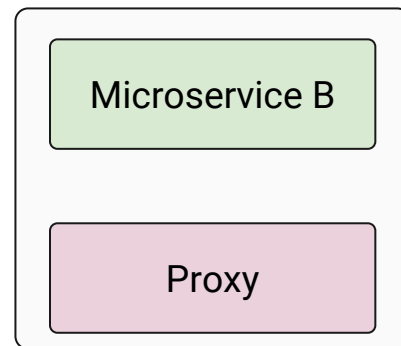
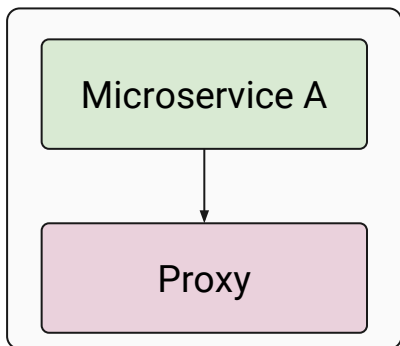
Fault injection

...





Service calls <http://service-b/>



Service calls <http://service-b/>

Service Mesh transparently intercepts request, forwards to local proxy



Service calls <http://service-b/>

Service Mesh transparently intercepts request, forwards to local proxy

Proxy has a list of destinations, load balances the request to a destination proxy



Service calls <http://service-b/>

Service Mesh transparently intercepts request, forwards to local proxy

Proxy has a list of destinations, load balances the request to a destination proxy

If allowed, destination proxy forwards the request to Service B instance



Service calls <http://service-b/>

Service Mesh transparently intercepts request, forwards to local proxy

Proxy has a list of destinations, load balances the request to a destination proxy

Destination proxy checks with a mixer to enforce policy, quota, ACL, etc

Service B response goes back to the caller



Service calls <http://service-b/>

Service Mesh transparently intercepts request, forwards to local proxy

Proxy has a list of destinations, load balances the request to a destination proxy

Destination proxy checks with a mixer to enforce policy, quota, ACL, etc

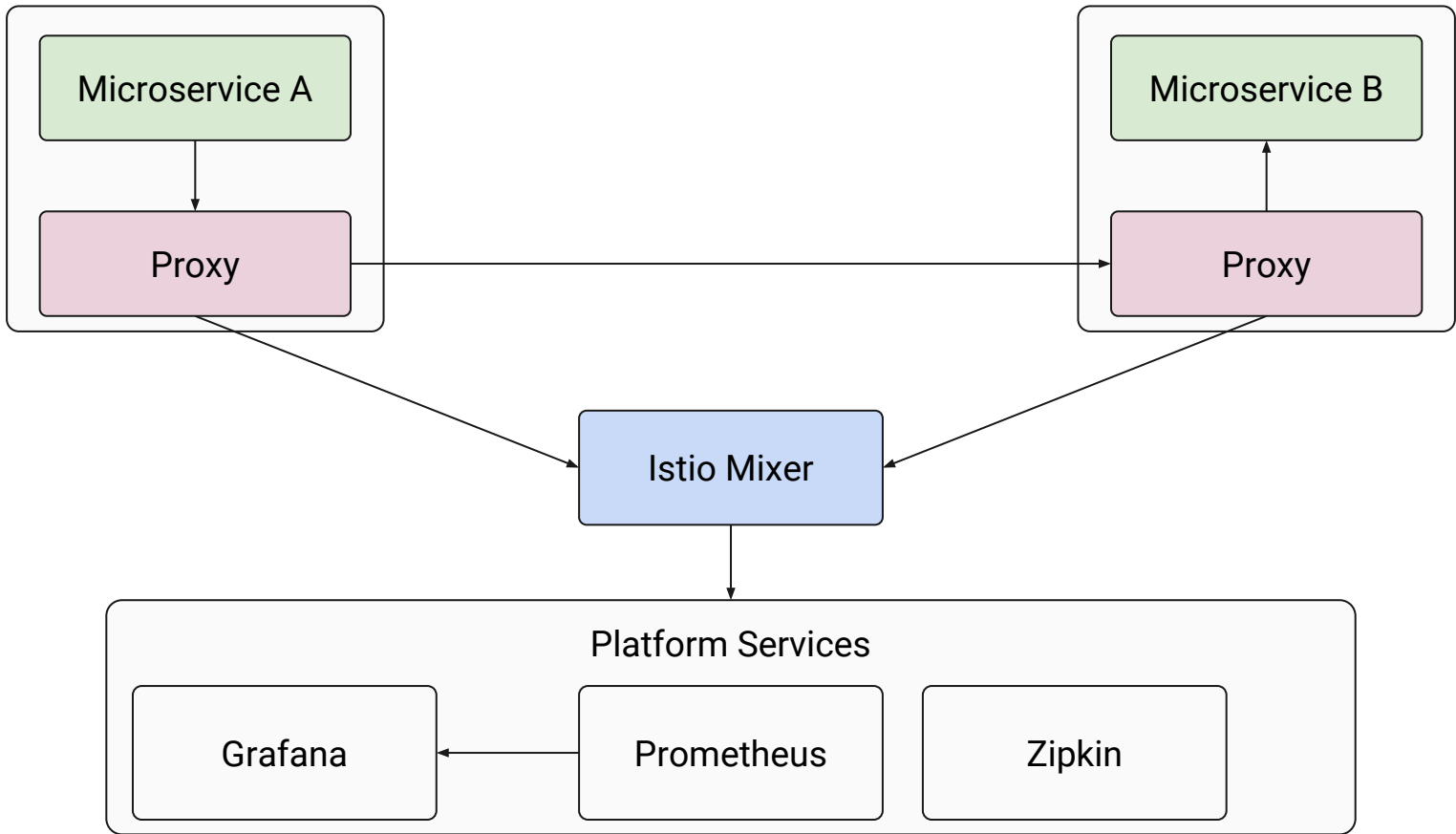
If allowed, destination proxy forwards the request to Service B instance

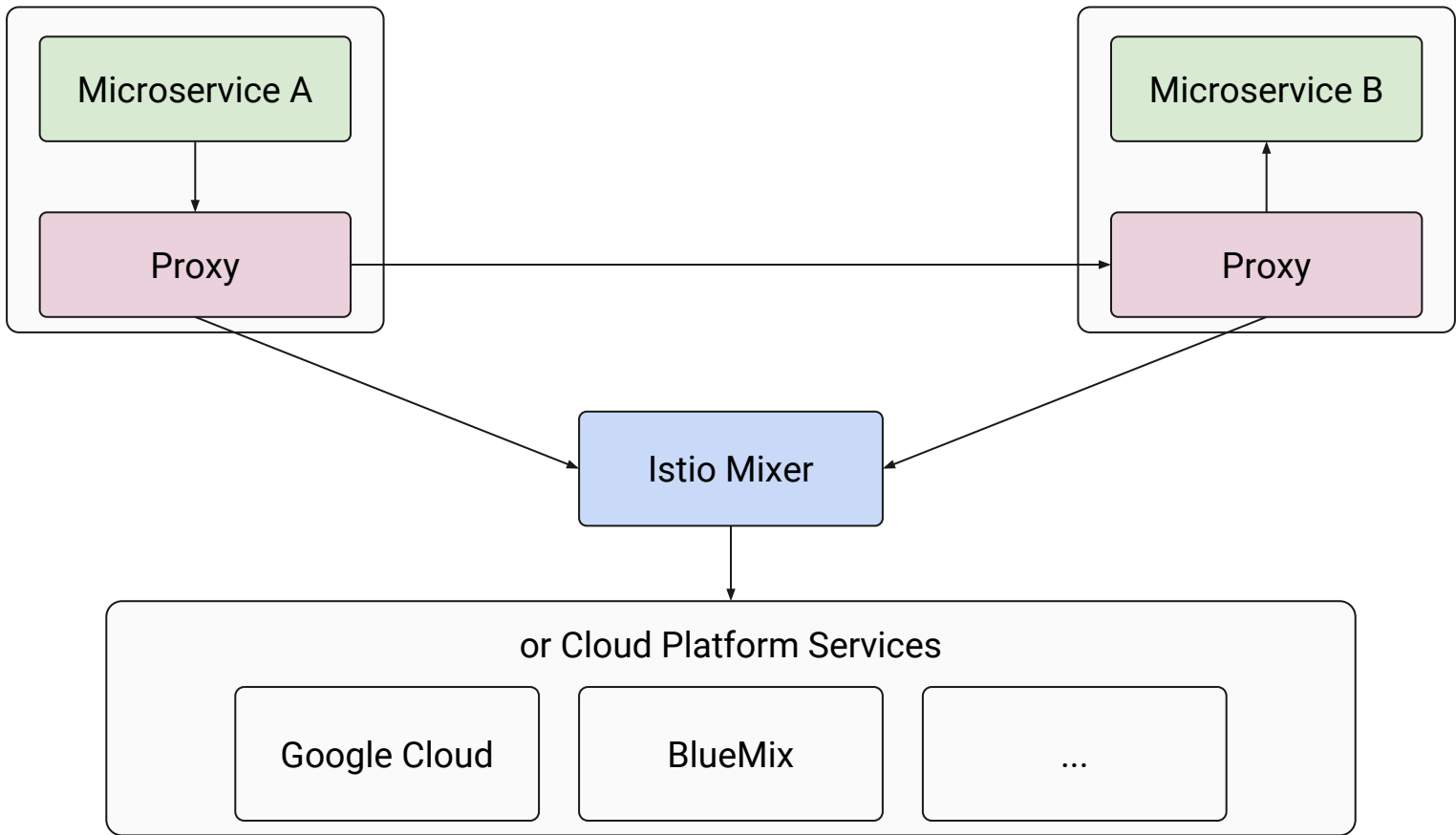
Service B response goes back to the caller

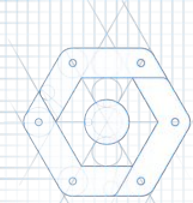


Through the Proxy

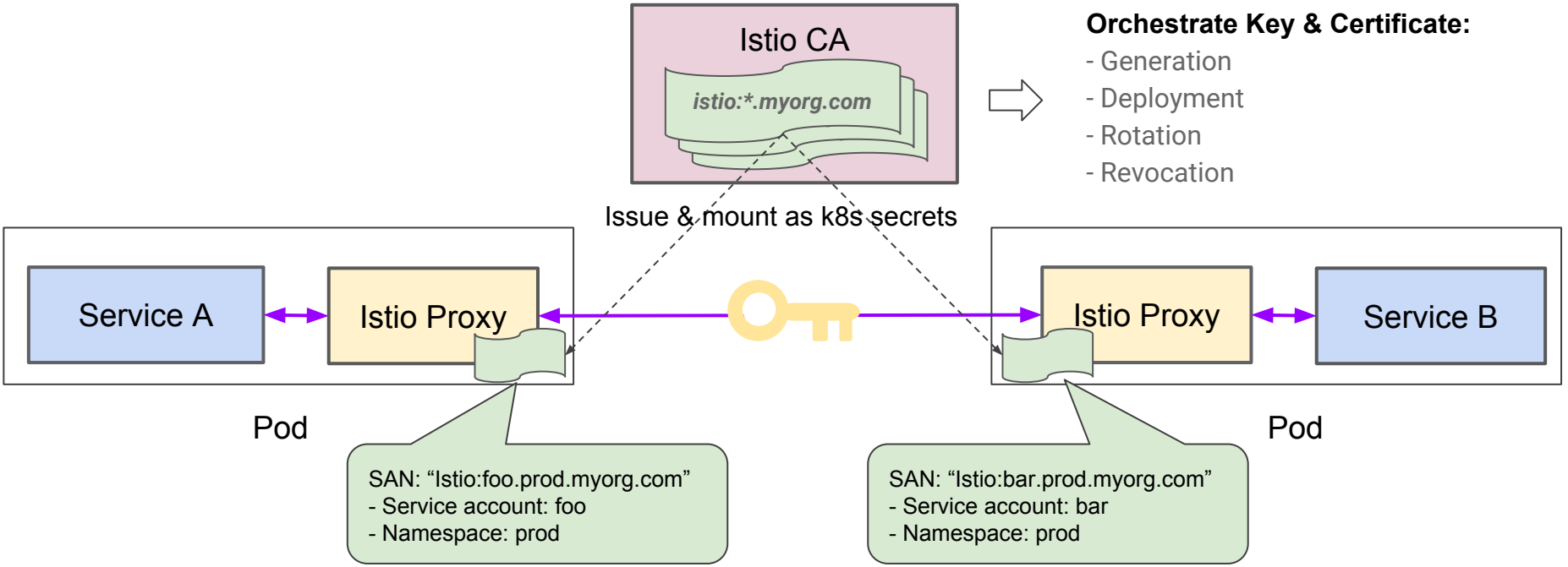
- Traffic Control - enforce routing rules & policies
- Resiliency - Circuit Breaker, Retries
- Monitoring - Record metrics
- Observability - Record traces
- Security - Mutual TLS! Encryption







Let's see it...





Visibility

Resiliency & Efficiency

Traffic Control

Security

Policy Enforcement

- 0.1: a single Kubernetes namespace
- 0.2 (just launched): a single Kubernetes cluster and external VMs
- 0.3 (by end of year): production readiness within a single cluster
- 1.0: (2018): complete mesh across all environments



Getting started

- Install Kubernetes (v1.7+ for Initializers)
 - [Google Container Engine Alpha clusters](#)
- [istio.io](#) quickstart
- [Helm chart](#)
 - `helm install incubator/istio`
- Take a lab!



Thank you!

Learn more on istio.io

Let us know on istio-users@googlegroups.com

Examples on github.com/saturnism/istio-by-example-java

Try our Code Labs g.co/codelabs/cloud!