

Istio on Kubernetes: Enter the Service Mesh

✉️ benevides@redhat.com

🐦 [@rafabene](https://twitter.com/rafabene)

Link → <http://bit.ly/istio-kubernetes>

Rafael Benevides



Director of Developer Experience at Red Hat

 benevides@redhat.com
 @rafabene

Java Certifications:

SCJA / SCJP / SCWCD / SCBCD / SCEA

JBoss Certifications:

JBCD / JBCAA

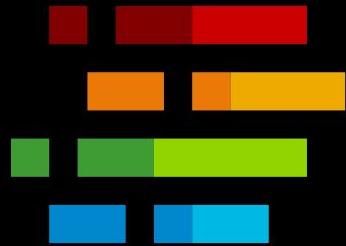
Red Hat Certifications:

OpenShift / Containers / Ansible

Other Certifications:

SAP Netweaver / ITIL / IBM Software Quality



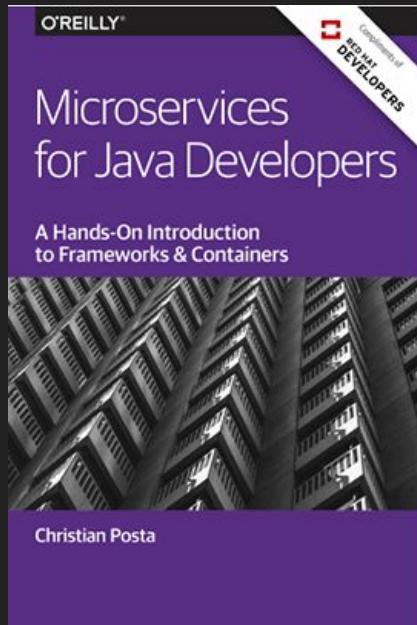


RED HAT® DEVELOPER

Get software and know-how.
Get started with Red Hat technologies.

Join at **developers.redhat.com**.

bit.ly/javamicroservicesbook



Free eBooks from developers.redhat.com

Microservices Introductory Materials

Demo: bit.ly/msa-instructions

Slides: bit.ly/microservicesdeepdive

Video Training: bit.ly/microservicesvideo

[Kubernetes for Java Developers](#)

Advanced Materials

bit.ly/istio-tutorial

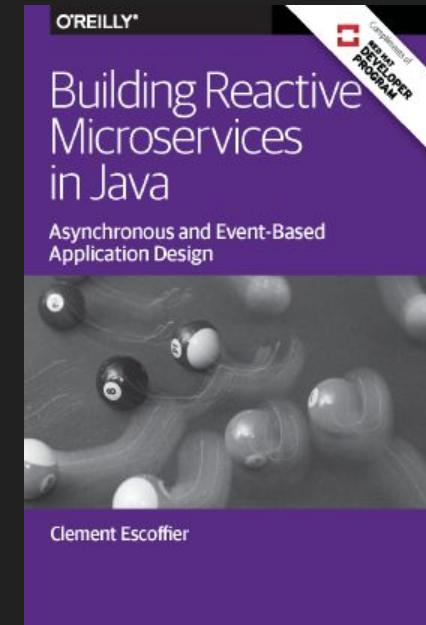
learn.openshift.com/servicemesh

bit.ly/faas-tutorial

learn.openshift.com/serverless

<http://bit.ly/istio-kubernetes>

bit.ly/reactivemicroservicesbook



O'REILLY®

Migrating to Microservice Databases

From Relational Monolith
to Distributed Data



Edson Yanaga

Compliments of
**RED HAT
DEVELOPERS**

bit.ly/mono2microdb

O'REILLY®



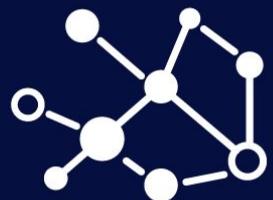
Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant Cloud-Native Applications



Christian Posta & Burr Sutter

bit.ly/istio-book

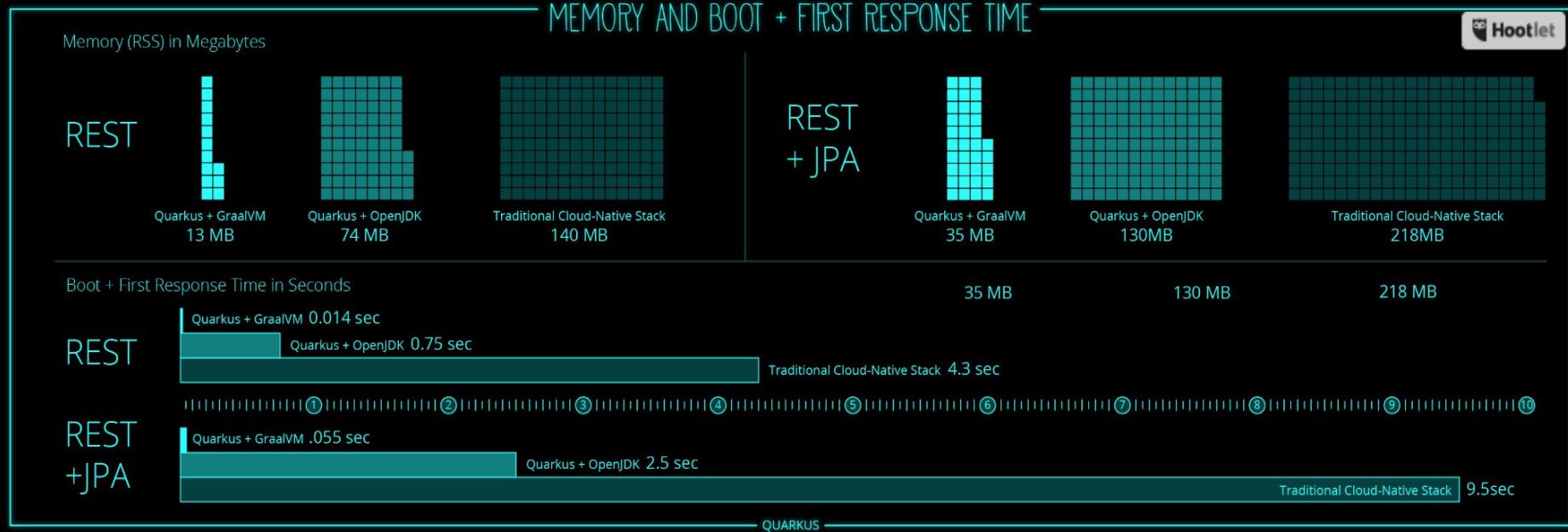


MicroProfile

microprofile.io

Container First

<https://quarkus.io/>



Quarkus tailors your application for GraalVM and HotSpot. Amazingly fast boot time, incredibly low RSS memory (not just heap size!) offering near instant scale up and high density memory utilization in container orchestration platforms like Kubernetes. We use a technique we call compile time boot. [Learn more](#).

```
$ ./my-native-java-rest-app  
Quarkus started in 0.008s
```

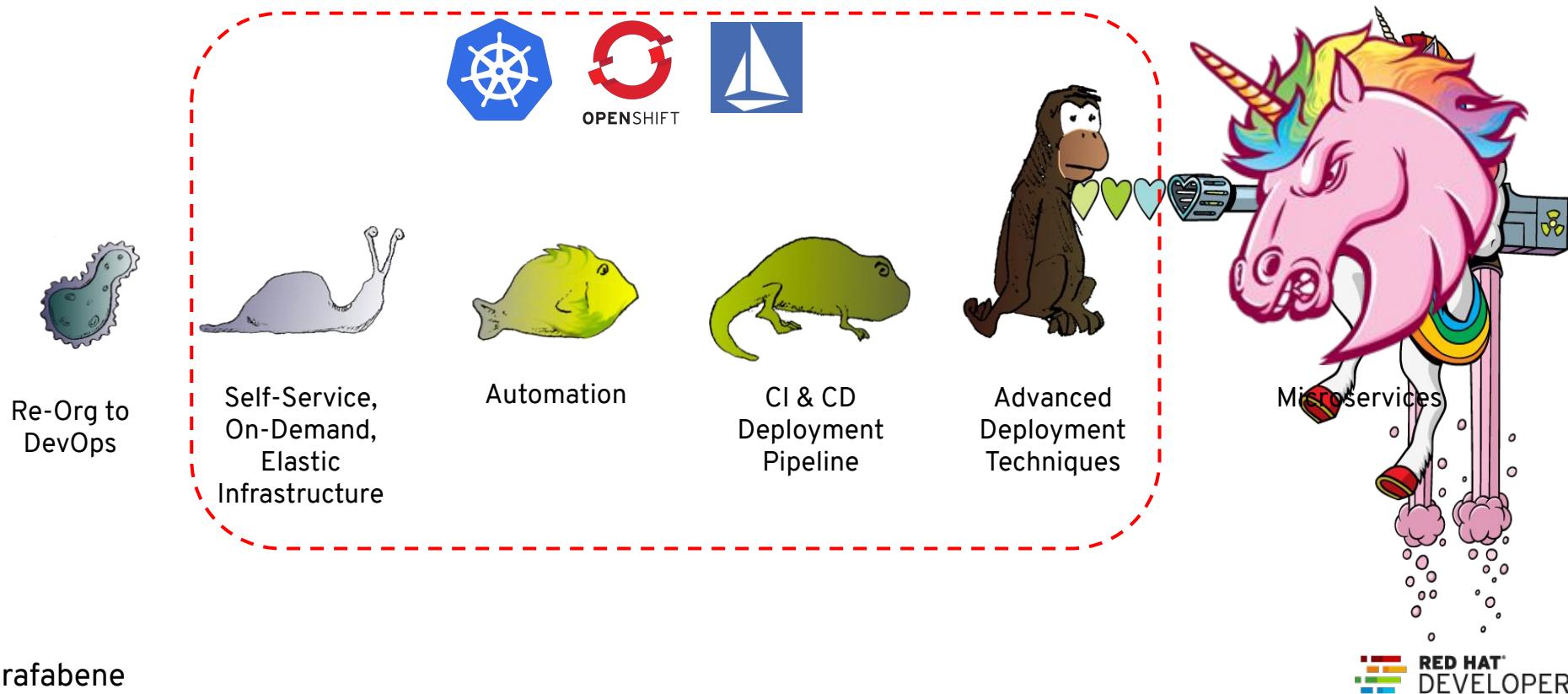
Raffle Rules (applicable in the real)

1. Follow: @rhdevelopers 
2. With selfie of the booth
3. With hashtag #REDHATnoTDC

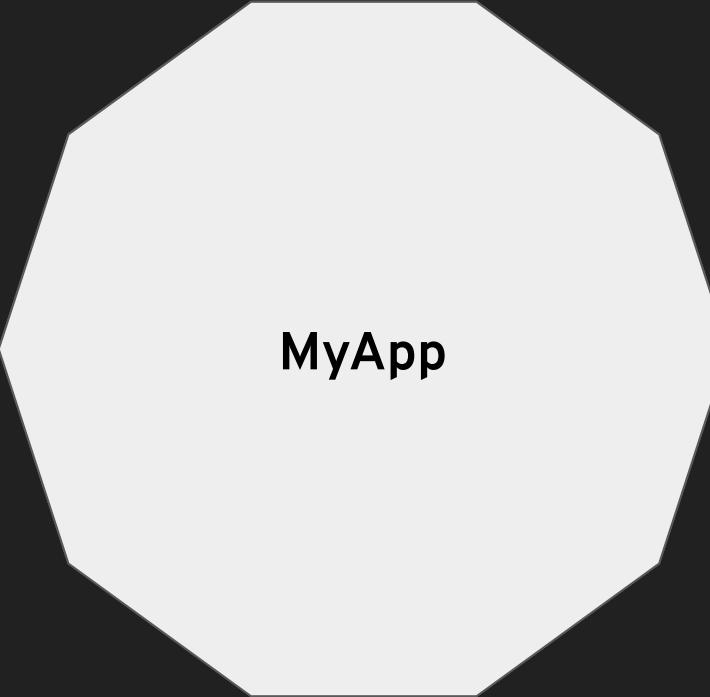
Raffle Rules (applicable in the real)

1. Follow: **@rafabene** 
2. With picture of the session
3. Mention **@rafabene**
4. With hashtag **#VDBUH2019**

Your Journey to Awesomeness

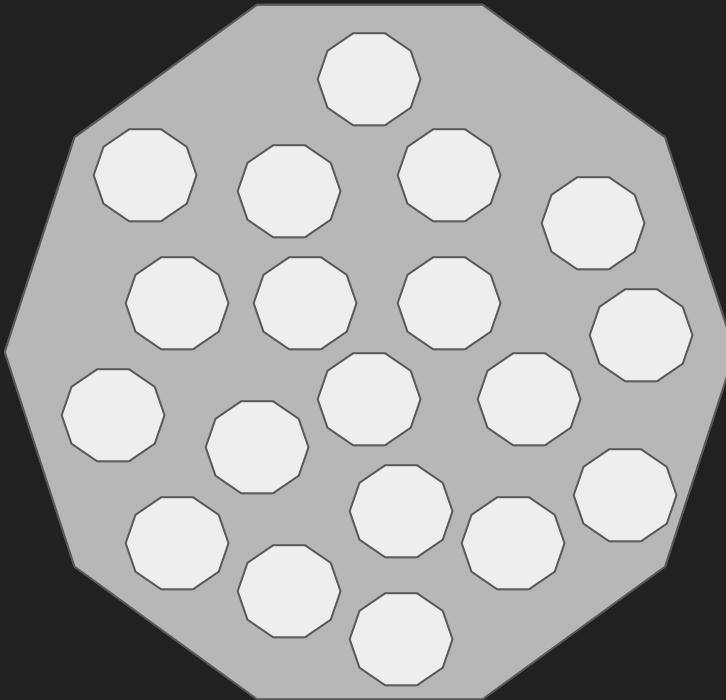


Monolith

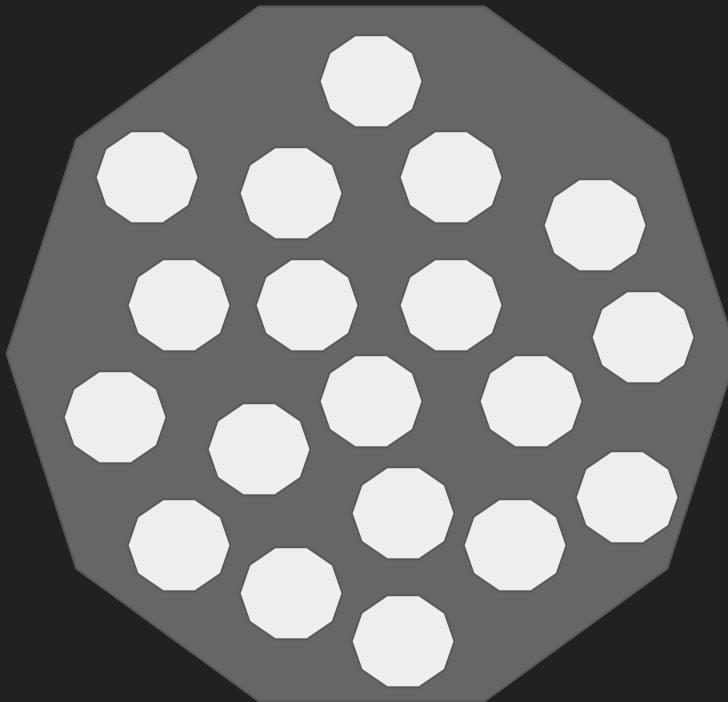


MyApp

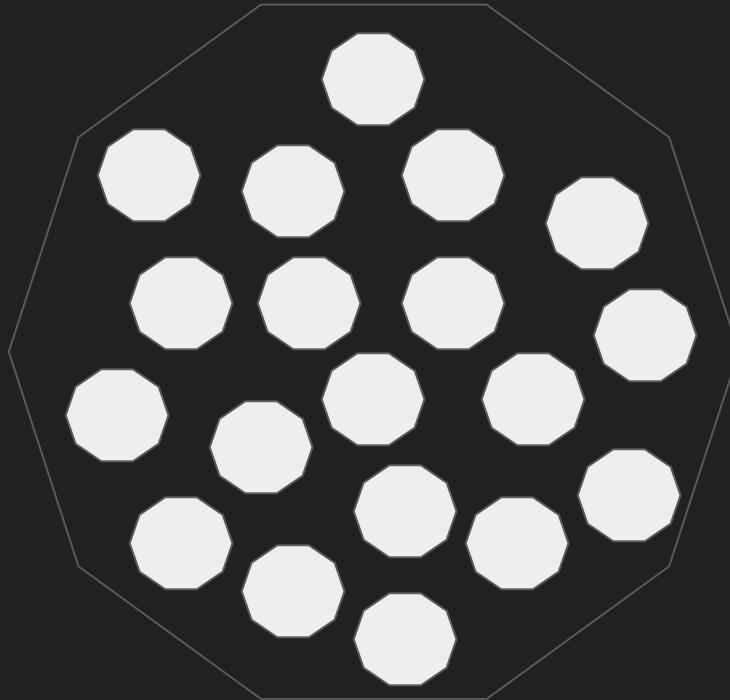
Modules



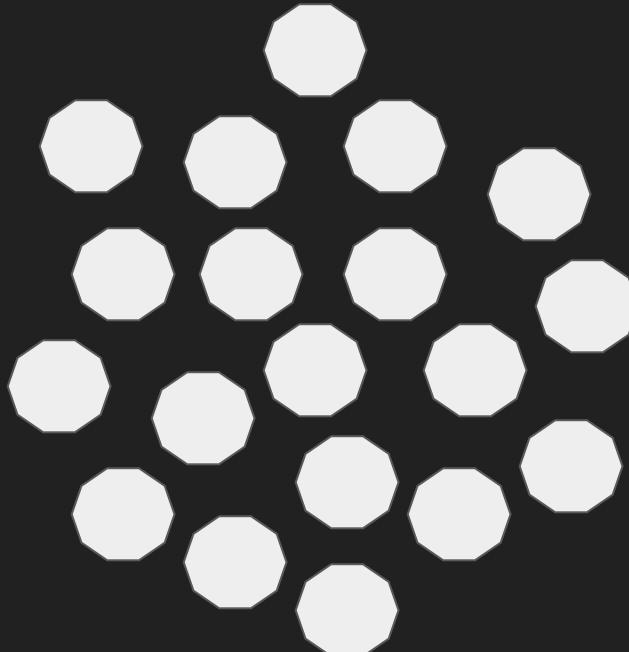
Microservices



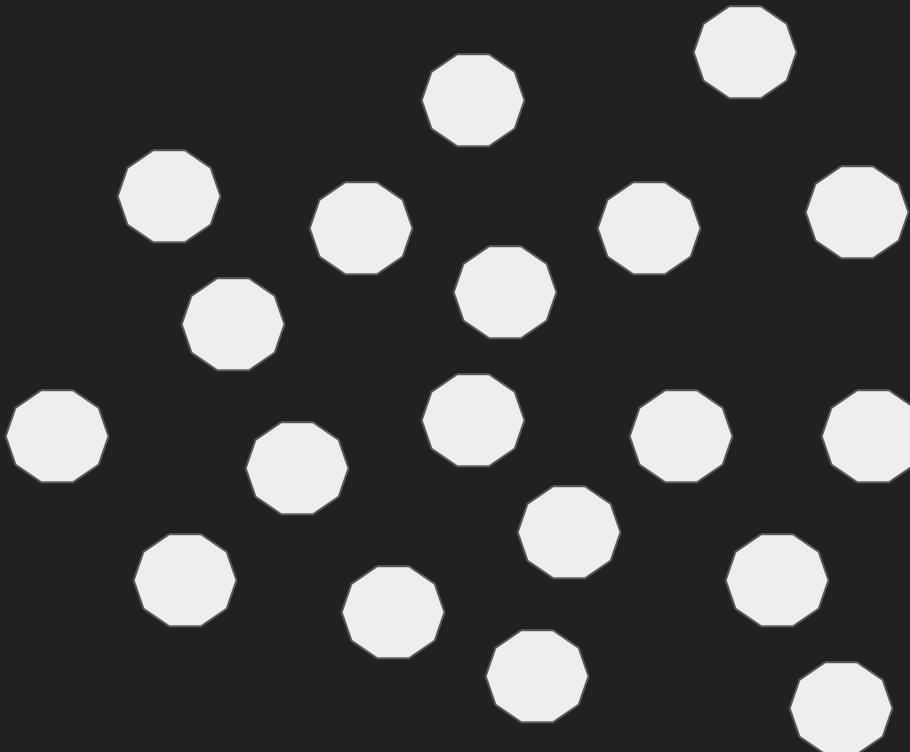
Microservices



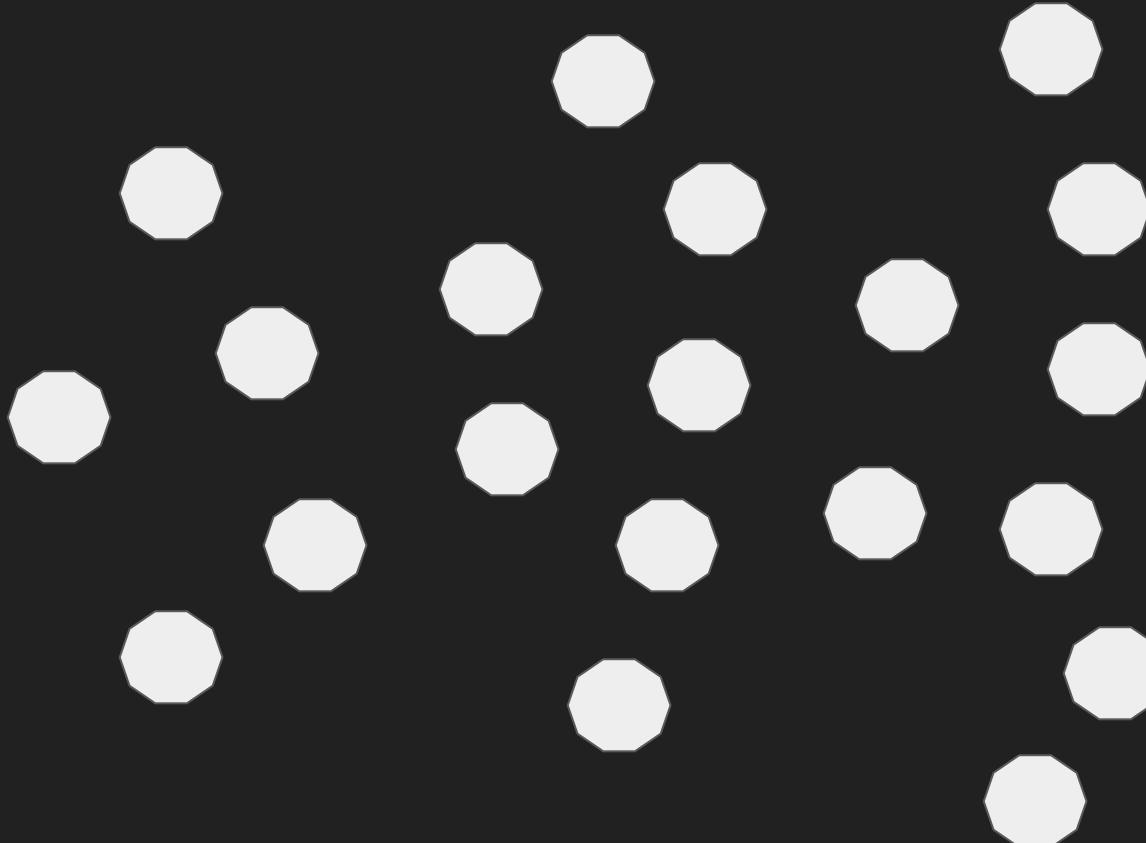
Microservices



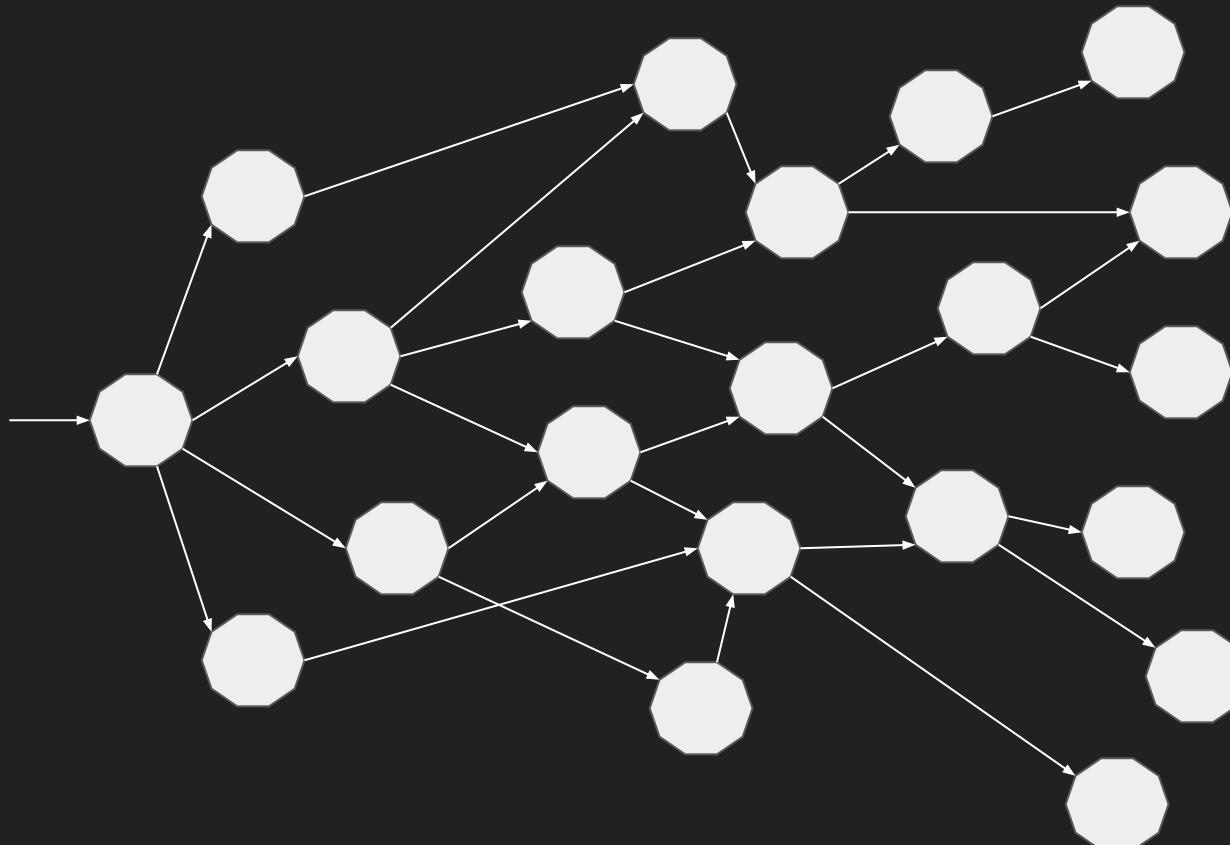
Microservices



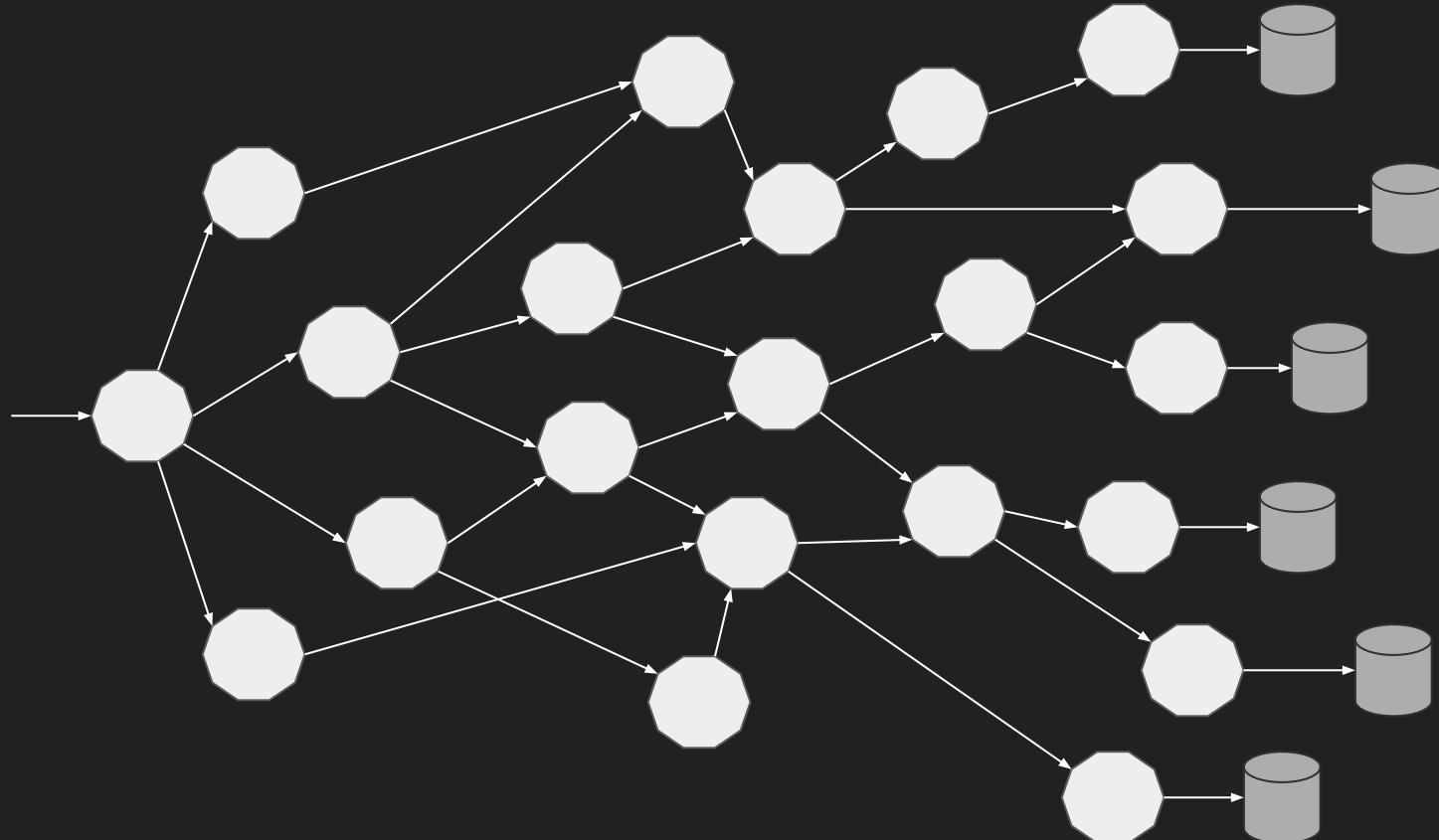
Microservices



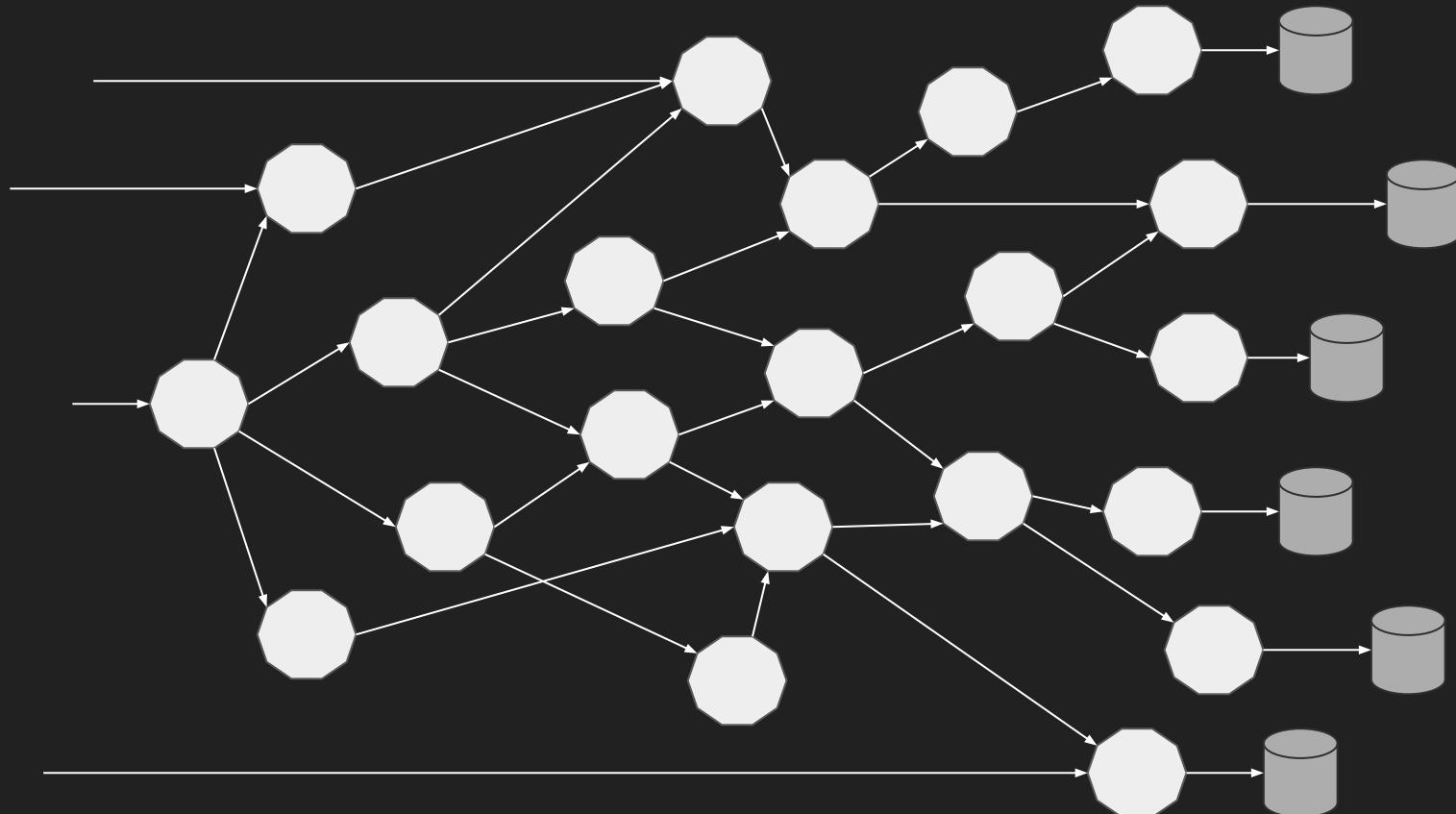
Network of Services - Mesh



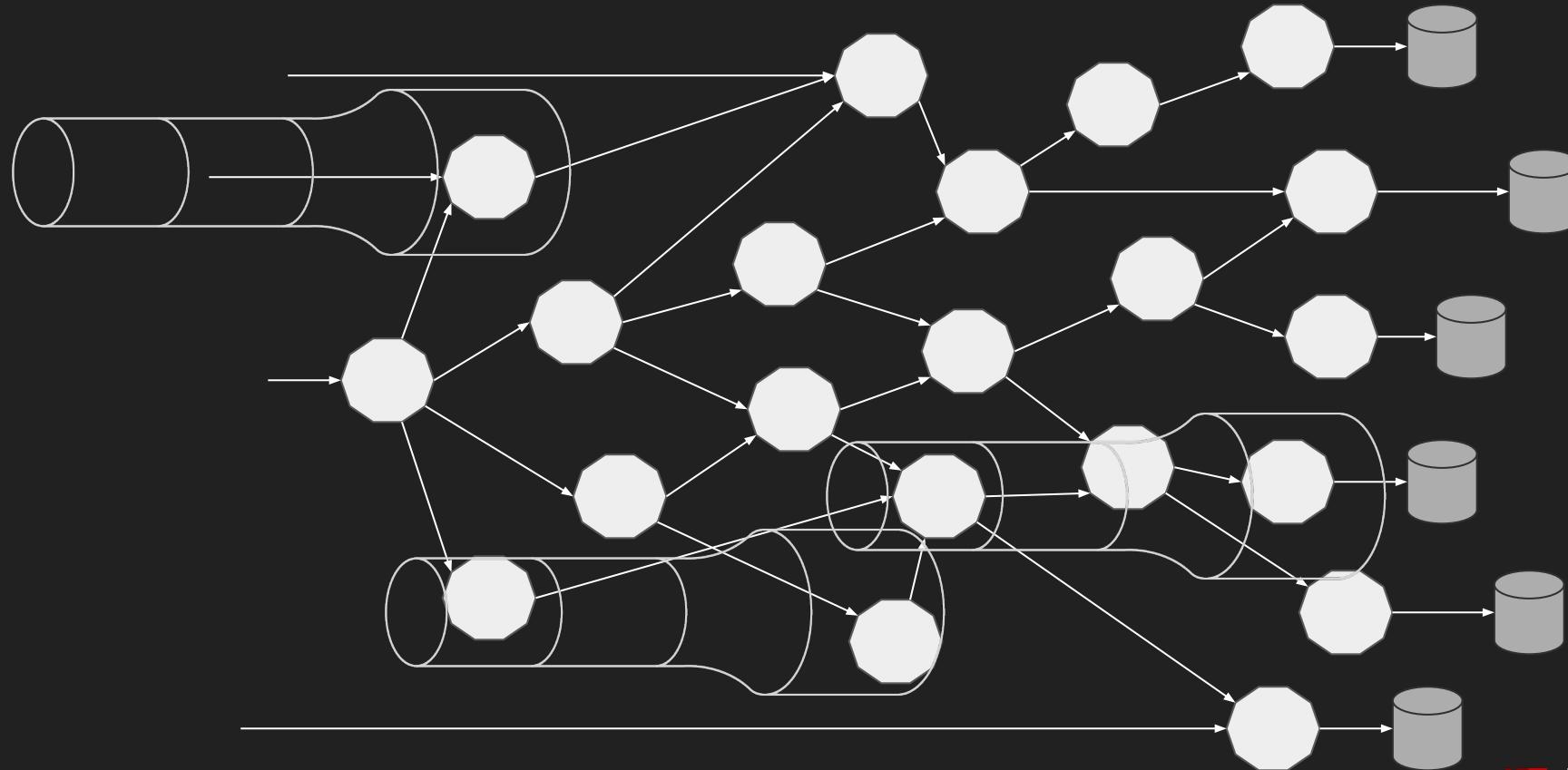
Microservices own their Data



Multiple Points of Entry

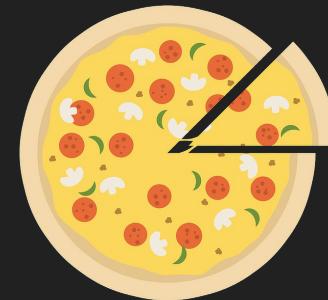


Multiple Pipelines



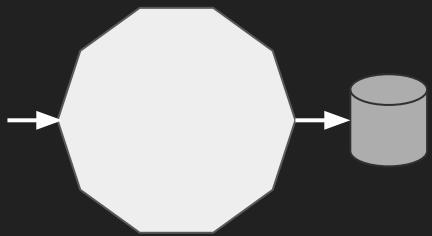
Microservices Principles

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



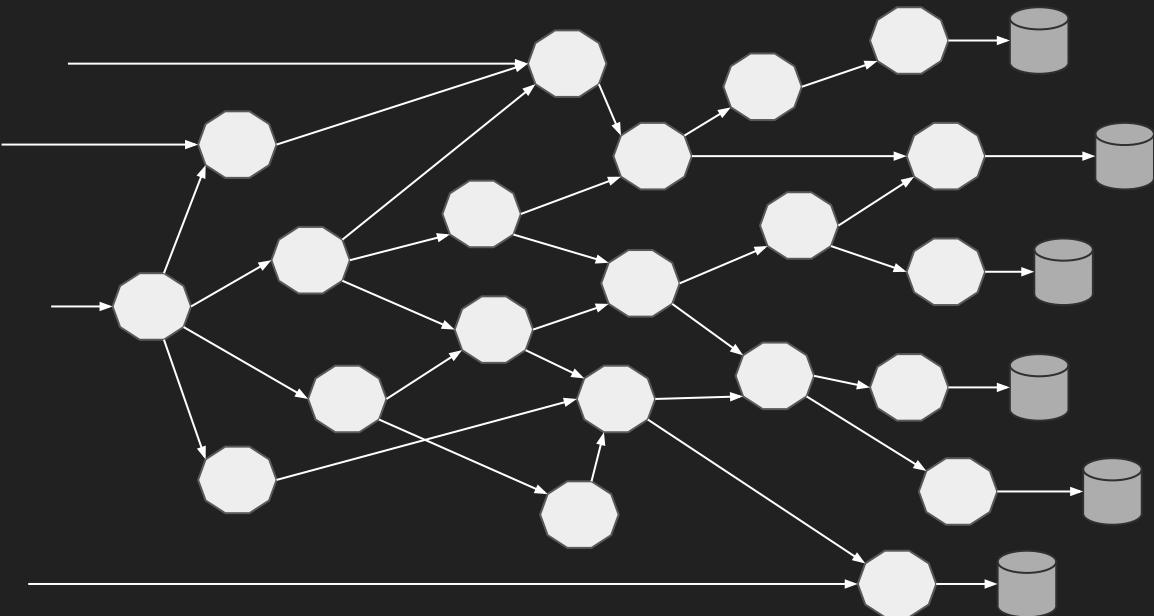
2 Pizza Team

Old School



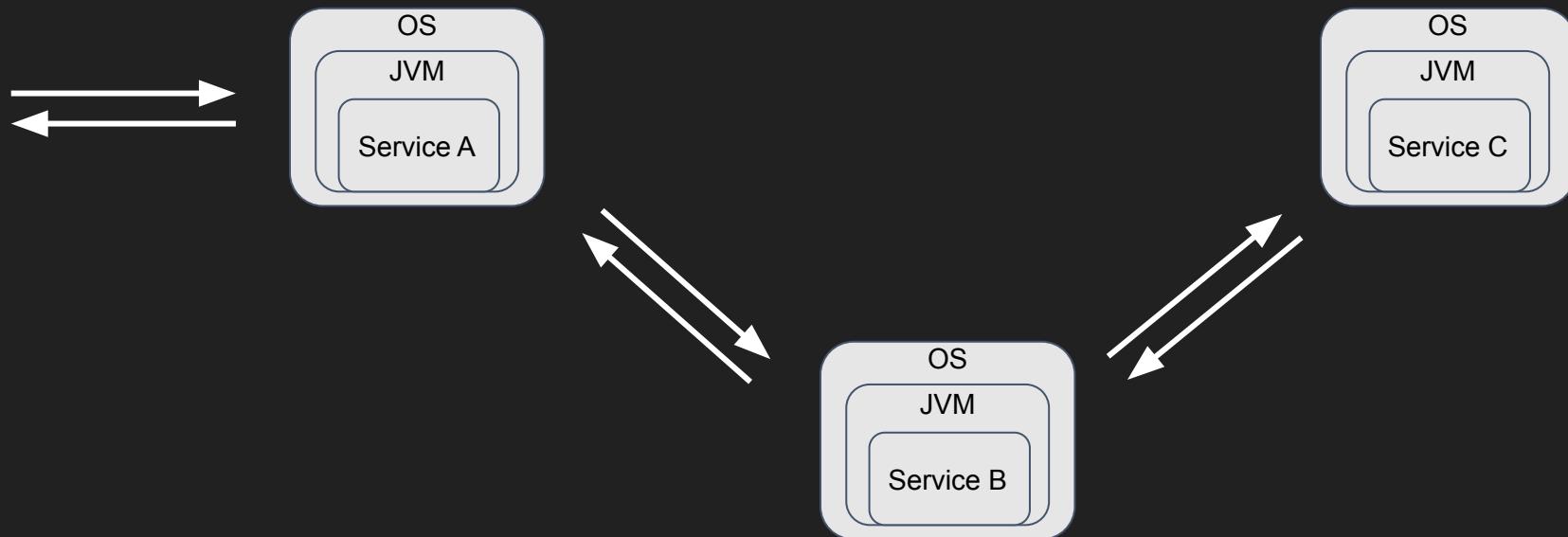
Love Thy Mono

New School



OPENSHIFT

Microservices == Distributed Computing

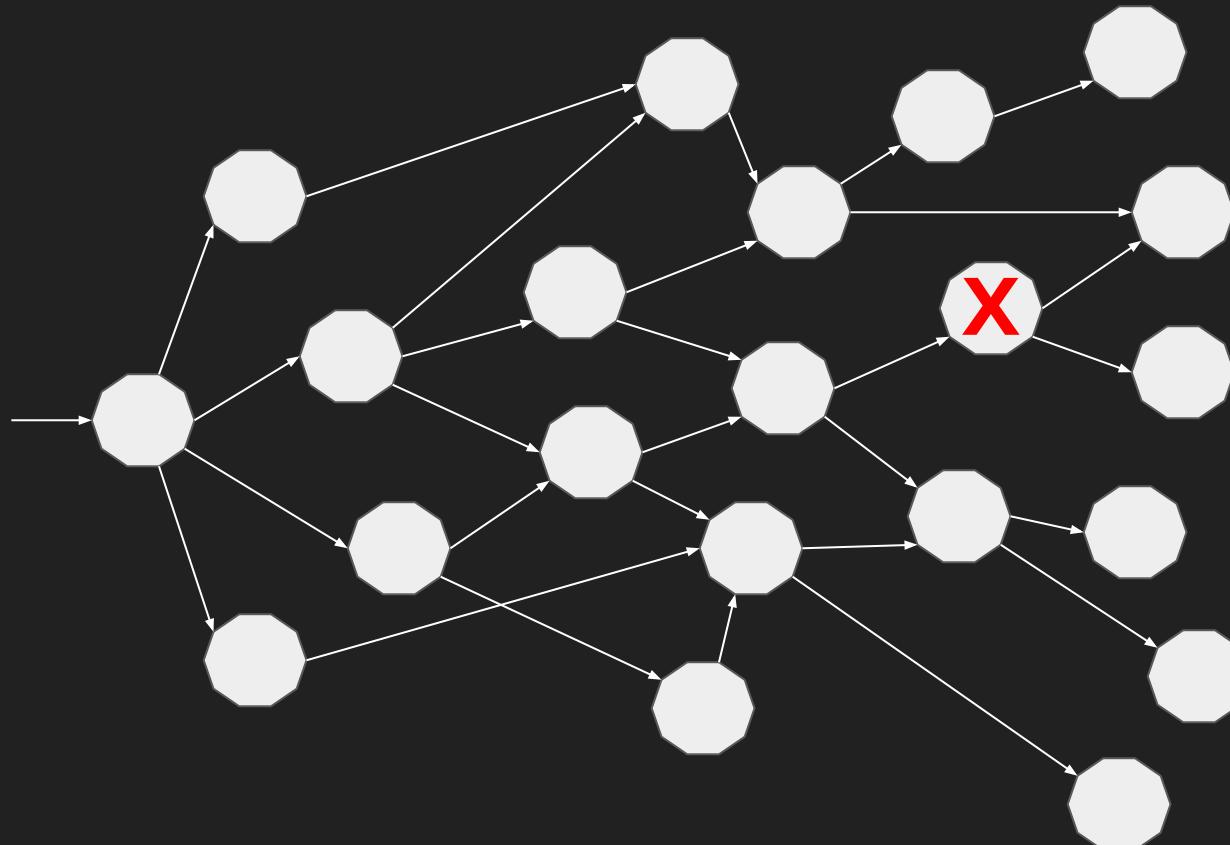


Fallacies of Distributed Computing

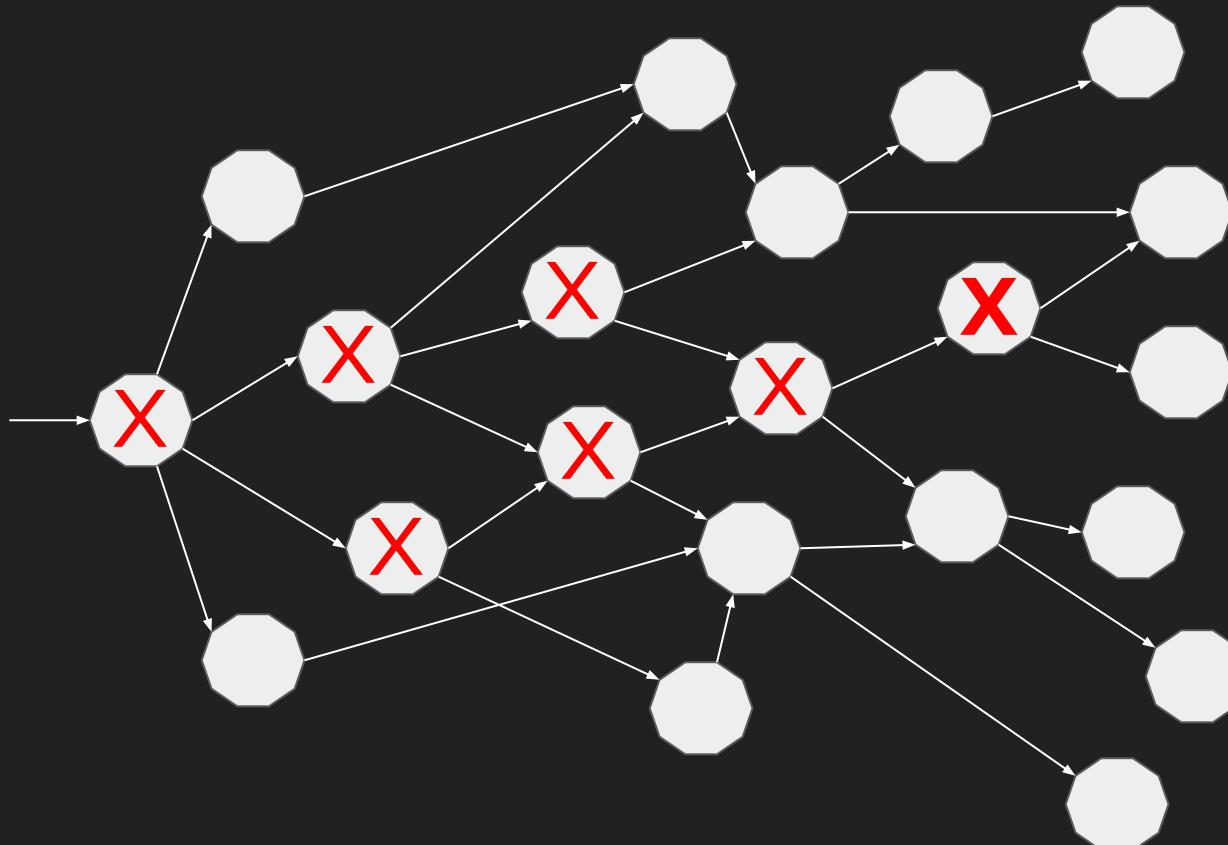
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

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

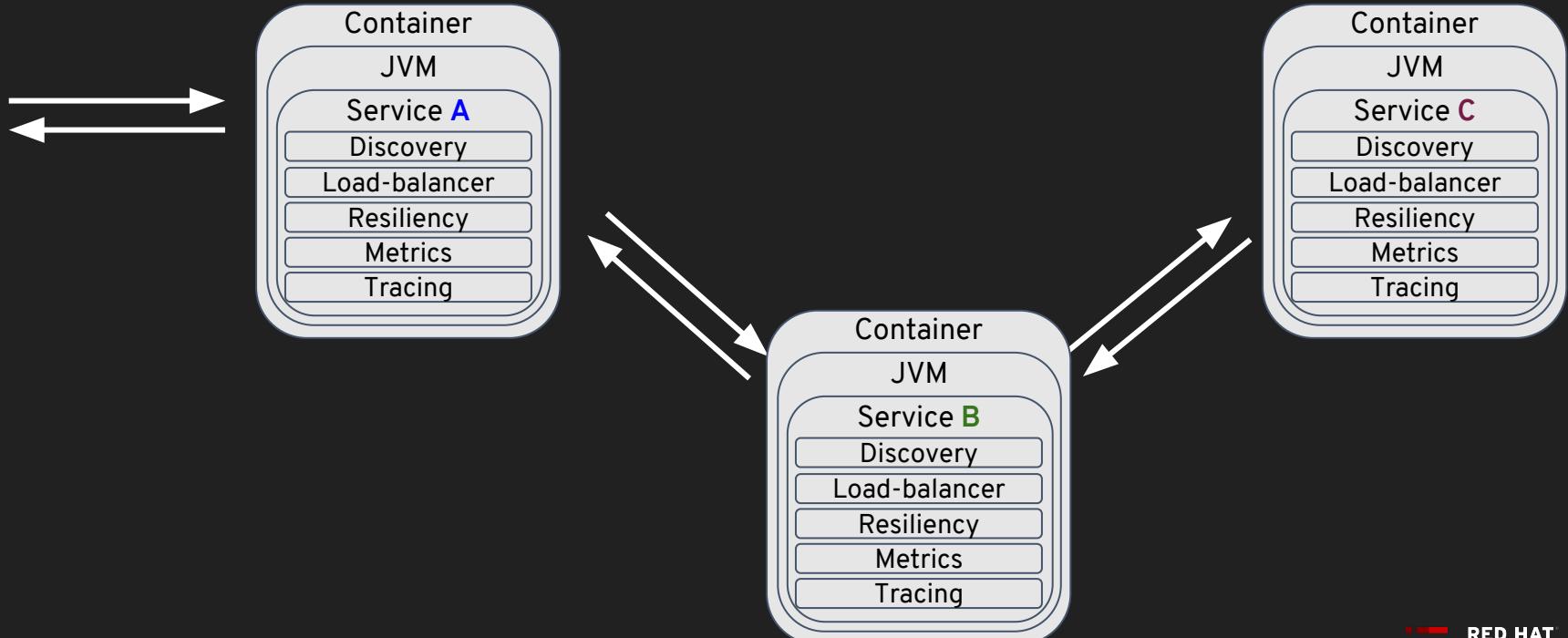
Failure of a Service



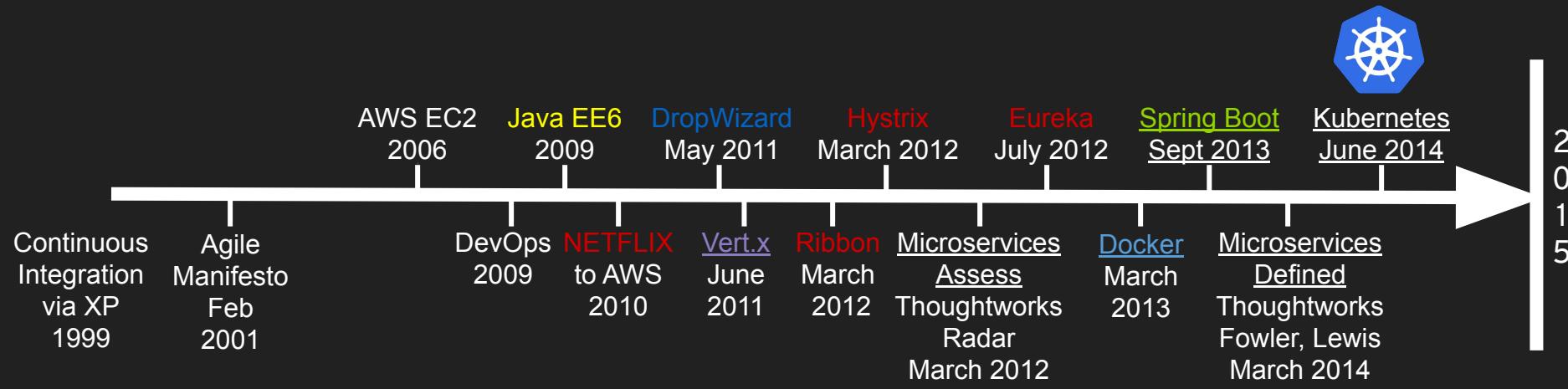
Cascading Failure



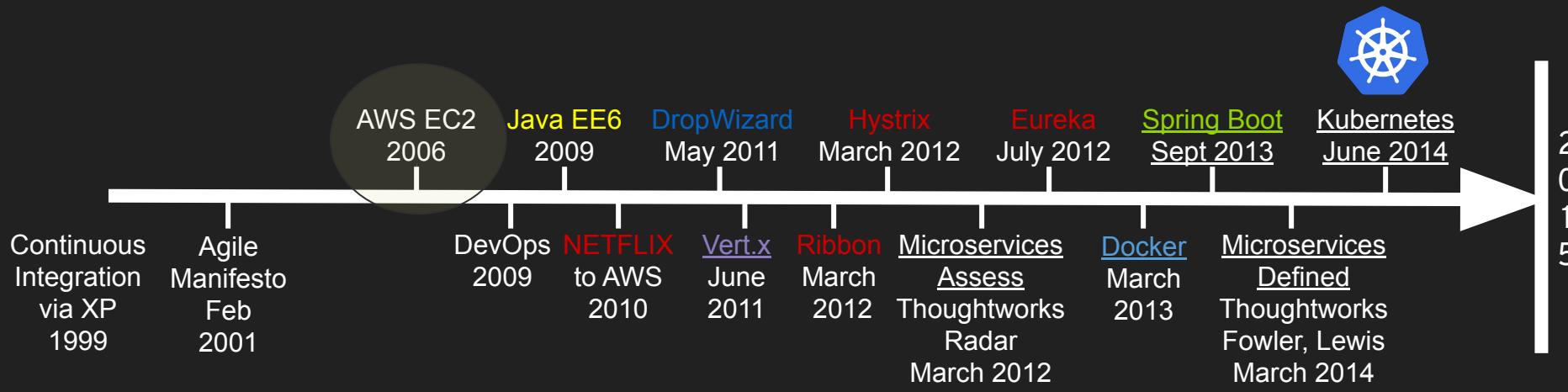
Microservices embedding Capabilities



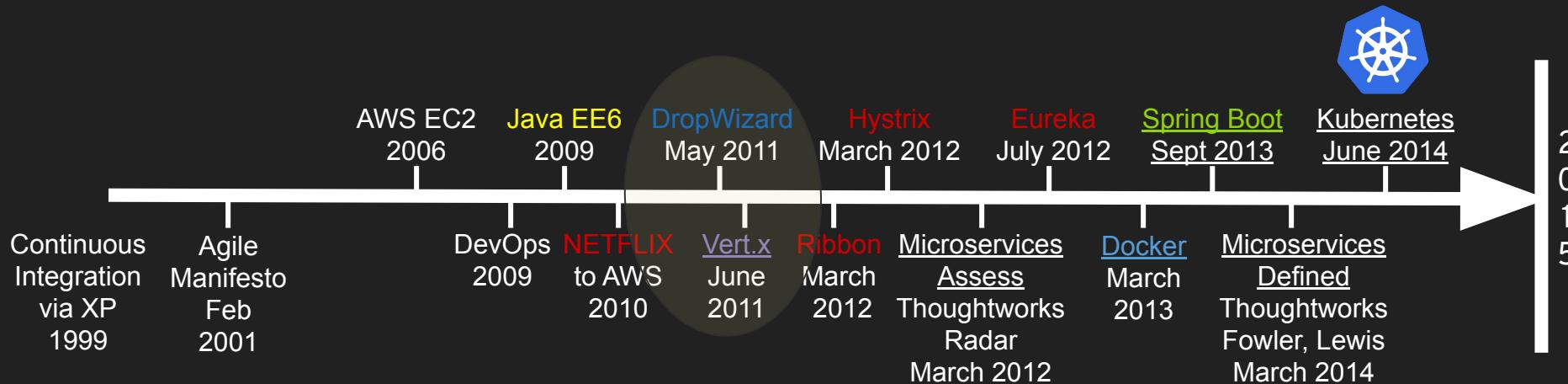
History of Microservices



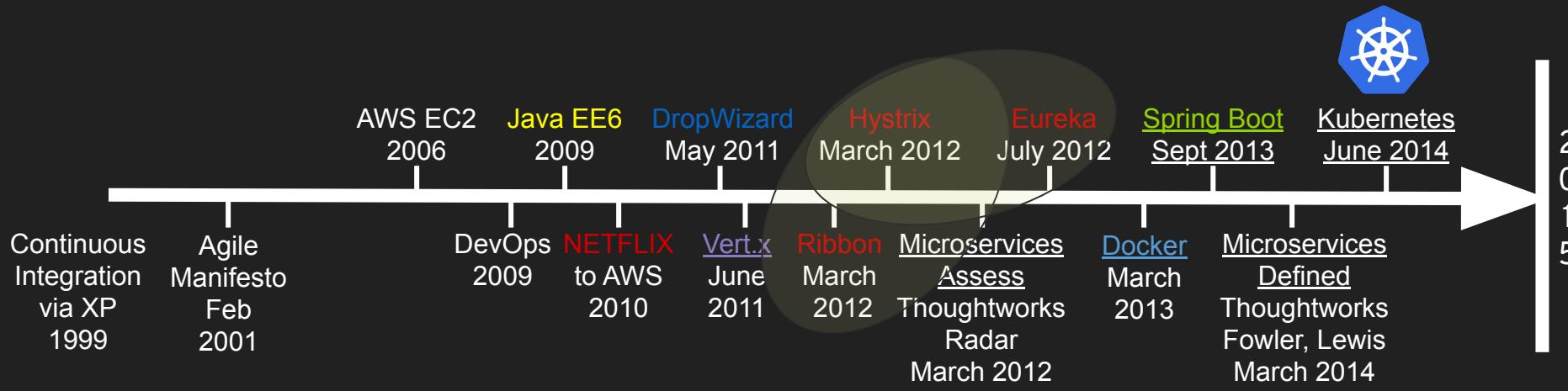
The Cloud is Born



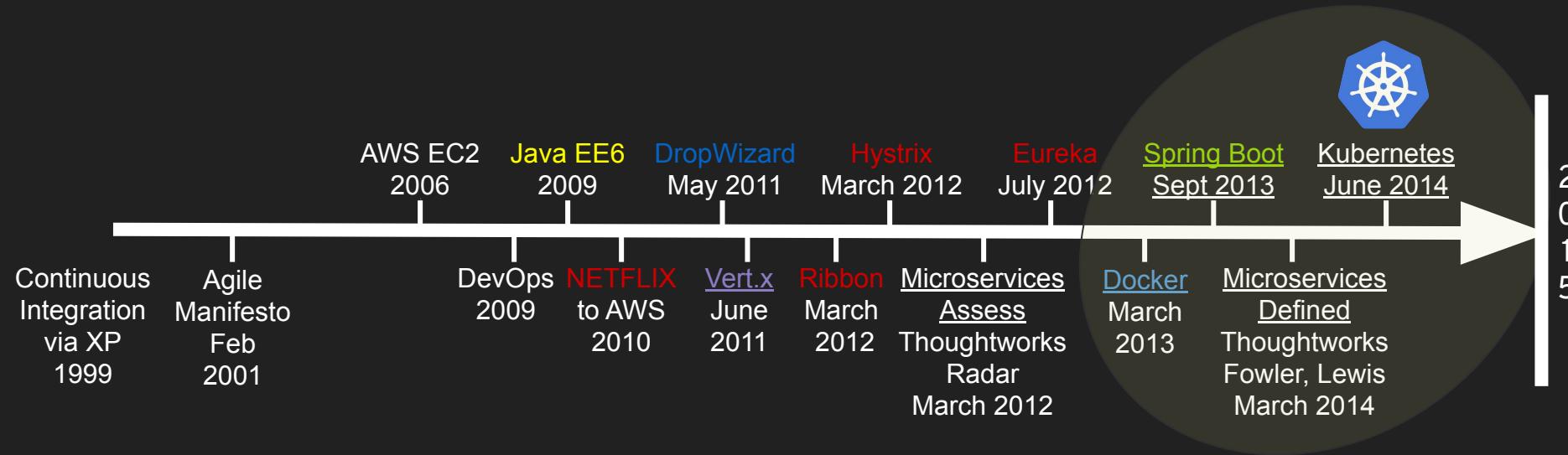
Fat Jars



Netflix goes Open Source



Perfect Storm for Microservices



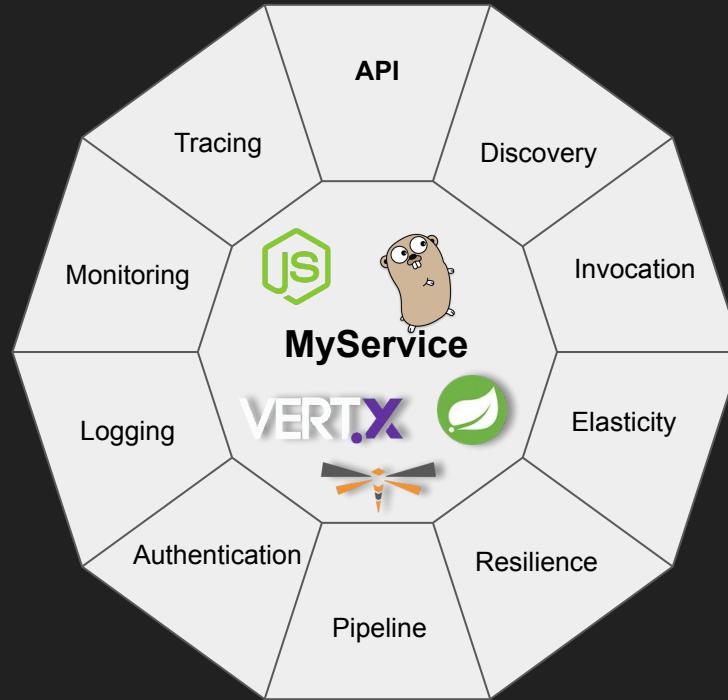
What's Wrong with Netflix OSS?

Java Only

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



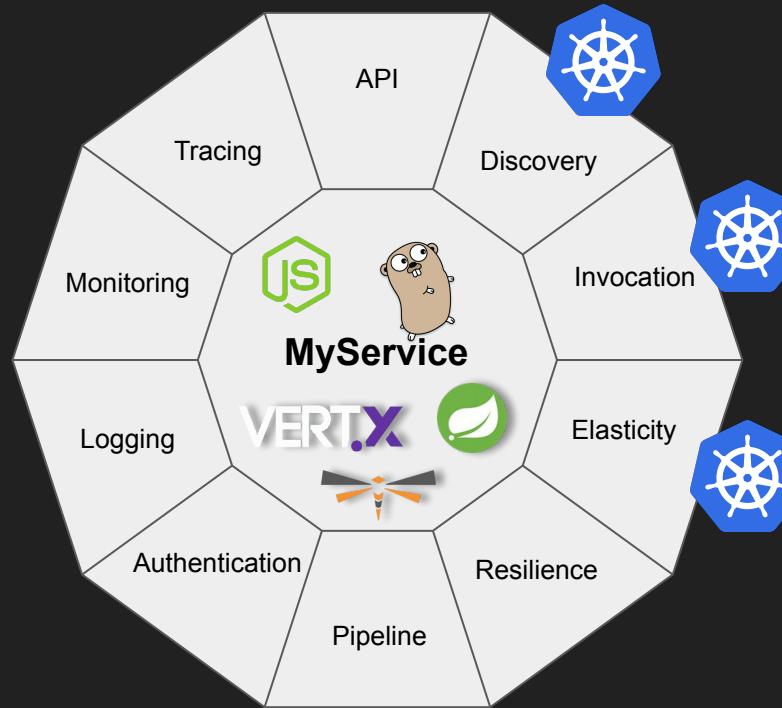
Microservices'ilities



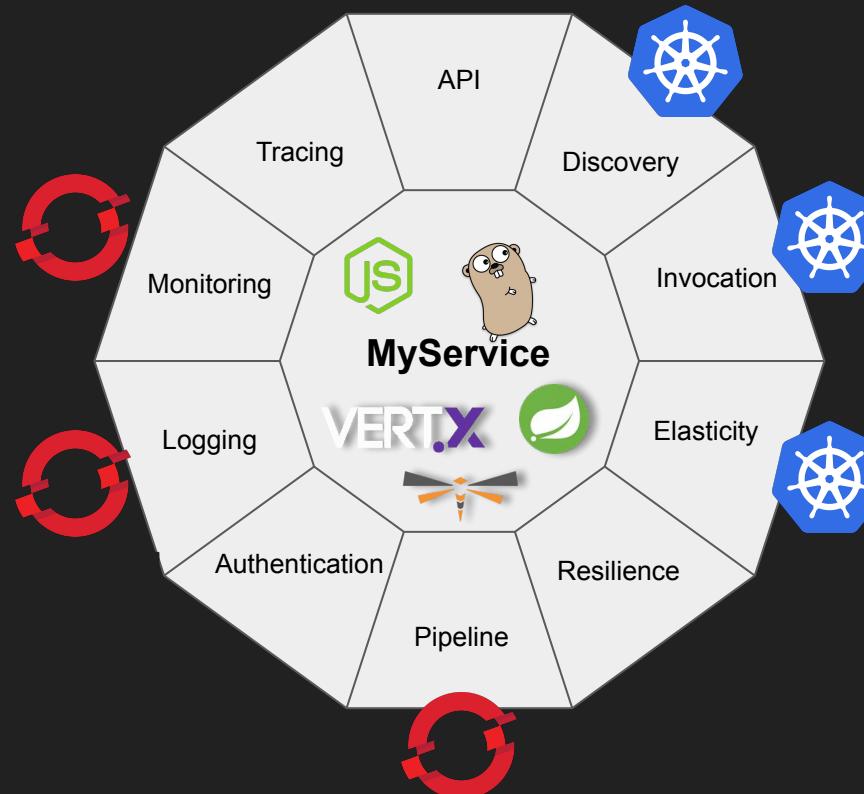


OPENSIFT

Microservices'ilities + Kubernetes



Microservices'ilities + OpenShift





Istio - Sail

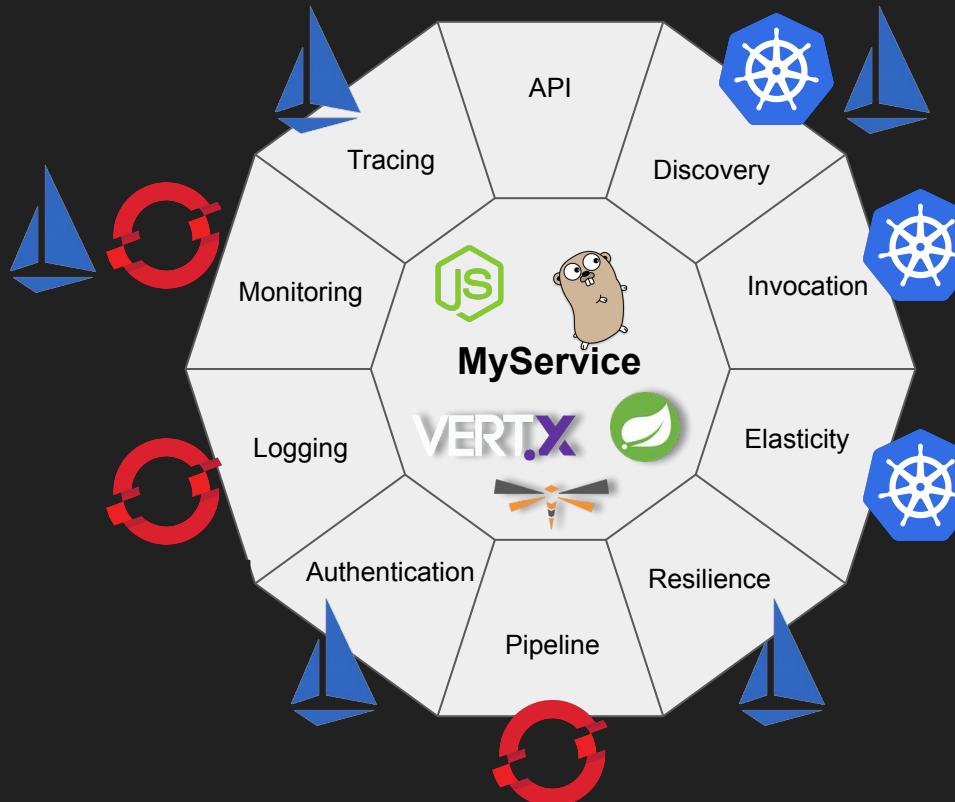
(Kubernetes - Helmsman or ship's pilot)

Service Mesh Defined

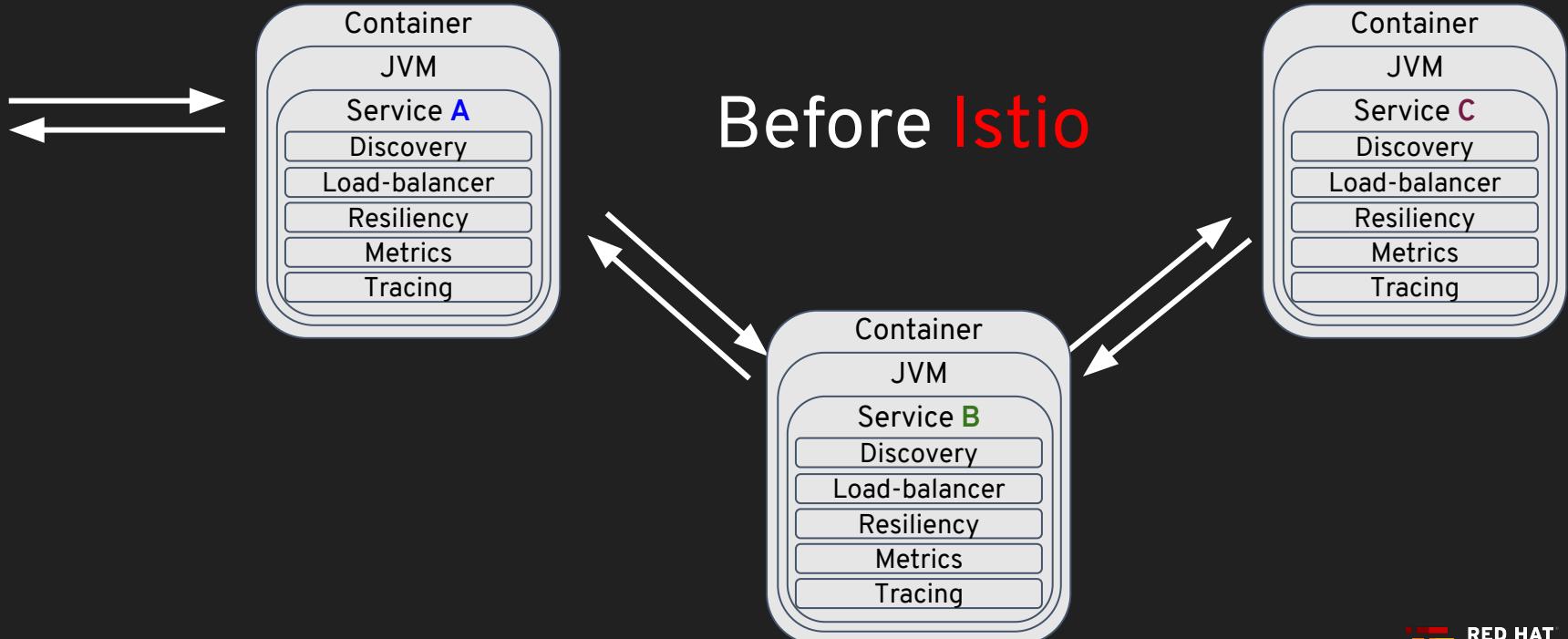
A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as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

<https://buoyant.io/2017/04/25/whats-a-service-mesh-and-why-do-i-need-one/>

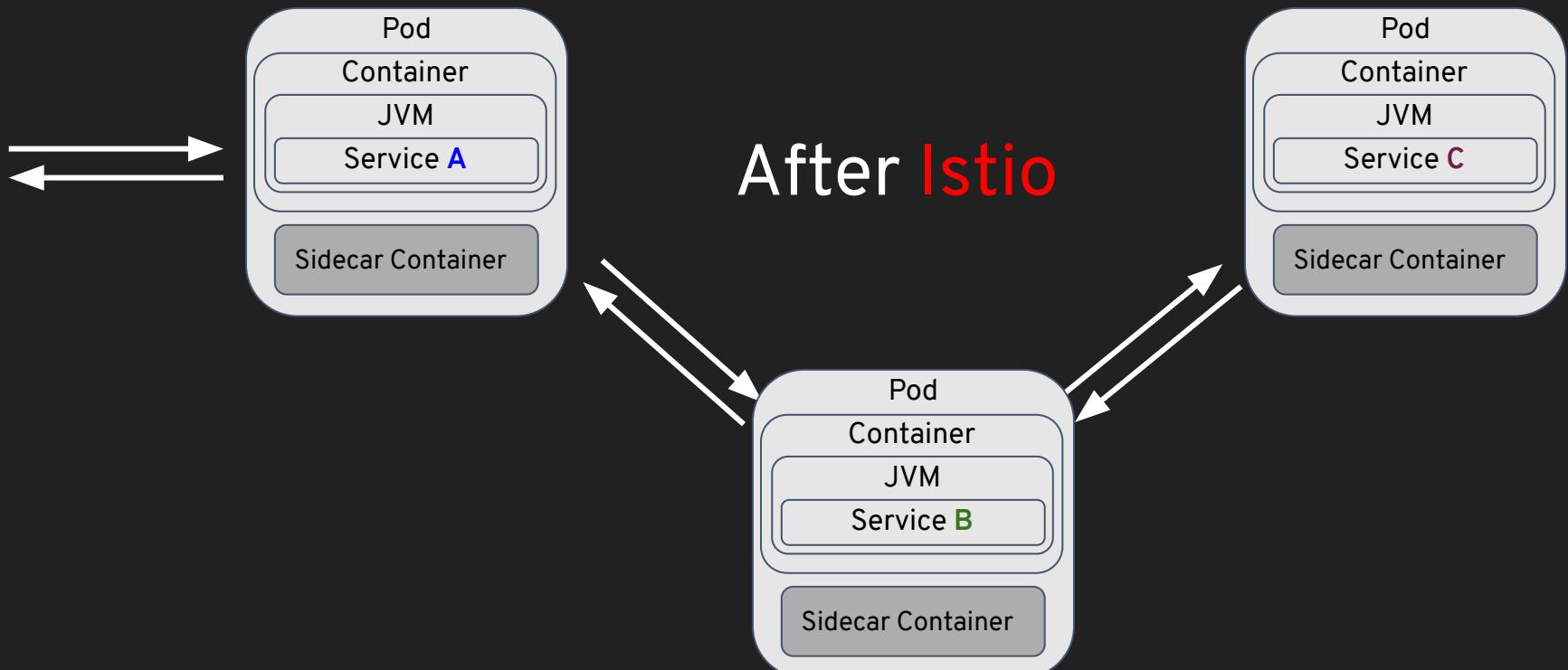
Microservices'ilities + Istio



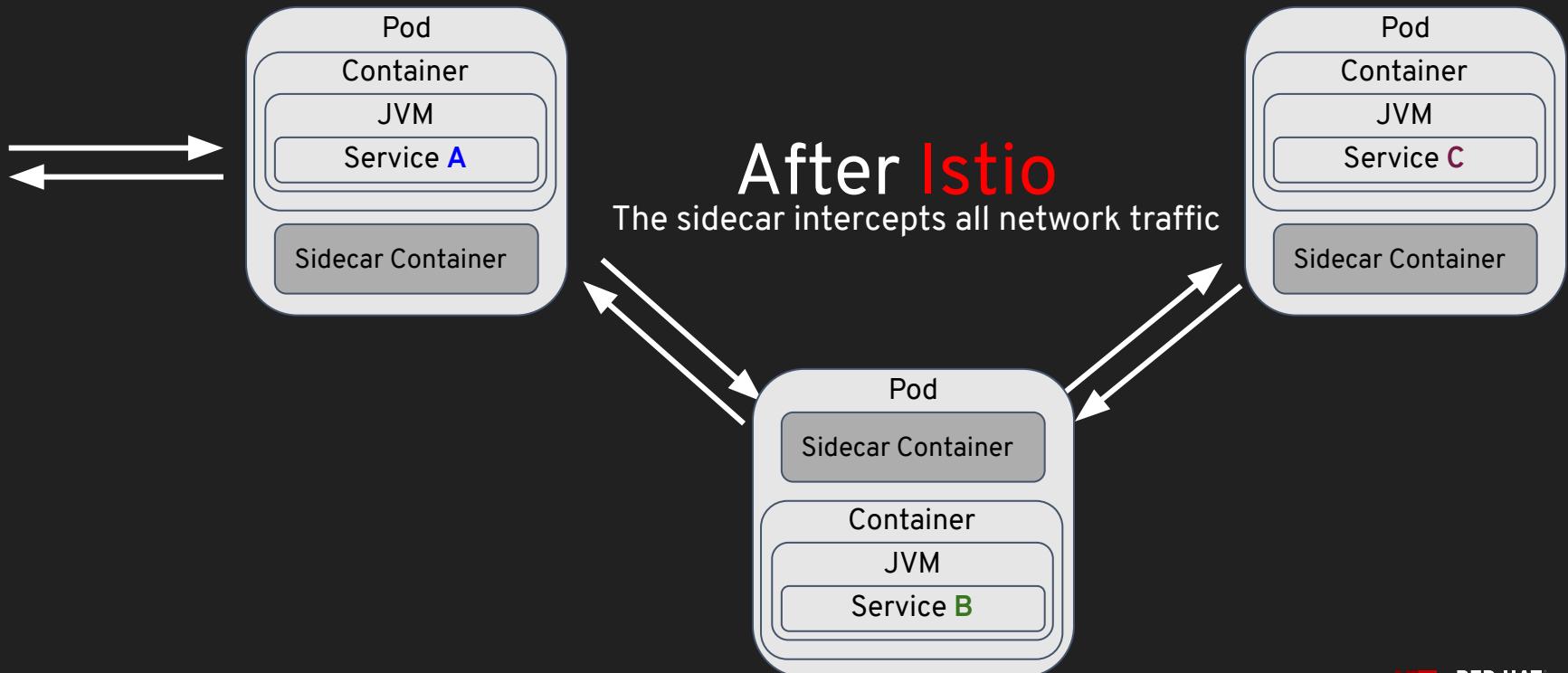
Microservices embedding Capabilities



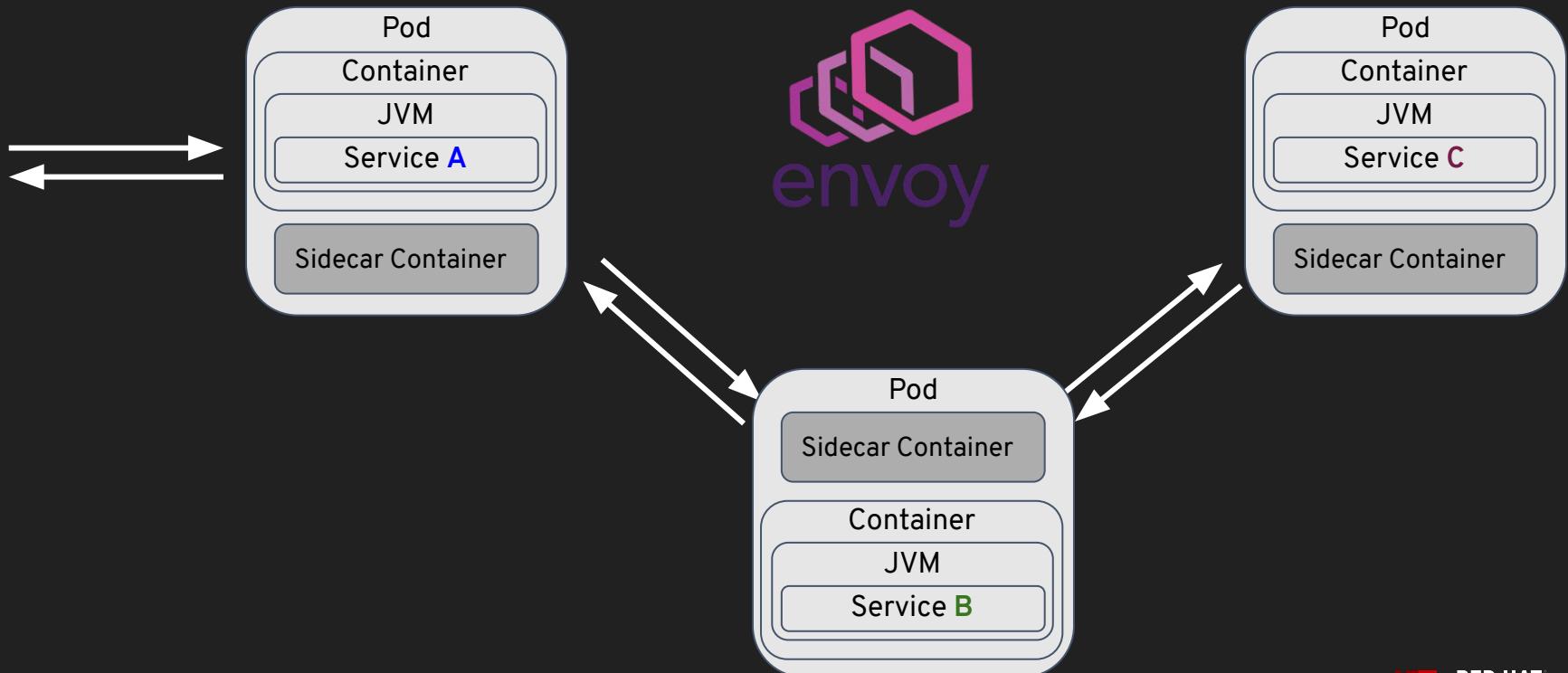
Microservices externalizing Capabilities



Microservices externalizing Capabilities



Envoy is the current sidecar





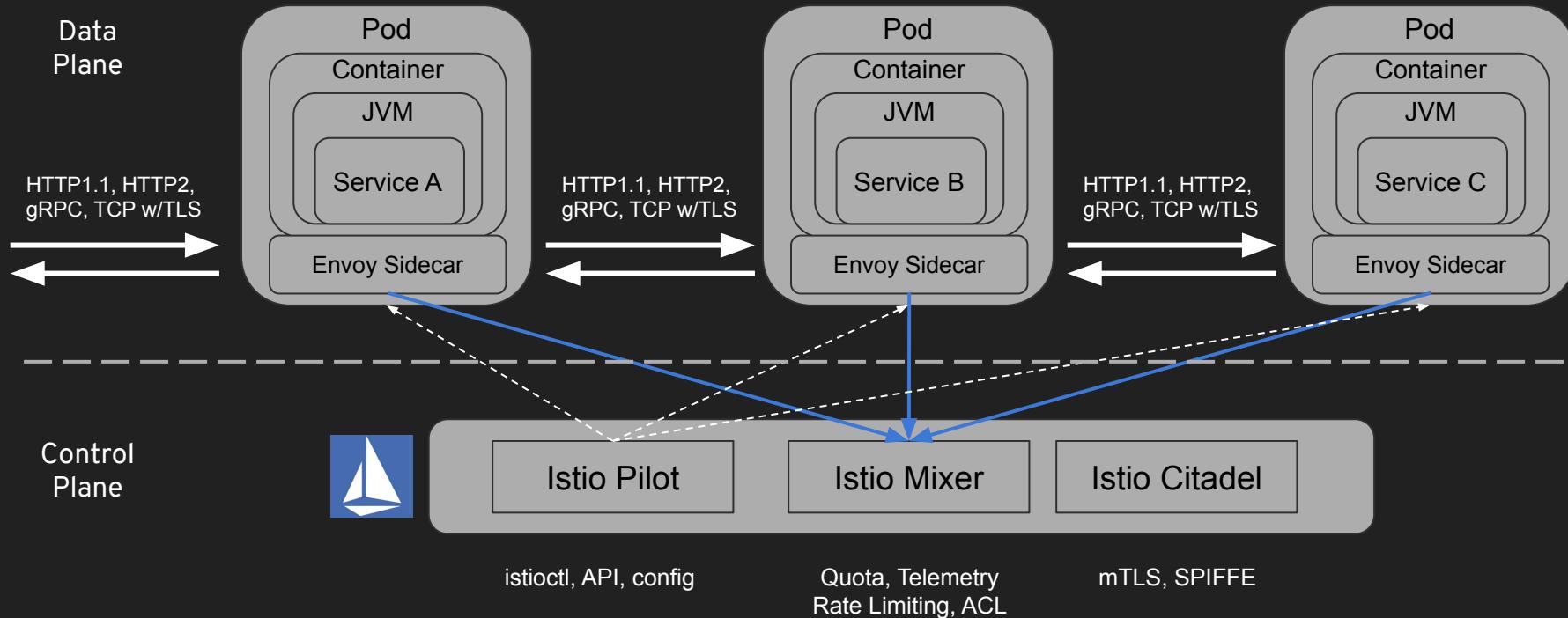
<https://www.imz-ural.com/blog/waffles-the-sidecar-dog>

Next Generation Microservices - Service Mesh

Code Independent (Polyglot)

- Intelligent Routing and Load-Balancing
 - A/B Tests
 - Smarter Canary Releases
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability: Metrics and Tracing
- Fleet wide policy enforcement

Istio Data Plane vs Control Plane



Polyglot Microservices Platform circa 2019

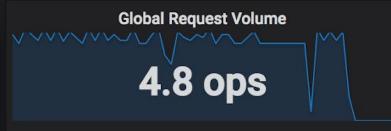


Observability



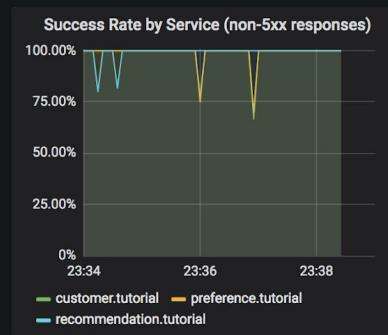
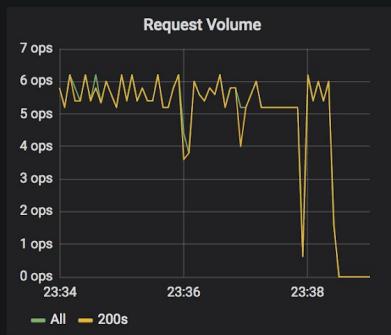
Istio Dashboard

Last 5 minutes Refresh every 5s



Service Mesh

Service Mesh



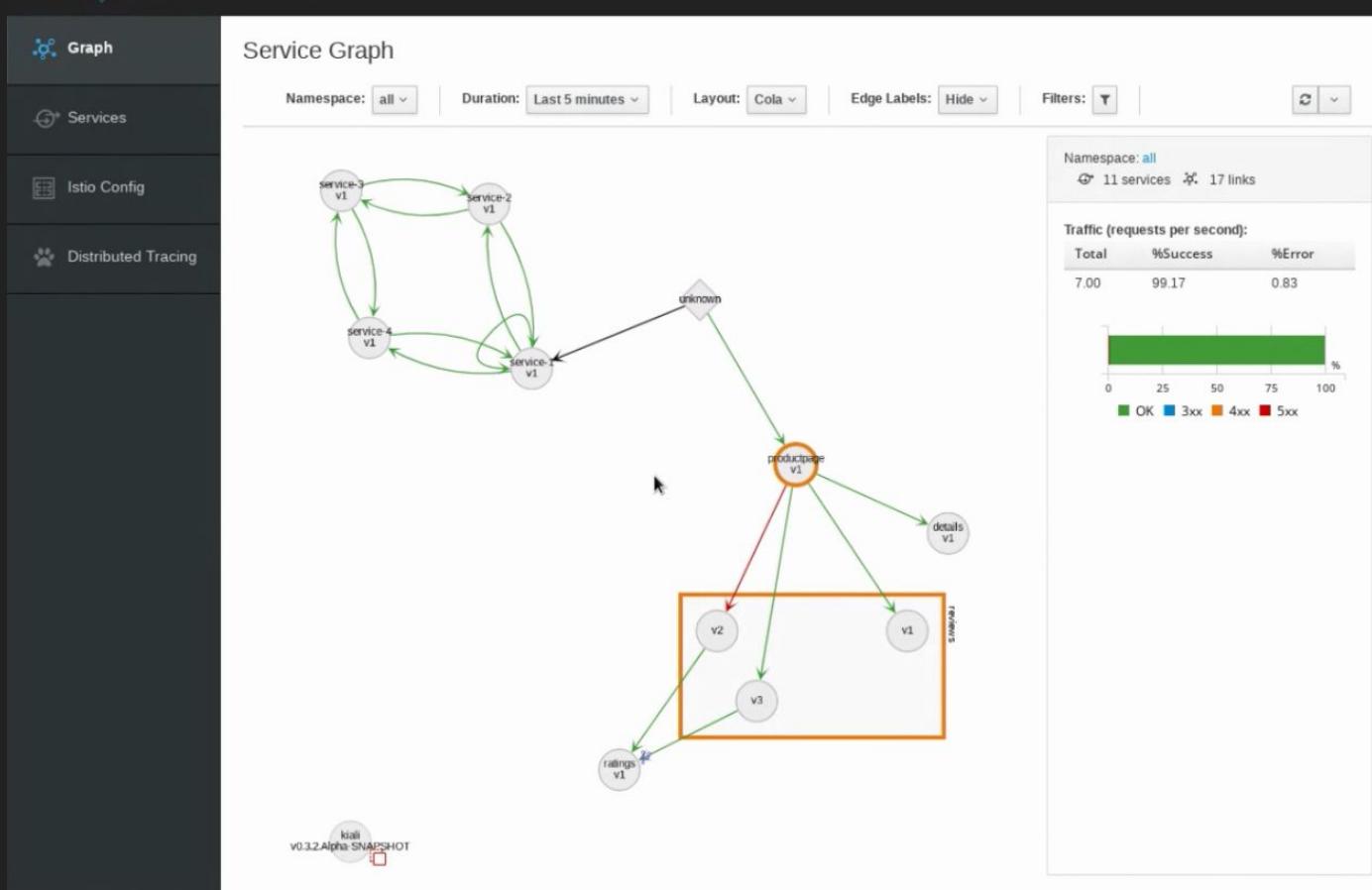
Services

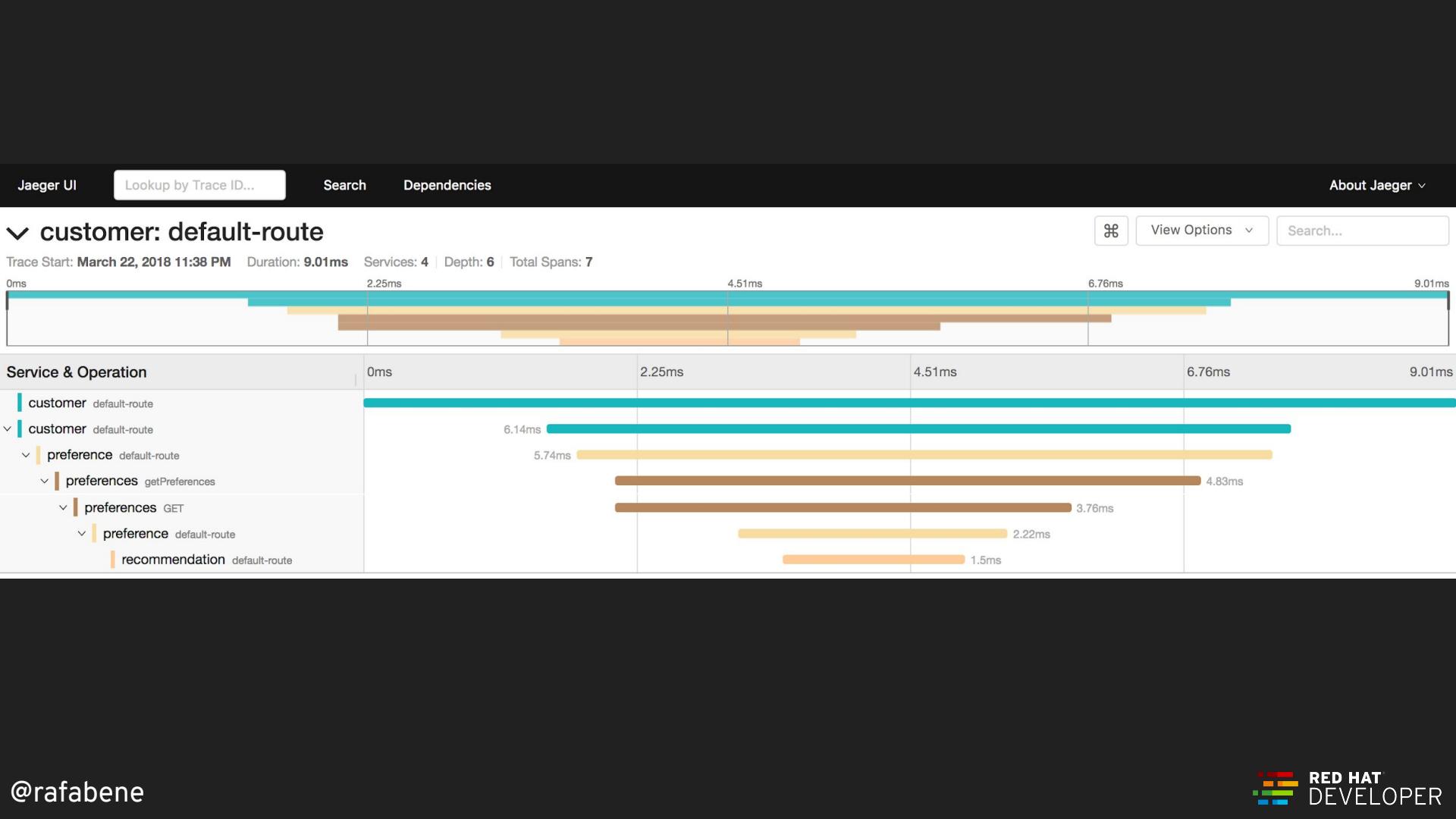
HTTP Services

customer.tutorial.svc.cluster.local



Kiali.io New Service Graph





Prometheus



How to add an Istio-Proxy (sidecar)?

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

OR

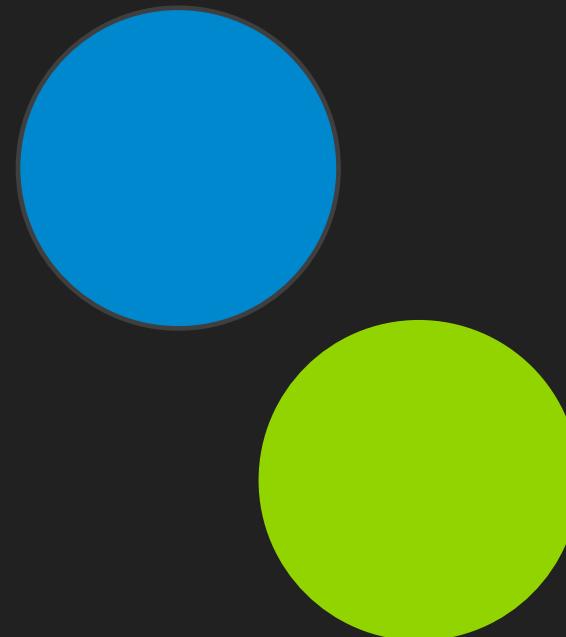
```
kubectl label namespace myspace istio-injection=enabled
```

To "see" the sidecar:

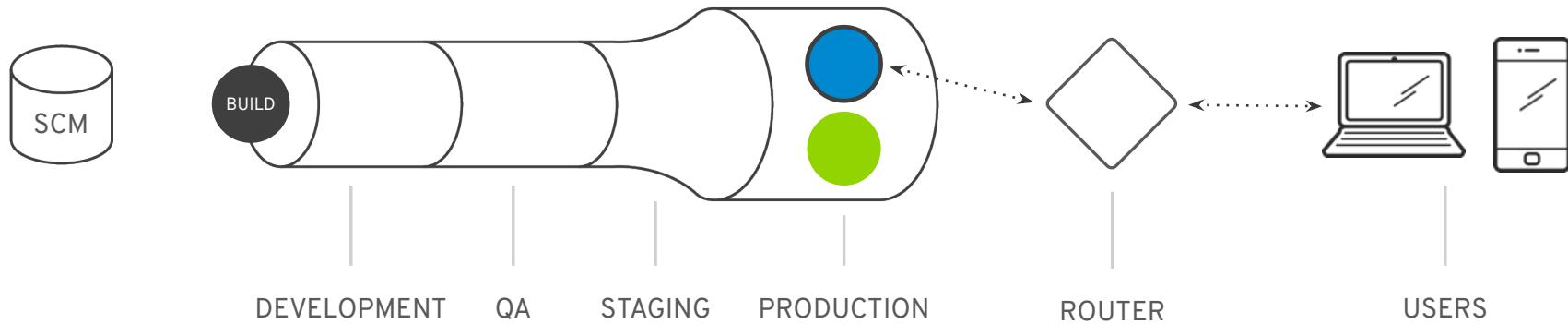
```
kubectl describe deployment customer
```

Traffic Control

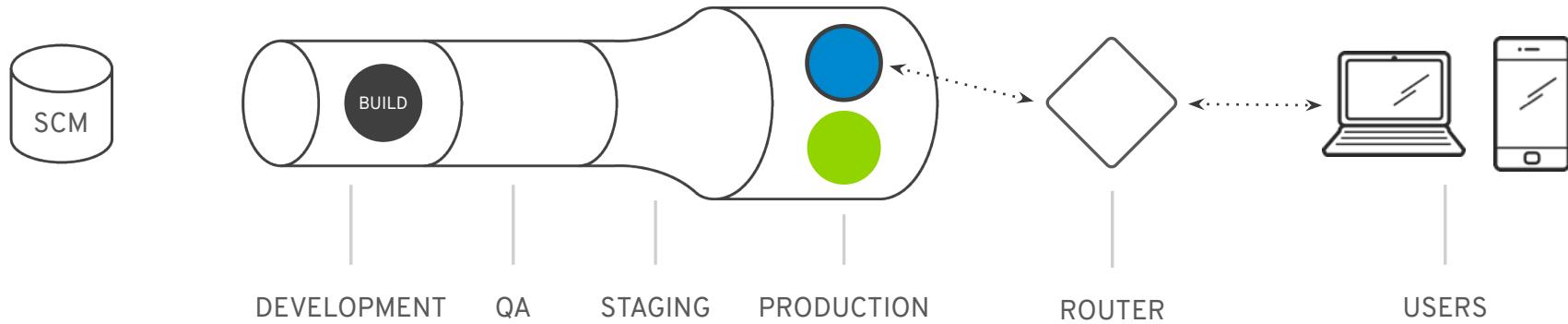
Blue/Green Deployment



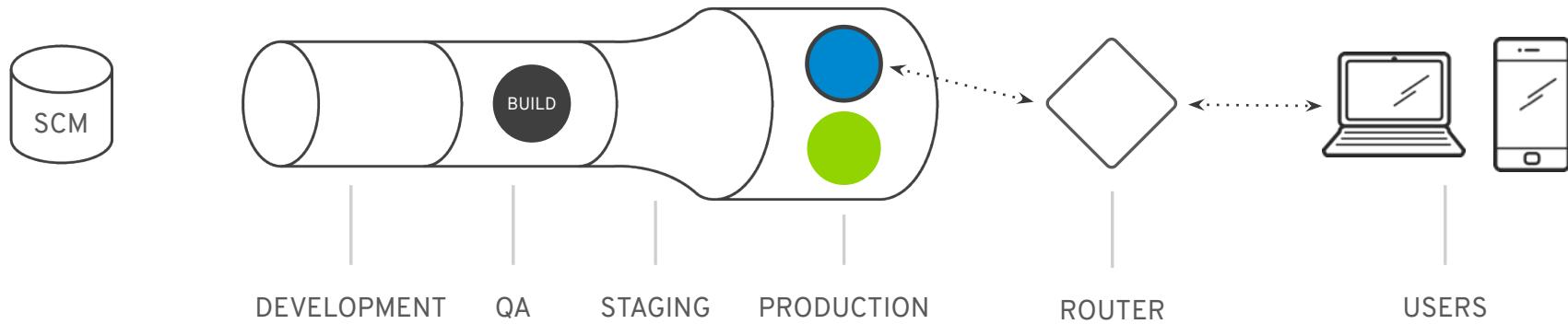
Blue/Green Deployment



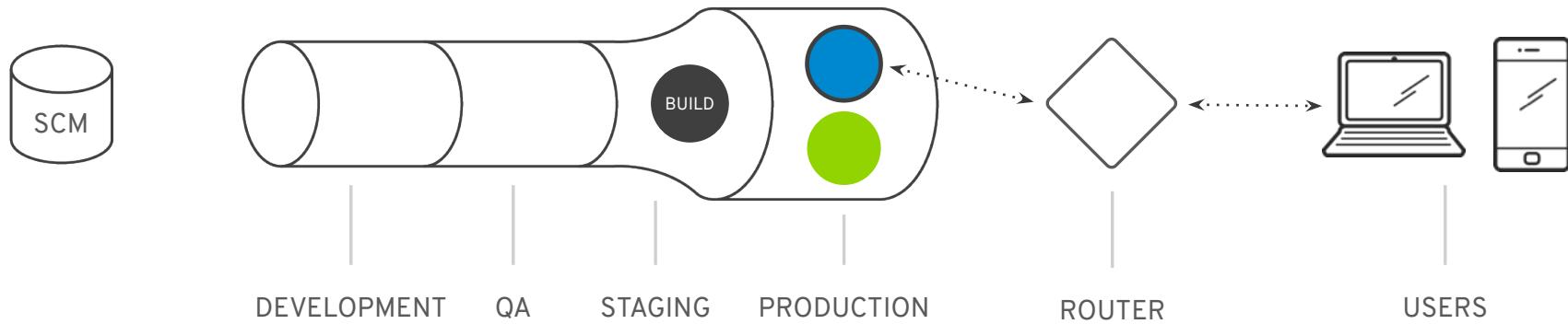
Blue/Green Deployment



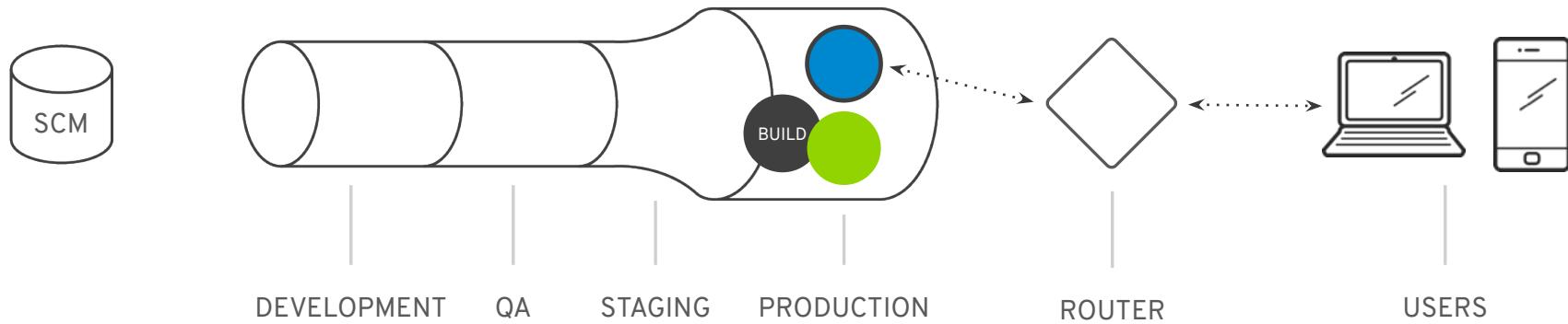
Blue/Green Deployment



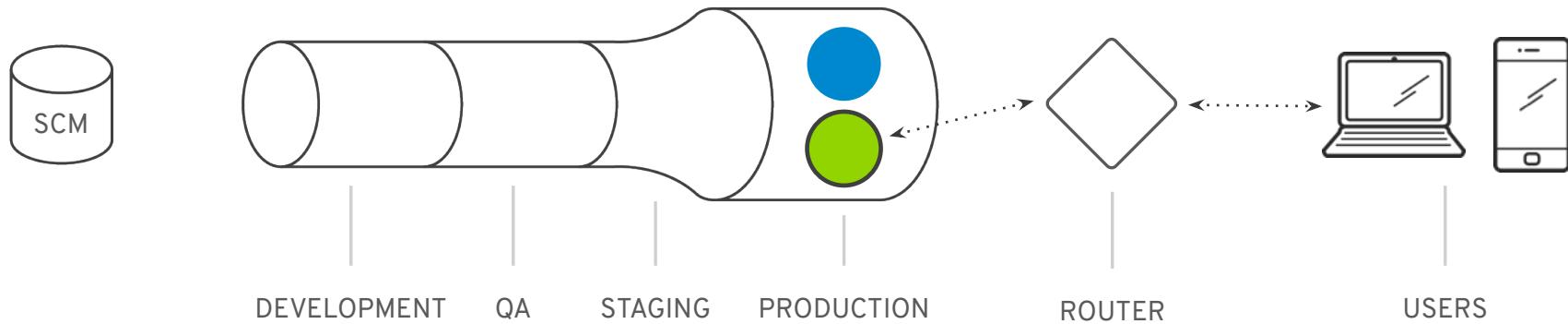
Blue/Green Deployment



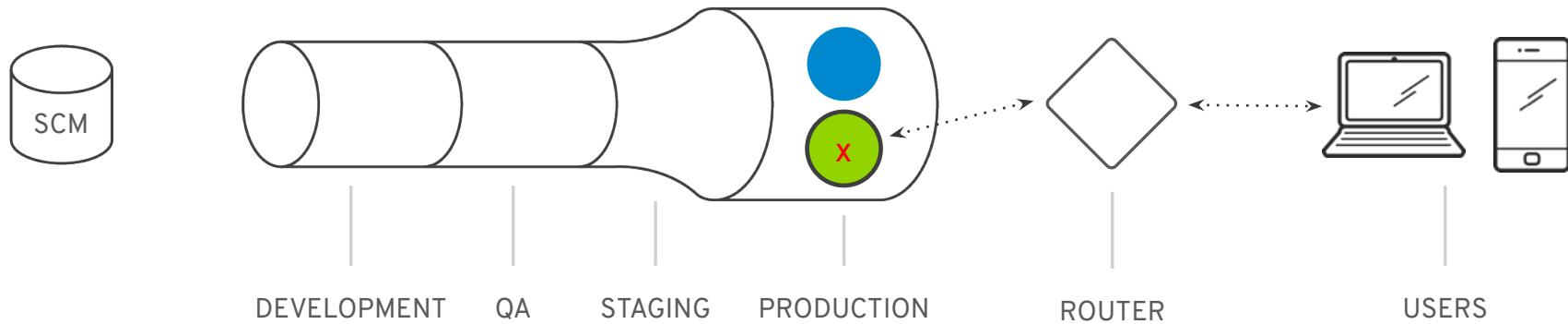
Blue/Green Deployment



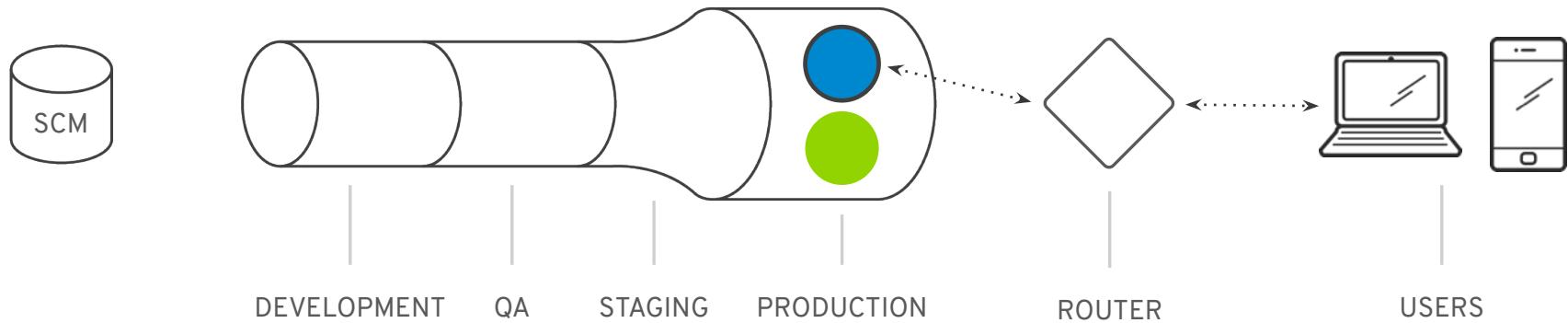
Blue/Green Deployment



Blue/Green Deployment



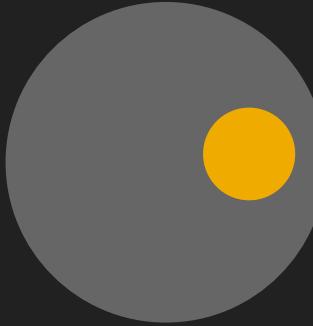
Blue/Green Deployment



Demo Blue/Green

- Only Recommendation-v2
- Only Recommendation-v1
- Both (Delete Rule)

Canary Deployment



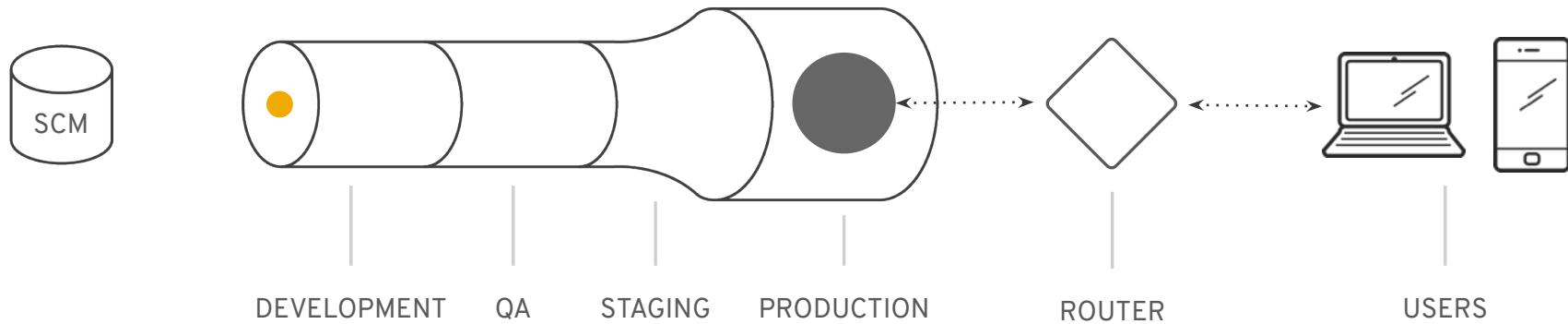


Canary Resuscitator

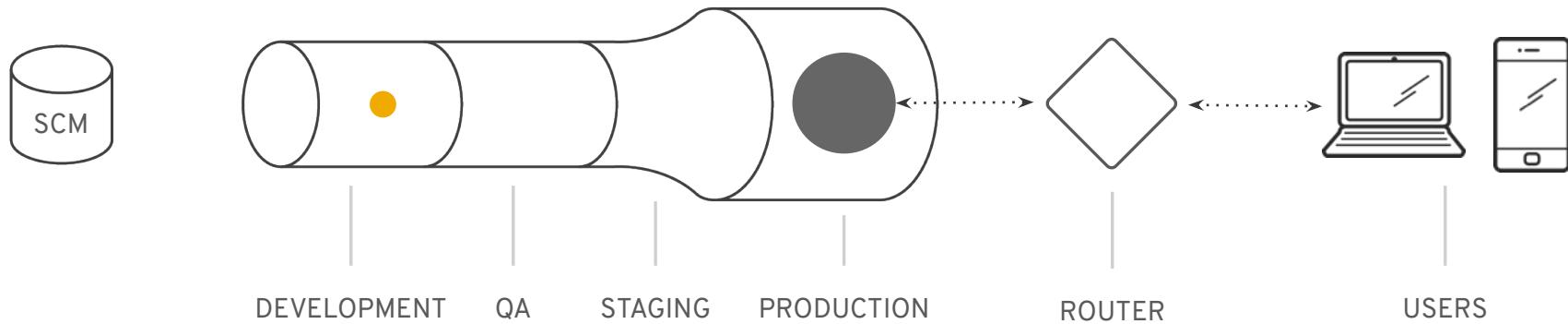


<http://www.openculture.com/2018/05/the-device-invented-to-resuscitate-canaries-in-coal-mines-circa-1896.html>
Thanks to **Paolo Antinori!**

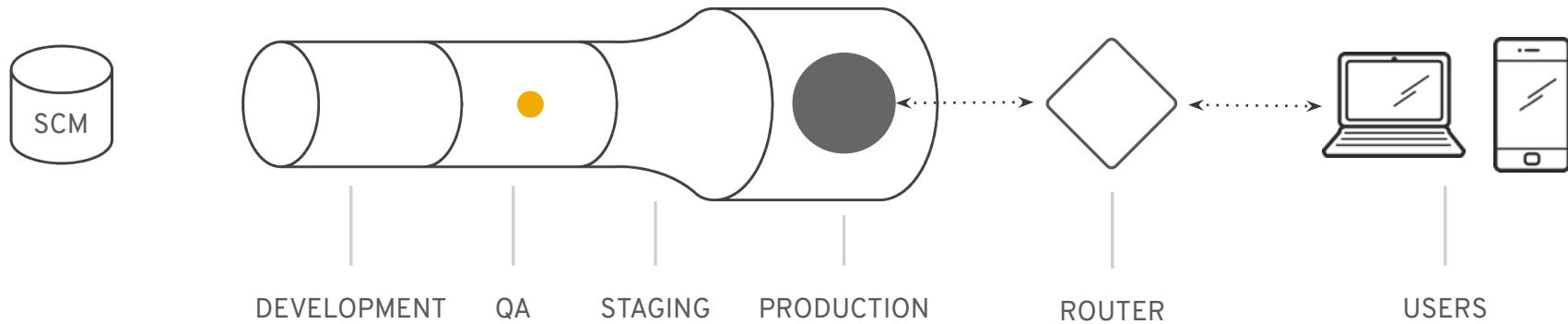
Canary Deployment



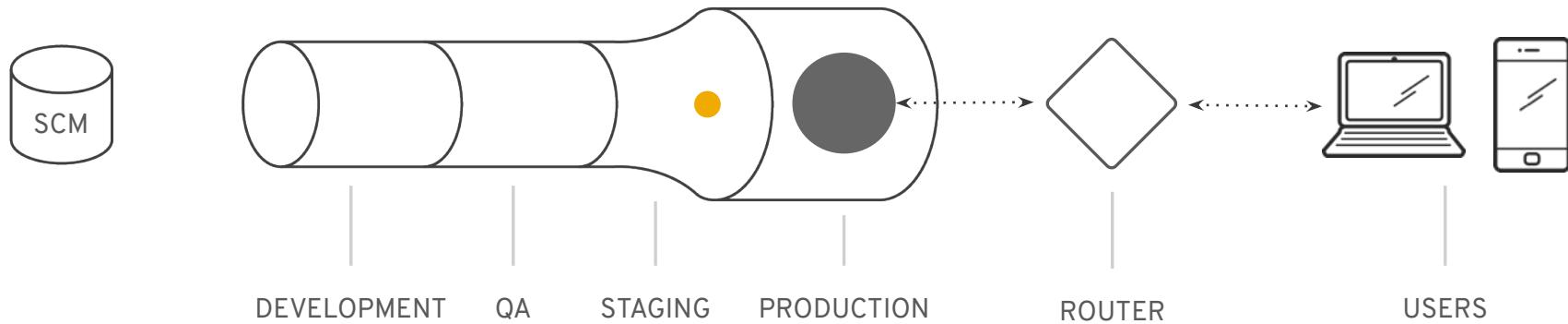
Canary Deployment



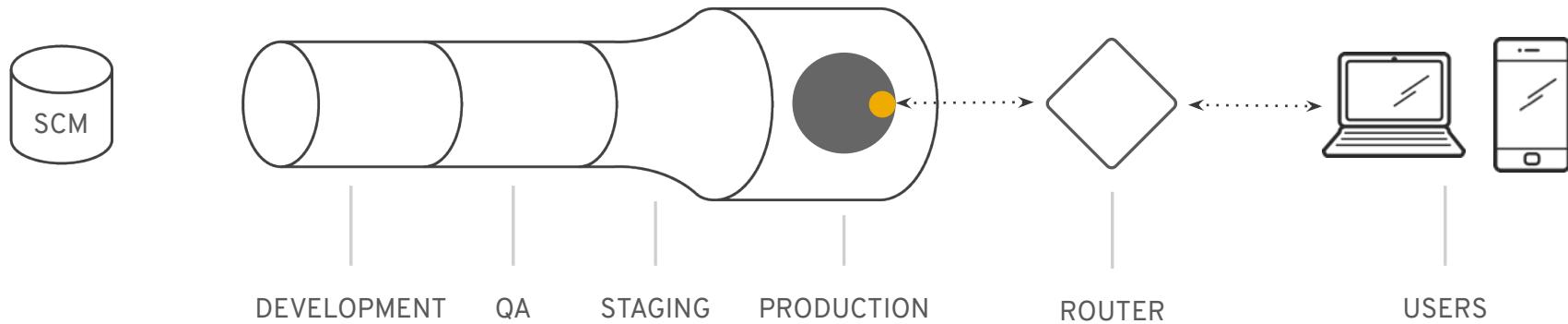
Canary Deployment



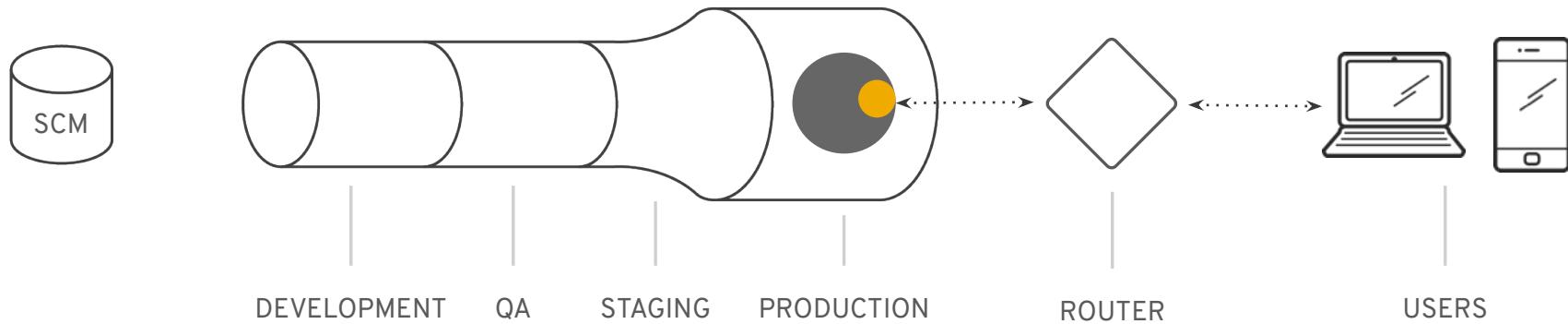
Canary Deployment



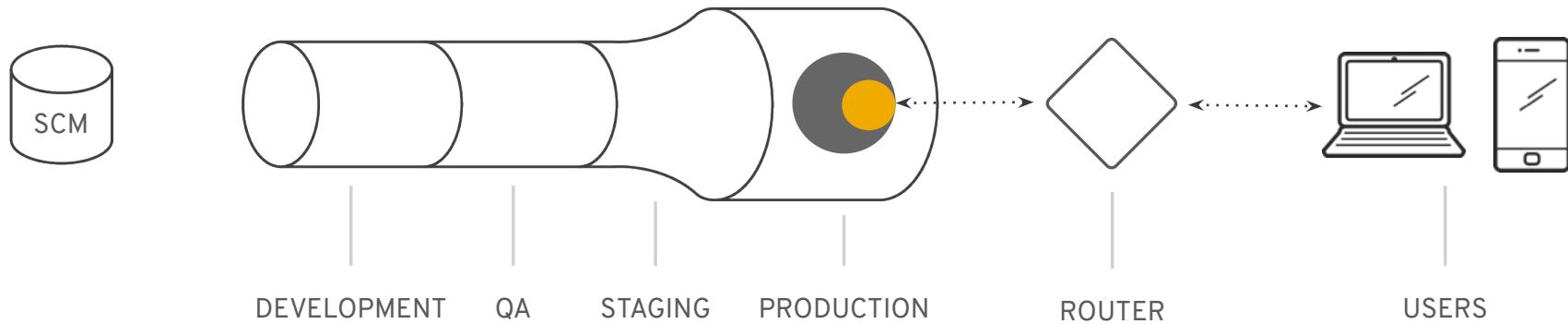
Canary Deployment



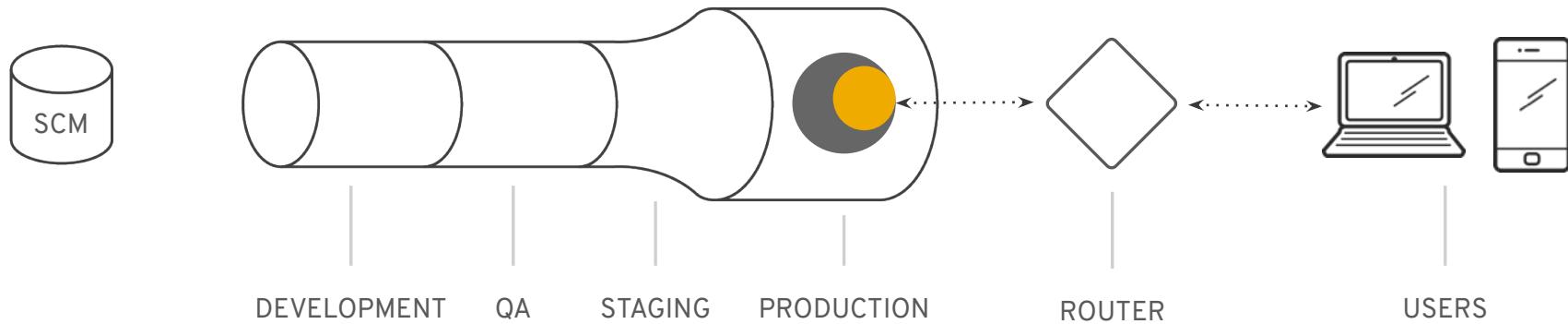
Canary Deployment



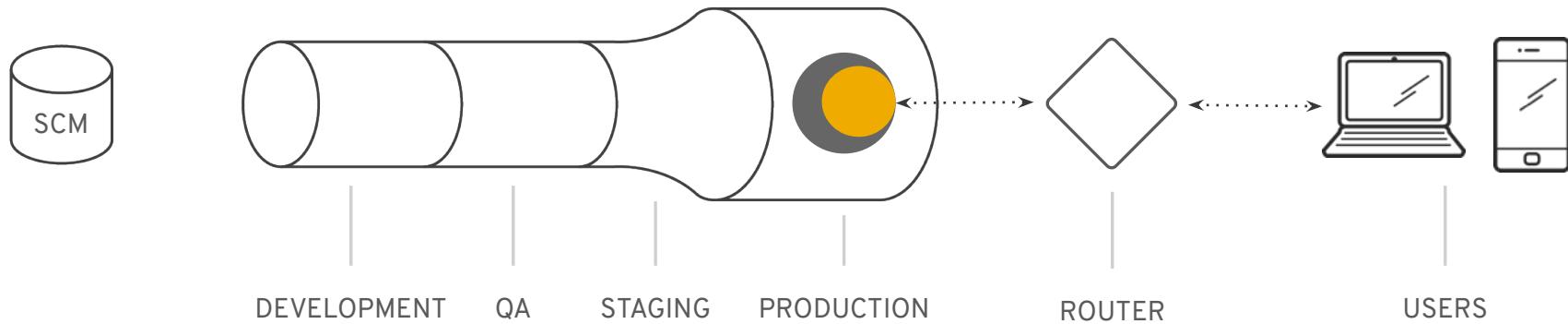
Canary Deployment



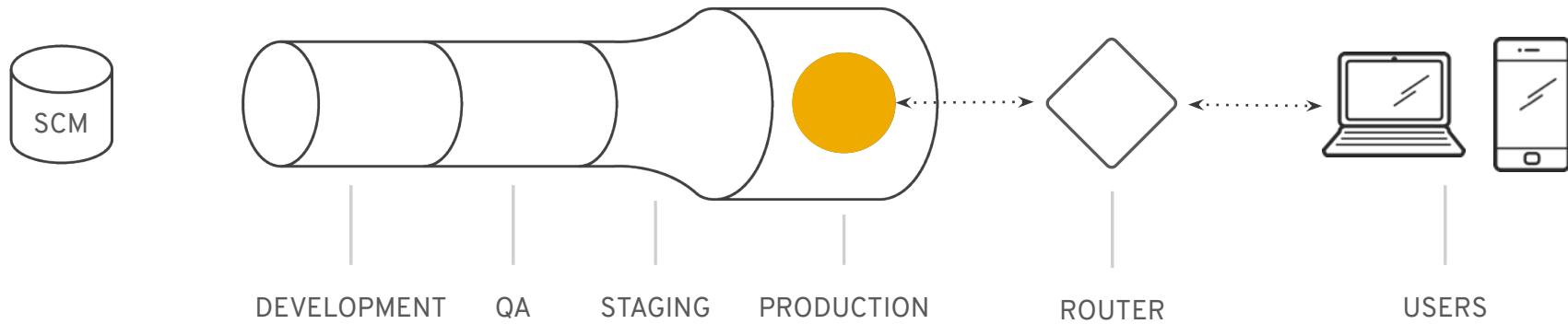
Canary Deployment



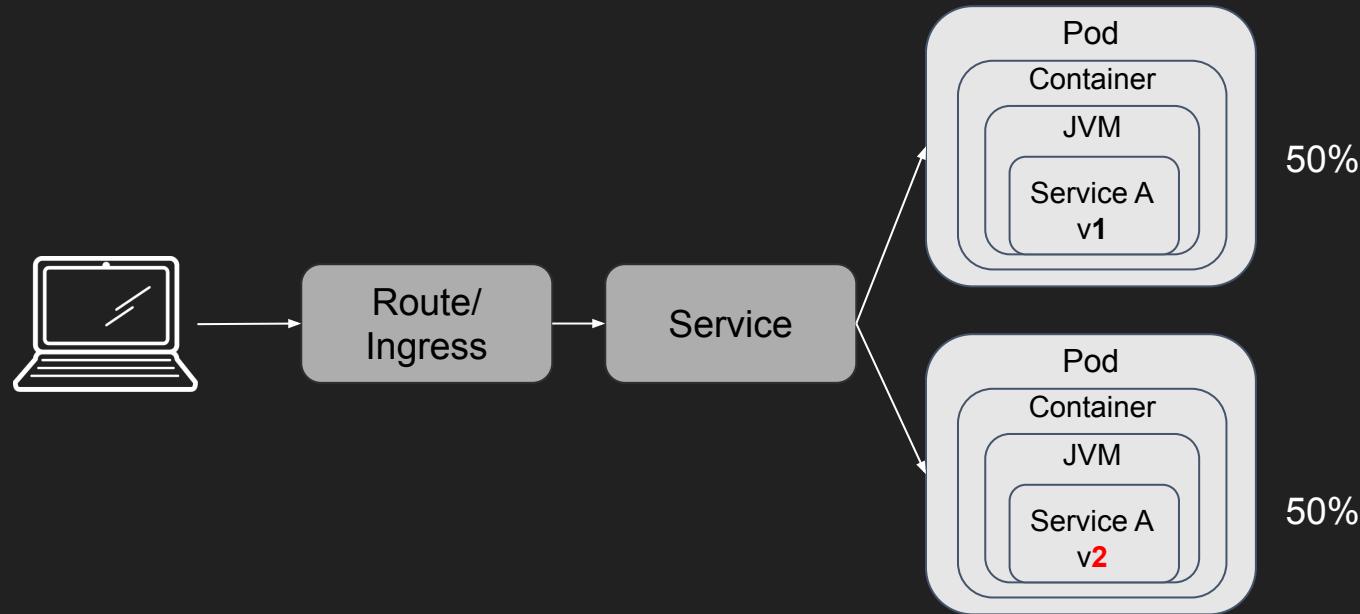
Canary Deployment



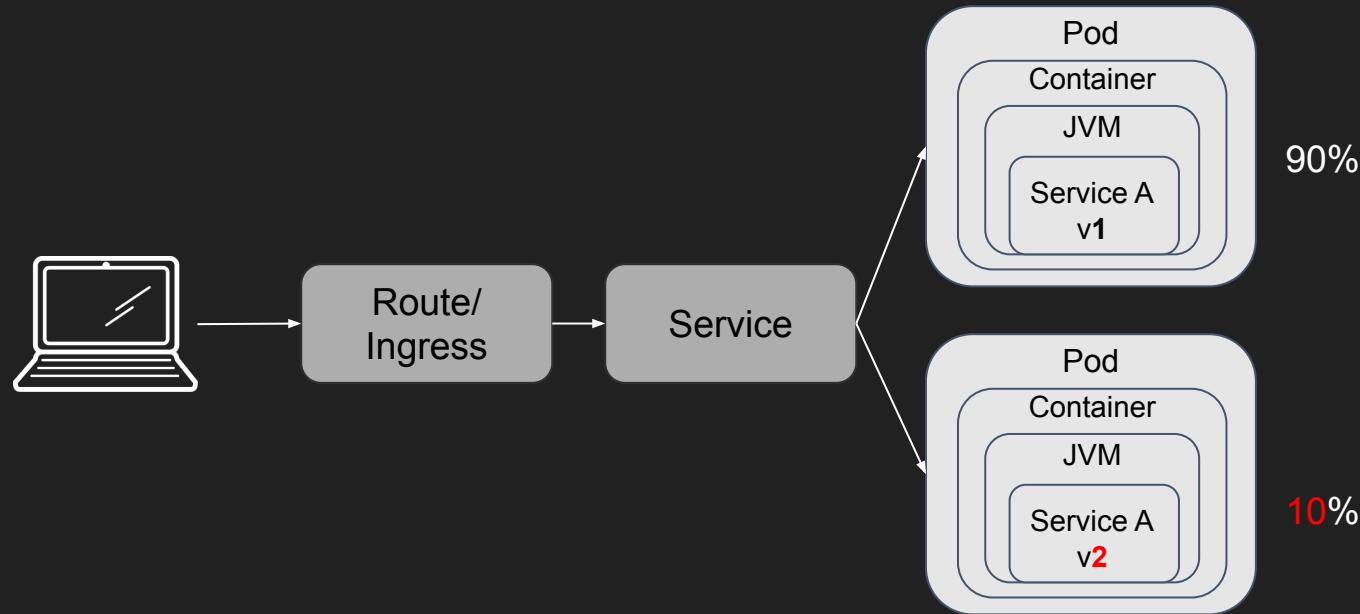
Canary Deployment



Canaries with Kubernetes



Canaries with Istio



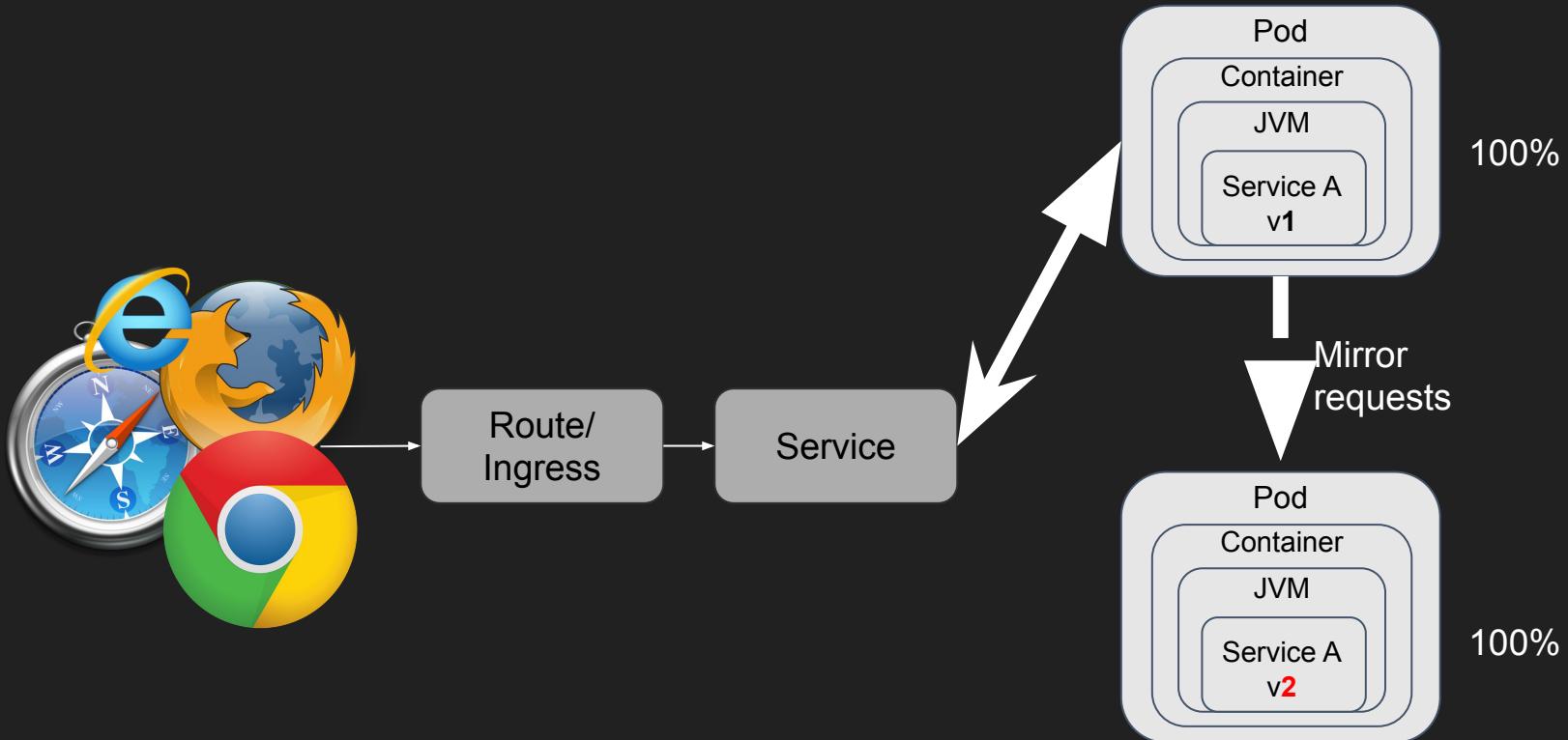
Demo Canary

- 90/10
- 75/25
- Based on User-Agent



Dark Launch

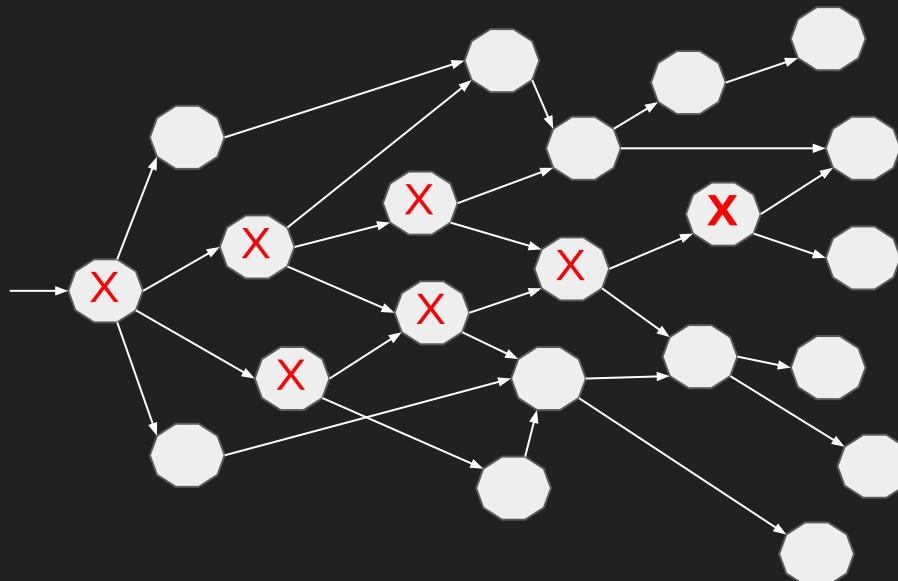
Dark Launches with Istio



Demo Dark Launch

Service Resiliency

- Retry
- Kiali



Chaos Testing

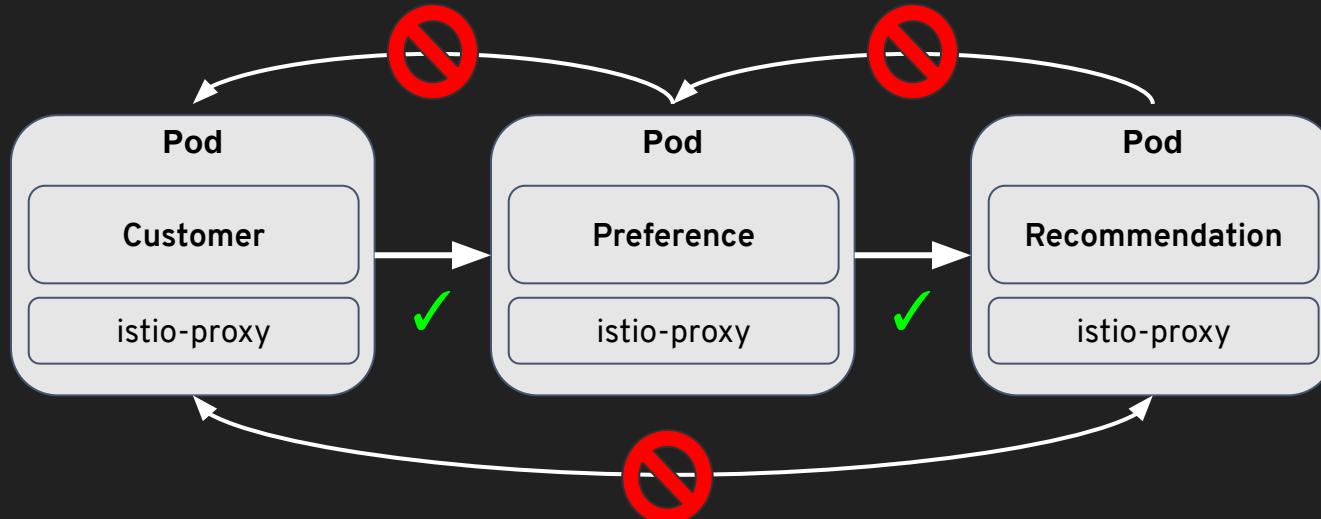


By Netflix - <https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png>, Apache License 2.0,
<https://commons.wikimedia.org/w/index.php?curid=63503083>

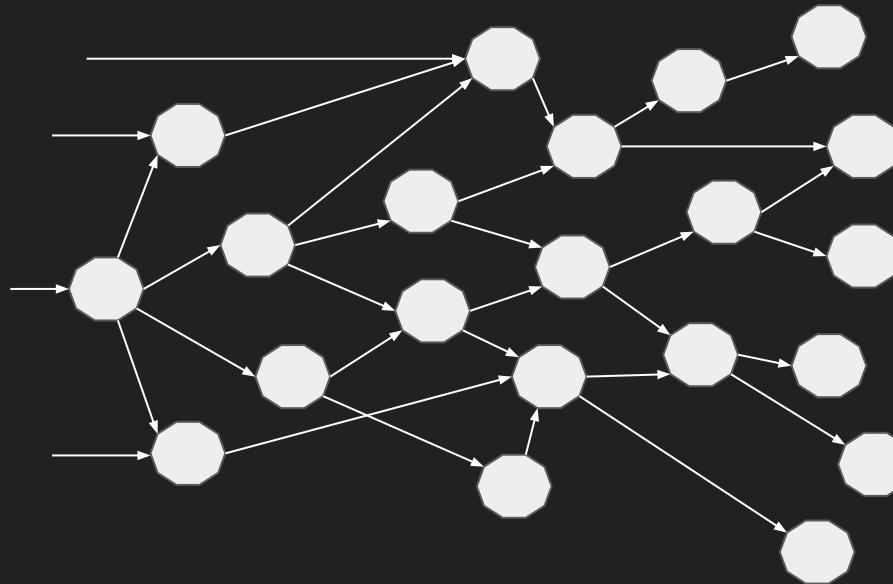
Demo Caos

- 503
- Delay

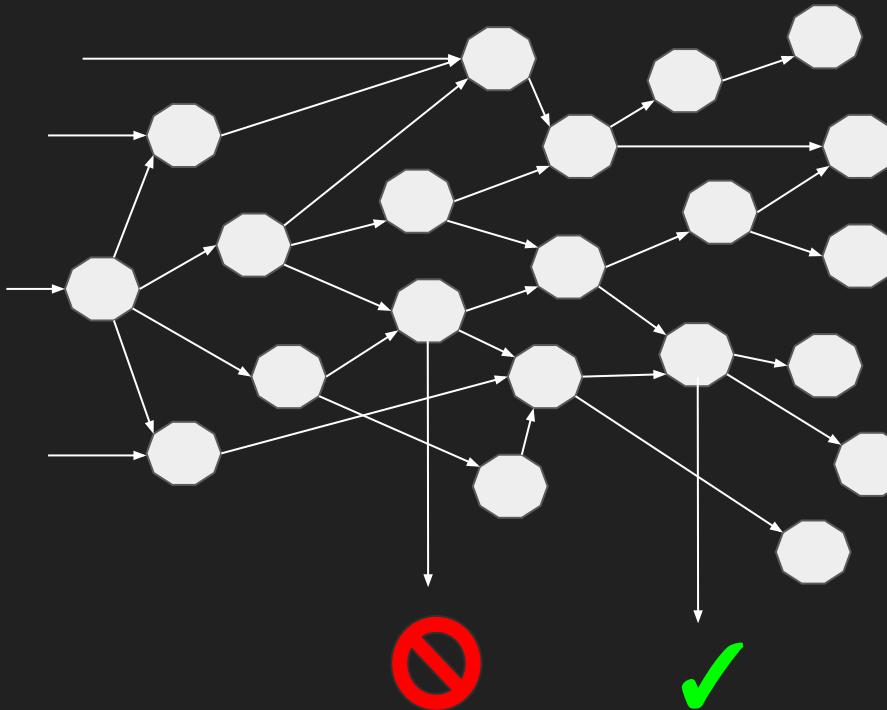
Access Control



Most Communication Inbound & Internal



Outbound/Egress Blocked By Default



Demo Egress

- Access <http://worldclockapi.com>

O'REILLY®



Introducing Istio Service Mesh for Microservices

Build and Deploy Resilient, Fault-Tolerant Cloud-Native Applications

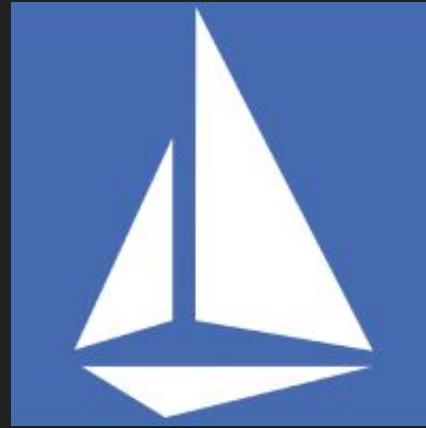


Christian Posta & Burr Sutter

bit.ly/istio-book

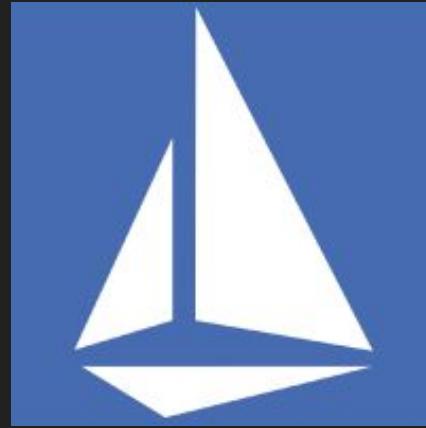
<https://learn.openshift.com/servicemesh>

Istio 1.0.x workshop: Istio Introduction START SCENARIO	Istio 1.0.x workshop: Deploy microservices START SCENARIO	Istio 1.0.x workshop: Monitoring and Tracing START SCENARIO
Istio 1.0.x workshop: Simple Routing START SCENARIO	Istio 1.0.x workshop: Advanced RouteRules START SCENARIO	Istio 1.0.x workshop: Fault Injection START SCENARIO
Istio 1.0.x workshop: Circuit Breaker START SCENARIO	Istio 1.0.x workshop: Egress START SCENARIO	Istio 1.0.x Advanced: Observing with Kiali START SCENARIO



Demo

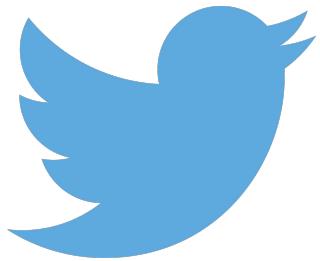
bit.ly/istio-tutorial



Workshop

bit.ly/the-istio-workshop

The End
(but Serverless is coming)



@RAFABENE

