



Istio on Kubernetes: Enter the Service Mesh

Burr Sutter (burrsutter.com)

bit.ly/istio-tutorial

Upcoming 3 hour classes/workshops

Istio on Kubernetes: Enter the Service Mesh (Advanced Class)

March 14, 2019

<https://www.safaribooksonline.com/live-training/courses/istio-on-kubernetes-enter-the-service-mesh/0636920231745/>

9 Steps to Awesome with Kubernetes (Introductory Class)

February 5

<https://www.safaribooksonline.com/live-training/courses/9-steps-to-awesome-with-kubernetes/0636920231363/>

March 12, 2019

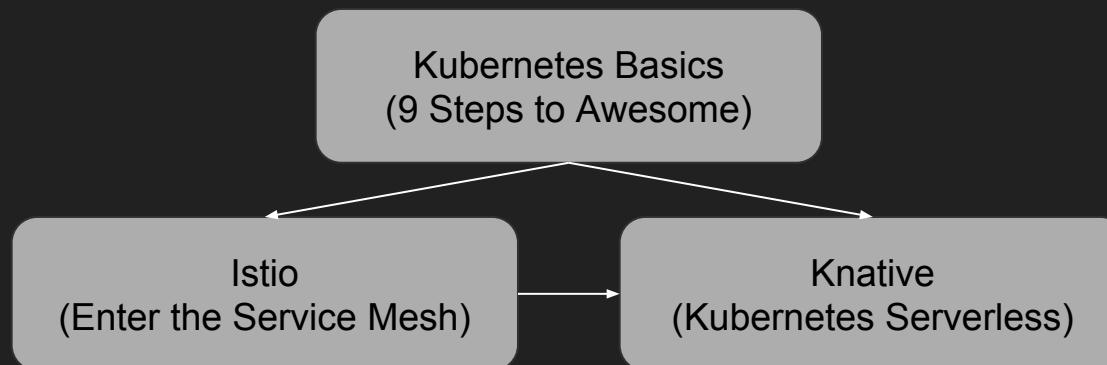
<https://www.safaribooksonline.com/live-training/courses/9-steps-to-awesome-with-kubernetes/0636920231783/>

New 3-hour Deep Dive

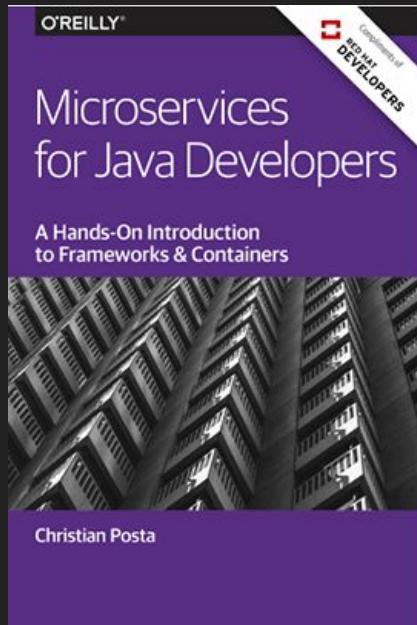
Kubernetes Serverless with Knative

March 15, 2019

<https://www.safaribooksonline.com/live-training/courses/kubernetes-serverless-with-knative/0636920257226/>



bit.ly/javamicroservicesbook



Free eBooks from developers.redhat.com

Microservices Introductory Materials

MSA Demo: bit.ly/msa-instructions

MSA Slides: bit.ly/microservicesdeepdive

Video Training: bit.ly/microservicesvideo

[Kubernetes for Java Developers](#)

[9 Steps to Awesome with Kubernetes](#)

Advanced Materials

bit.ly/istio-tutorial

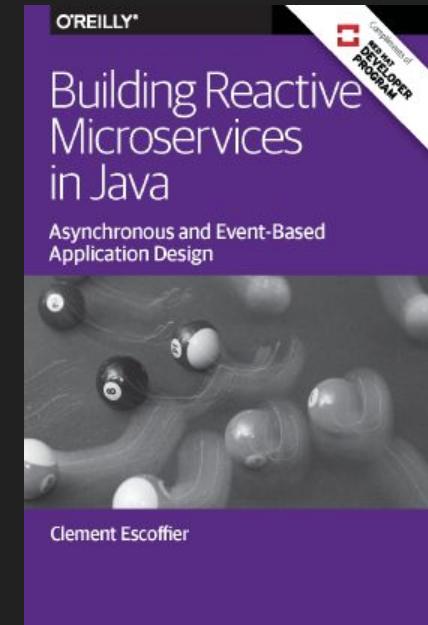
learn.openshift.com/servicemesh

bit.ly/faas-tutorial

learn.openshift.com/serverless

bit.ly/istio-intro

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

Being Updated to 2nd Edition with Istio 1.x

Exercise Setup

Minishift

<https://redhat-developer-demos.github.io/istio-tutorial/istio-tutorial/1.0.0/1setup.html>

<https://redhat-developer-demos.github.io/istio-tutorial/istio-tutorial/1.0.0/2deploy-microservices.html>

Minikube

<https://istio.io/docs/setup/kubernetes/platform-setup/minikube/>

```
kubectl apply -f install/kubernetes/helm/istio/templates/crds.yaml  
kubectl apply -f install/kubernetes/istio-demo.yaml
```

<https://istio.io/docs/setup/kubernetes/quick-start/#verifying-the-installation>

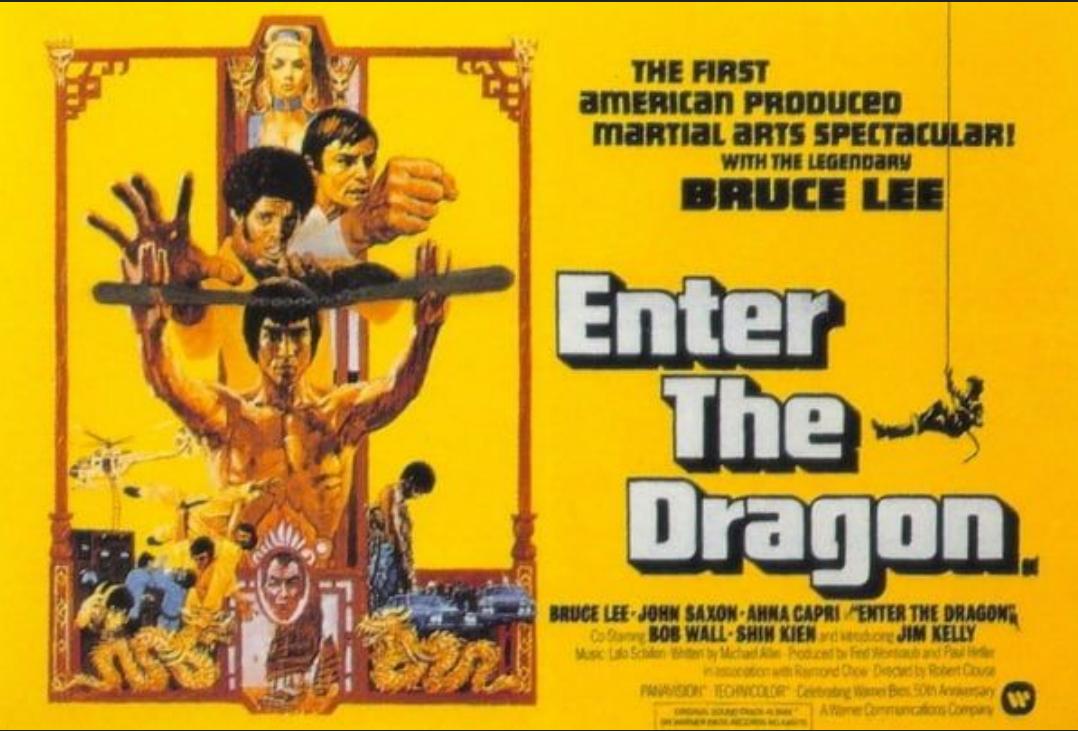
Testing/Demo/Minishift & Minikube Scripts
<https://github.com/burrsutter/scripts-istio>

Istio Tutorial Exercises (bit.ly/istio-tutorial)

- Setup
- Deploy Microservices
- Traffic Control: Simple Routing
- Traffic Control: Advanced Routing
- Chaos
- Service Resiliency & Circuit Breaking
- Security

Agenda

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

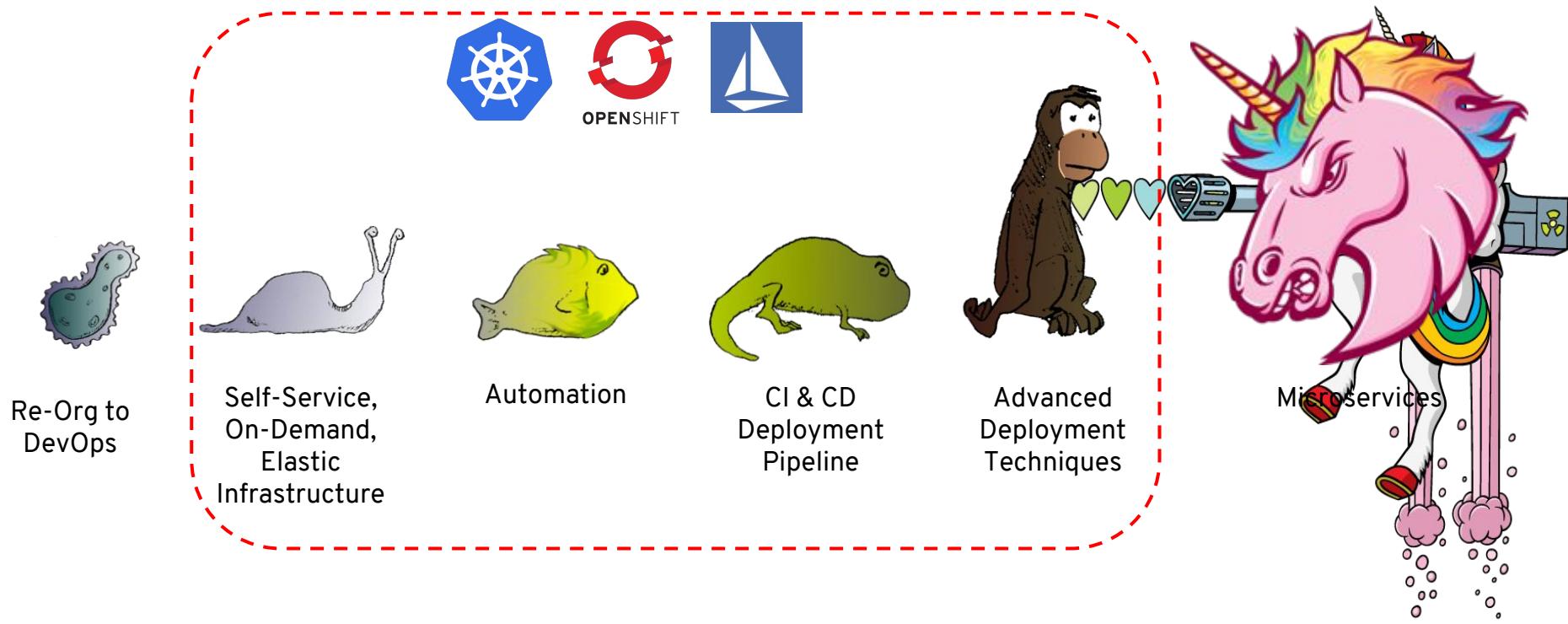


<https://www.flickeringmyth.com/2018/07/deadpool-2-director-in-talks-for-enter-the-dragon-remake/>

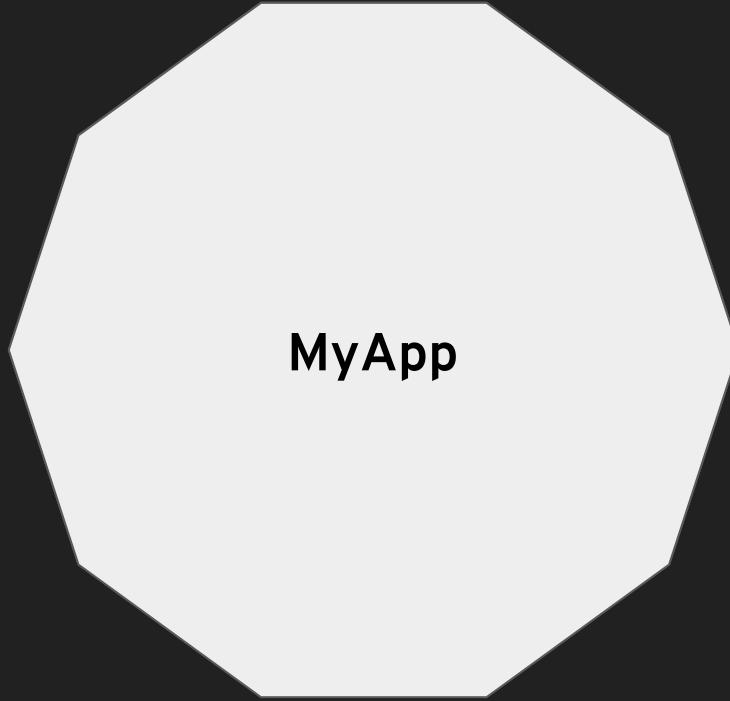


<https://www.watershed.co.uk/whatson/9037/enter-the-dragon>

Your Journey to Awesomeness

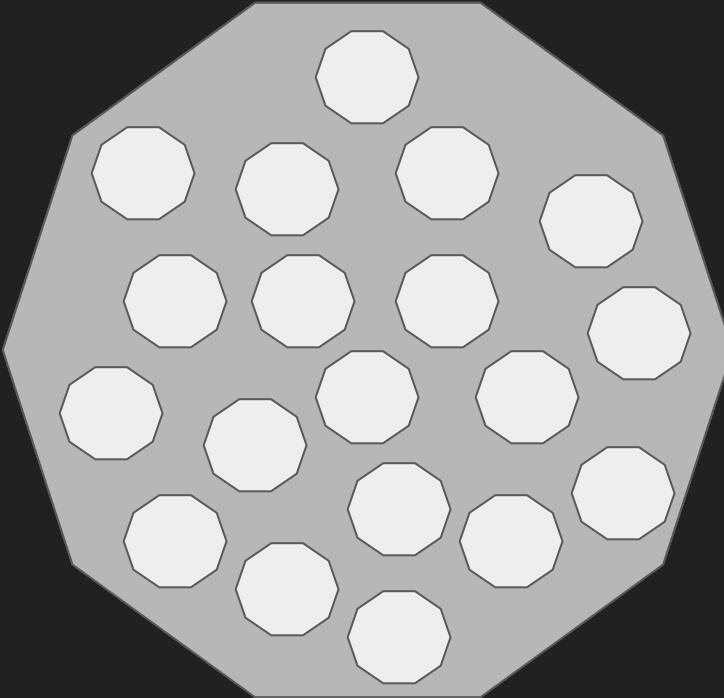


Monolith

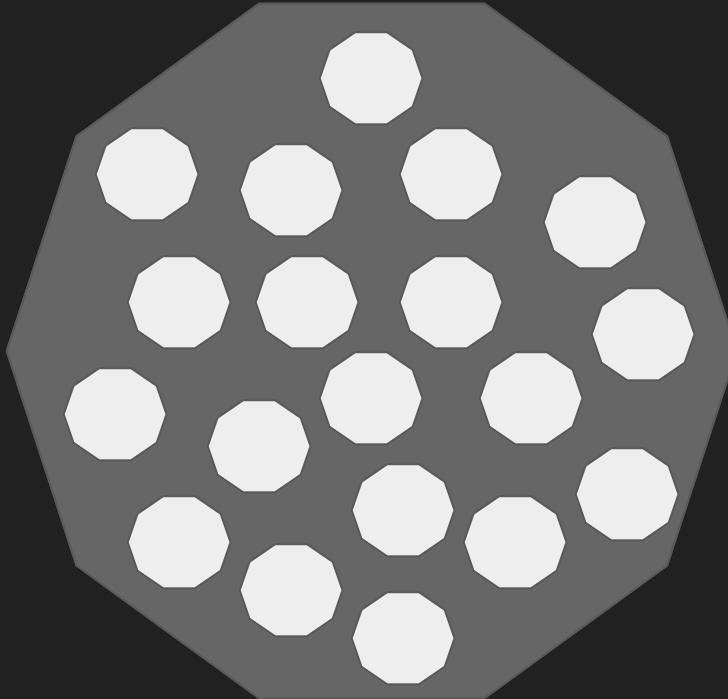


MyApp

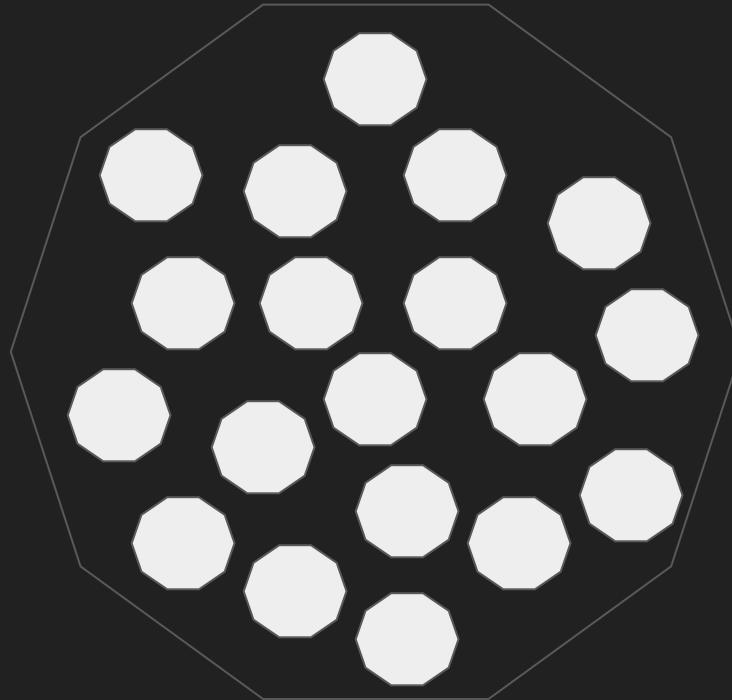
The Application



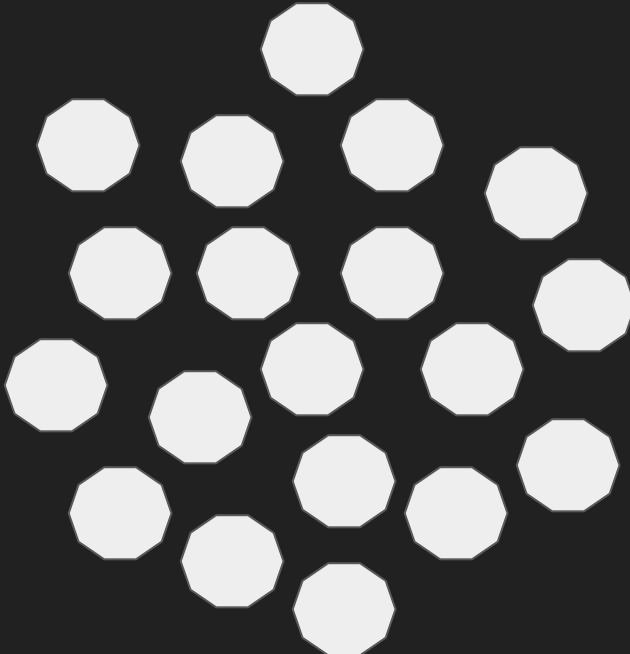
Modules



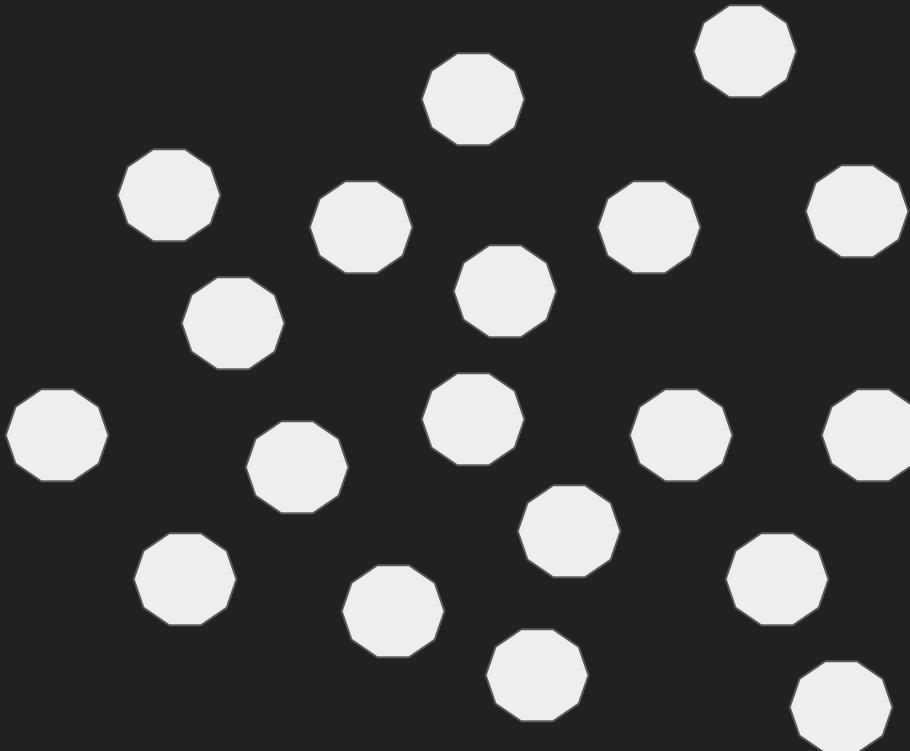
Microservices



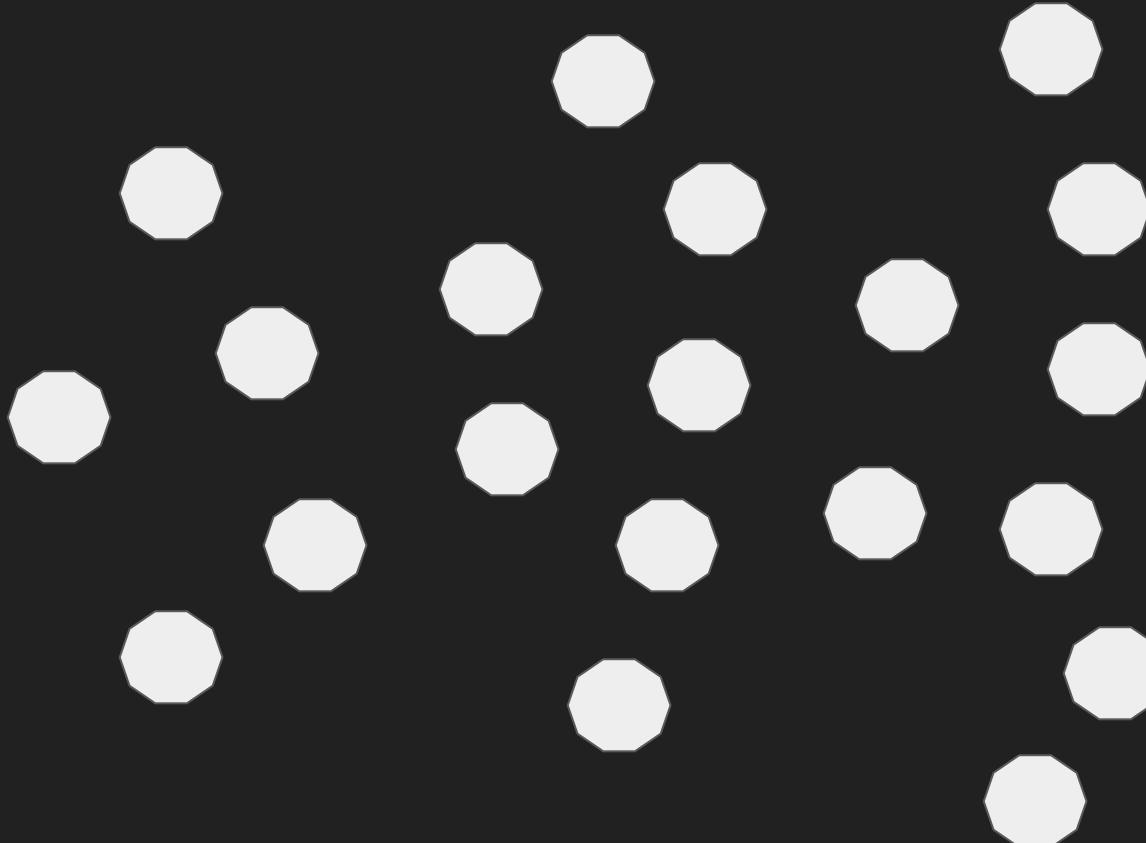
Microservices



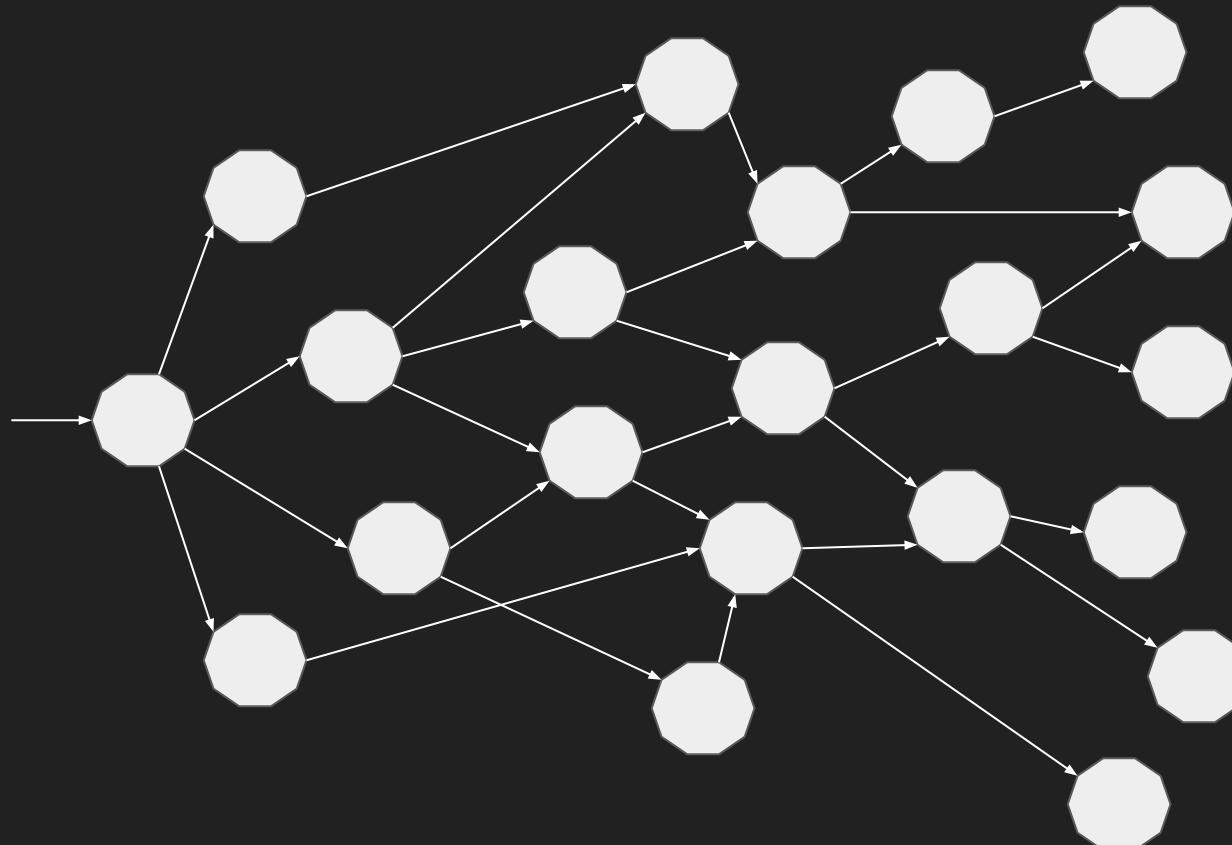
Microservices



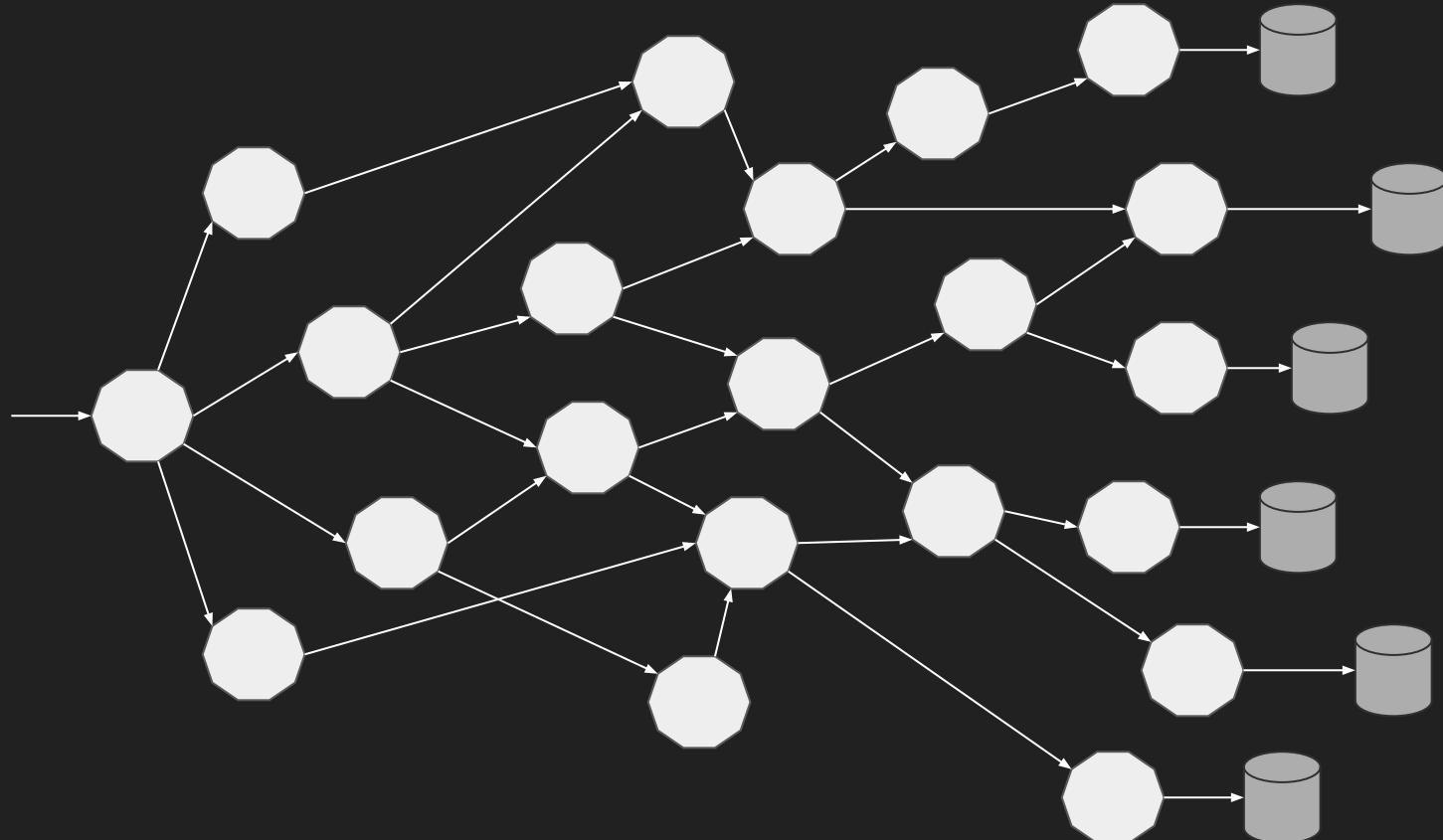
Microservices



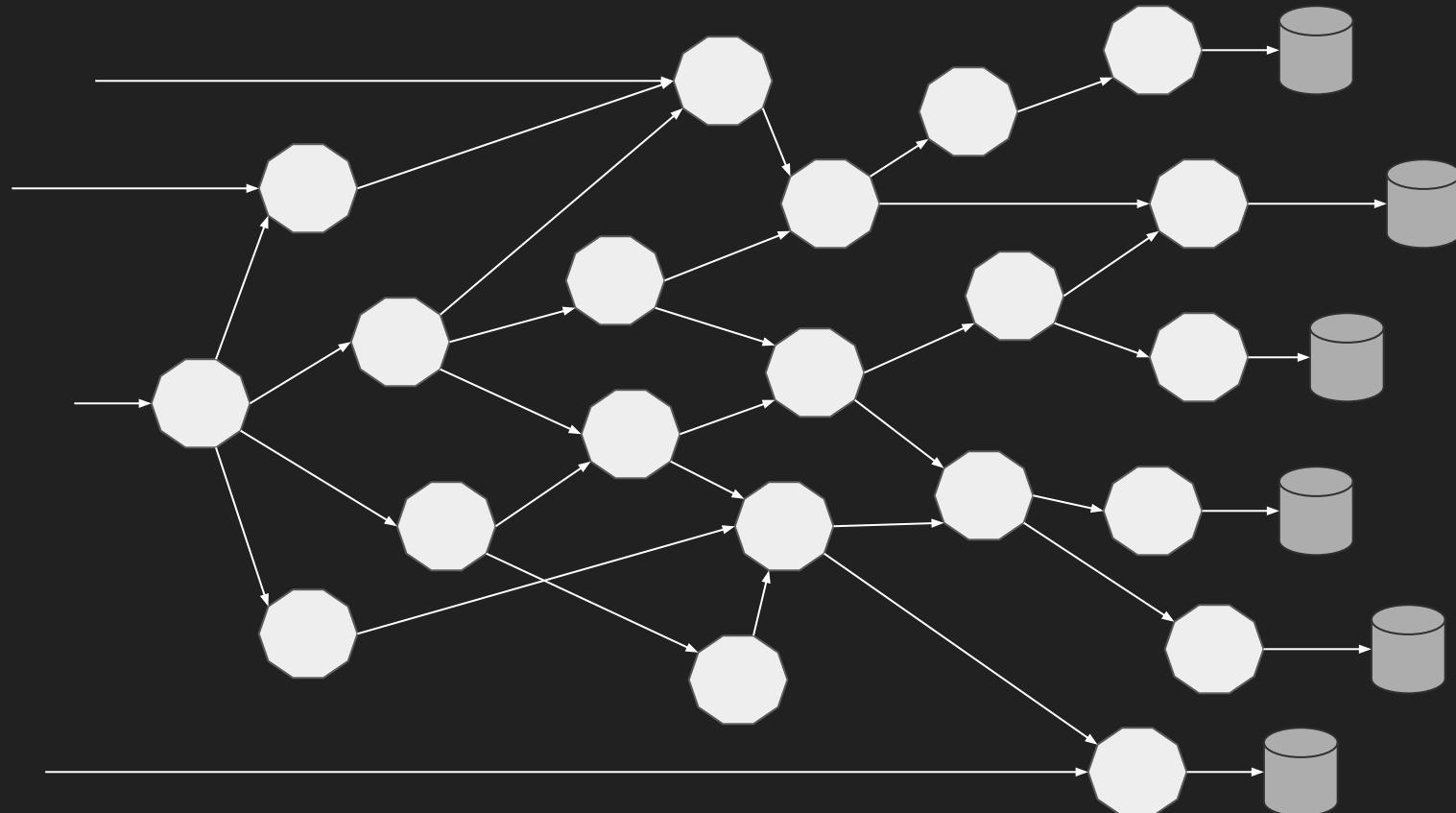
Network of Services



Microservices own their Data



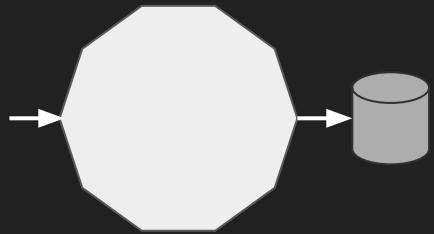
Multiple Points of Entry



Microservices Principles

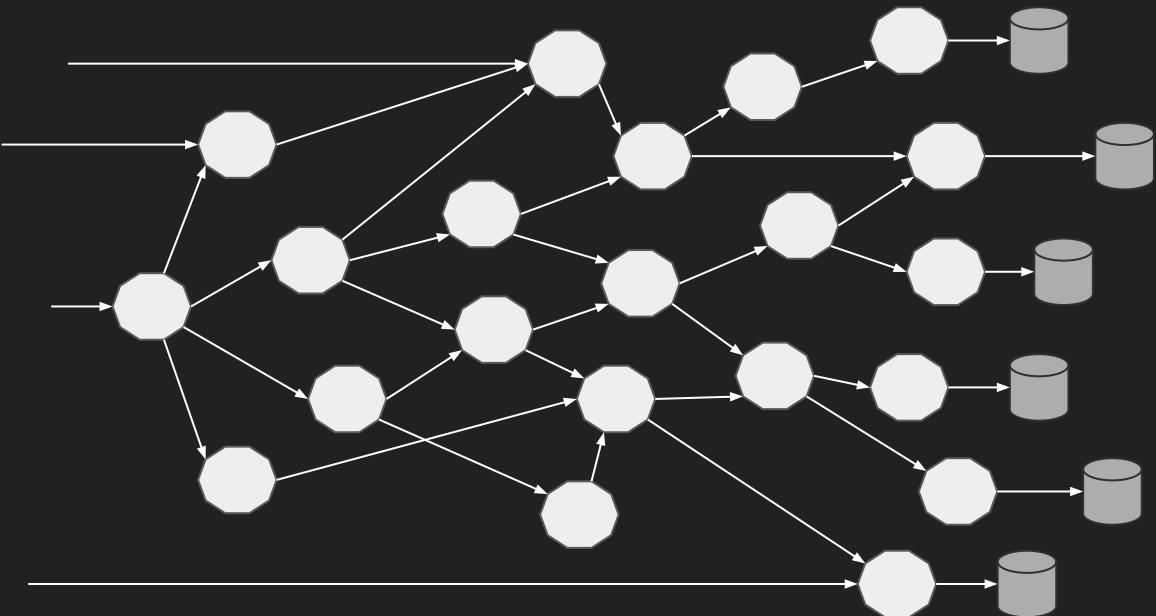
1. Deployment **Independence** - updates to an individual microservice have no negative impact to any other component of the system. Optimized for **Replacement**
2. Organized around **business** capabilities
3. **Products** not Projects
4. API Focused
5. Smart endpoints and dumb pipes
6. Decentralized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

Old School



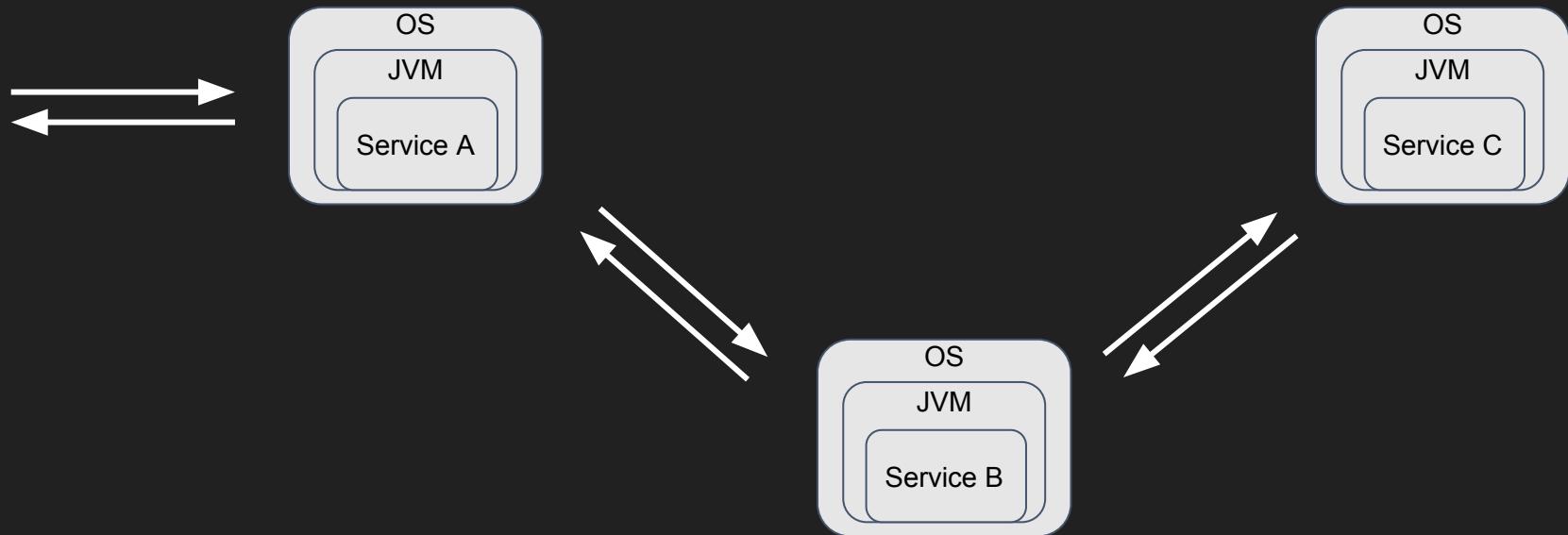
Love Thy Mono

New School



OPENSHIFT

Microservices == Distributed Computing

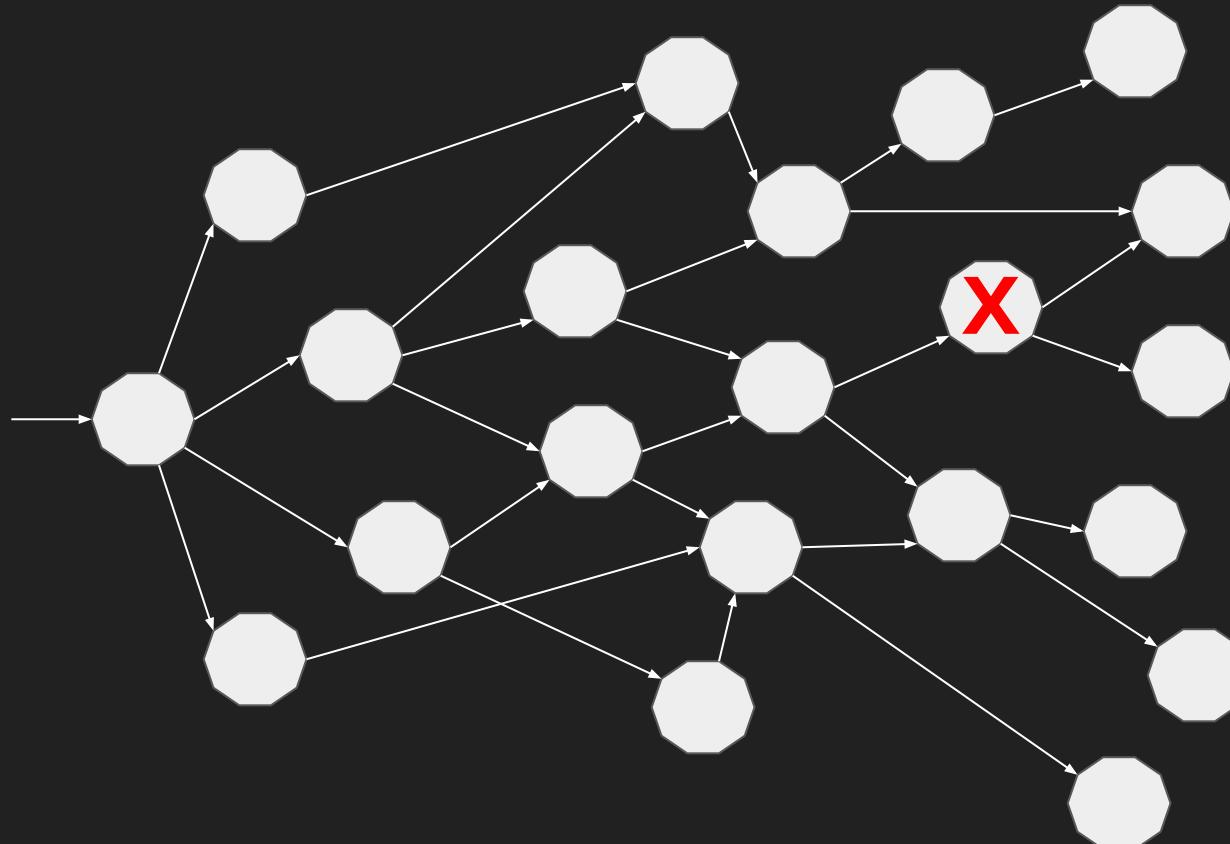


Fallacies of Distributed Computing

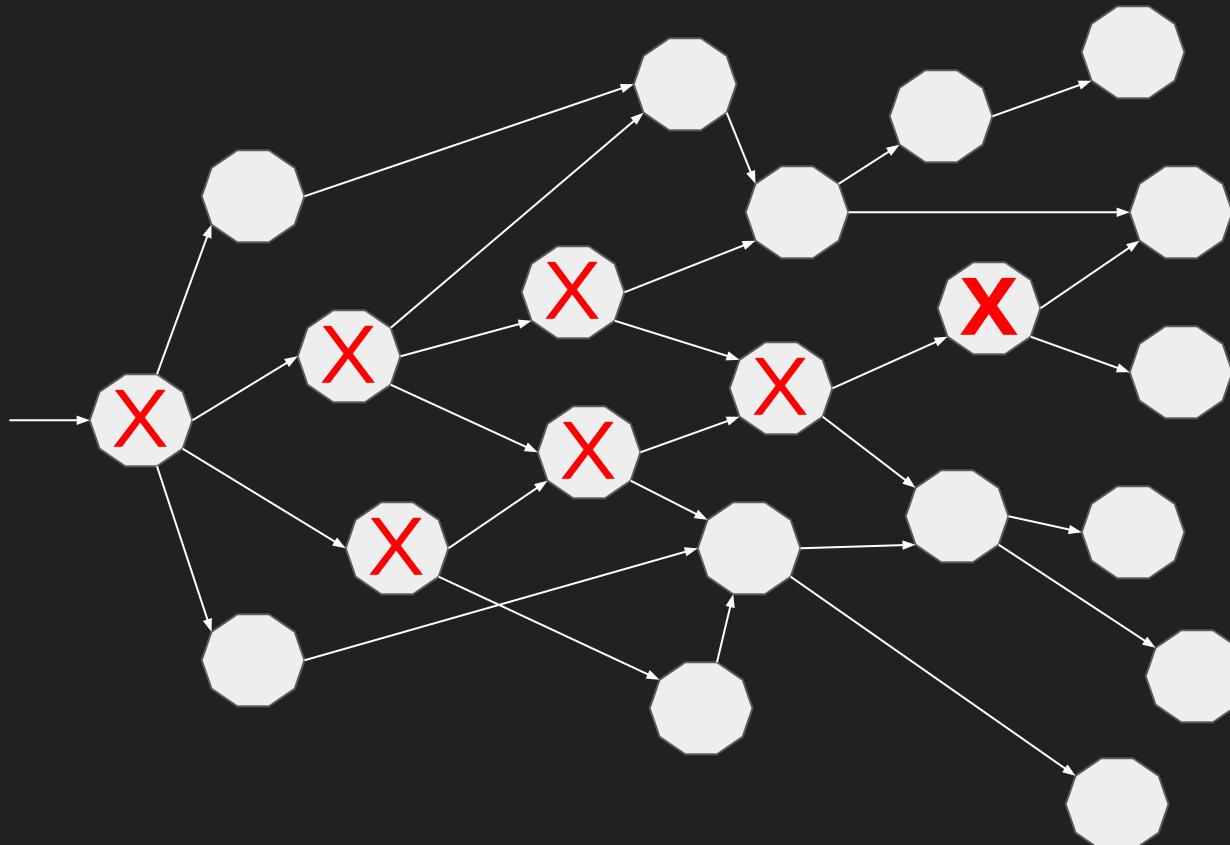
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

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

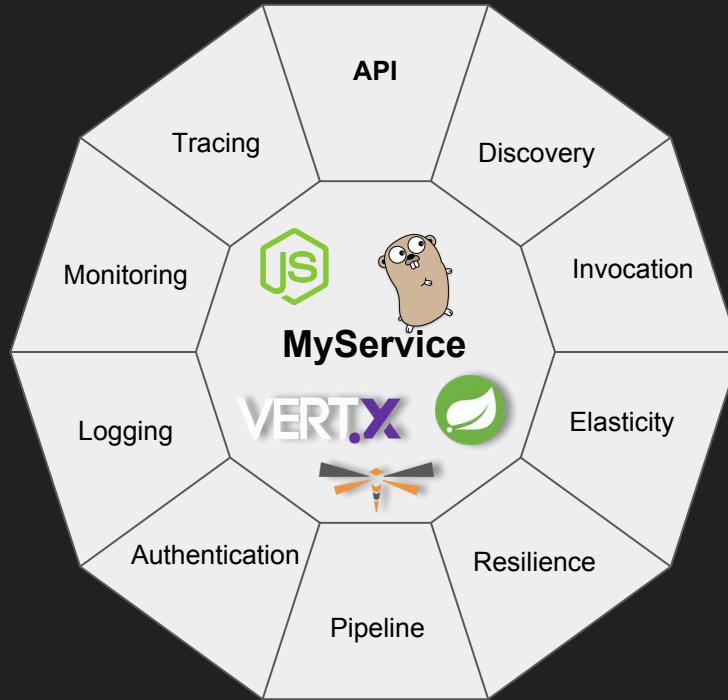
Failure of a Service



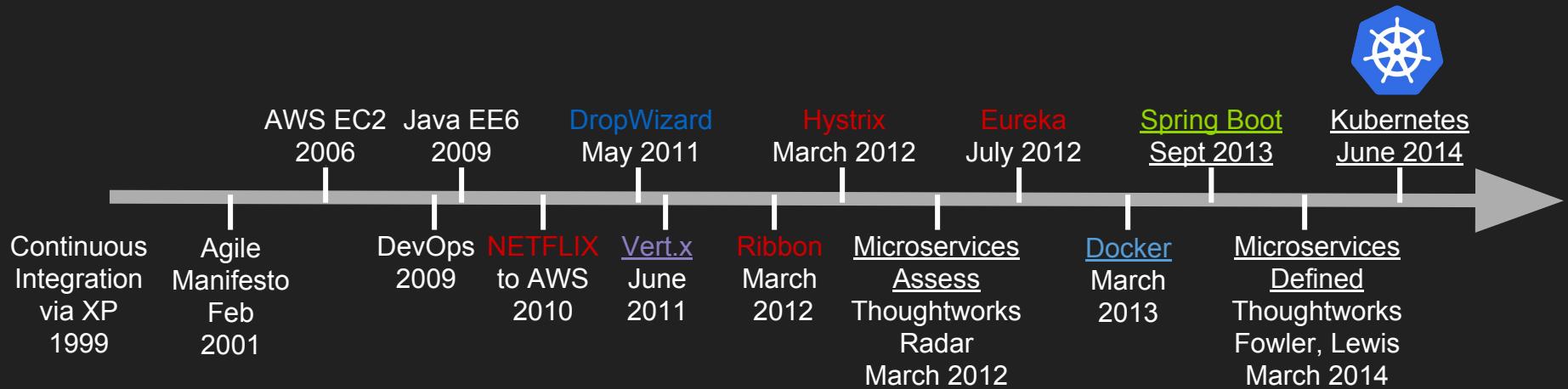
Cascading Failure



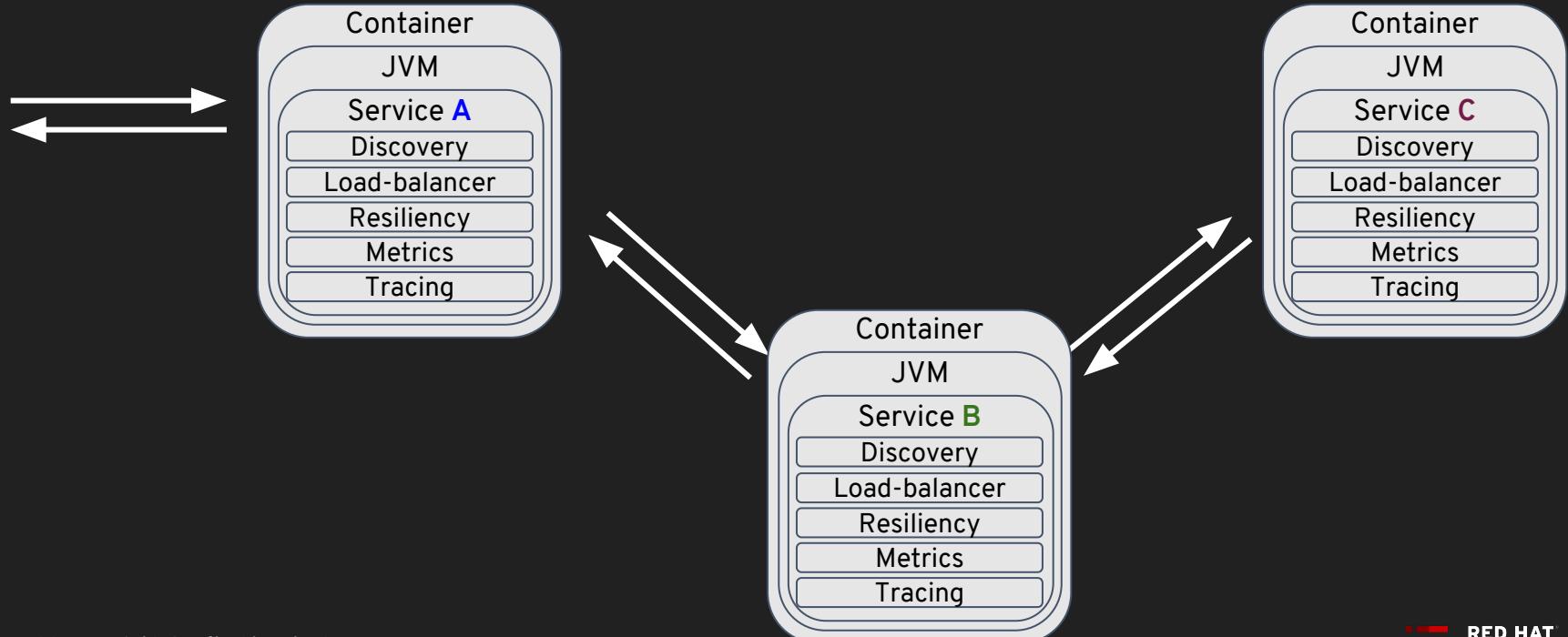
Microservices'ilities



History of Microservices



Microservices embedding Capabilities



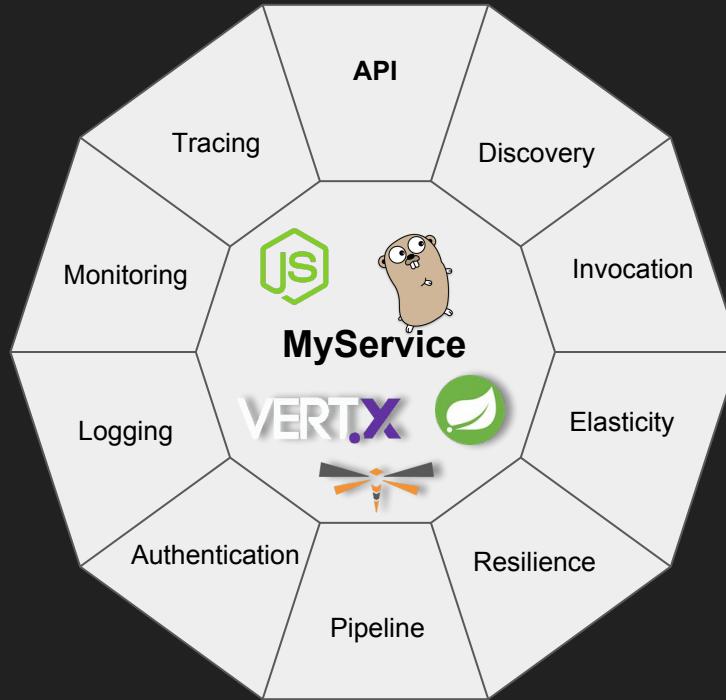
What's Wrong with Netflix OSS?

Java Only

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



Microservices'ilities



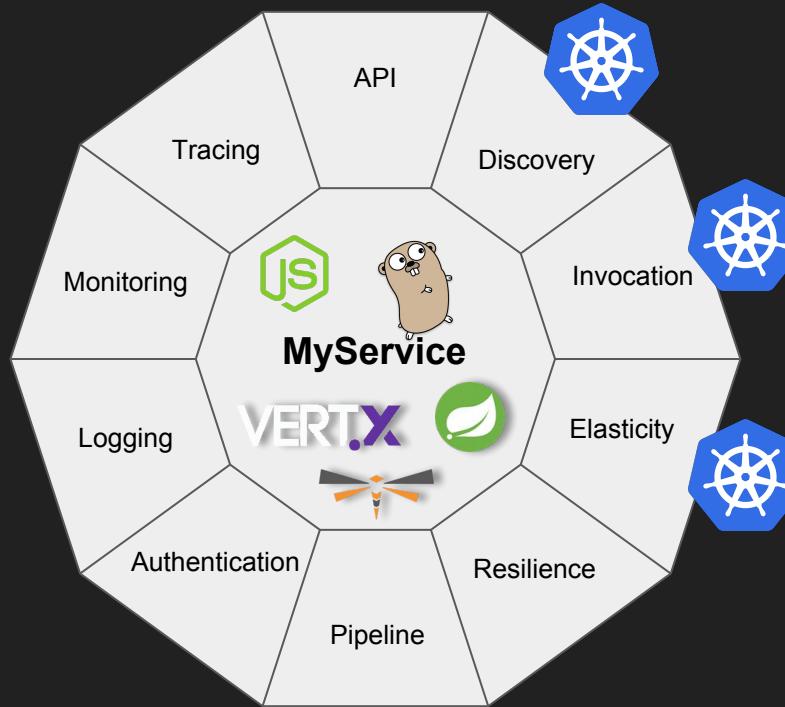


OPENSIFT

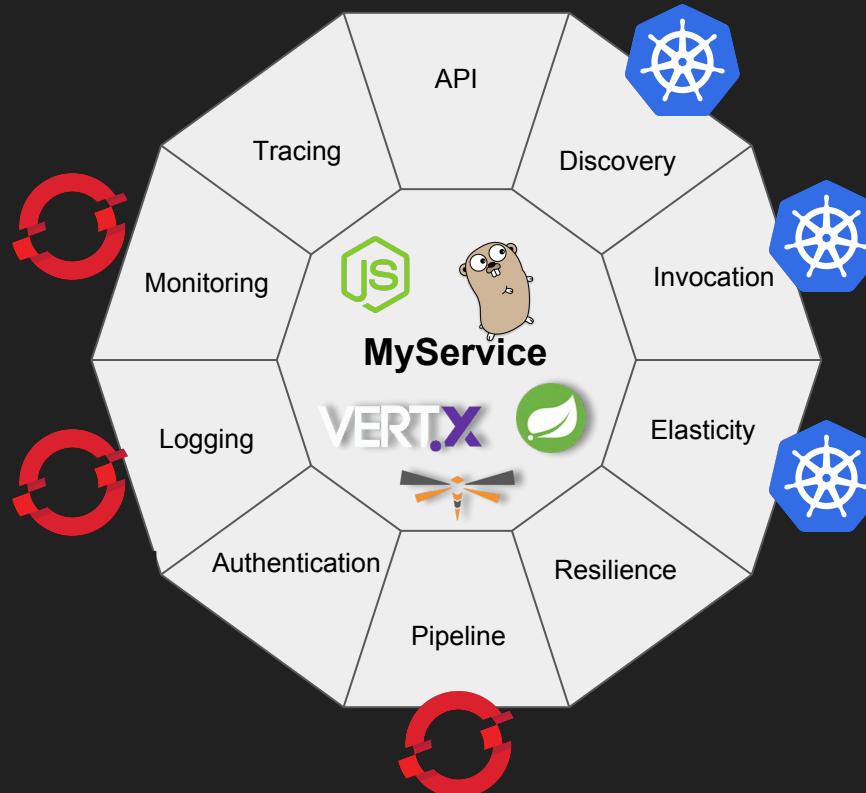
Kubernetes/OpenShift Re-cap Demo

`mvn package, docker build, kubectl apply -f deploy.yml`

Microservices'ilities + Kubernetes



Microservices'ilities + OpenShift





Istio - Sail

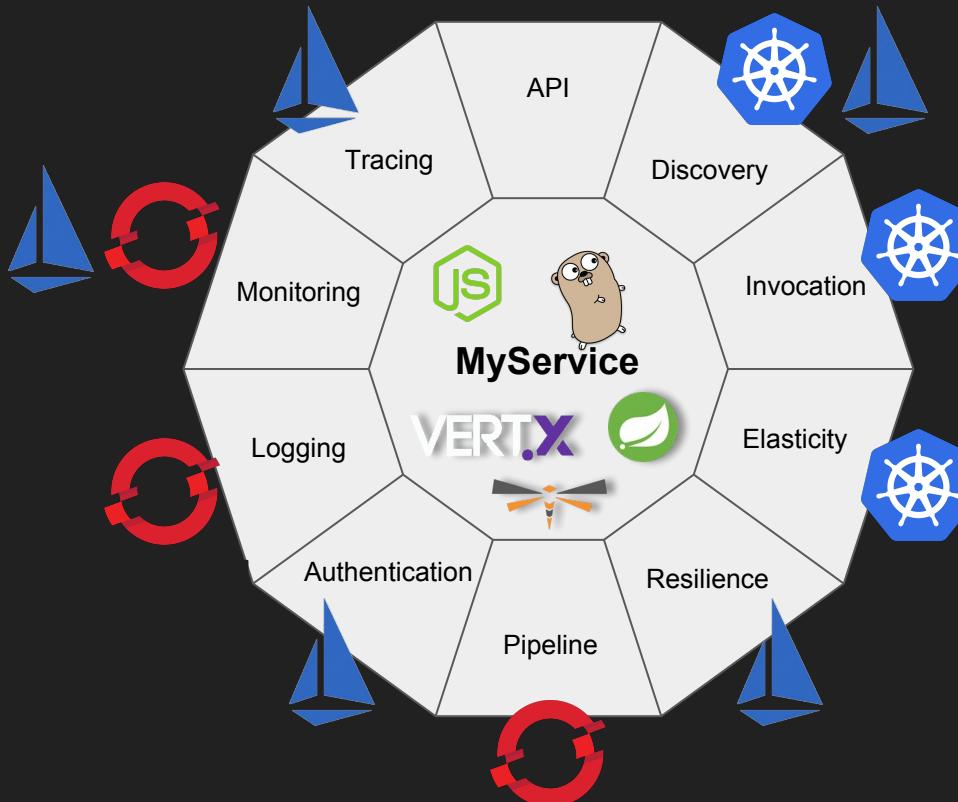
(Kubernetes - Helmsman or ship's pilot)

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



Observability



Istio Dashboard

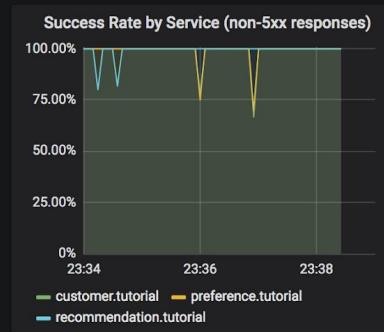
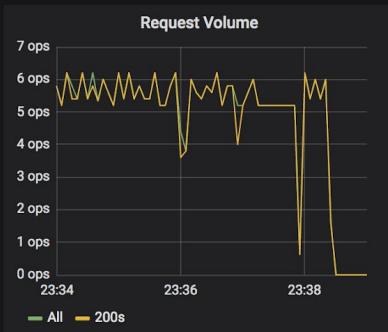


Last 5 minutes Refresh every 5s



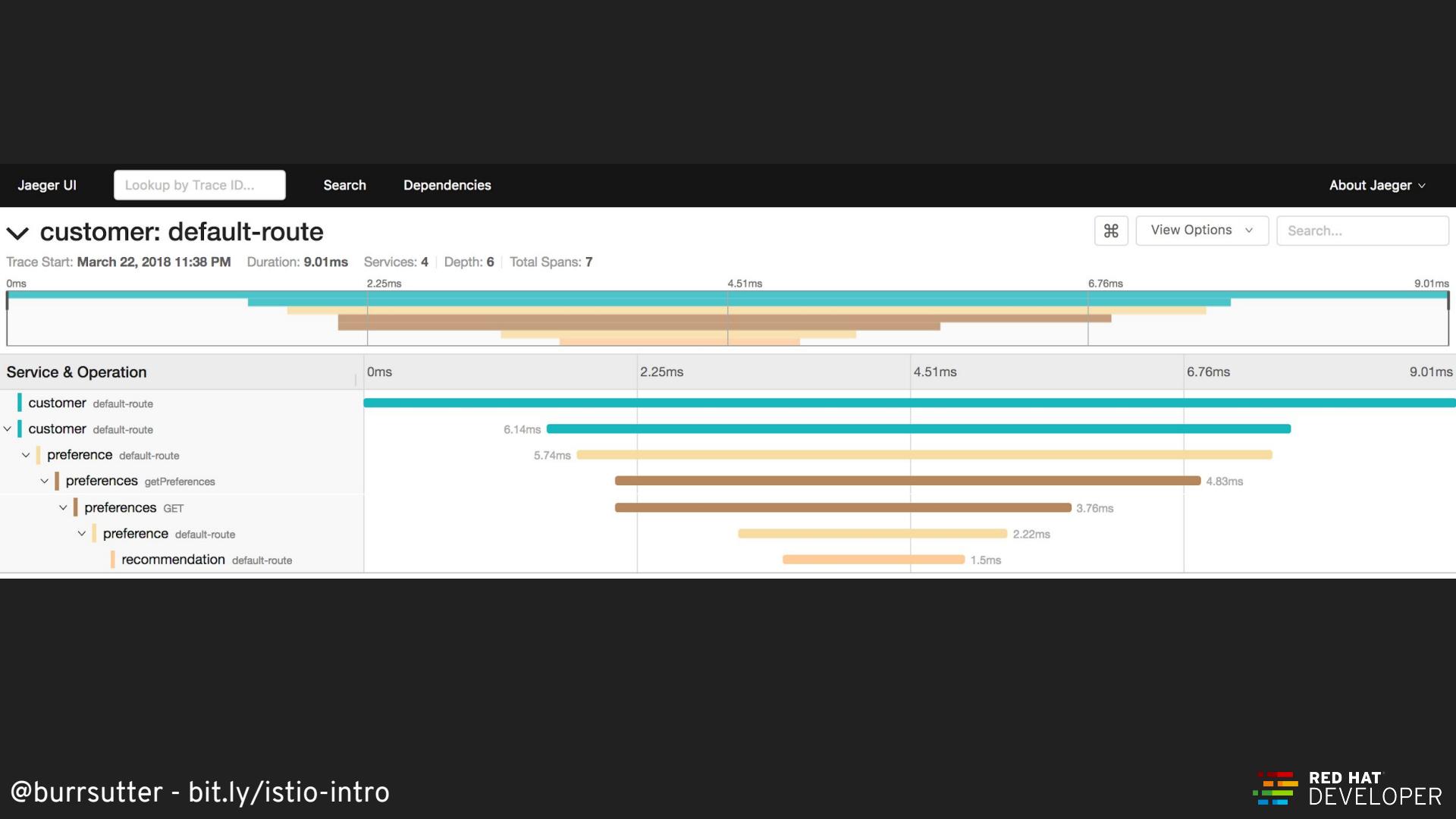
Service Mesh

Service Mesh

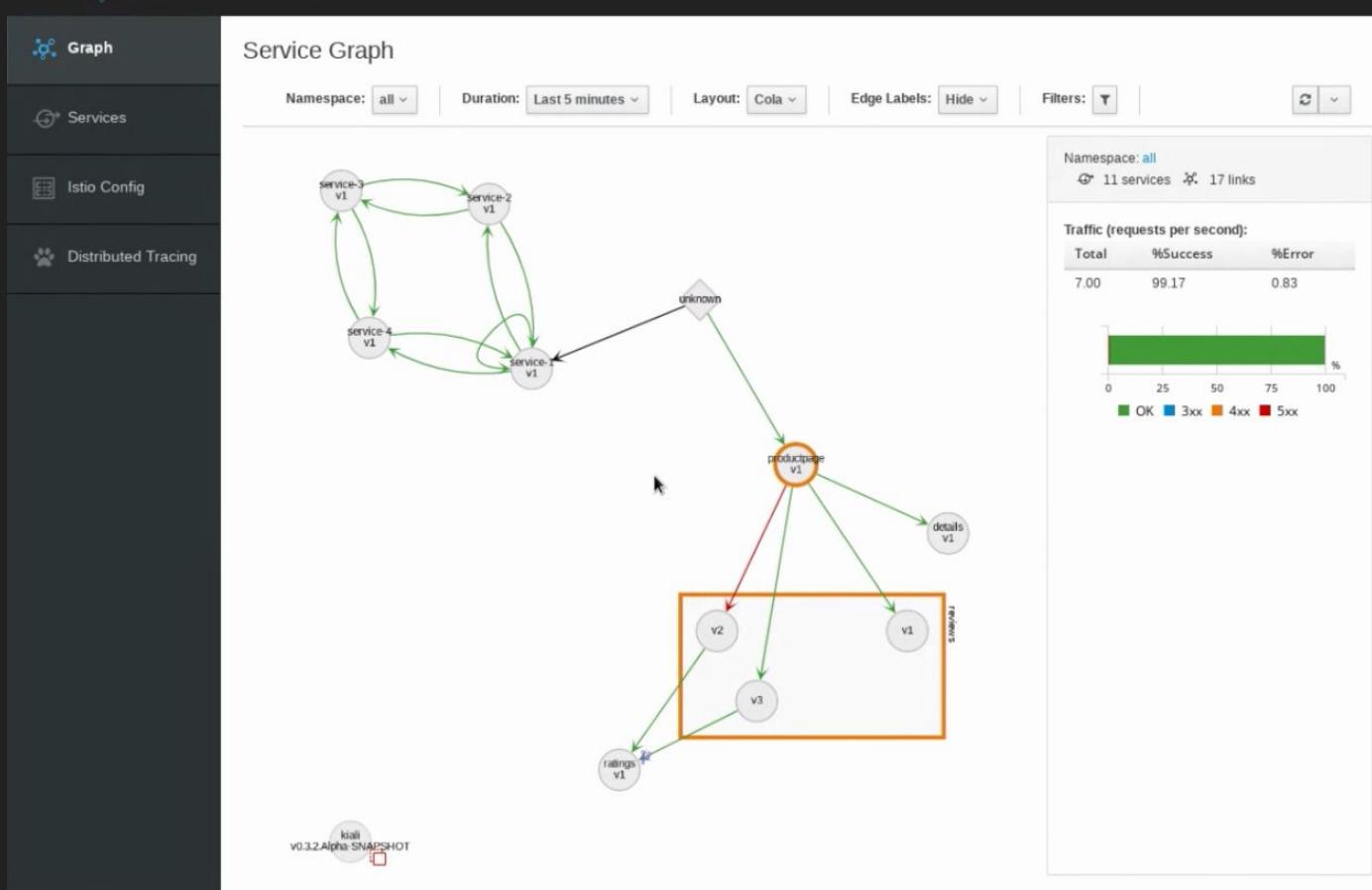


Services

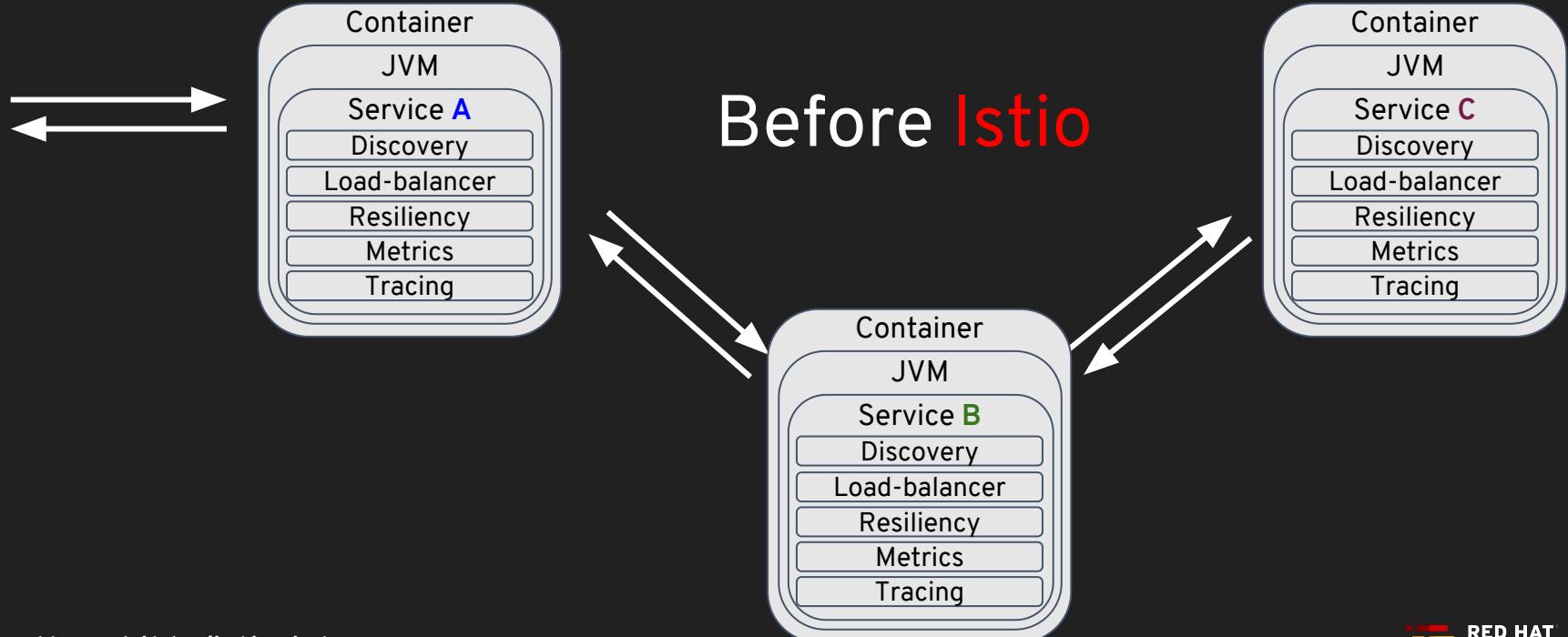
HTTP Services



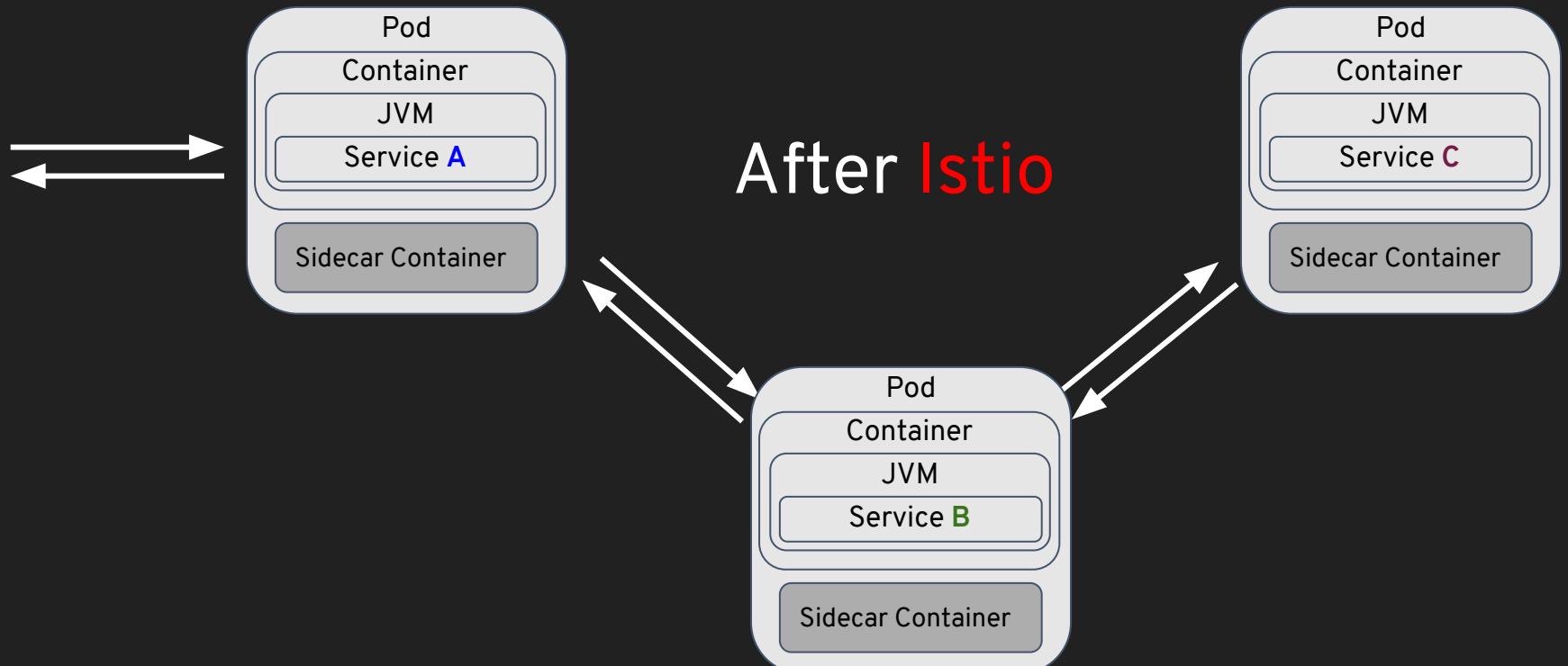
Kiali.io New Service Graph



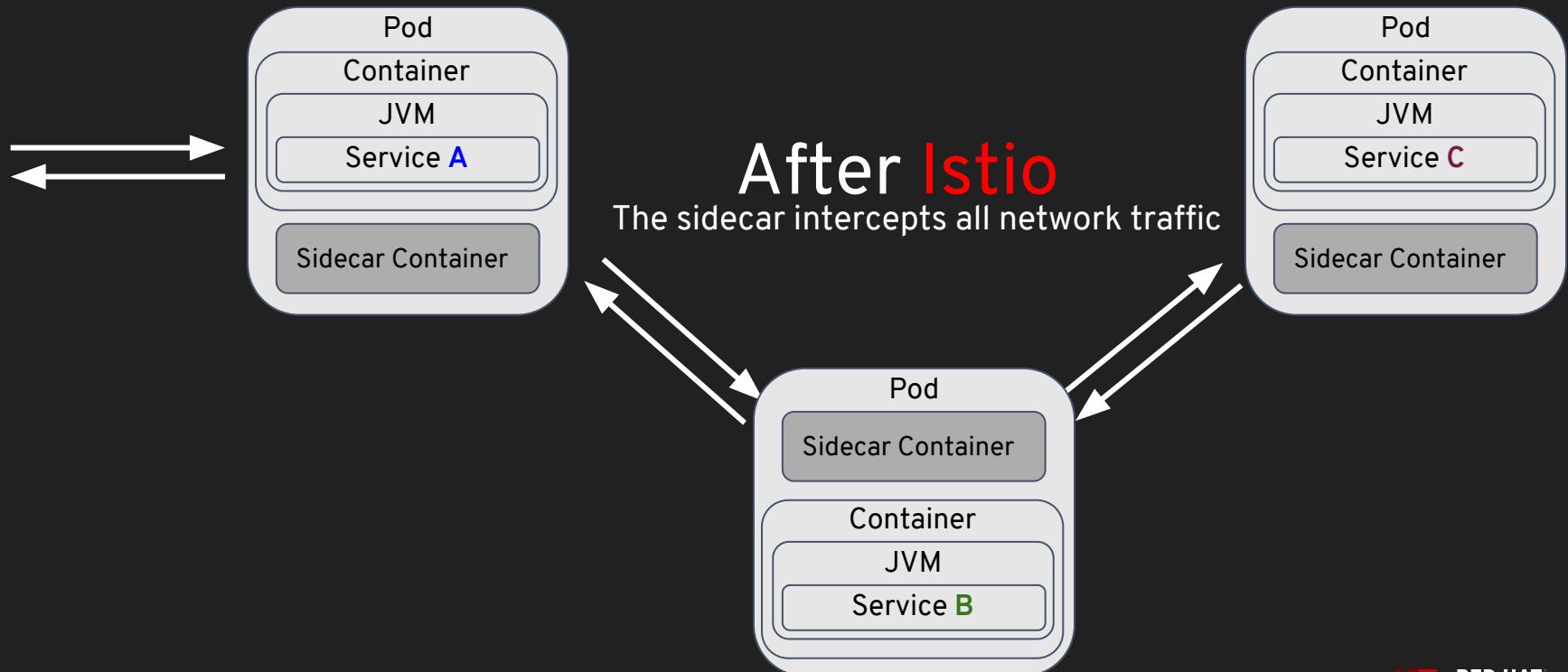
Microservices embedding Capabilities



Microservices externalizing Capabilities



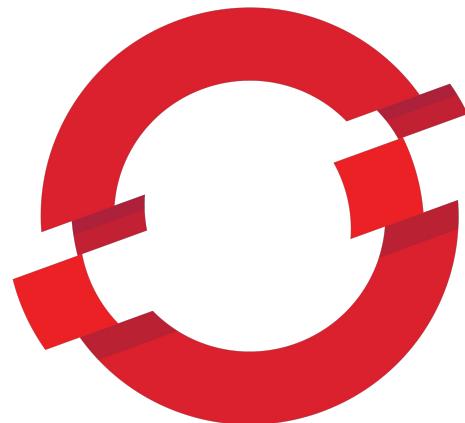
Microservices externalizing Capabilities





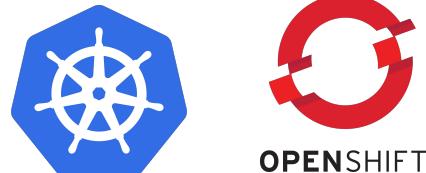
<https://www.imz-ural.com/blog/waffles-the-sidecar-dog>

Better Microservices Platform circa 2018

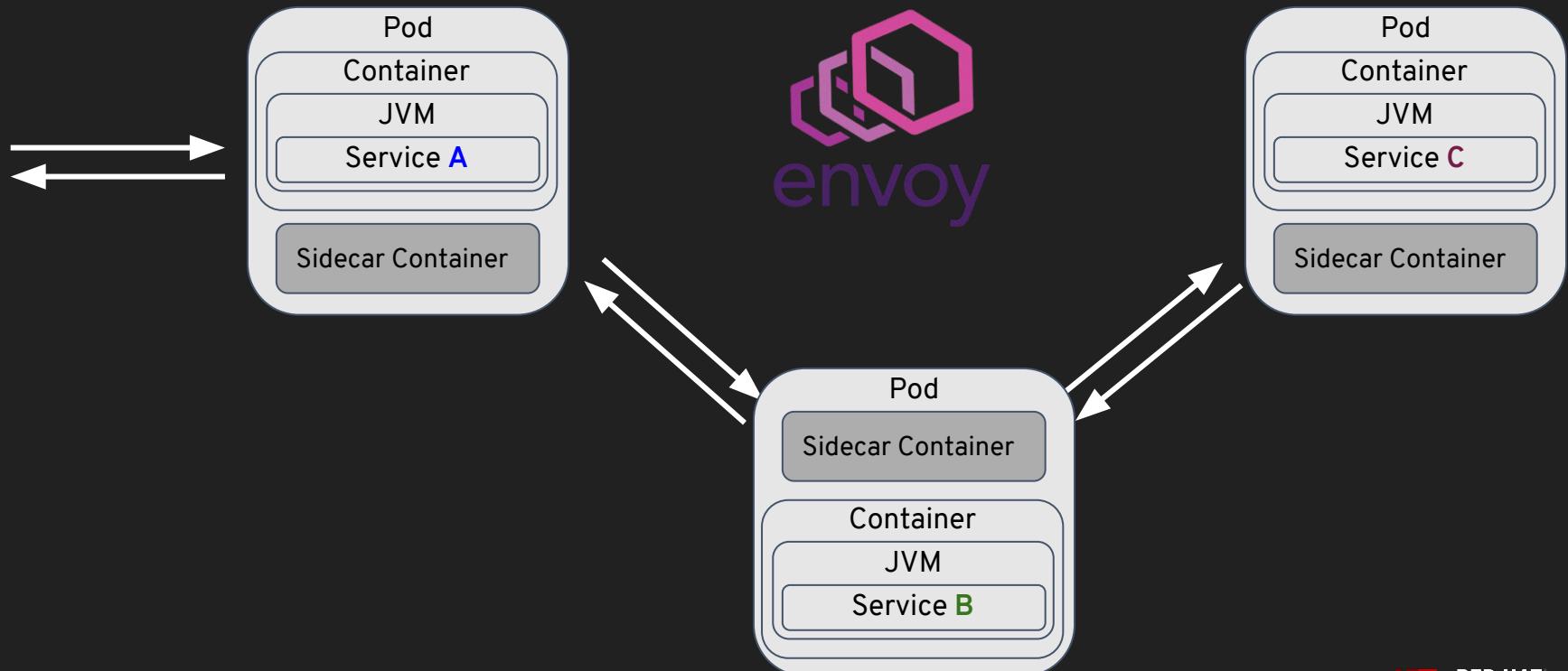


OPENSIFT

Polyglot Microservices Platform circa 2018



Envoy is the current sidecar

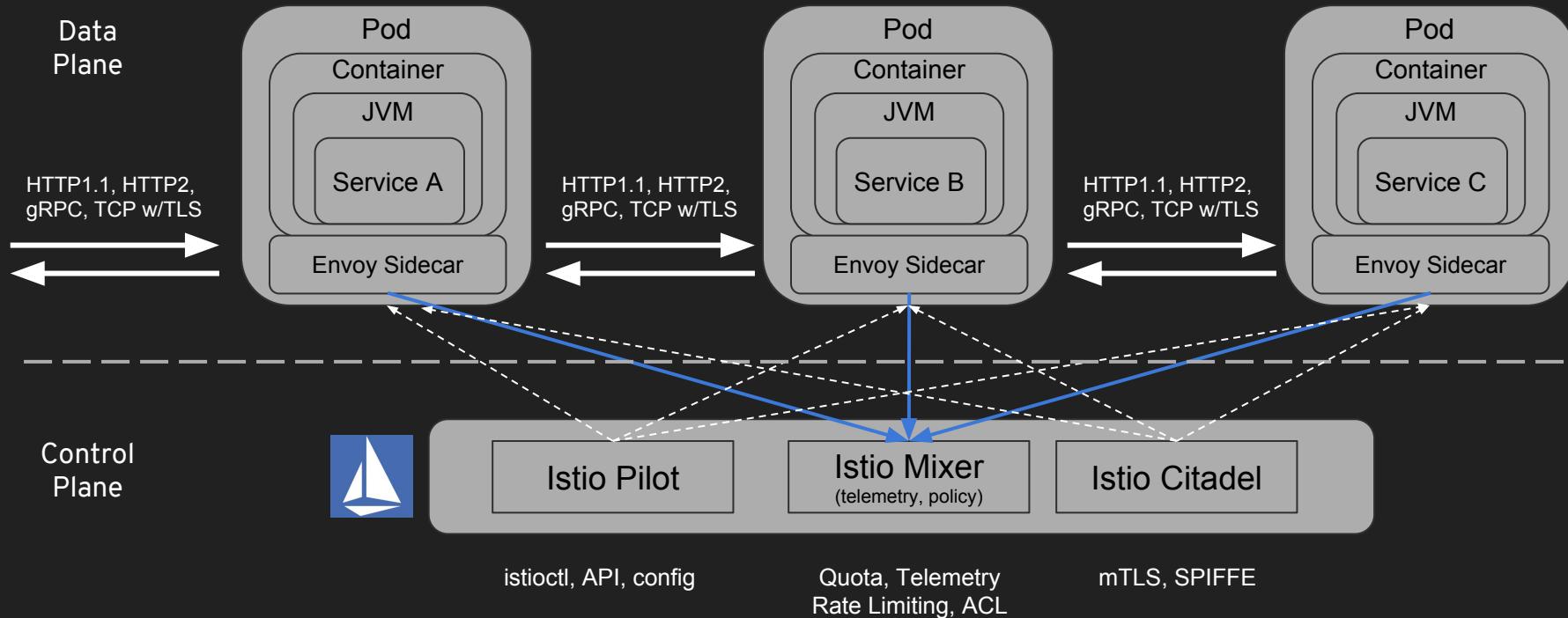


Next Generation Microservices - Service Mesh

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



kubectl get crds

Adapters.config.istio.io	Metrics.config.istio.io
Apikeys.config.istio.io	Noops.config.istio.io
Attributemanifests.config.istio.io	Opas.config.istio.io
Authorizations.config.istio.io	Policies.authentication.istio.io
Bypasses.config.istio.io	Prometheuses.config.istio.io
Checknothings.config.istio.io	Quotas.config.istio.io
Circonuses.config.istio.io	Quotaspecbindings.config.istio.io
Cloudwatches.config.istio.io	Quotaspecs.config.istio.io
Deniers.config.istio.io	Rbacconfigs.rbac.istio.io
Destinationrules .networking.istio.io	Rbacs.config.istio.io
Dogstatsds.config.istio.io	Redisquotas.config.istio.io
Edges.config.istio.io	Reportnothings.config.istio.io
Envoyfilters.networking.istio.io	Rules .config.istio.io
Fluentds.config.istio.io	Servicecontrolreports.config.istio.io
Gateways .networking.istio.io	Servicecontrols.config.istio.io
Handlers.config.istio.io	Serviceentries.networking.istio.io
Httpapispecbindings.config.istio.io	Servicerolebindings.rbac.istio.io
Httpapispecs.config.istio.io	Serviceroles.rbac.istio.io
Instances.config.istio.io	Signalfxs.config.istio.io
Kubernetesenvs.config.istio.io	Solarwindses.config.istio.io
Kuberneteses.config.istio.io	Stackdrivers.config.istio.io
Listcheckers.config.istio.io	Statsds.config.istio.io
Listentries.config.istio.io	Stdios.config.istio.io
Logentries.config.istio.io	Templates.config.istio.io
Memquotas.config.istio.io	Tracespans.config.istio.io
Meshpolicies.authentication.istio.io	Virtualservices .networking.istio.io
Metrics.config.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

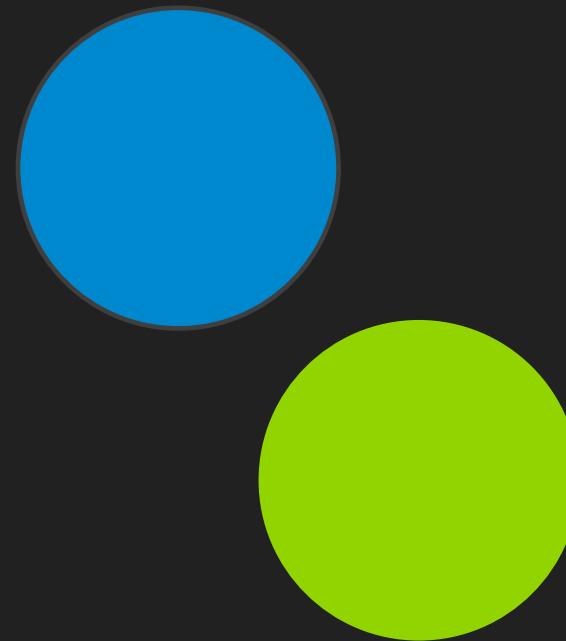
bit.ly/istio-tutorial

Traffic Control

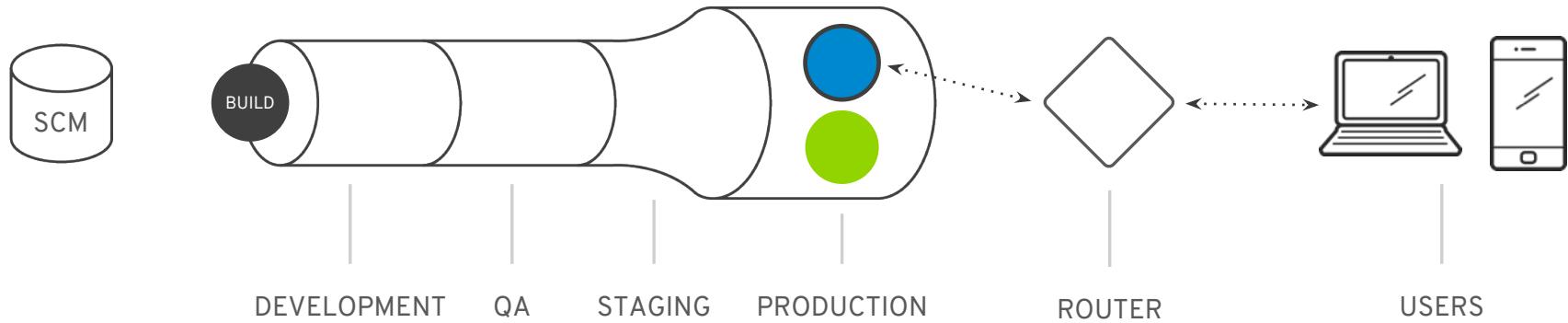
Traffic Control

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

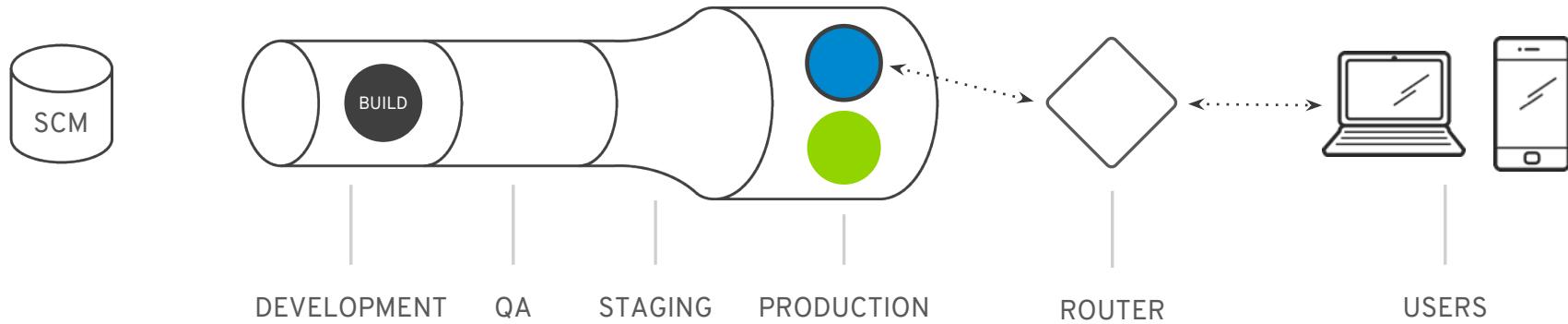
Blue/Green Deployment



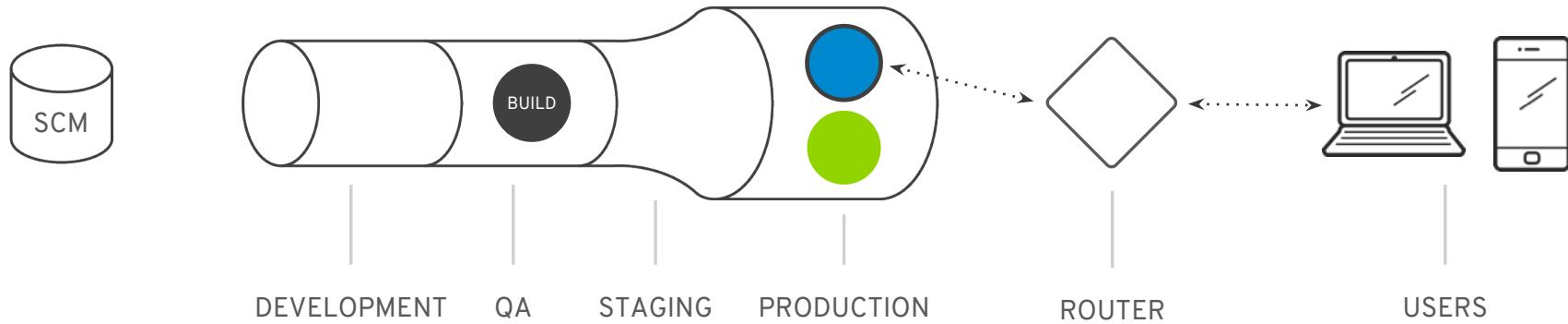
Blue/Green Deployment



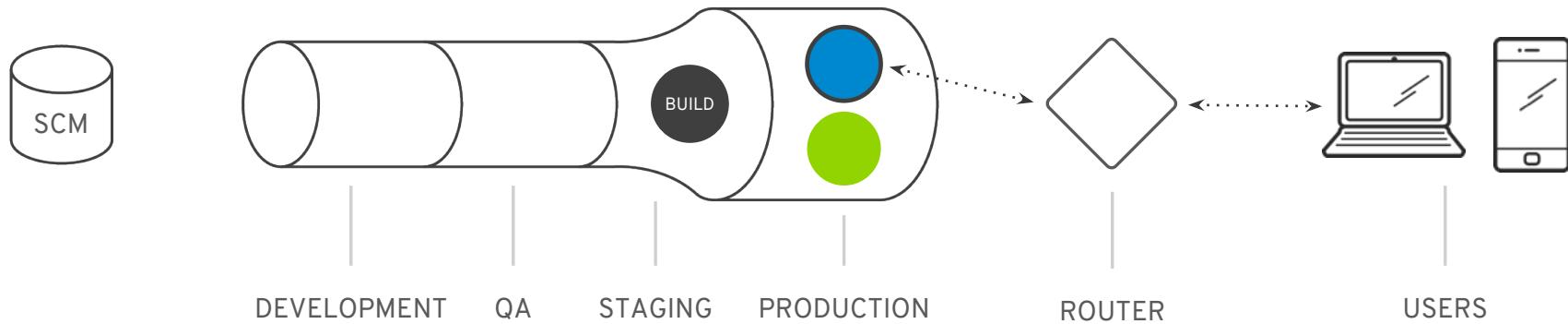
Blue/Green Deployment



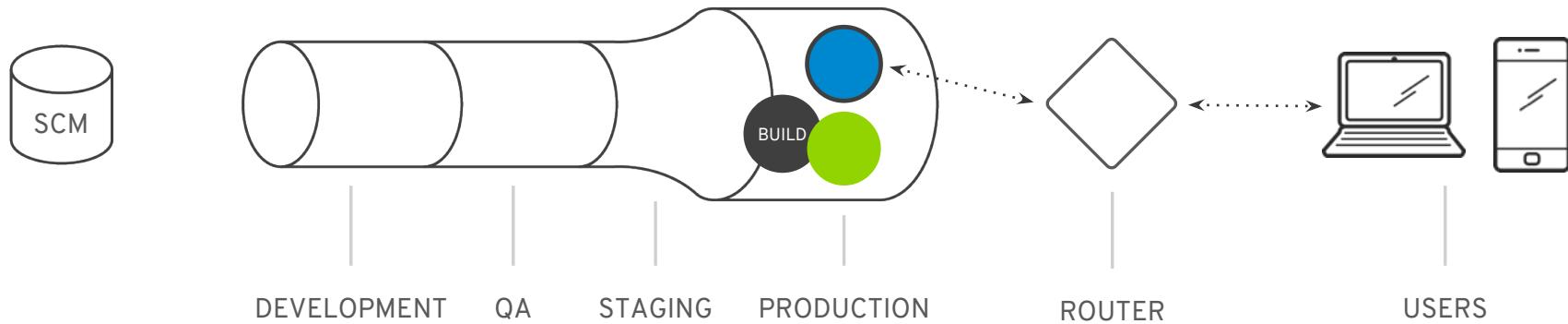
Blue/Green Deployment



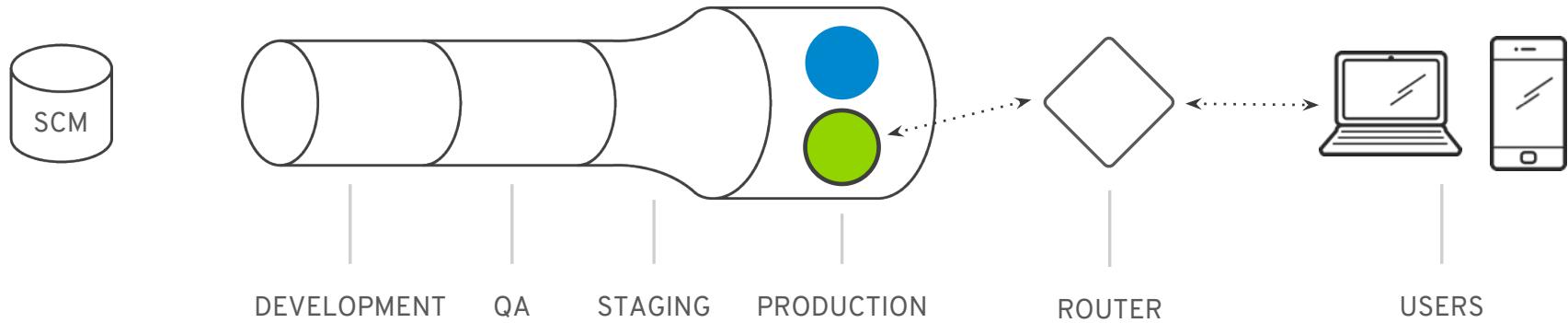
Blue/Green Deployment



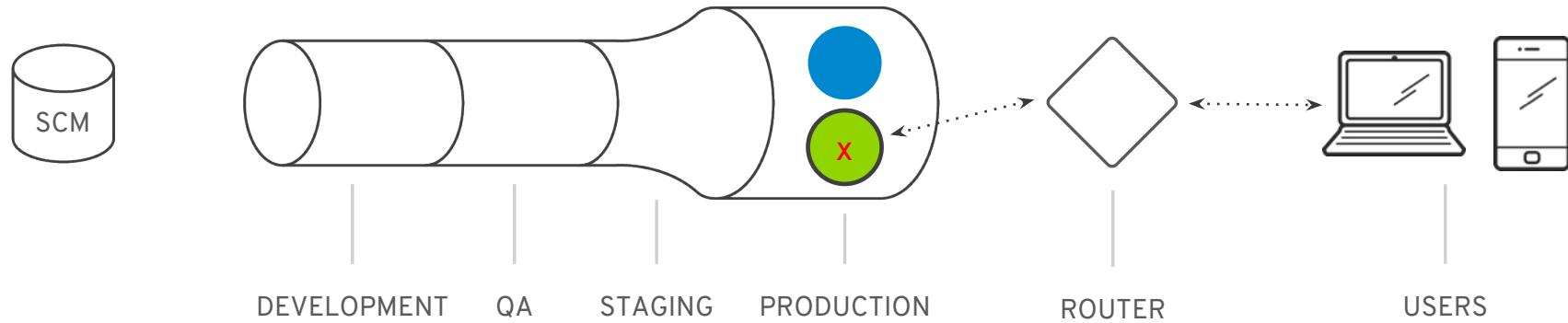
Blue/Green Deployment



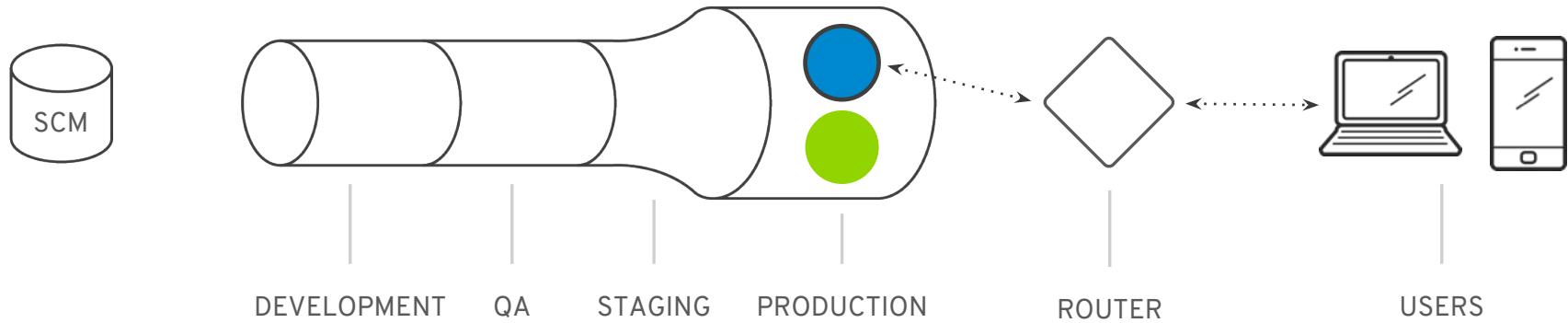
Blue/Green Deployment



Blue/Green Deployment



Blue/Green Deployment

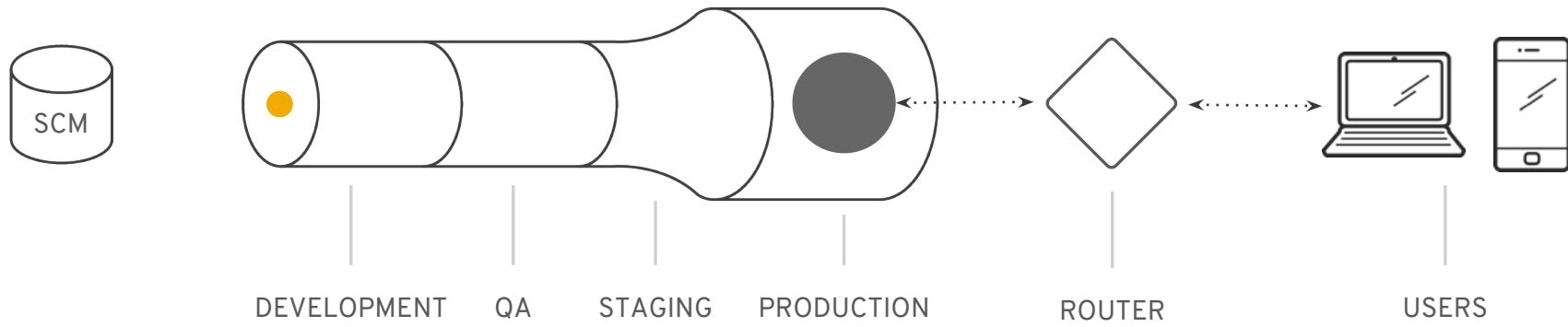


Canary Deployment

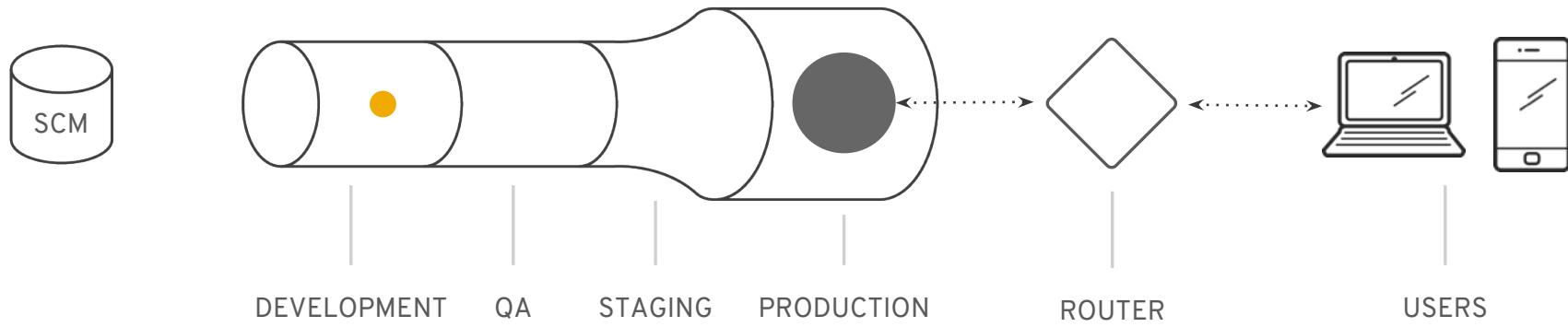




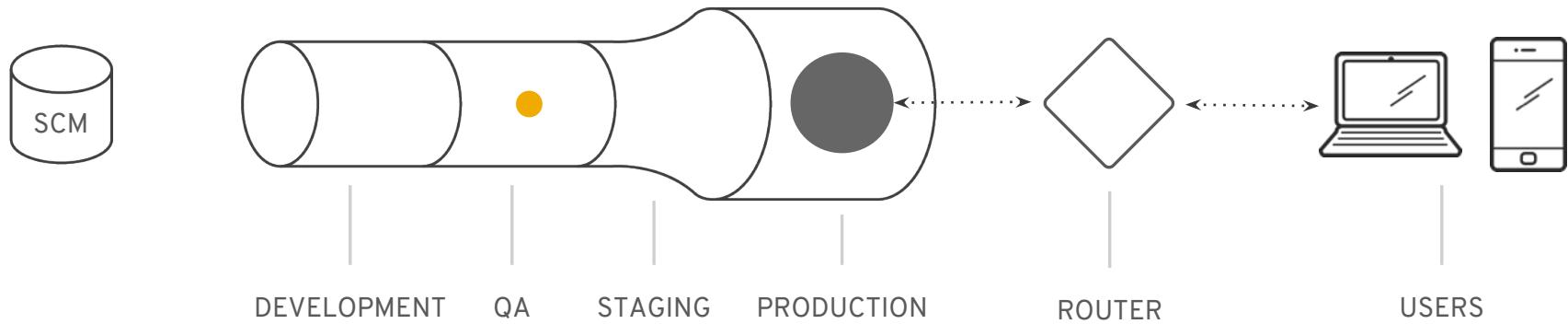
Canary Deployment



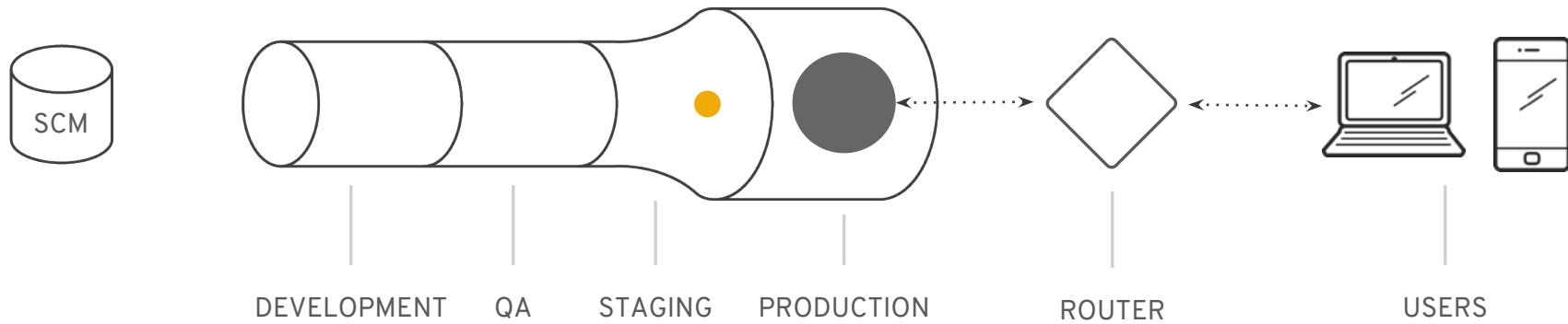
Canary Deployment



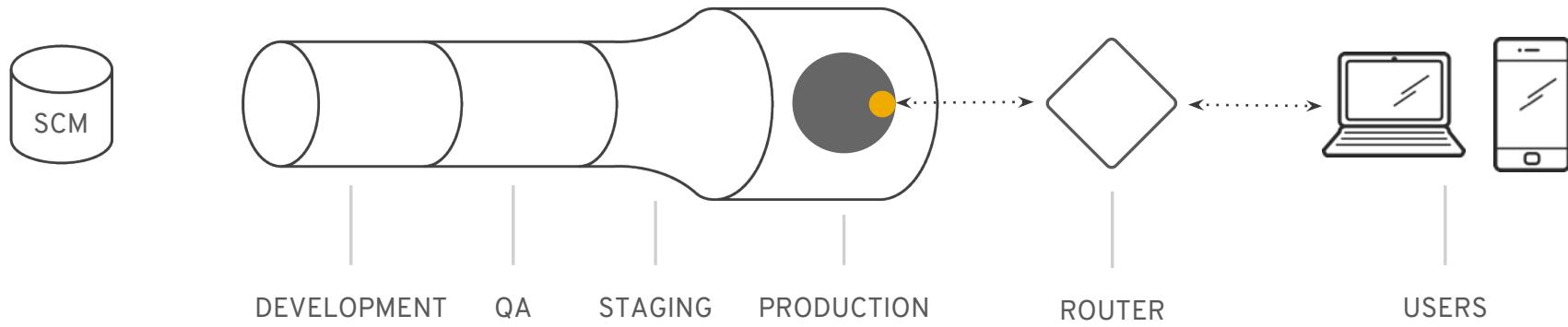
Canary Deployment



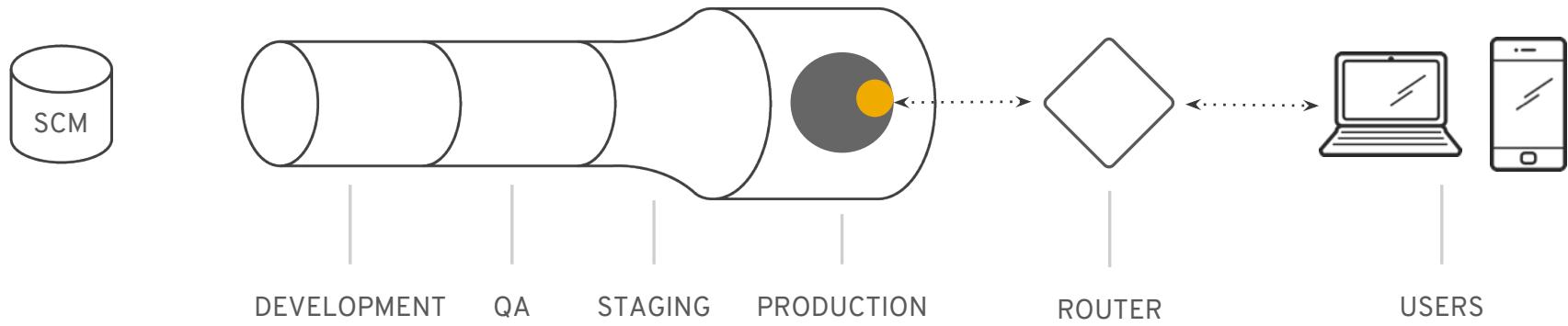
Canary Deployment



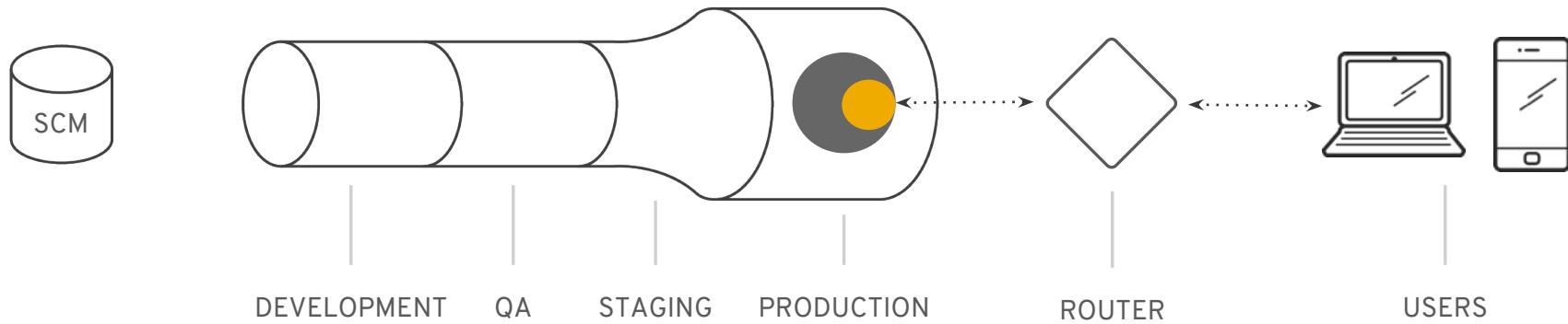
Canary Deployment



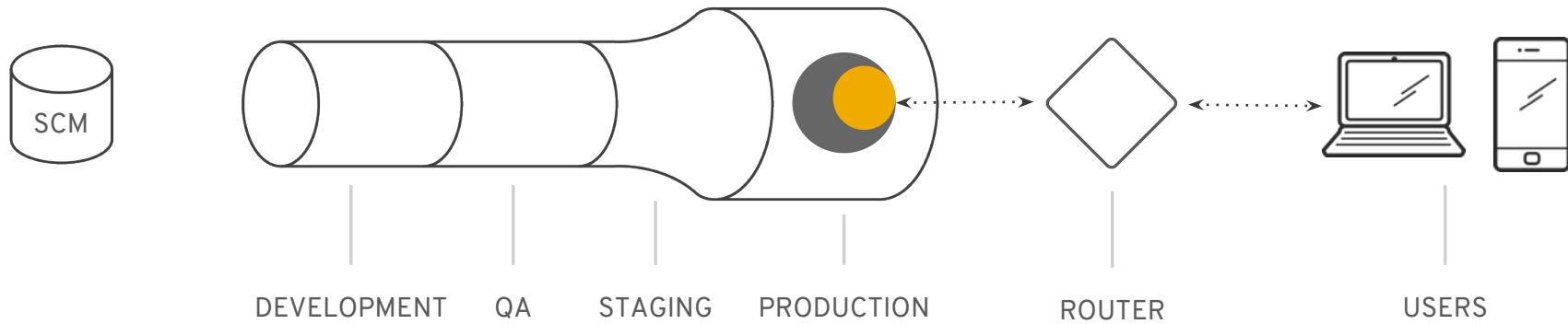
Canary Deployment



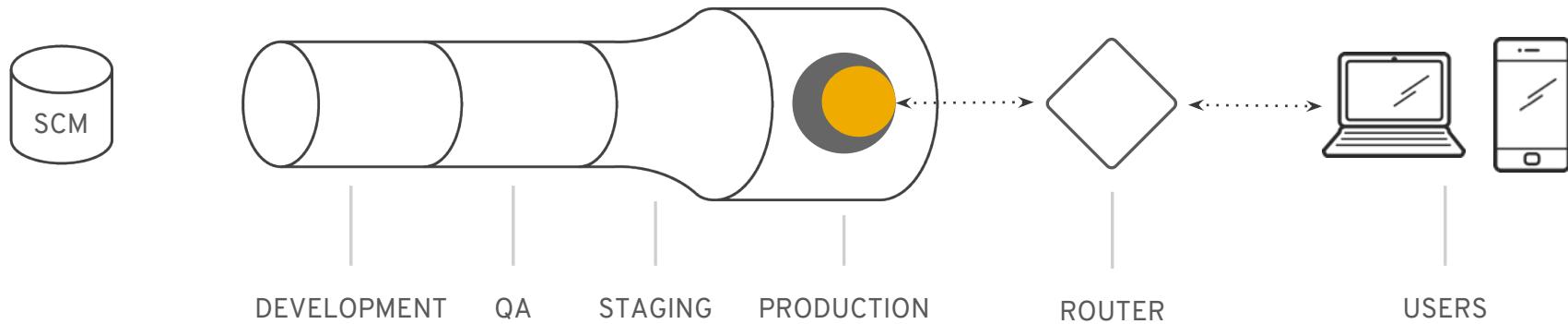
Canary Deployment



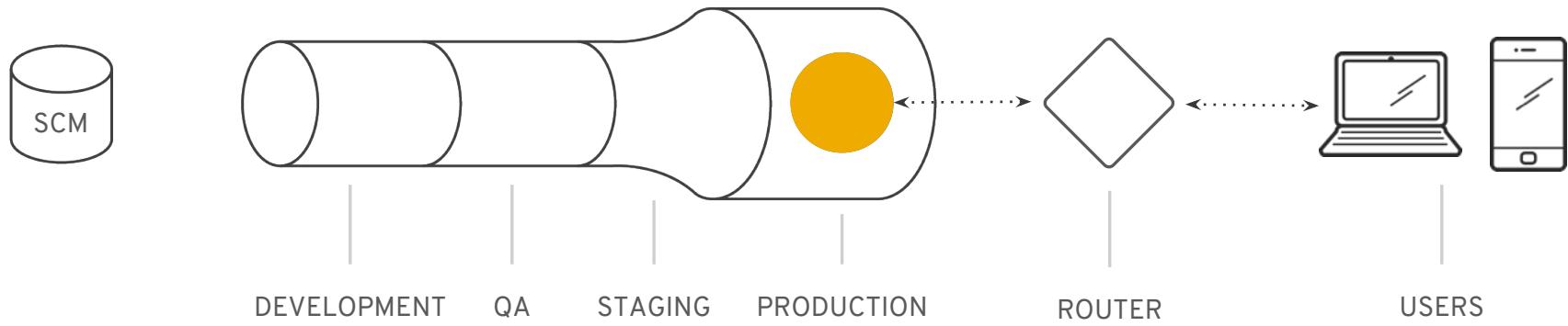
Canary Deployment



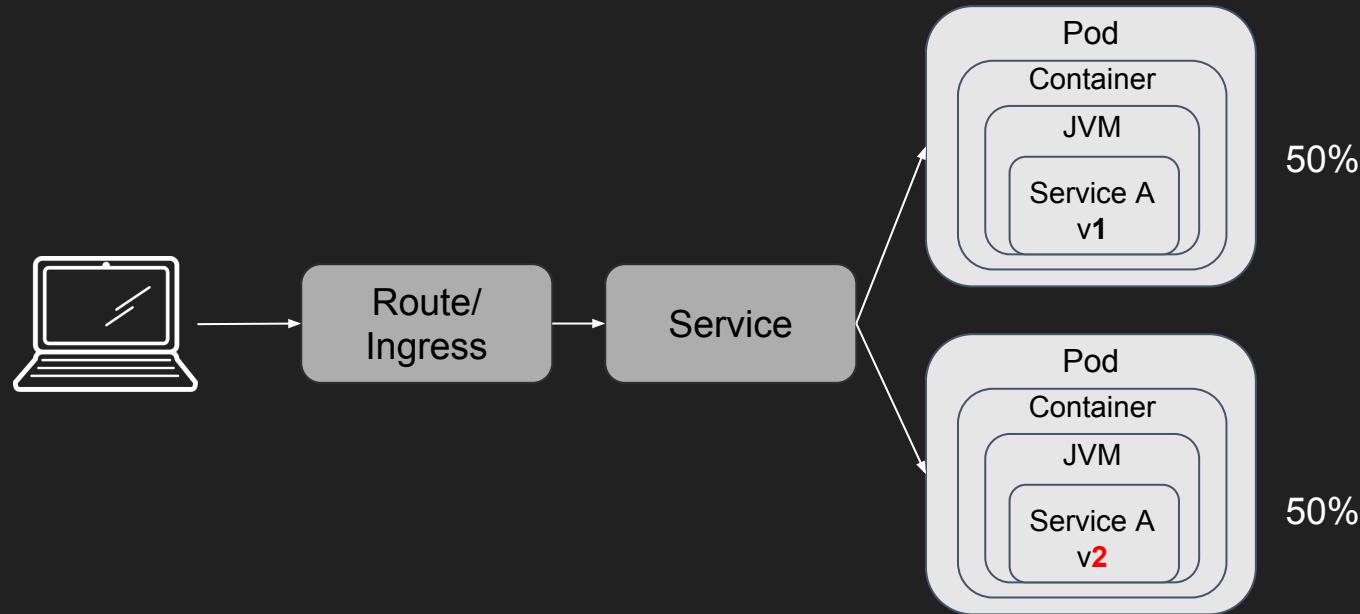
Canary Deployment



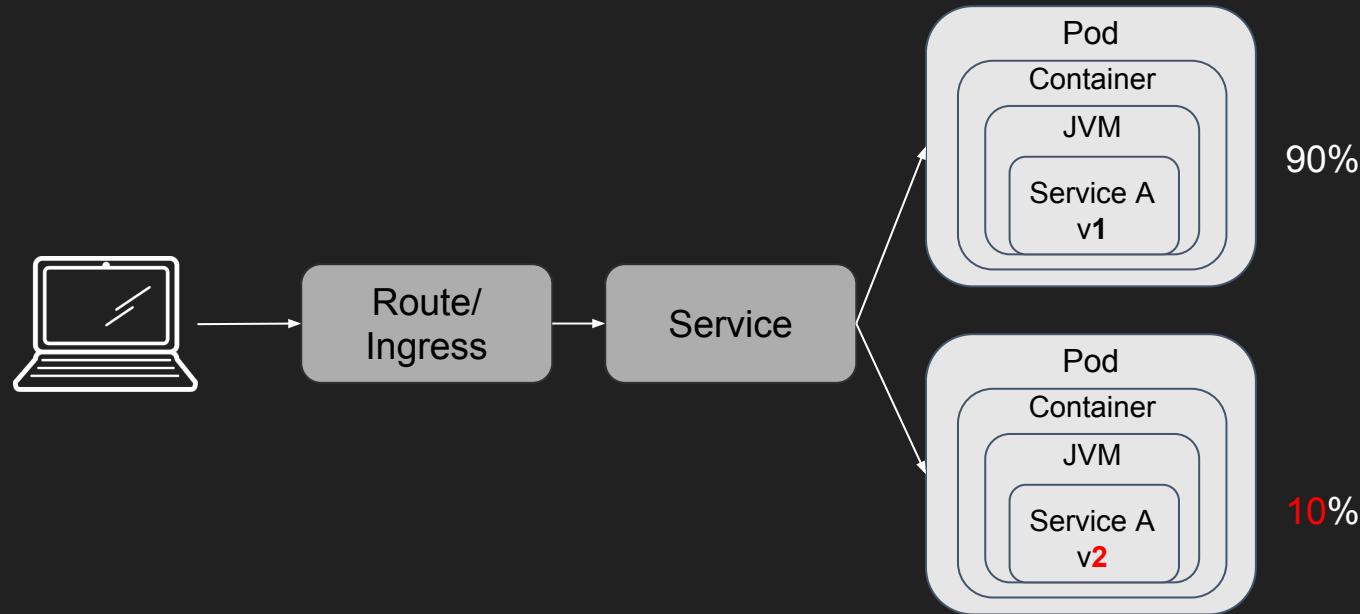
Canary Deployment



Canaries with Kubernetes



Canaries with Istio



Canary Resuscitator

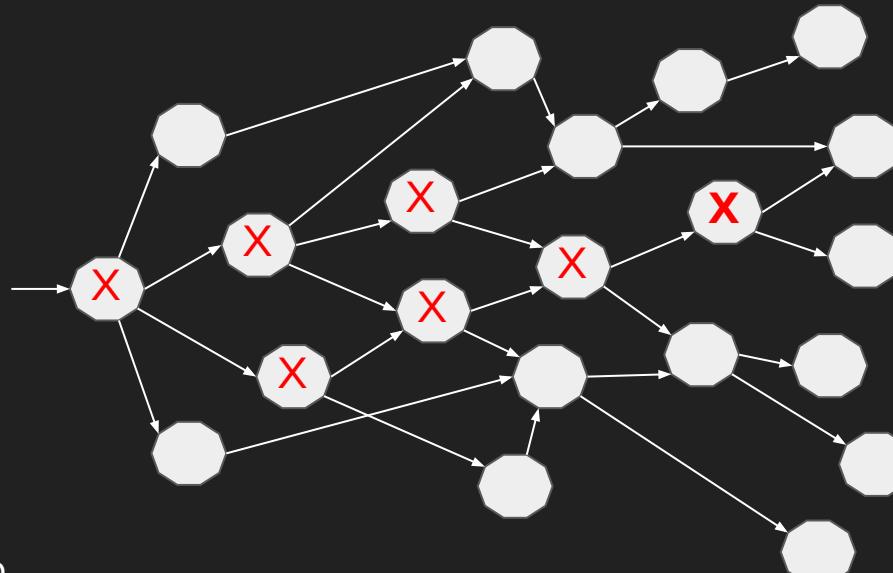


<http://www.openculture.com/2018/05/the-device-invented-to-resuscitate-canaries-in-coal-mines-circa-1896.html>
Thanks to **Paolo Antinori!**

Service Resiliency

Service Resiliency

- Fail Fast: Latency Circuit Breaker
- Pool Ejection
- Circuit Breaker + Pool Ejection + Retry



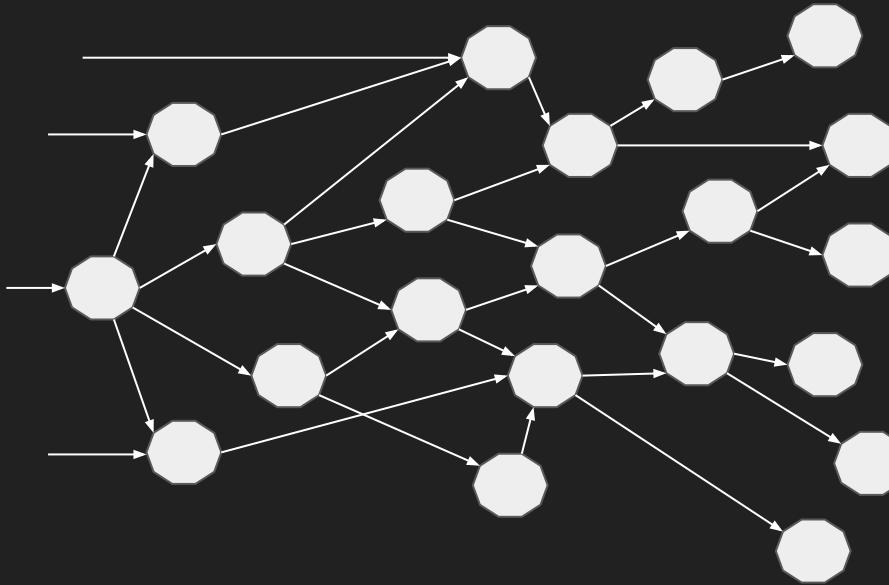
Chaos Testing



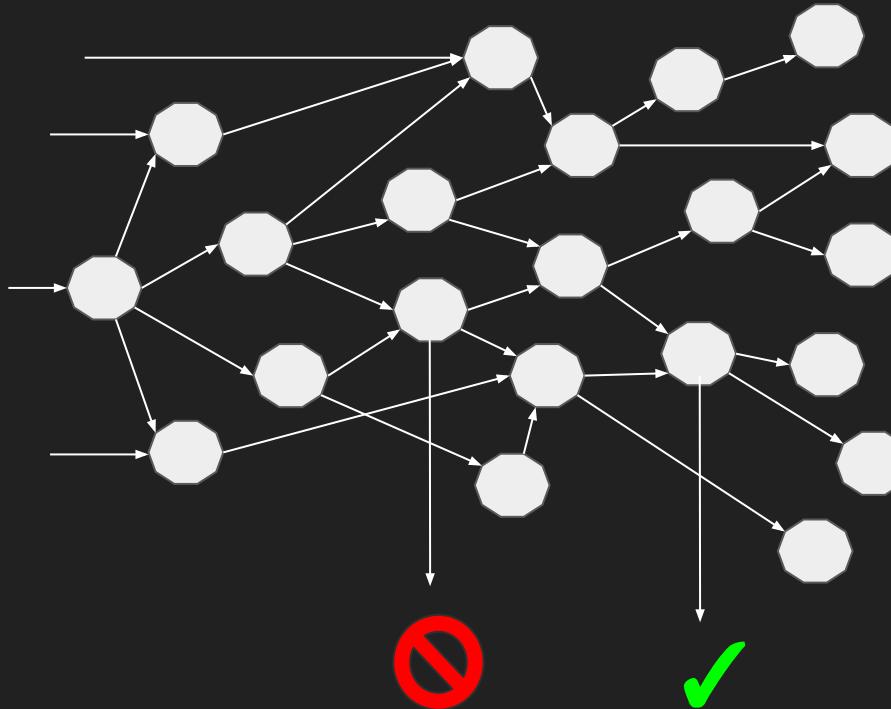
By Netflix - <https://github.com/Netflix/SimianArmy/blob/master/assets/SimianArmy.png>, Apache License 2.0,
@burrsutter - bit.ly/istio-intro <https://commons.wikimedia.org/w/index.php?curid=63503083>

Egress

Most Communication Inbound & Internal



Outbound/Egress Blocked By Default



Security

Why Security?

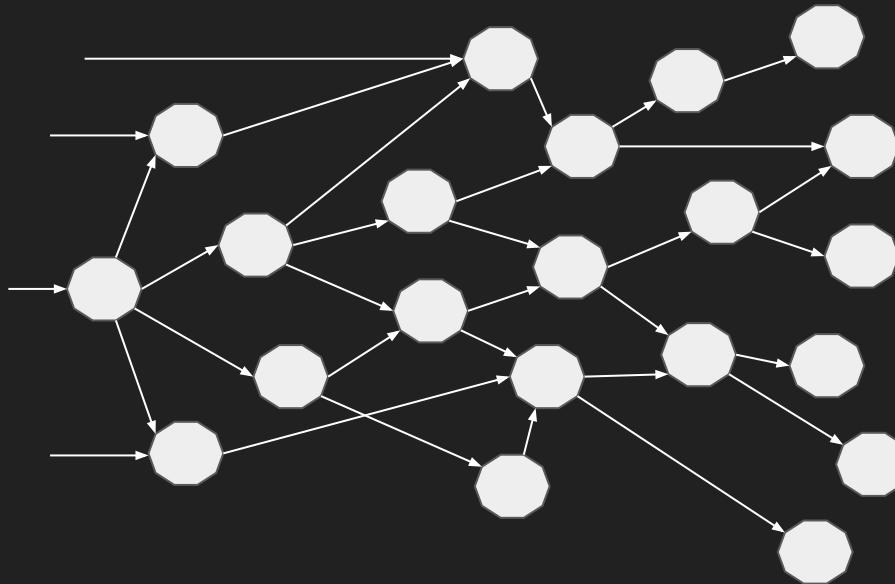
Our Teams:

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

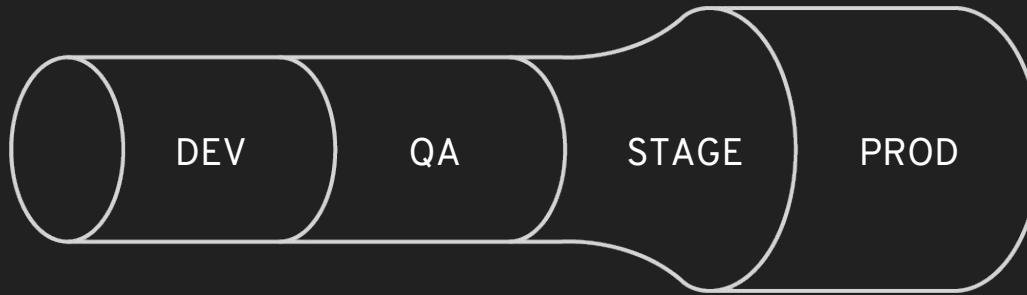
Shared Resources



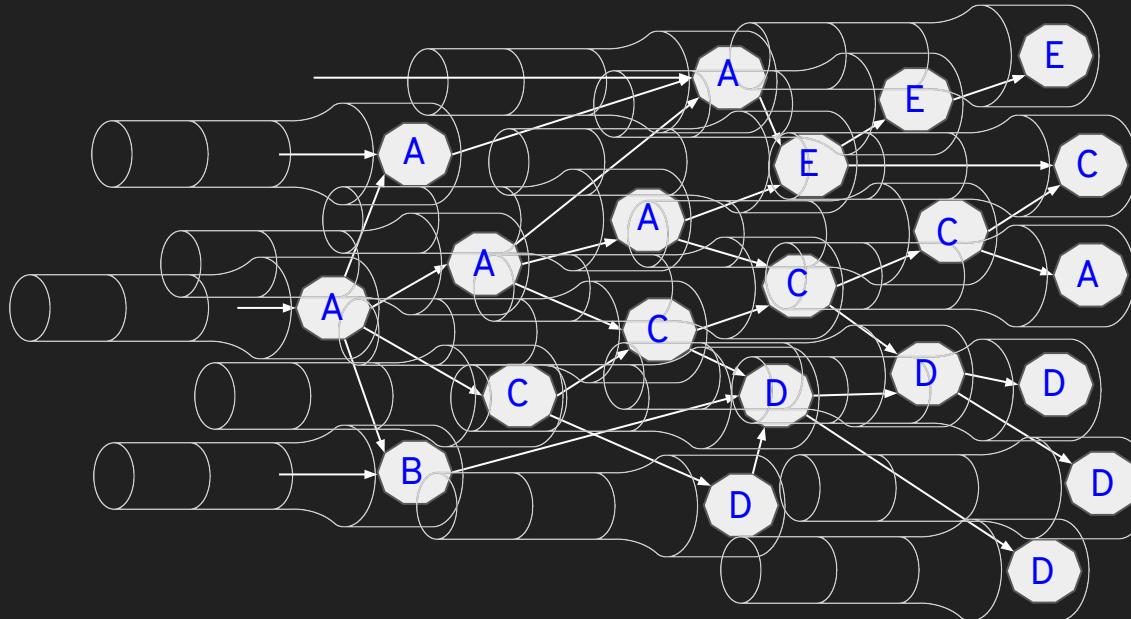
Our Service Mess



Our Pipelines



Our Services, Our Pipelines, Our Teams



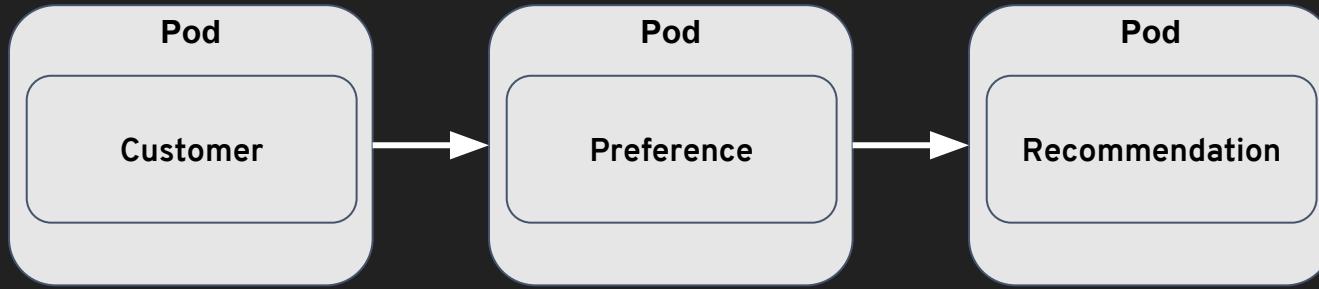
Customer Success Engineering Team A
Human Resources Engineering Team B
Marketing Engineering Team C

Manufacturing Engineering Team D
Big Money Customer Engineering Team E

Istio Security Capabilities

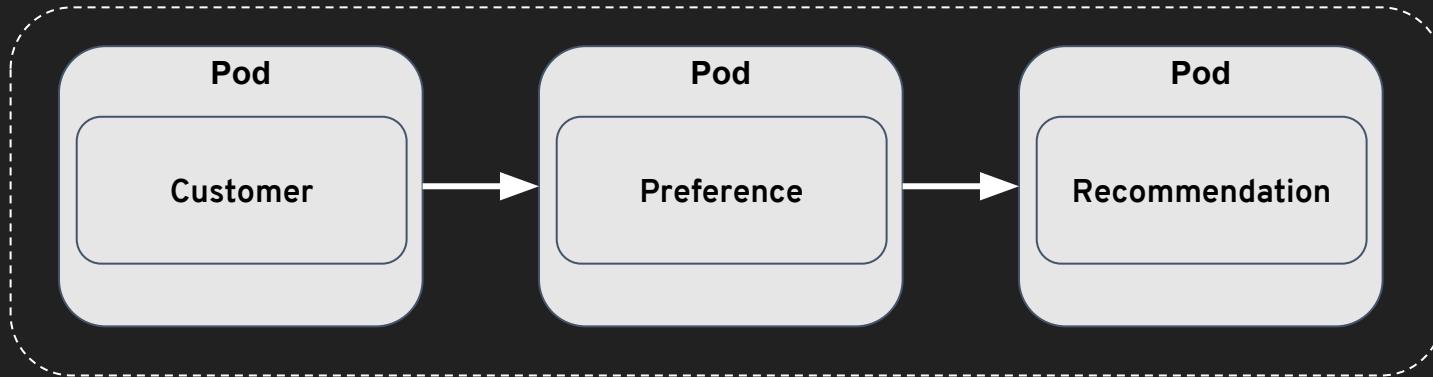
- mTLS - Encryption
- Access Control
- JSON Web Token (JWT) Authentication
- Role-based Access Control (RBAC) Authorization

Why Encryption?



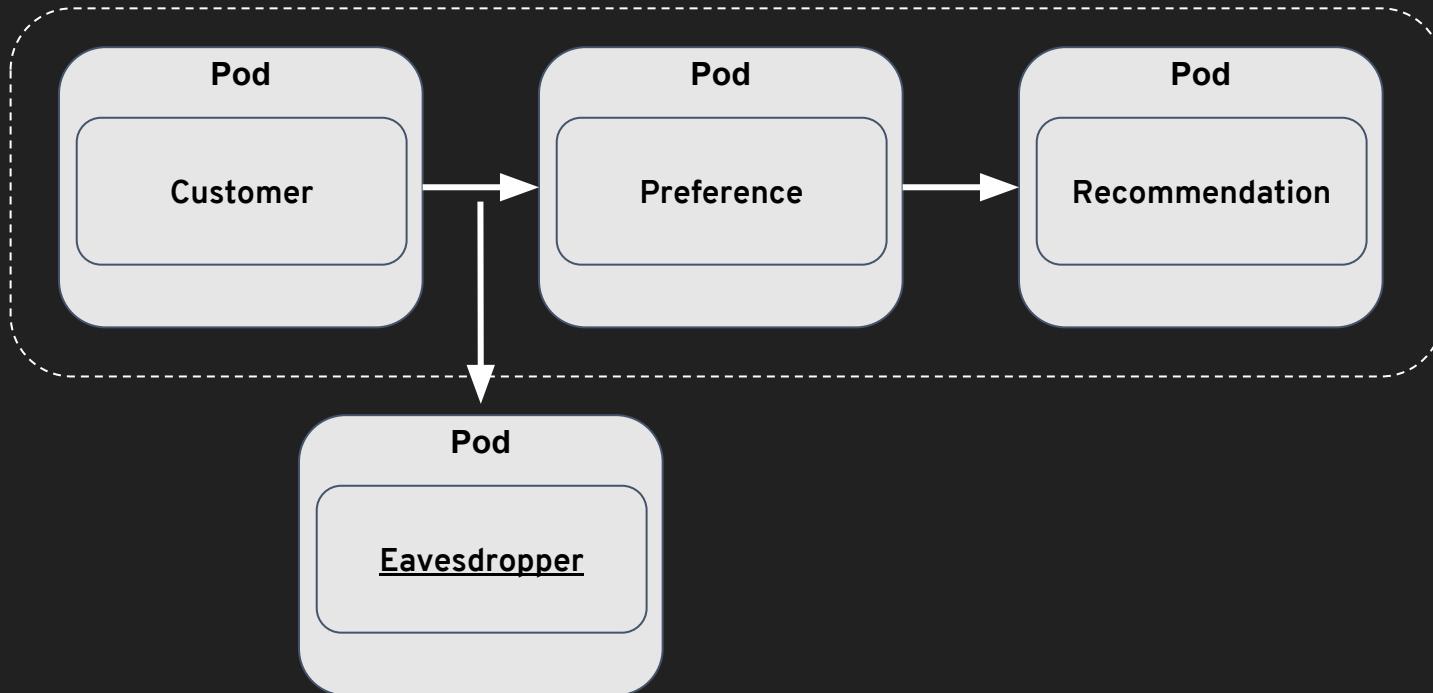
Why Encryption?

Big Money Customer Engineering Team



Why Encryption?

Big Money Customer Engineering Team



istio Shell ...e-v1-5485dc6f49-fspbb: /

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

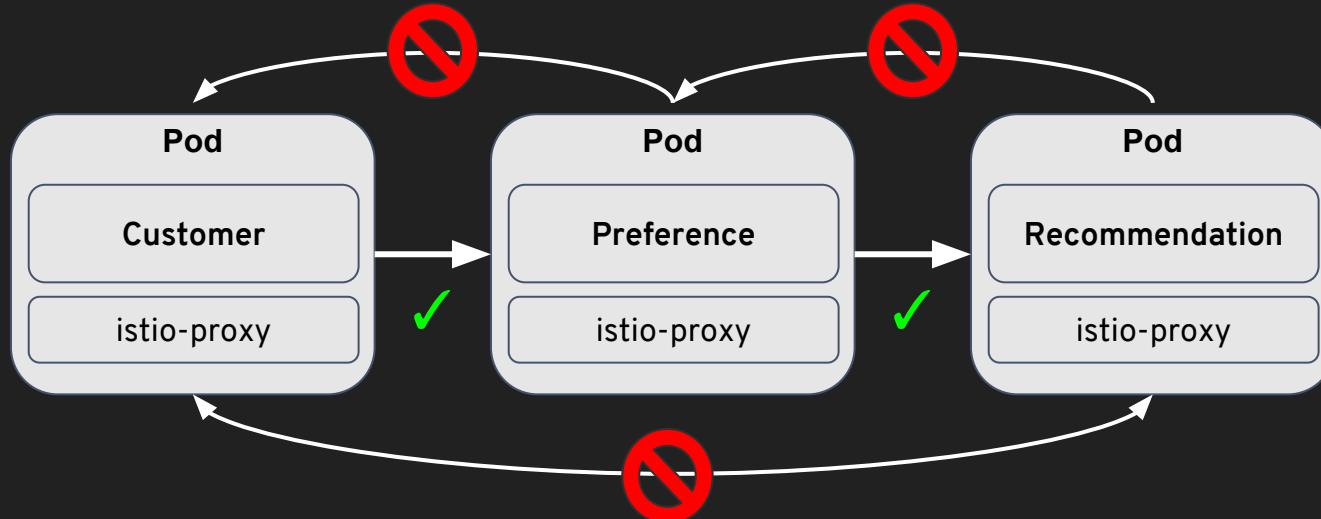


```
[jboss@preference-v1-5485dc6f49-fspbb ~]$ curl recommendation:8080
recommendation v2 from '7cbd9f9c79-nd69g': 341
[jboss@preference-v1-5485dc6f49-fspbb ~]$ curl recommendation:8080
recommendation v1 from '66b7c9779c-75fp1': 347
[jboss@preference-v1-5485dc6f49-fspbb ~]$ 
```

istio

```
~/minishift_1.27.0/istio-tutorial $ 
```

Access Control



JWT Issuer

The screenshot shows the Keycloak Admin UI interface. The left sidebar is dark-themed with white icons and text, showing navigation options like 'Master', 'Add realm', 'Realm Settings', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', 'Authentication', 'Manage', 'Groups', and 'Users'. The main content area has a light background and displays the 'Master' realm settings. The 'General' tab is selected, showing fields for 'Name' (set to 'master'), 'Display name' (set to 'Keycloak'), 'HTML Display name' (containing a snippet of HTML: '<div class="kc-logo-text">Keycloak</div>'), 'Enabled' (set to 'ON'), 'User-Managed Access' (set to 'OFF'), and an 'Endpoints' section with a link to 'OpenID Endpoint Configuration'. At the bottom are 'Save' and 'Cancel' buttons.

1.0 Changes

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

The End (but Serverless is coming)

Extra Slides

Raffle Rules (applicable in the real)

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