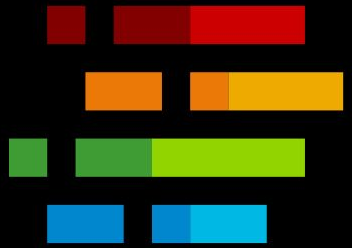


RED HAT® DEVELOPER

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

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

I am a Developer

Kamesh Sampath

Director of Developer Experience at Red Hat

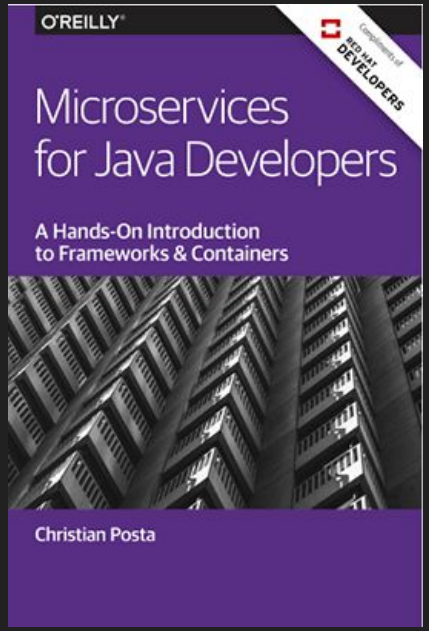
- Active Open Source Contributor
  - Knative
  - Minishift
  - Eclipse Che
  - fabric8 Platform https://fabric8.io/
- Creator vert.x-maven-plugin → https://vmp.fabric8.io/

kamesh.sampath@hotmail.com

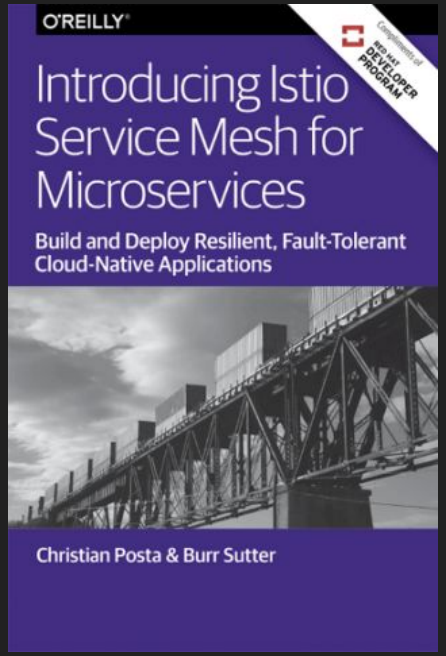
kameshsampath

@kamesh_sampath

bit.ly/javamicroservicesbook



bit.ly/istio-book



bit.ly/reactivemicroservicesbook

Demo: bit.ly/msa-instructions
Slides: bit.ly/microservicesdeepdive
Video Training: bit.ly/microservicesvideo

bit.ly/mono2microdb

Kubernetes for Java Developers

@kamesh_sampath

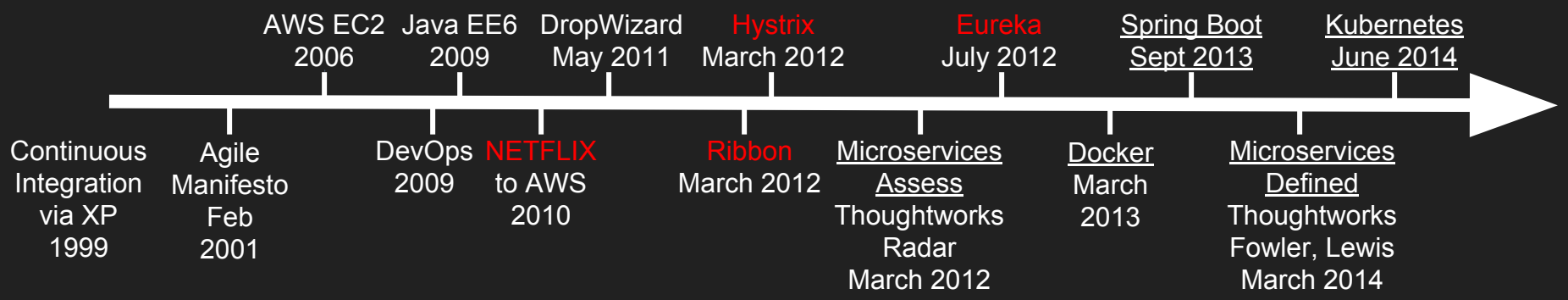
What is a microservice ?

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.

These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

Martin Fowler

Short History of Microservices

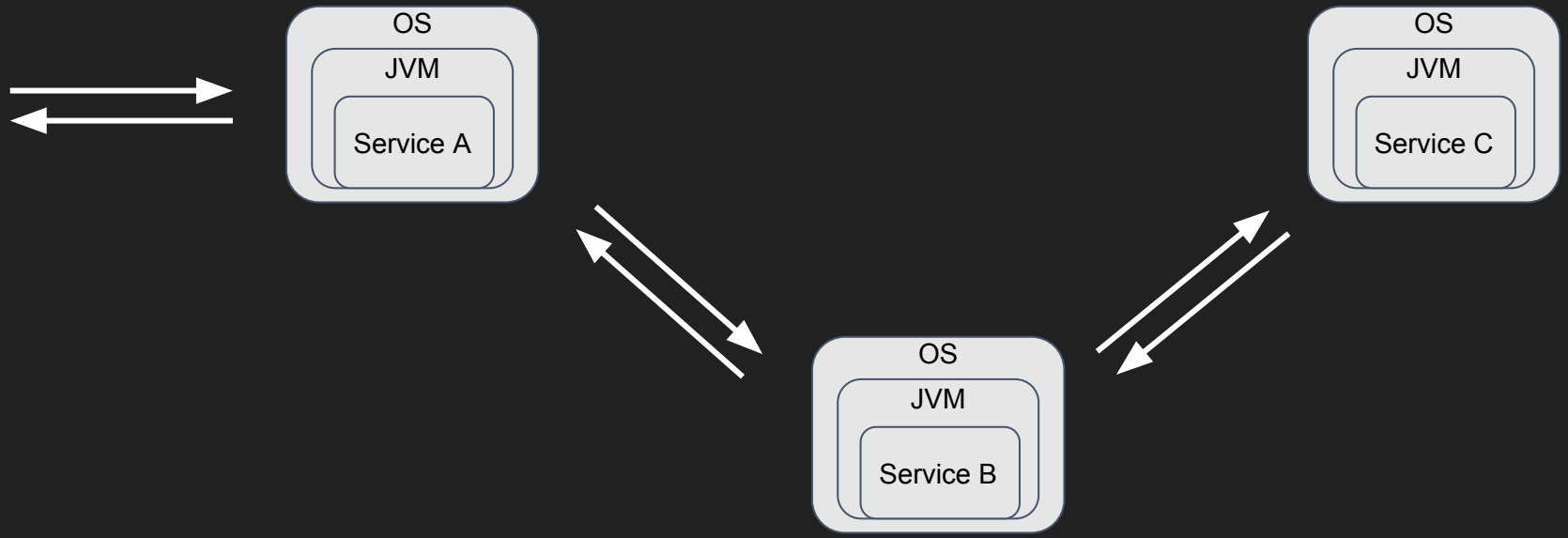


NETFLIX | OSS

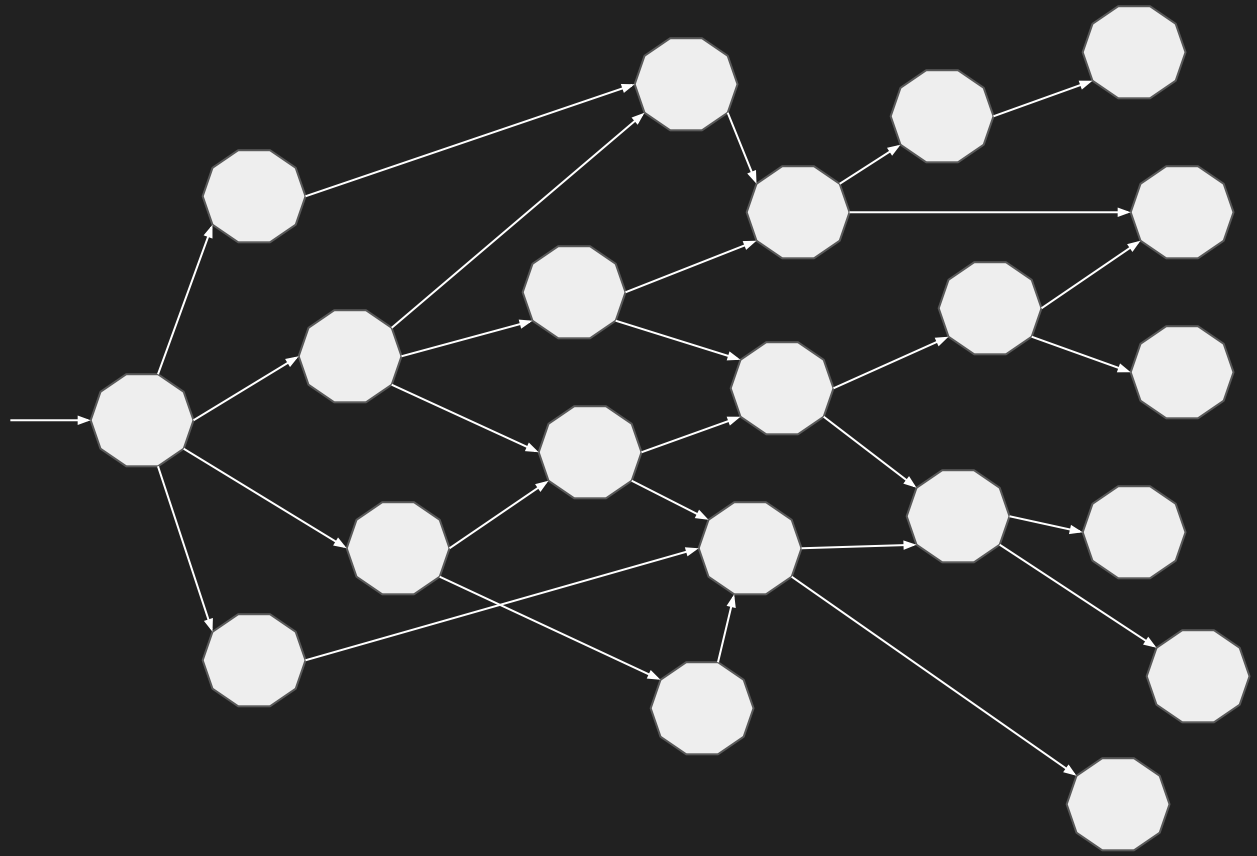
@kamesh_sampath

RED HAT DEVELOPER

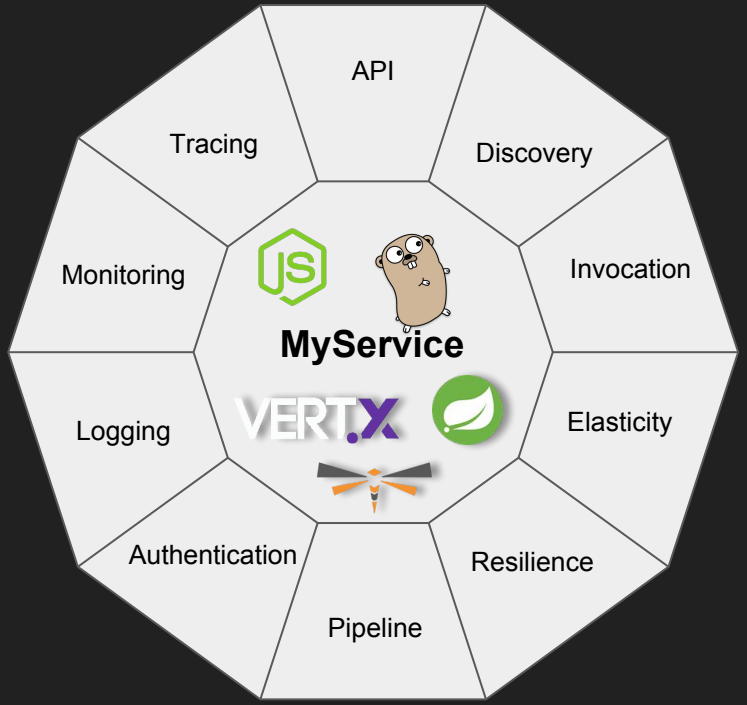
Microservices == Distributed Computing



Distributed Computing == Network of Services

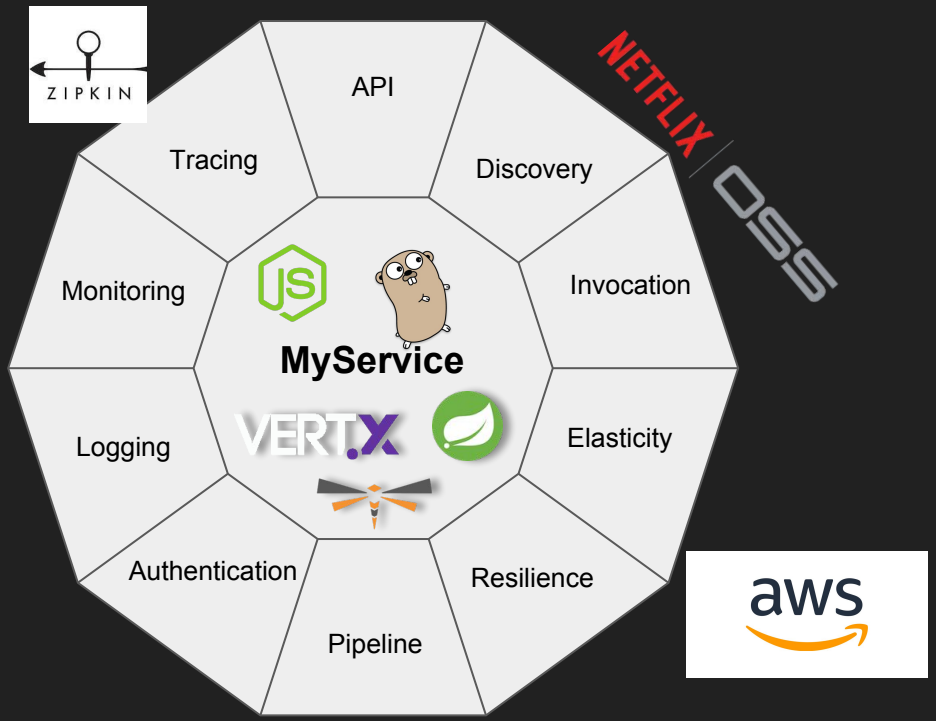


Microservices'ilities



RED HAT DEVELOPER

Microservices'ilities



RED HAT DEVELOPER

Java Microservices Platform circa 2014-

Config Server

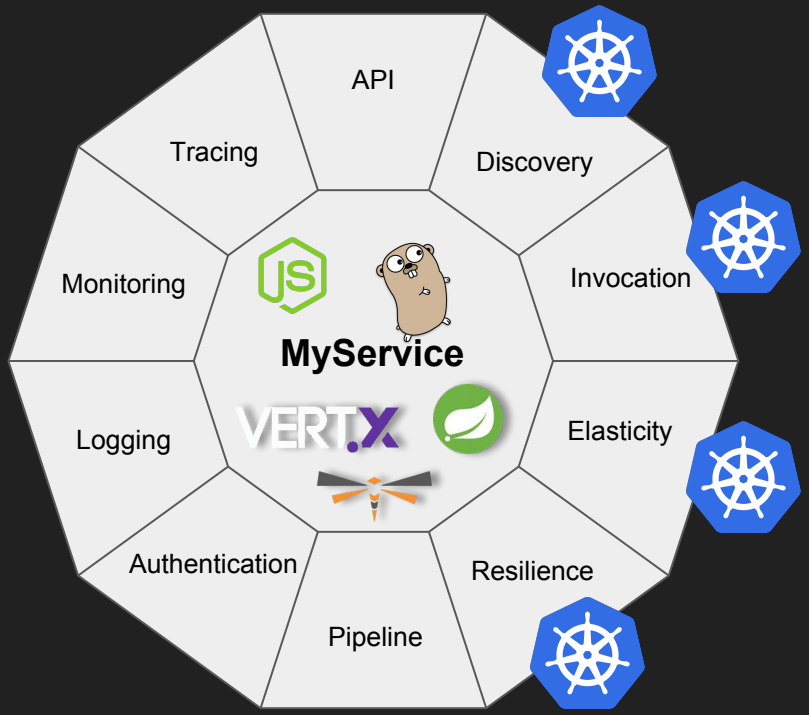
NETFLIX Ribbon

DevOps Challenges for Multiple Containers

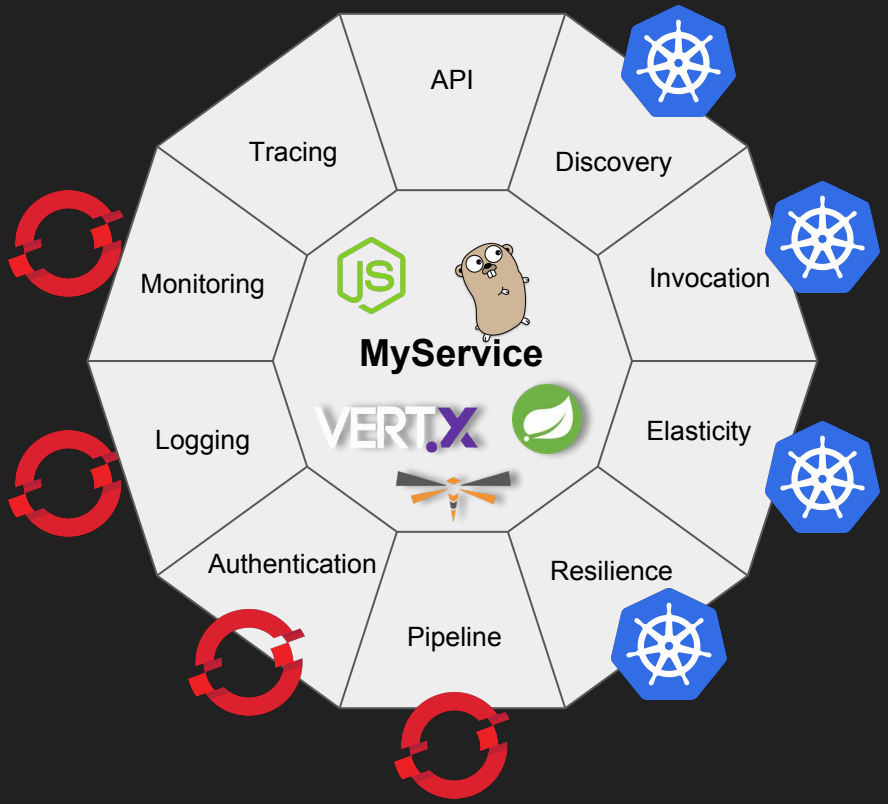
- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Where are my containers?

OPENSHIFT

Microservices'ilities + Kubernetes

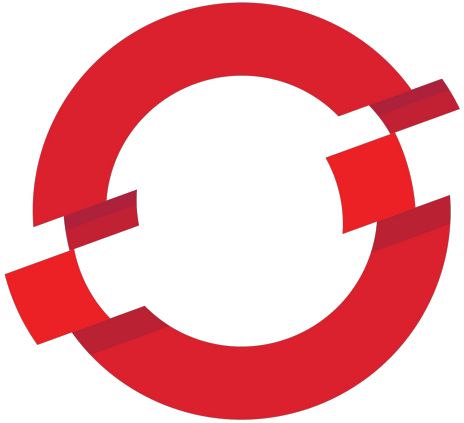


Microservices'ilities + OpenShift



Better Spring Boot Microservices Platform circa 2016

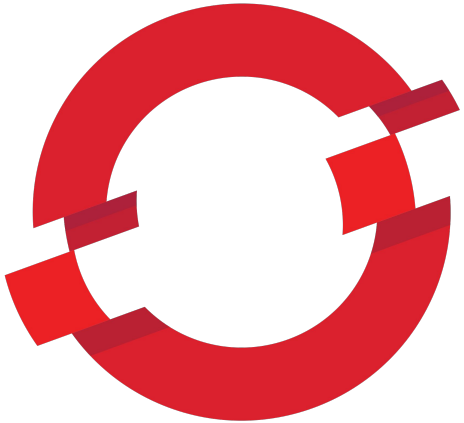
NETFLIX Ribbon



OPENSHIFT

@kamesh_sampath

Better Spring Boot Microservices Platform circa 2017



OPENSHIFT

@kamesh_sampath

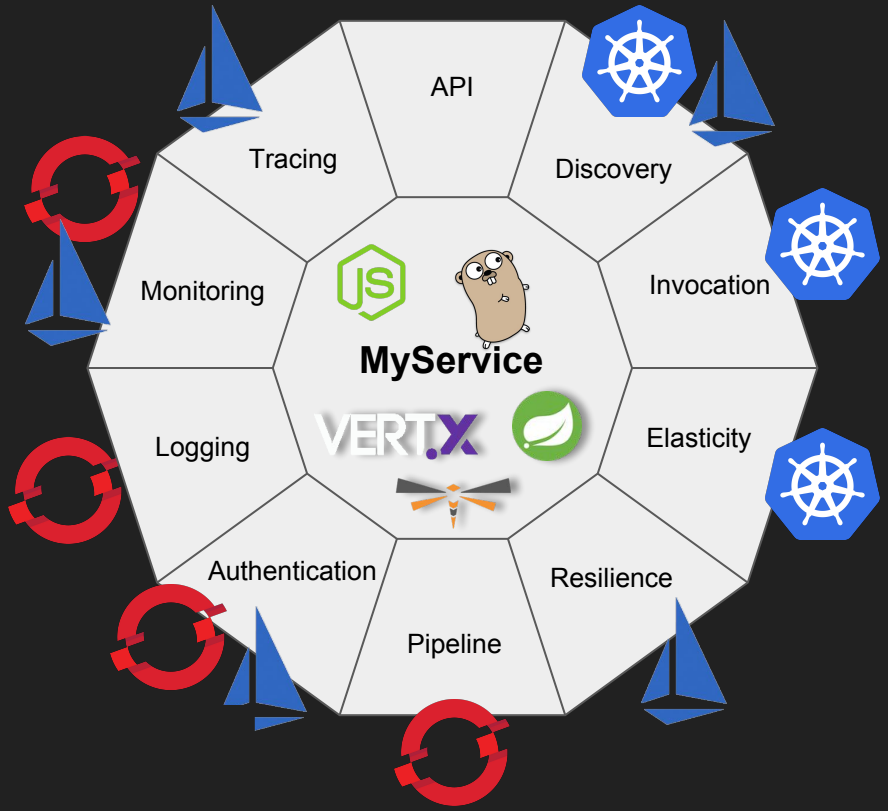
Microservice(Yes) Pain Points

- Discovery
- Distributed Tracing
- Circuit Breakers
- Metrics and Monitoring
- Operational Requirements
  - A/B Testing
  - Canary Release
  - Rate Limiting
  - Access Policies

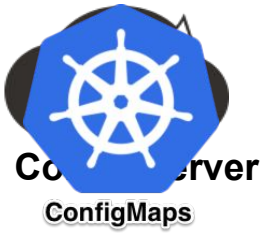
Istio - Sail

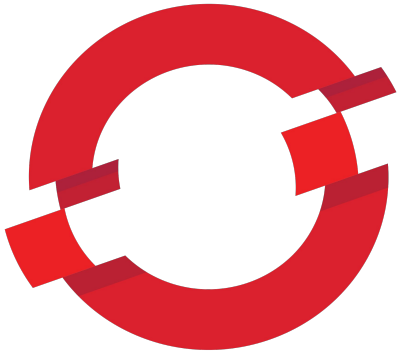
(Kubernetes - Helmsman or ship’s pilot)

Microservices'ilities + Istio



Better Spring Boot Microservices Platform circa 2018

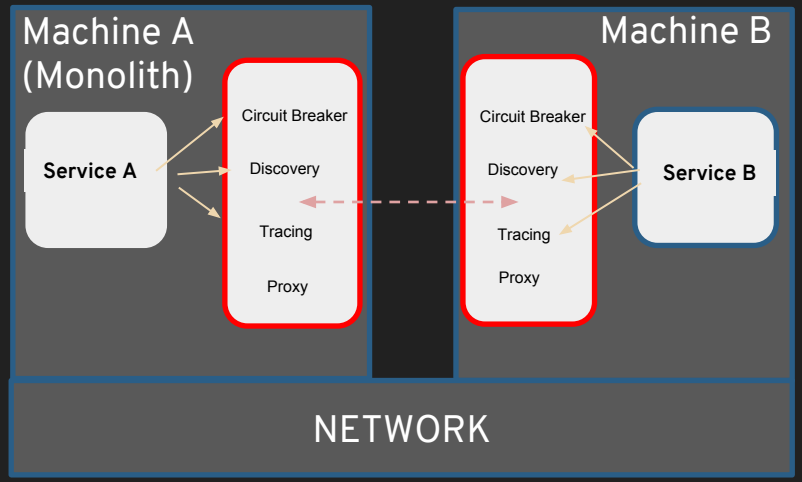




OPENSHIFT

@kamesh_sampath

What is a service mesh ?



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.

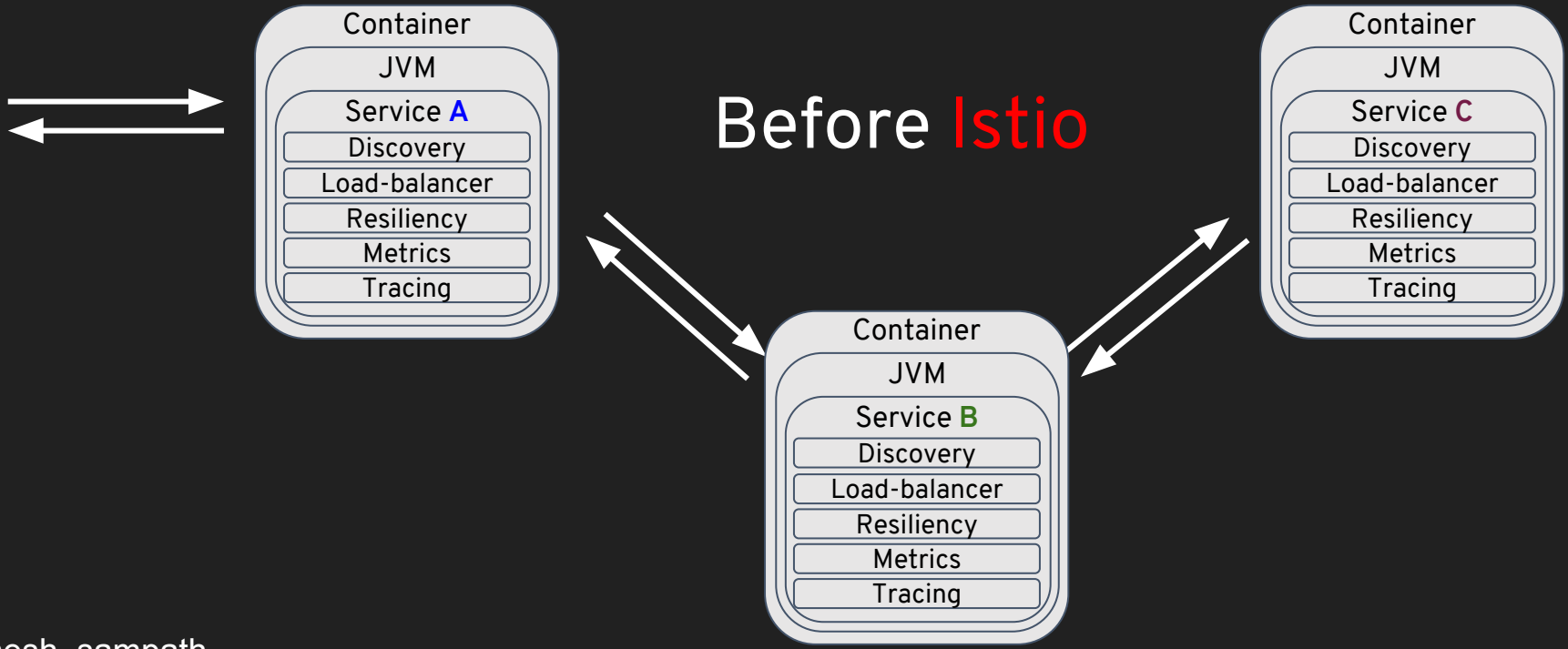
https://www.cncf.io/blog/2017/04/26/service-mesh-critical-component-cloud-native-stack/

What's Wrong with Netflix OSS?

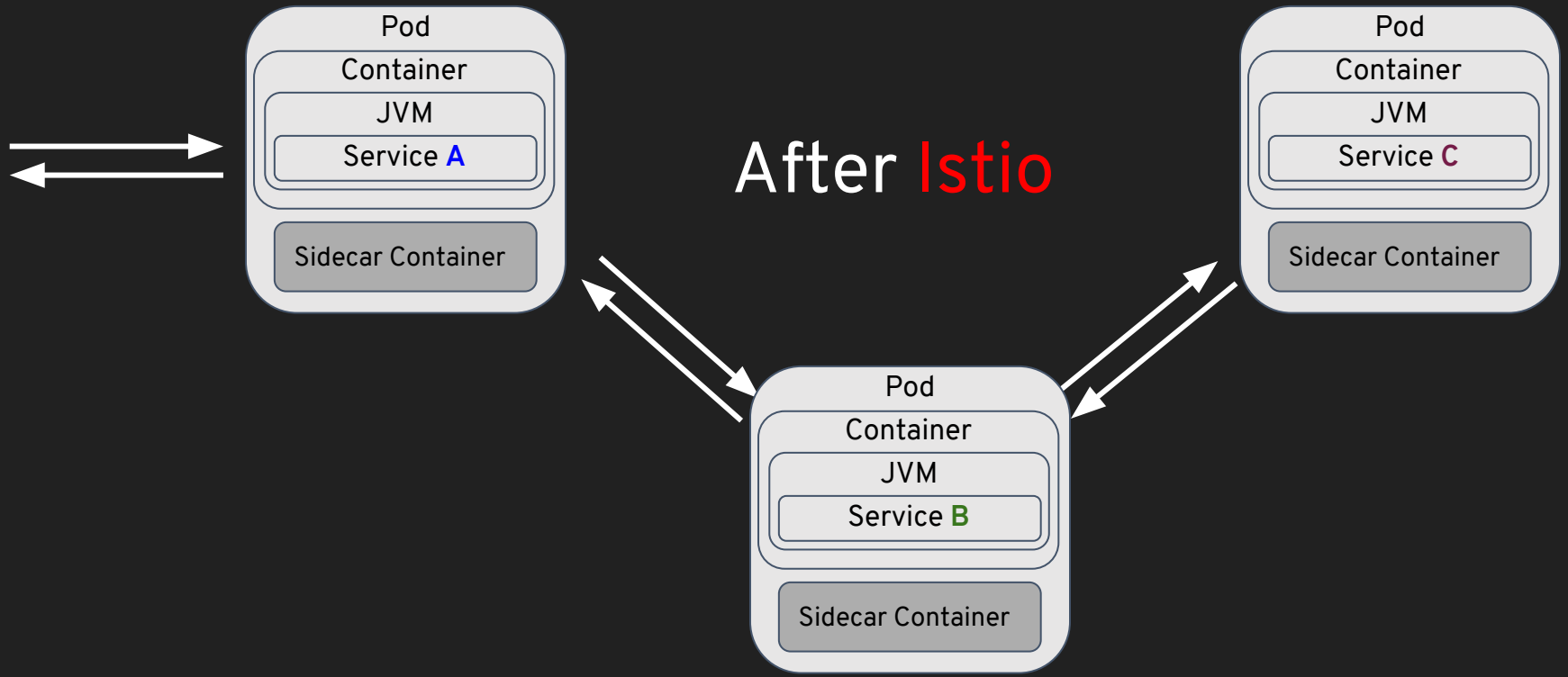
Java Only

Adds a lot of libraries to YOUR code

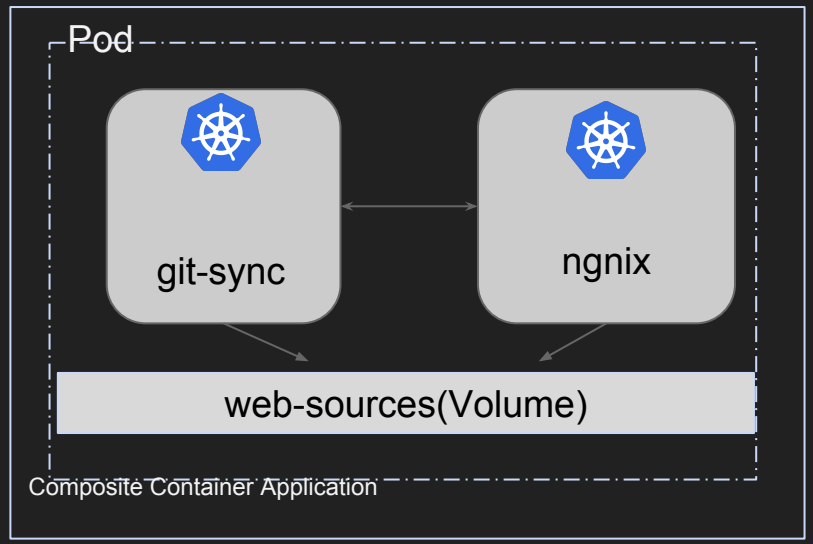
Microservices embedding Capabilities



Microservices externalizing Capabilities



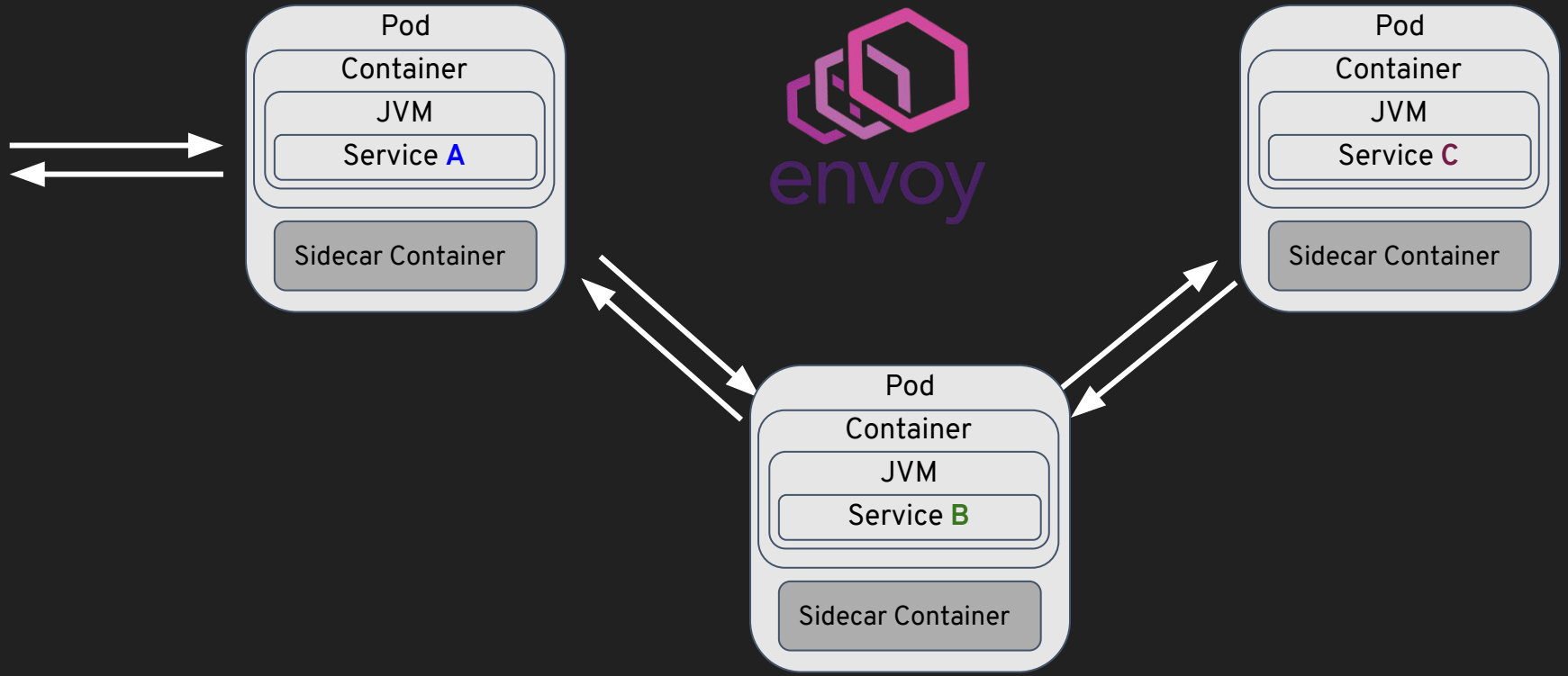
SideCars



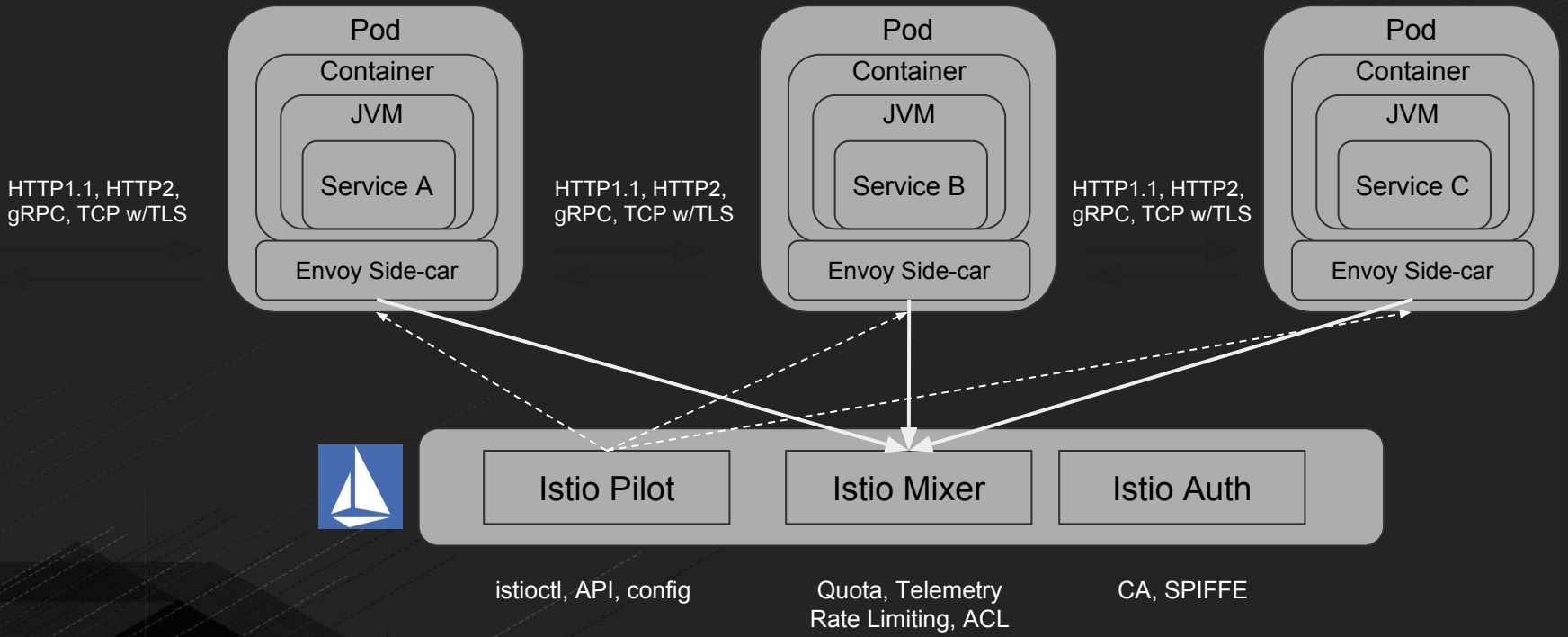
- Two or more containers deployed to same pod
- Share
  - same namespace
  - same Pod IP
  - Shared lifecycle
- Used to enhance the containers

Source: http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html

Kubernetes, Istio, Envoy



Istio Control Plane

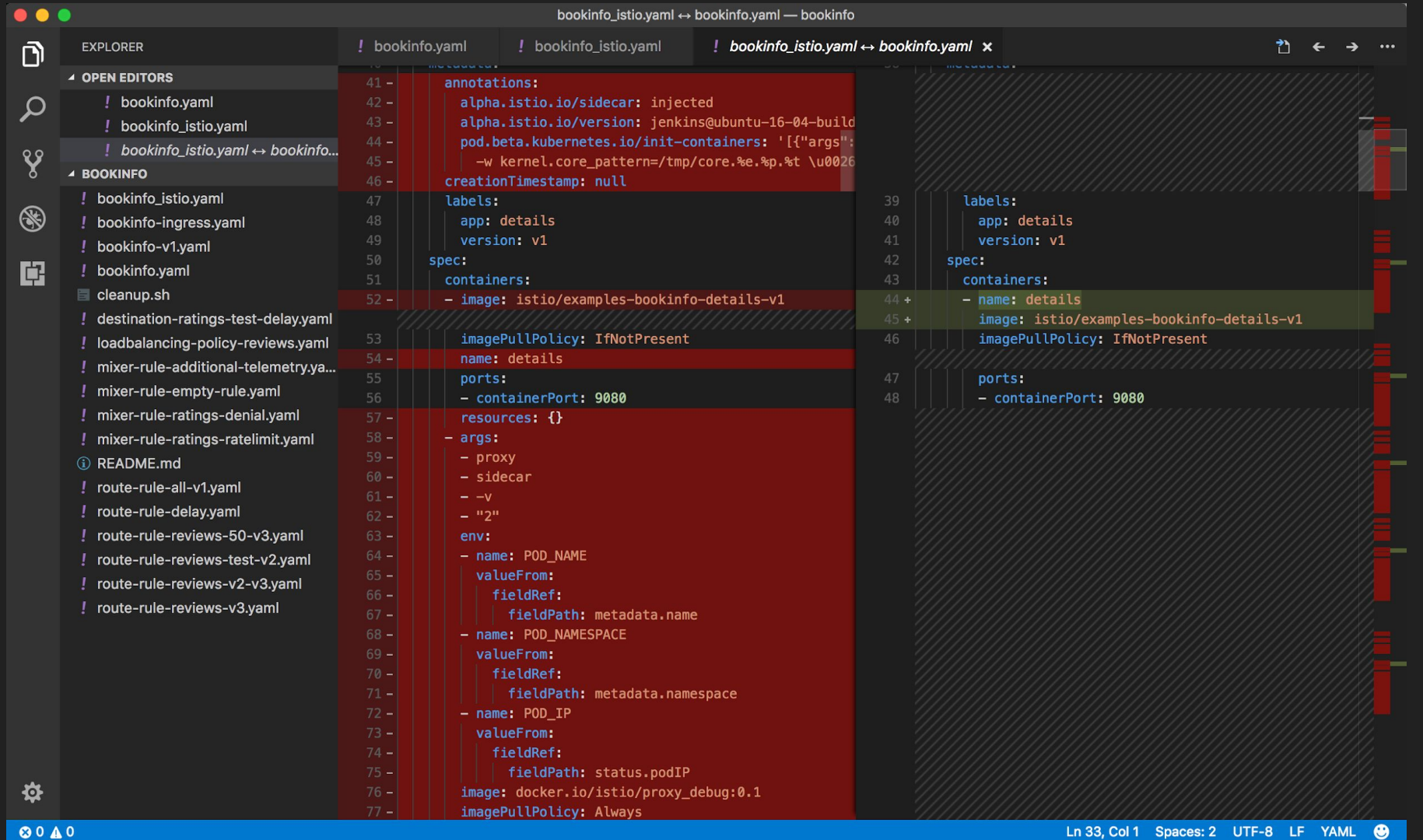


RED HAT DEVELOPERS

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

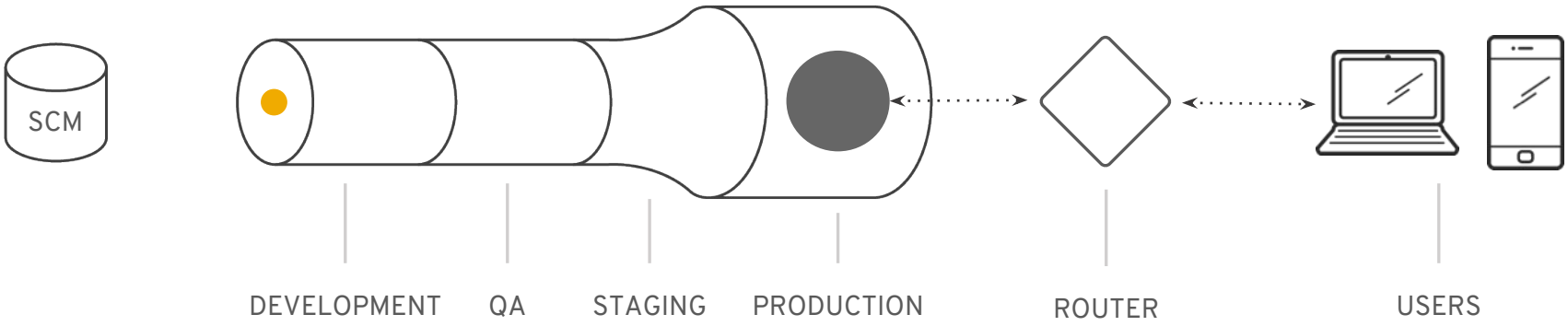

bookinfo_istio.yaml ↔ bookinfo.yaml — bookinfo
EXPLORER
OPEN EDITORS
bookinfo.yaml
bookinfo_istio.yaml
bookinfo_istio.yaml ↔ bookinfo...
BOOKINFO
bookinfo_istio.yaml
bookinfo-ingress.yaml
bookinfo-v1.yaml
bookinfo.yaml
cleanup.sh
destination-ratings-test-delay.yaml
loadbalancing-policy-reviews.yaml
mixer-rule-additional-telemetry.ya...
mixer-rule-empty-rule.yaml
mixer-rule-ratings-denial.yaml
mixer-rule-ratings-ratelimit.yaml
README.md
route-rule-all-v1.yaml
route-rule-delay.yaml
route-rule-reviews-50-v3.yaml
route-rule-reviews-test-v2.yaml
route-rule-reviews-v2-v3.yaml
route-rule-reviews-v3.yaml
bookinfo.yaml
bookinfo_istio.yaml
bookinfo_istio.yaml ↔ bookinfo.yaml
annotations:
alpha.istio.io/sidecar: injected
alpha.istio.io/version: jenkins@ubuntu-16-04-build
pod.beta.kubernetes.io/init-containers: '[{"args":
-w kernel.core_pattern=/tmp/core.%e.%p.%t \u0026
creationTimestamp: null
labels:
app: details
version: v1
spec:
containers:
- image: istio/examples-bookinfo-details-v1
imagePullPolicy: IfNotPresent
name: details
ports:
- containerPort: 9080
resources: {}
- args:
- proxy
- sidecar
- -v
- "2"
env:
- name: POD_NAME
valueFrom:
fieldRef:
fieldPath: metadata.name
- name: POD_NAMESPACE
valueFrom:
fieldRef:
fieldPath: metadata.namespace
- name: POD_IP
valueFrom:
fieldRef:
fieldPath: status.podIP
image: docker.io/istio/proxy_debug:0.1
imagePullPolicy: Always
labels:
app: details
version: v1
spec:
containers:
- name: details
image: istio/examples-bookinfo-details-v1
imagePullPolicy: IfNotPresent
ports:
- containerPort: 9080
0 0
Ln 33, Col 1
Spaces: 2
UTF-8
LF
YAML

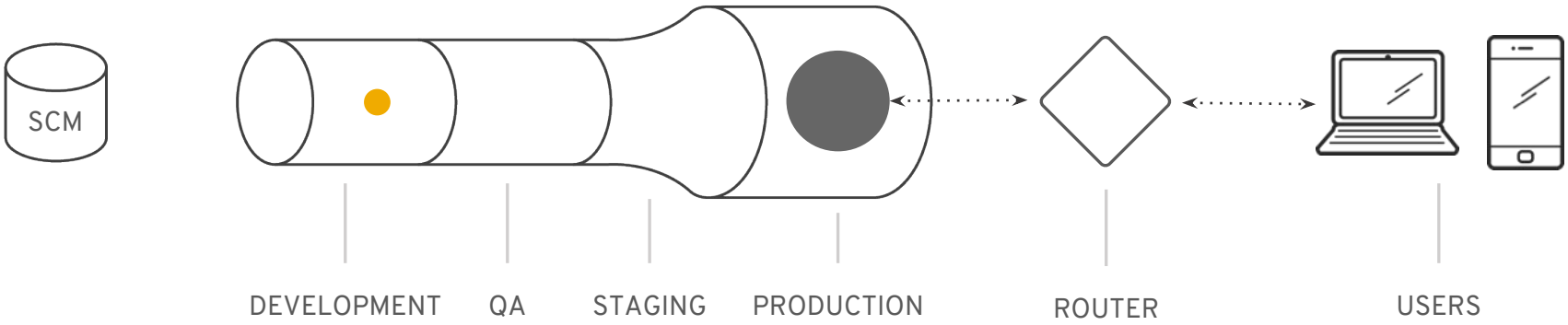
Microservice Architecture - Principles

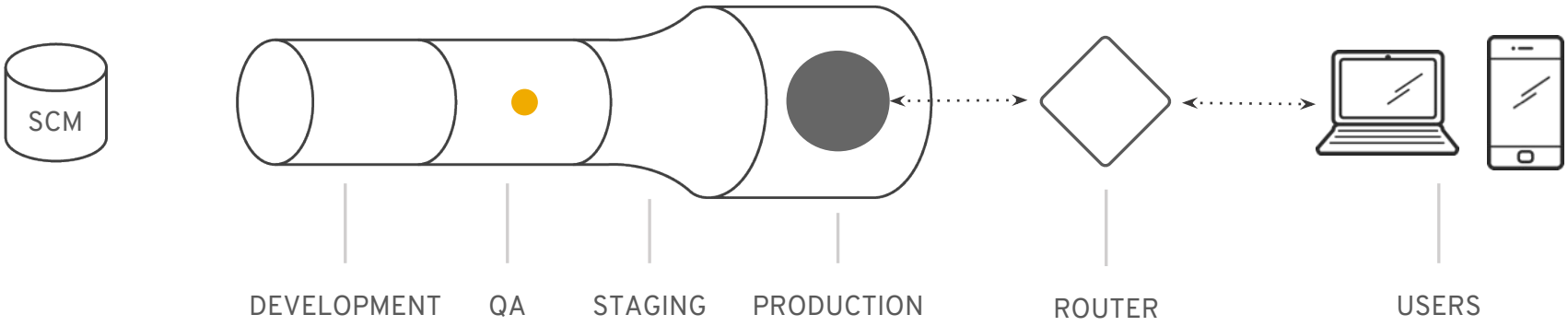
Canary Release

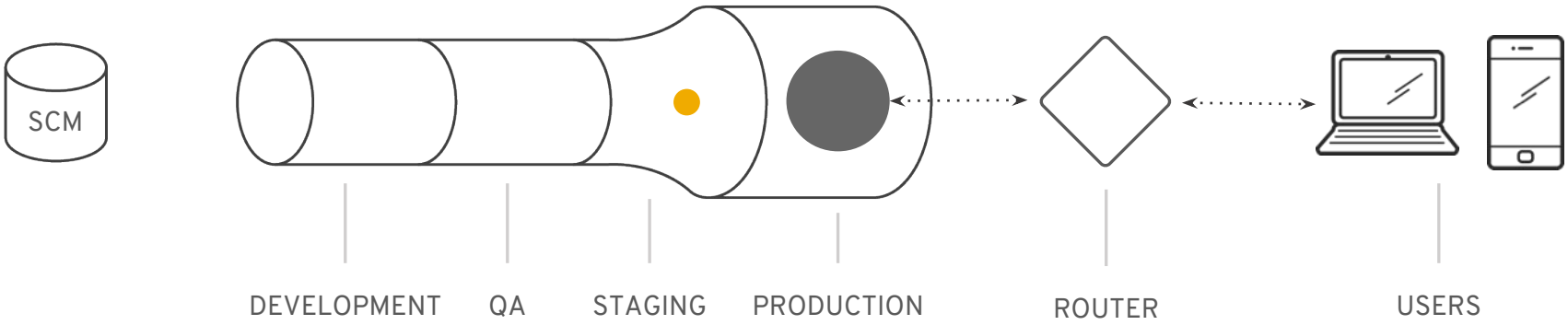
Canary release is a technique to reduce the risk of introducing a new software version in production by slowly rolling out the change to a small subset of users before rolling it out to the entire infrastructure and making it available to everybody.

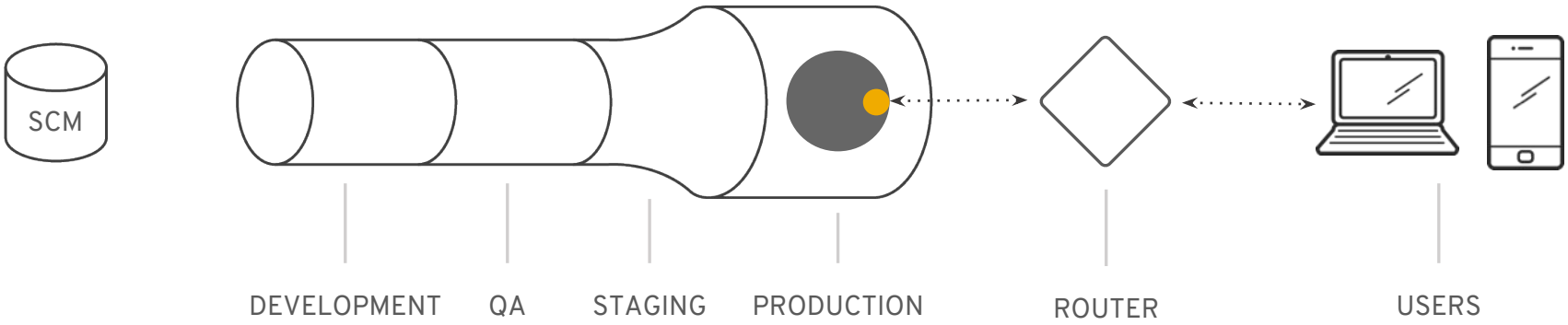
Source: https://martinfowler.com/bliki/CanaryRelease.html

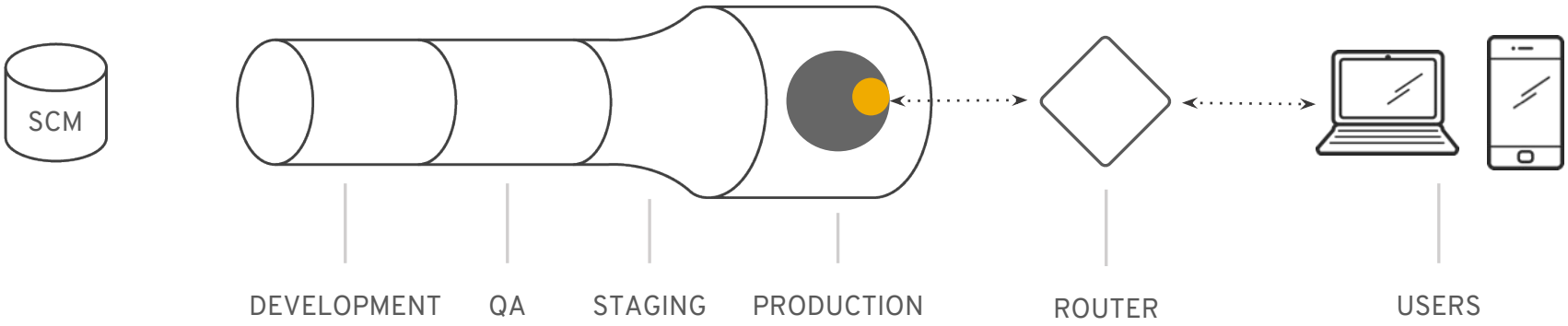

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

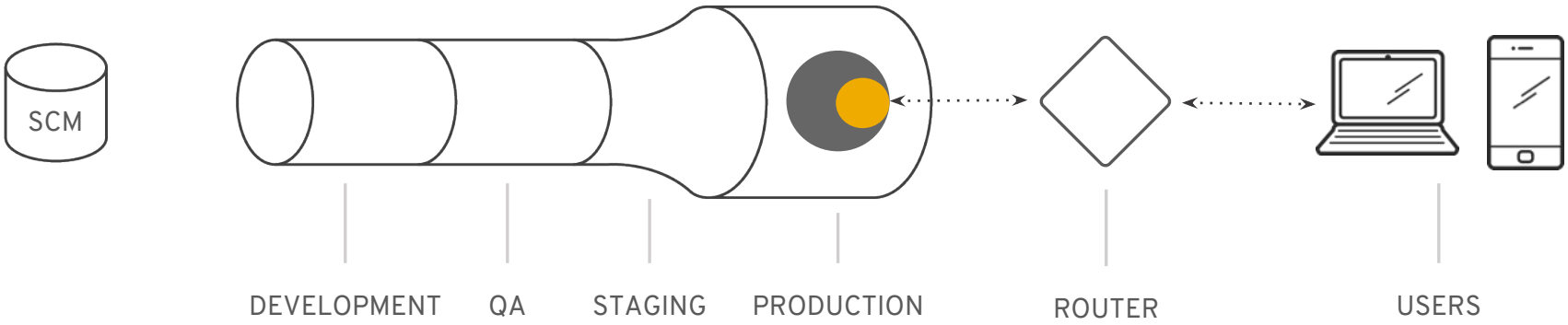

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

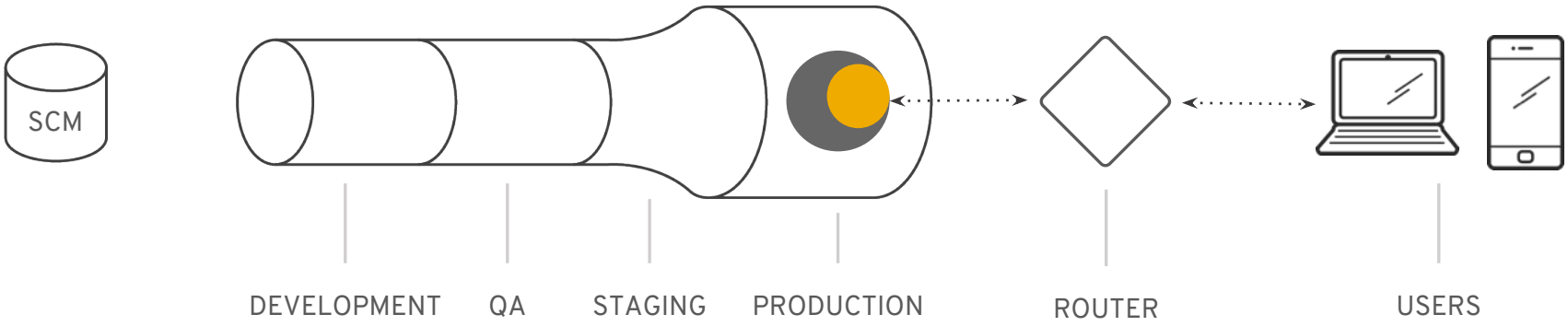

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

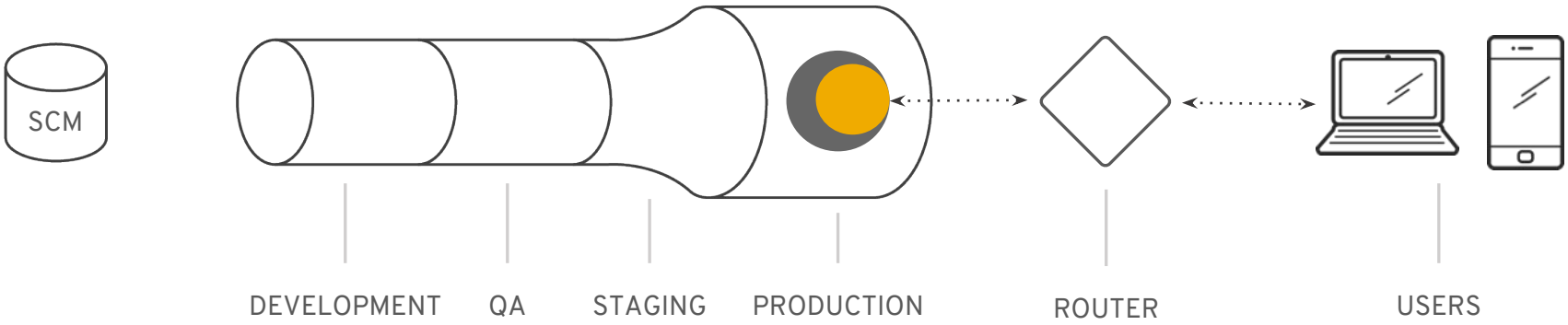

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

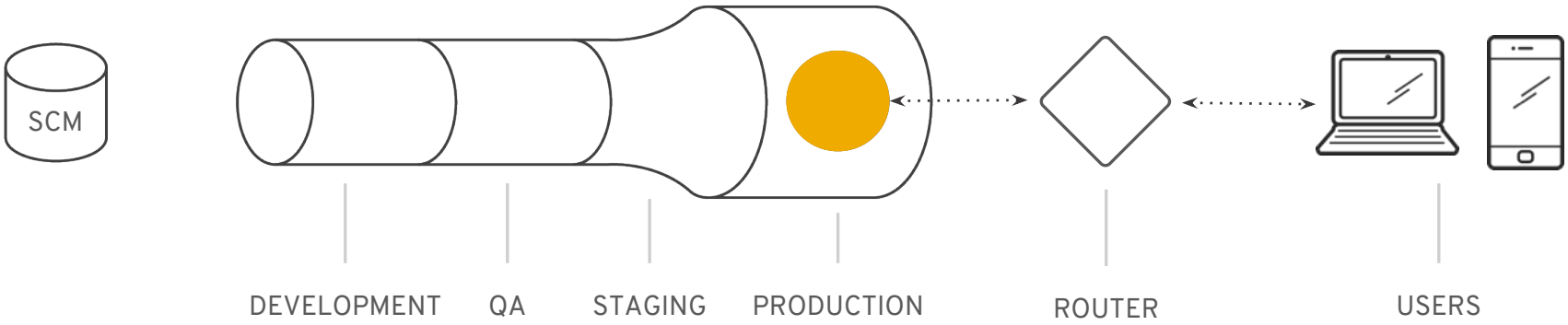

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS


SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

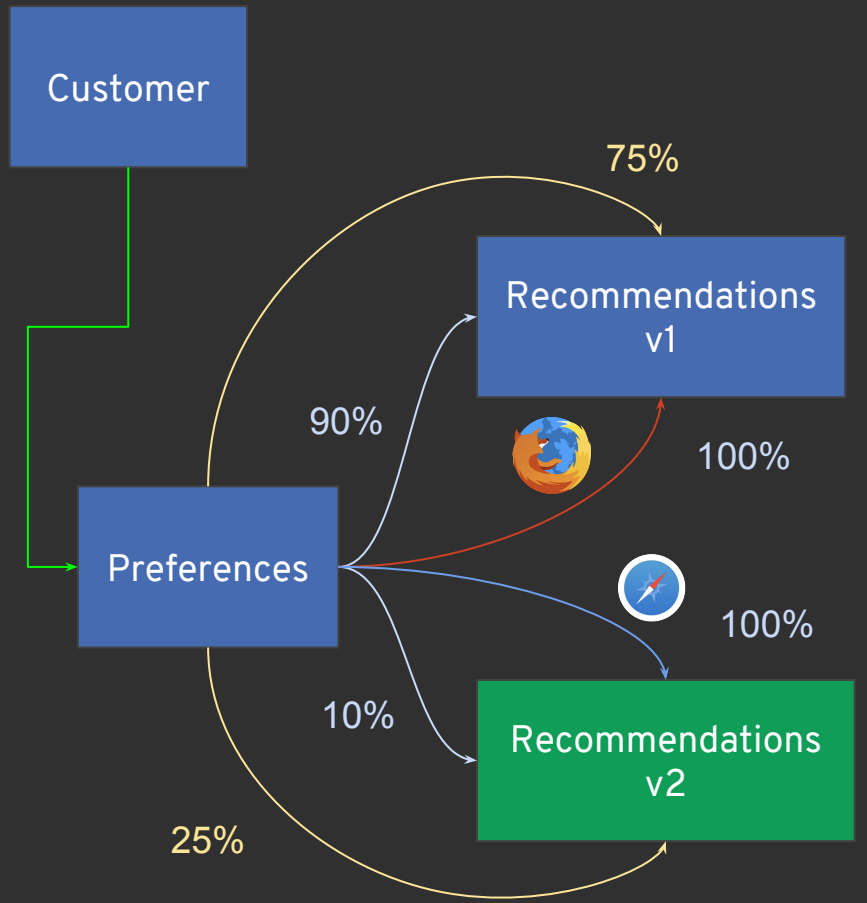

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS


SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

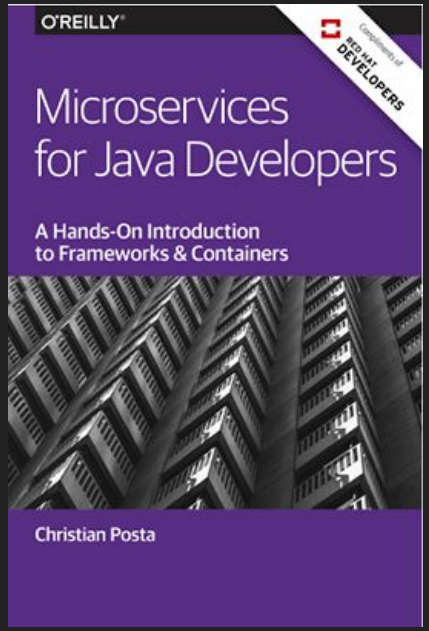

SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS


SCM
DEVELOPMENT
QA
STAGING
PRODUCTION
ROUTER
USERS

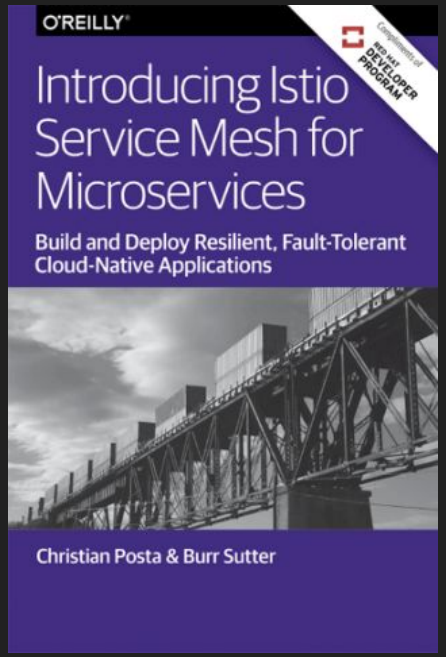
Demo


Customer
75%
Recommendations v1
90%
100%
Preferences
100%
10%
Recommendations v2
25%

bit.ly/javamicroservicesbook



bit.ly/istio-book



bit.ly/reactivemicroservicesbook

Demo: bit.ly/msa-instructions
Slides: bit.ly/microservicesdeepdive
Video Training: bit.ly/microservicesvideo

bit.ly/mono2microdb

Kubernetes for Java Developers

@kamesh_sampath

Summary

@kamesh_sampath

- Minishift
- https://istio.io
- bit.ly/sail-into-cloud
  - Demo sources
    https://github.com/workspace7/kubeboot
- https://kiali.io/
- Istio Tutorials
  - bit.ly/istio-tutorial
  - learn.openshift.com/servicemesh