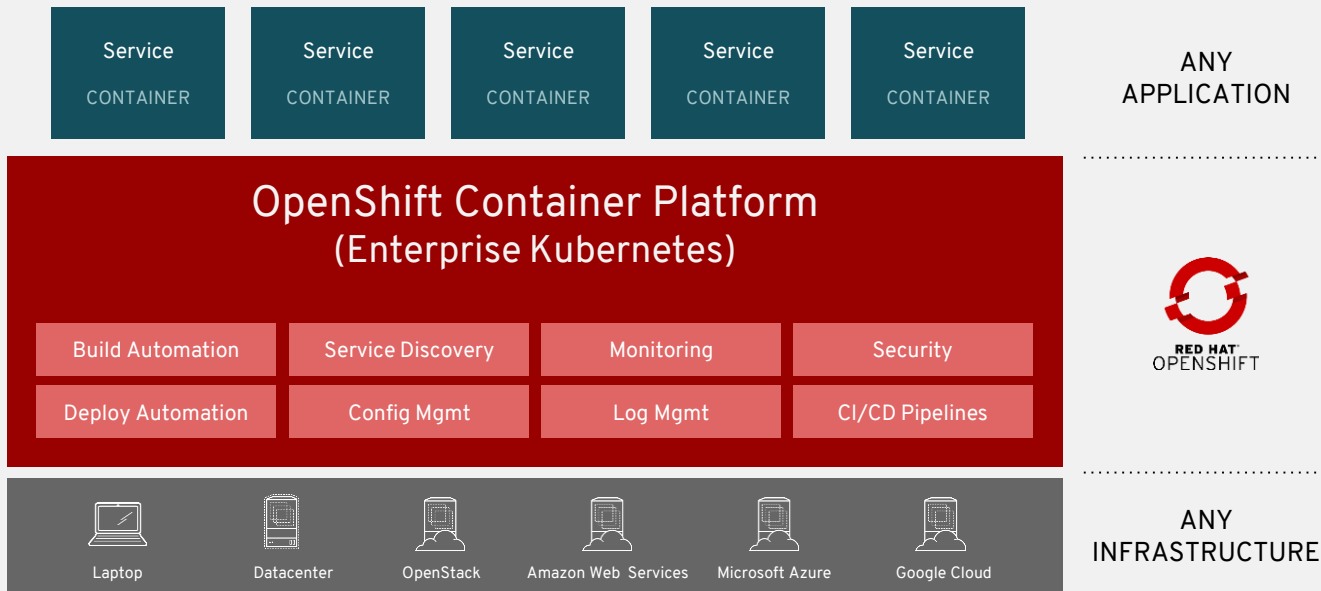




Simplify Microservices with Service-Mesh Technology

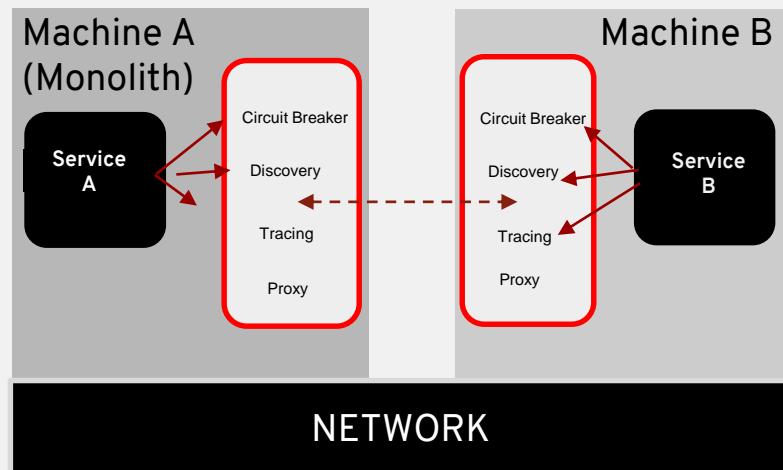
Vaibhav Jain
Senior Specialist Solutions Architect

BUILD AND DEPLOY CLOUD-NATIVE APPS WITH RED HAT OPENSSHIFT

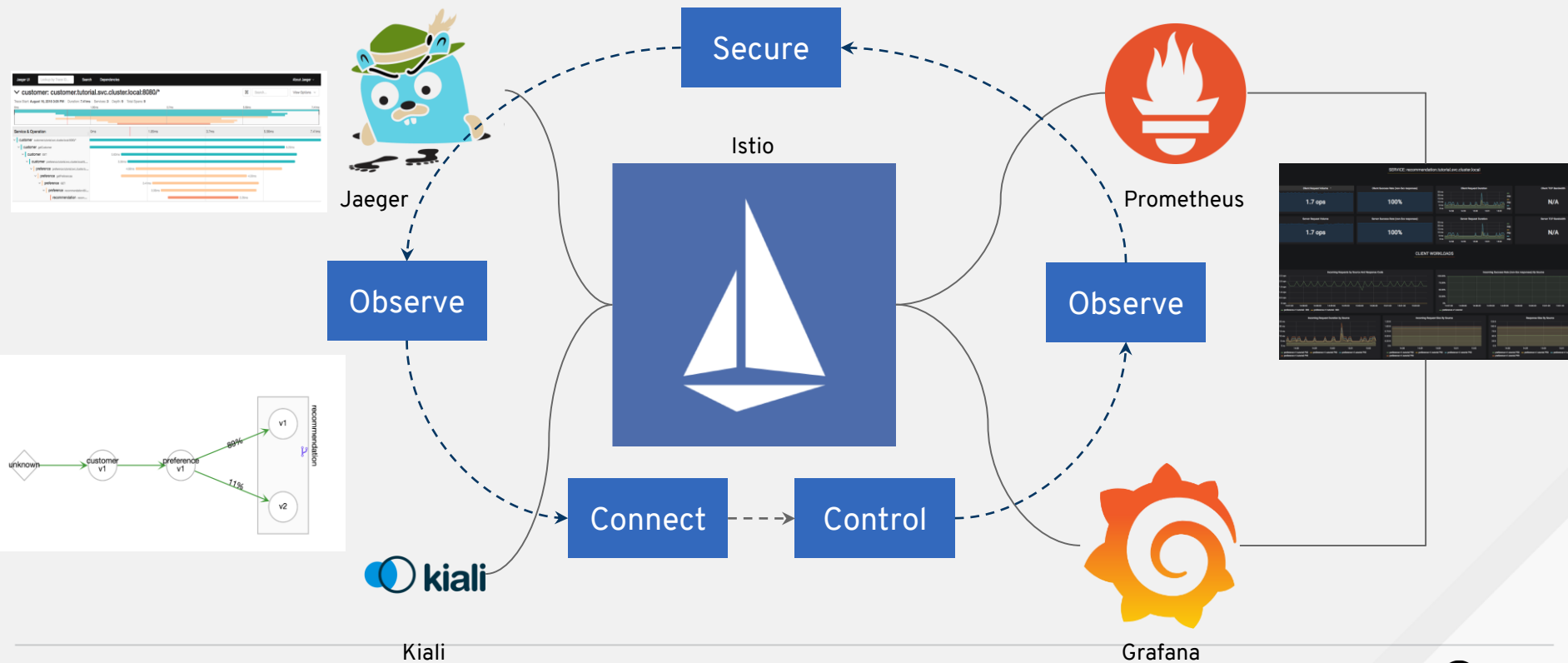


OVERVIEW

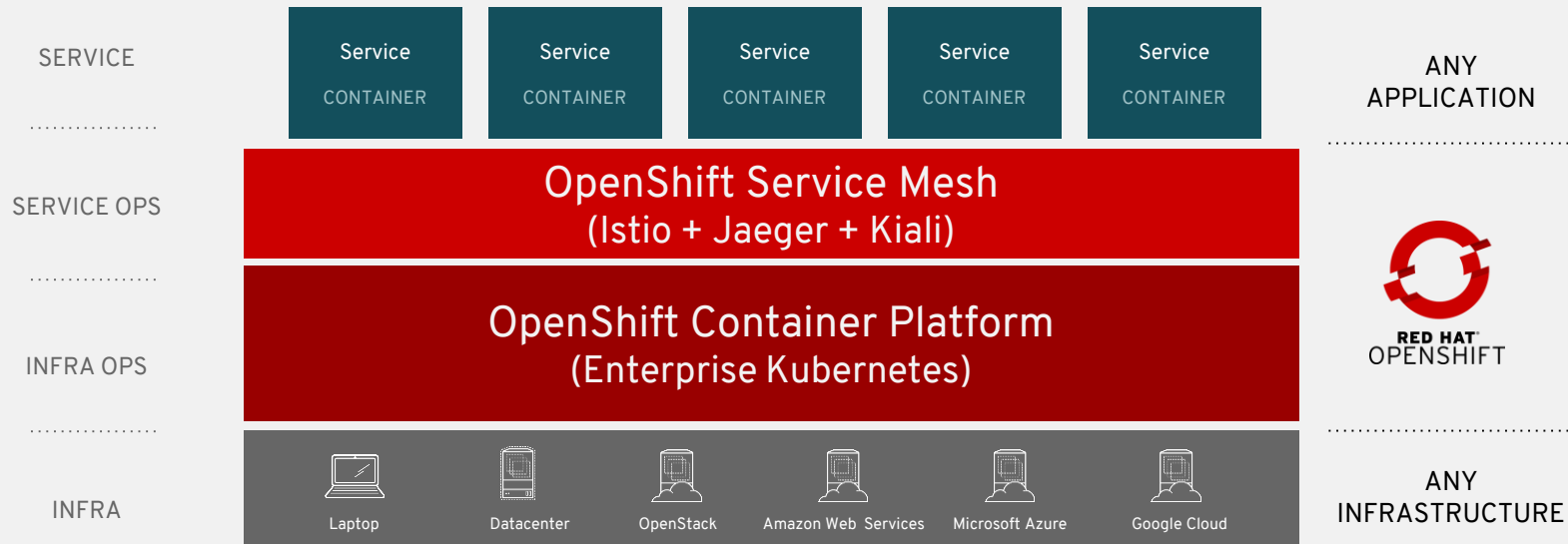
WHAT IS A SERVICEMESH ?



SERVICE MESH ECOSYSTEM

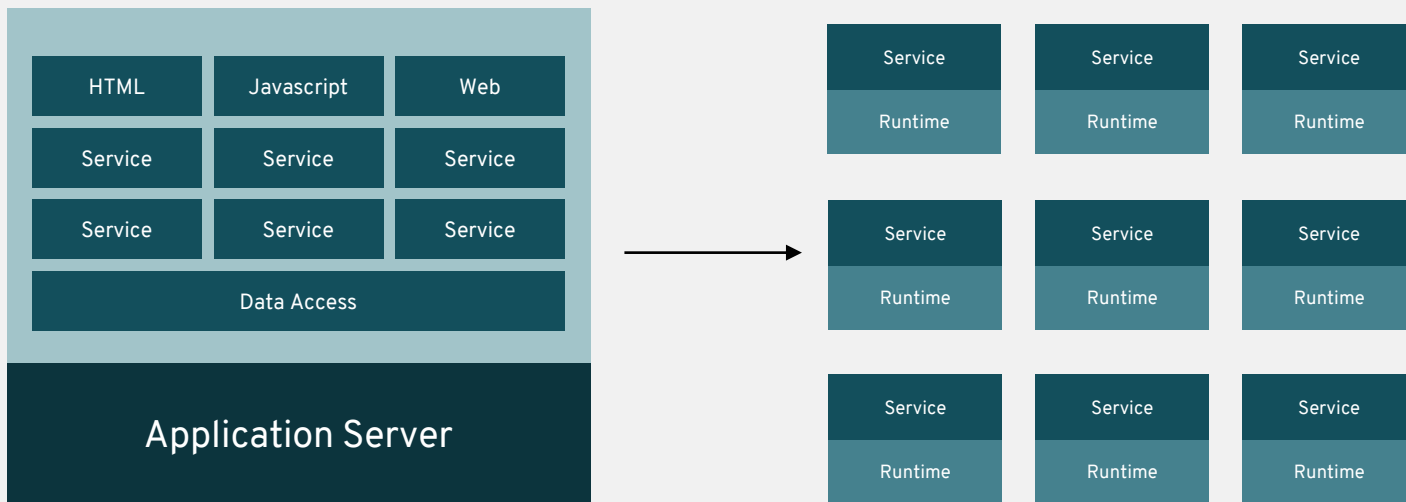


DISTRIBUTED SERVICES WITH RED HAT OPENSIFT SERVICE MESH



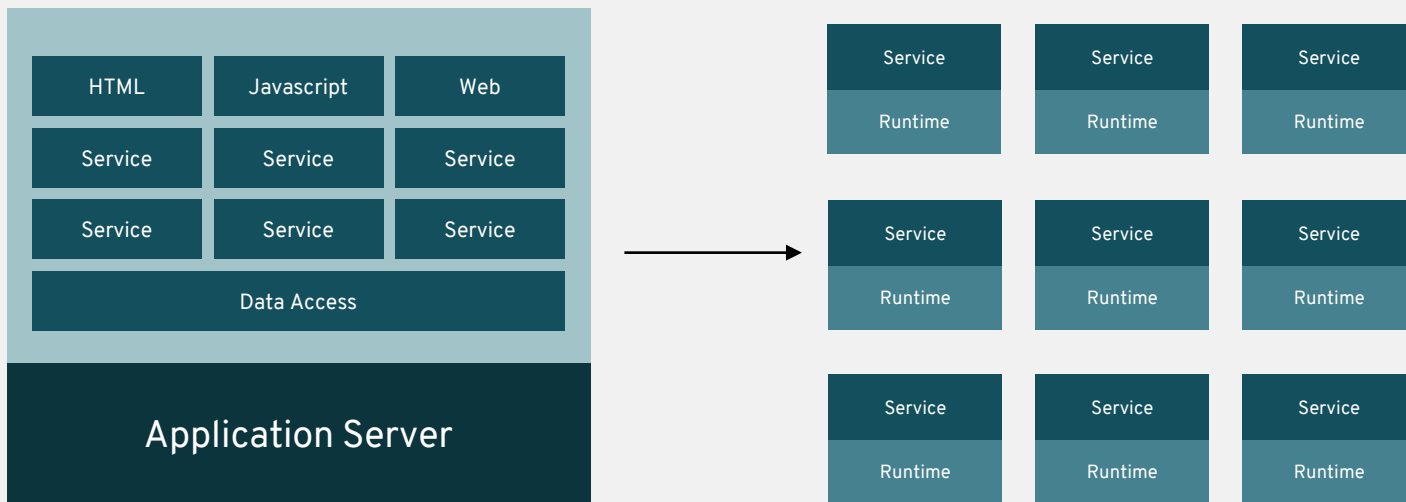
UNDER THE HOOD

MICROSERVICES ARCHITECTURE

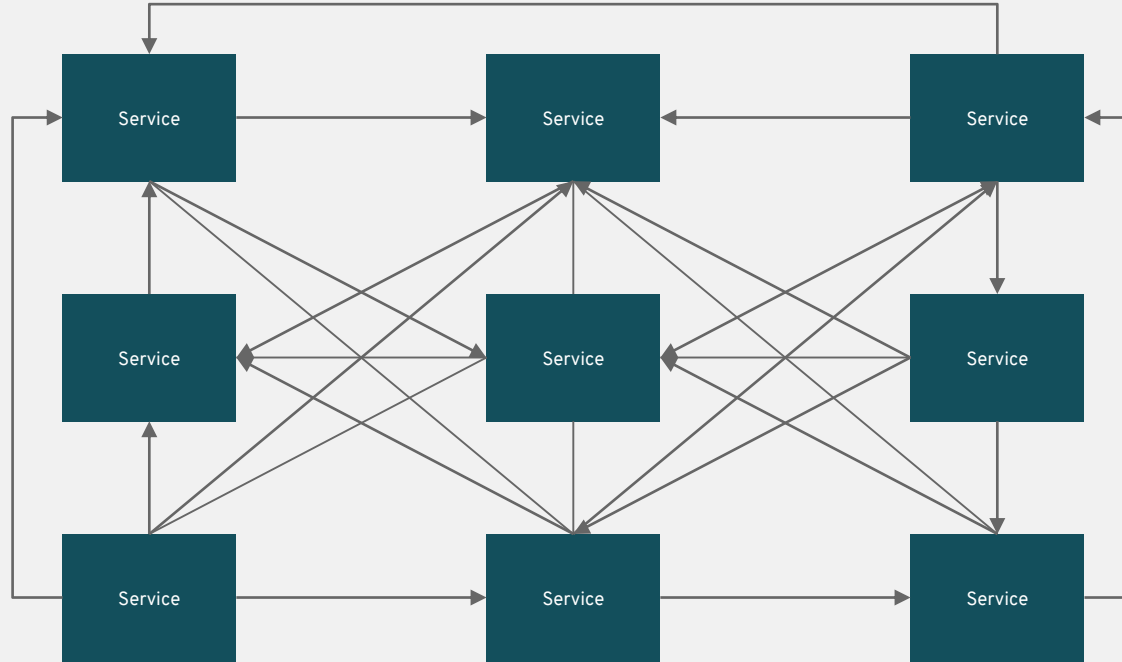


~~MICROSERVICES~~ ARCHITECTURE

DISTRIBUTED



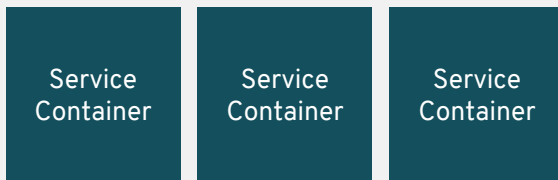
DISTRIBUTED ARCHITECTURE





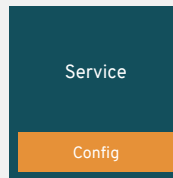
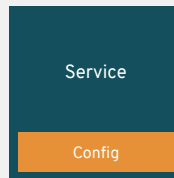
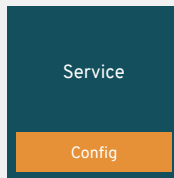
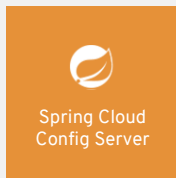
HOW TO DEAL WITH THE COMPLEXITY?

DEPLOYMENT



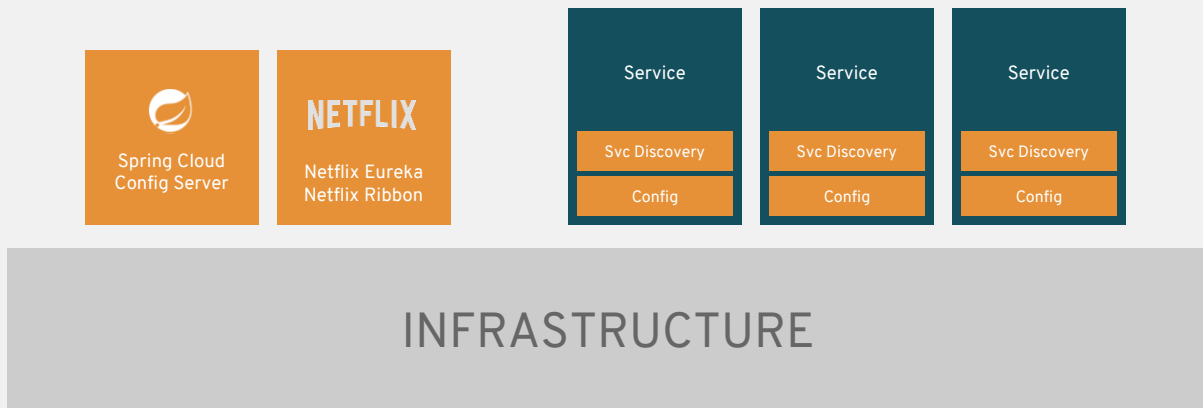
INFRASTRUCTURE

CONFIGURATION

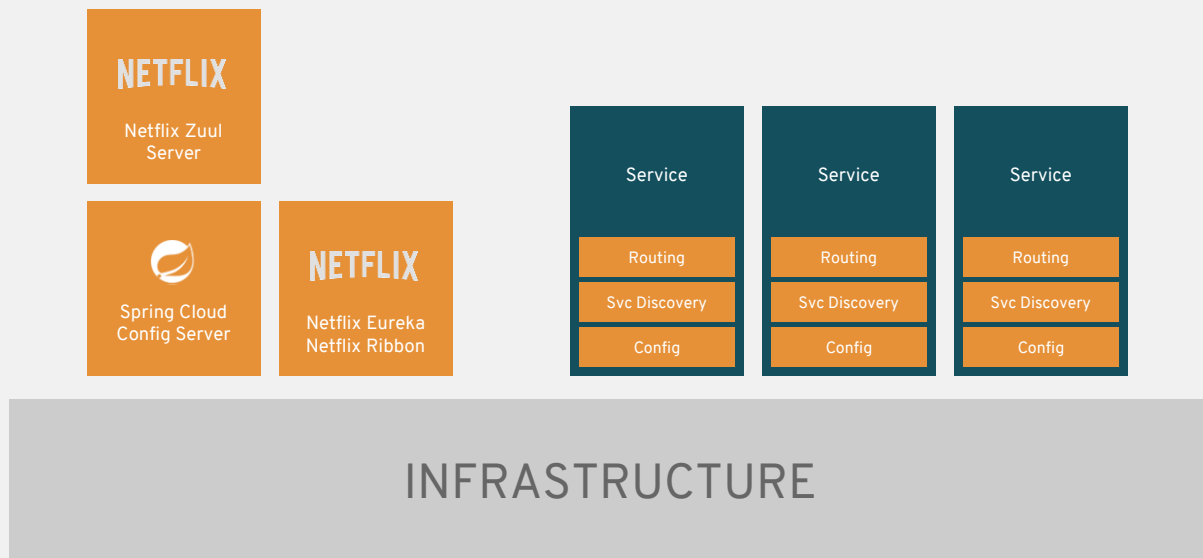


INFRASTRUCTURE

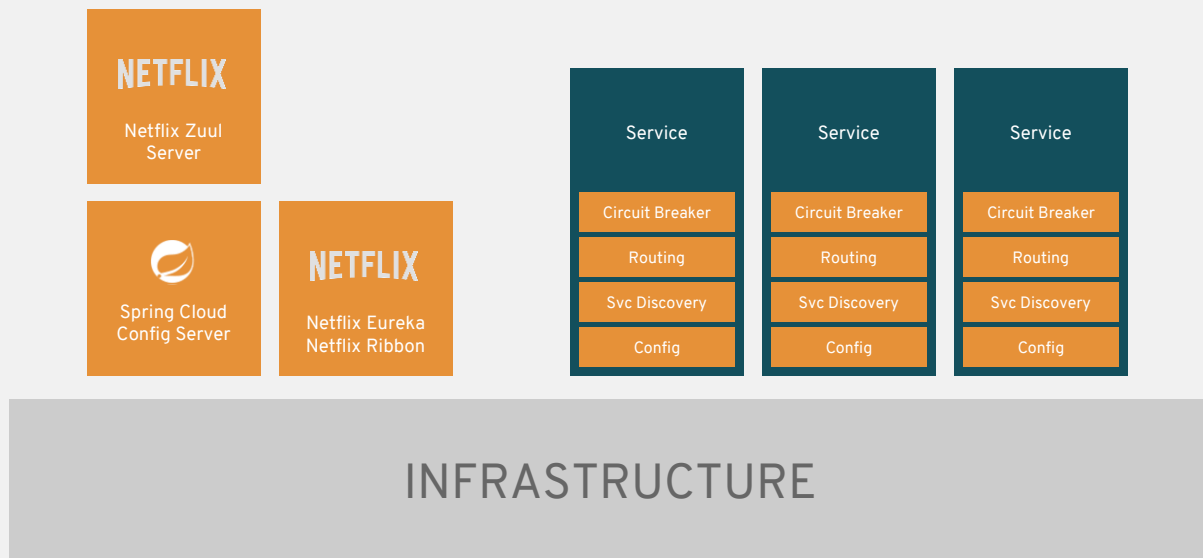
SERVICE DISCOVERY



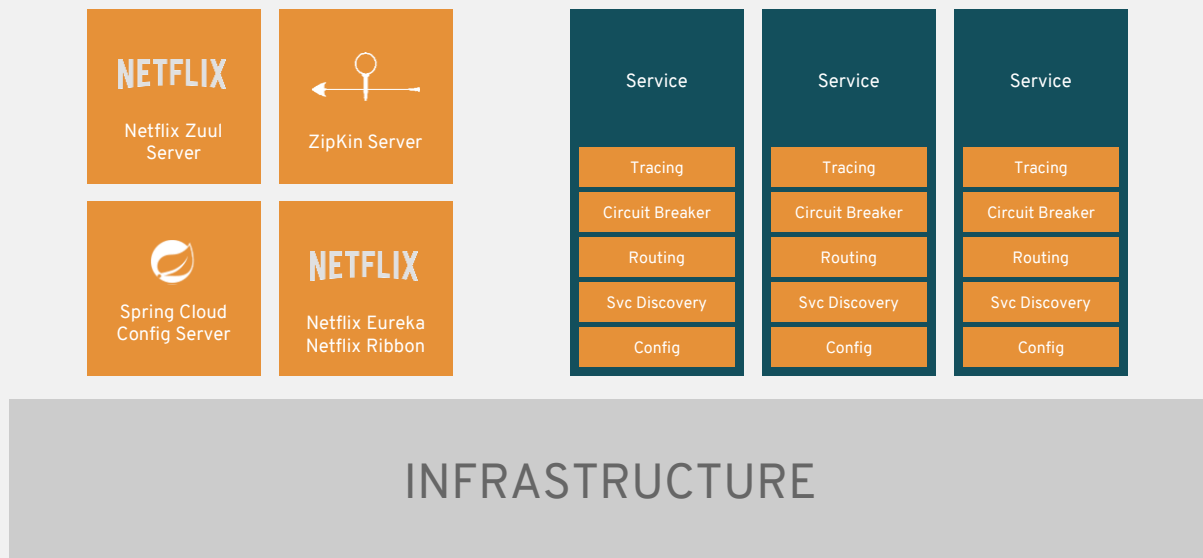
DYNAMIC ROUTING



FAULT TOLERANCE



TRACING AND VISIBILITY



WHAT ABOUT...?

POLYGLOT
APPS



EXISTING
APPS

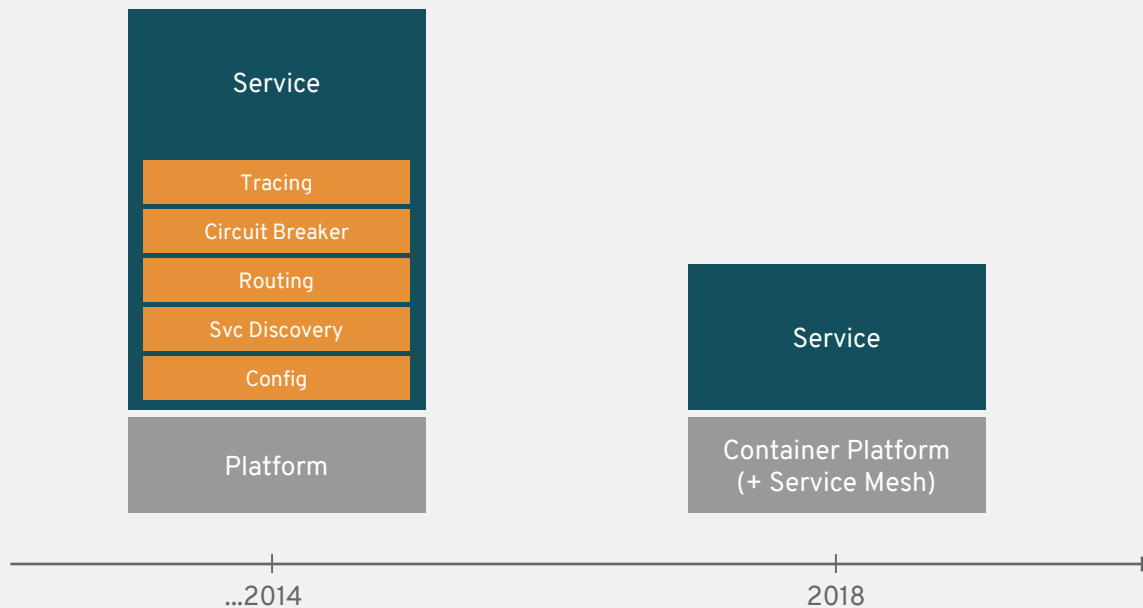
**THERE SHOULD BE A
BETTER WAY**

ADDRESS THE COMPLEXITY IN THE INFRASTRUCTURE

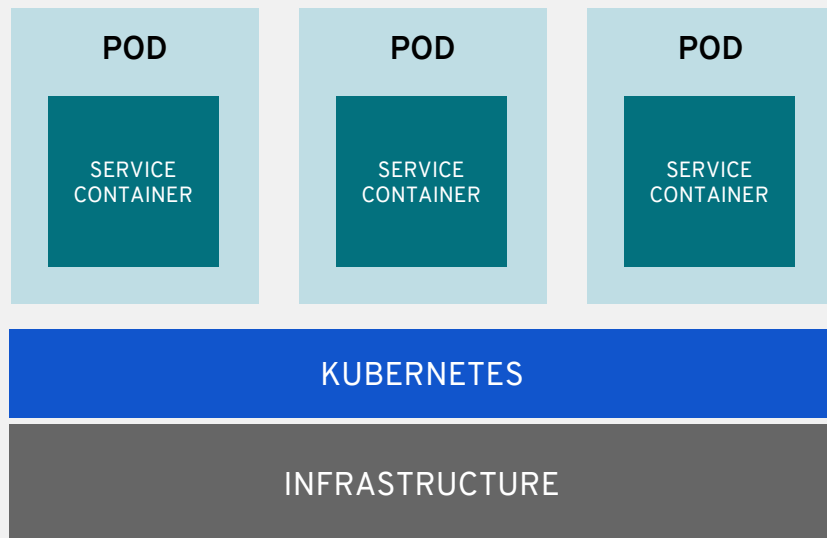
SERVICE MESH

A dedicated infrastructure layer for
service-to-service communications

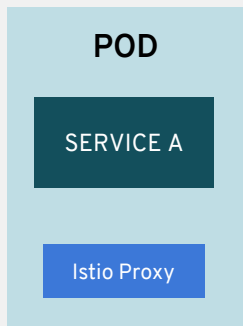
MICROSERVICES EVOLUTION



AUTOMATING CONTAINER DEPLOYMENT



SIDECARS

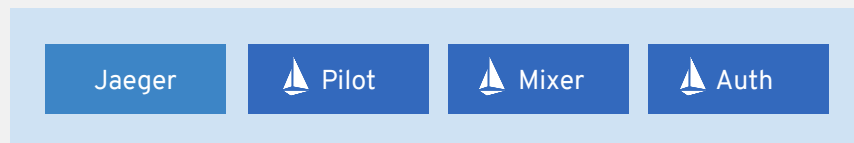


- Two or more containers deployed to same pod
- Share
 - Same
 - Namespace
 - Pod IP
 - Shared lifecycle
- Used to enhance the co-located containers
- Istio Proxy (L7 Proxy)
 - Proxy all network traffic in and out of the app container

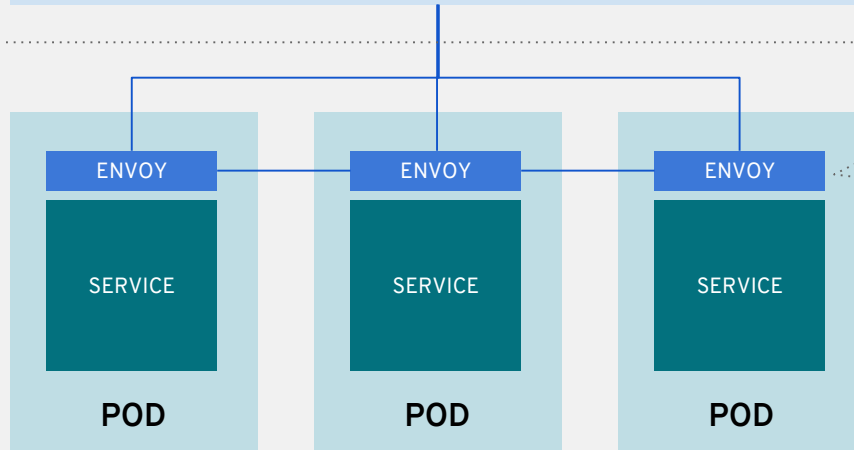
Source: <http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html>

SERVICE MESH ARCHITECTURE

Control Plane



Data Plane

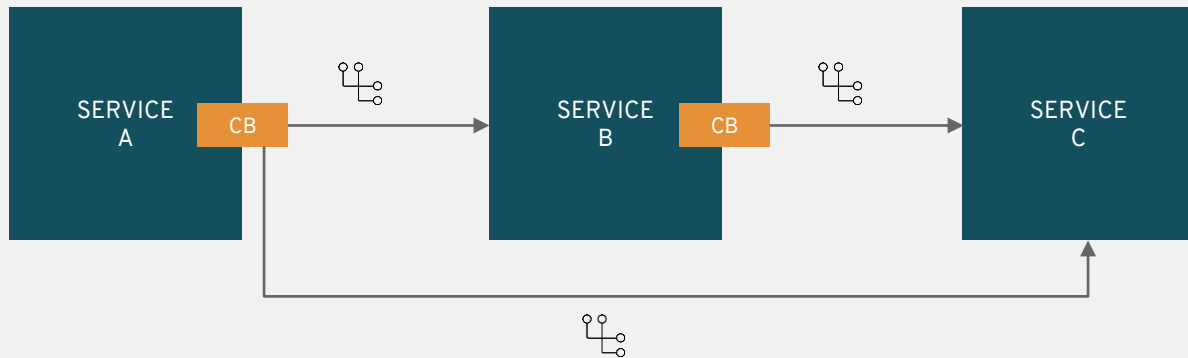


Applies security, route rules, policies and reports traffic telemetry at the pod level

MAJOR FUNCTIONALITY

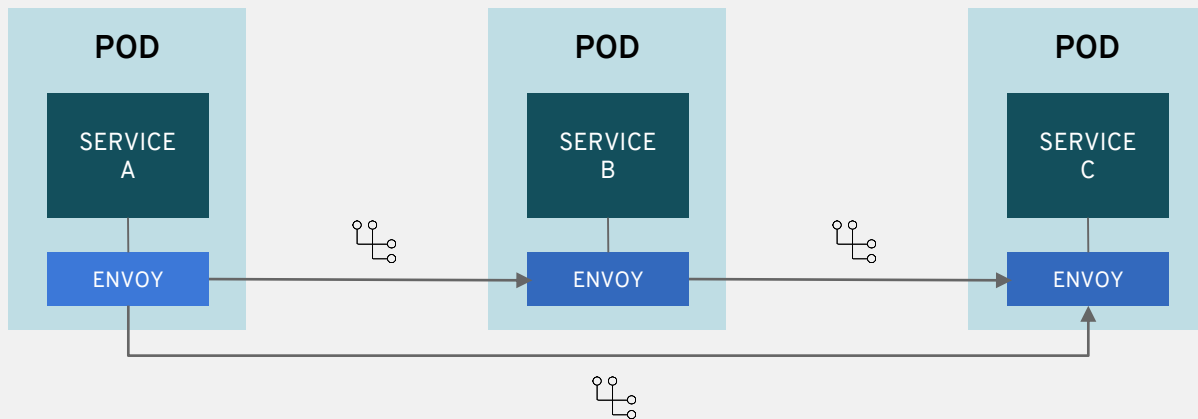
FAULT TOLERANCE

CIRCUIT BREAKERS **WITHOUT** ISTIO



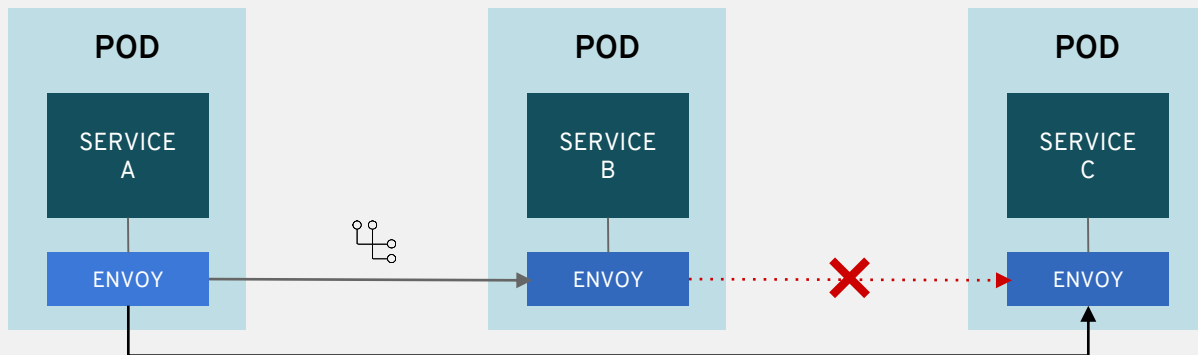
coupled to the service code

CIRCUIT BREAKERS WITH ISTIO



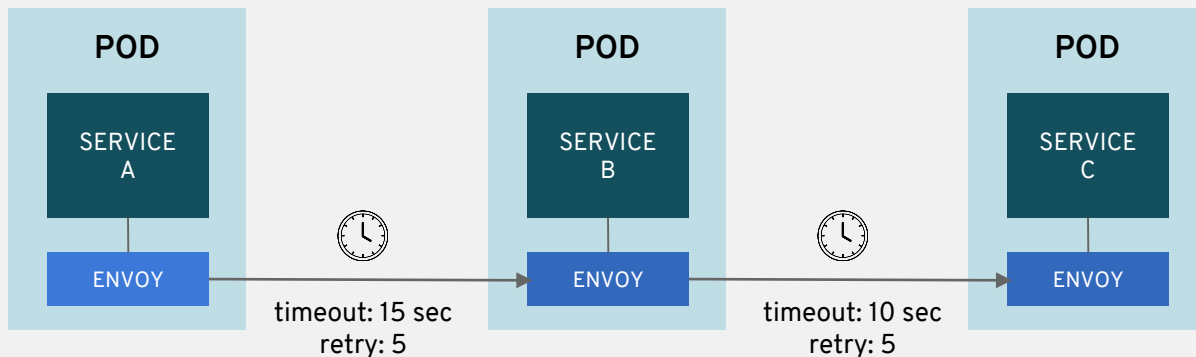
transparent to the services

CIRCUIT BREAKERS WITH ISTIO



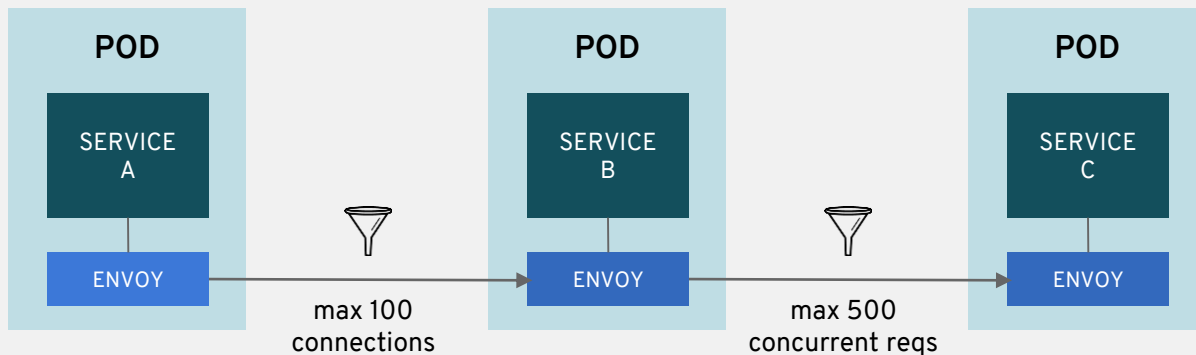
improved response time with global circuit status

TIMEOUTS AND RETRIES WITH ISTIO



configure timeouts and retries, transparent to the services

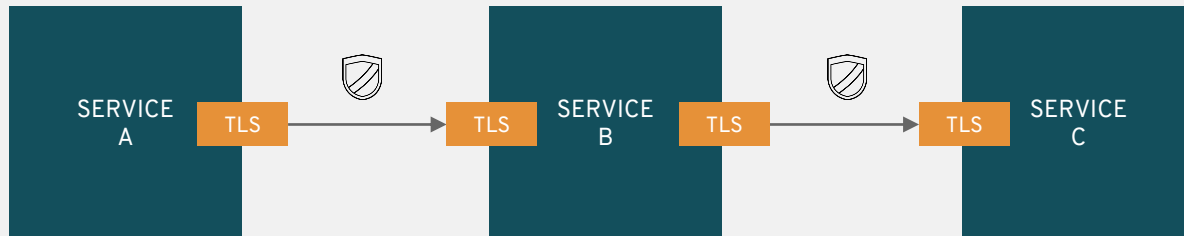
RATE LIMITING WITH ISTIO



limit invocation rates, transparent to the services

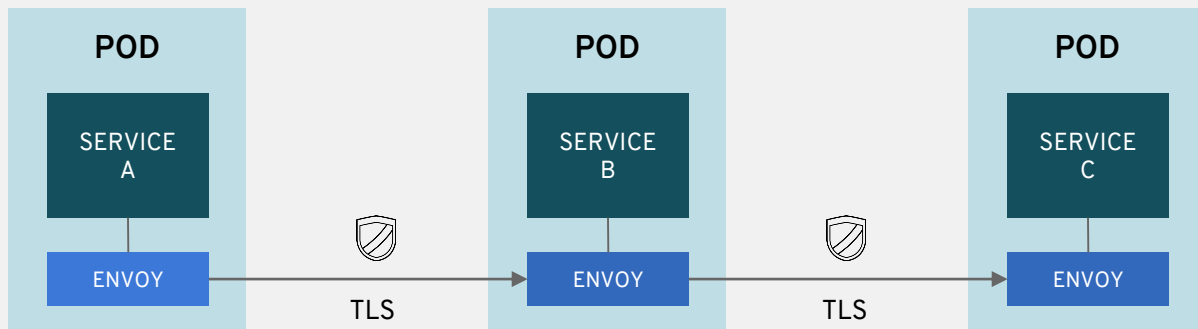
SERVICE SECURITY

SECURE COMMUNICATION **WITHOUT** ISTIO



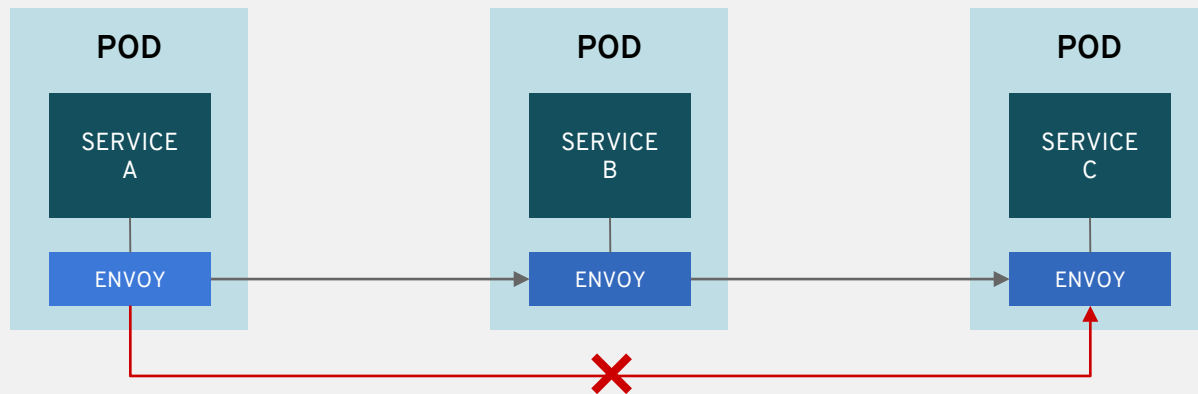
coupled to the service code

SECURE COMMUNICATION WITH ISTIO



mutual TLS authentication, transparent to the services

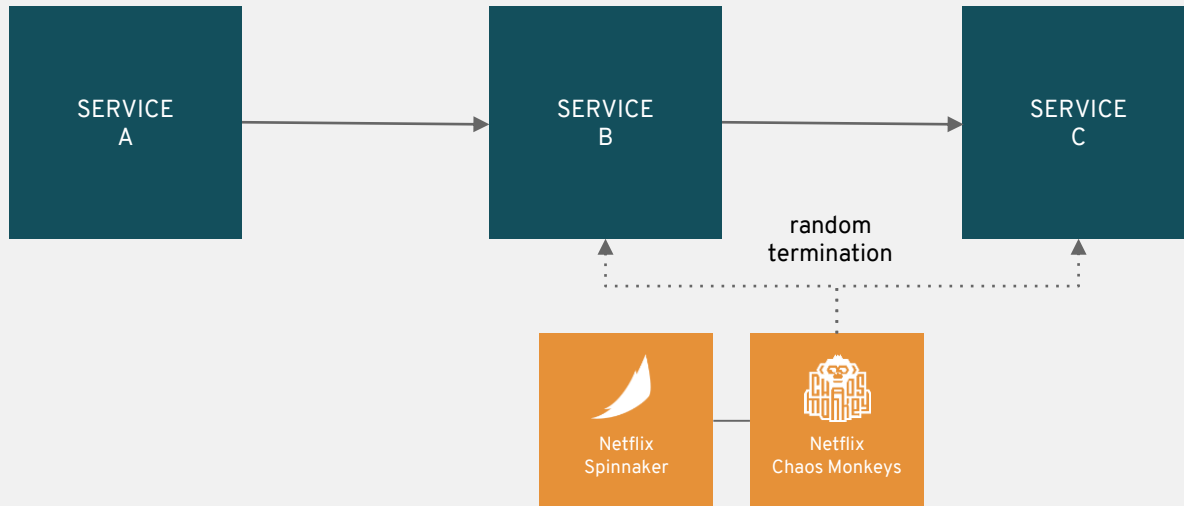
CONTROL SERVICE ACCESS WITH ISTIO



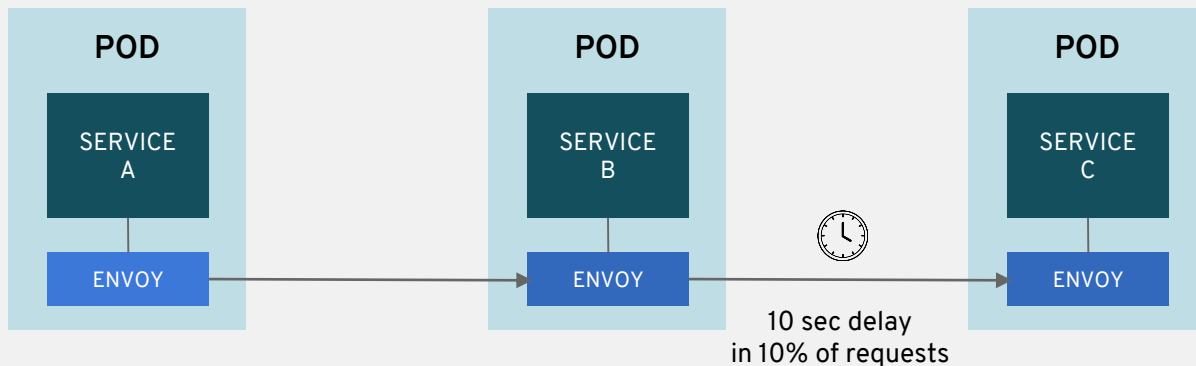
control the service access flow, transparent to the services

CHAOS ENGINEERING

CHAOS ENGINEERING WITHOUT ISTIO

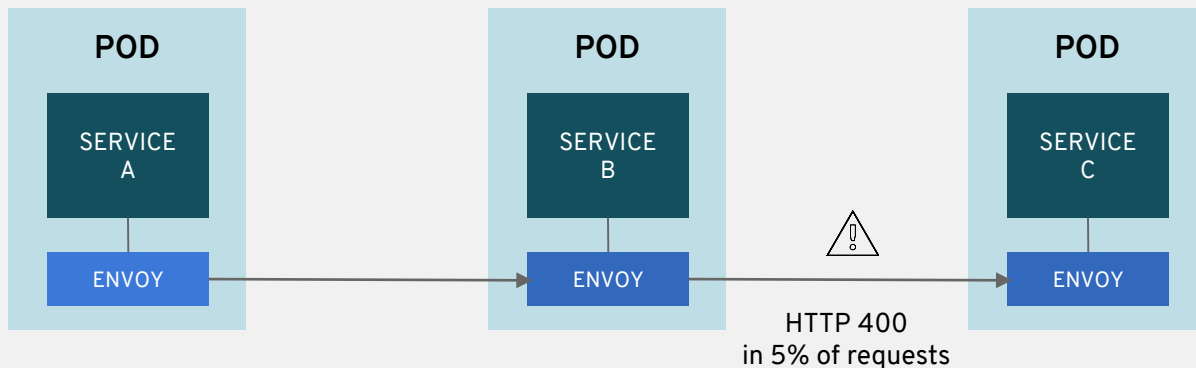


CHAOS ENGINEERING WITH ISTIO



inject delays, transparent to the services

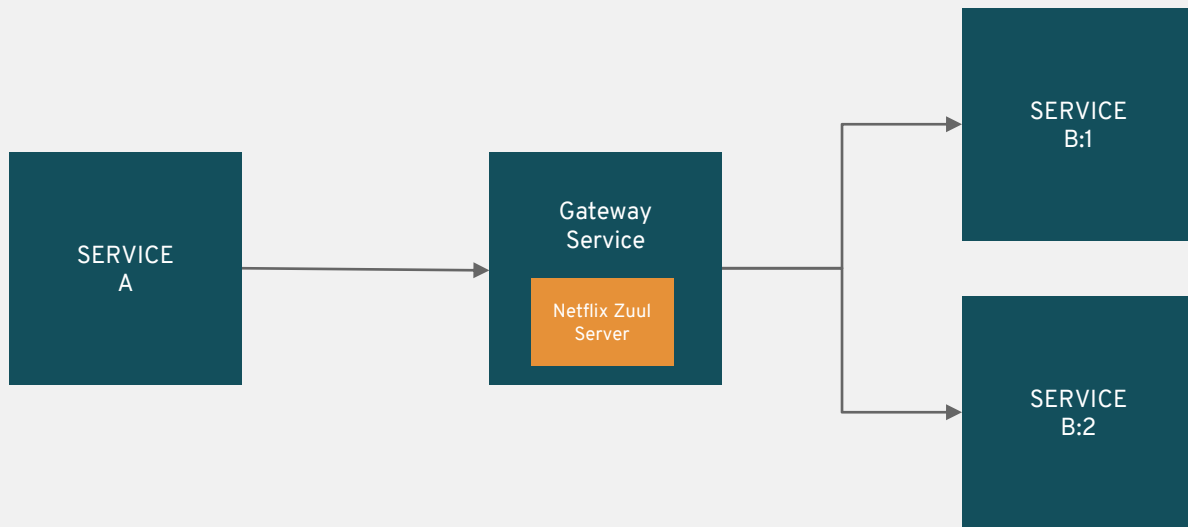
CHAOS ENGINEERING WITH ISTIO



inject protocol-specific errors, transparent to the services

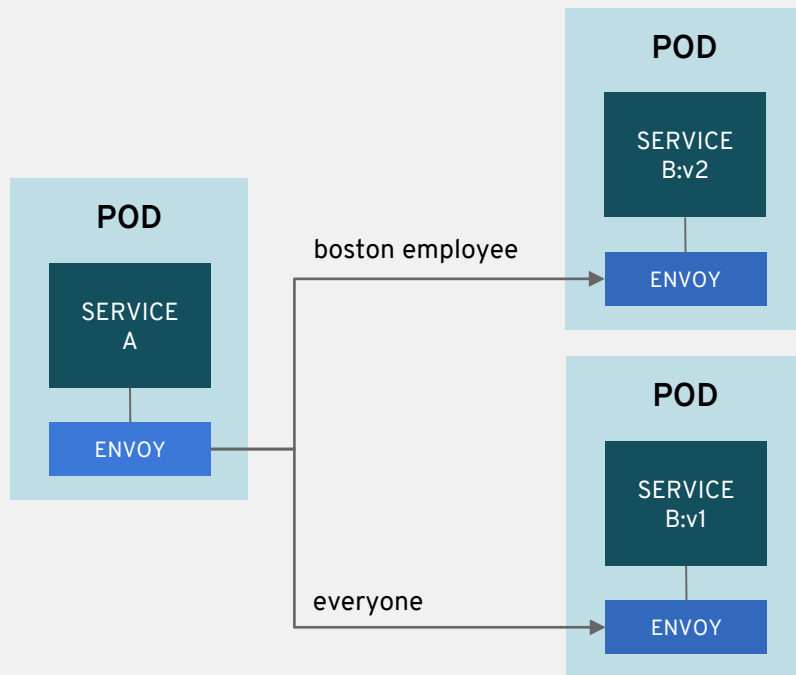
DYNAMIC ROUTING

DYNAMIC ROUTING **WITHOUT** ISTIO

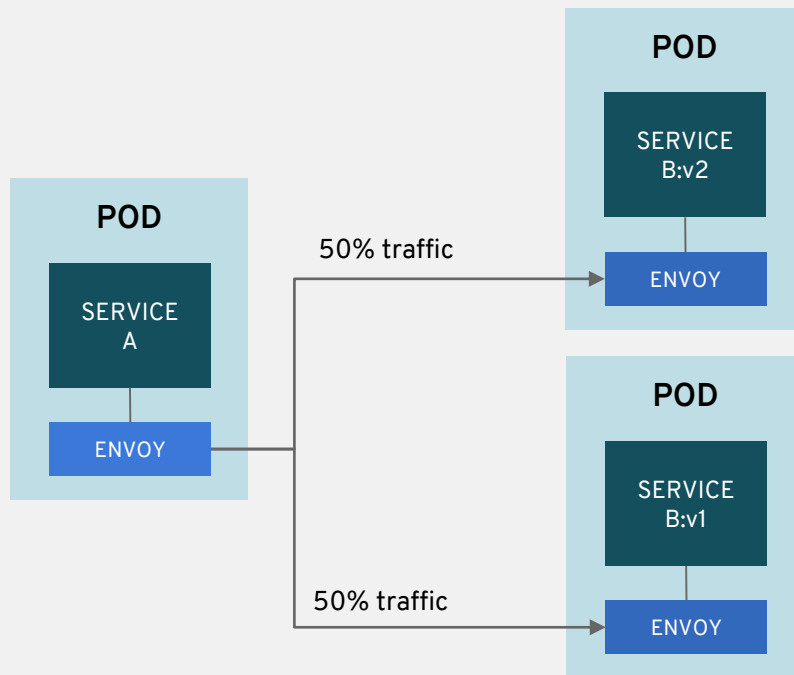


custom code to enable dynamic routing

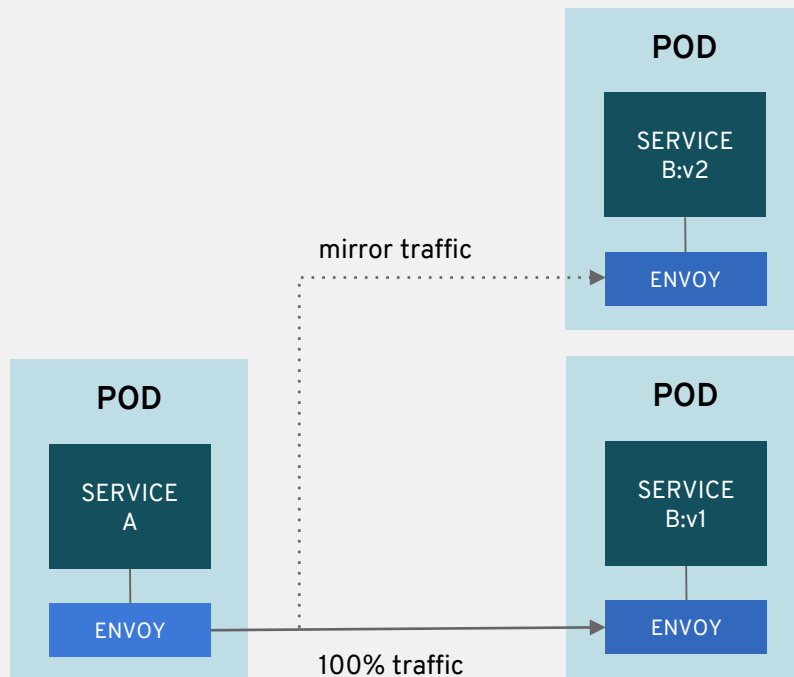
CANARY DEPLOYMENT WITH ISTIO



A/B DEPLOYMENT WITH ISTIO

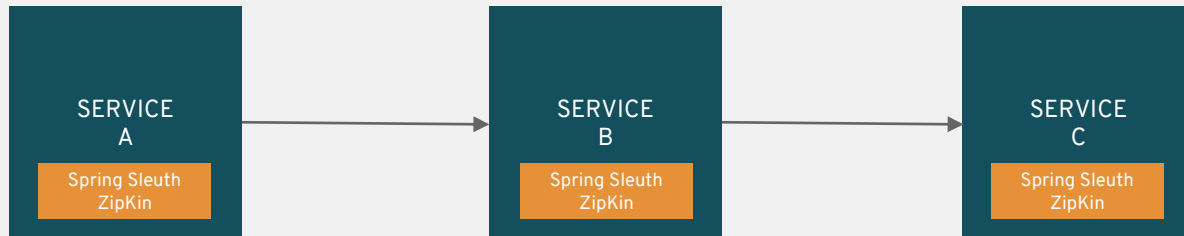


DARK LAUNCHES WITH ISTIO



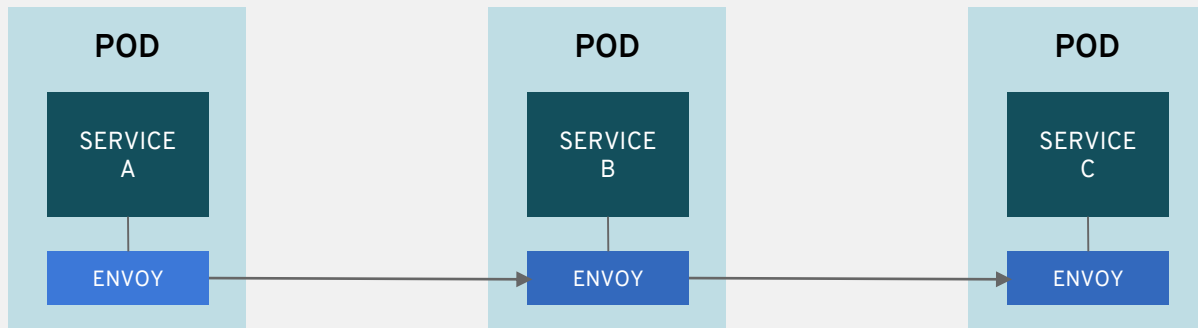
DISTRIBUTED TRACING (JAEGER)

DISTRIBUTED TRACING **WITHOUT** ISTIO

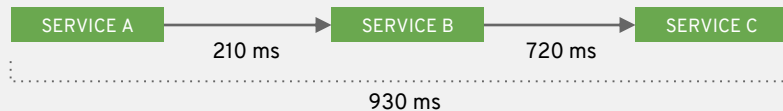


code to enable dynamic tracing

DISTRIBUTED TRACING WITH ISTIO & JAEGER



discovers service relationships and process times,
transparent to the services



SERVICE MESH OBSERVABILITY (KIALI)

- Overview
- Graph**
- Applications
- Workloads
- Services
- Istio Config
- Distributed Trac...

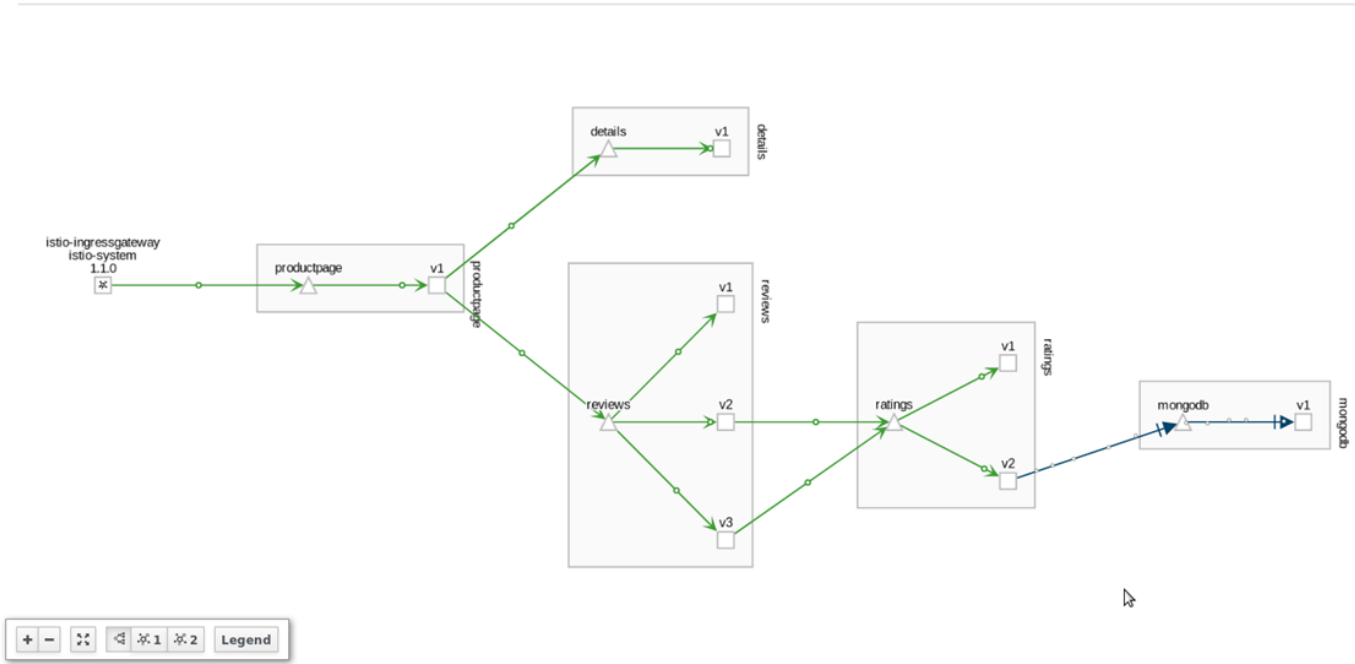
Namespace: **bookinfo**

Graph

Feb 18, 16:07:37 ... Feb 18, 16:08:37

Display Edge Labels Graph Type Versioned app Find... Hide...

Fetching Last min Every 15 sec



Namespace: bookinfo
[applications](#), [services](#), [workloads](#)

Current Graph:

- 9 apps
- 5 services
- 14 edges

HTTP Traffic (requests per second):

Total	%Success	%Error
3.68	100.00	0.00

0255075100

OK3xx4xx5xx

HTTP - Total Request Traffic min / max:
RPS: 3.60 / 3.60 , %Error 0.00 / 0.00

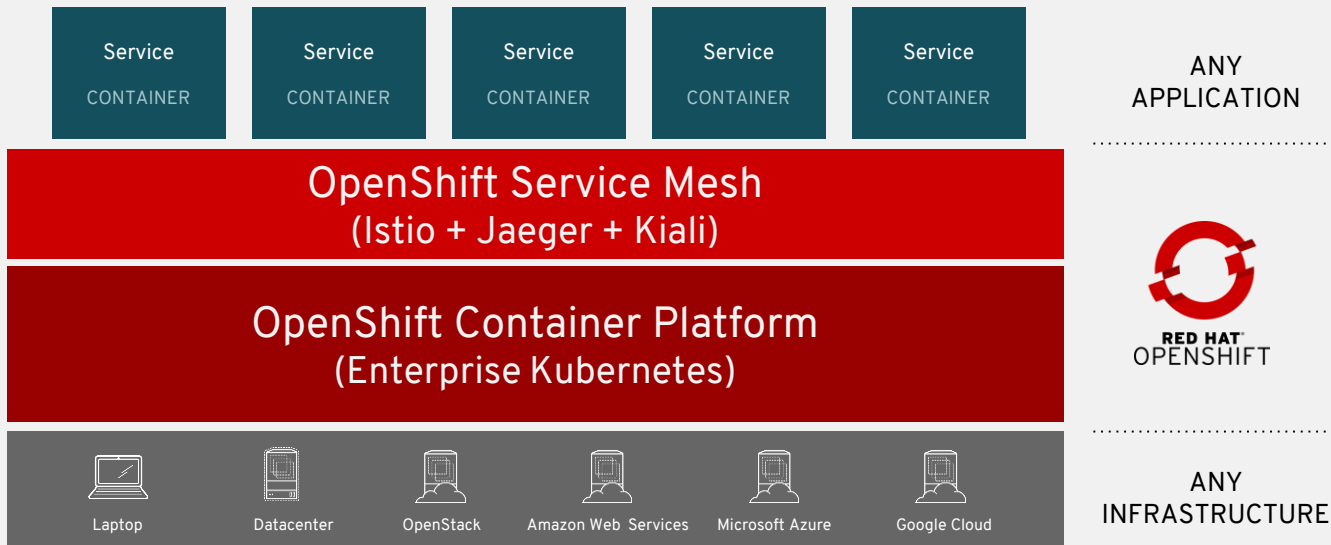
TCP - Total Traffic - min / max:

Sent: 143.00 / 143.00 B/s

Received: 115.67 / 115.67 B/s

TAKEWAY

DISTRIBUTED SERVICES PLATFORM





THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHat



youtube.com/user/RedHatVideos