Burr Sutter

Istio on Kubernetes: Canaries, Chaos and Dark Launches

VOXXED DAYS
BUCHAREST

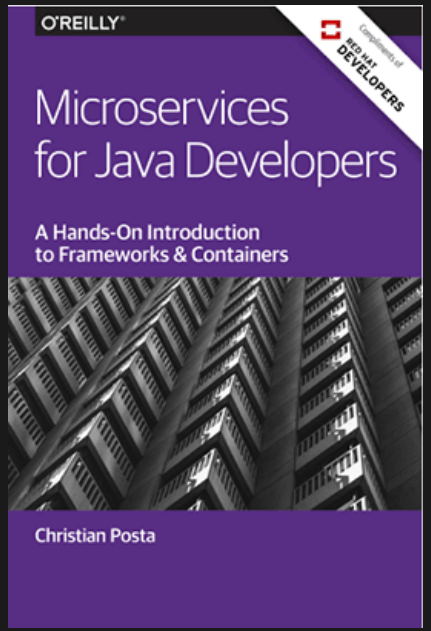
RED HAT®
DEVELOPER
PROGRAM

Istio: Canaries, Chaos, Dark Launches

@burrsutter burr@redhat.com

Link → bit.ly/istio-intro
bit.ly/istio-tutorial
https://learn.openshift.com/servicemesh

bit.ly/javamicroservicesbook

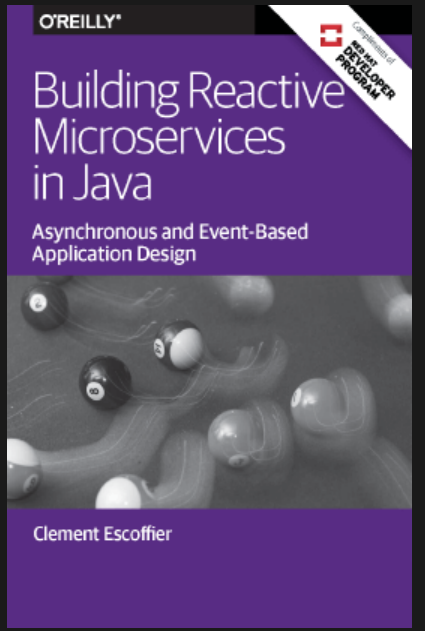


Free eBooks from developers.redhat.com

Microservices Introductory Materials

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

bit.ly/reactivemicroservicesbook



Microservices Advanced Materials

bit.ly/istio-tutorial

bit.ly/mono2microdb

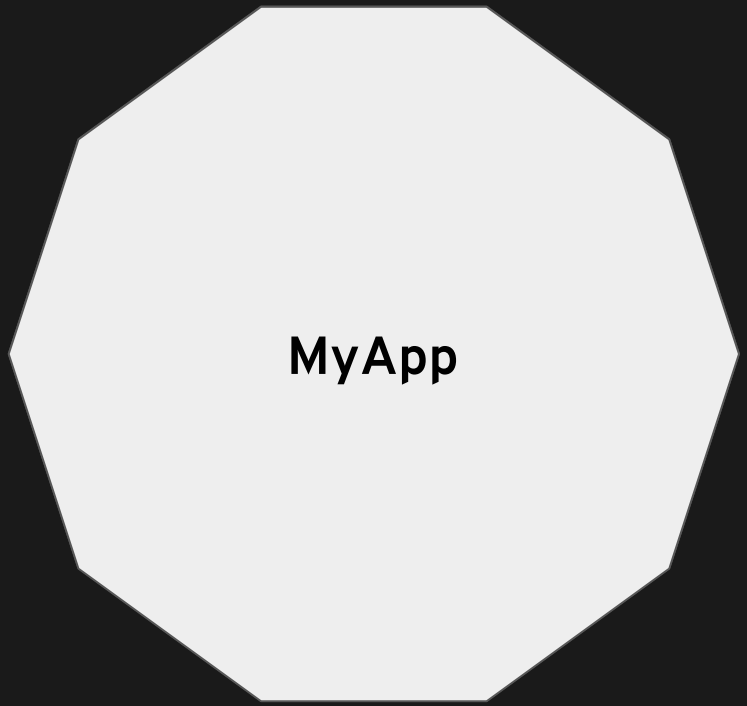
http://bit.ly/istio-intro

Istio.io

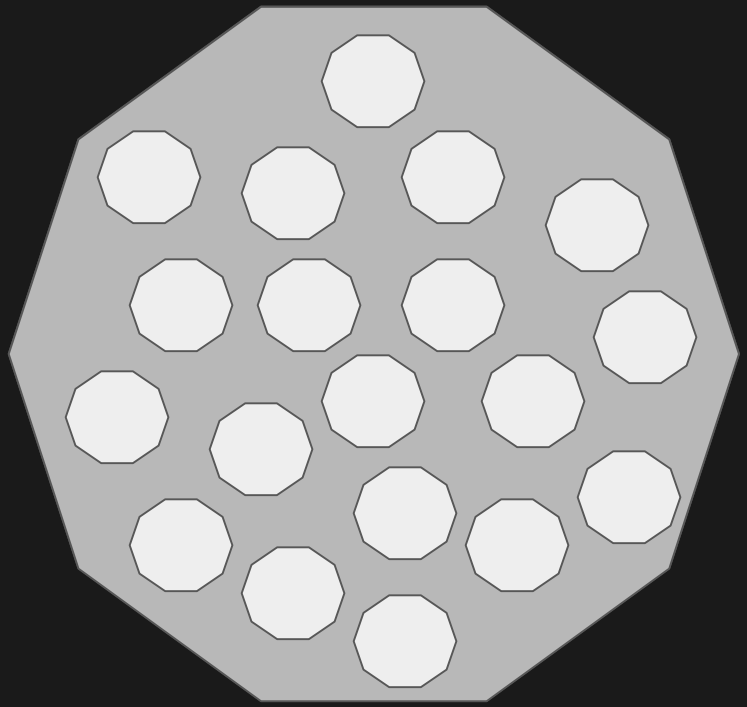
@burrsutter

RED HAT DEVELOPER PROGRAM

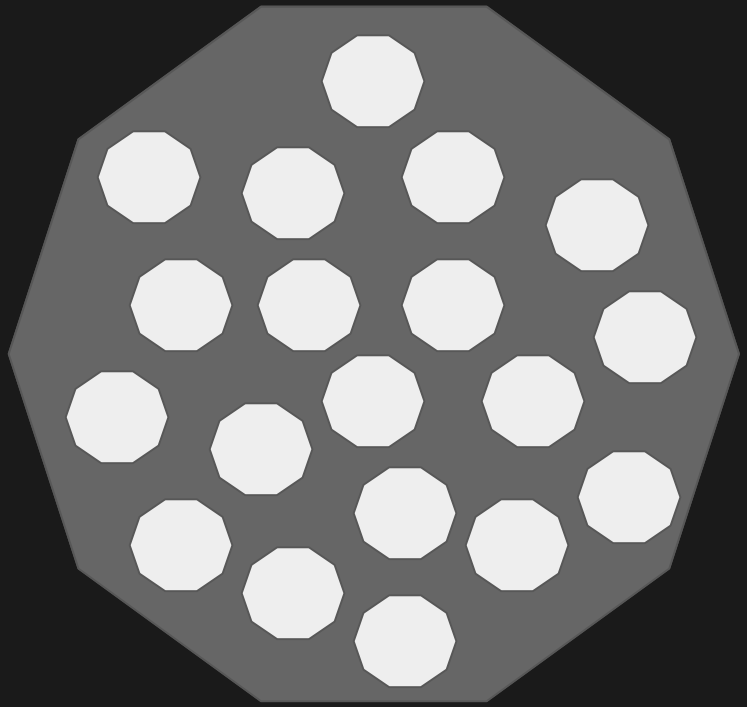
Monolith



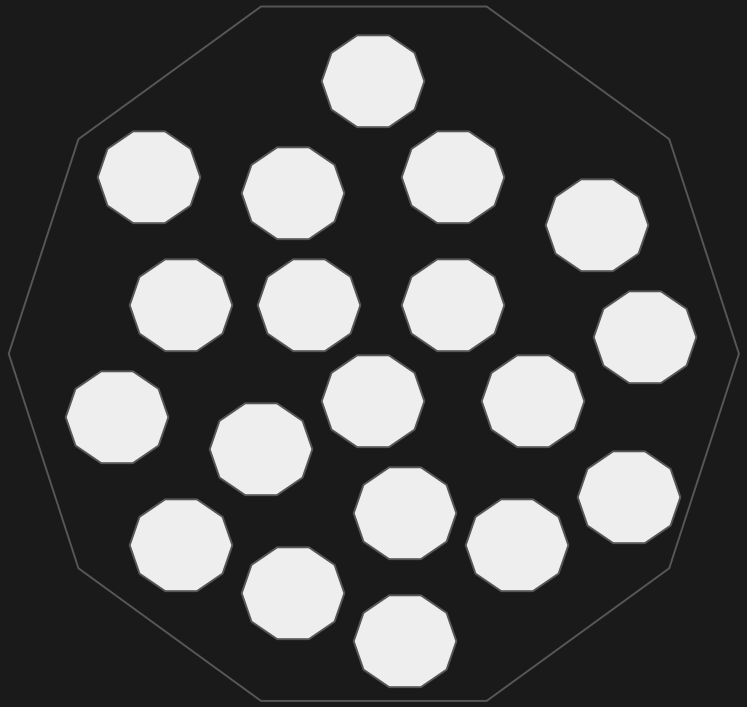
Microservices



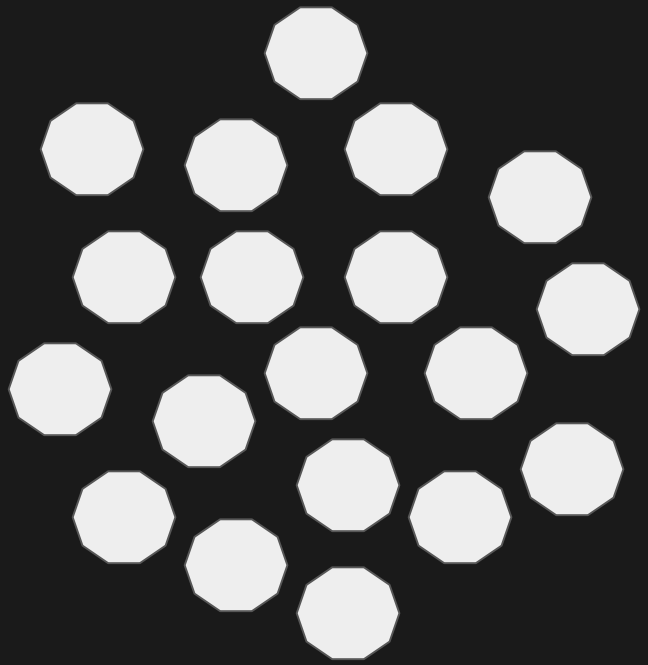
Microservices



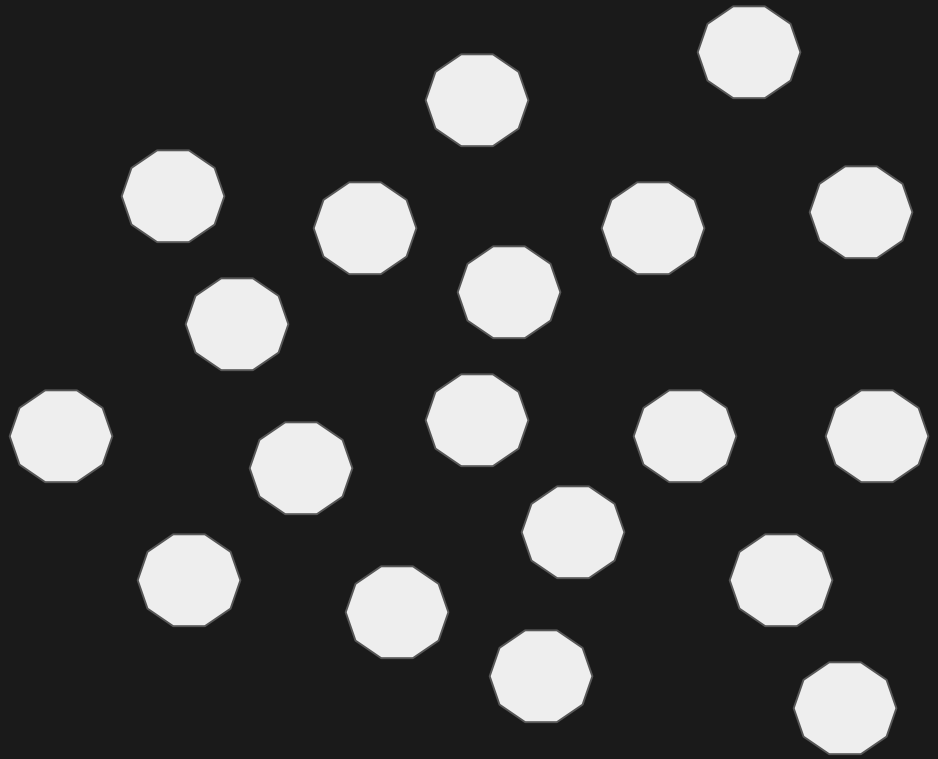
Microservices



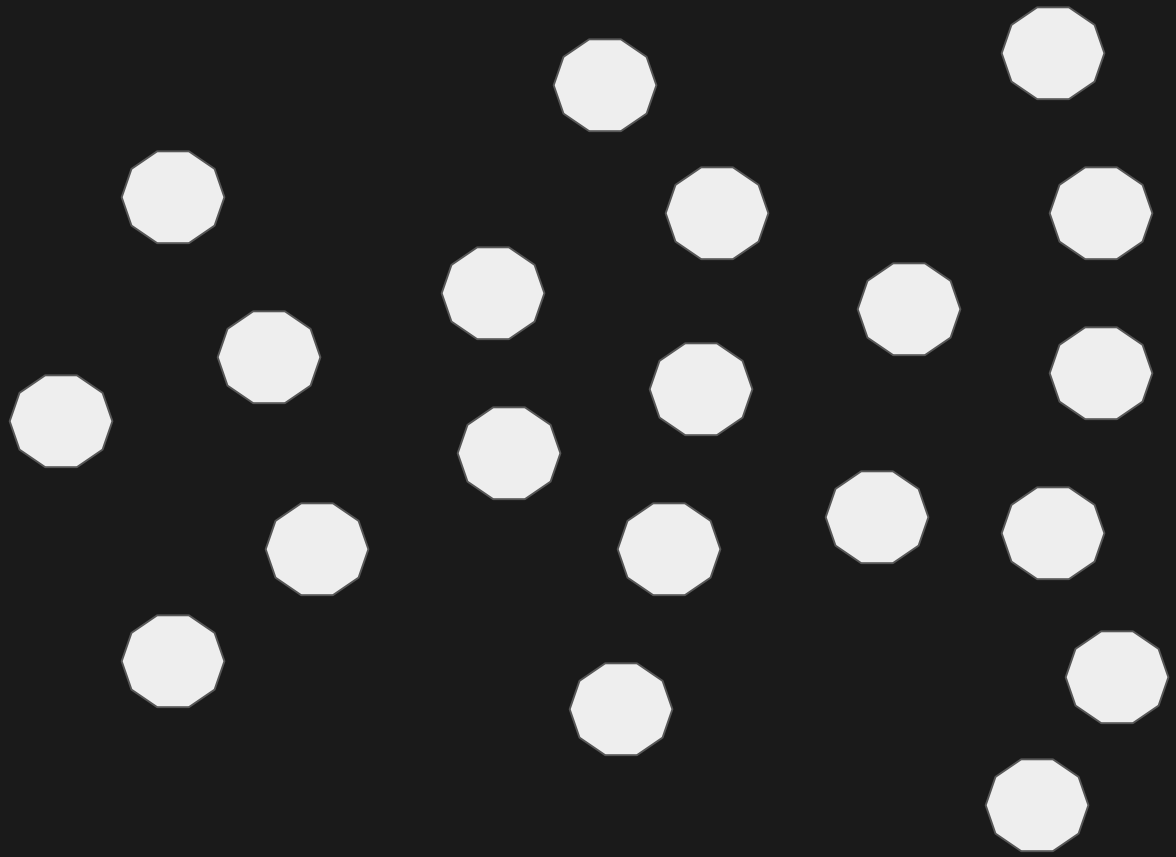
Microservices



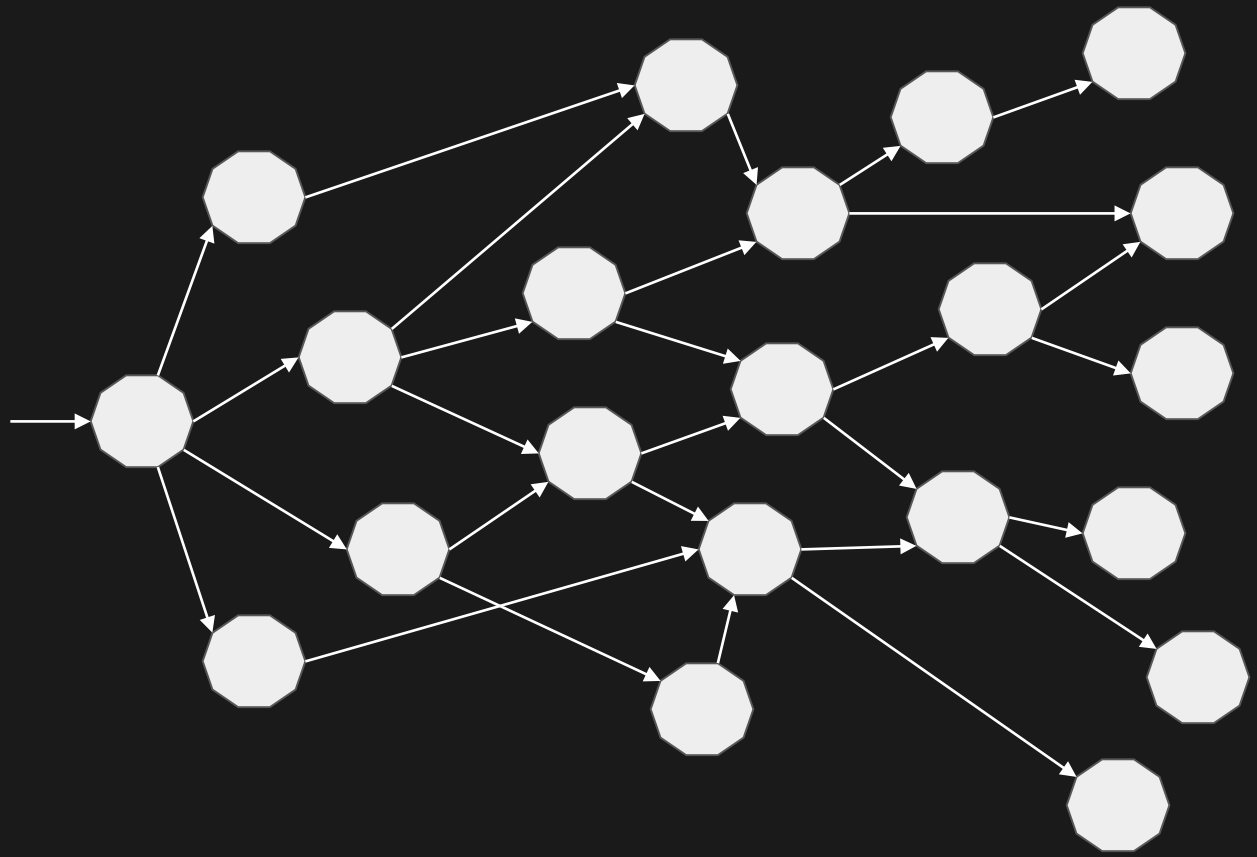
Microservices



Microservices



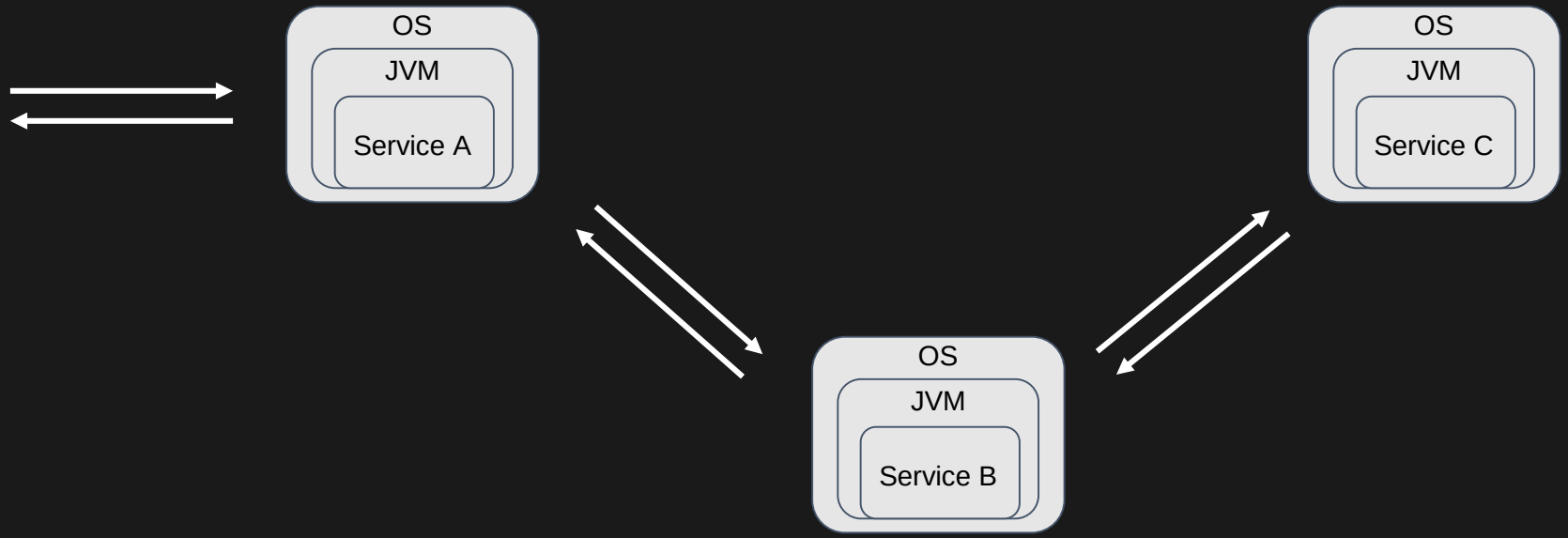
Network of Services



Demo

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

Microservices == Distributed Computing

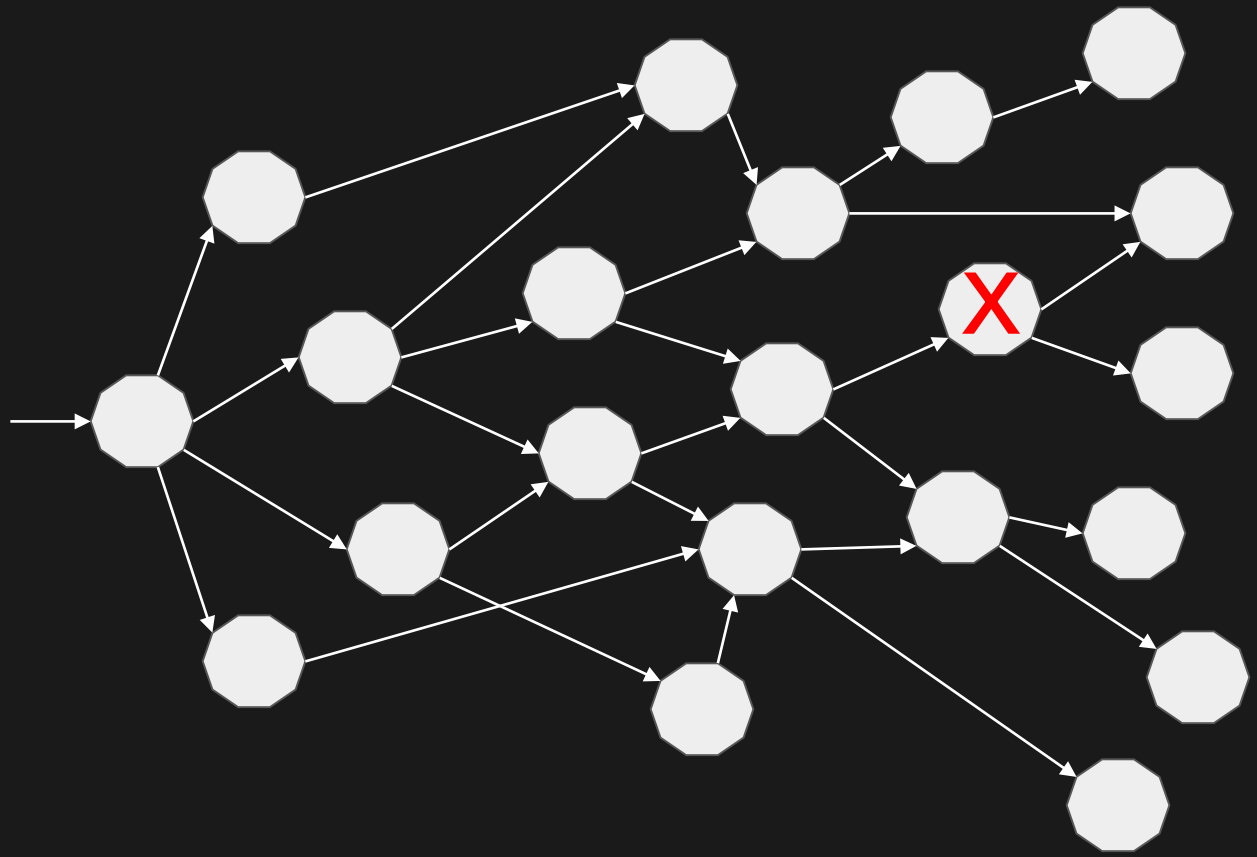


Fallacies of Distributed Computing

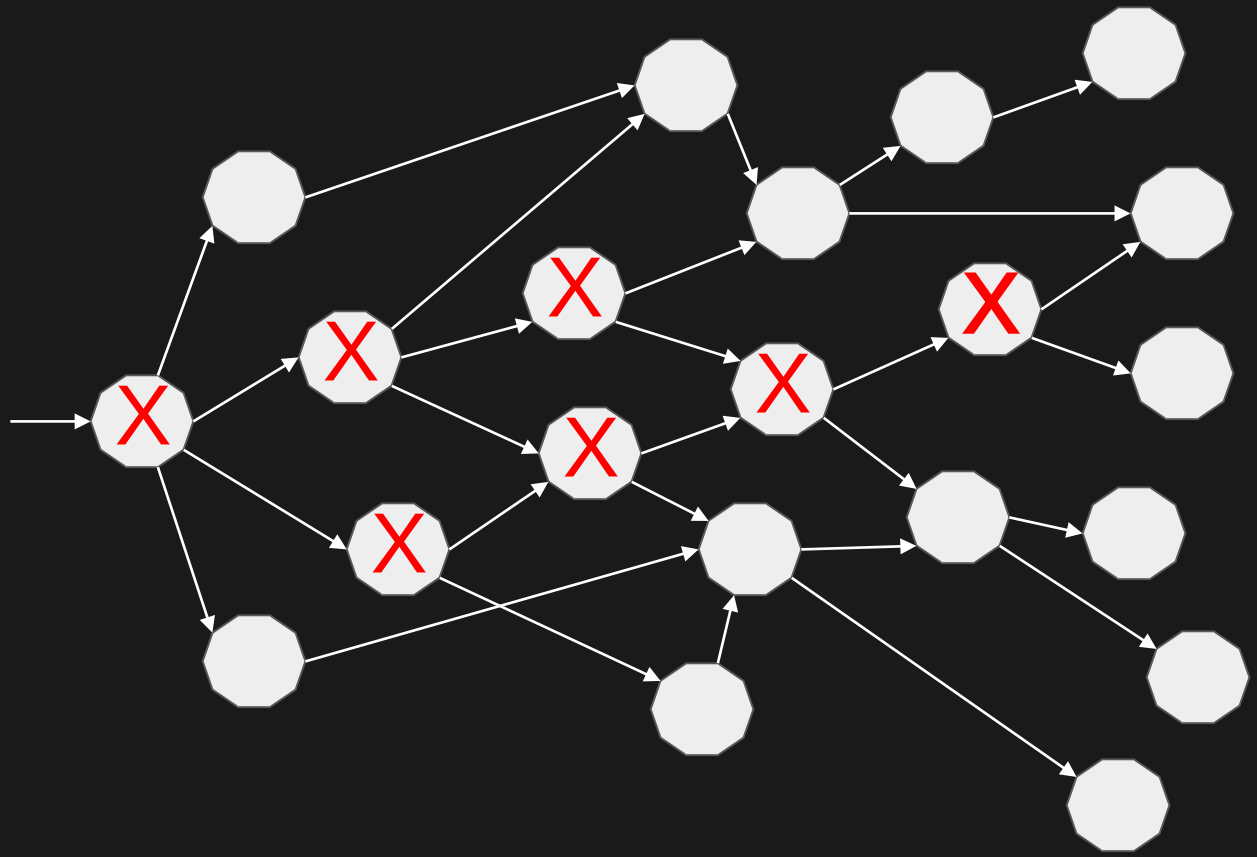
- The Network is Reliable
- Latency is zero
- Bandwidth is infinite
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

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

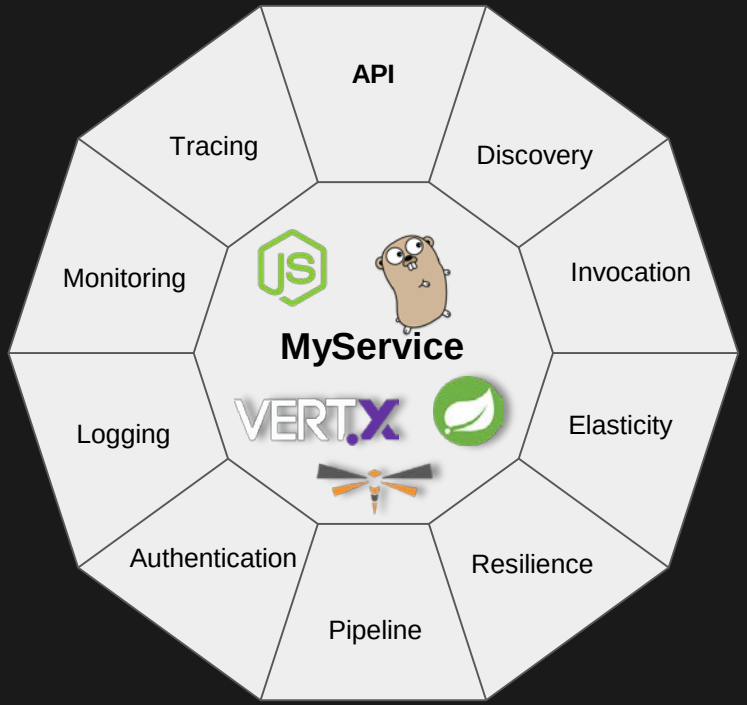
Failure of a Service



Cascading Failure



Microservices'ilities



Short History of Microservices

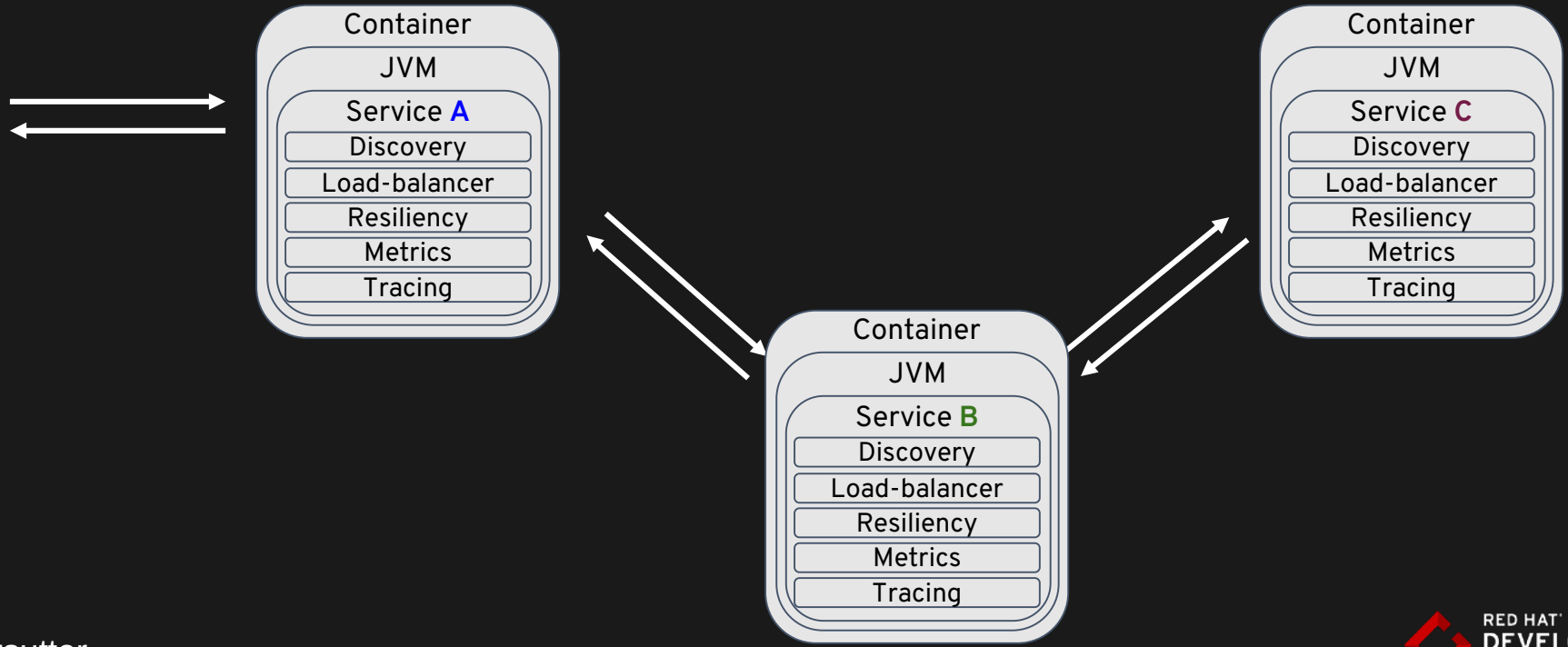
NETFLIX | OSS

What's Wrong with Netflix OSS?

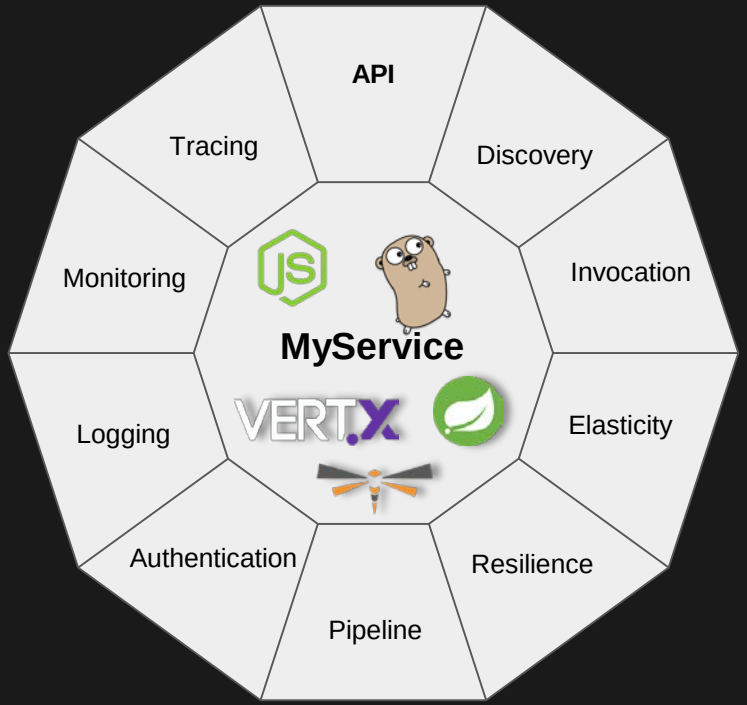
Java Only

Adds a lot of libraries to YOUR code

Microservices embedding Capabilities

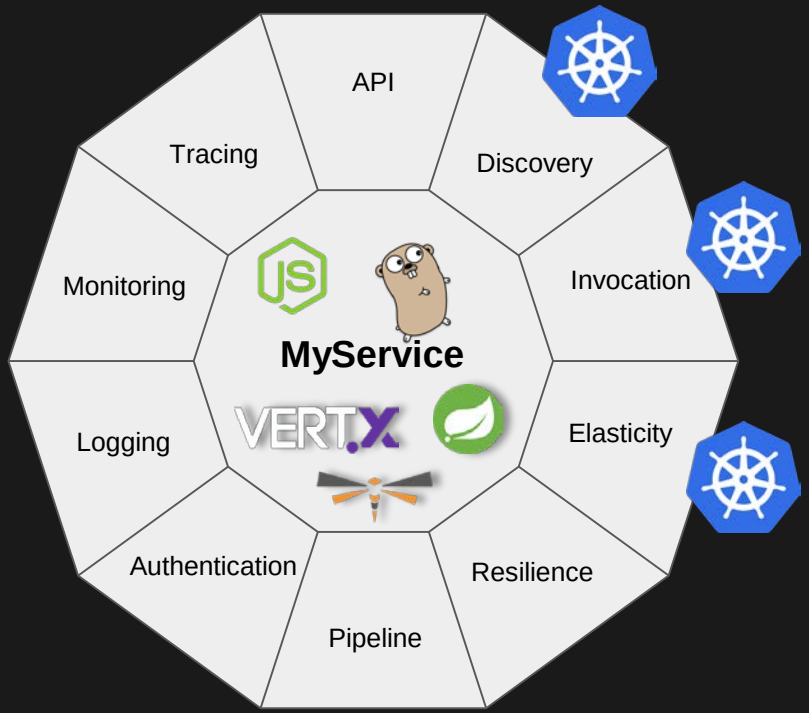


Microservices'ilities

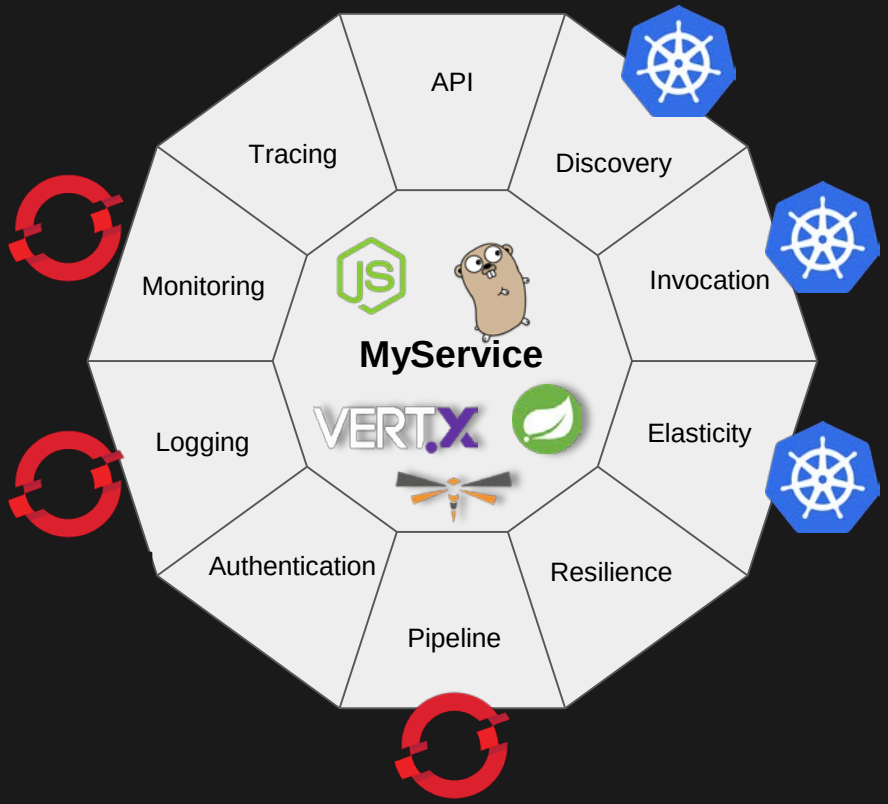


OPENSHIFT

Microservices'ilities + Kubernetes



Microservices'ilities + OpenShift



Istio - Sail

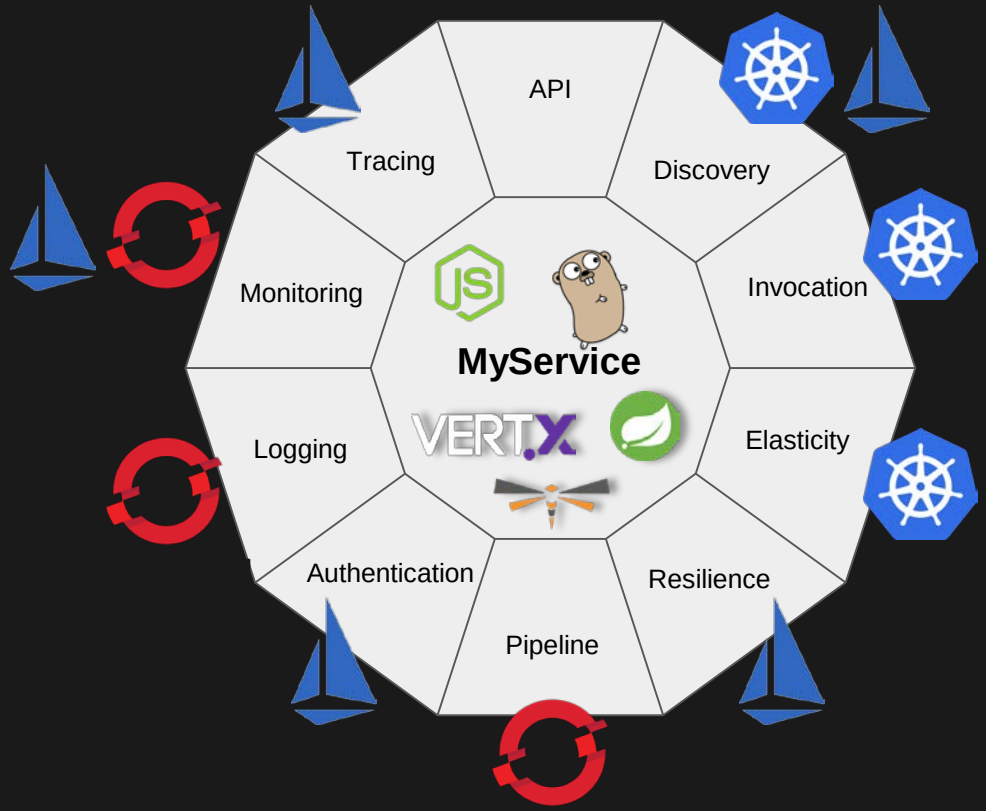
(Kubernetes - Helmsman or ship’s pilot)

Service Mesh Defined

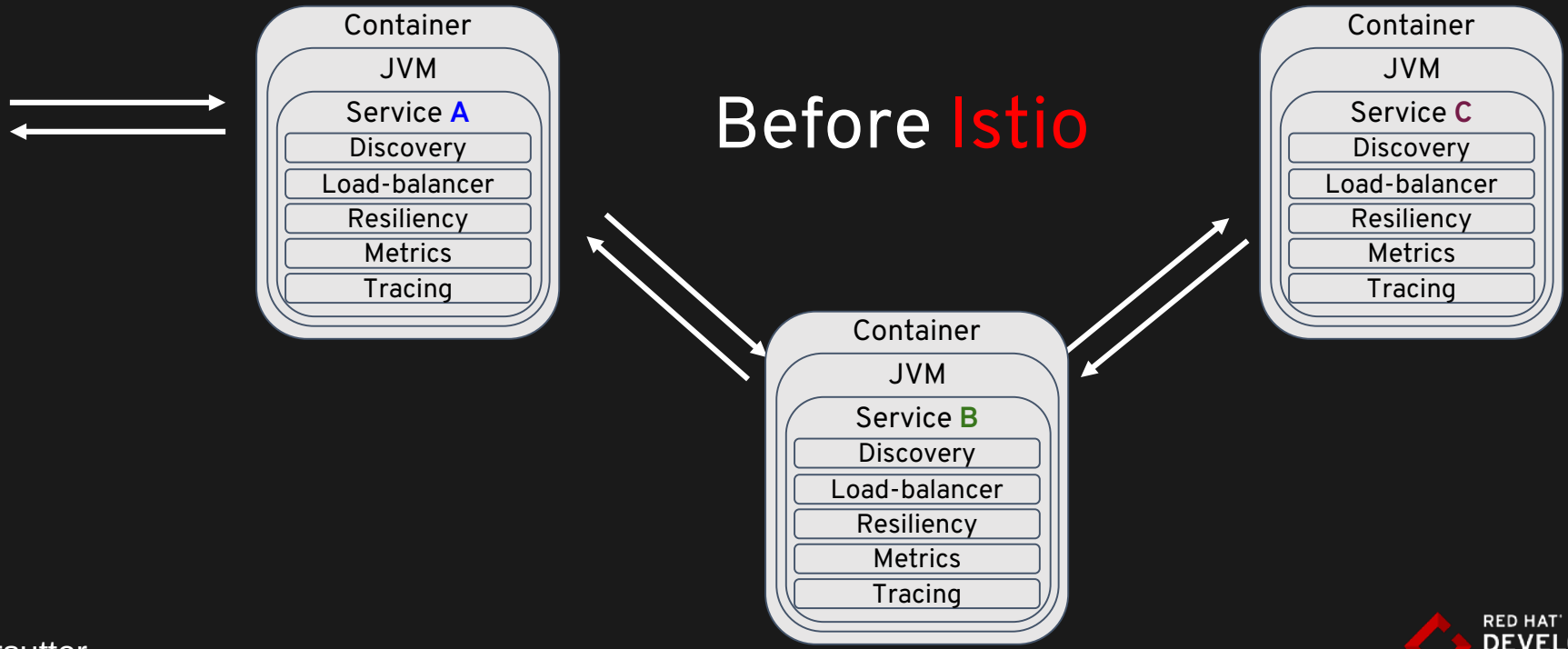
A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as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

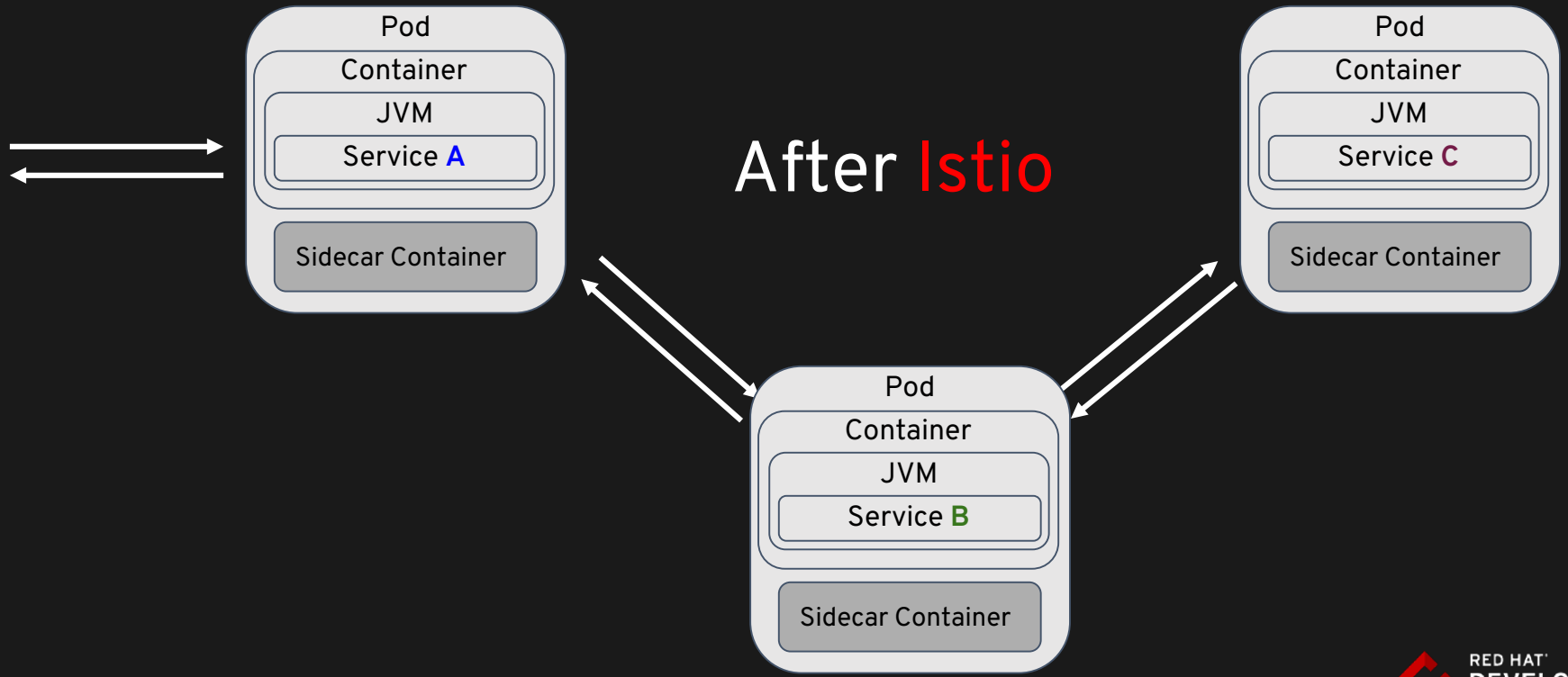
Microservices'ilities + Istio



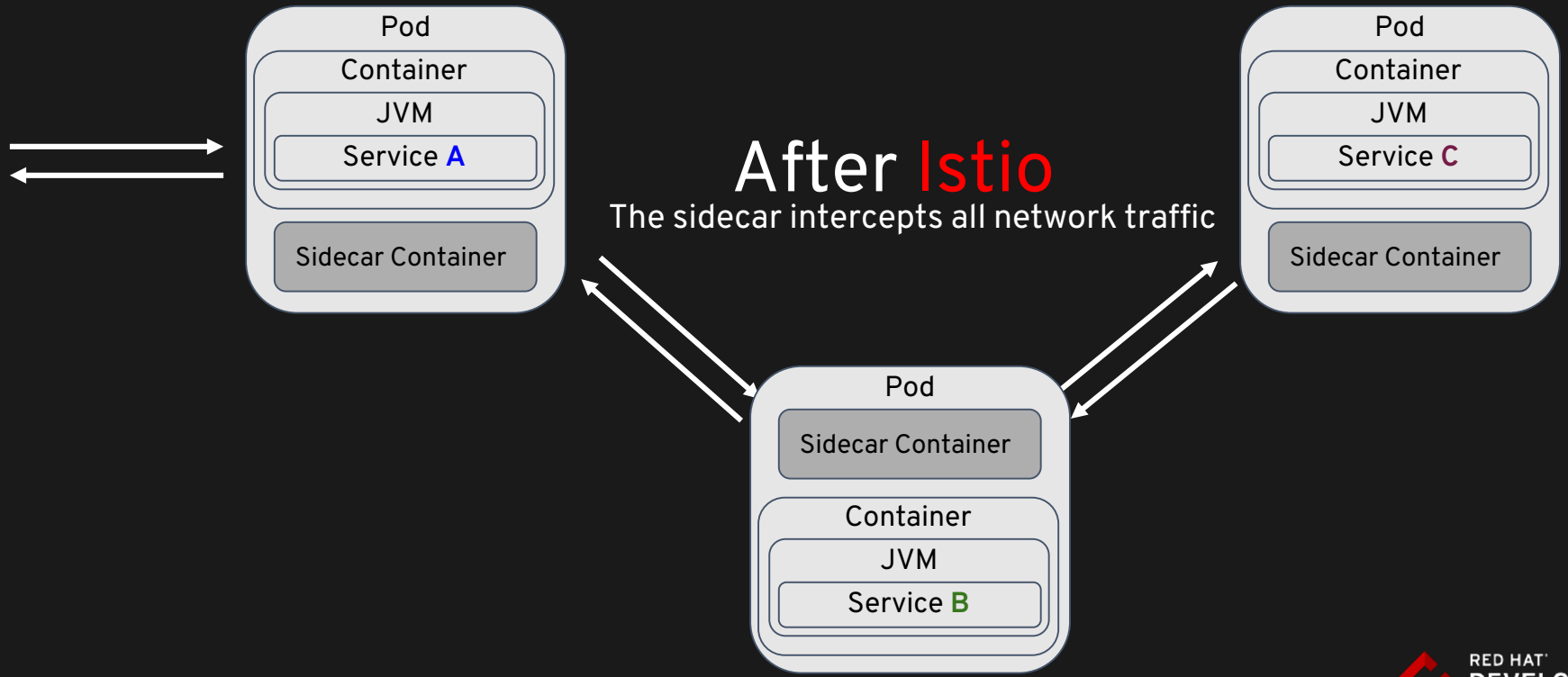
Microservices embedding Capabilities



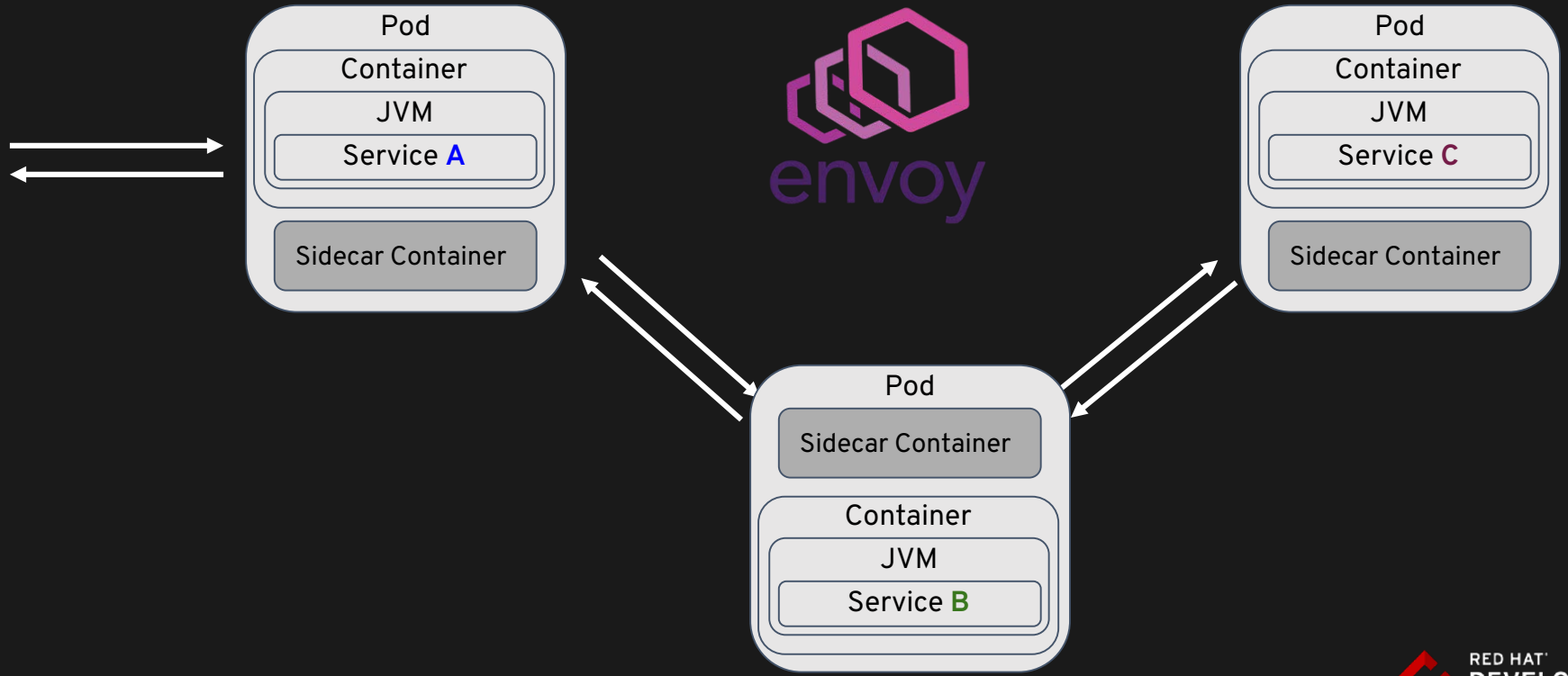
Microservices externalizing Capabilities



Microservices externalizing Capabilities



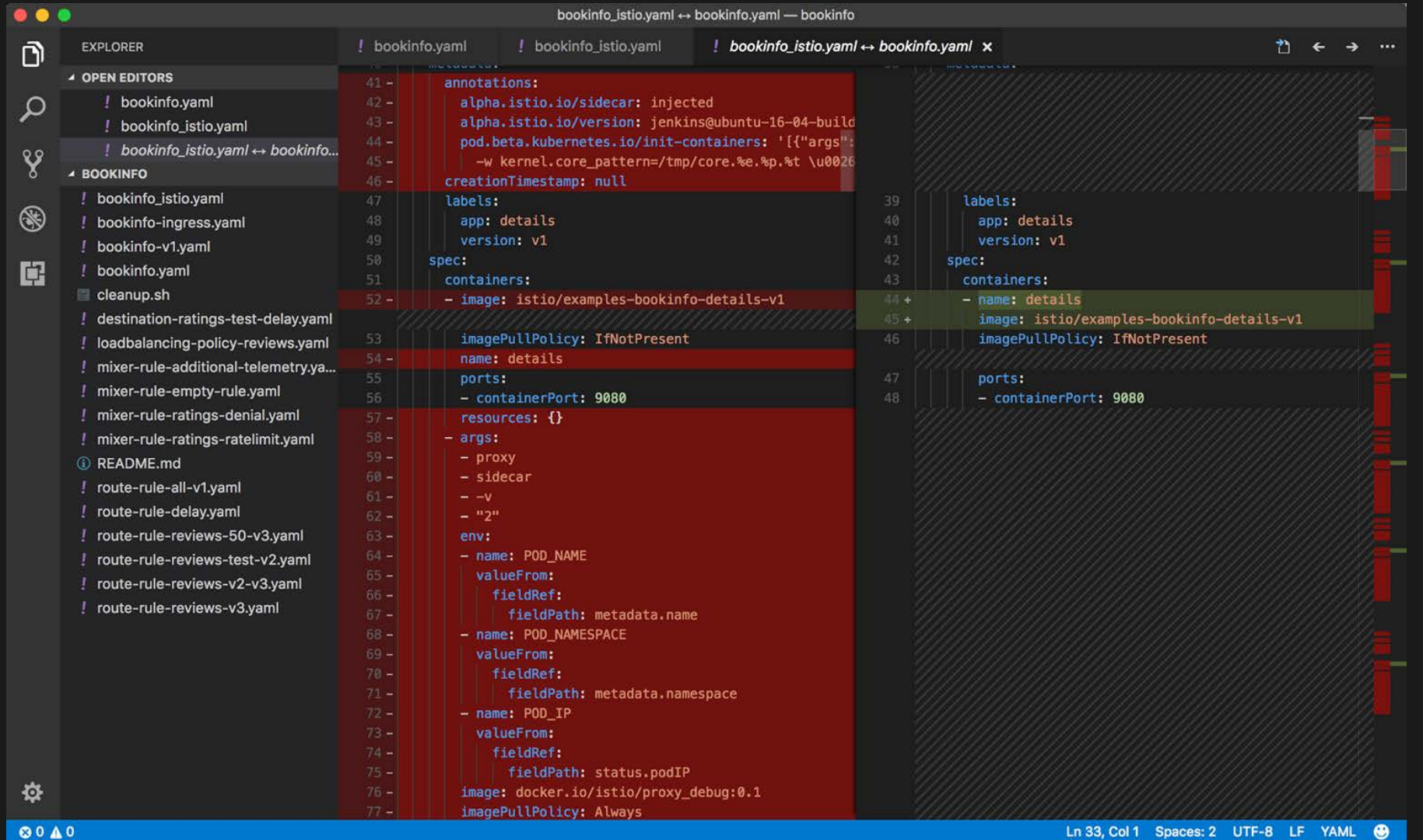
Envoy is the current sidecar



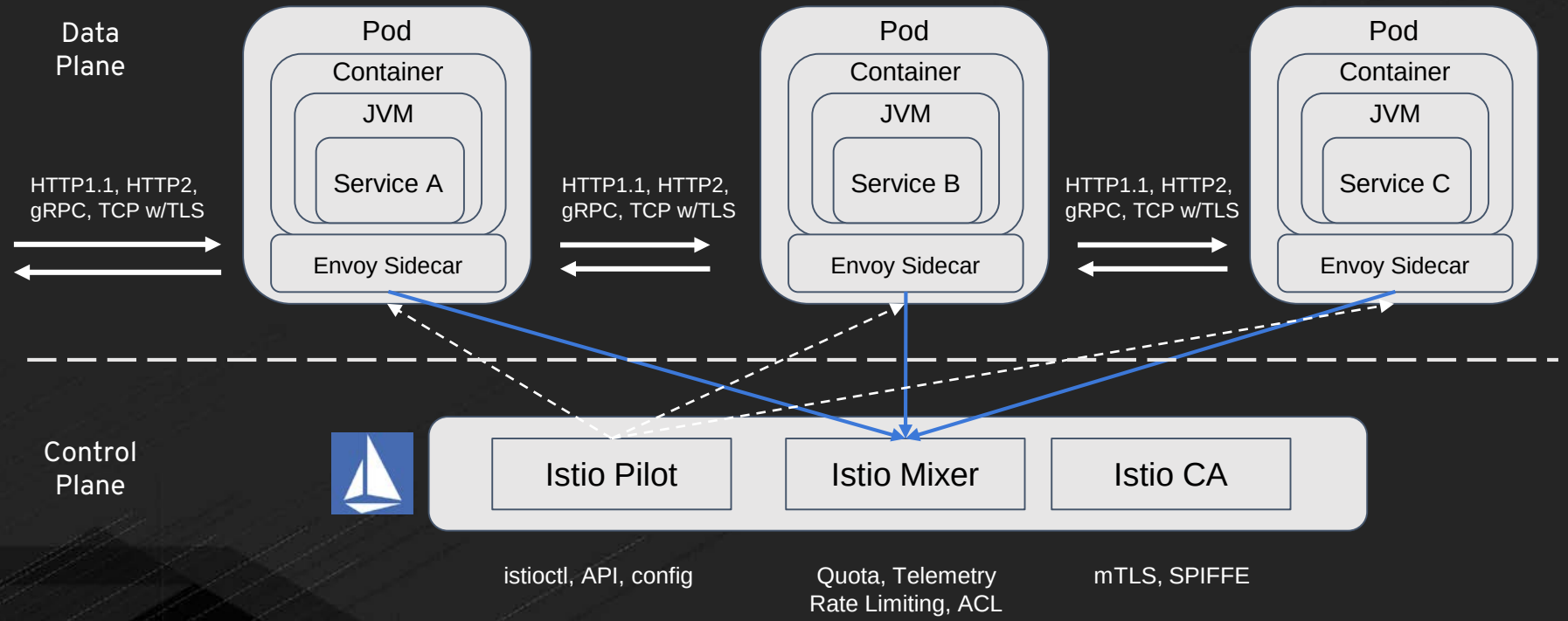
Next Generation Microservices - Service Mesh

Code Independent (Polyglot)

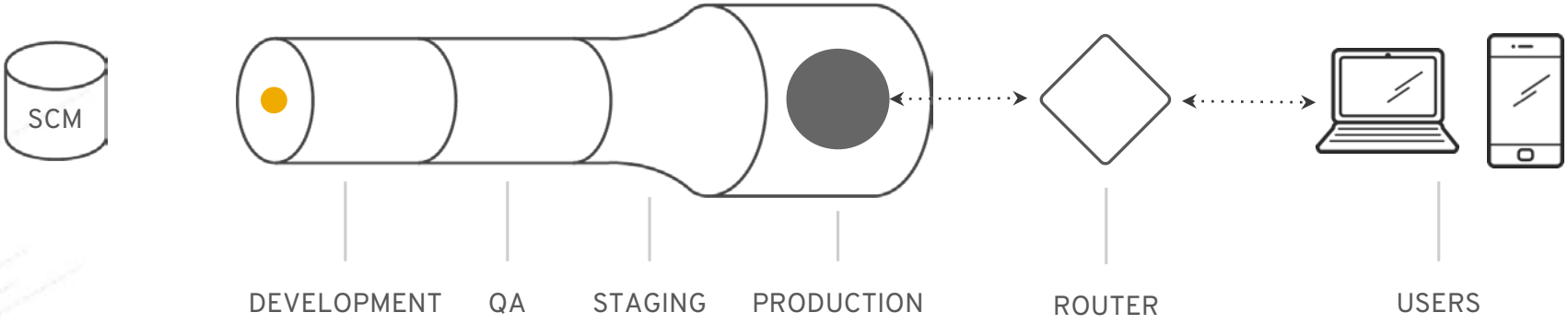
- Intelligent Routing and Load-Balancing
  - A/B Tests
  - Smarter Canary Releases
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability: Metrics and Tracing
- Fleet wide policy enforcement


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

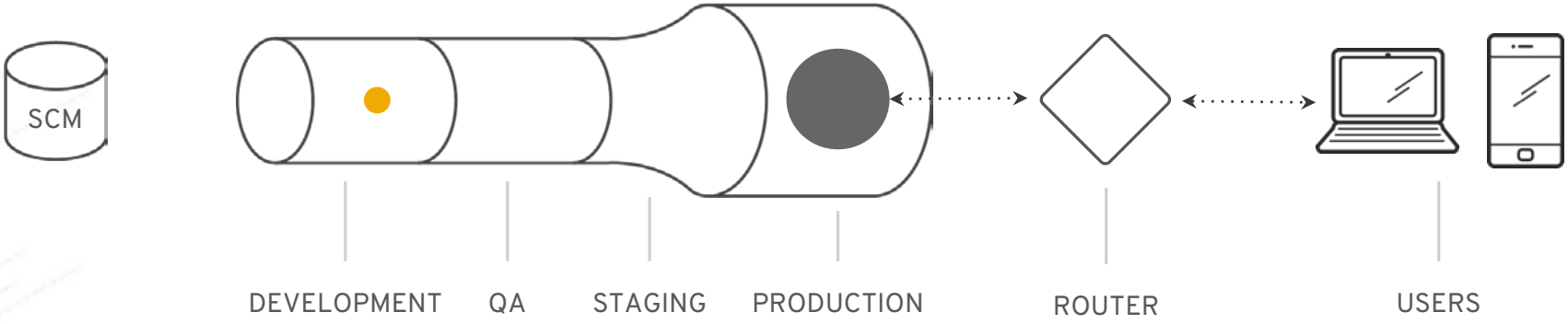
Istio Data Plane vs Control Plane



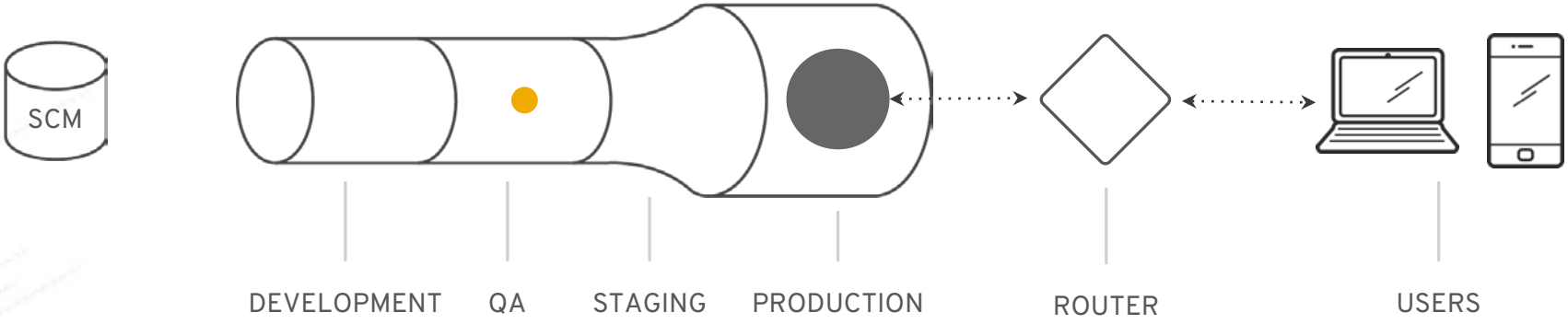
Canary Deployment



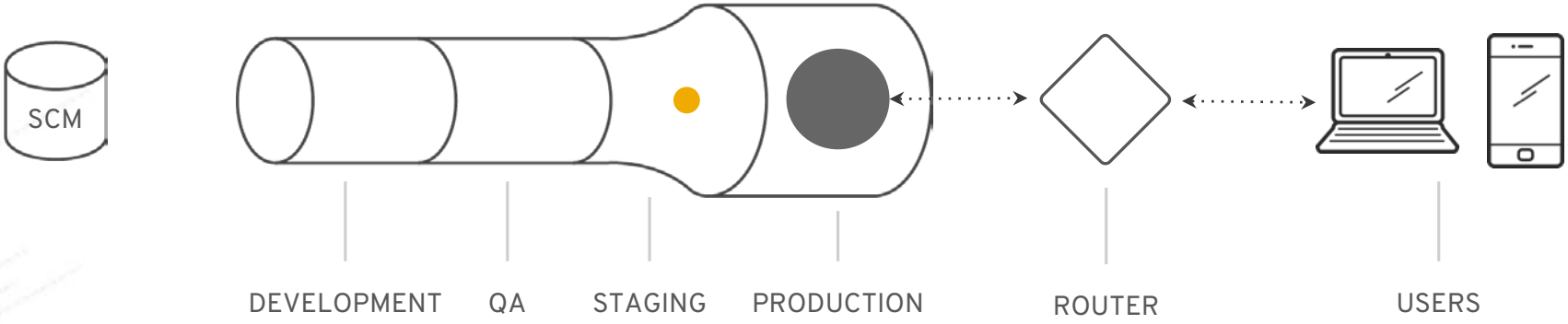
Canary Deployment



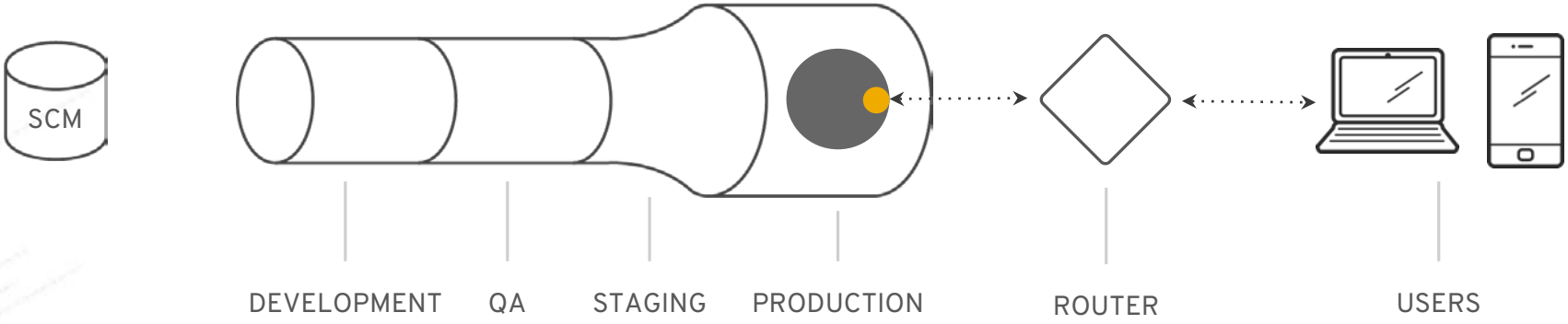
Canary Deployment



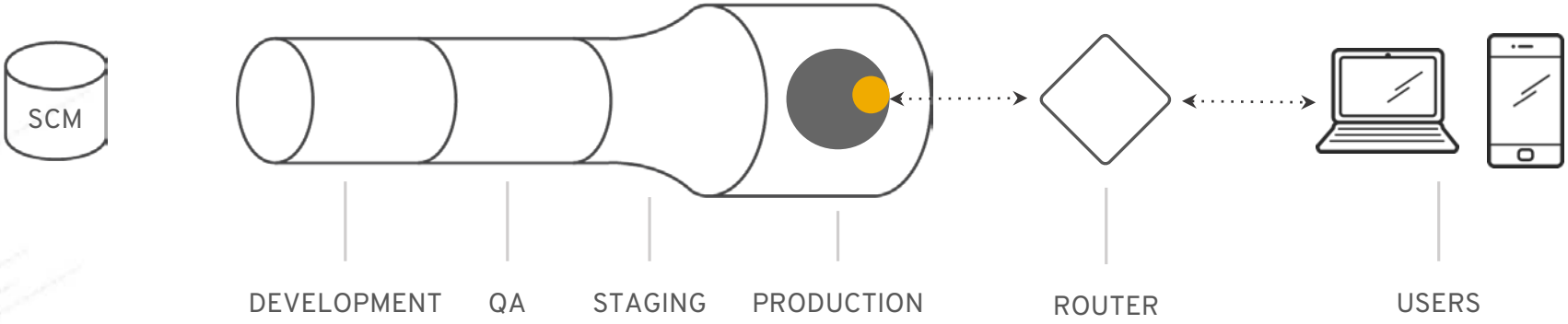
Canary Deployment



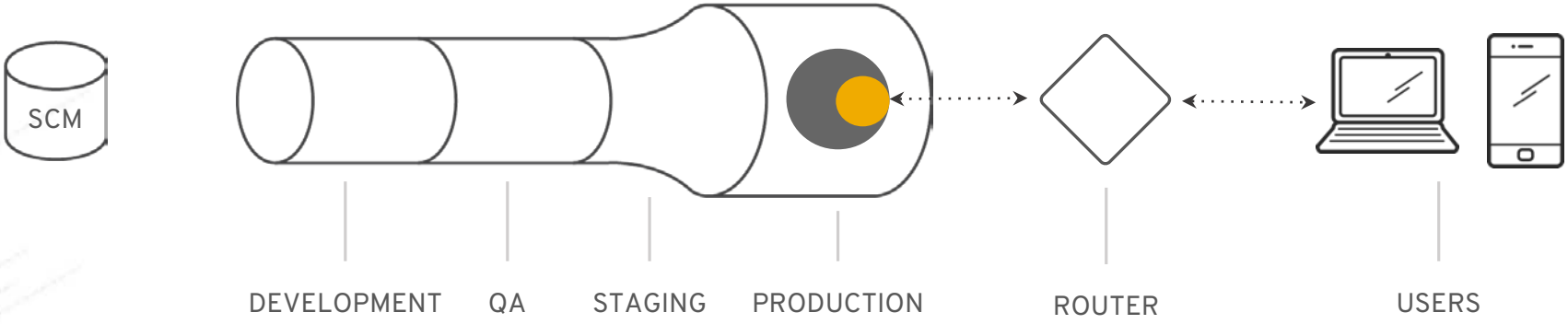
Canary Deployment



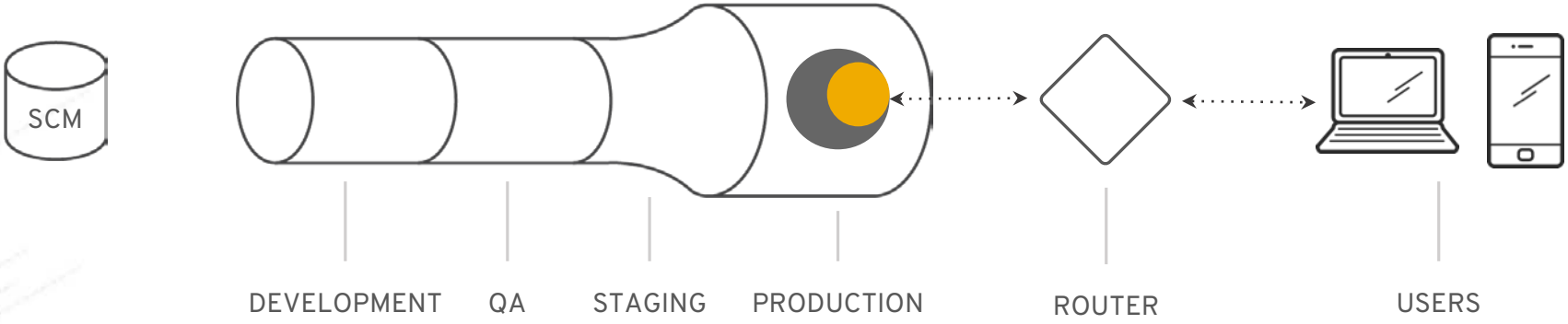
Canary Deployment



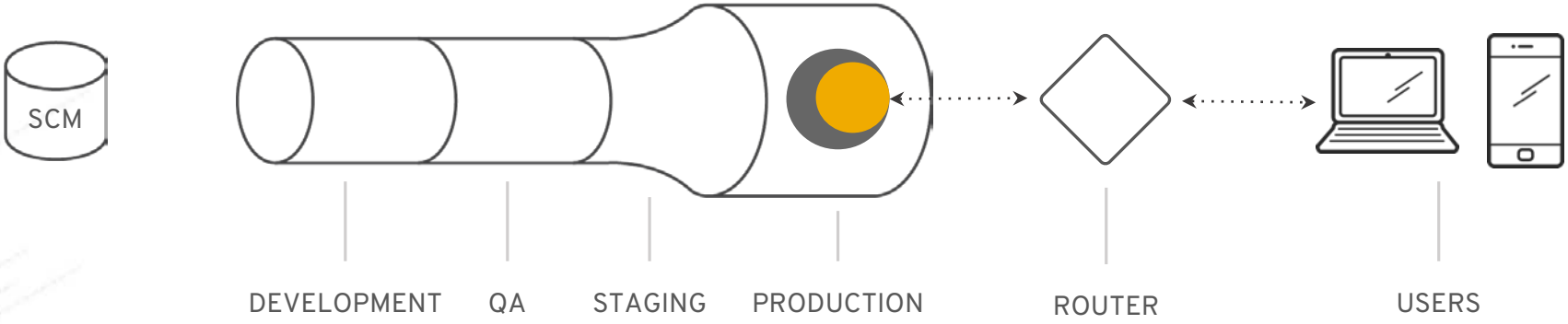
Canary Deployment



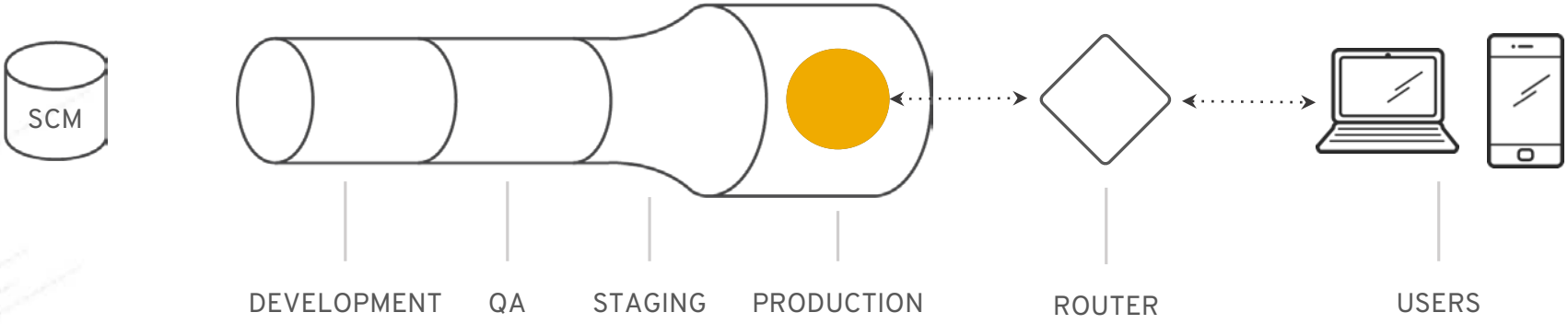
Canary Deployment



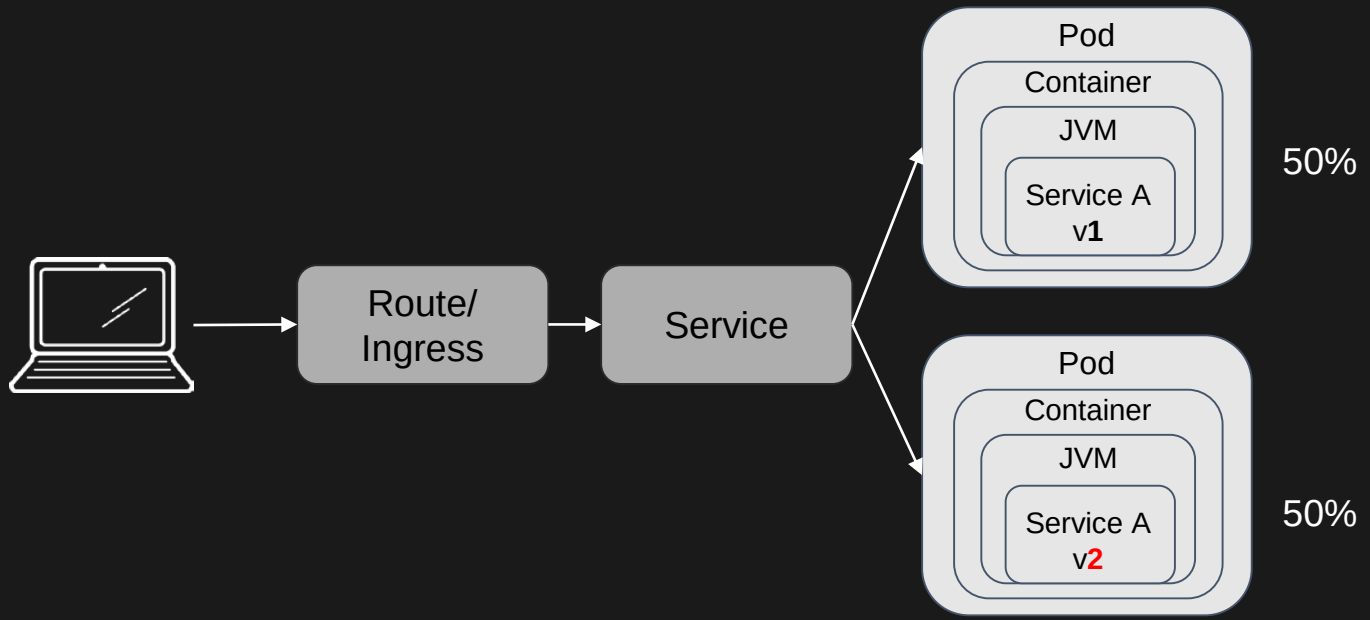
Canary Deployment



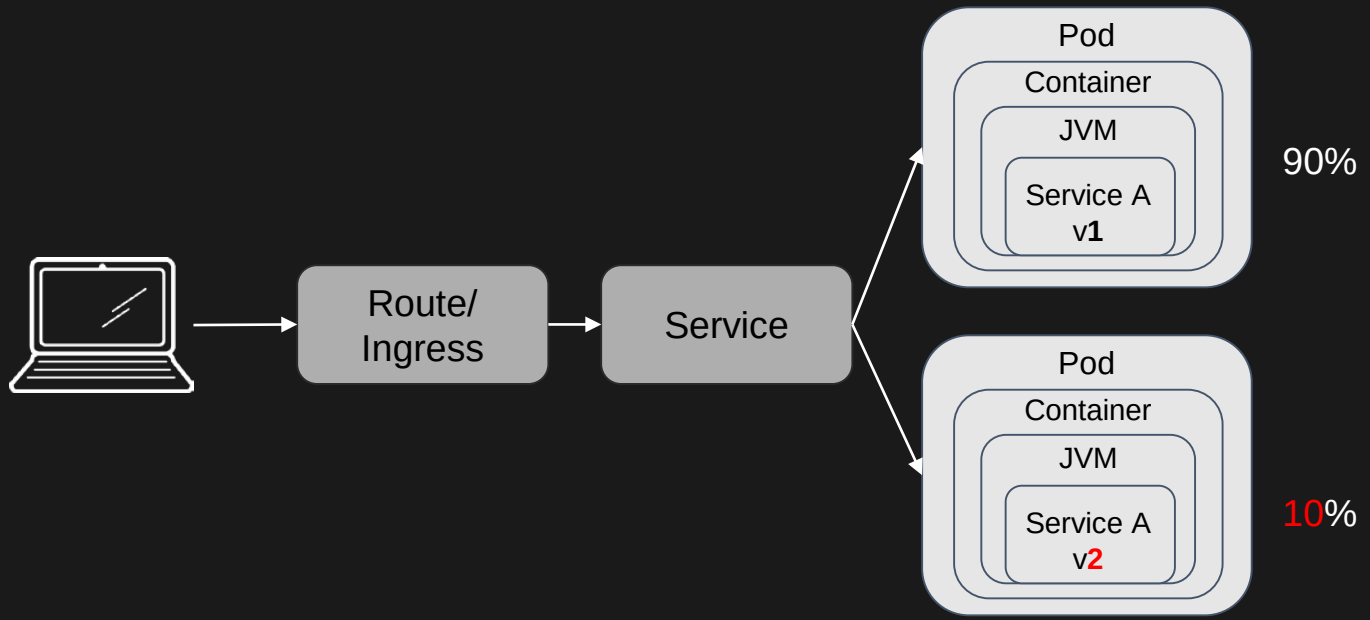
Canary Deployment



Canaries with Kubernetes



Canaries with Istio



Demo

bit.ly/istio-tutorial

Grafana - Istio Dashboard

grafana-istio-system.192.168.99.103.nip.io/d/1/istio-dashboard?refresh=5s&orgId=1

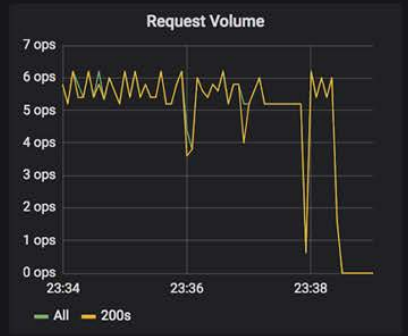
Istio Dashboard

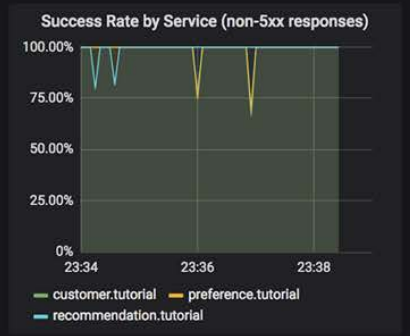
Last 5 minutes | Refresh every 5s

Global Request Volume	Global Success Rate (non-5xx responses)	4xxs	5xxs
4.8 ops	98.9%	N/A	0.056 ops

Service Mesh

Service Mesh





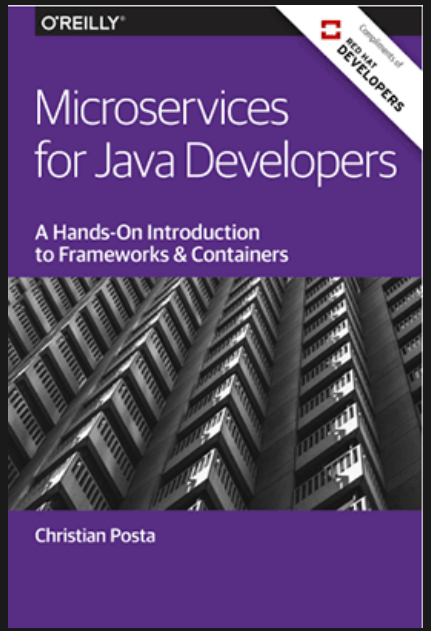
Services

HTTP Services

customer.tutorial.svc.cluster.local


Jaeger UI
Lookup by Trace ID...
Search
Dependencies
About Jaeger
customer: default-route
View Options
Search...
Trace Start: March 22, 2018 11:38 PM
Duration: 9.01ms
Services: 4
Depth: 6
Total Spans: 7
0ms
2.25ms
4.51ms
6.76ms
9.01ms
Service & Operation
customer default-route
customer default-route
preference default-route
preferences getPreferences
preferences GET
preference default-route
recommendation default-route
6.14ms
5.74ms
4.83ms
3.76ms
2.22ms
1.5ms

bit.ly/javamicroservicesbook

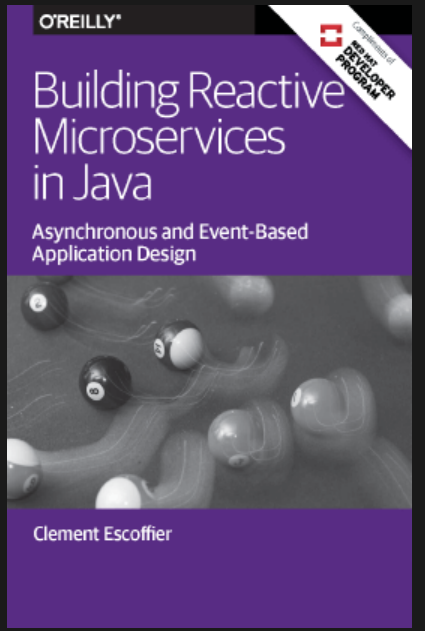


Free eBooks from developers.redhat.com

Microservices Introductory Materials

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

bit.ly/reactivemicroservicesbook



Microservices Advanced Materials

bit.ly/istio-tutorial

bit.ly/mono2microdb

http://bit.ly/istio-intro

Istio.io

@burrsutter

RED HAT DEVELOPER PROGRAM

istio pilot: Kinds

EgressRule: Routing (to services outside of the istio service mesh)

RouteRule: Routing (within the service mesh), Retries, Mirroring, Fault Injection

DestinationPolicy: Load Balancing, Pool Ejection, Circuit Breaker, CORS

istio mixer: Kinds (1 of 2)

denier: Blacklist

checknothing: Blacklist

listchecker: Whitelist

listentry: Whitelist

metric: Request Count (for monitoring)

Prometheus: Metrics for Prometheus monitoring

istio mixer: Kinds (2 of 2)

quota: Rate Limits

memquota: Rate Limits

rule: Whitelist, Rate Limits, Request Count (monitoring)