RED HAT® DEVELOPER

Istio on Kubernetes: Enter the Service Mesh

benevides@redhat.com

@rafabene

Link → http://bit.ly/istio-kubernetes

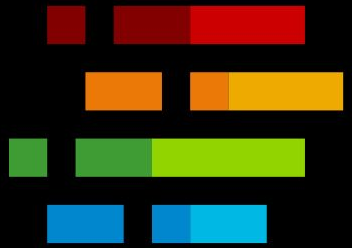
Rafael Benevides

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

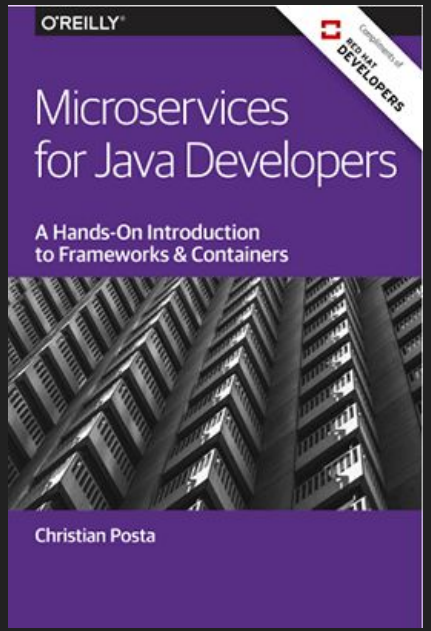


RED HAT® DEVELOPER

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

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

bit.ly/javamicroservicesbook

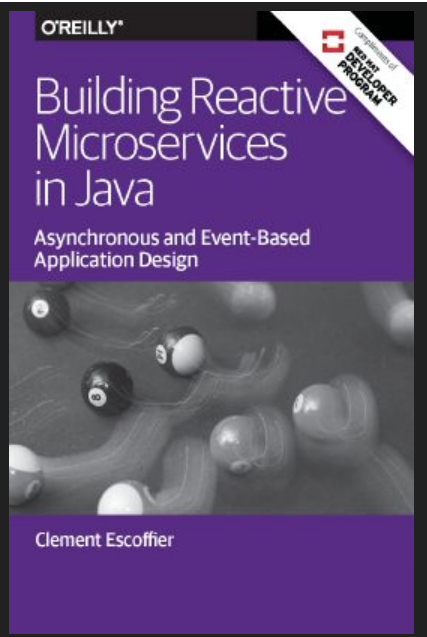


Free eBooks from developers.redhat.com

Microservices Introductory Materials

Demo: bit.ly/msa-instructions
Slides: bit.ly/microservicesdeepdive
Video Training: bit.ly/microservicesvideo
Kubernetes for Java Developers

bit.ly/reactivemicroservicesbook



Advanced Materials

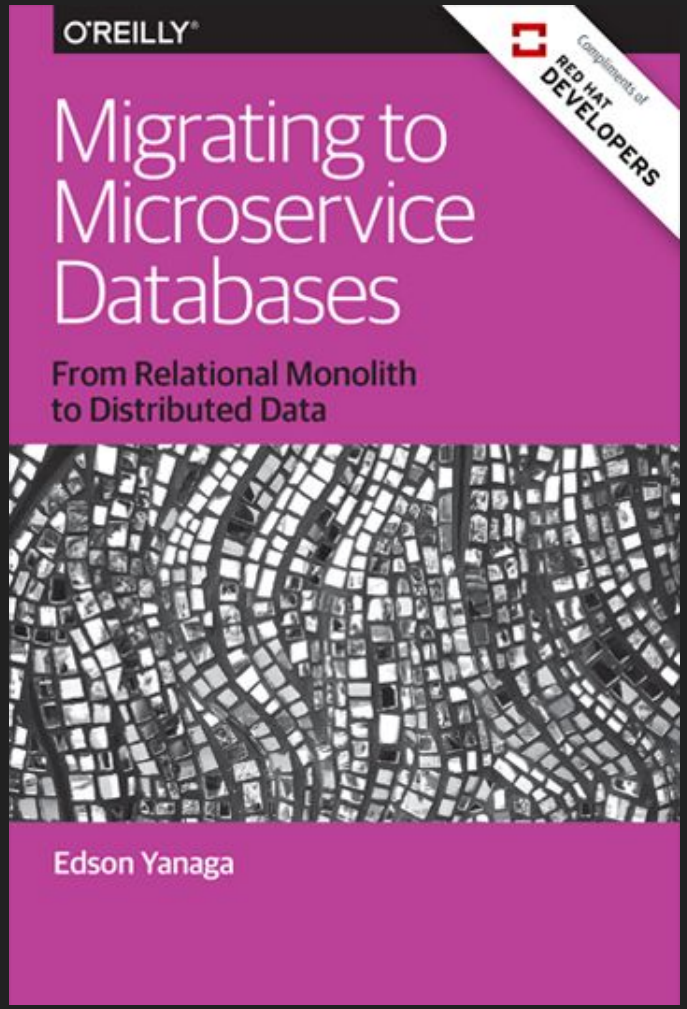
bit.ly/istio-tutorial
learn.openshift.com/servicemesh

bit.ly/faas-tutorial
learn.openshift.com/serverless

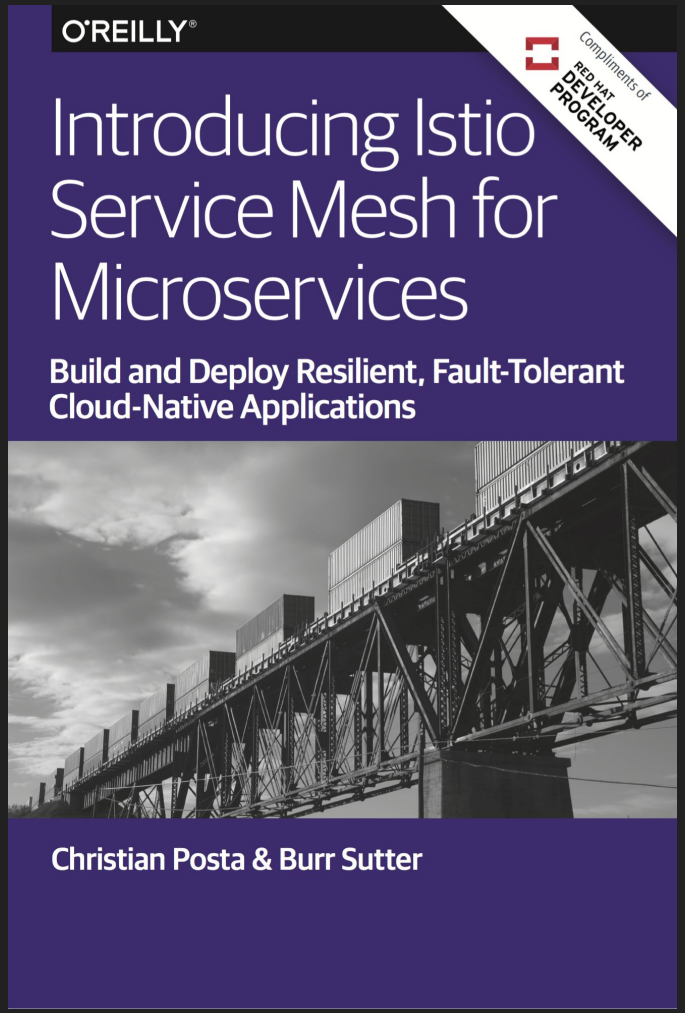
http://bit.ly/istio-kubernetes

@rafabene

RED HAT DEVELOPER



bit.ly/mono2microdb



bit.ly/istio-book

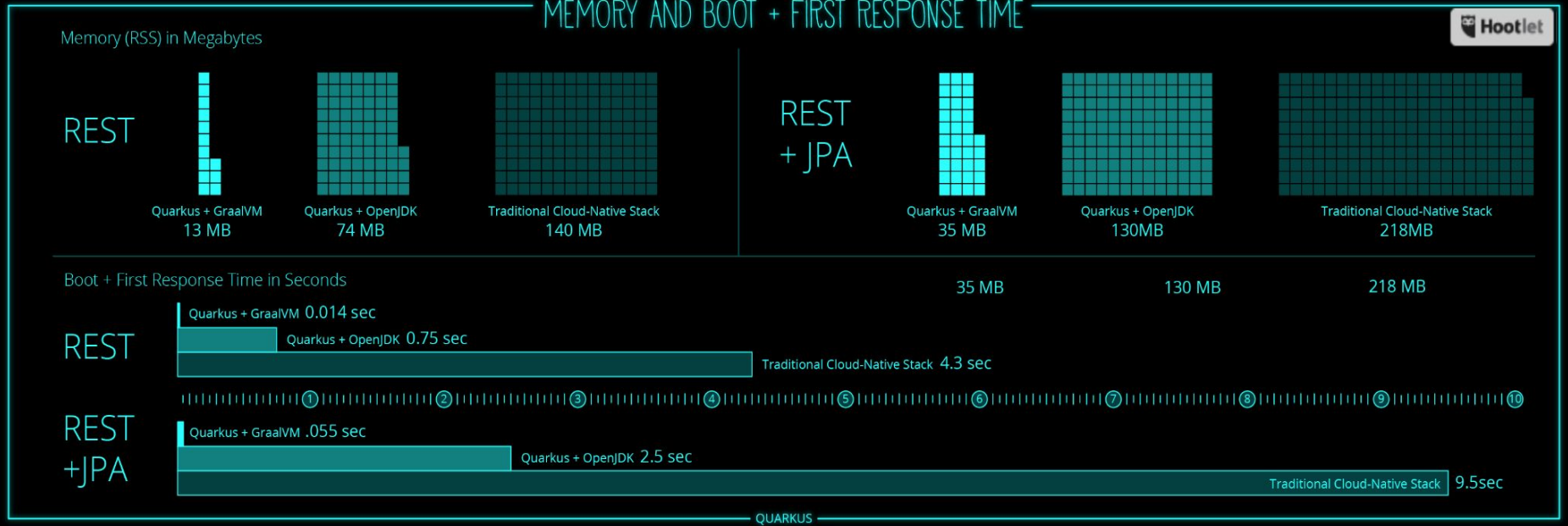
MicroProfile

microprofile.io

RED HAT DEVELOPER

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)

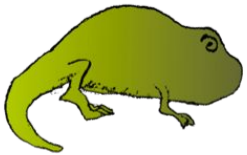
1. Follow: @rafabene 
2. With picture of the session
3. Mention @rafabene
4. With hashtag #VDBUH2019

Your Journey to Awesomeness

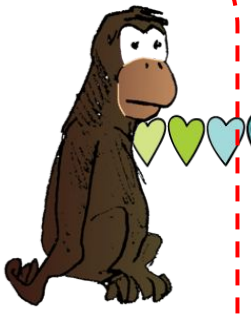
Re-Org to DevOps

Self-Service, On-Demand, Elastic Infrastructure

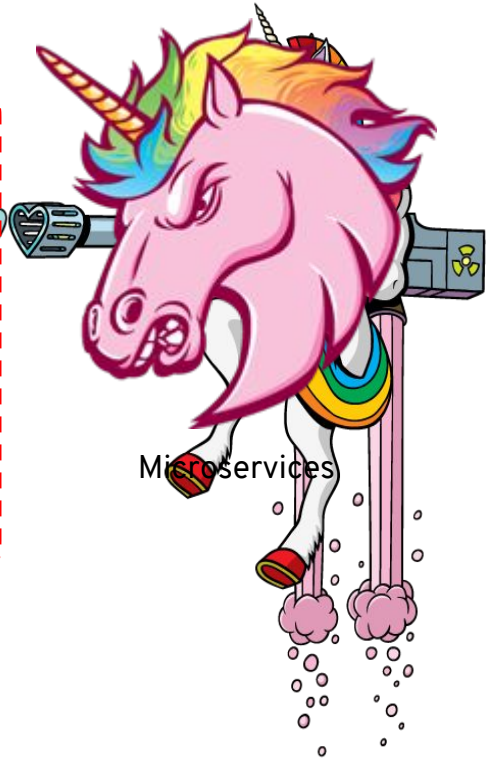
Automation



CI & CD Deployment Pipeline



Advanced Deployment Techniques

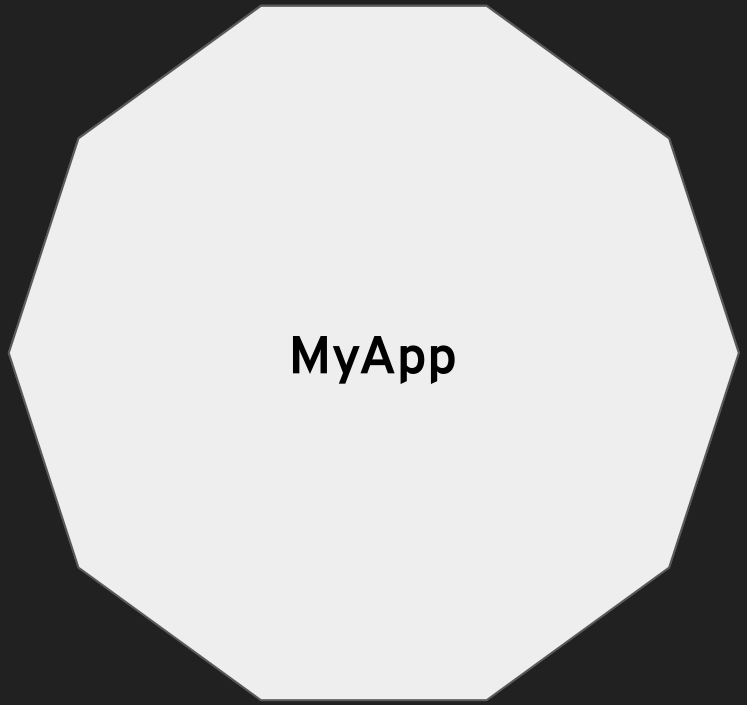


Microservices

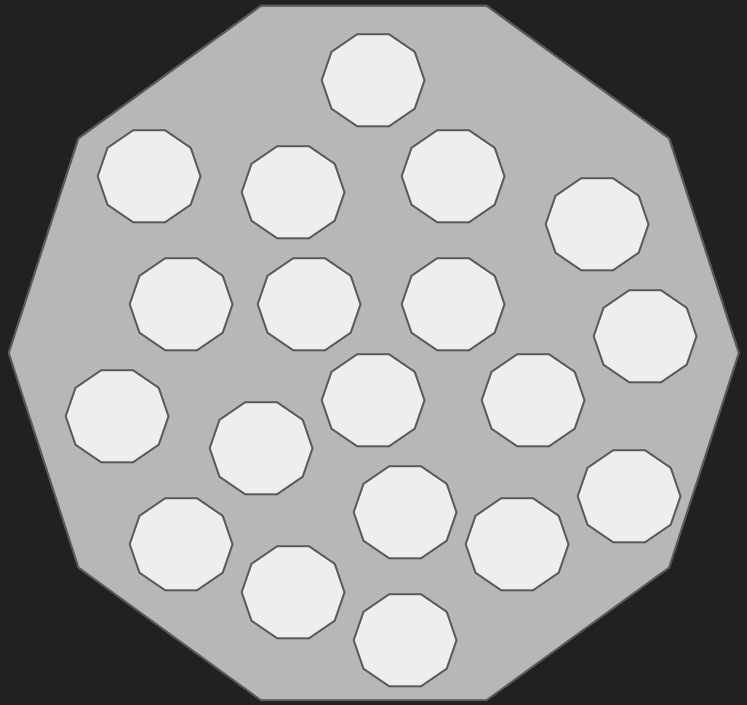
@rafabene

RED HAT DEVELOPER

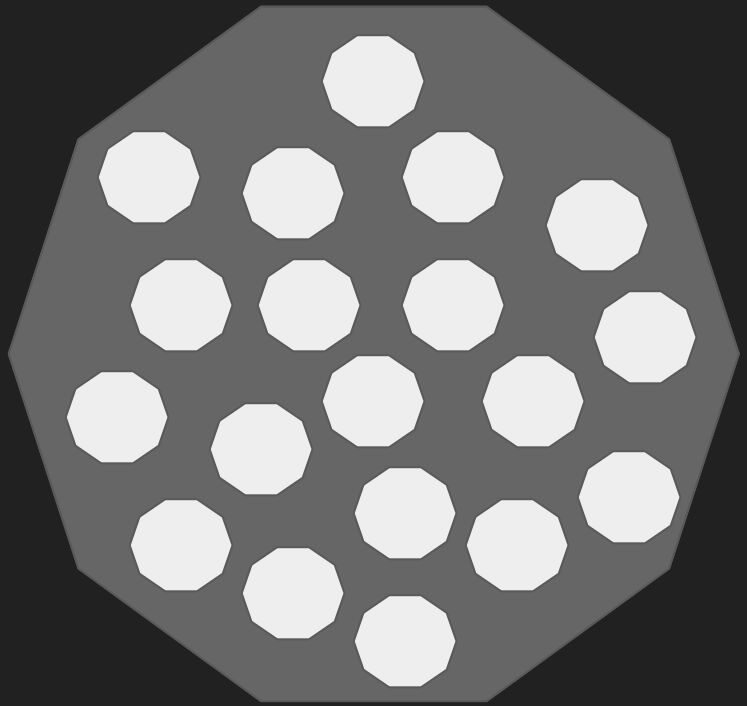
Monolith



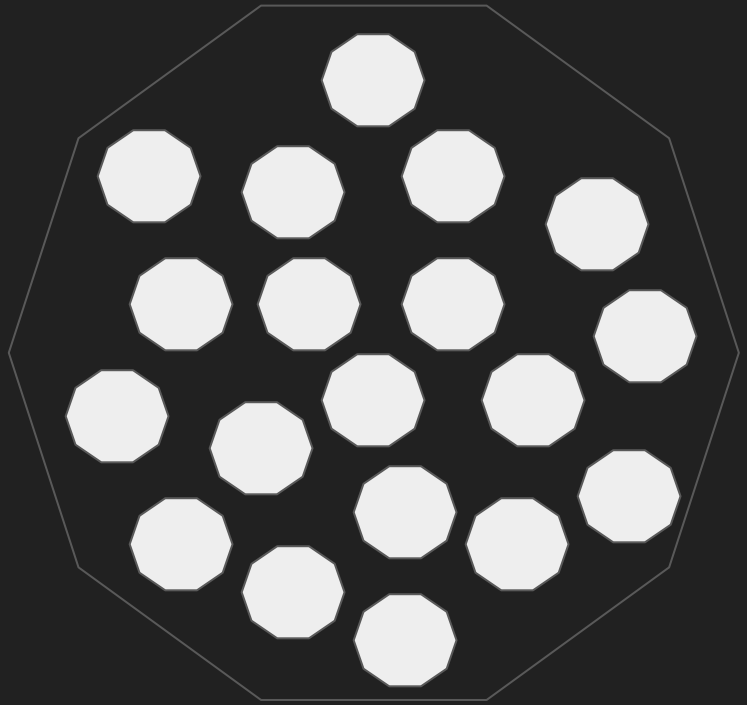
Modules



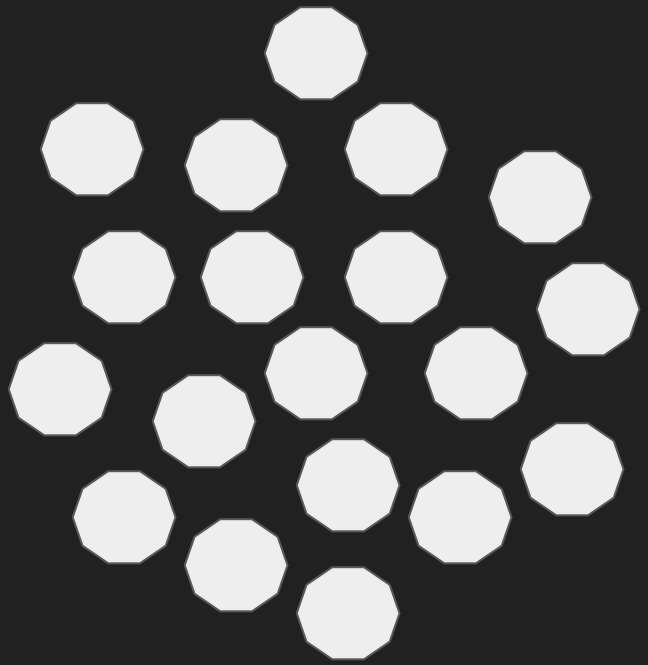
Microservices



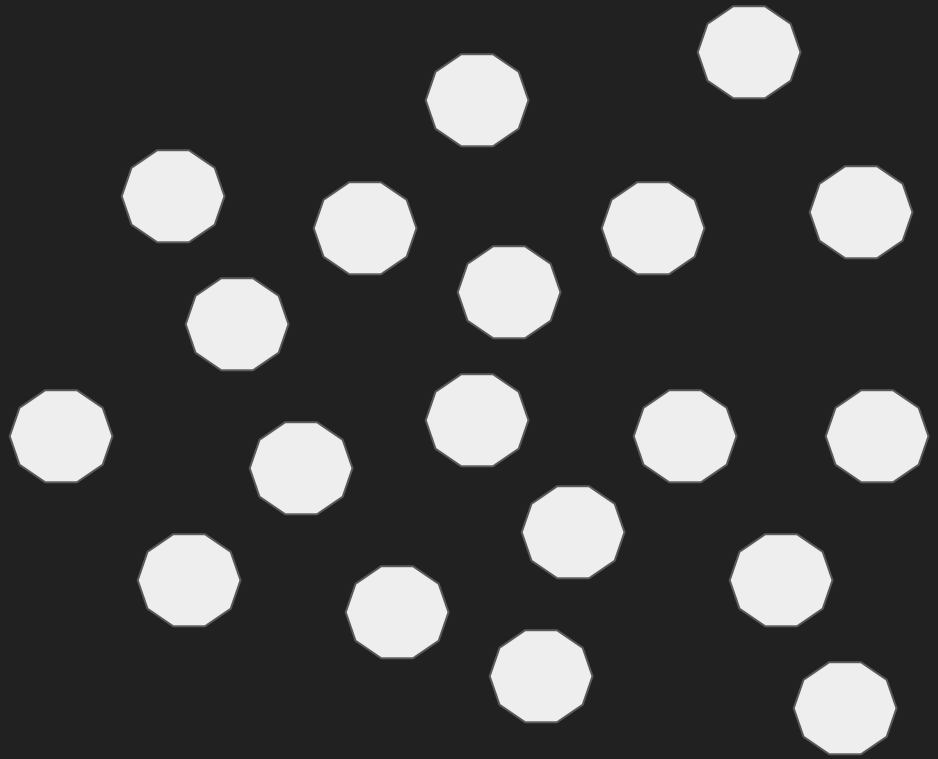
Microservices



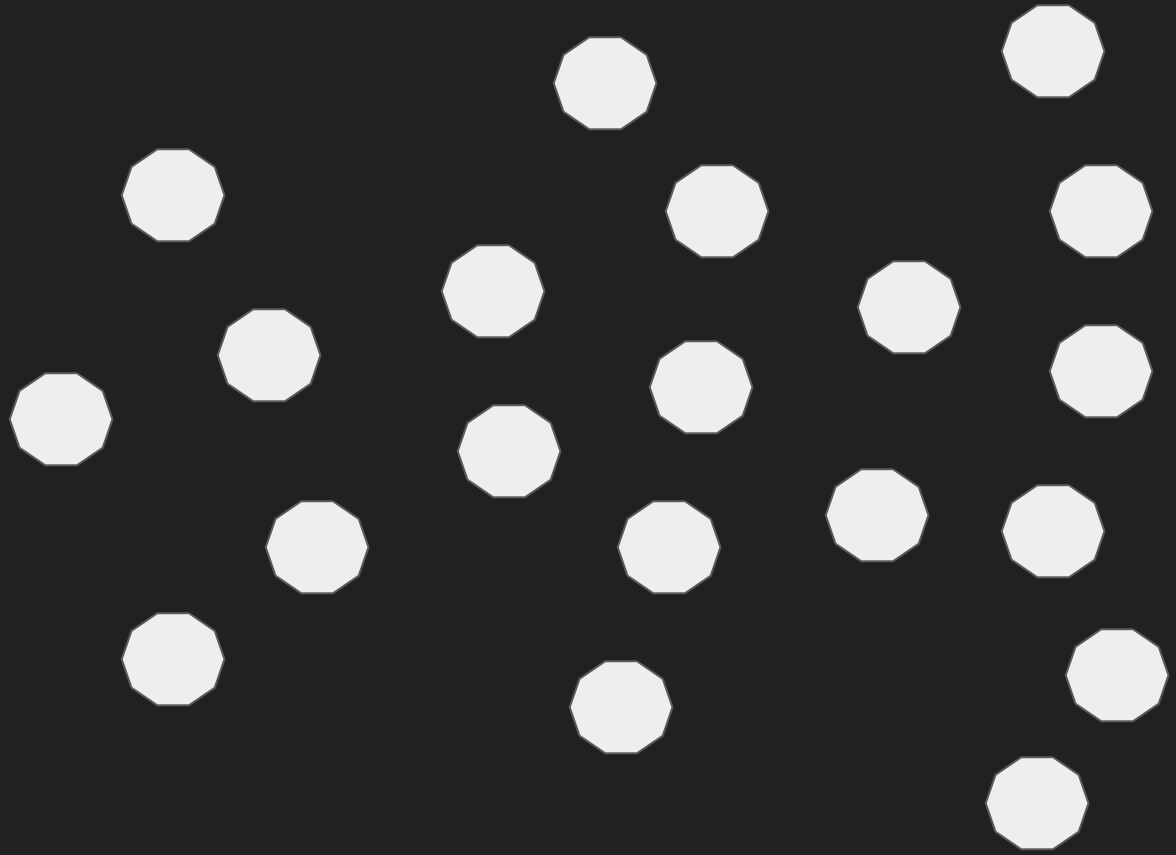
Microservices



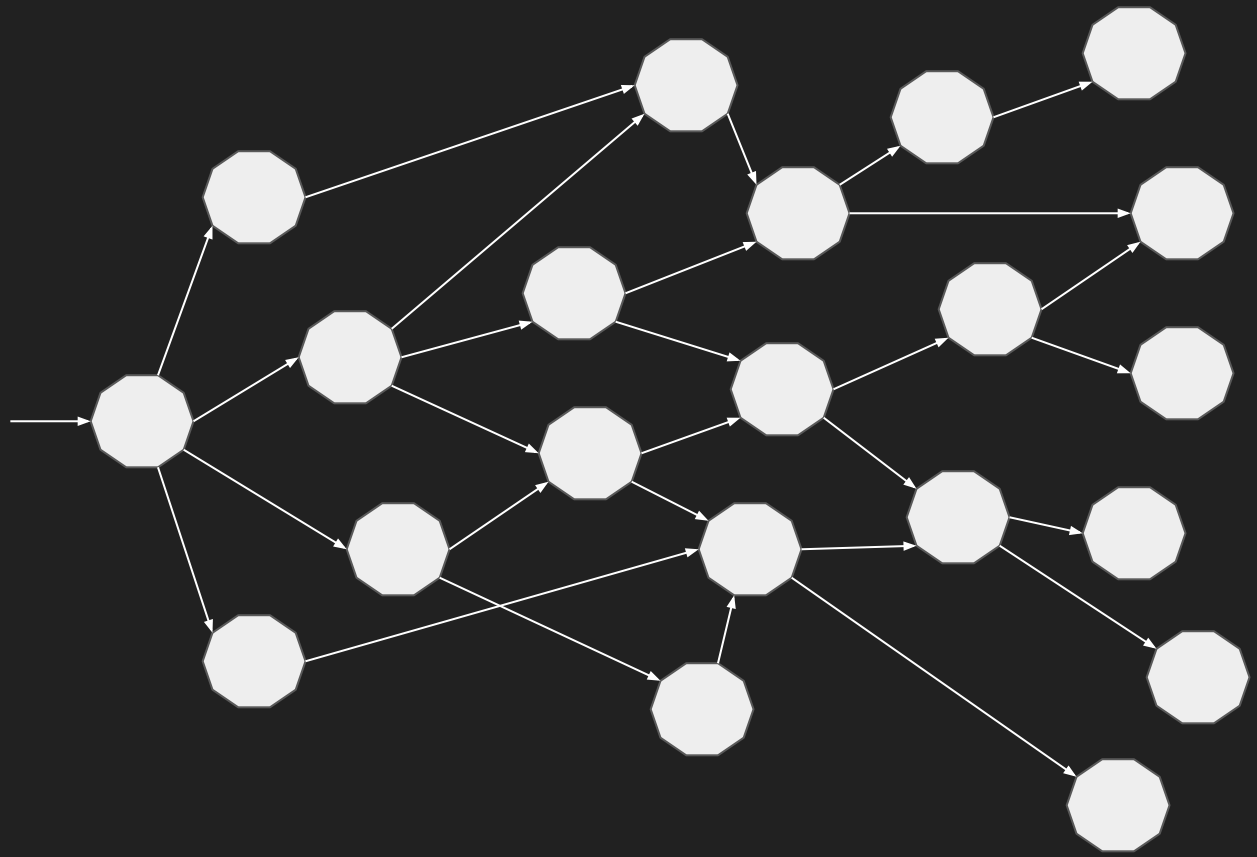
Microservices



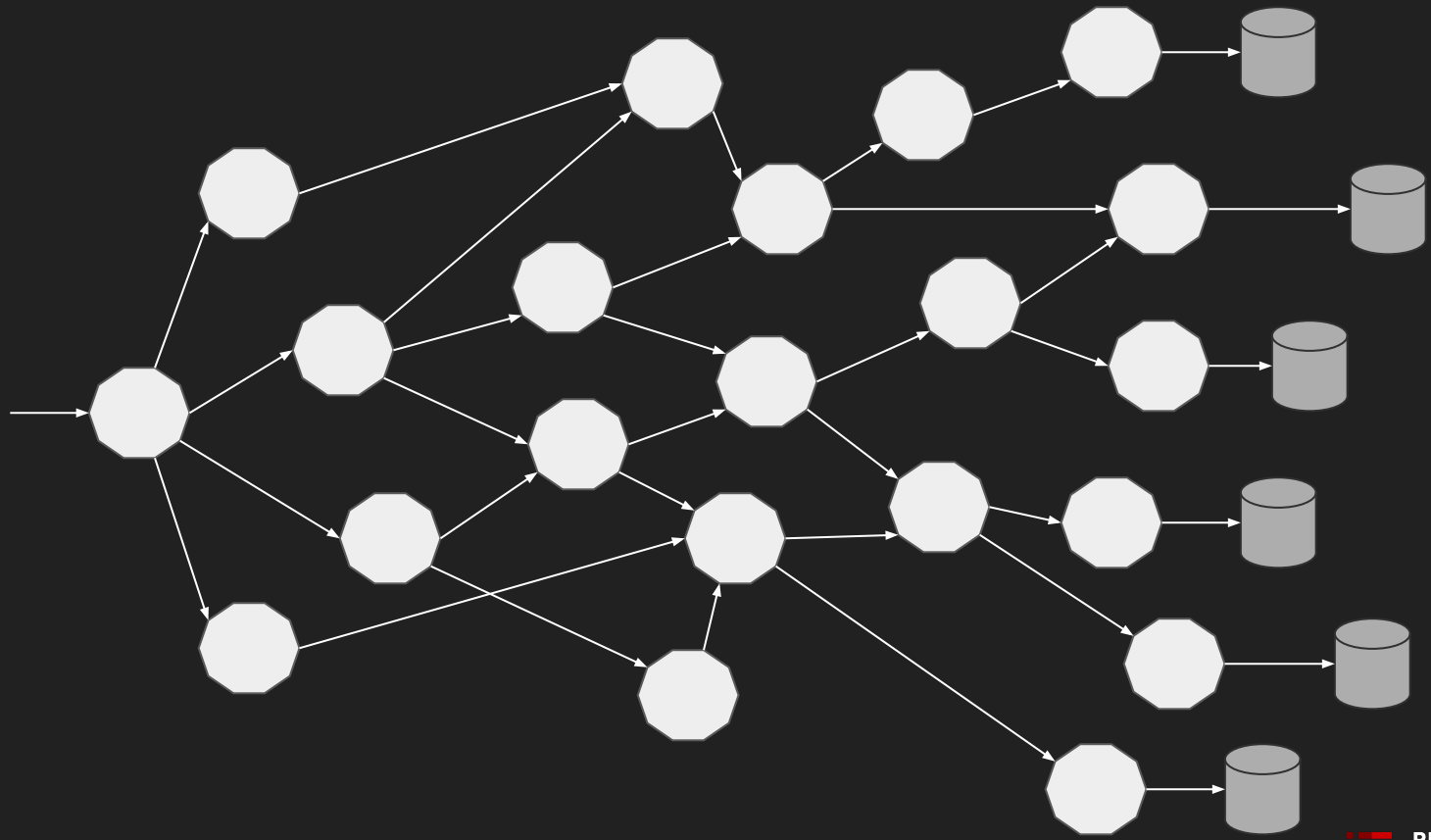
Microservices



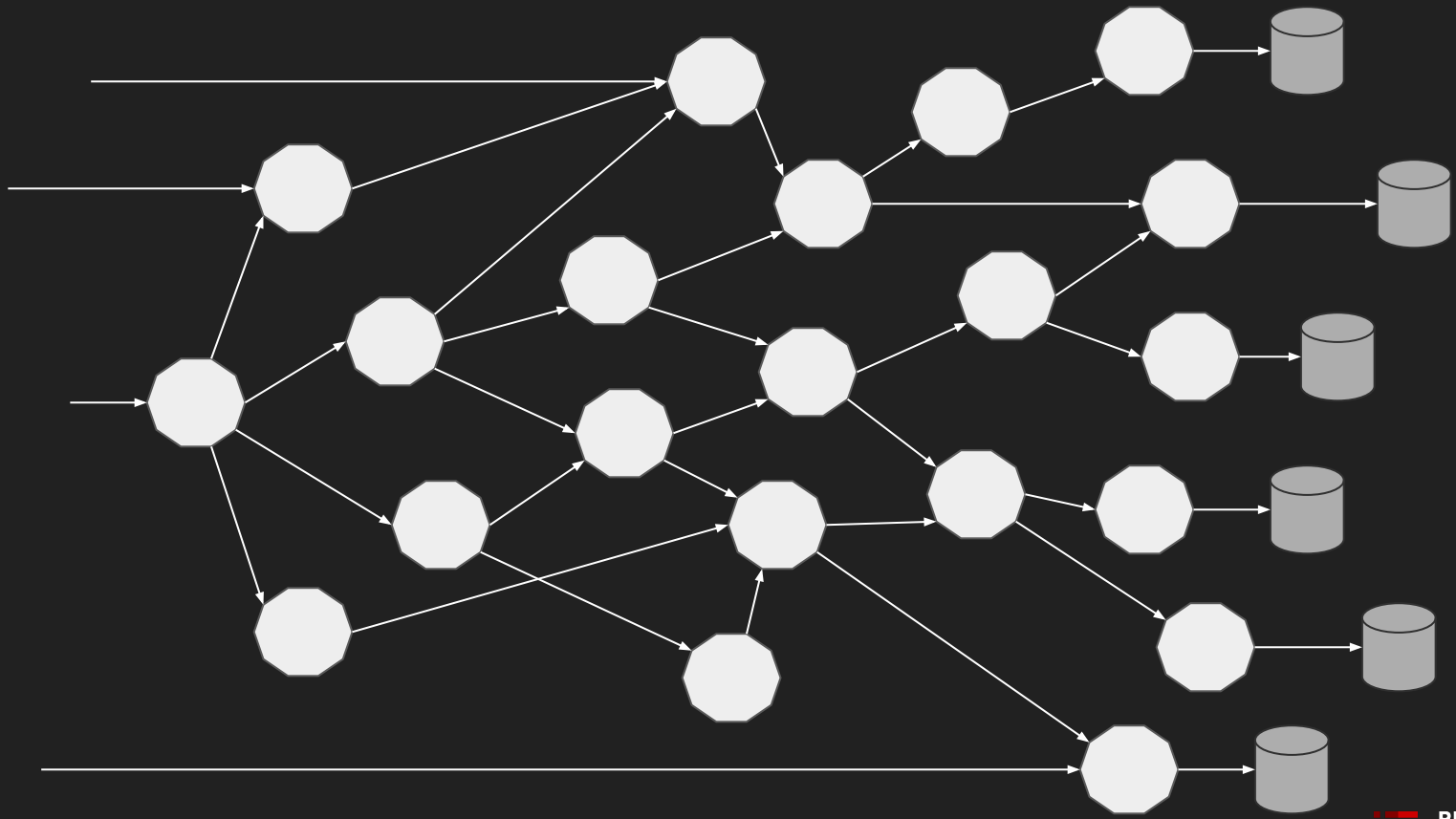
Network of Services - Mesh



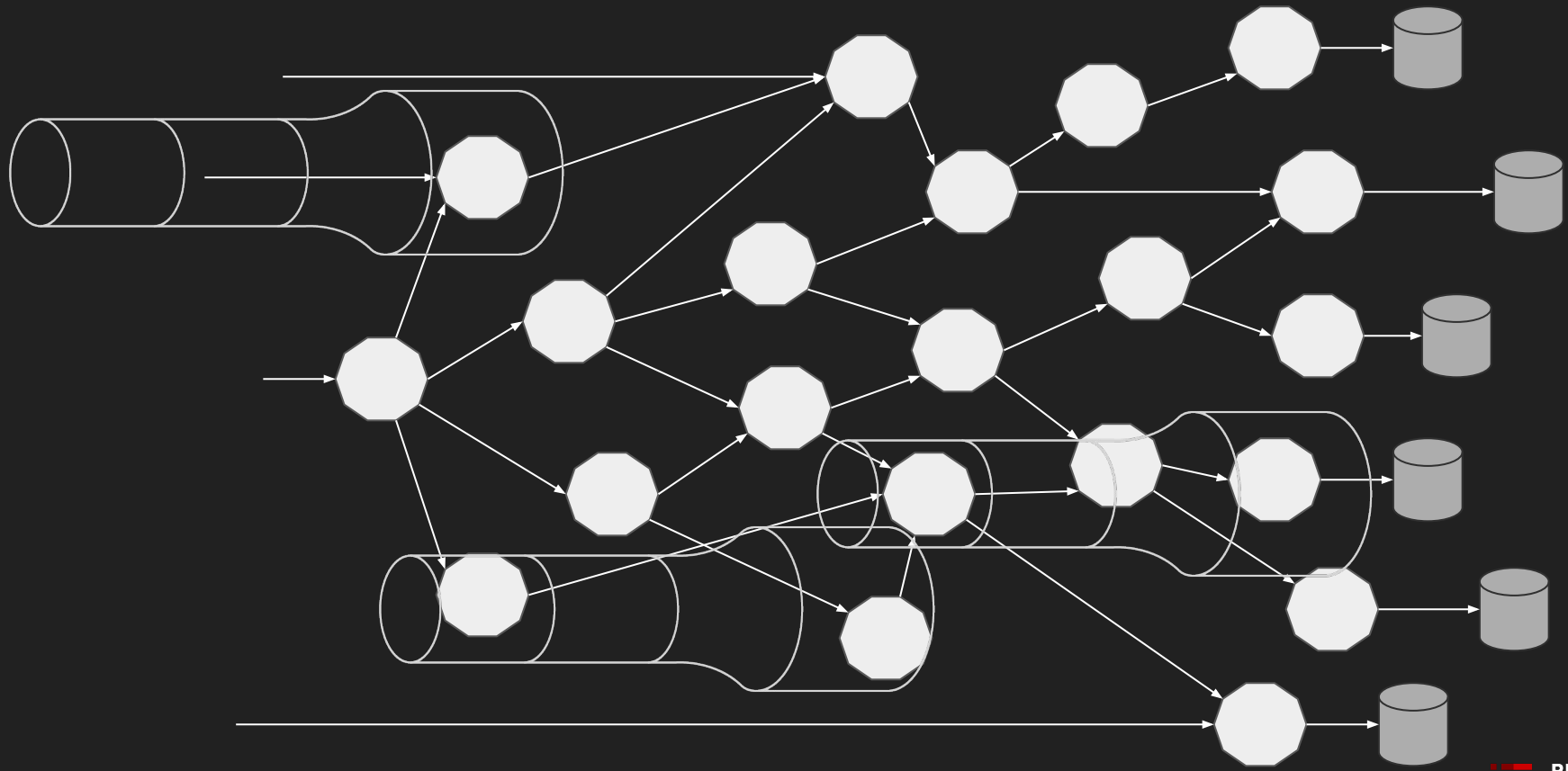
Microservices own their Data



Multiple Points of Entry

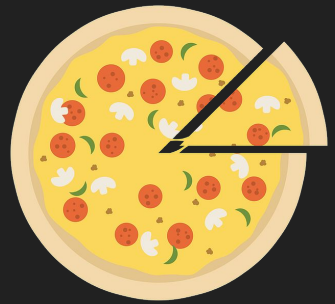


Multiple Pipelines



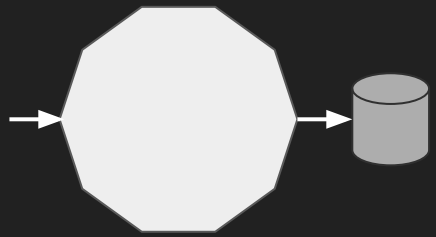
Microservices Principles

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



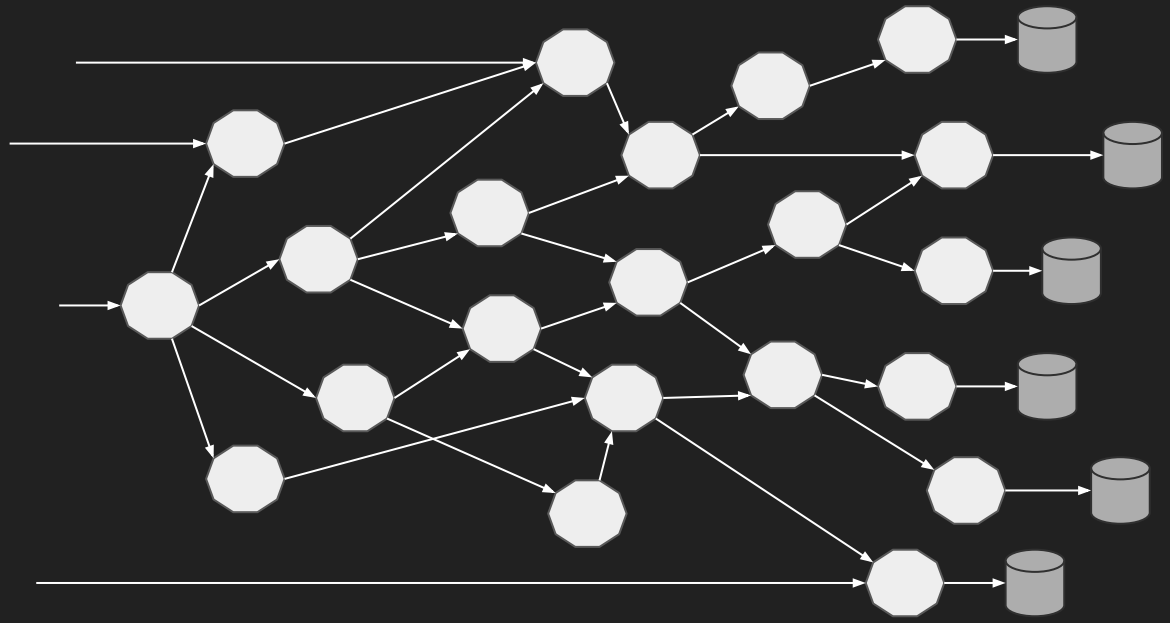
2 Pizza Team

Old School

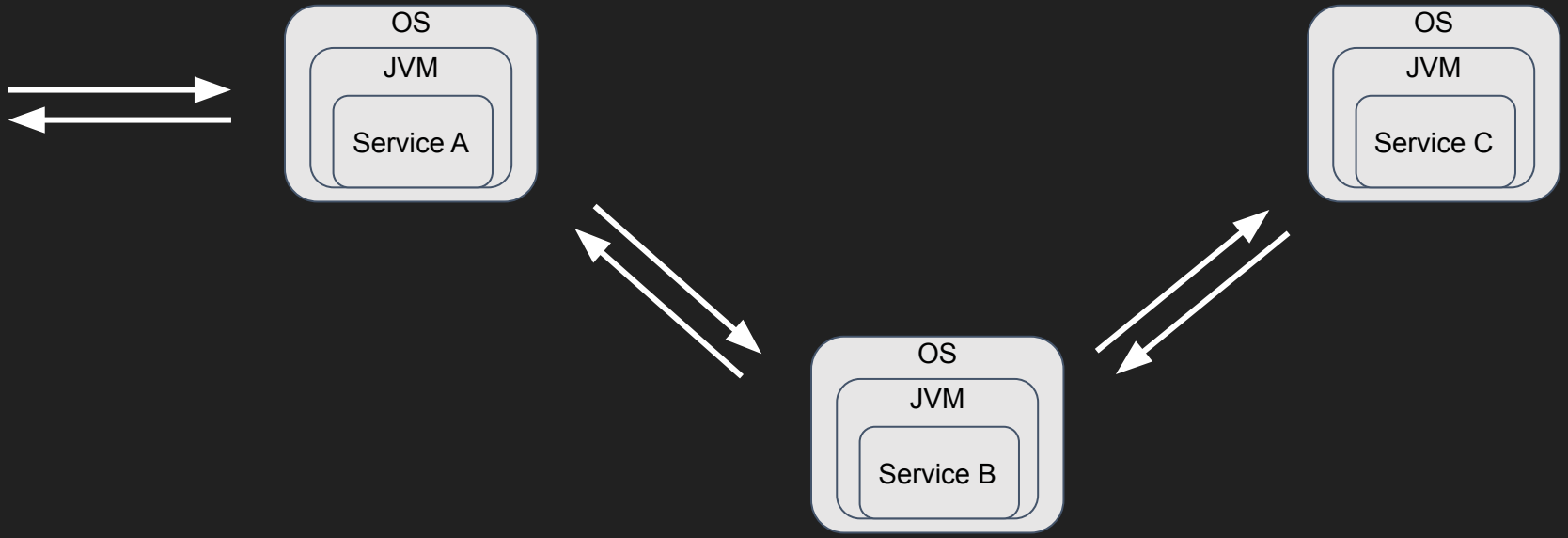


Love Thy Mono

New School



Microservices == Distributed Computing

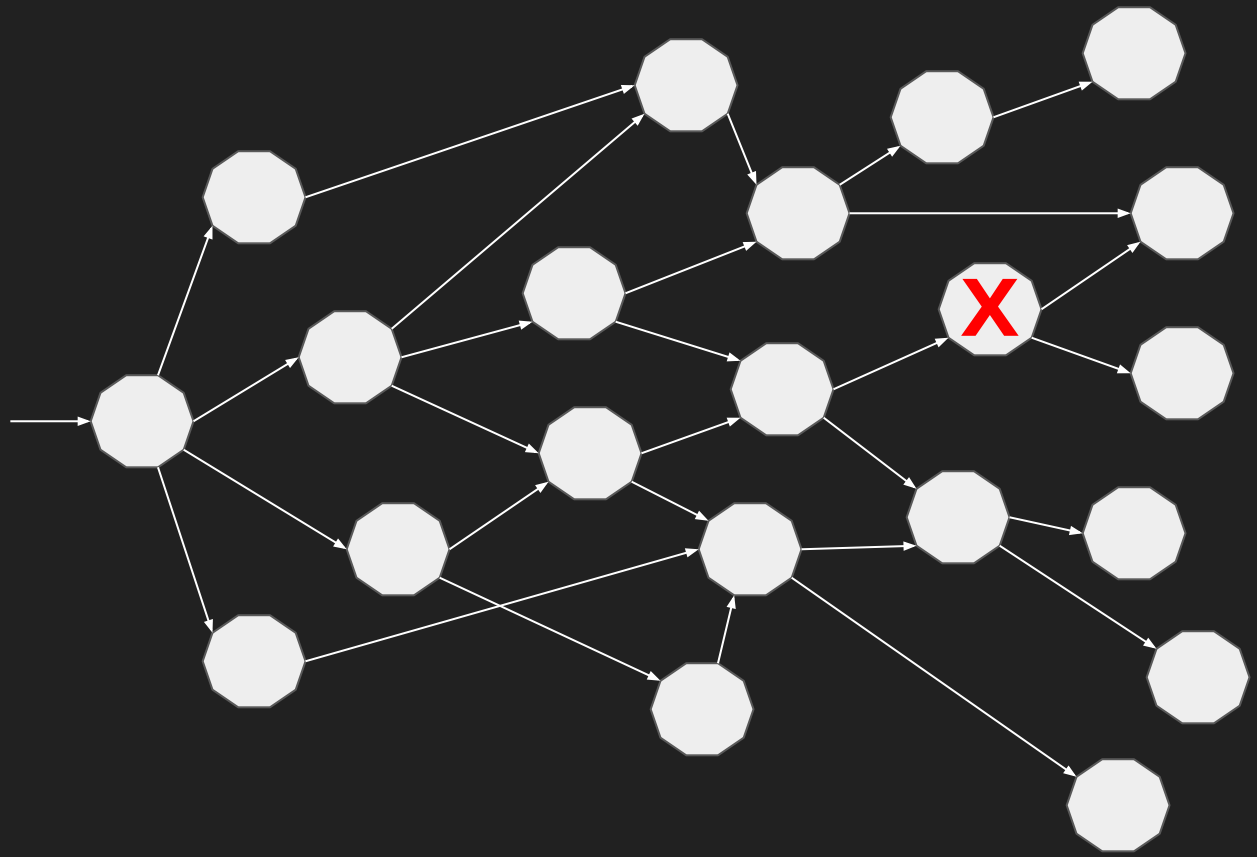


Fallacies of Distributed Computing

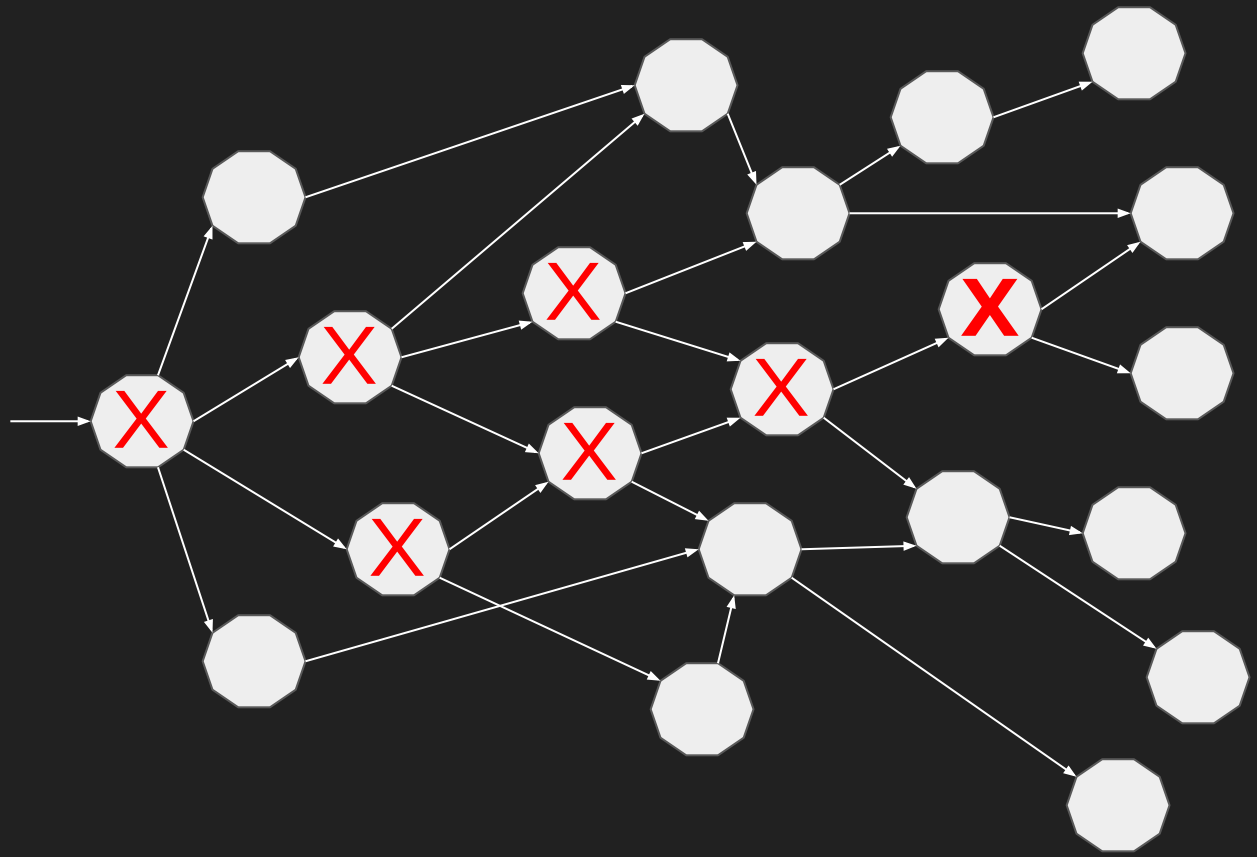
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

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

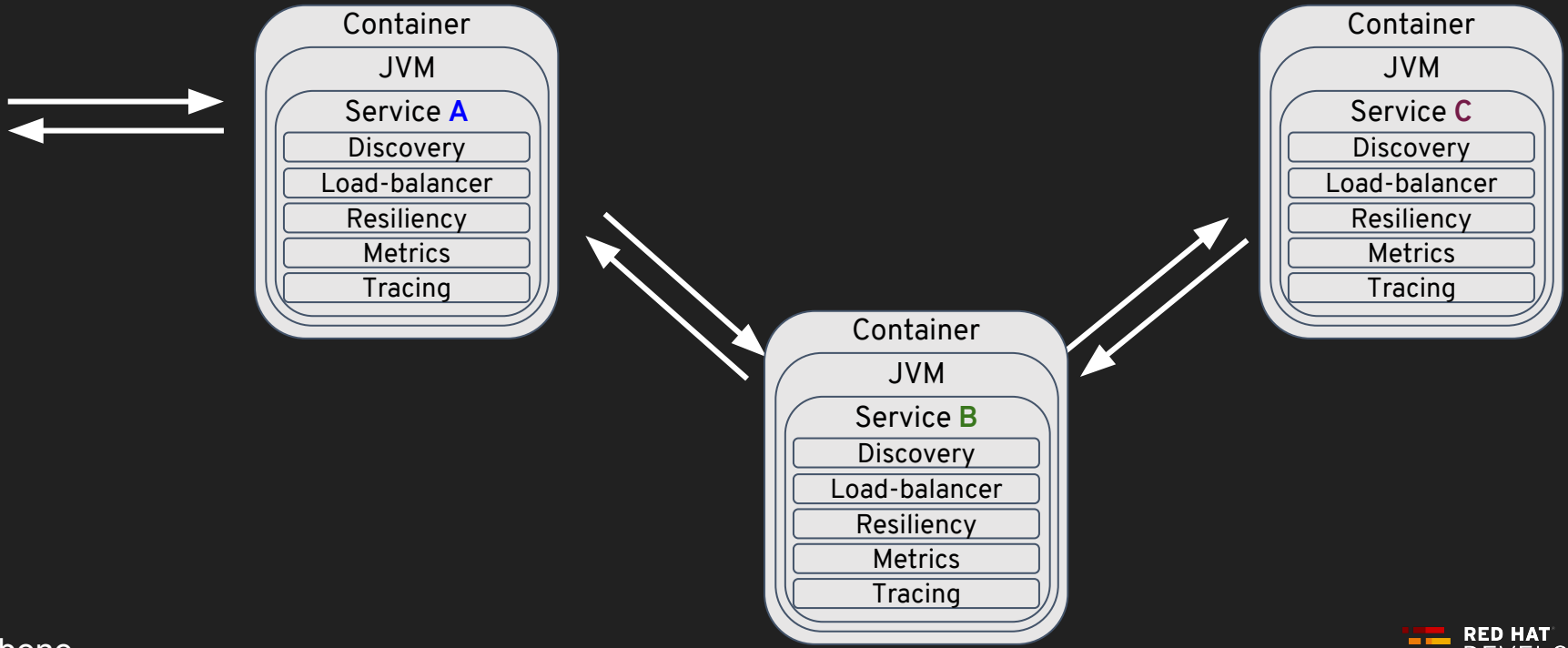
Failure of a Service



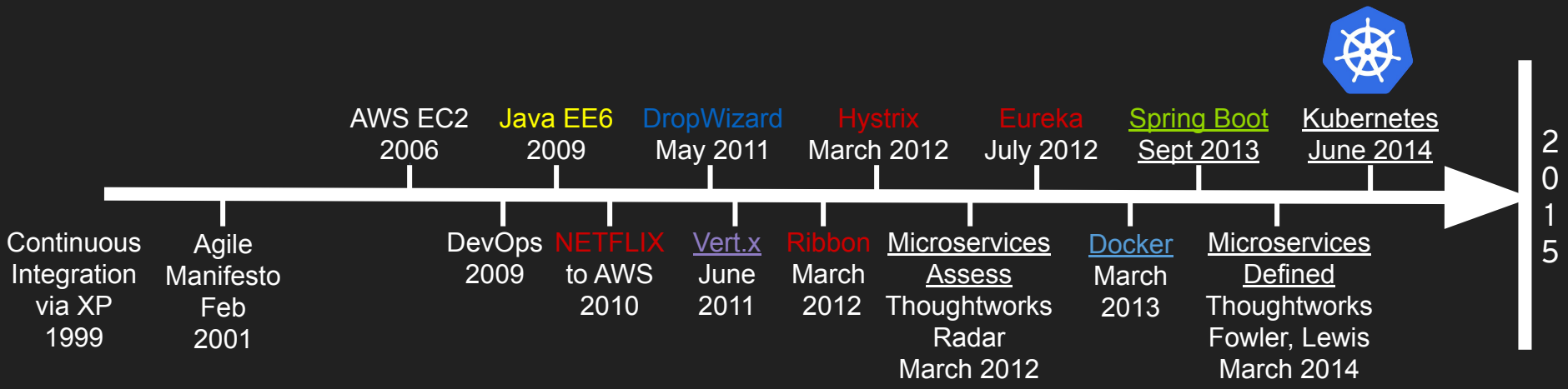
Cascading Failure



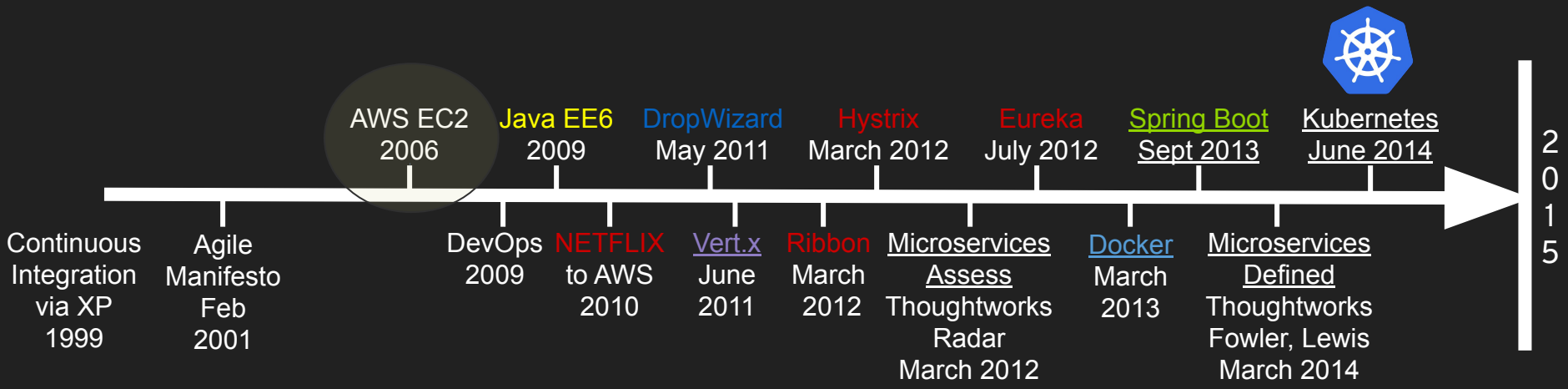
Microservices embedding Capabilities



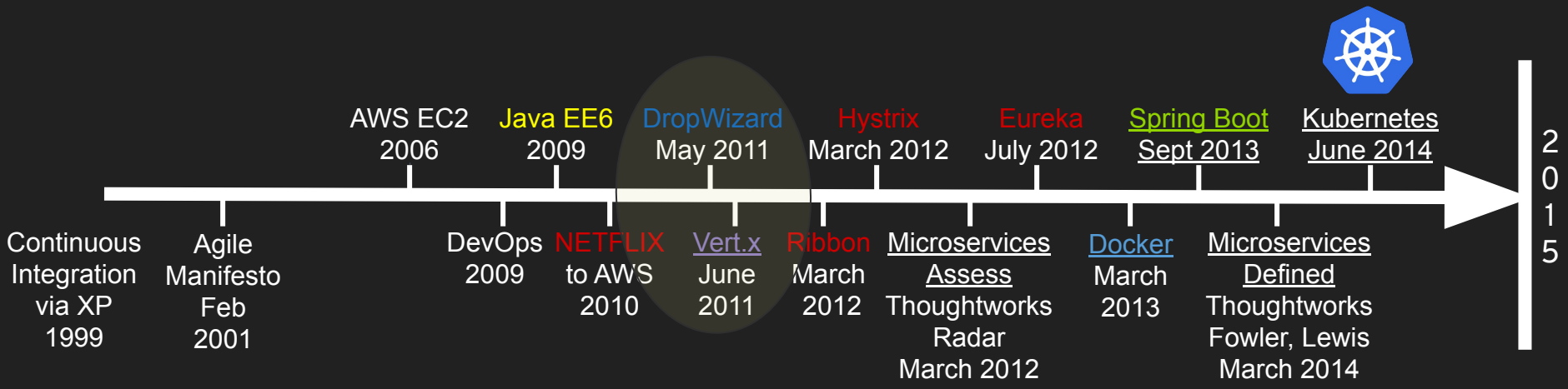
History of Microservices



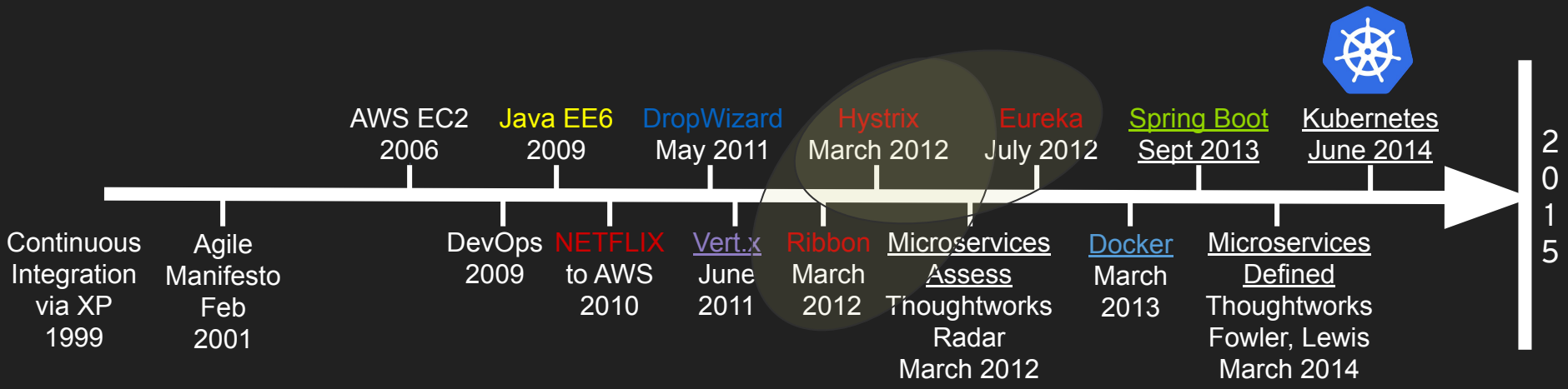
The Cloud is Born



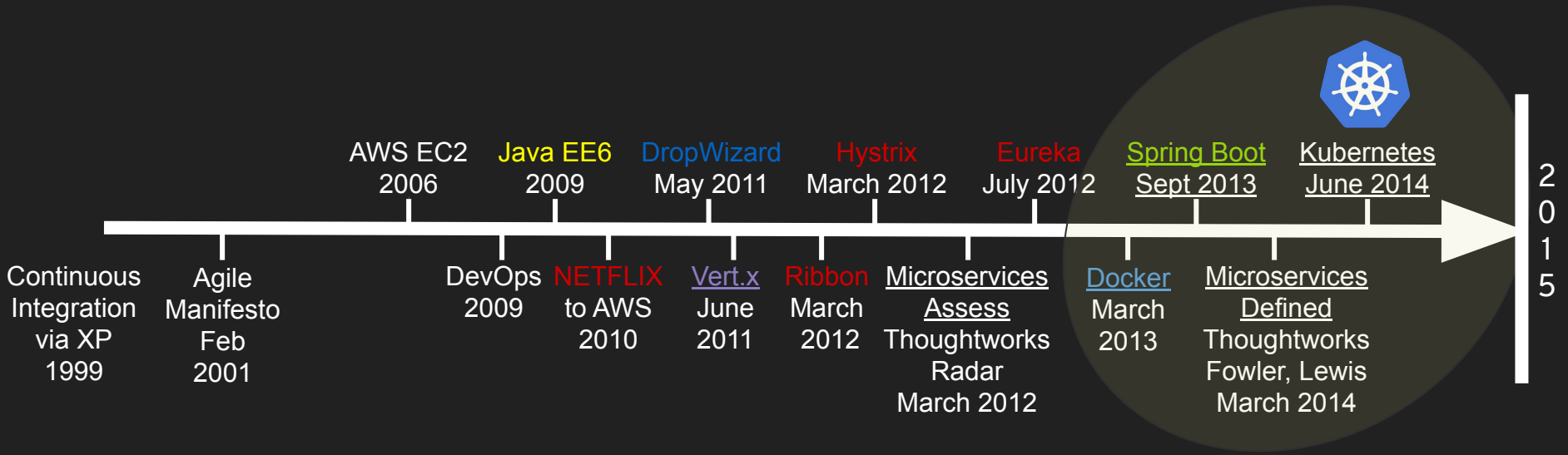
Fat Jars



Netflix goes Open Source



Perfect Storm for Microservices

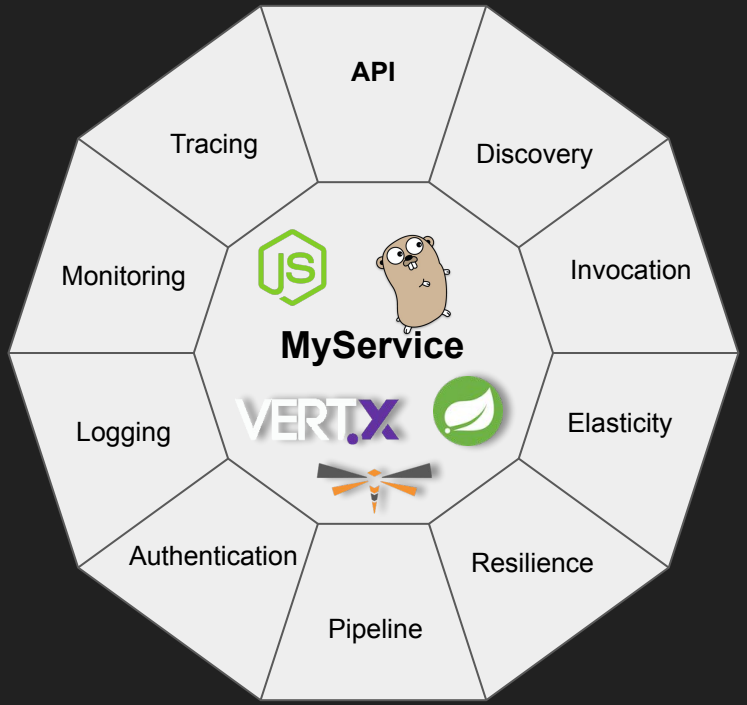


What's Wrong with Netflix OSS?

Java Only

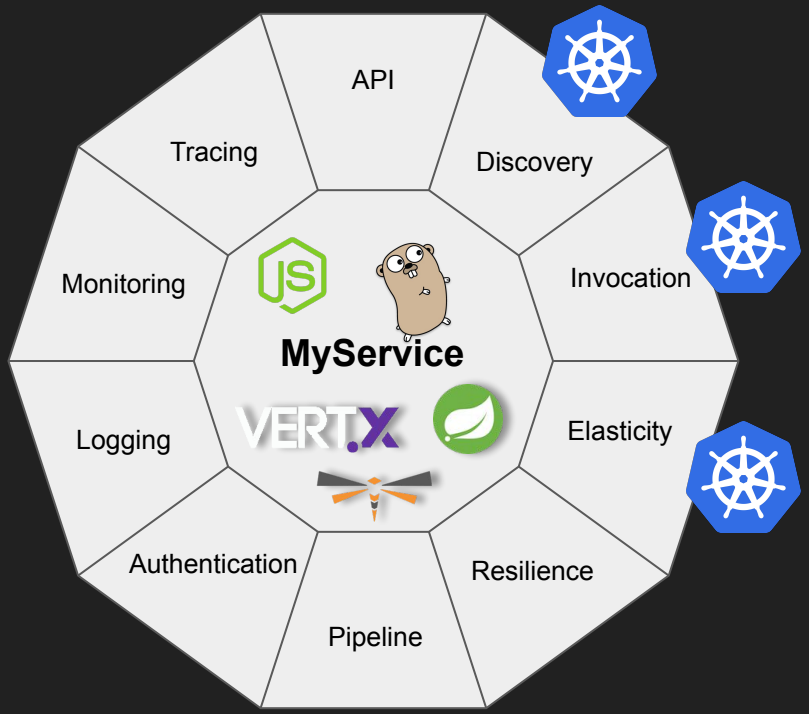
Adds a lot of libraries to YOUR code

Microservices'ilities

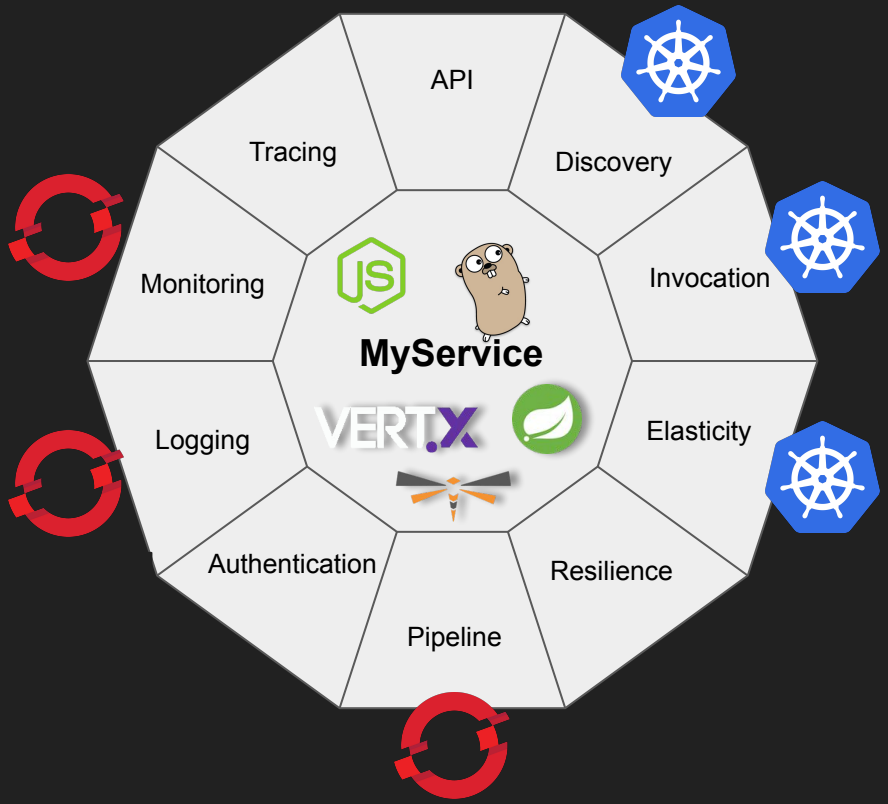


OPENSHIFT

Microservices'ilities + Kubernetes



Microservices'ilities + OpenShift



Istio - Sail

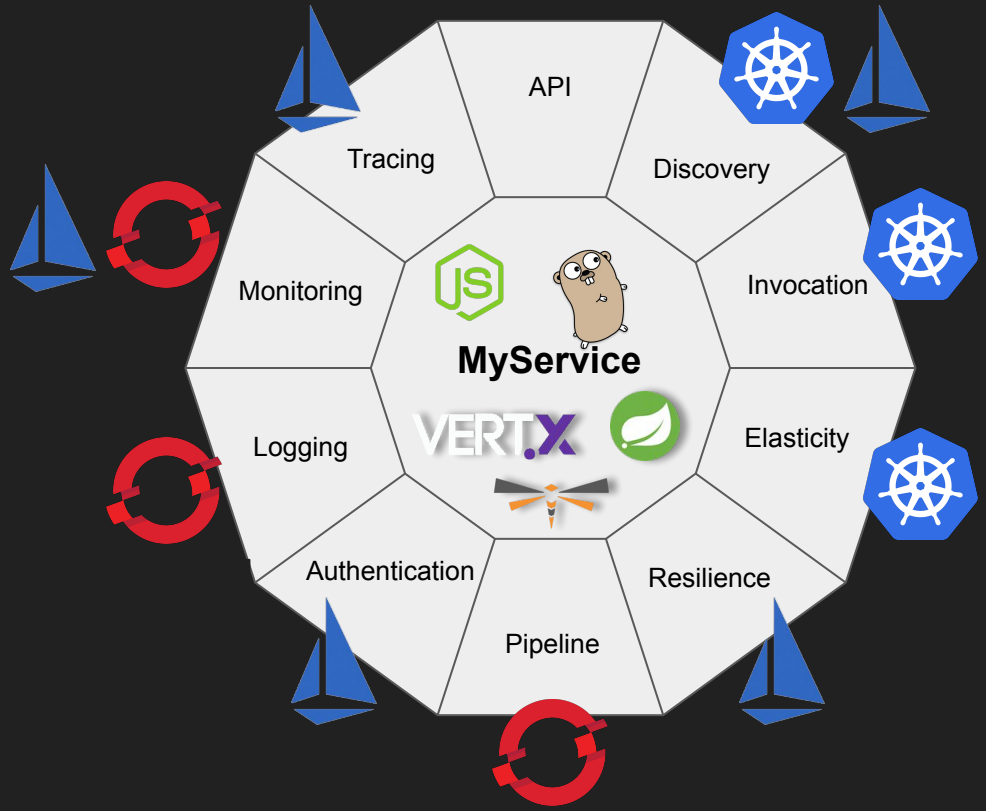
(Kubernetes - Helmsman or ship's pilot)

Service Mesh Defined

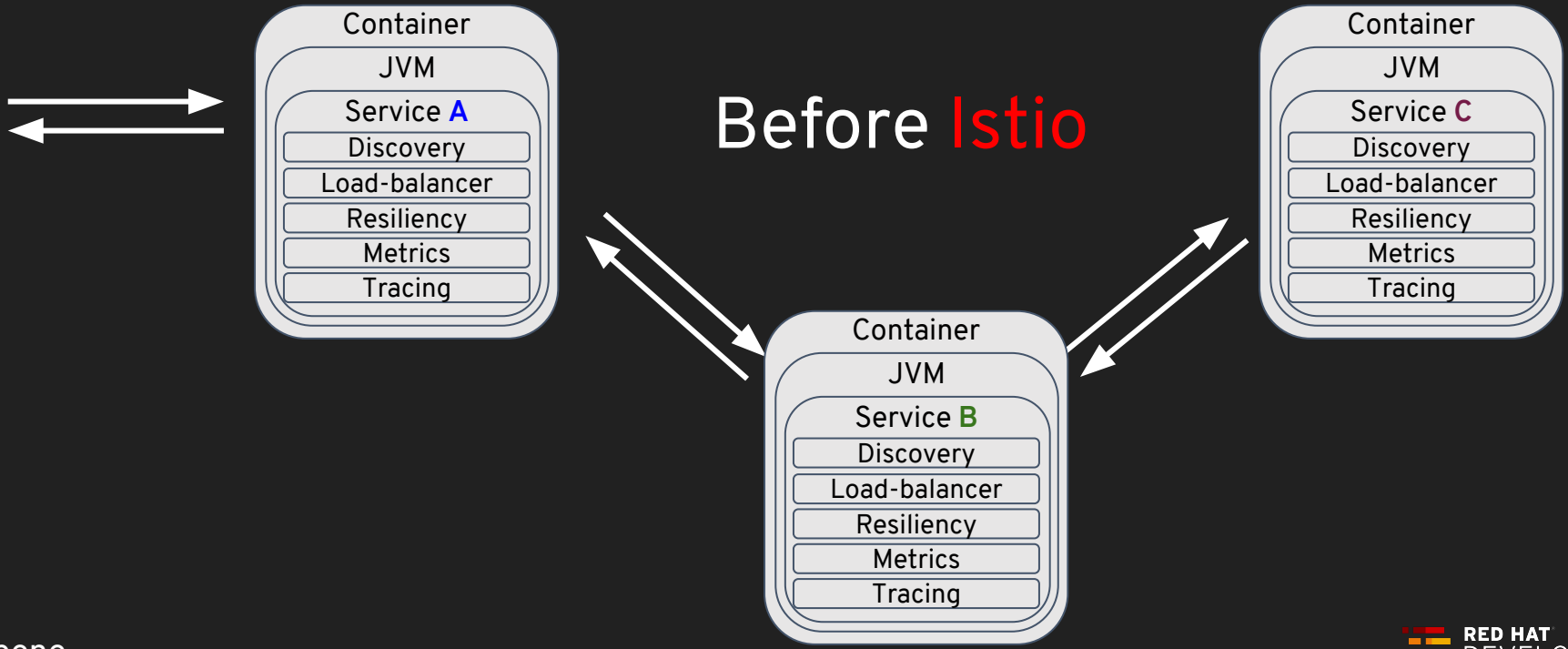
A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as 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/

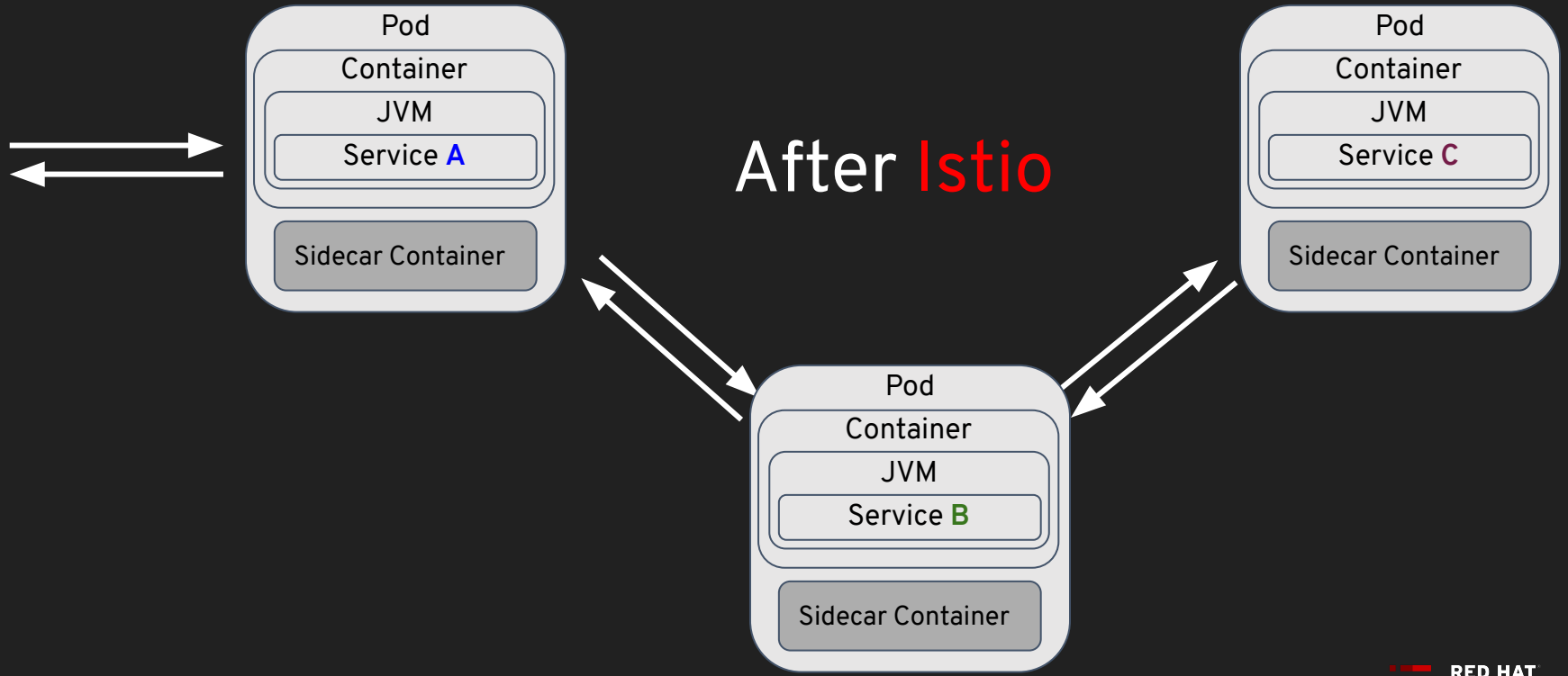
Microservices'ilities + Istio



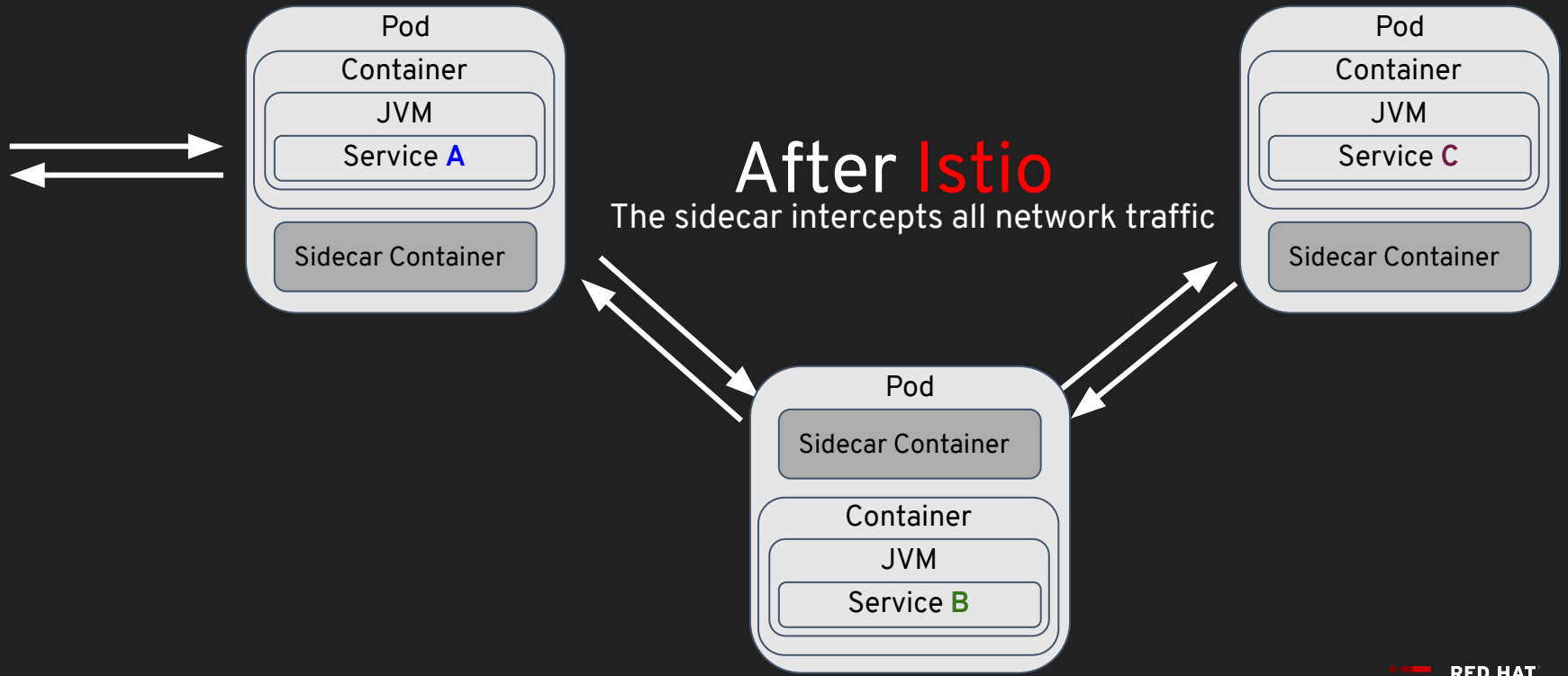
Microservices embedding Capabilities



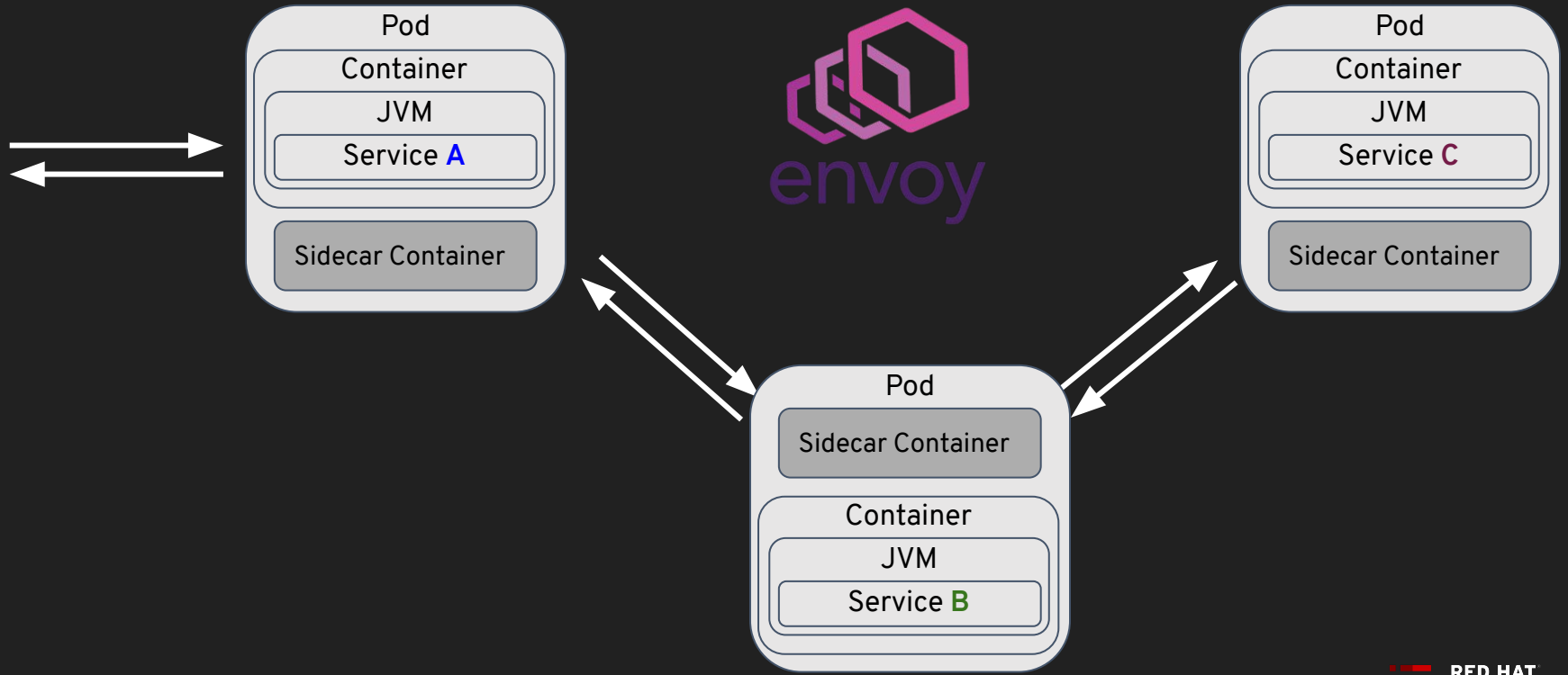
Microservices externalizing Capabilities



Microservices externalizing Capabilities



Envoy is the current sidecar



Sidecar

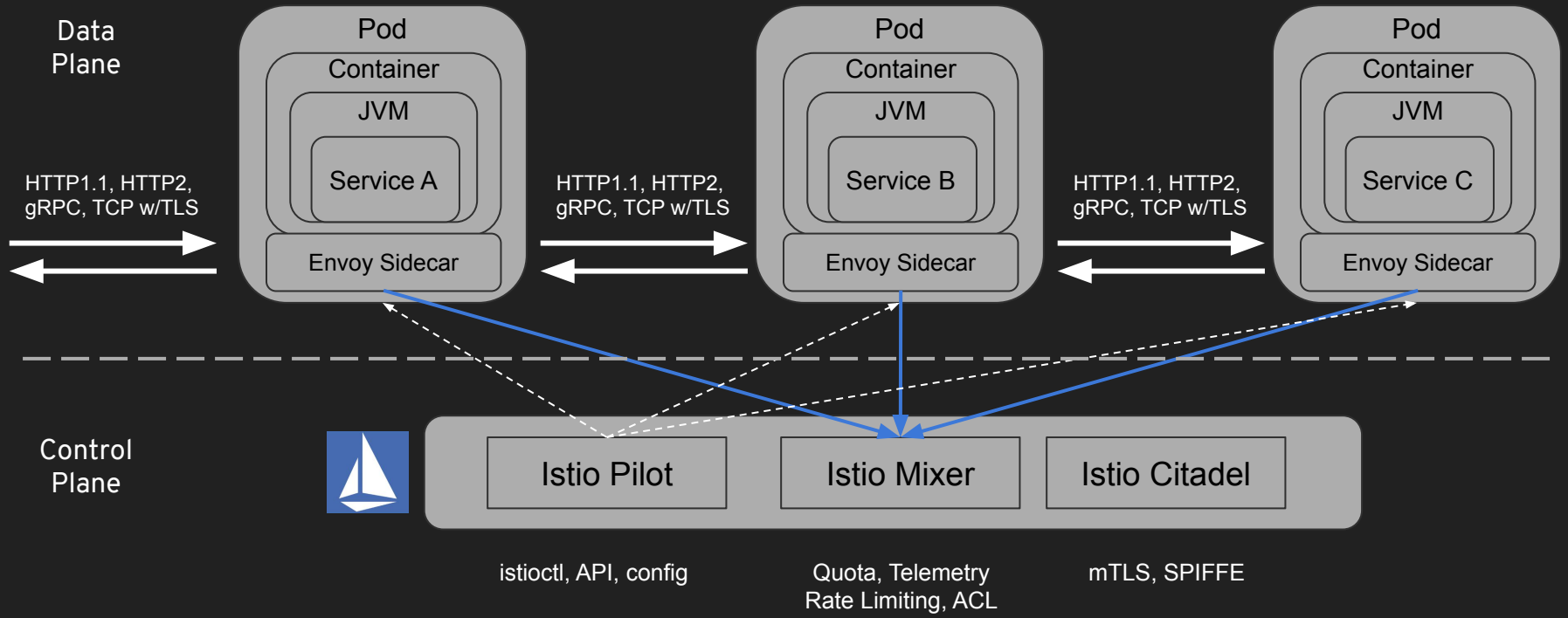
https://www.imz-ural.com/blog/waffles-the-sidecar-dog

Next Generation Microservices - Service Mesh

Code Independent (Polyglot)

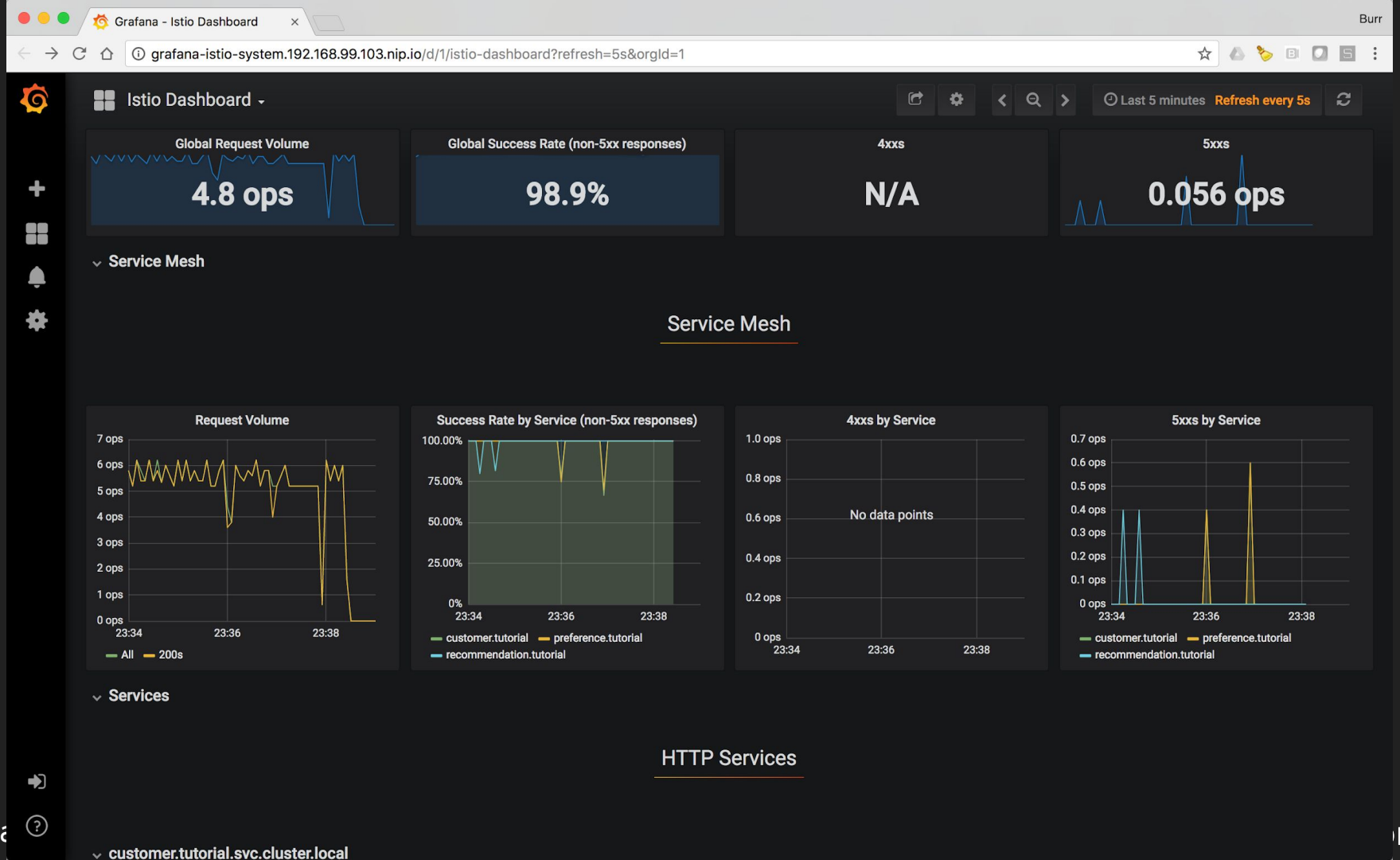
- Intelligent Routing and Load-Balancing
  - A/B Tests
  - Smarter Canary Releases
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability: Metrics and Tracing
- Fleet wide policy enforcement

Istio Data Plane vs Control Plane

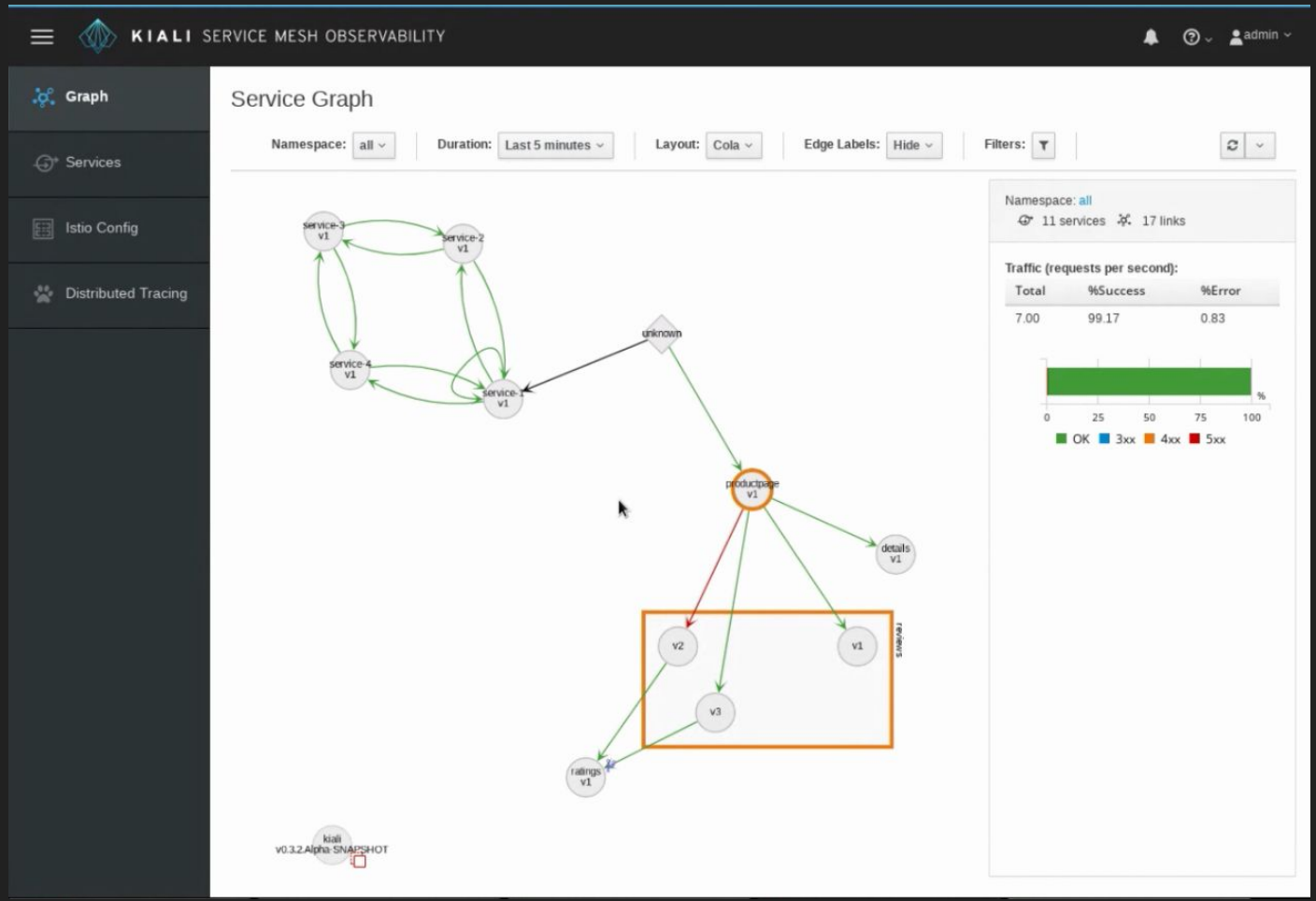


Polyglot Microservices Platform circa 2019

Observability

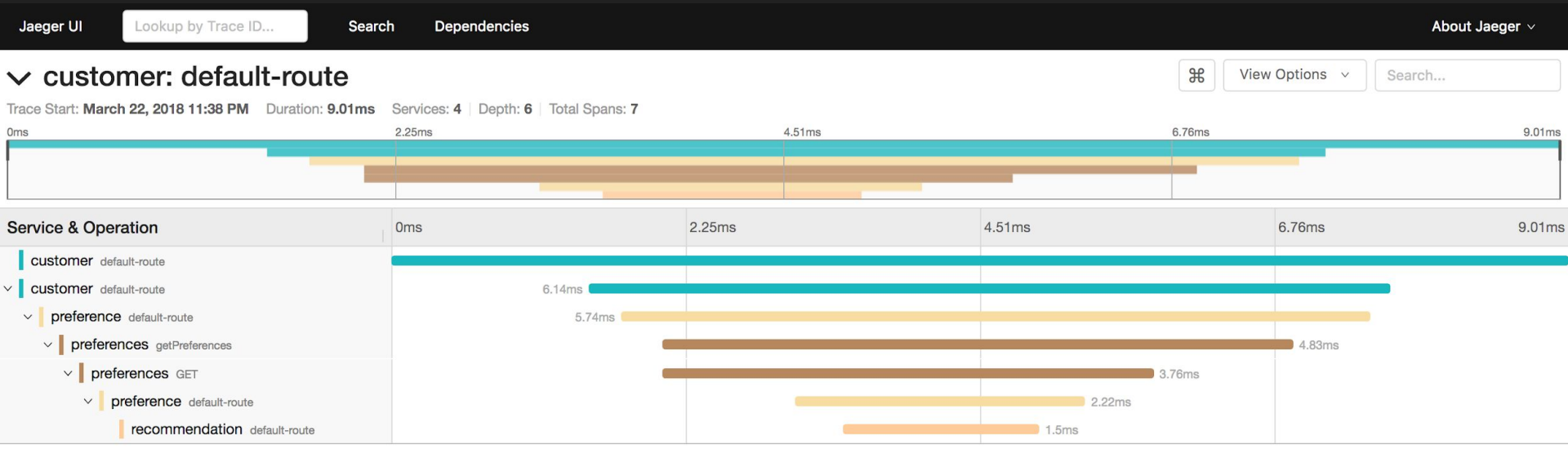

Grafana - Istio Dashboard
Burr
grafana-istio-system.192.168.99.103.nip.io/d/1/istio-dashboard?refresh=5s&orgId=1
Istio Dashboard
Last 5 minutes
Refresh every 5s
Global Request Volume
4.8 ops
Global Success Rate (non-5xx responses)
98.9%
4xxs
N/A
5xxs
0.056 ops
Service Mesh
Service Mesh
Request Volume
7 ops
6 ops
5 ops
4 ops
3 ops
2 ops
1 ops
0 ops
23:34
23:36
23:38
All
200s
Success Rate by Service (non-5xx responses)
100.00%
75.00%
50.00%
25.00%
0%
customer.tutorial
preference.tutorial
recommendation.tutorial
4xxs by Service
1.0 ops
0.8 ops
0.6 ops
0.4 ops
0.2 ops
0 ops
No data points
5xxs by Service
0.7 ops
0.6 ops
0.5 ops
0.4 ops
0.3 ops
0.2 ops
0.1 ops
Services
HTTP Services
customer.tutorial.svc.cluster.local

Kiali.io
New Service Graph

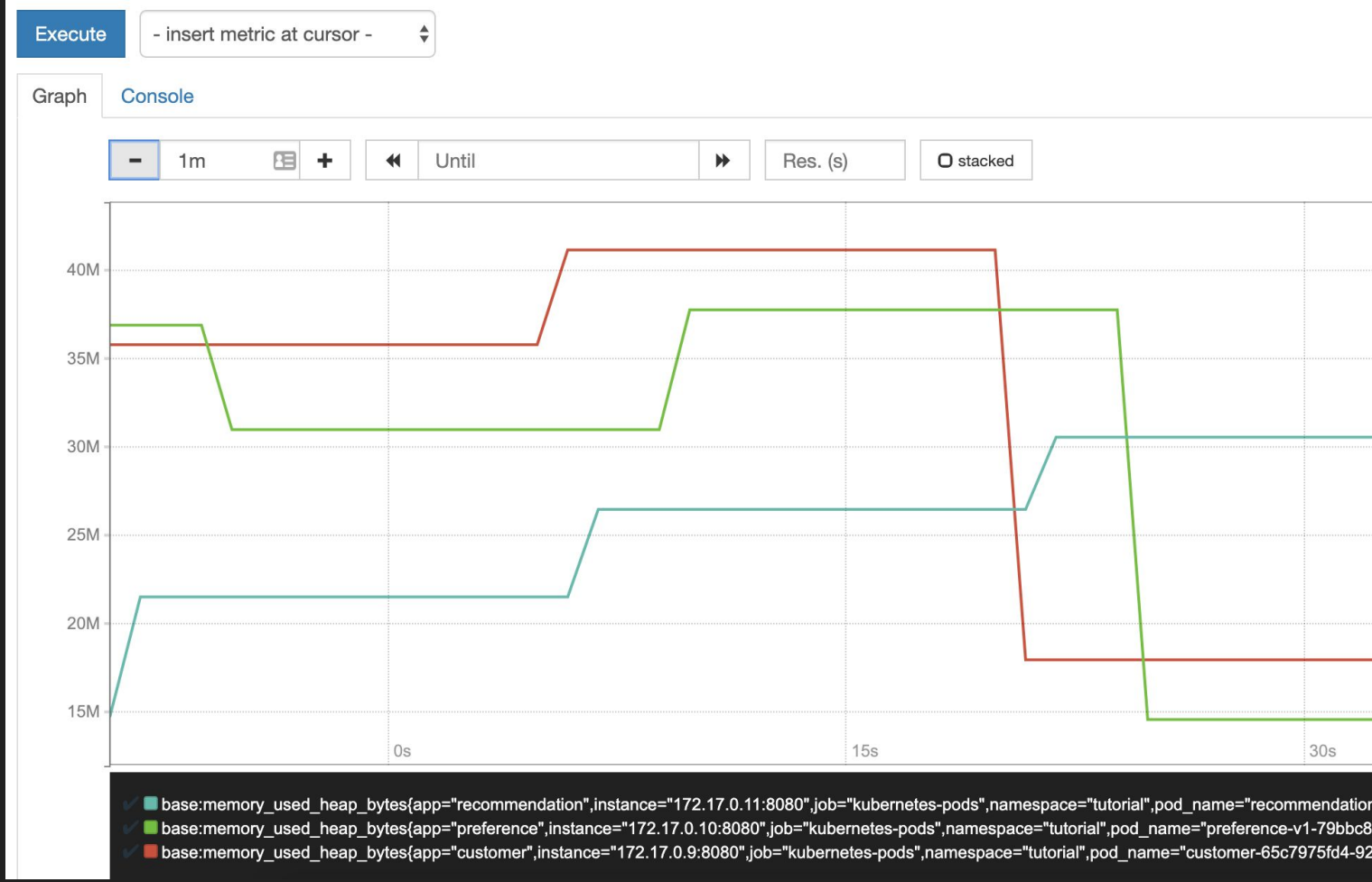


@rafabene

RED HAT DEVELOPER



Prometheus



How to add an Istio-Proxy (sidecar)?

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

OR

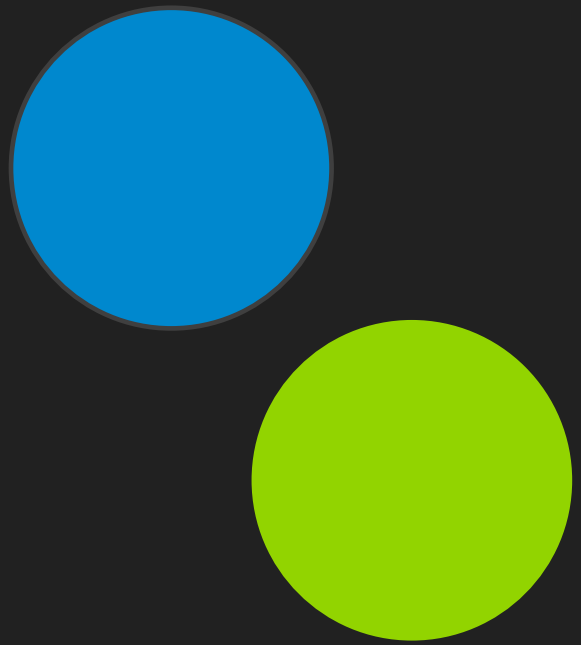
```
kubectl label namespace myspace istio-injection=enabled
```

To "see" the sidecar:

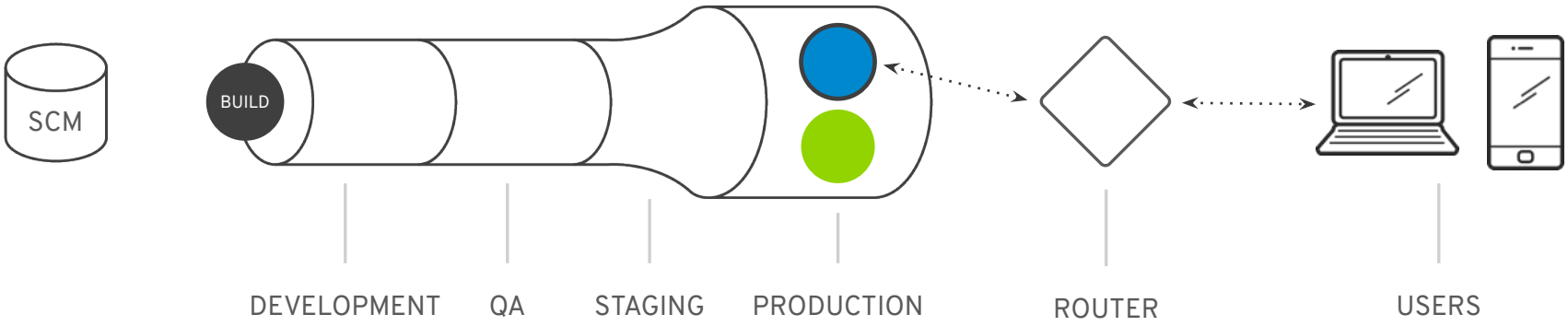
```
kubectl describe deployment customer
```

Traffic Control

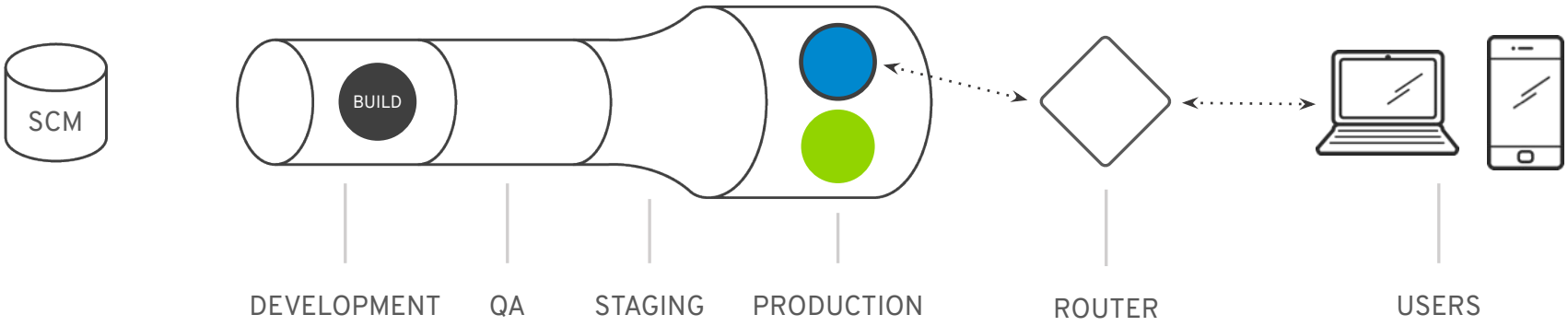
Blue/Green Deployment



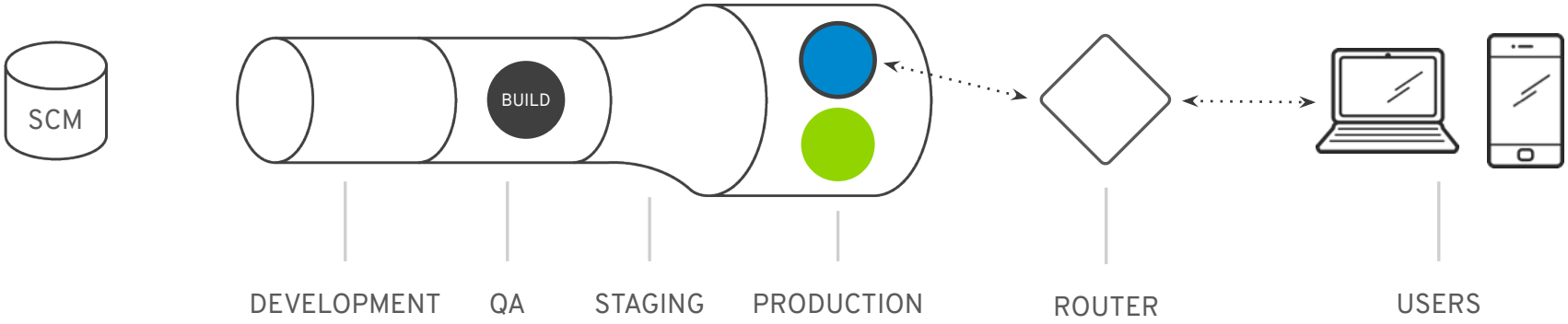
Blue/Green Deployment



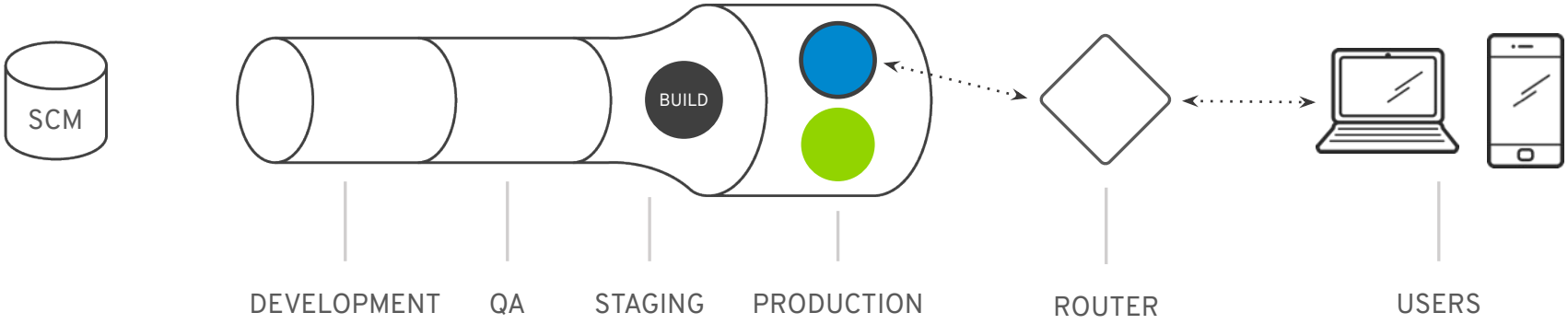
Blue/Green Deployment



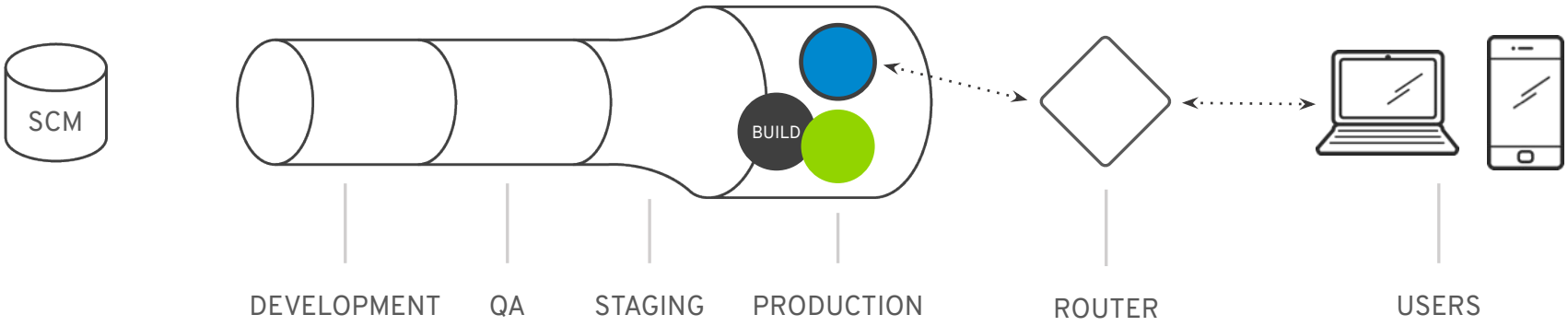
Blue/Green Deployment



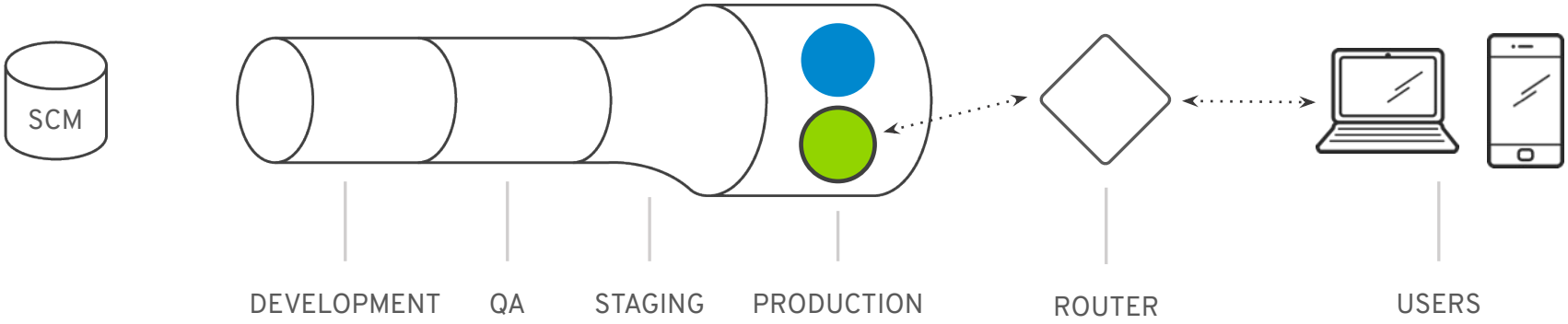
Blue/Green Deployment



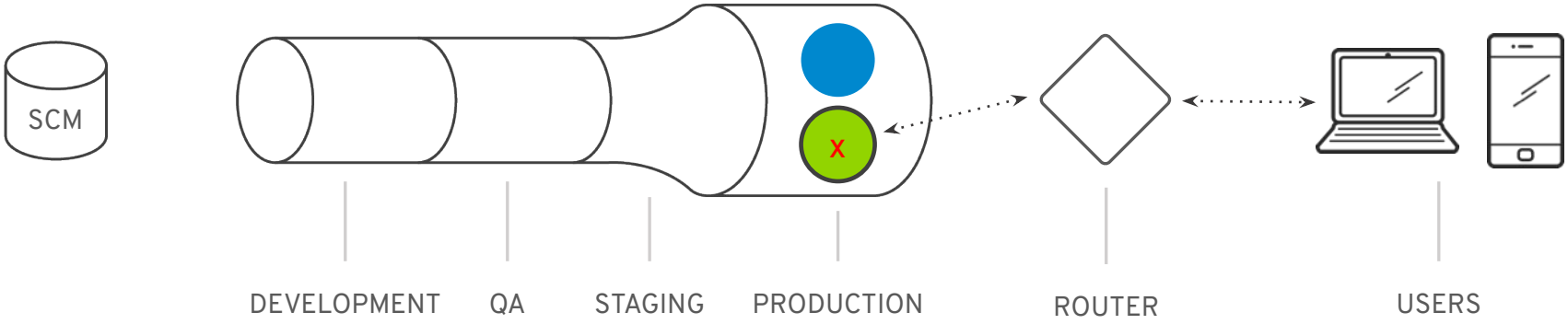
Blue/Green Deployment



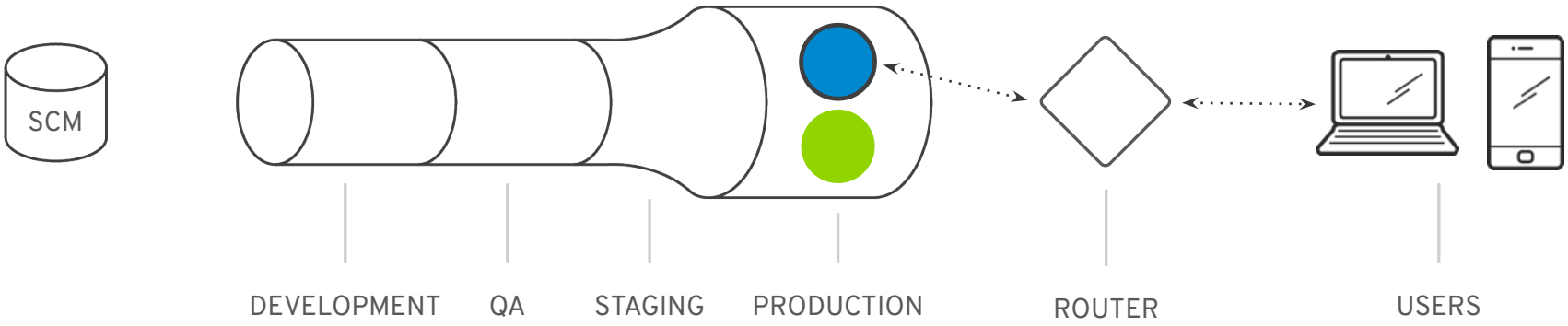
Blue/Green Deployment



Blue/Green Deployment



Blue/Green Deployment

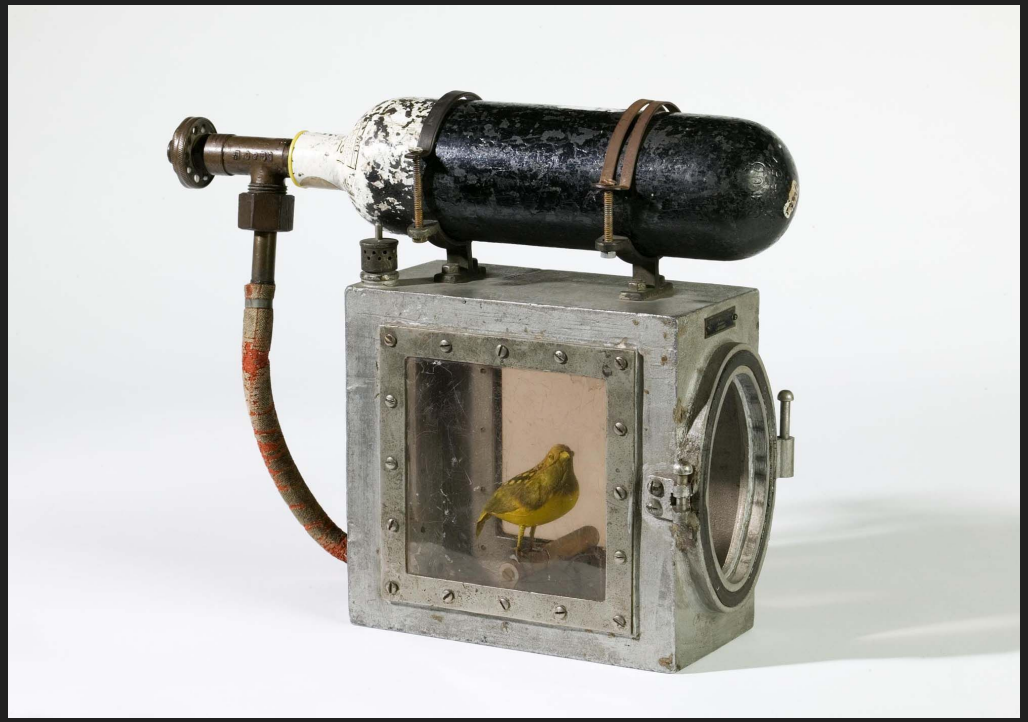


Demo
Blue/Green

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

Canary Deployment

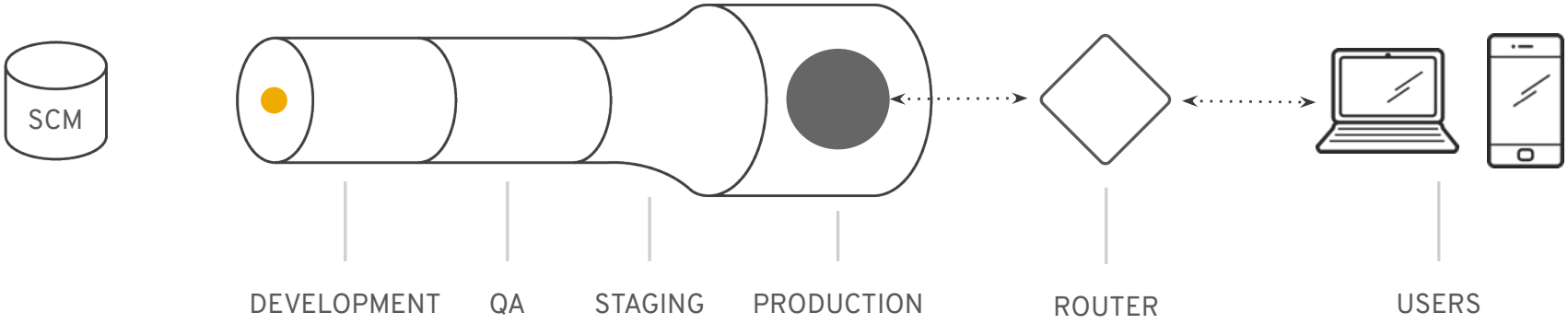
Canary Resuscitator



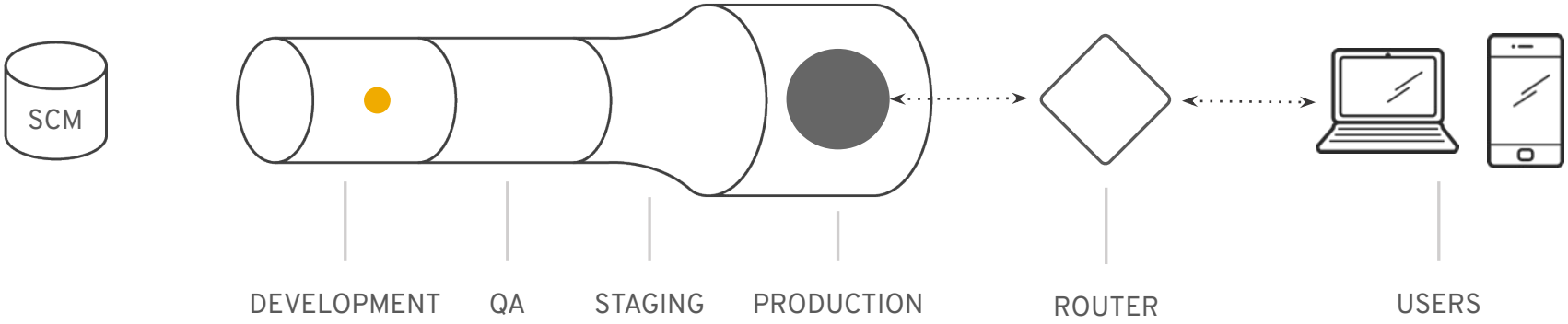
http://www.openculture.com/2018/05/the-device-invented-to-resuscitate-canaries-in-coal-mines-circa-1896.html

Thanks to **Paolo Antinori!**

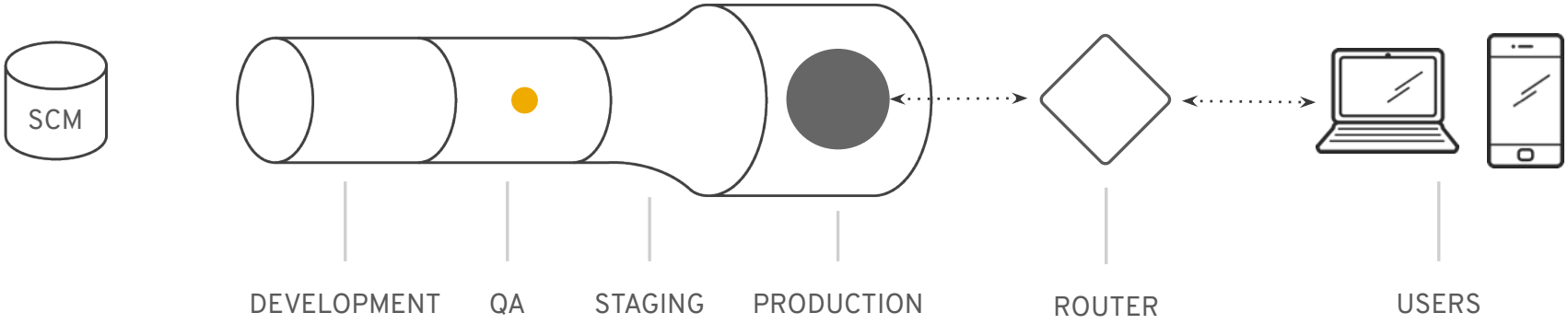
Canary Deployment



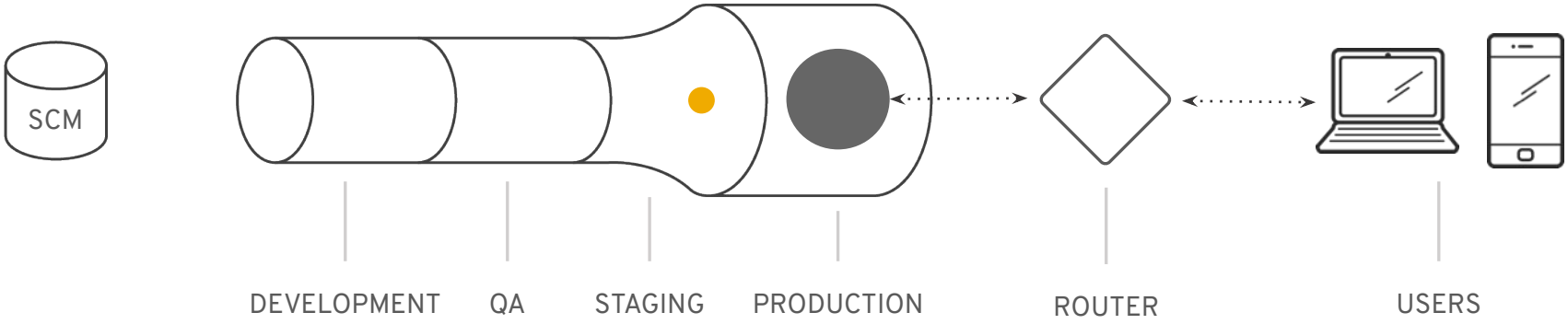
Canary Deployment



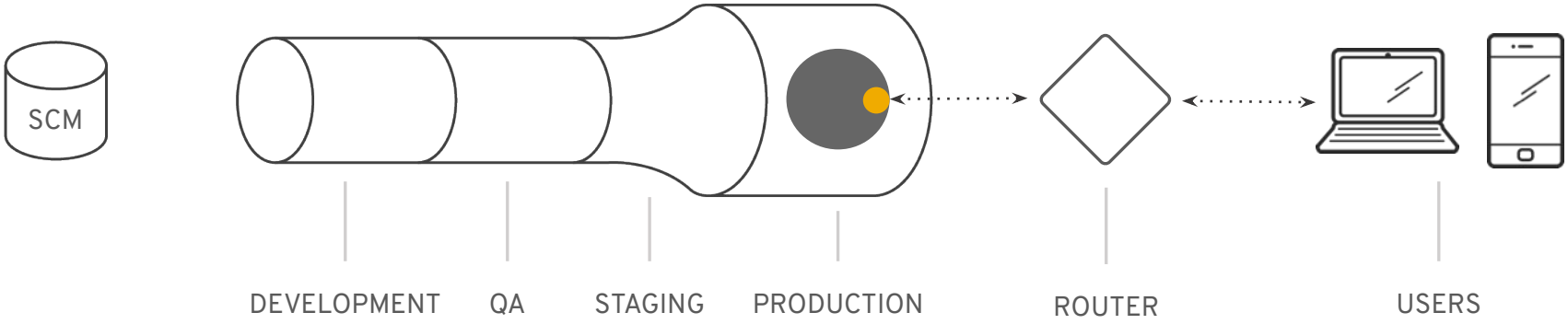
Canary Deployment



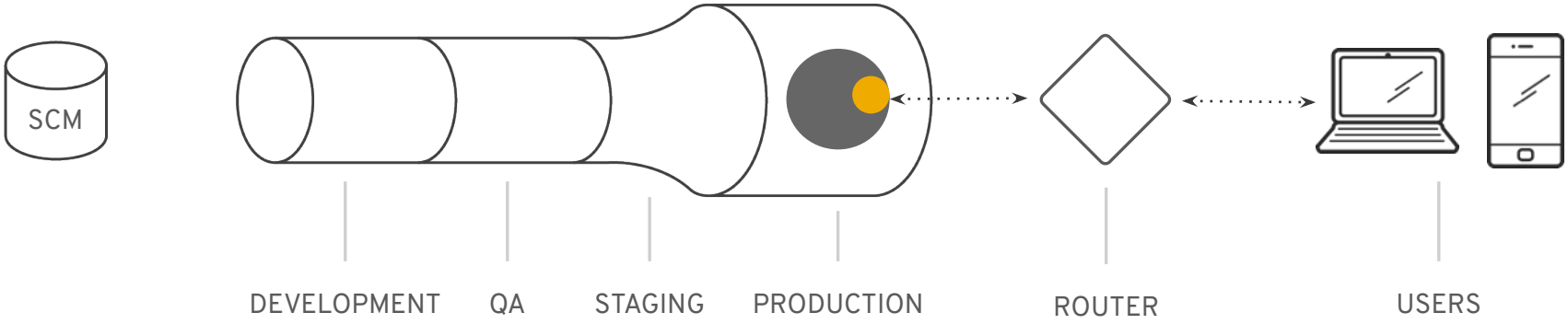
Canary Deployment



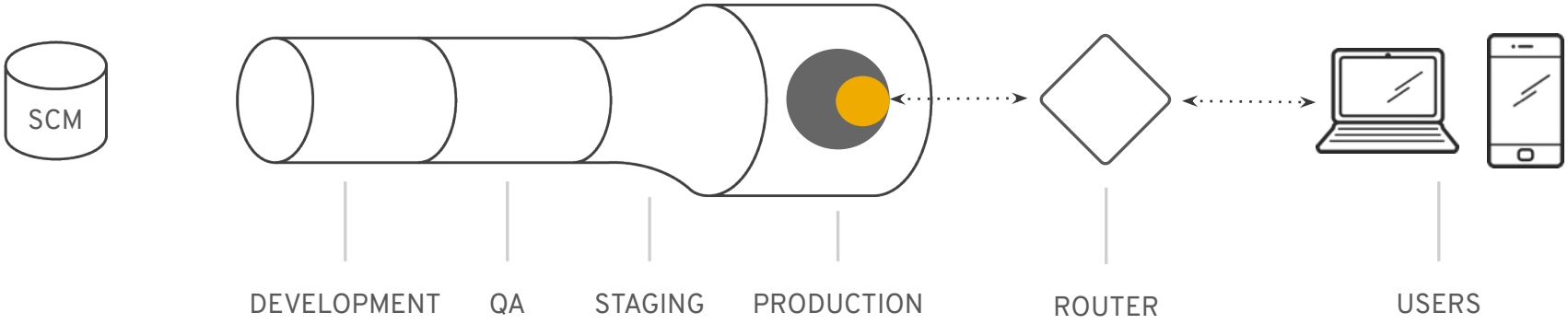
Canary Deployment



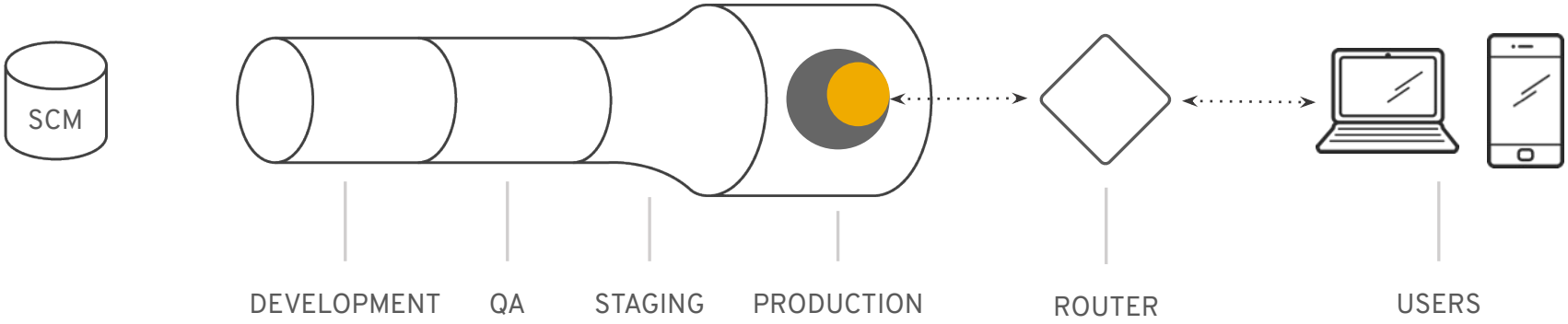
Canary Deployment



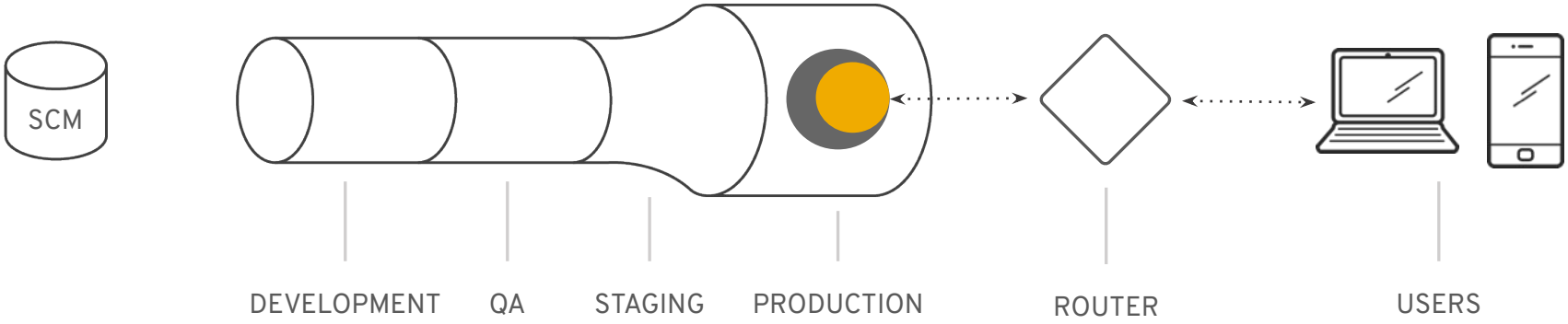
Canary Deployment



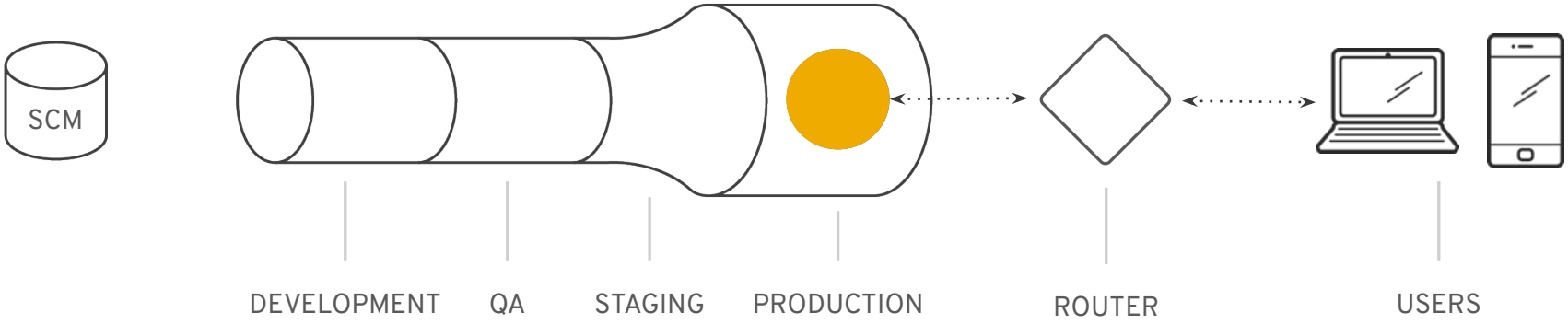
Canary Deployment



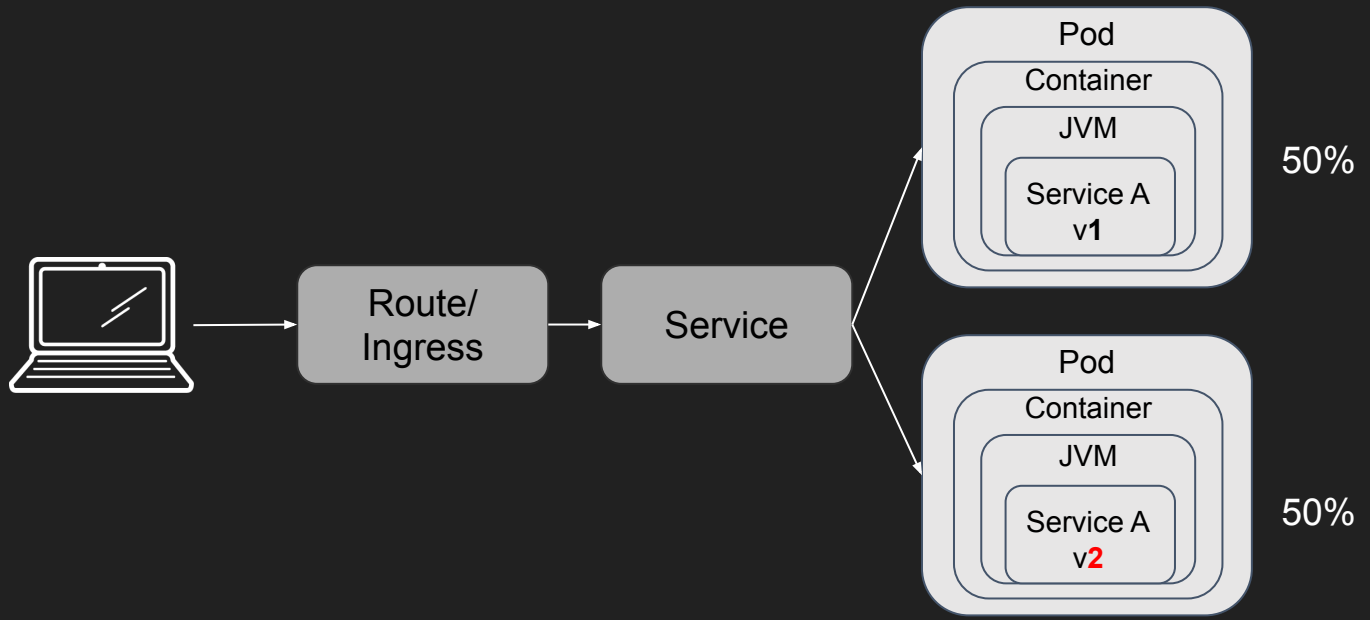
Canary Deployment



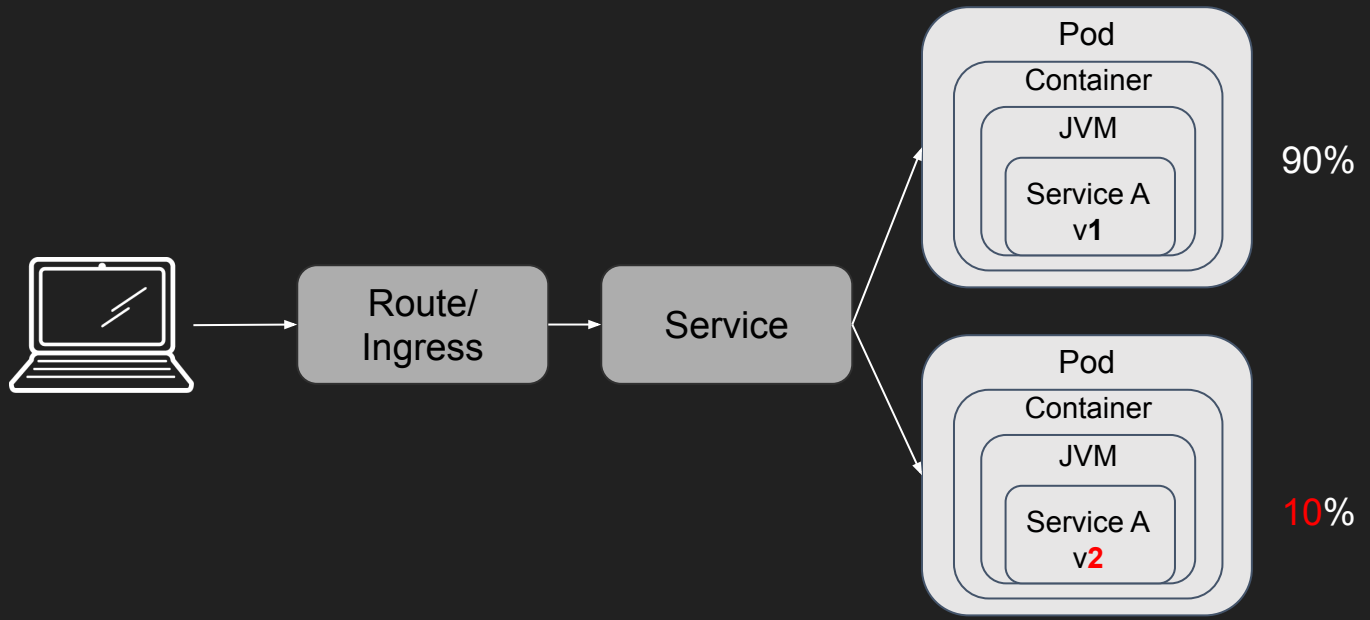
Canary Deployment



Canaries with Kubernetes



Canaries with Istio

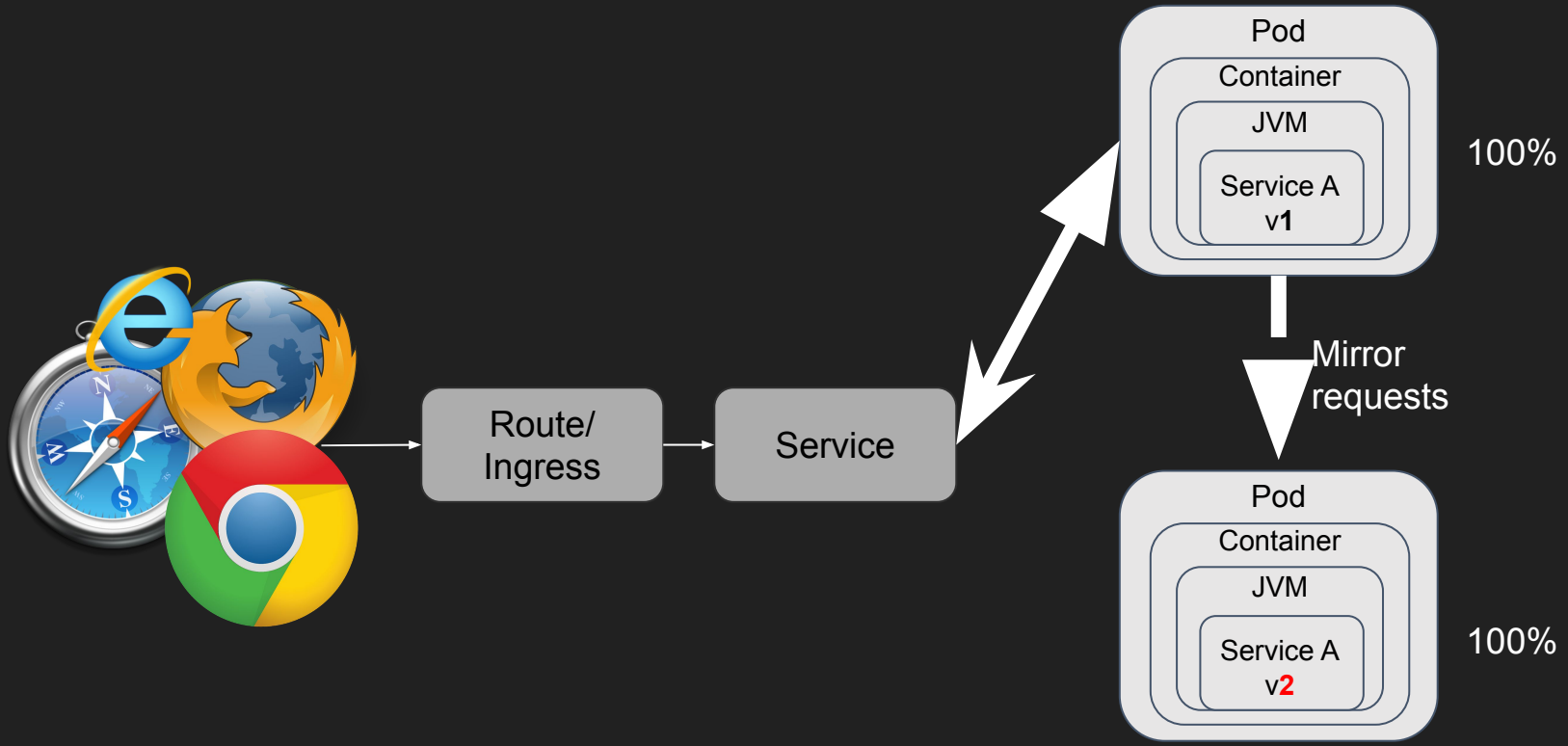


Demo Canary

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

Dark Launch

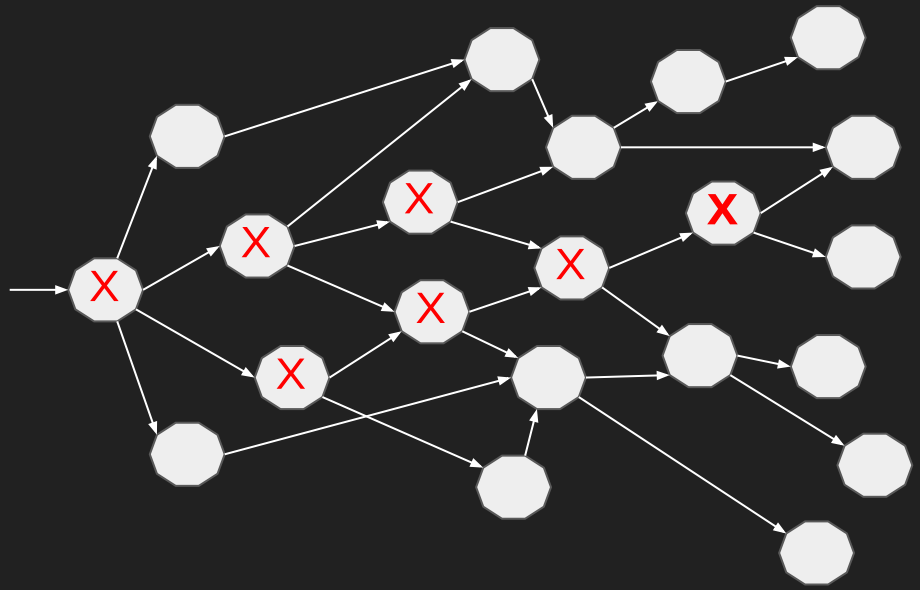
Dark Launches with Istio



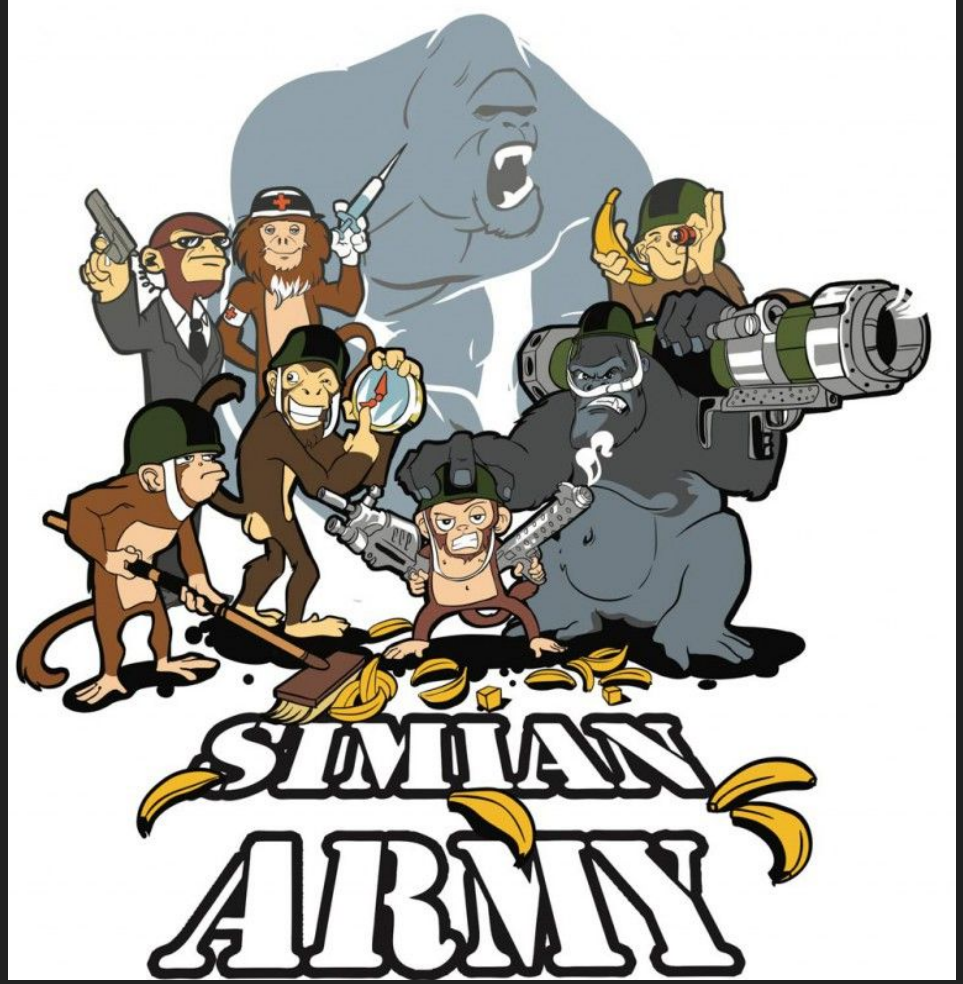
Demo Dark Launch

Service Resiliency

- Retry
- Kiali



Chaos Testing

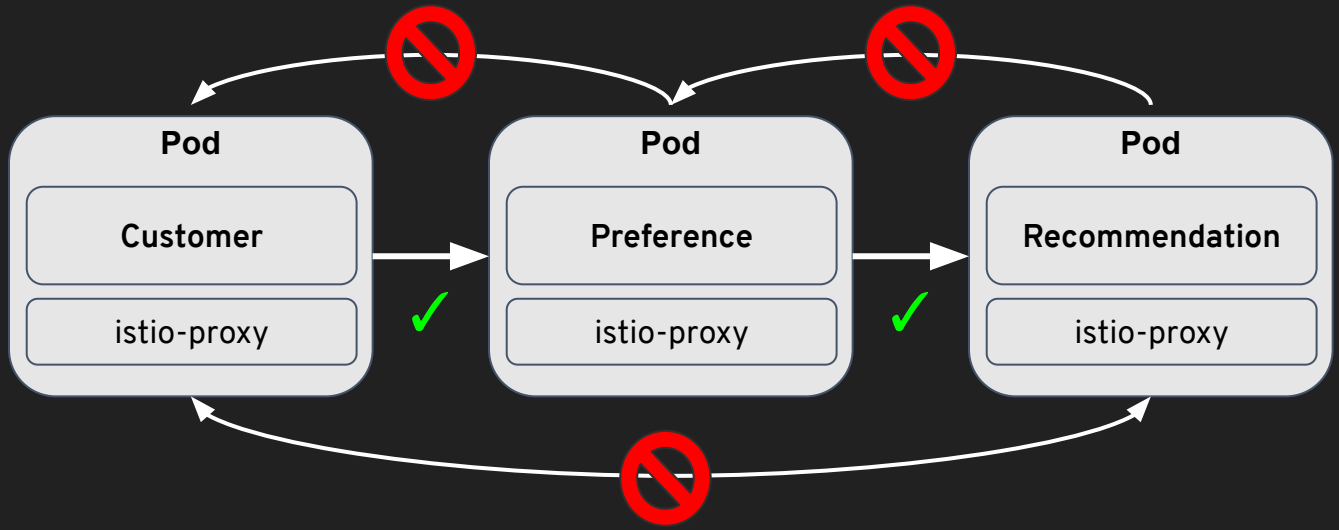


By Netflix - https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png, Apache License 2.0, https://commons.wikimedia.org/w/index.php?curid=63503083

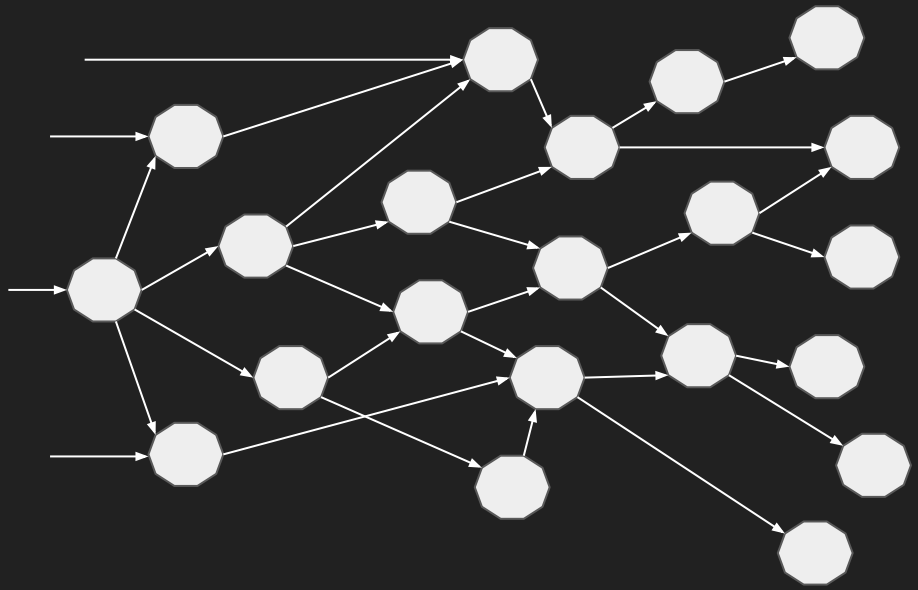
Demo Caos

- 503
- Delay

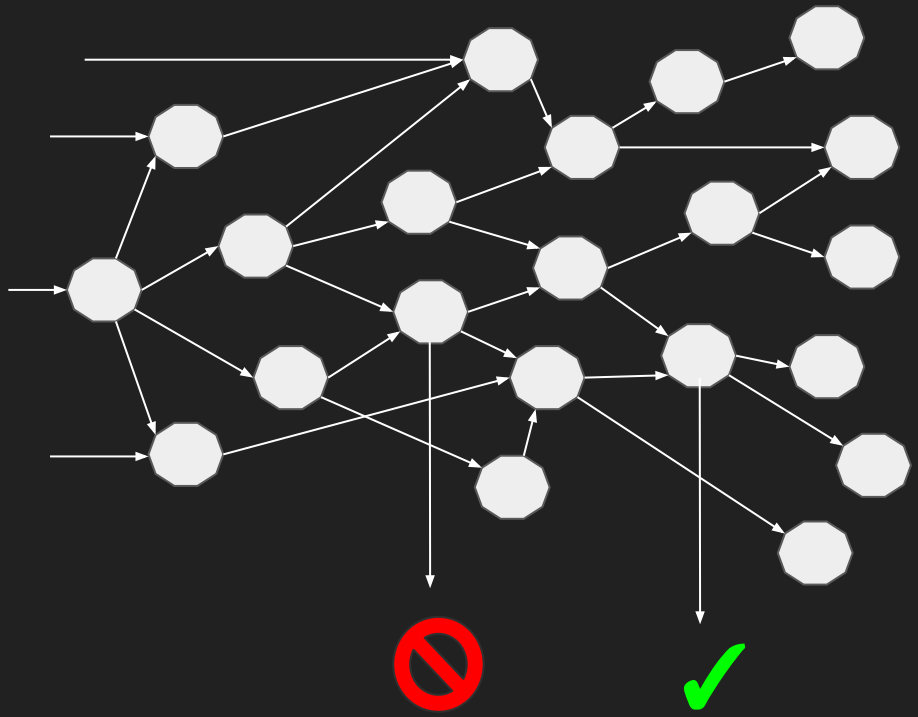
Access Control



Most Communication Inbound & Internal

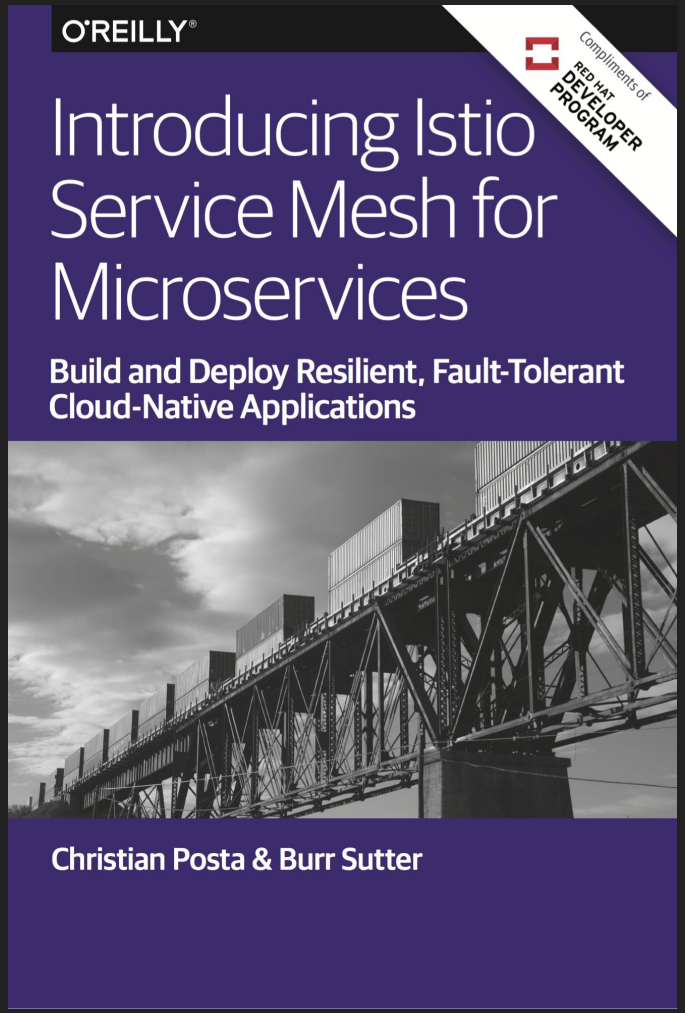


Outbound/Egress Blocked By Default



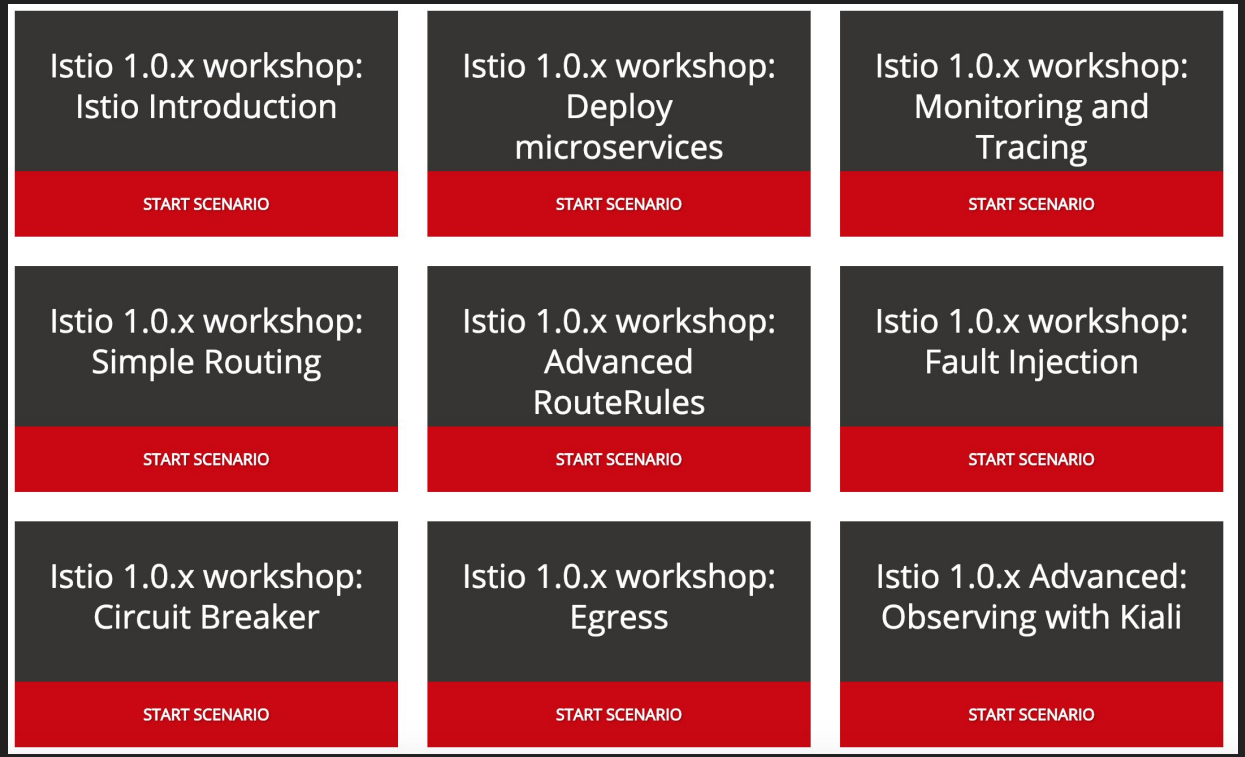
Demo Egress

- Access http://worldclockapi.com



bit.ly/istio-book

https://learn.openshift.com/servicemesh



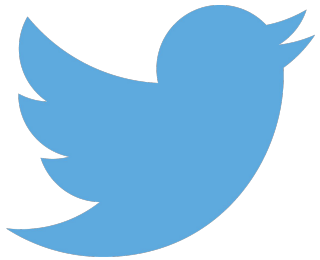
Demo

bit.ly/istio-tutorial

Workshp

bit.ly/the-istio-workshop

The End
(but Serverless is coming)



@RAFABENE