



Solucionando Problemas de Microsserviços com Service

Mesh: Istio e Envoy

Edson Yanaga (@yanaga)

bit.ly/istio-tutorial



@yanaga



Edson Yanaga



Raffle Rules



Edson Yanaga

- Follow:  @yanaga
- With a picture of the session
- Mention @yanaga
- With hashtag #QConSP

O'REILLY®

Migrating to Microservice Databases

From Relational Monolith
to Distributed Data



Edson Yanaga

Compliments of
**RED HAT
DEVELOPERS**

bit.ly/mono2microdb

O'REILLY®



Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant
Cloud Native Applications



Burr Sutter & Christian Posta

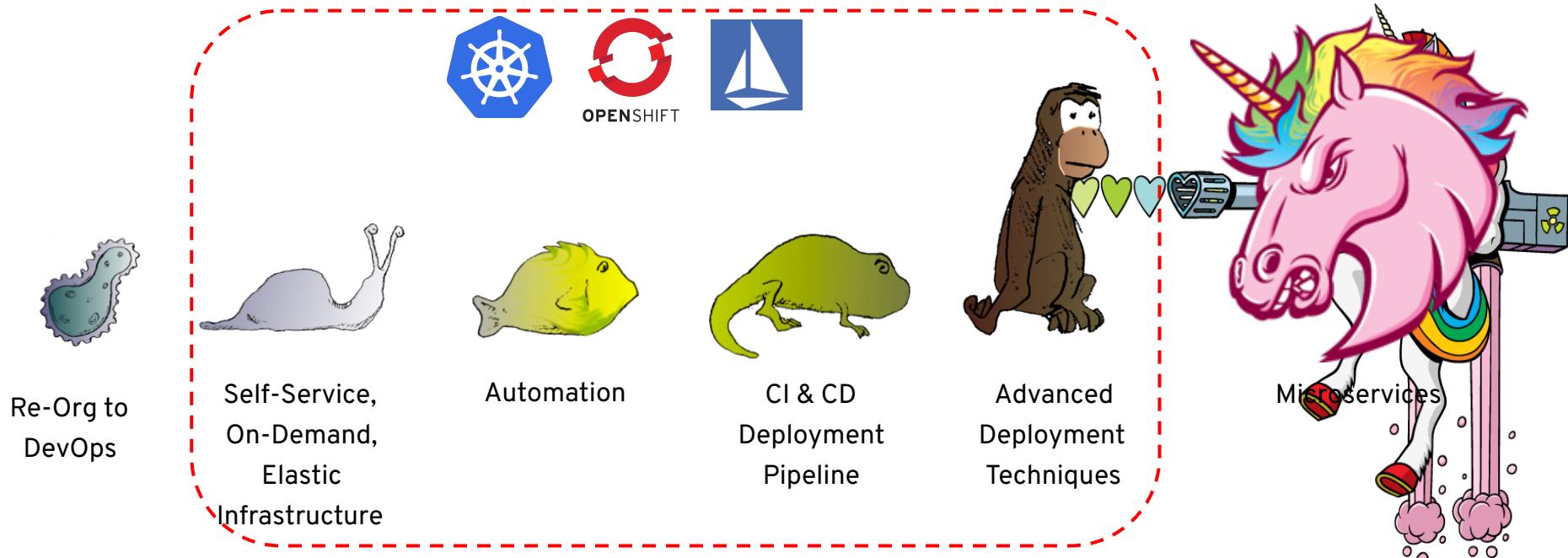
bit.ly/istiobook

2nd Edition

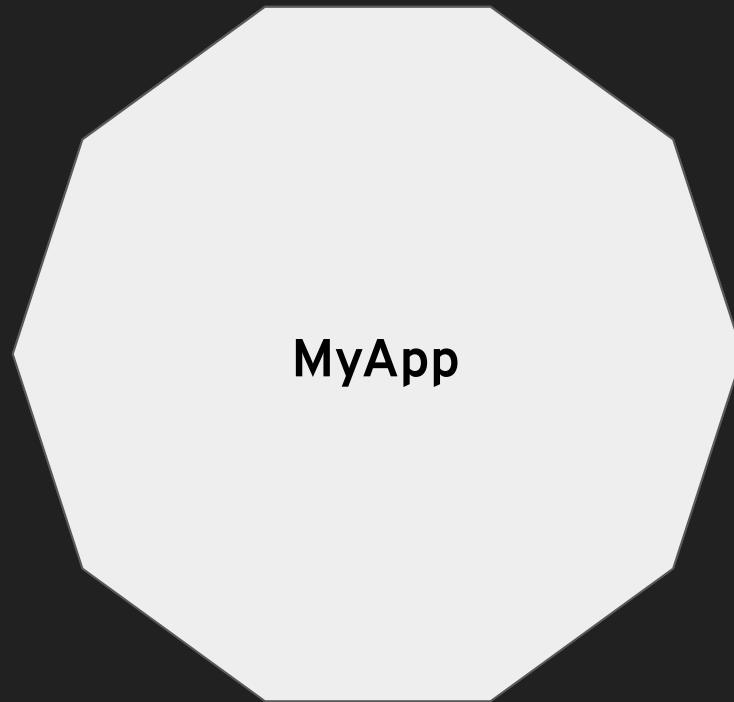
Agenda

- Why Service Mesh
- Observability
- Istio Architecture & Introduction
- Traffic Control
- Service Resiliency & Circuit Breaking
- Chaos Testing
- Egress
- Security

Your Journey to Awesomeness

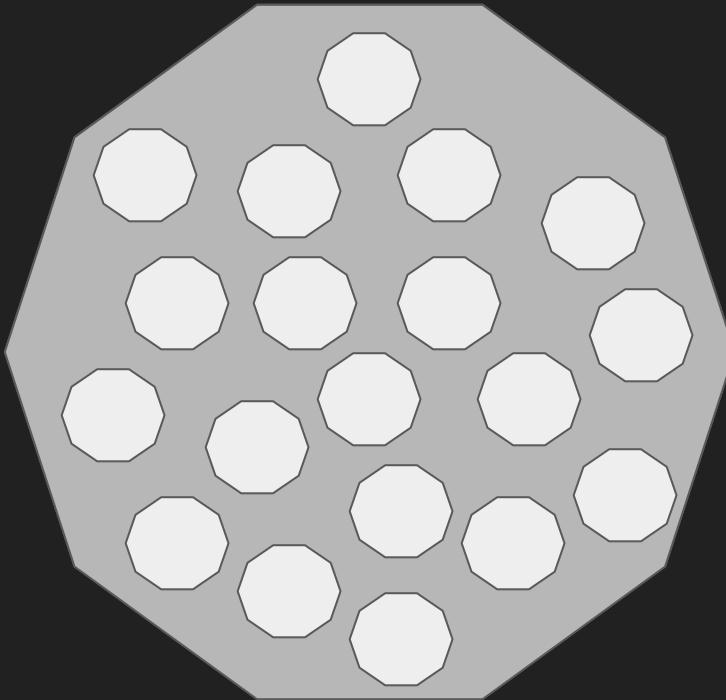


Monolith

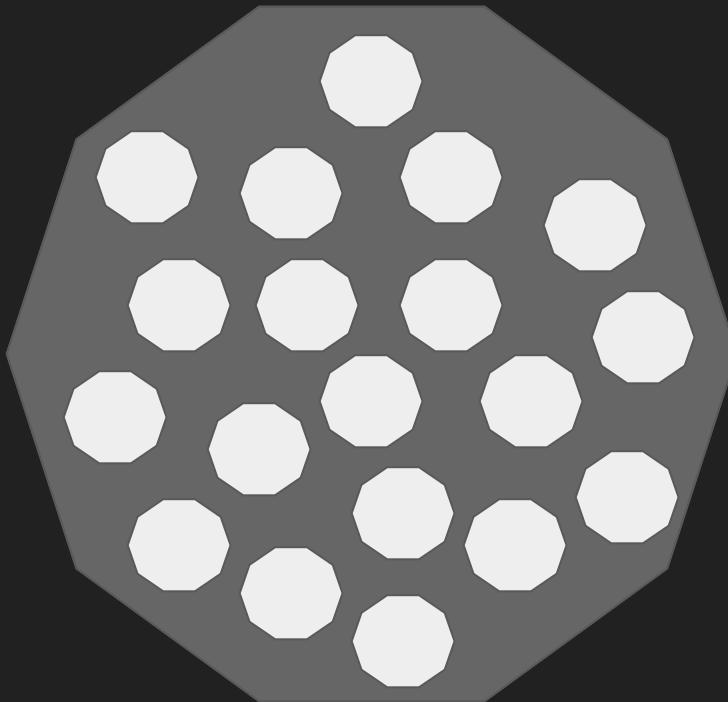


MyApp

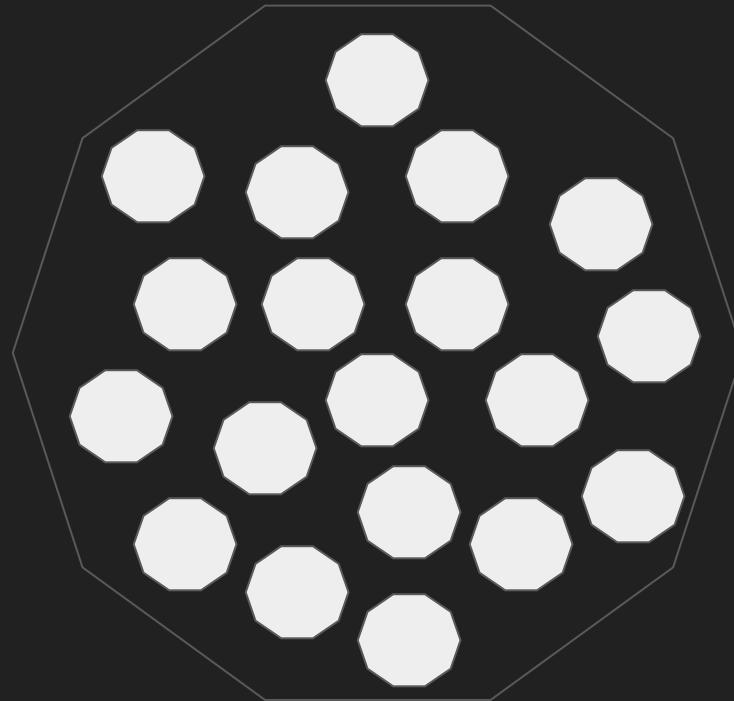
The Application



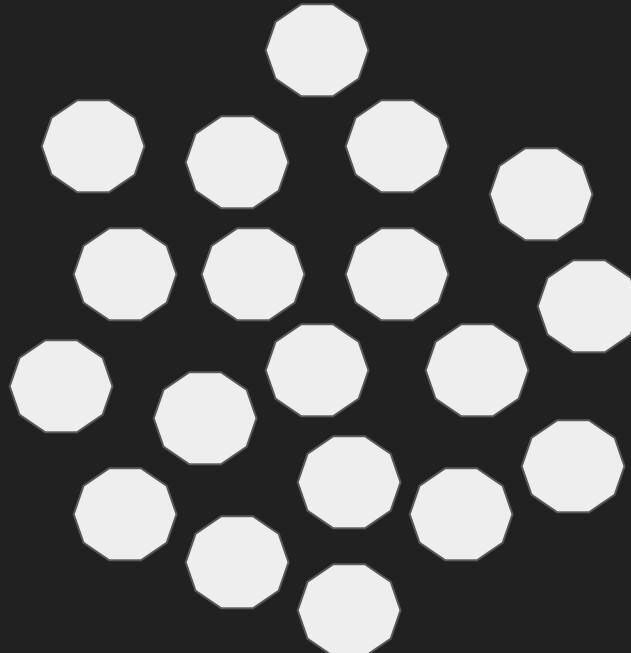
Modules



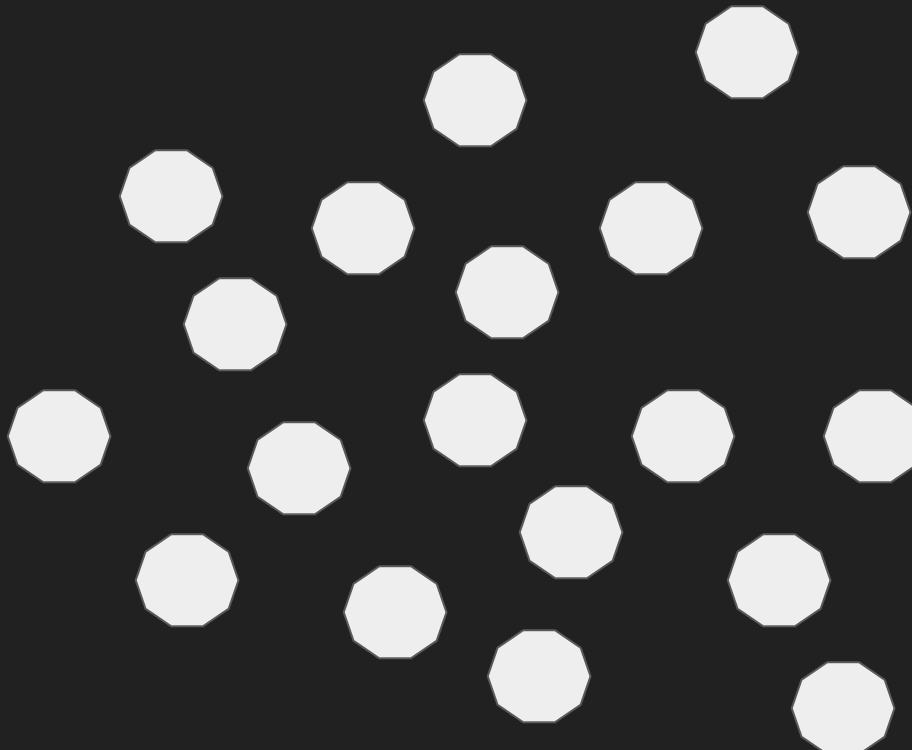
Microservices



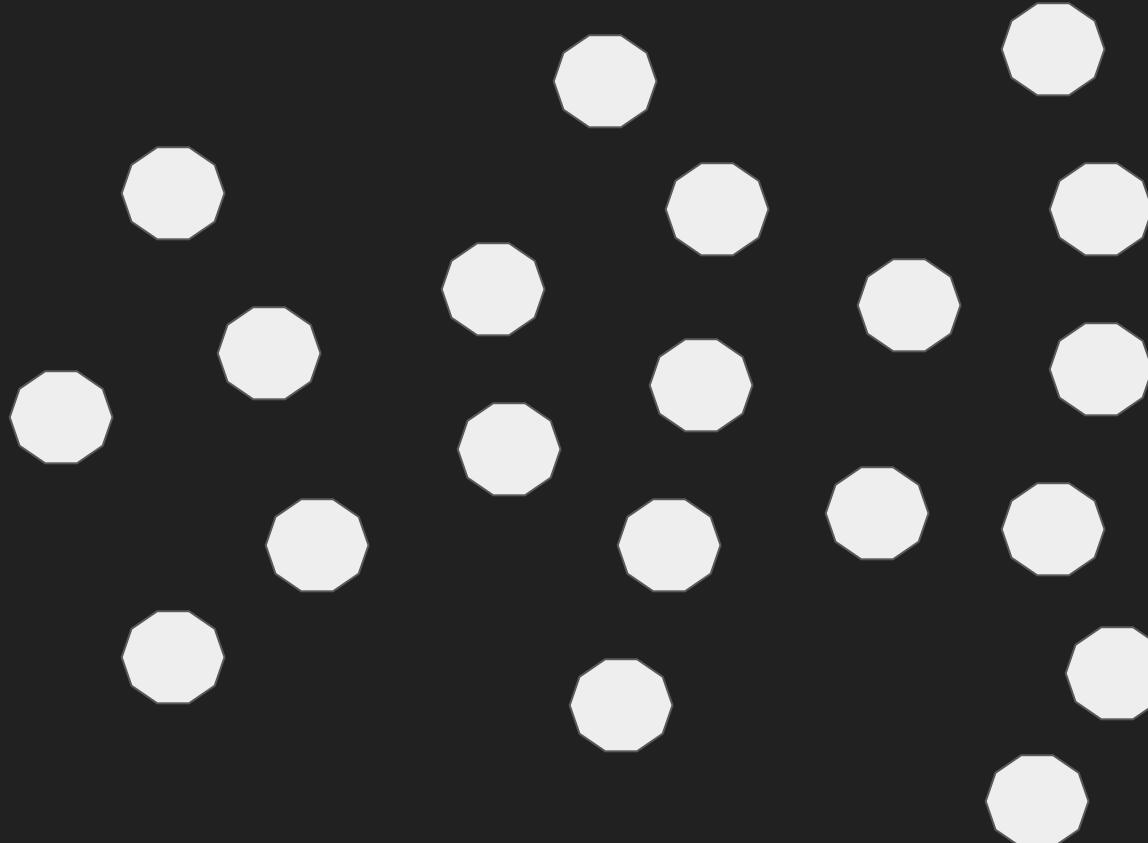
Microservices



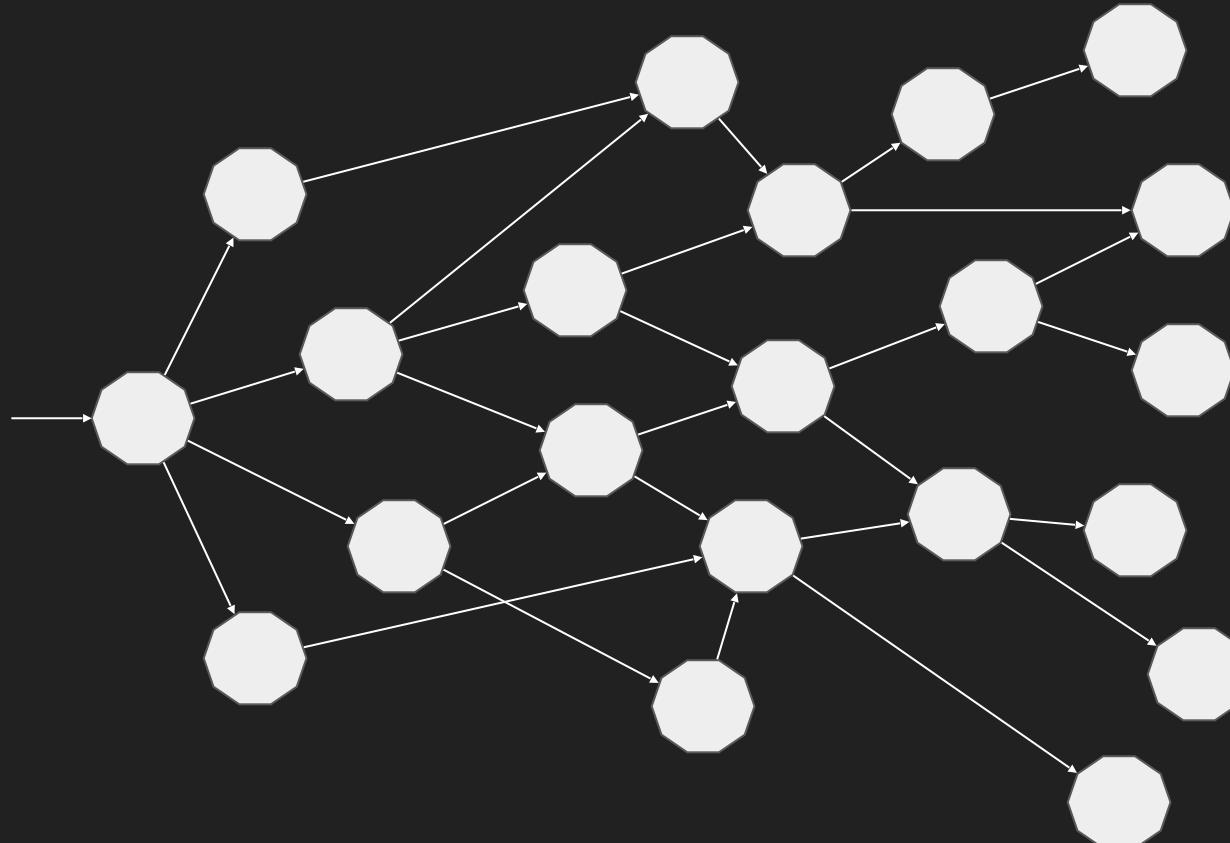
Microservices



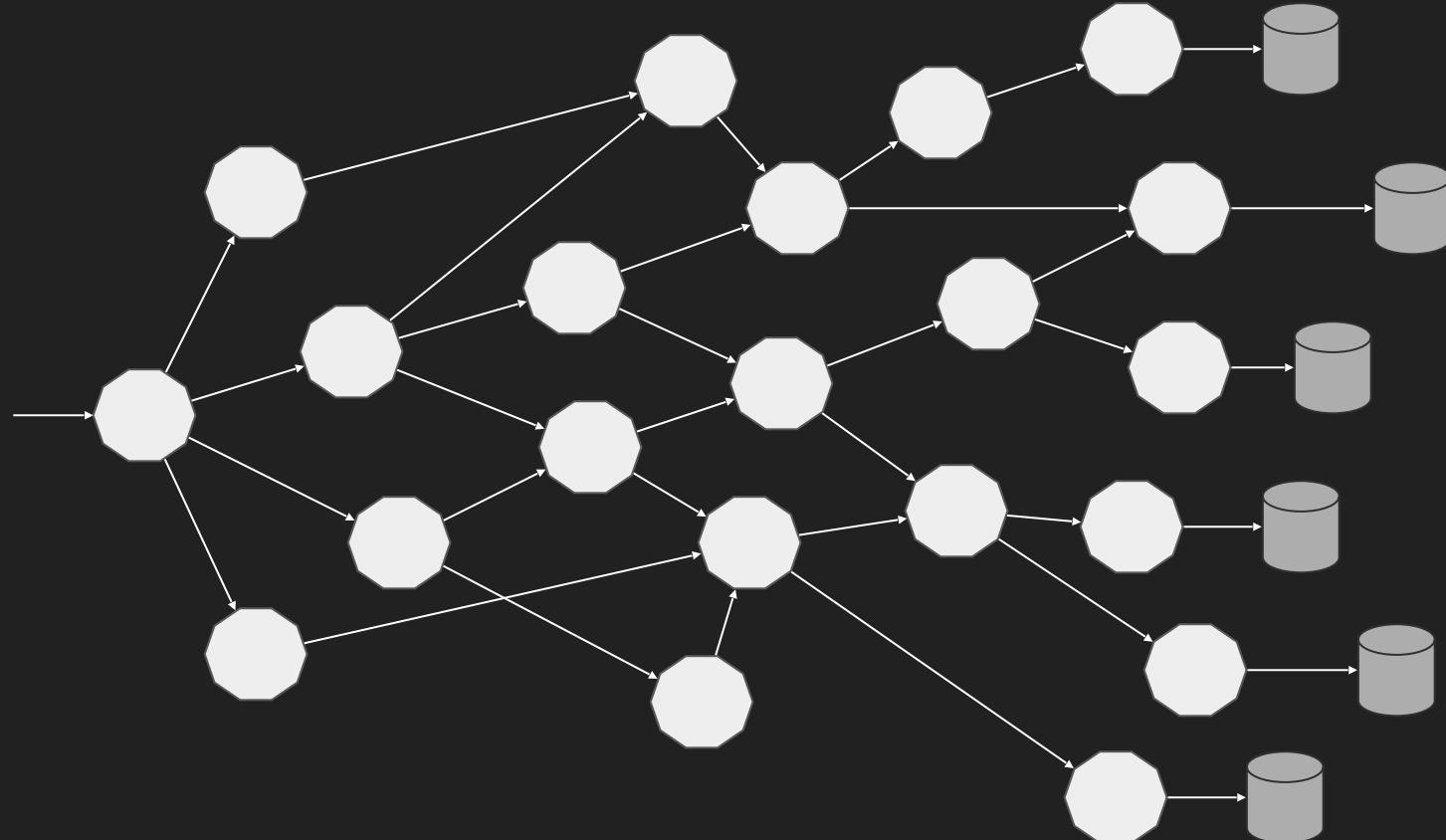
Microservices



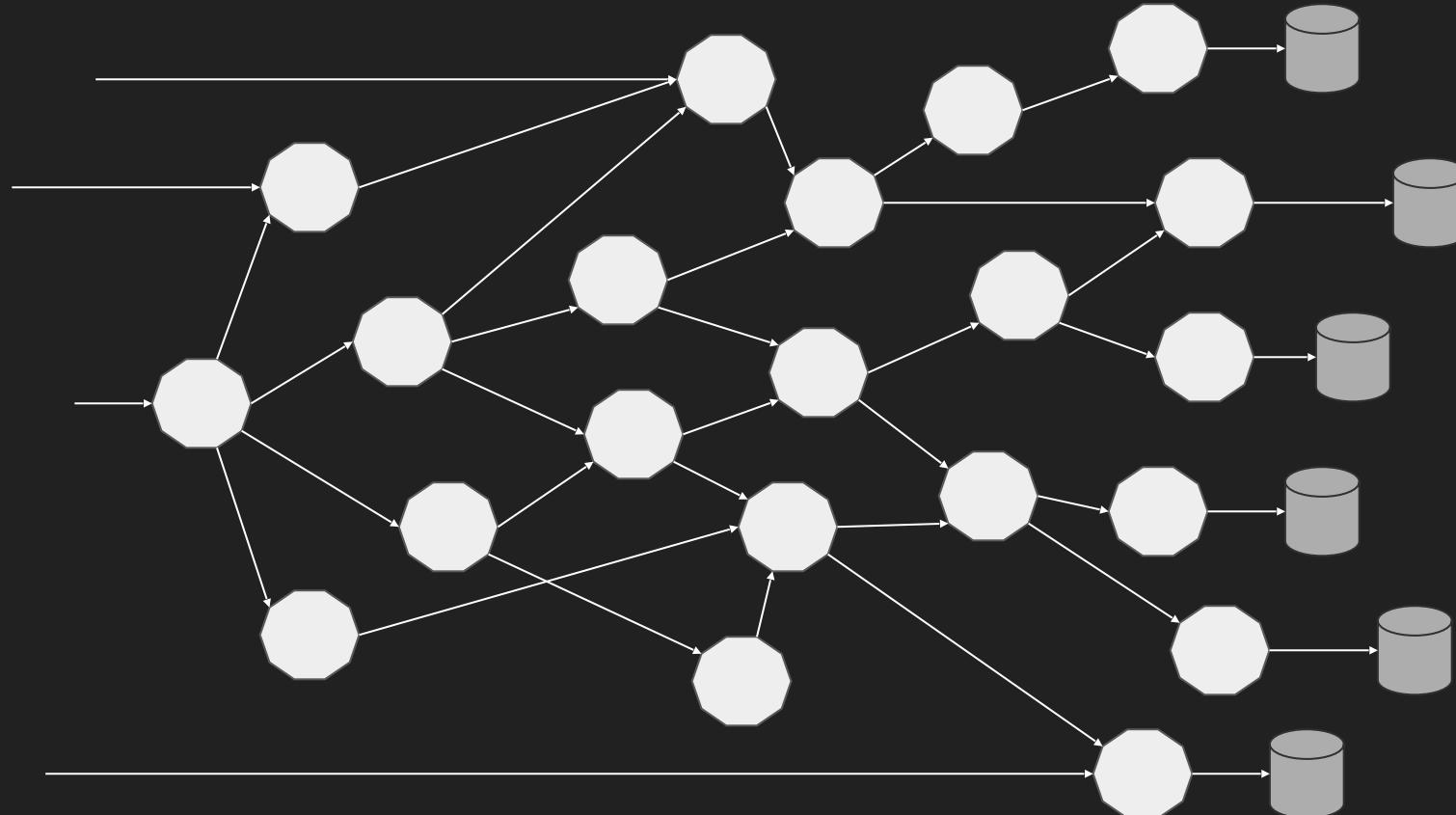
Network of Services



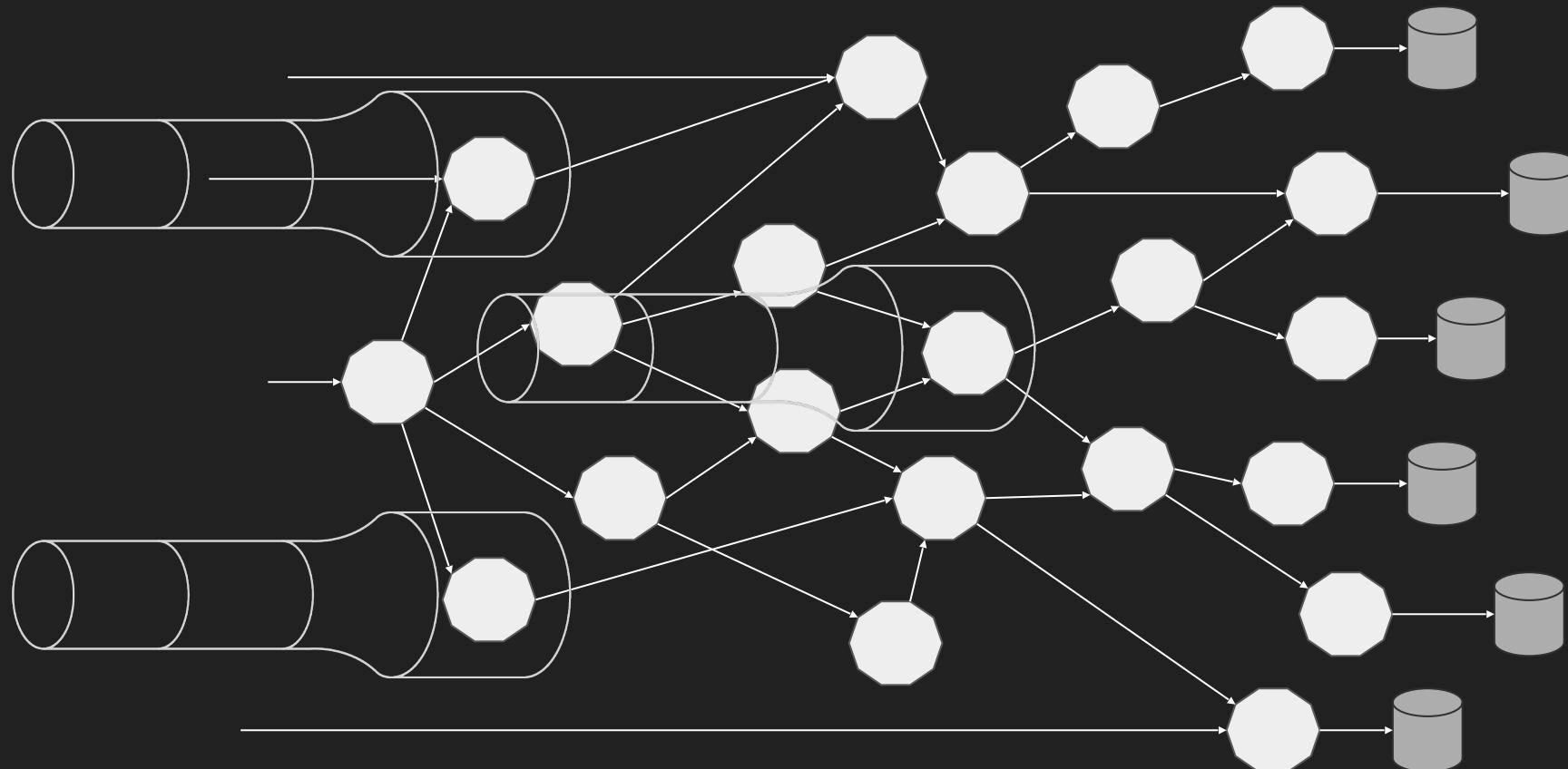
Microservices own their Data



Multiple Points of Entry

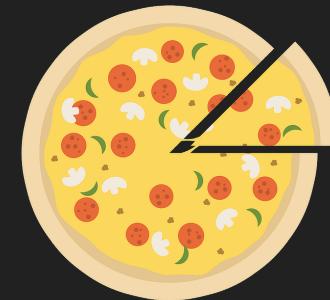


Multiple Teams, Multiple Pipelines



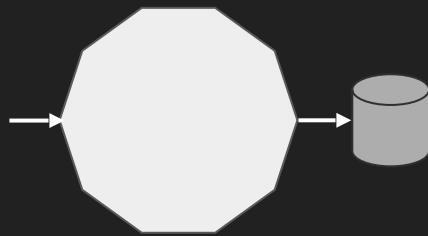
Microservices Principles

1. Deployment **Independence** - updates to an individual microservice have no negative impact to any other component of the system. Optimized for **Replacement**
2. Organized around **business** capabilities
3. **Products** not Projects
4. API Focused
5. Smart endpoints and dumb pipes
6. Decentralized Governance
7. Decentralized Data Management
8. Infrastructure Automation (infrastructure as code)
9. Design for failure
10. Evolutionary Design



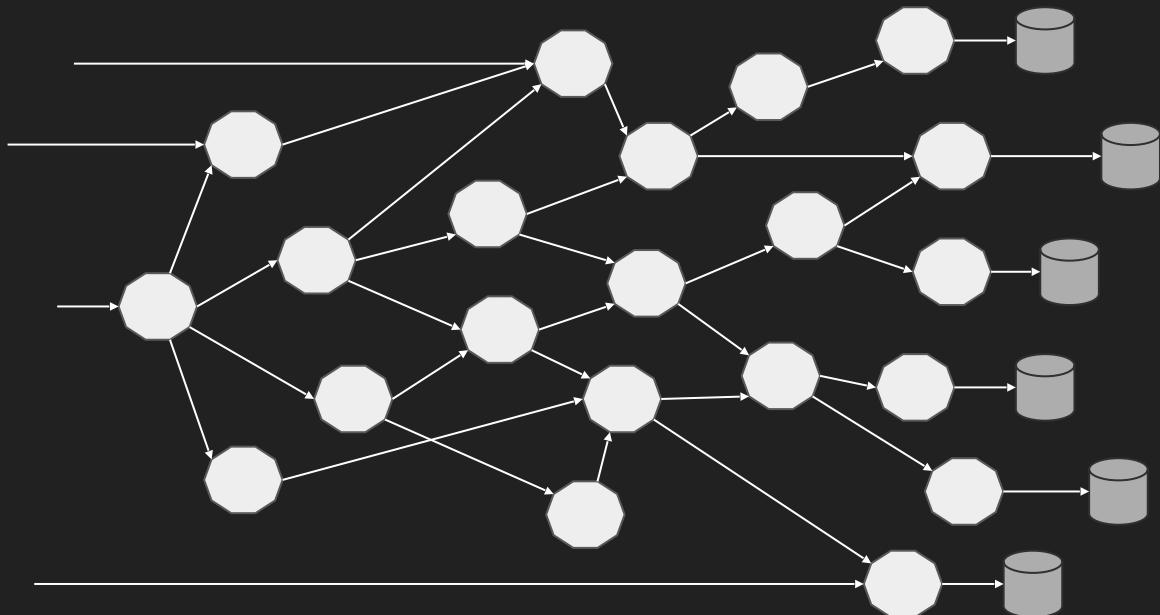
2 Pizza Team

Old School School



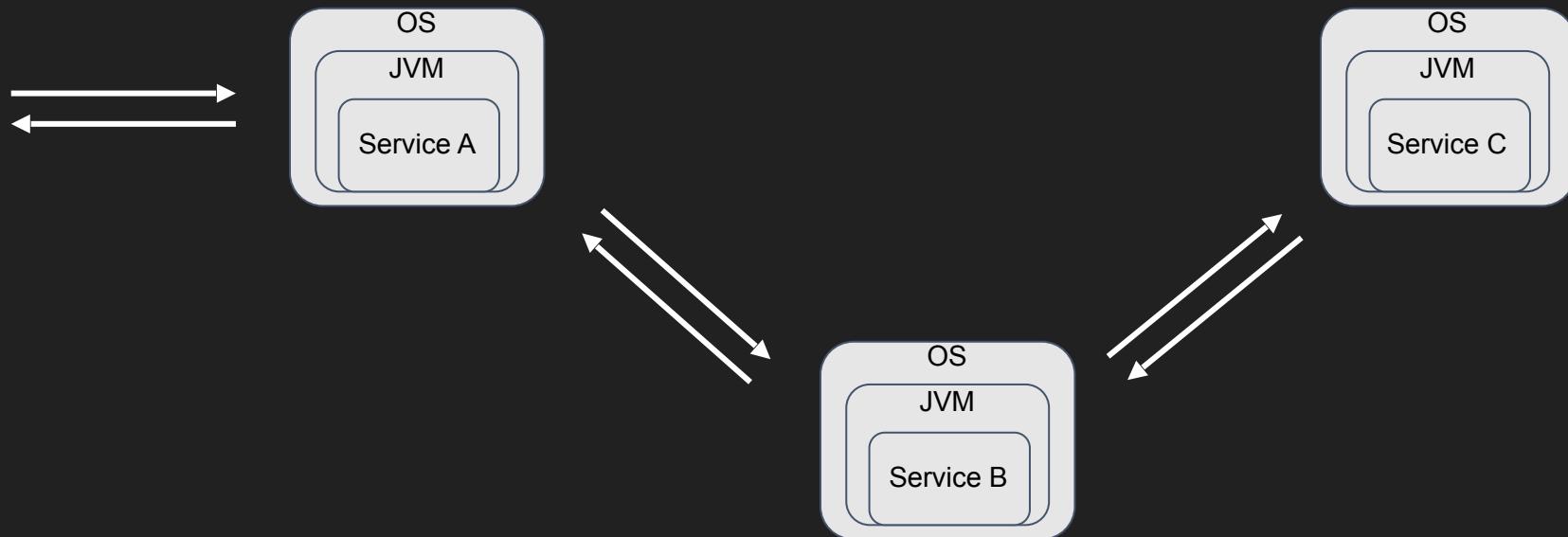
Love Thy Mono

New



OPENSHIFT

Microservices == Distributed Computing

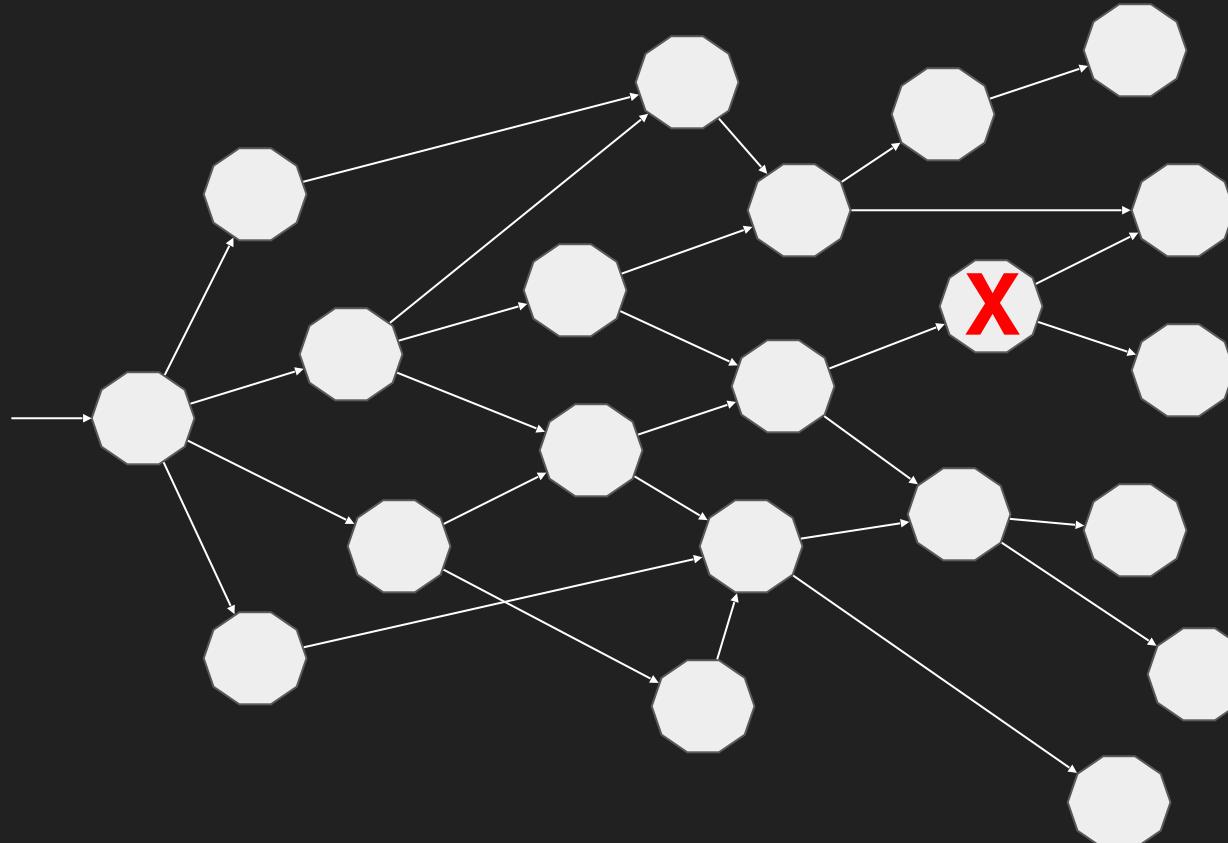


Fallacies of Distributed Computing

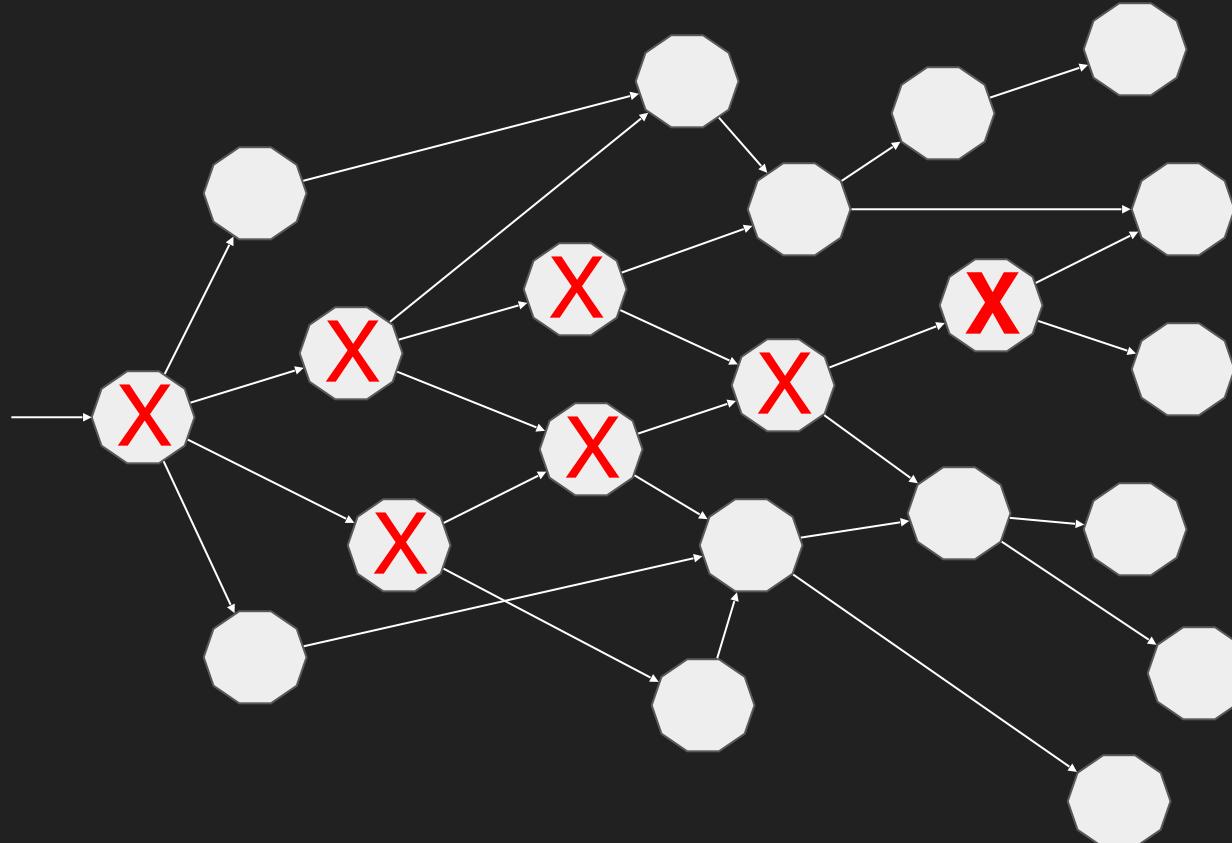
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

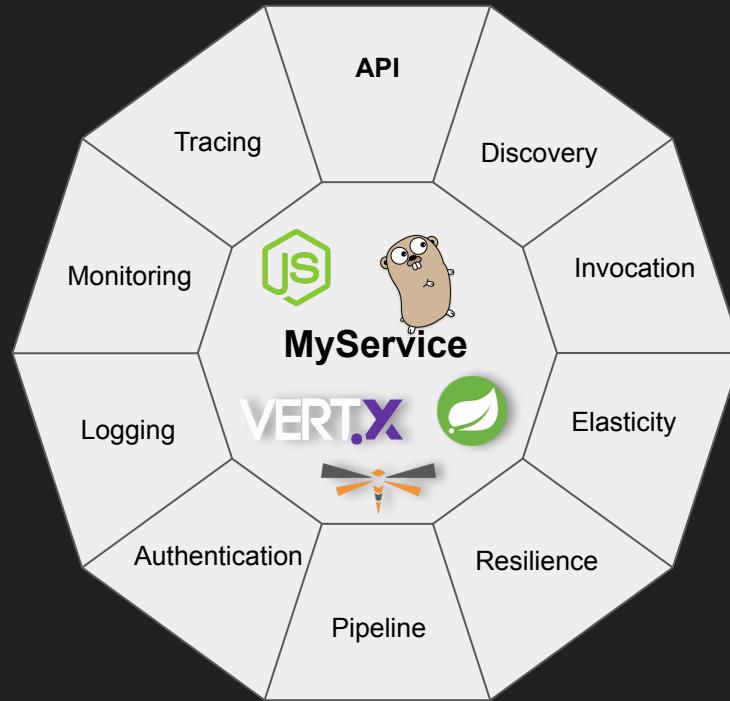
Failure of a Service



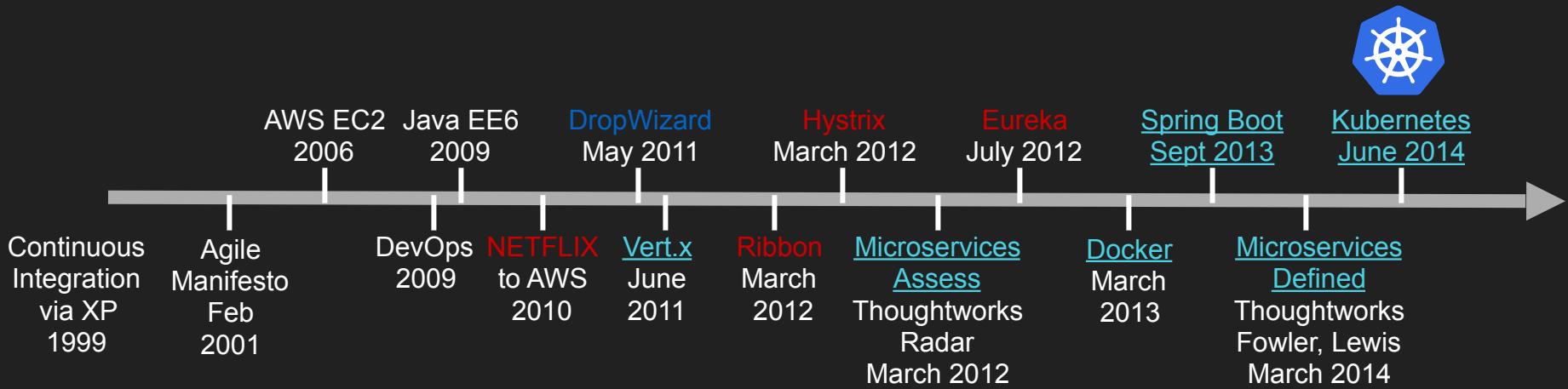
Cascading Failure



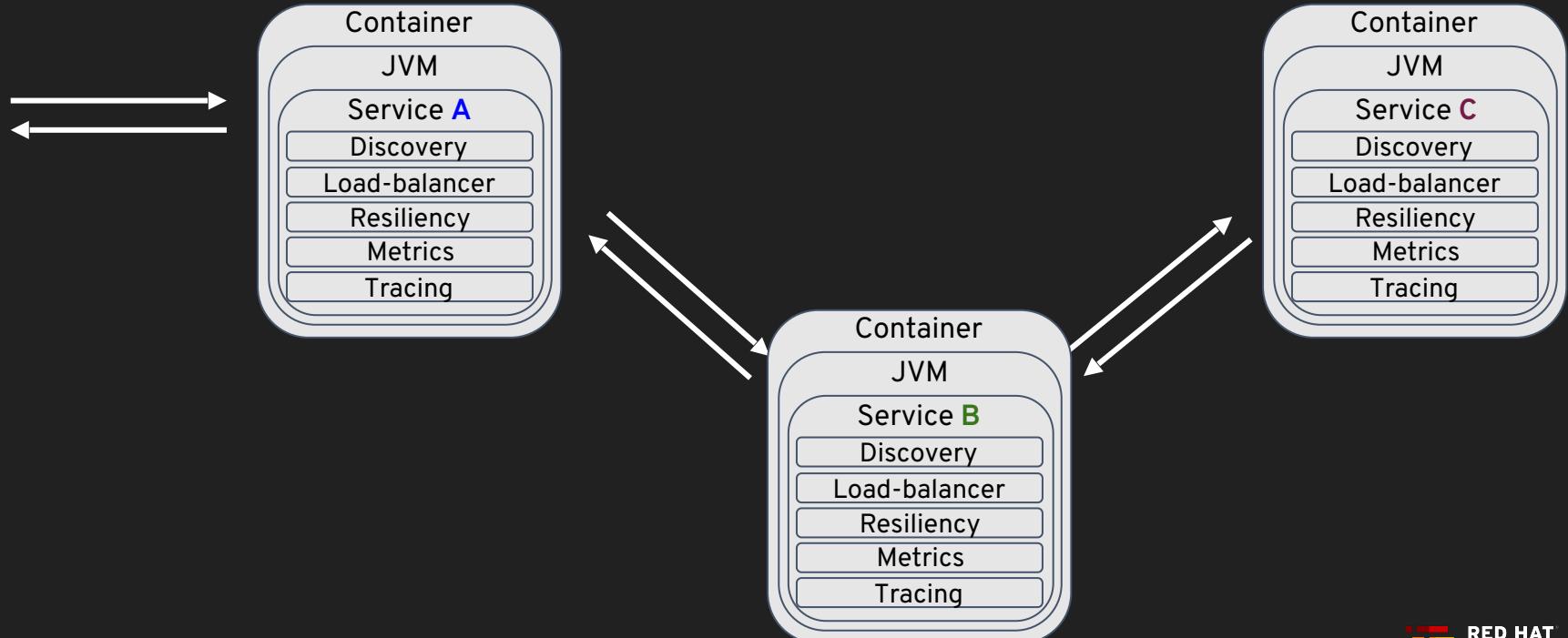
Microservices'ilities



History of Microservices



Microservices embedding Capabilities



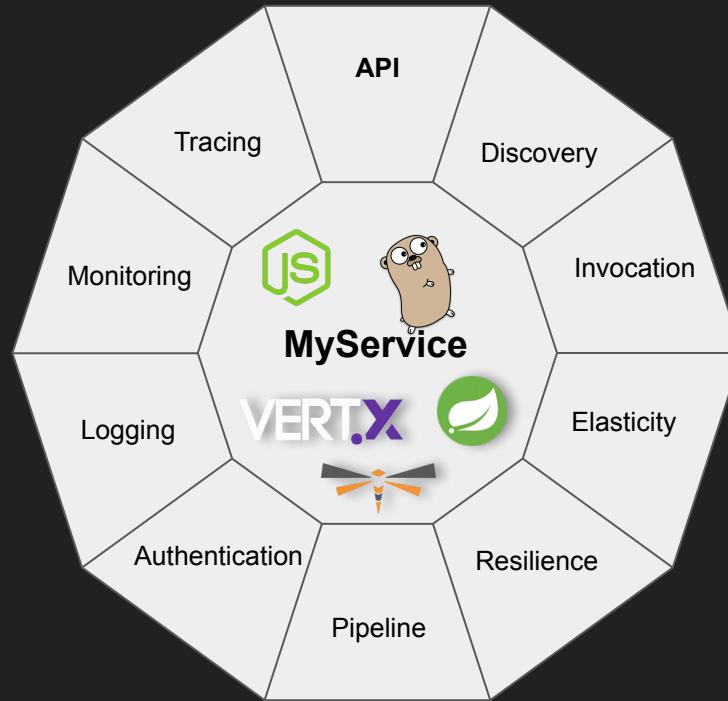
What's Wrong with Netflix OSS?

Java Only

Adds a lot of libraries to **YOUR** code



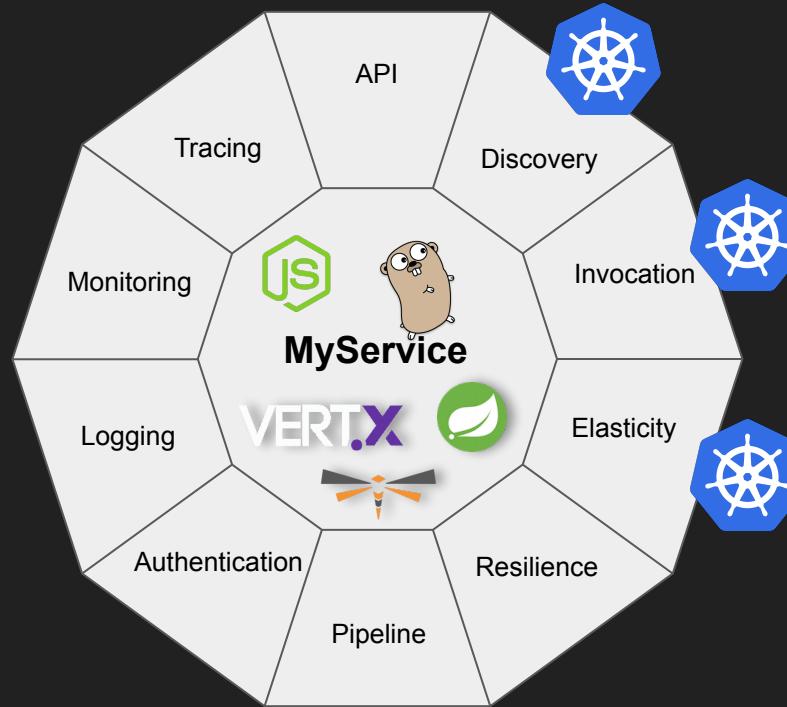
Microservices'ilities



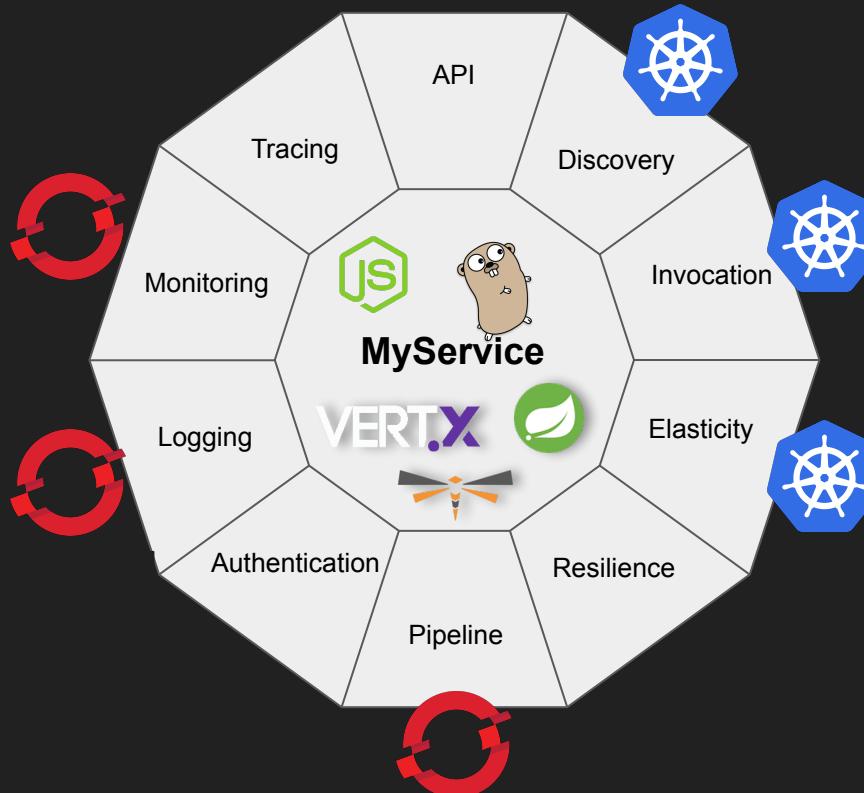


OPENSIFT

Microservices'ilities + Kubernetes



Microservices'ilities + OpenShift





Istio - Sail

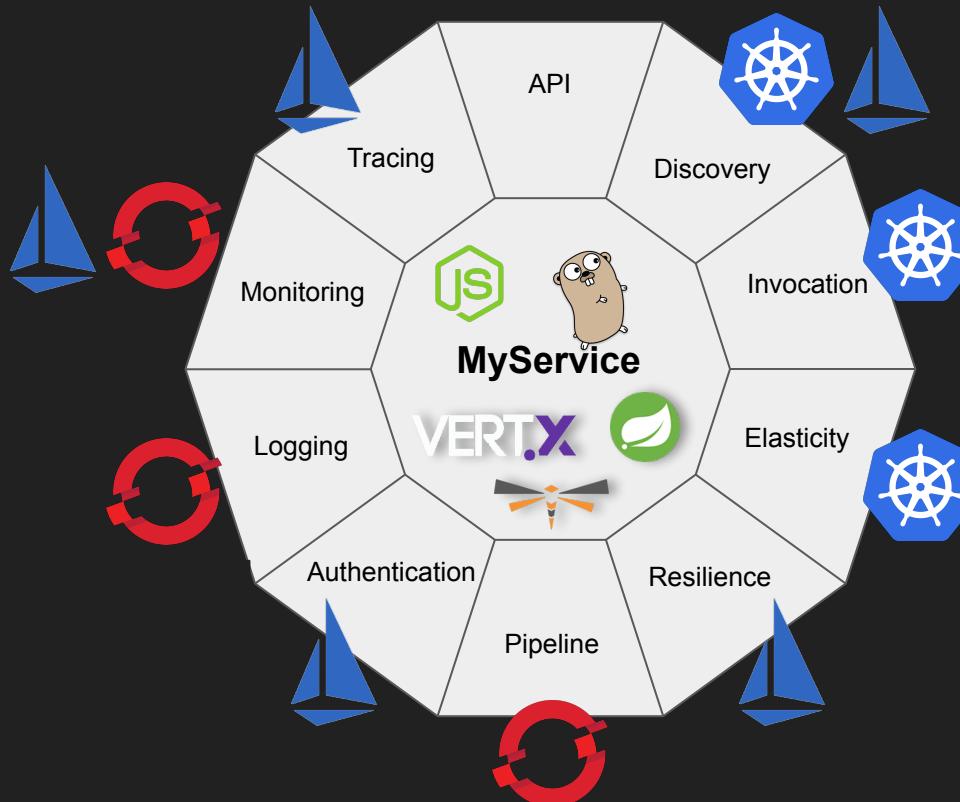
(Kubernetes - Helmsman or ship's pilot)

Service Mesh Defined

A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware

<https://www.infoq.com/2017/04/25/what-is-a-service-mesh-and-why-do-i-need-one/>

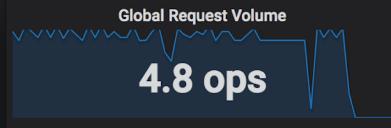
Microservices'ilities + Istio



Observability

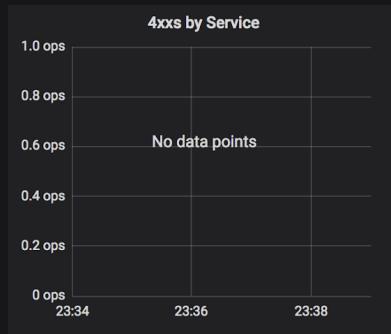
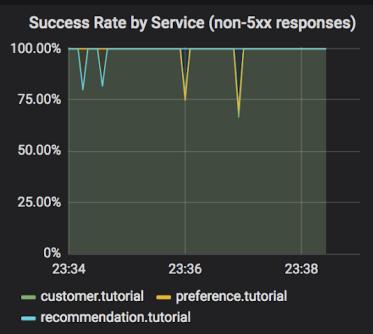
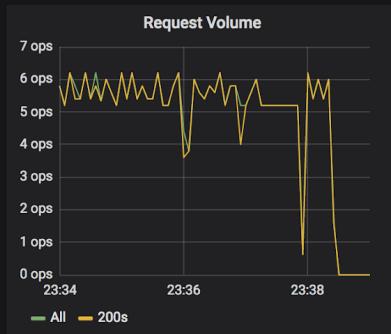


Istio Dashboard ▾



Service Mesh

Service Mesh



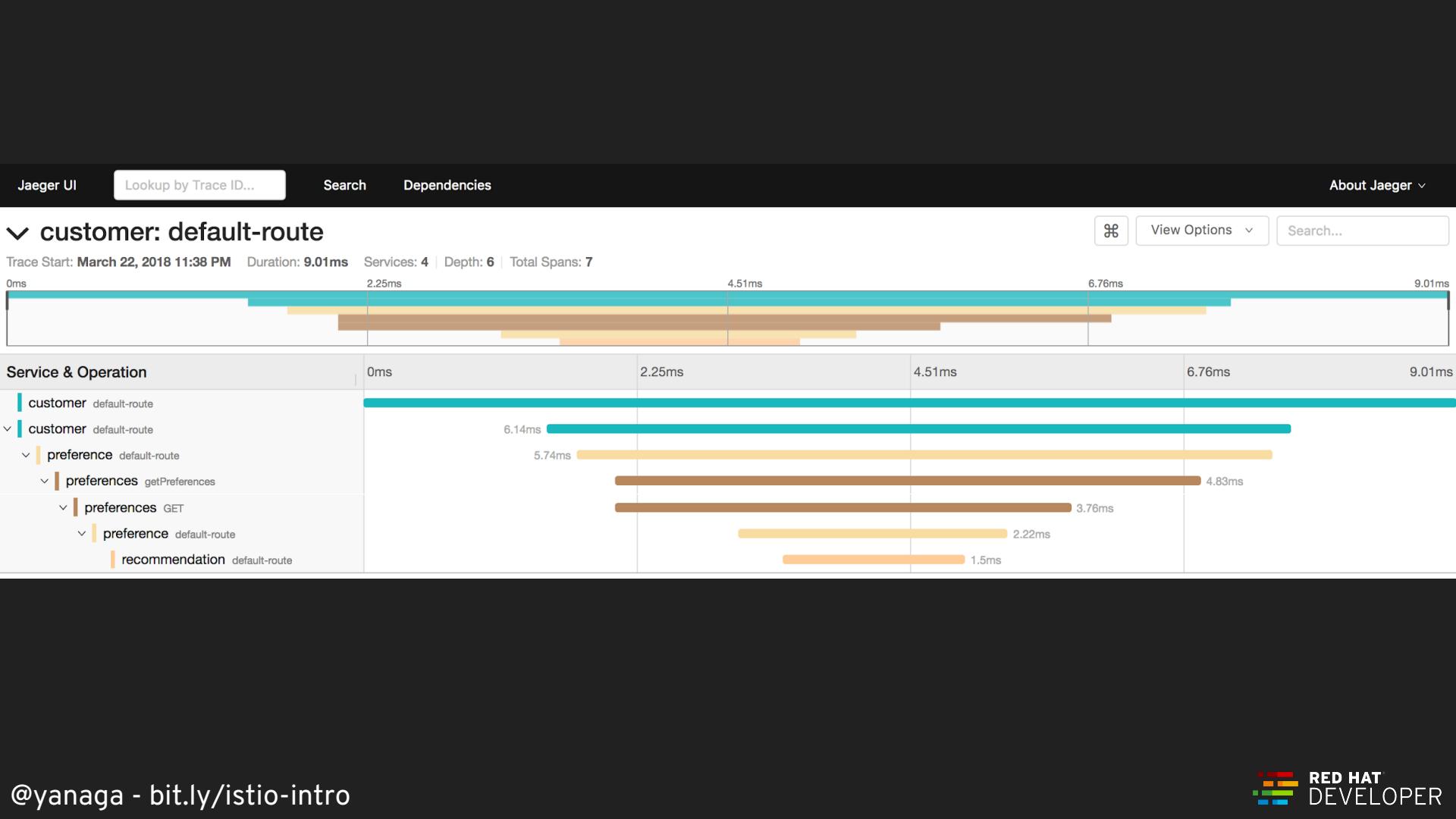
Services

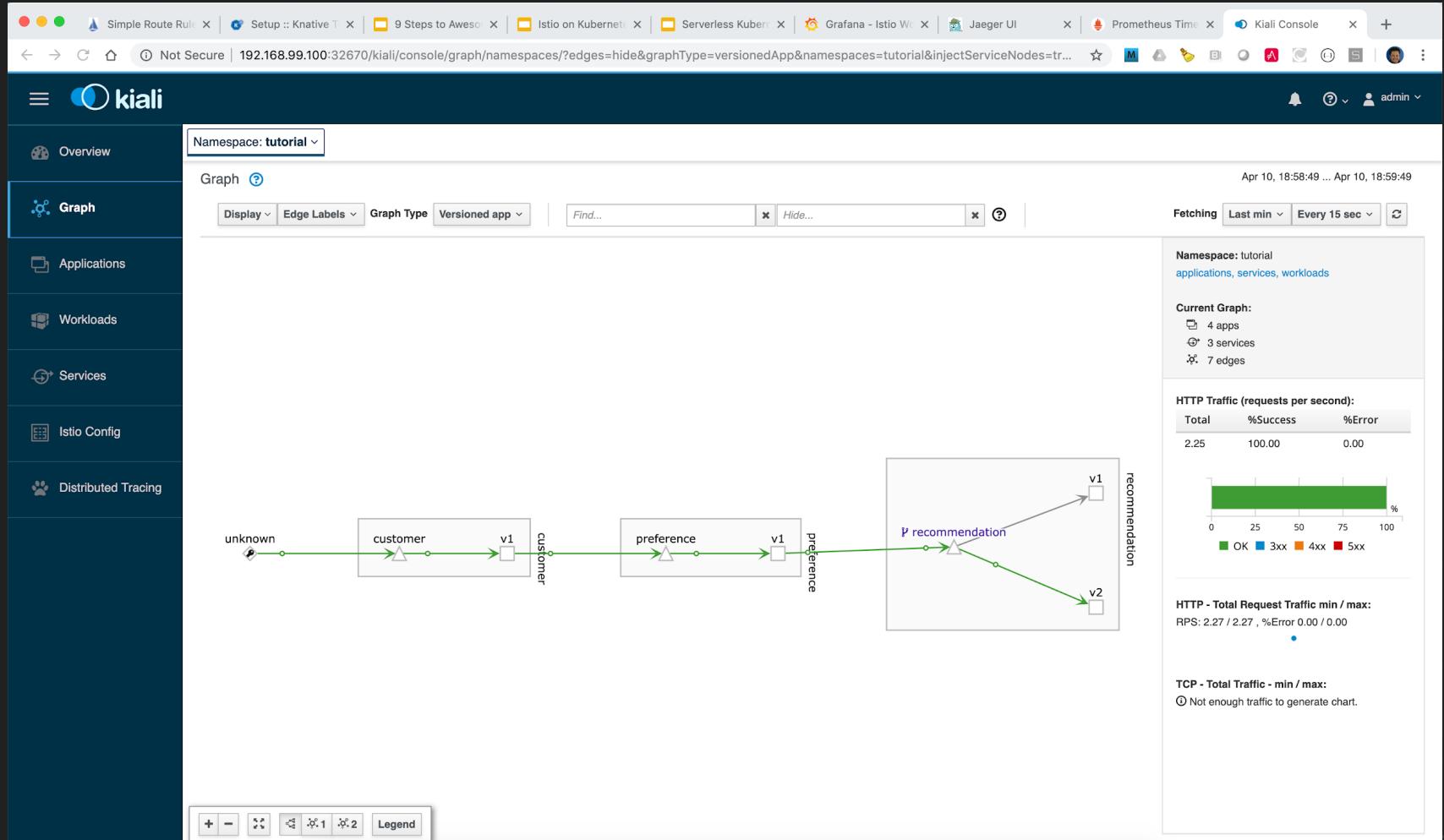
HTTP Services

customer.tutorial.svc.cluster.local

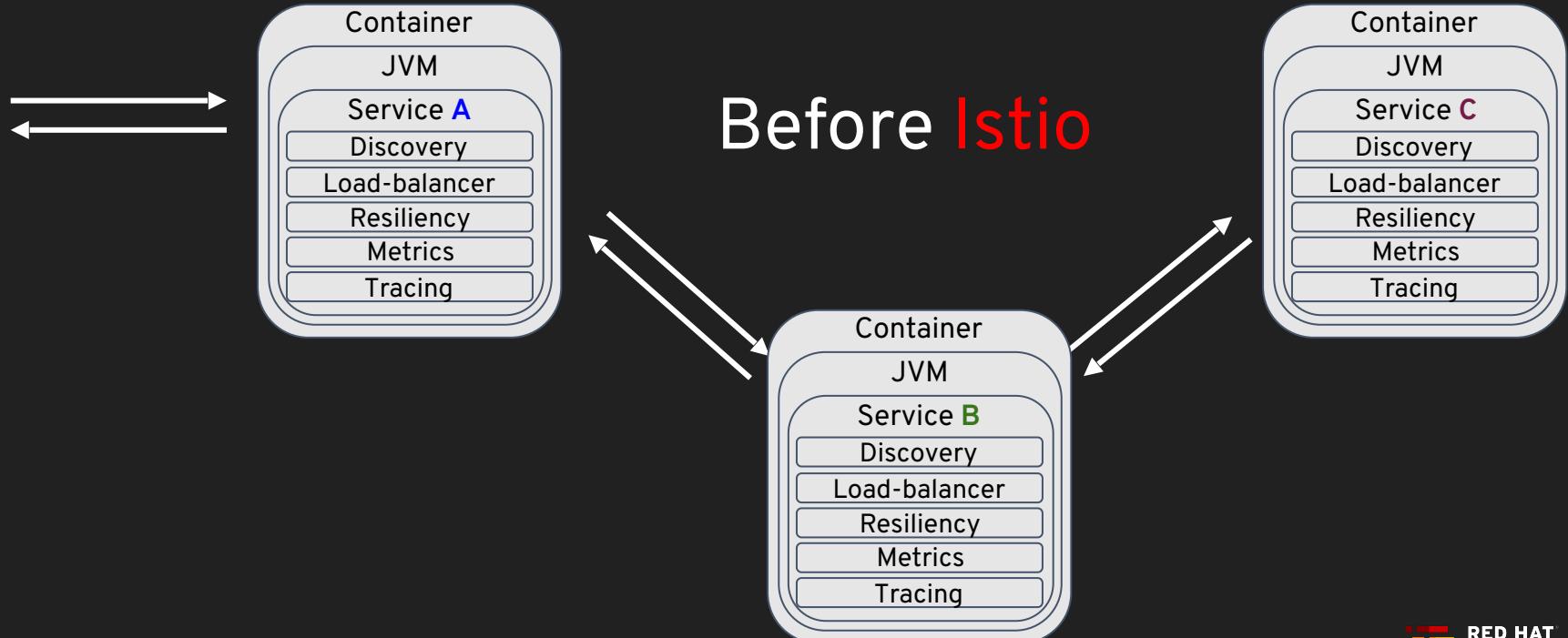
PER

@yan

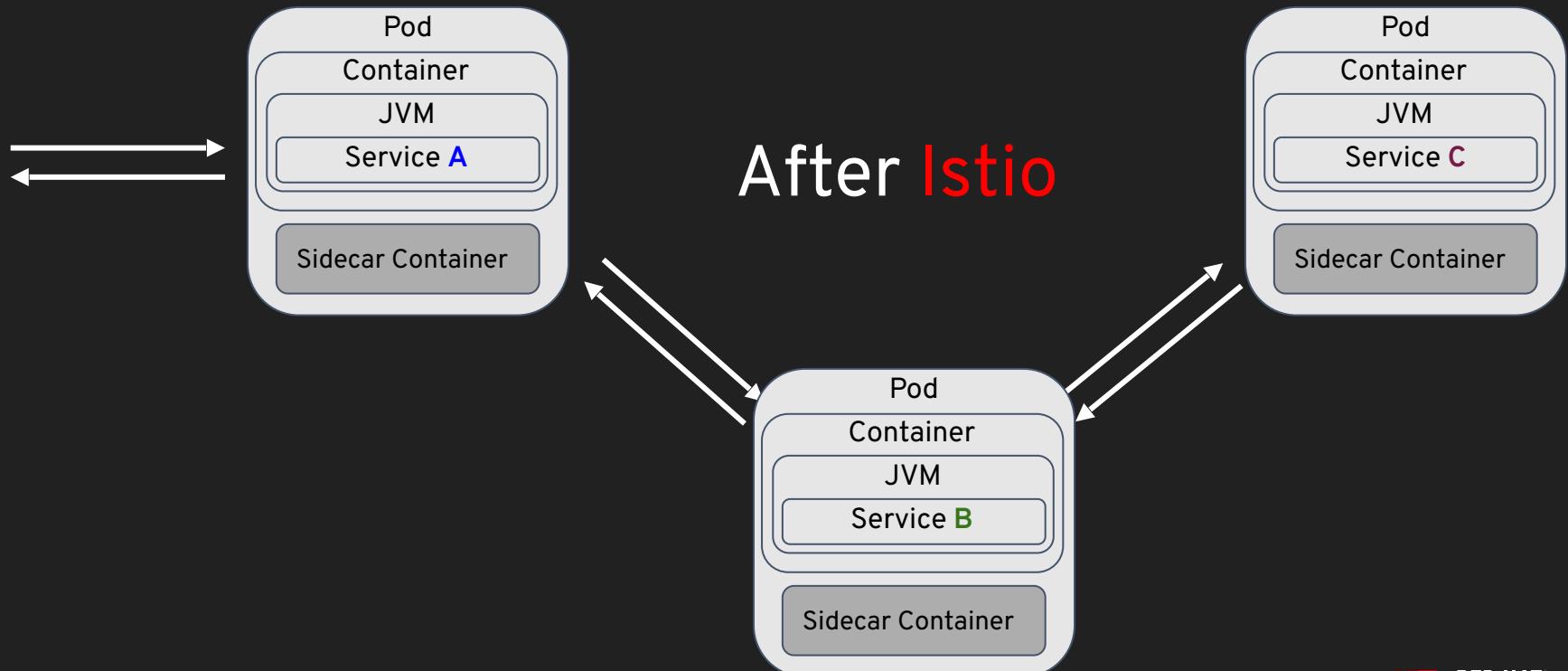




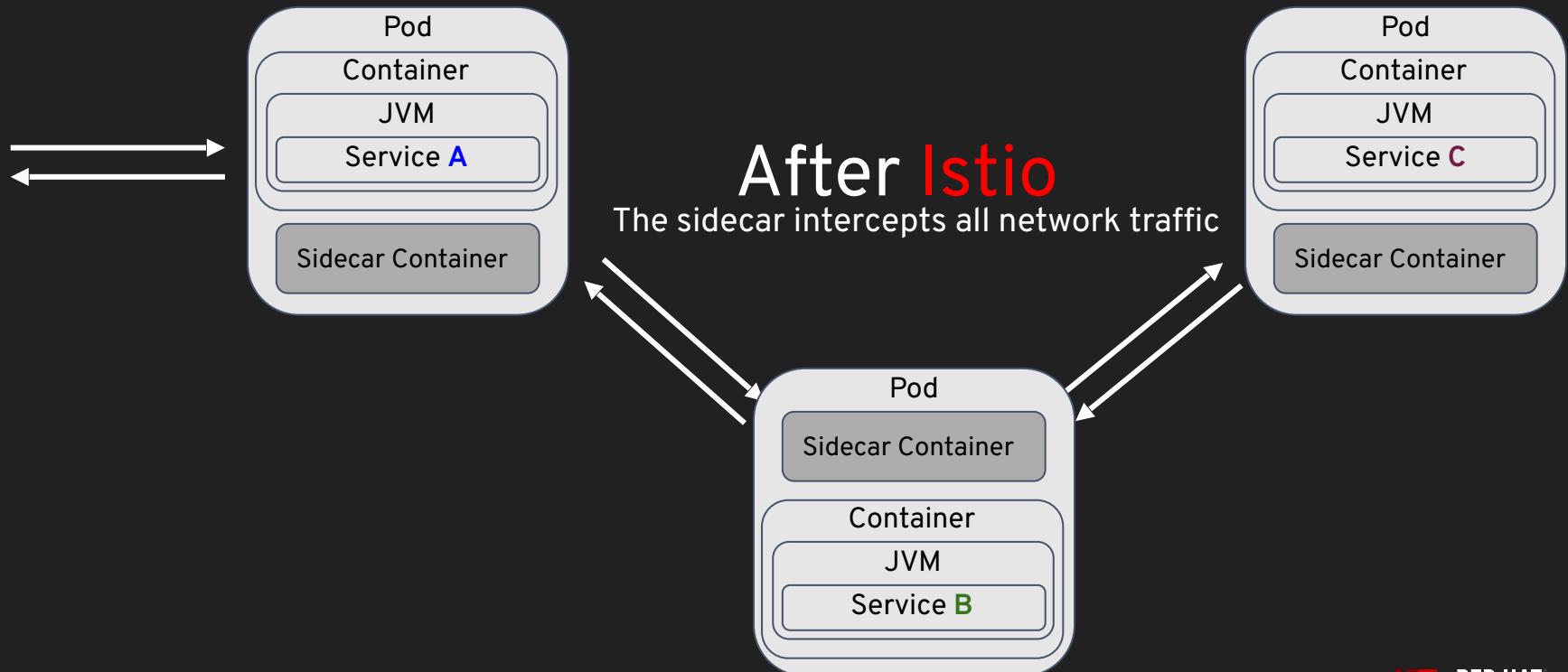
Microservices embedding Capabilities



Microservices externalizing Capabilities



Microservices externalizing Capabilities



Sidecar



<https://www.imz-ural.com/blog/waffles-the-sidecar-dog>

How to add an Istio-Proxy (sidecar)?

```
istioctl kube-inject -f NormalDeployment.yaml
```

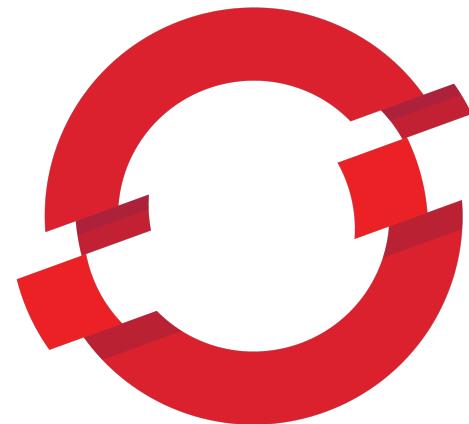
OR

```
kubectl label namespace myspace istio-  
injection=enabled
```

To "see" the sidecar:

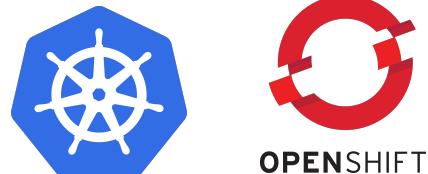
```
kubectl describe deployment customer
```

Better Microservices Platform circa 2018

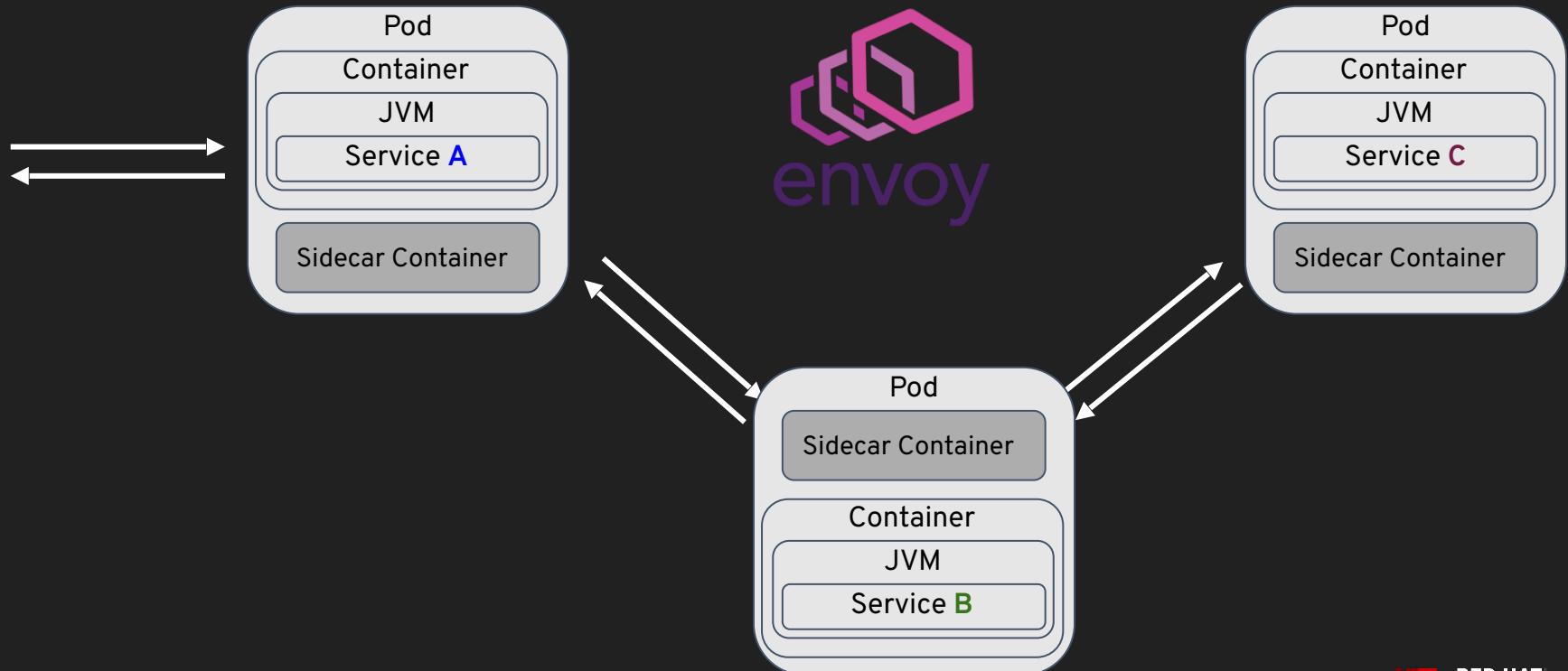


OPENSHIFT

Polyglot Microservices Platform circa 2018



Envoy is the current sidecar

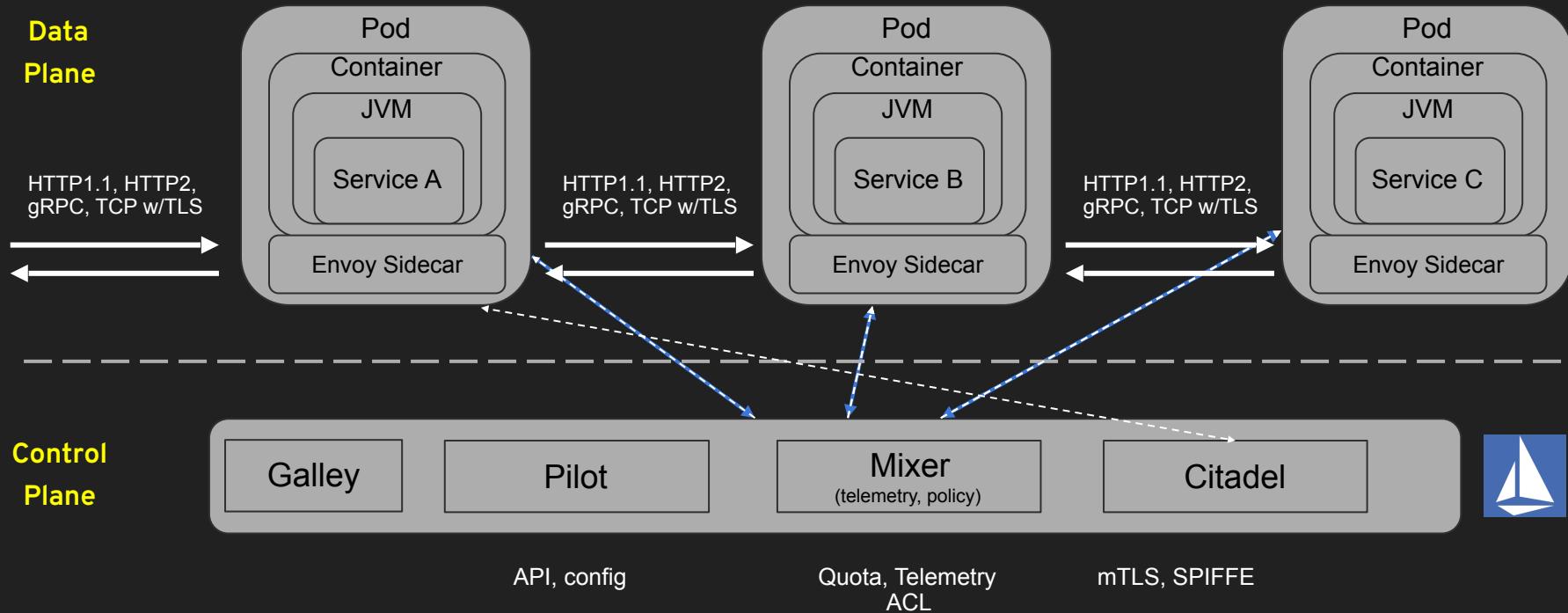


Next Generation Microservices - Service Mesh

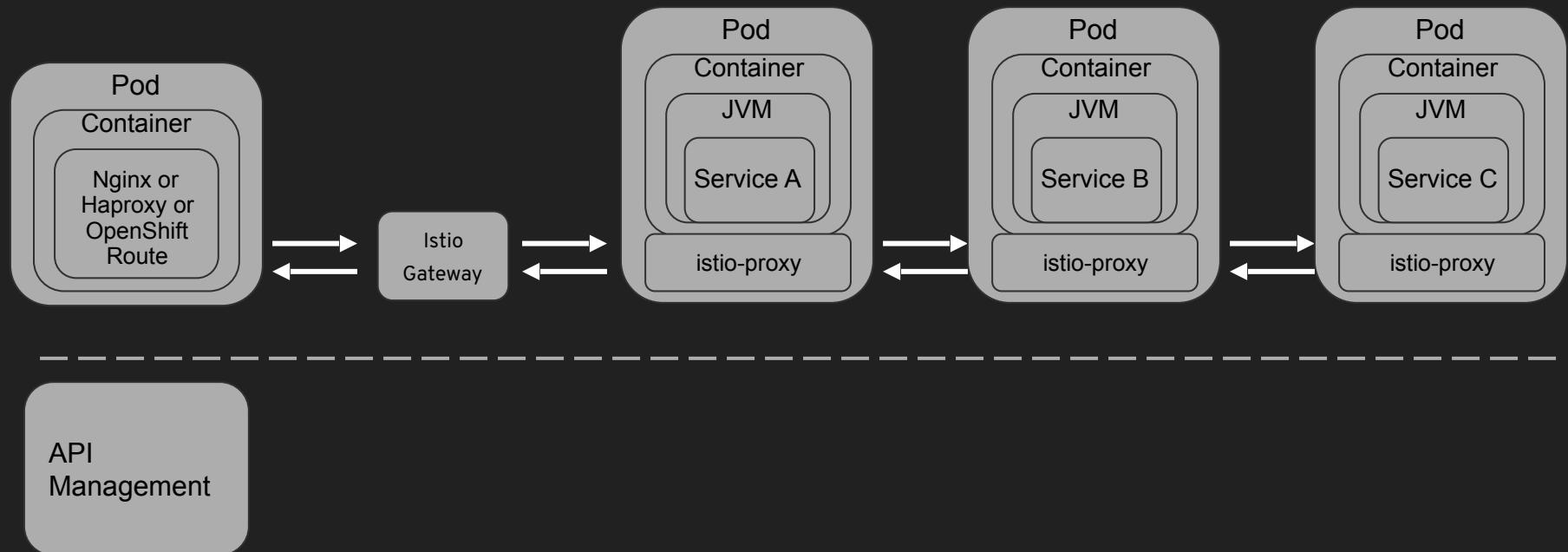
Code Independent (Polyglot)

- Intelligent Routing and Load-Balancing
 - Smarter Canary Releases
 - Dark Launch
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability & Telemetry: Metrics and Tracing
- Security: Encryption & Authorization
- Fleet wide policy enforcement

Istio Data Plane vs Control Plane



API Gateways



kubectl get crds

Adapters.config.istio.io
Apikeys.config.istio.io
Attributemanifests.config.istio.io
Authorizations.config.istio.io
Bypasses.config.istio.io
Checknothings.config.istio.io
Circonuses.config.istio.io
Cloudwatches.config.istio.io
Deniers.config.istio.io
Destinationrules.networking.istio.io
Dogstatsds.config.istio.io
Edges.config.istio.io
Envoyfilters.networking.istio.io
Fluenttds.config.istio.io
Gateways.networking.istio.io
Handlers.config.istio.io
Httpapispecbindings.config.istio.io
Httpapispecs.config.istio.io
Instances.config.istio.io
Kubernetesenvs.config.istio.io
Kuberneteses.config.istio.io
Listcheckers.config.istio.io
Listentries.config.istio.io
Logentries.config.istio.io
Memquotas.config.istio.io
Meshpolicies.authentication.istio.io
Metrics.config.istio.io

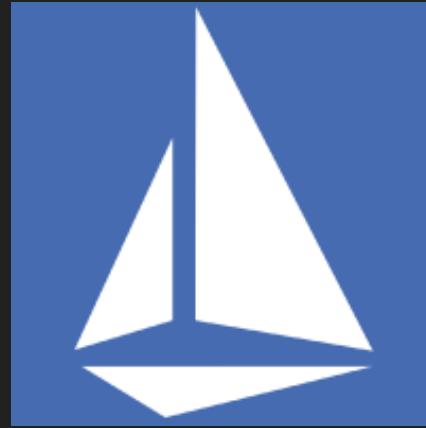
Metrics.config.istio.io
Noops.config.istio.io
Opas.config.istio.io
Policies.authentication.istio.io
Prometheuses.config.istio.io
Quotas.config.istio.io
Quotaspecbindings.config.istio.io
Quotaspecs.config.istio.io
Rbacconfigs.rbac.istio.io
Rbacs.config.istio.io
Redisquotas.config.istio.io
Reportnothings.config.istio.io
Rules.config.istio.io
Servicecontrolreports.config.istio.io
Servicecontrols.config.istio.io
Serviceentries.networking.istio.io
Servicerolebindings.rbac.istio.io
Serviceroles.rbac.istio.io
Signalfxs.config.istio.io
Solarwindses.config.istio.io
Stackdrivers.config.istio.io
Statsds.config.istio.io
Stdios.config.istio.io
Templates.config.istio.io
Tracespans.config.istio.io
Virtualservices.networking.istio.io

CustomResourceDefinitions
of Istio 1.0.x

kubectl api-resources | grep istio

Main Istio Resources (API Objects based on CRDs)

- VirtualService
 - defines the rules that control how requests for a service are routed within an Istio service mesh
 - routing logic, load weighting, chaos injection
- DestinationRule
 - configures the set of policies to be applied to a request after VirtualService routing has occurred
 - load-balancer, outlier, circuit breaker
- ServiceEntry - egress enablement
- Gateway - making a service external to cluster - Ingres
- Policy - enable mTLS
- ServiceRole - roles for RBAC
- ServiceRoleBinding - "users" for the ServiceRole



Exercises

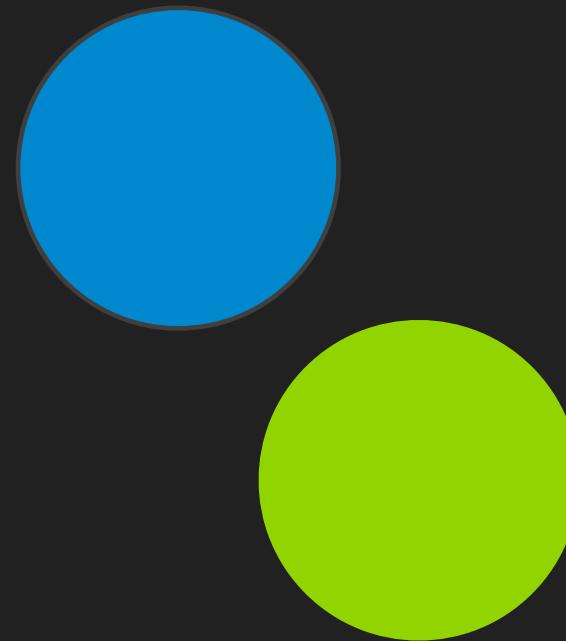
bit.ly/istio-tutorial

Traffic Control

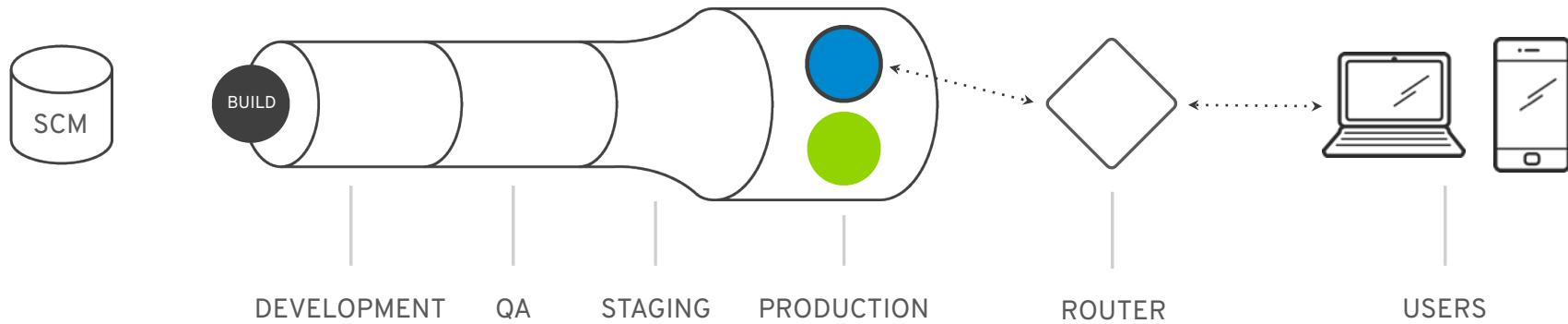
Traffic Control

- Blue/Green part of base Kubernetes/OpenShift
- Percentages not based on pod count - Canary Deployment
- Smart Canaries
- Dark Launch

Blue/Green Deployment

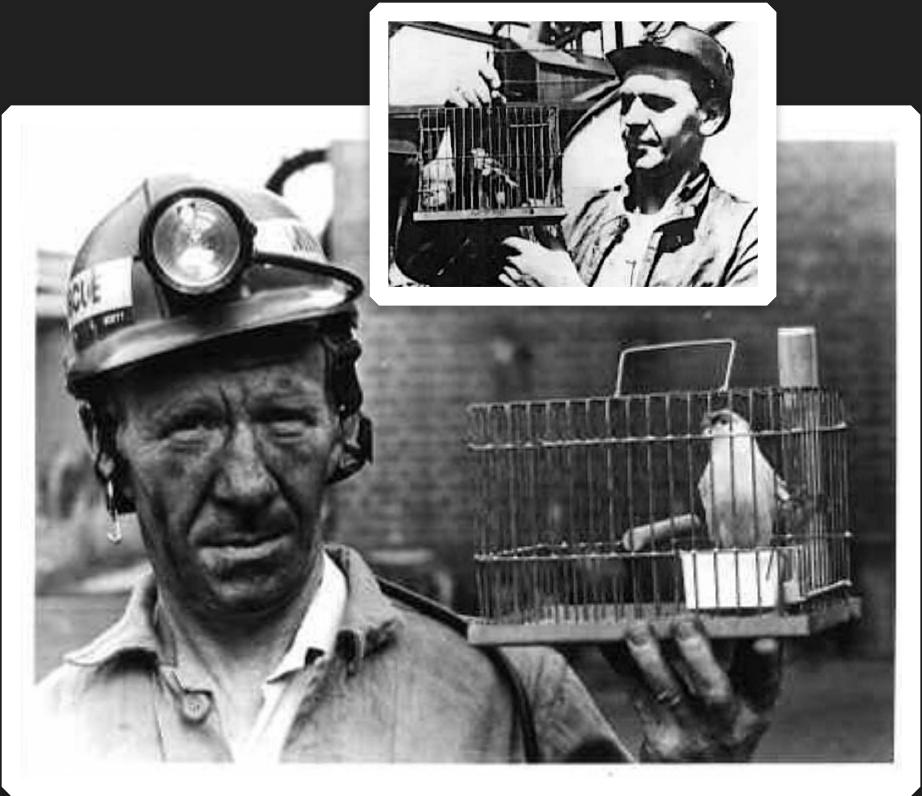


Blue/Green Deployment

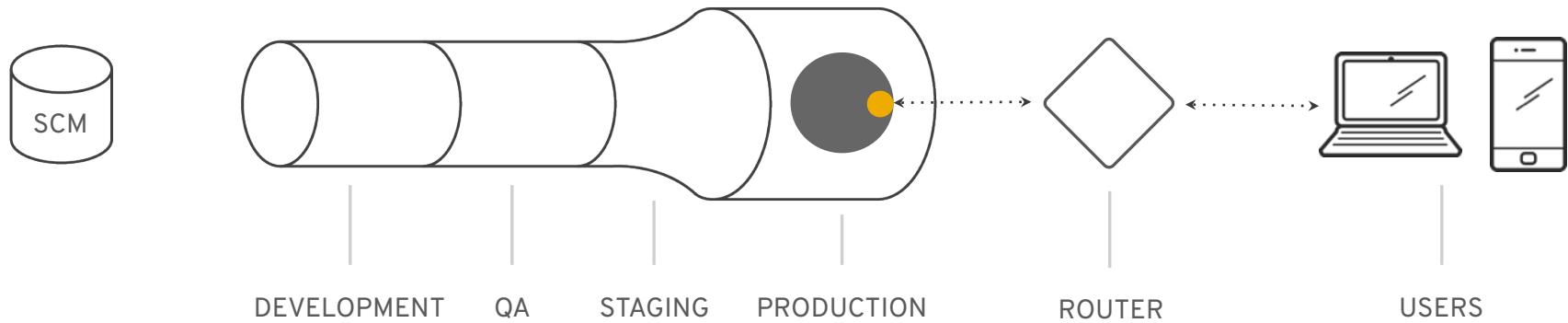


Canary Deployment

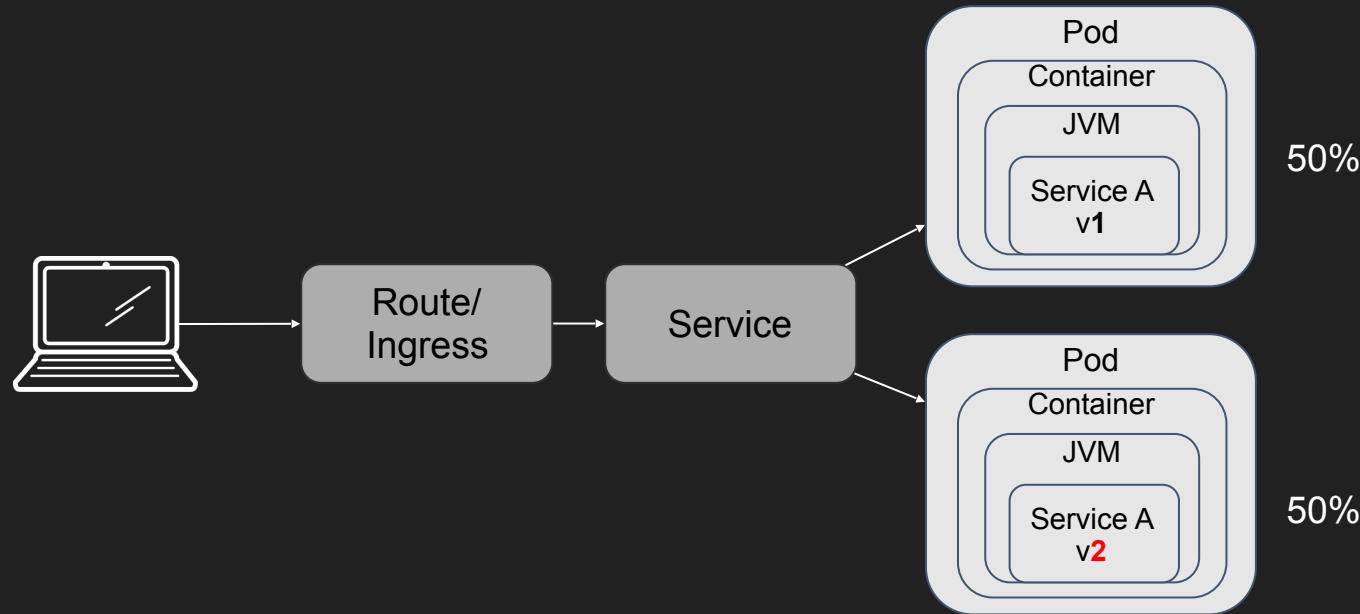




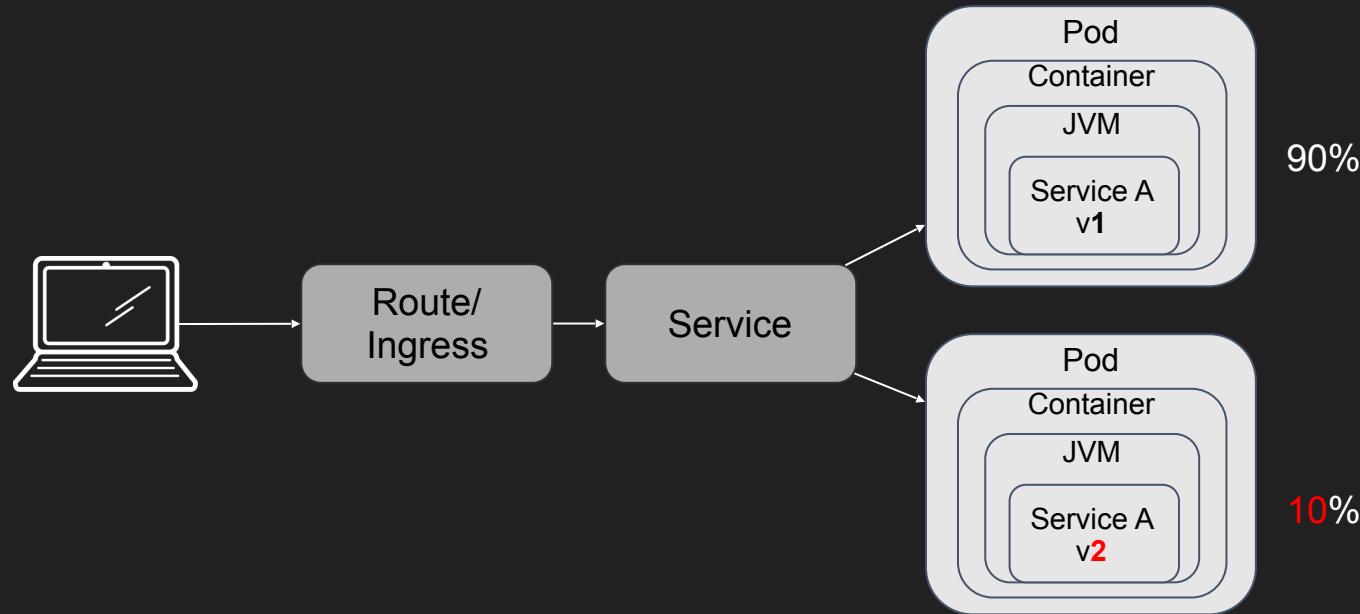
Canary Deployment



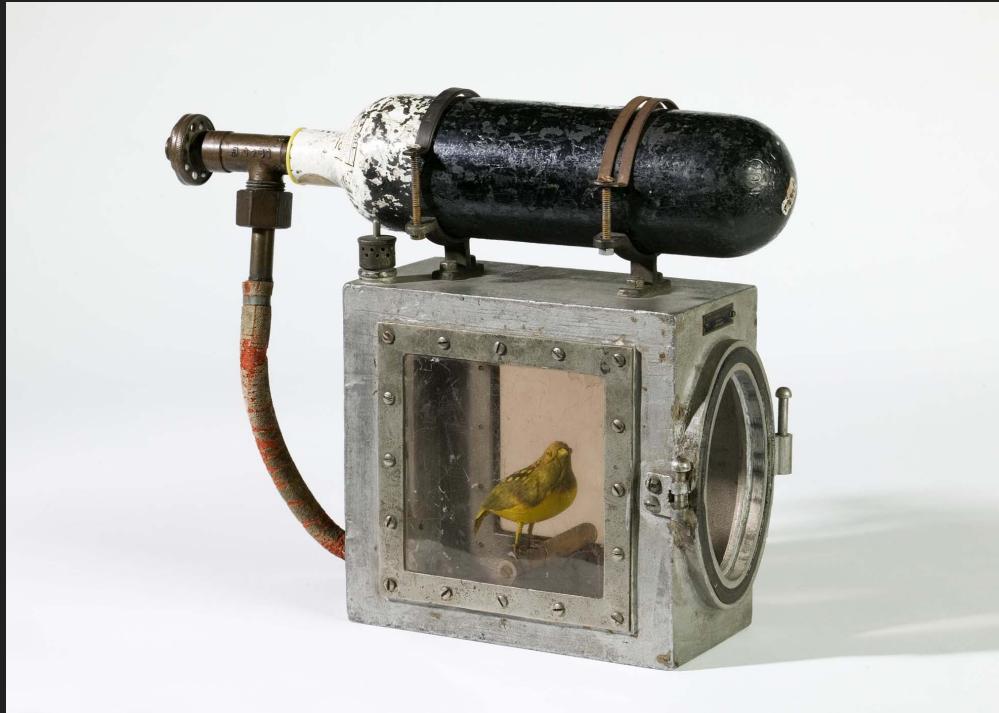
Canaries with Kubernetes



Canaries with Istio



Canary Resuscitator



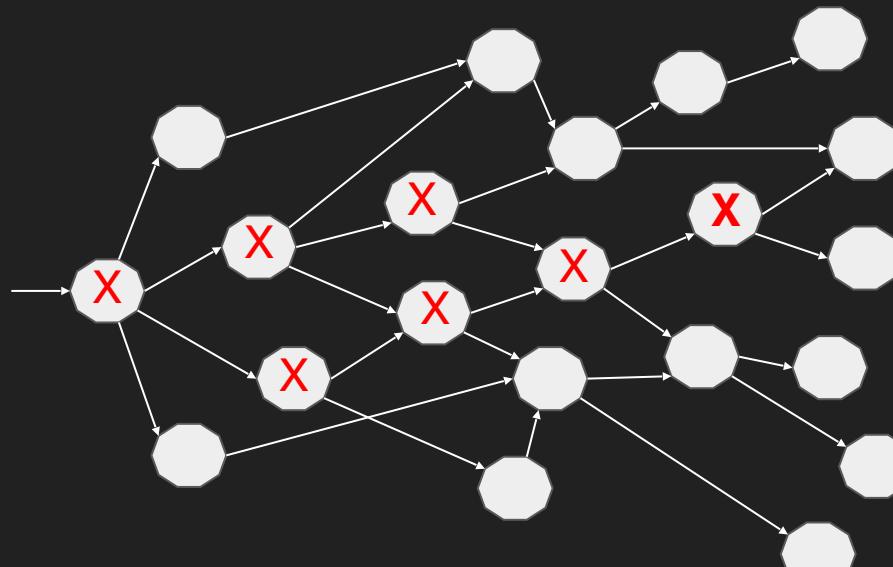
<http://www.openculture.com/2018/05/the-device-invented-to-resuscitate-canaries-in-coal-mines-circa-1896.html>

Thanks to Paolo Antinori!

Service Resiliency

Service Resiliency

- Fail Fast: Latency Circuit Breaker



Chaos Testing

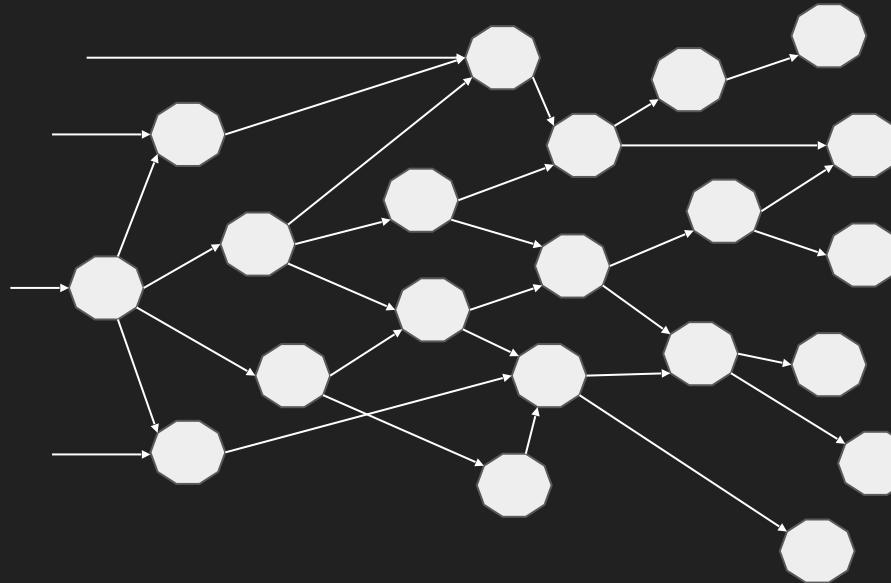
<https://principlesofchaos.org/>



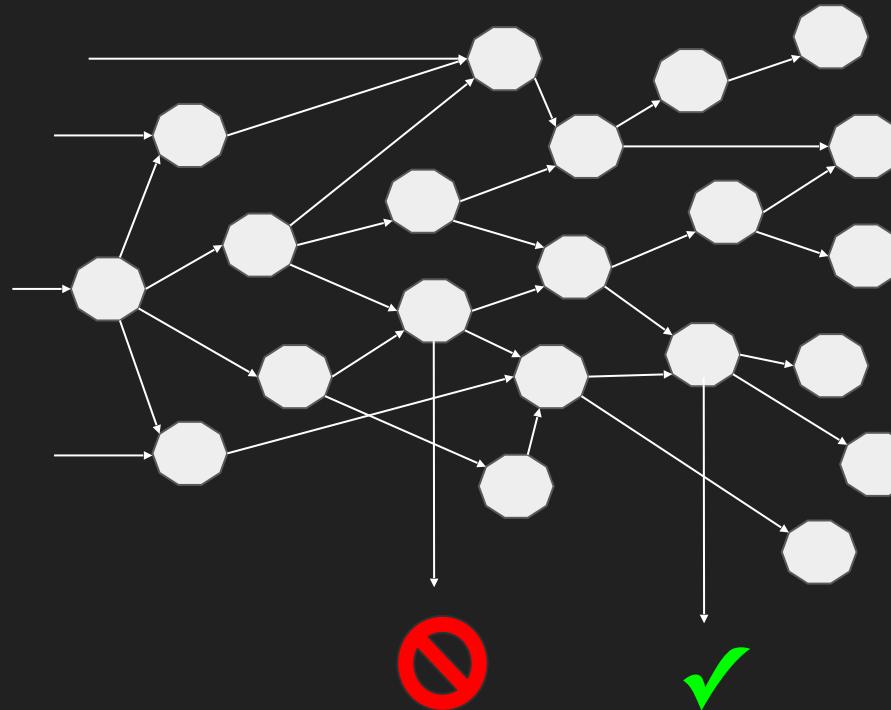
By Netflix - <https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png>, Apache License 2.0, <https://commons.wikimedia.org/w/index.php?curid=63503083>

Egress

Most Communication Inbound & Internal



Outbound/Egress Blocked By Default



Security

Why Security?

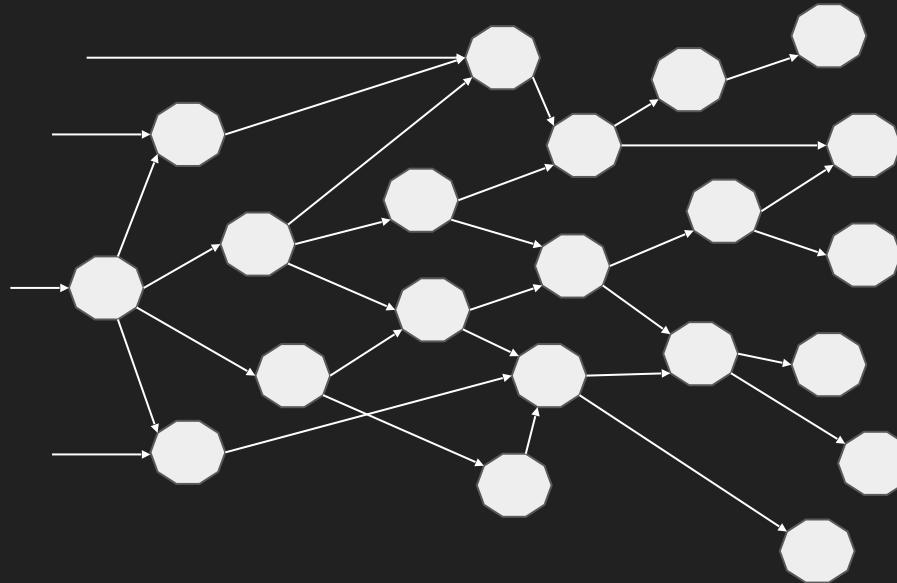
Our Teams:

- A) Customer Success Engineering Team
- B) Human Resources Engineering Team
- C) Marketing Engineering Team
- D) Manufacturing Engineering Team
- E) Big Money Customer Engineering Team

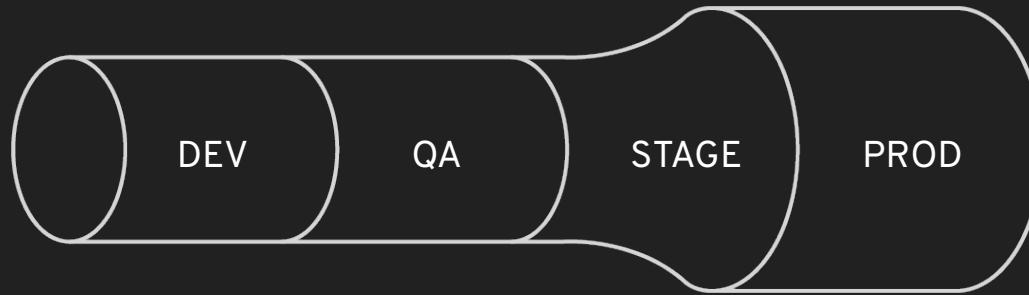
Shared Resources



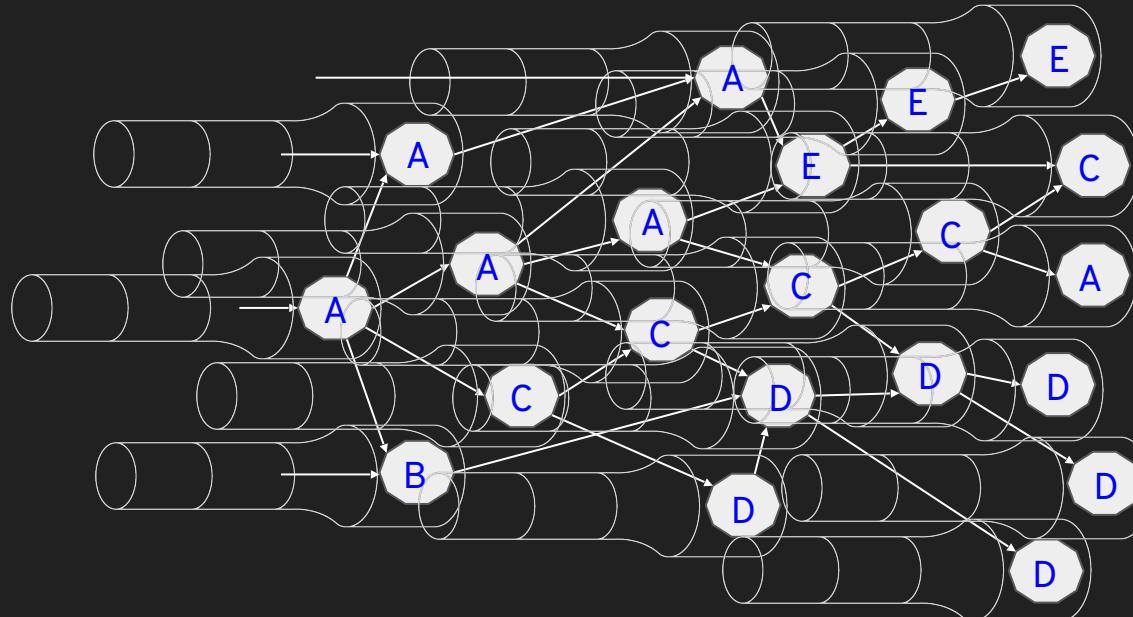
Our Service Mess



Our Pipelines



Our Services, Our Pipelines, Our Teams



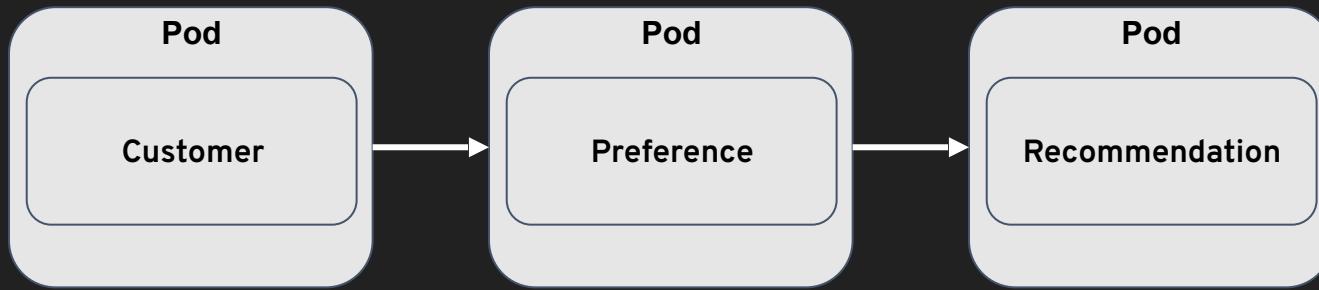
Customer Success Engineering Team A
Human Resources Engineering Team B
Marketing Engineering Team C

Manufacturing Engineering Team D
Big Money Customer Engineering Team E

Istio Security Capabilities

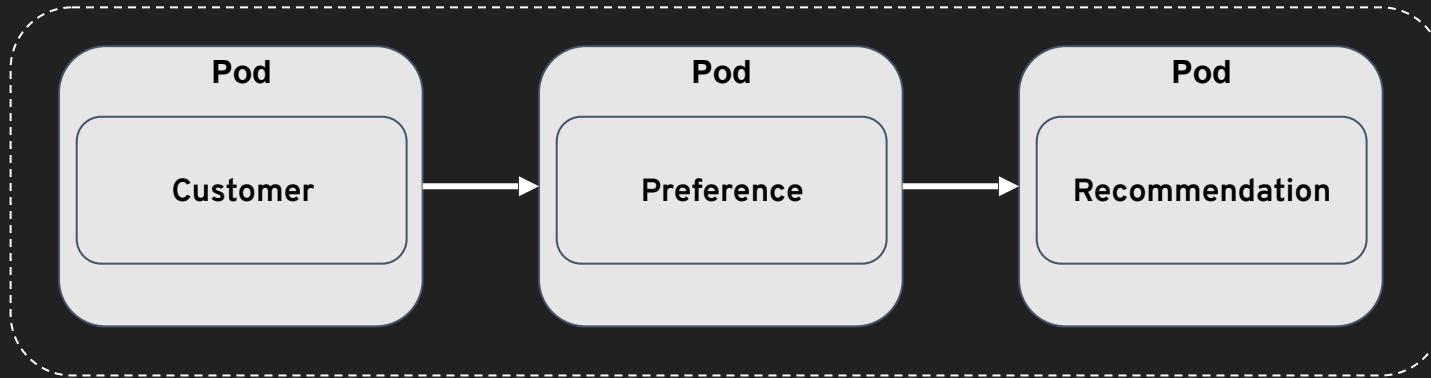
- mTLS - Encryption
- Access Control
- JSON Web Token (JWT) Authentication
- Role-based Access Control (RBAC) Authorization

Why Encryption?



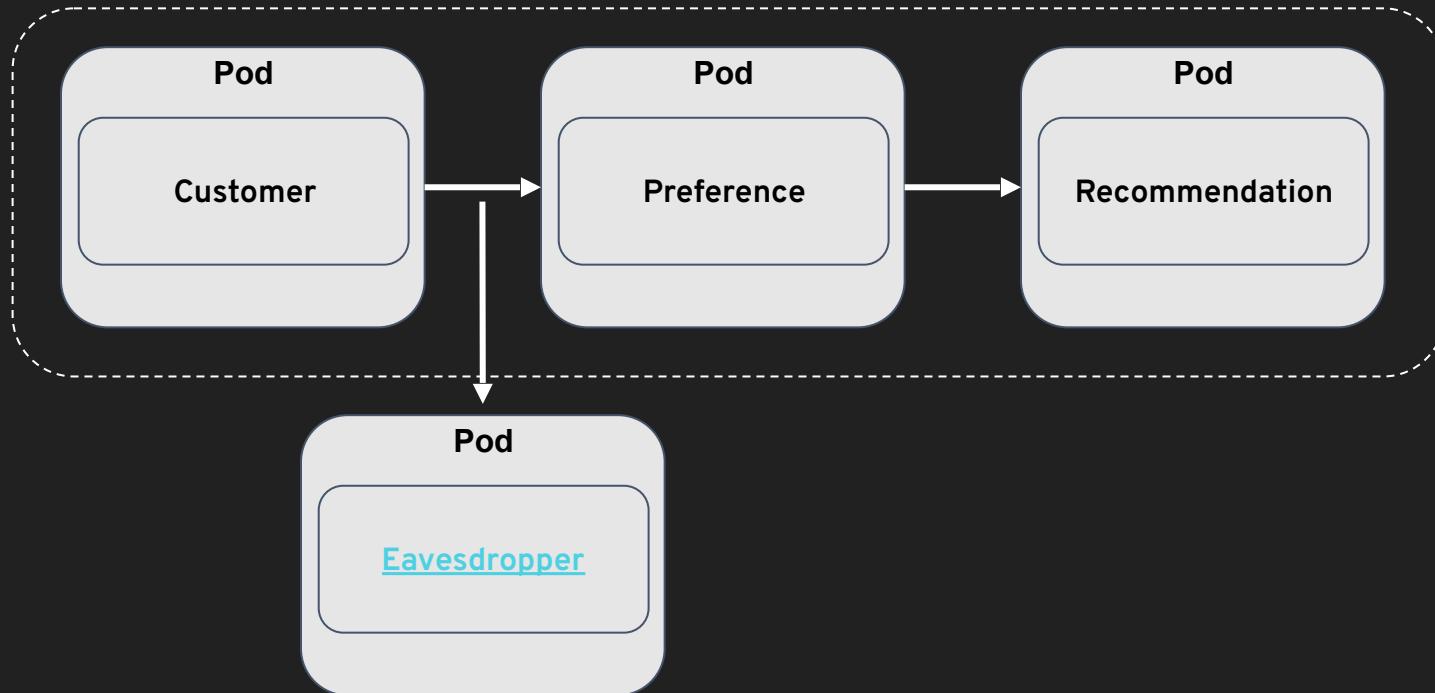
Why Encryption?

Big Money Customer Engineering Team



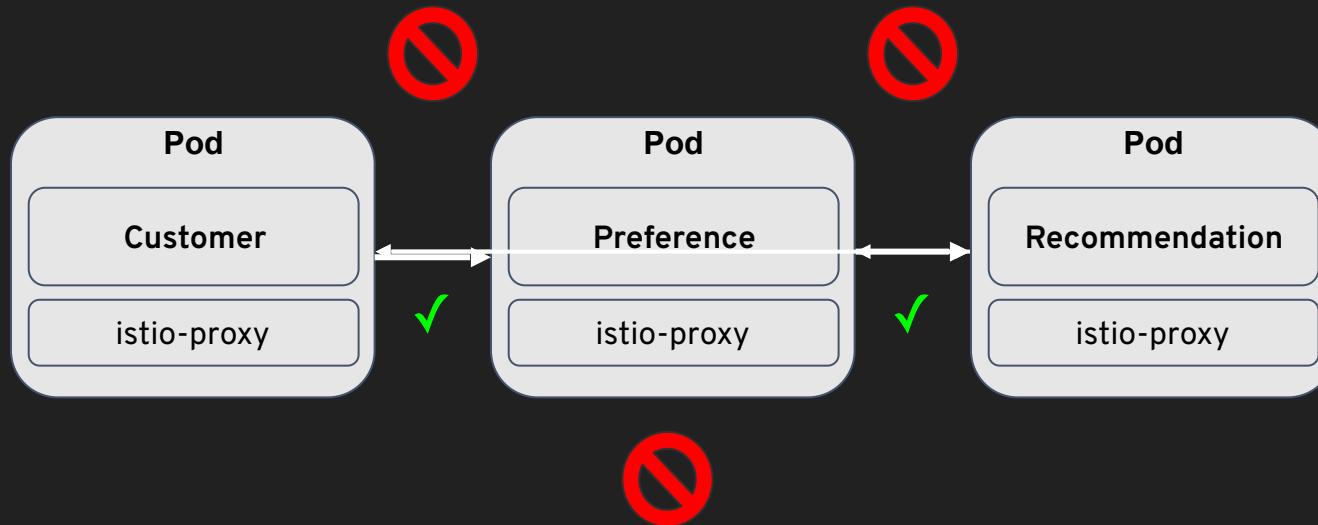
Why Encryption?

Big Money Customer Engineering Team



```
x istio   X Shell   X ...e-v1-5485dc6f49-fspbb: /  
x istio-proxy@preference-v1-5485dc6f49-fspbb: /  
  
p,nop,TS val 9122945 ecr 9122943], length 172: HTTP: HTTP/1.1 200 OK  
E.....@.1.....>..5.*E.....Y.....  
.4....4.HTTP/1.1 200 OK  
content-length: 47  
x-envoy-upstream-service-time: 0  
date: Mon, 24 Dec 2018 17:26:01 GMT  
server: envoy  
  
recommendation v1 from '66b7c9779c-75fp1': 347 ← Eavesdropper  
[]  
  
x @preference-v1-5485dc6f49-fspbb:~  
[jboss@preference-v1-5485dc6f49-fspbb ~]$ curl recommendation:8080  
recommendation v2 from '7cbd9f9c79-nd69g': 341  
[jboss@preference-v1-5485dc6f49-fspbb ~]$ curl recommendation:8080  
recommendation v1 from '66b7c9779c-75fp1': 347  
[jboss@preference-v1-5485dc6f49-fspbb ~]$ []  
~/minishift_1.27.0/istio-tutorial $ []
```

Access Control



JWT Issuer

The screenshot shows the Keycloak administration interface. On the left, a sidebar menu lists various realm management options: Add realm, Realm Settings (selected), Clients, Client Scopes, Roles, Identity Providers, User Federation, Authentication, Manage, Groups, and Users. The main content area is titled "Master" and displays the "General" tab of the realm settings. The configuration fields include:

- Name: master
- Display name: Keycloak
- HTML Display name: <div class="kc-logo-text">Keycloak</div>
- Enabled: ON
- User-Managed Access: OFF
- Endpoints: OpenID Endpoint Configuration

At the bottom right of the form are "Save" and "Cancel" buttons.

1.0 Changes

- RouteRule -> VirtualService
- DestinationPolicy -> DestinationRule
- EgressRule -> ServiceEntry
- Ingress -> Gateway

The End

(but Serverless is coming)