Red Hat OpenShift 4

Next Level Cloud Development with Service Mesh

HVOpen January 2020

Patrick Ladd – pladd@redhat.com
Technical Account Manager
Red Hat

Red Hat

QUICK TIP

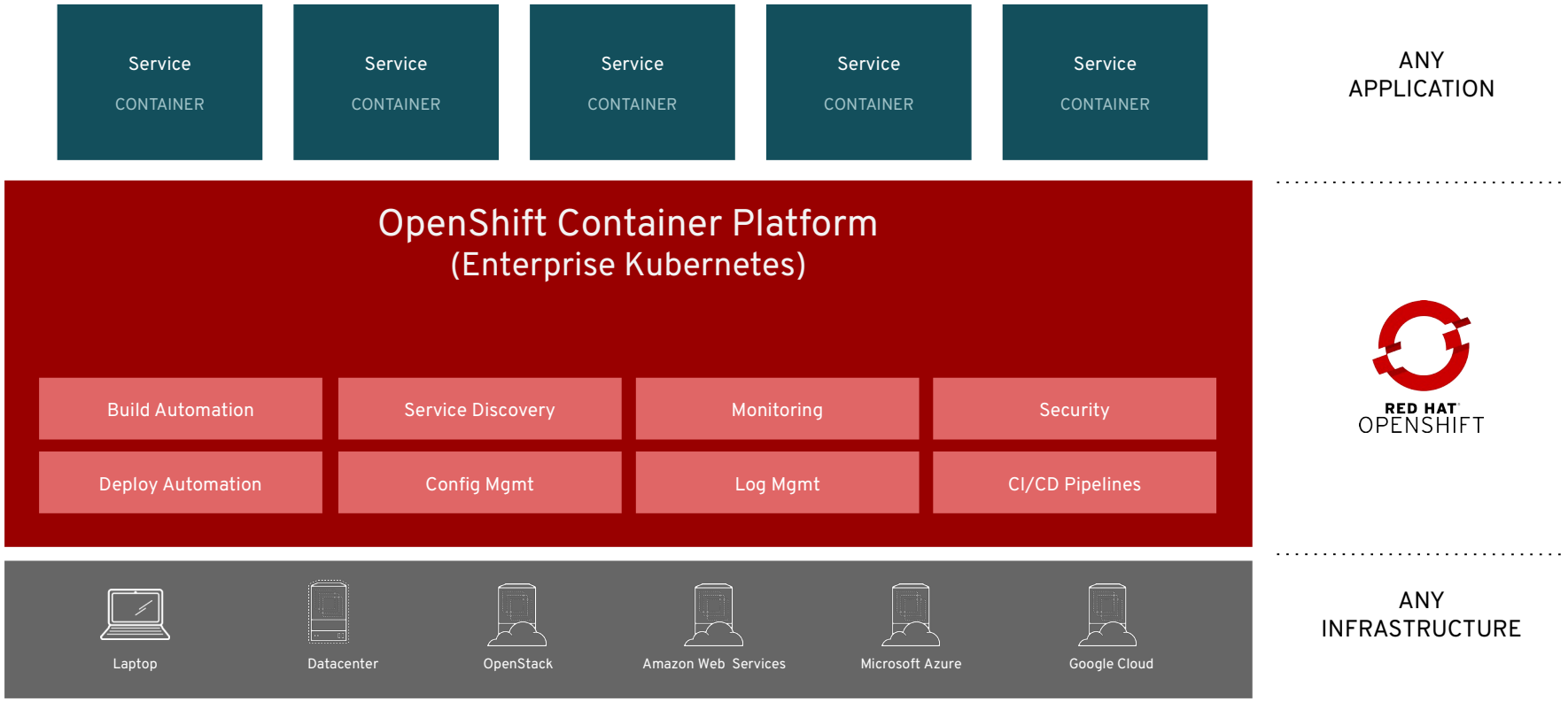
Try right clicking on the photo and using "Replace" to insert your own photo. You are also welcome to use this photo.

A mash-up of several better-known technologies: "A service mesh is a set of software components which act as the "glue" for a set of independent applications. The goal of the mesh is to guarantee secure communications between each application and be able to redirect traffic in the event of failures. Often the features of a service mesh look like a mash-up between a load balancer, a web application firewall, and an API gateway."

- Brian "Redbeard" Harrington, Product Manager at Red Hat

Red Hat

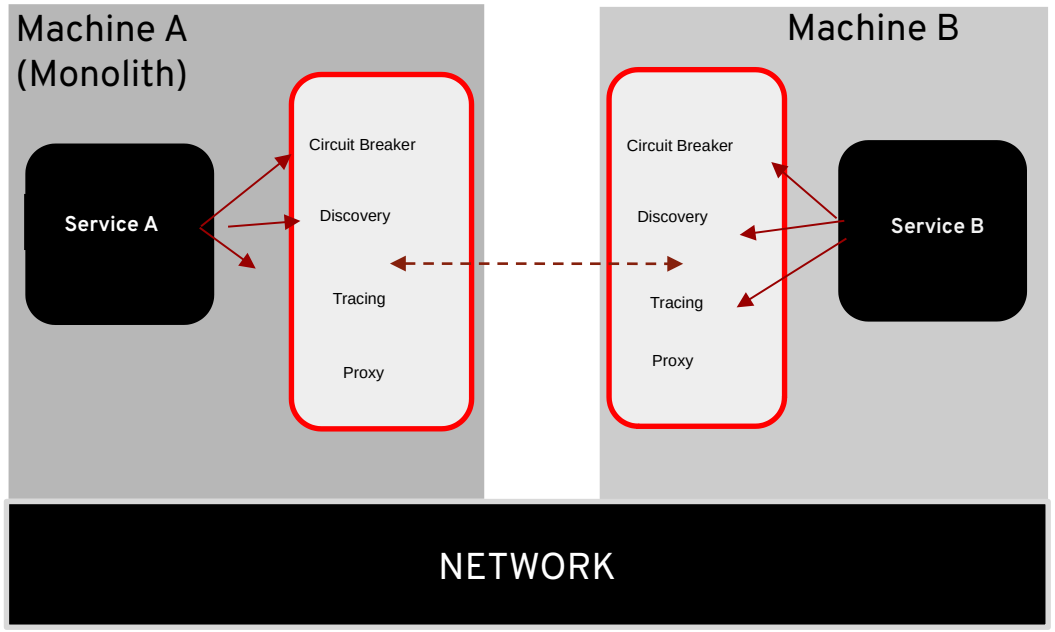
BUILD AND DEPLOY CLOUD-NATIVE APPS WITH RED HAT OPENSHIFT



Red Hat

OVERVIEW

WHAT IS A SERVICE MESH ?



Red Hat

SERVICE MESH ECOSYSTEM

Jaeger

Secure

Istio

Observe

Connect

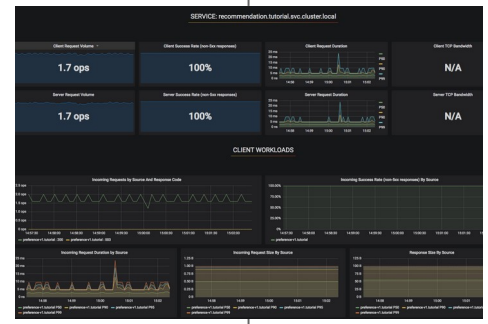
Control

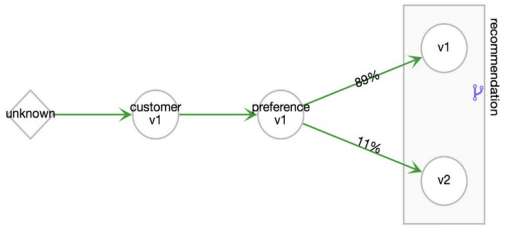
kiali

Kiali

Prometheus

Observe

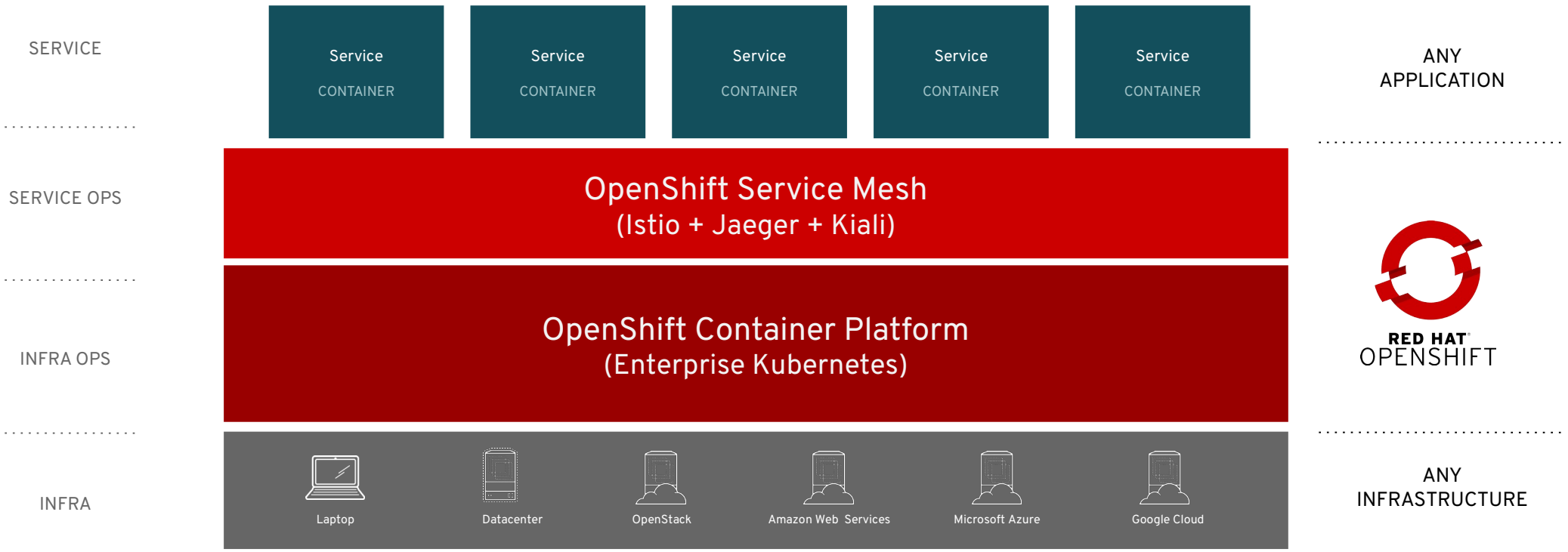




Grafana

Red Hat

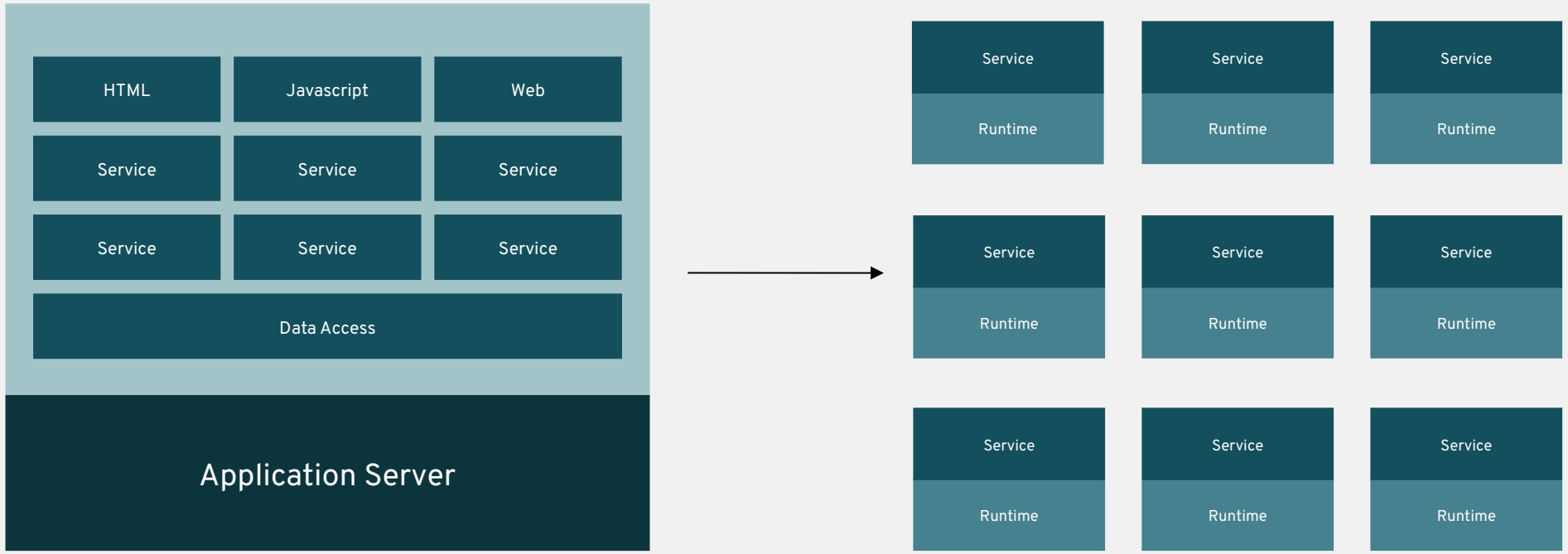
DISTRIBUTED SERVICES WITH RED HAT OPENSHIFT SERVICE MESH



Red Hat

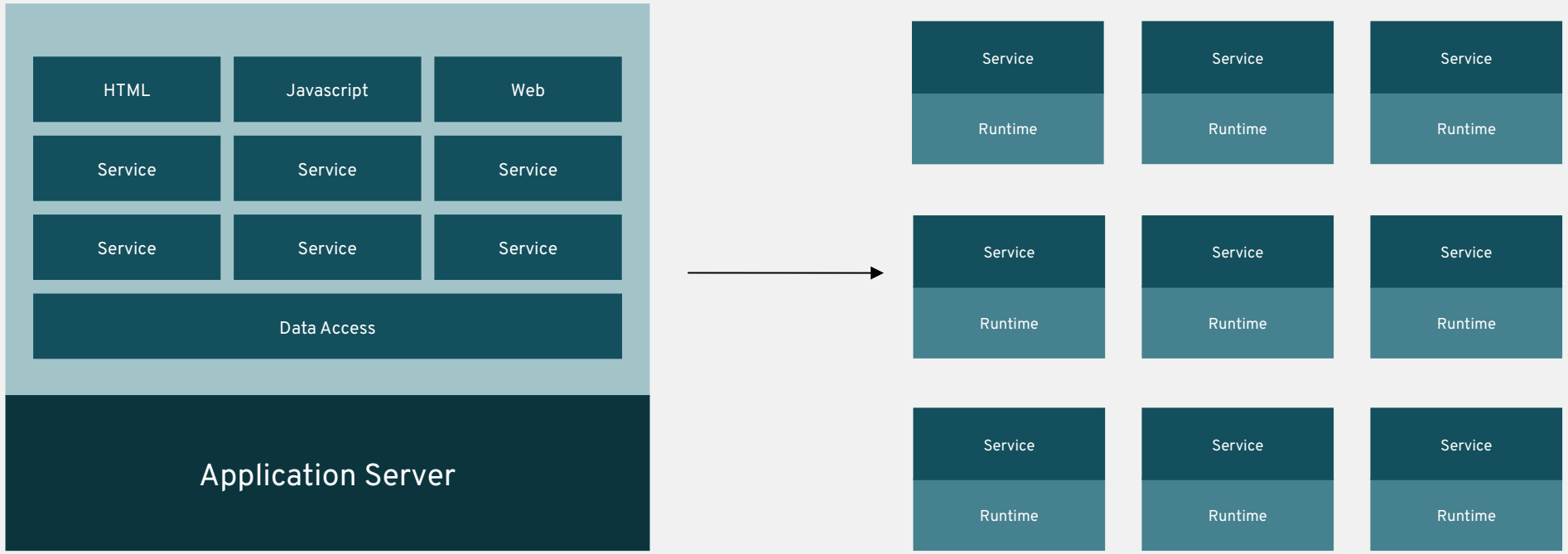
UNDER THE HOOD

MICROSERVICES ARCHITECTURE



~~MICROSERVICES~~ ARCHITECTURE

DISTRIBUTED



DISTRIBUTED ARCHITECTURE

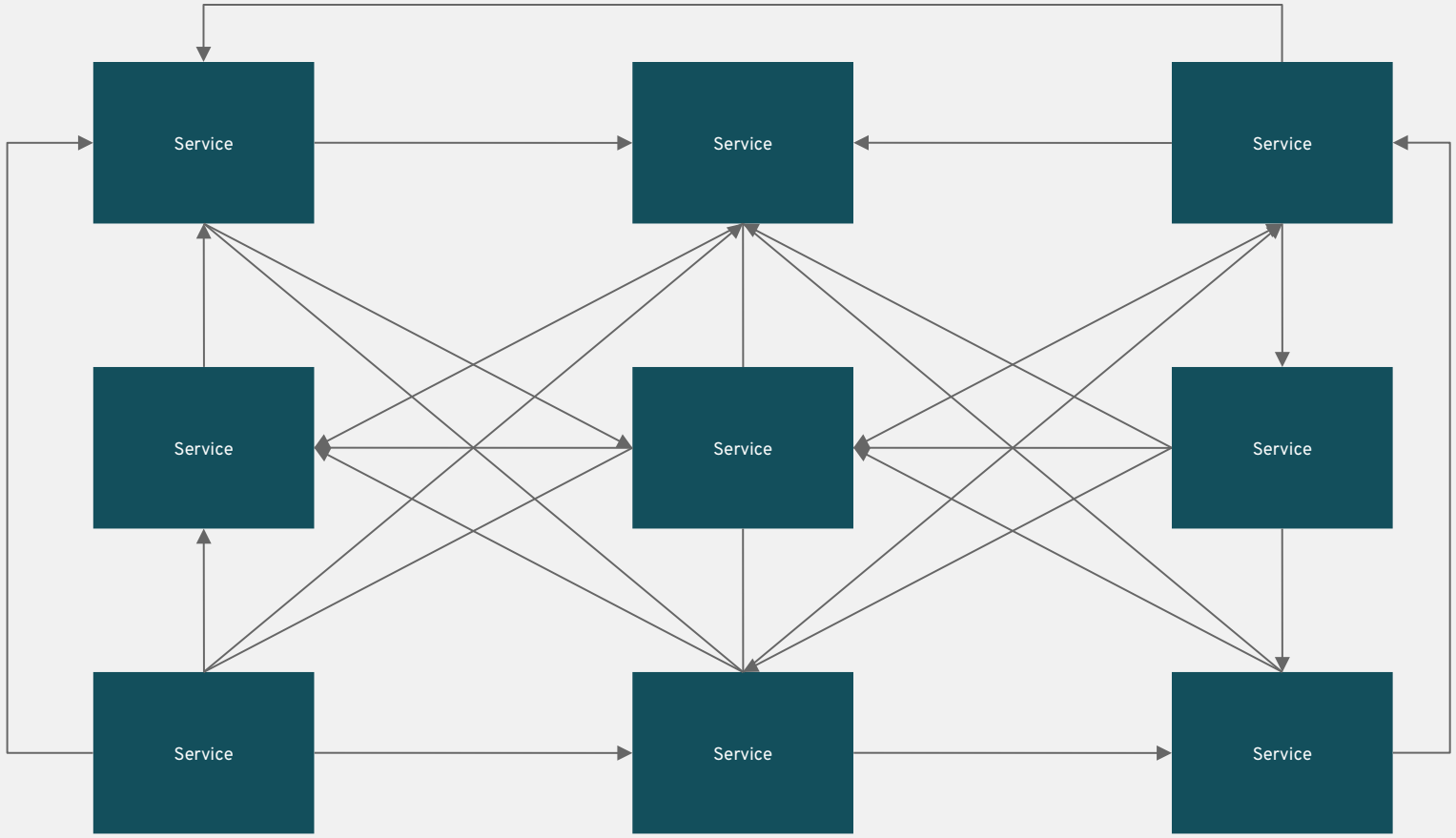
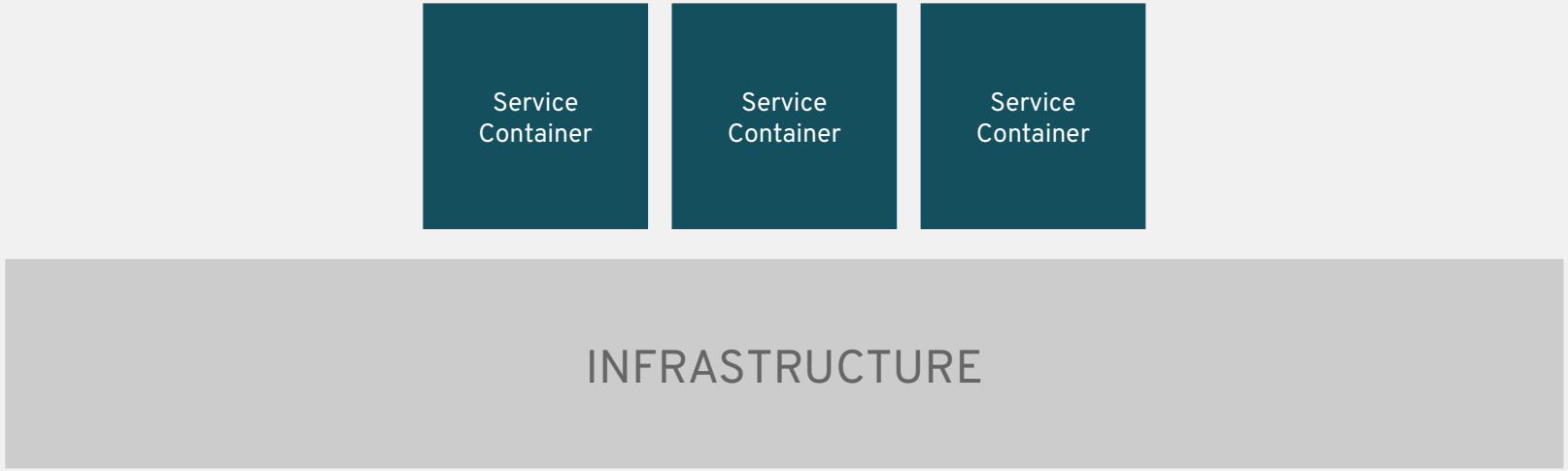
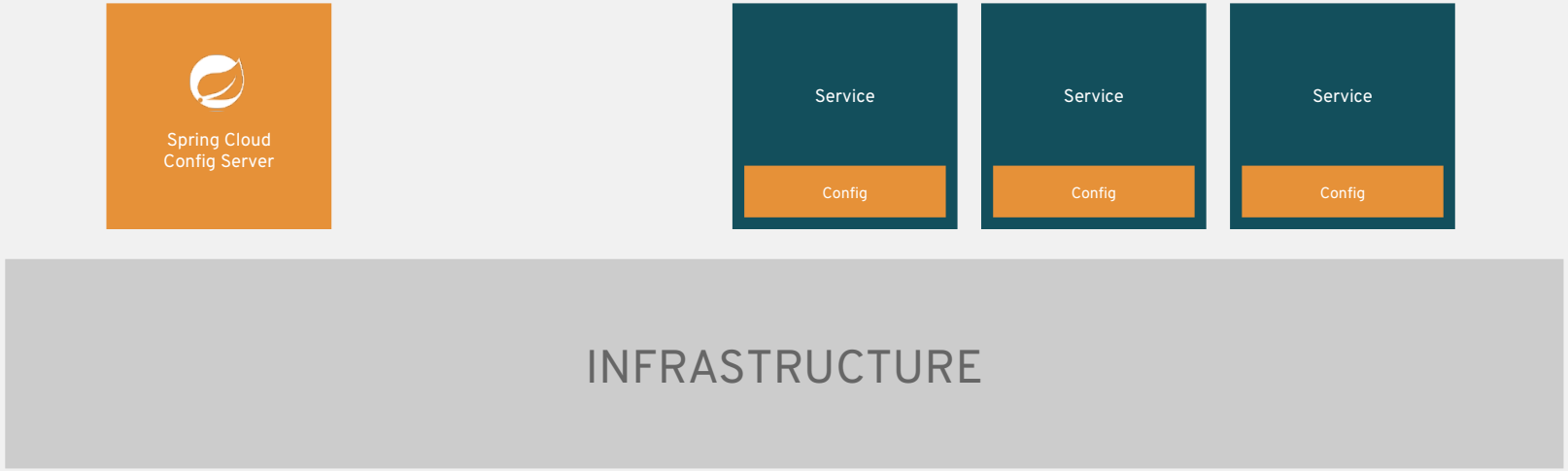


Photo by Clint Adair on Unsplash

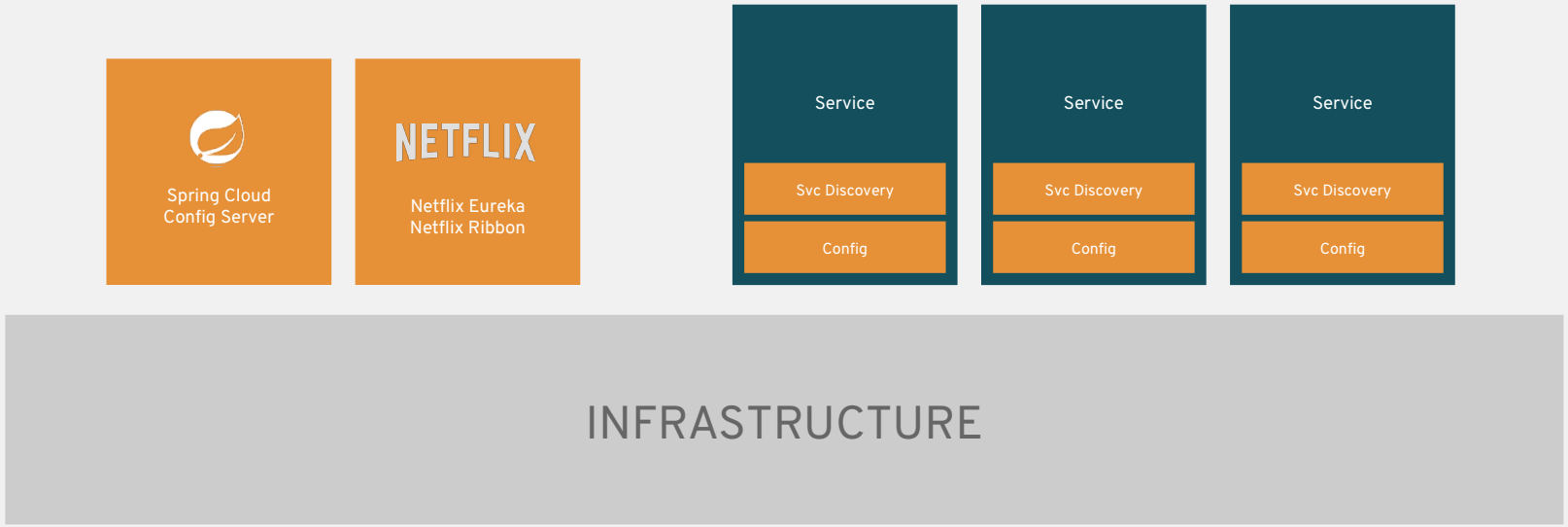
DEPLOYMENT


Service Container
Service Container
Service Container
INFRASTRUCTURE

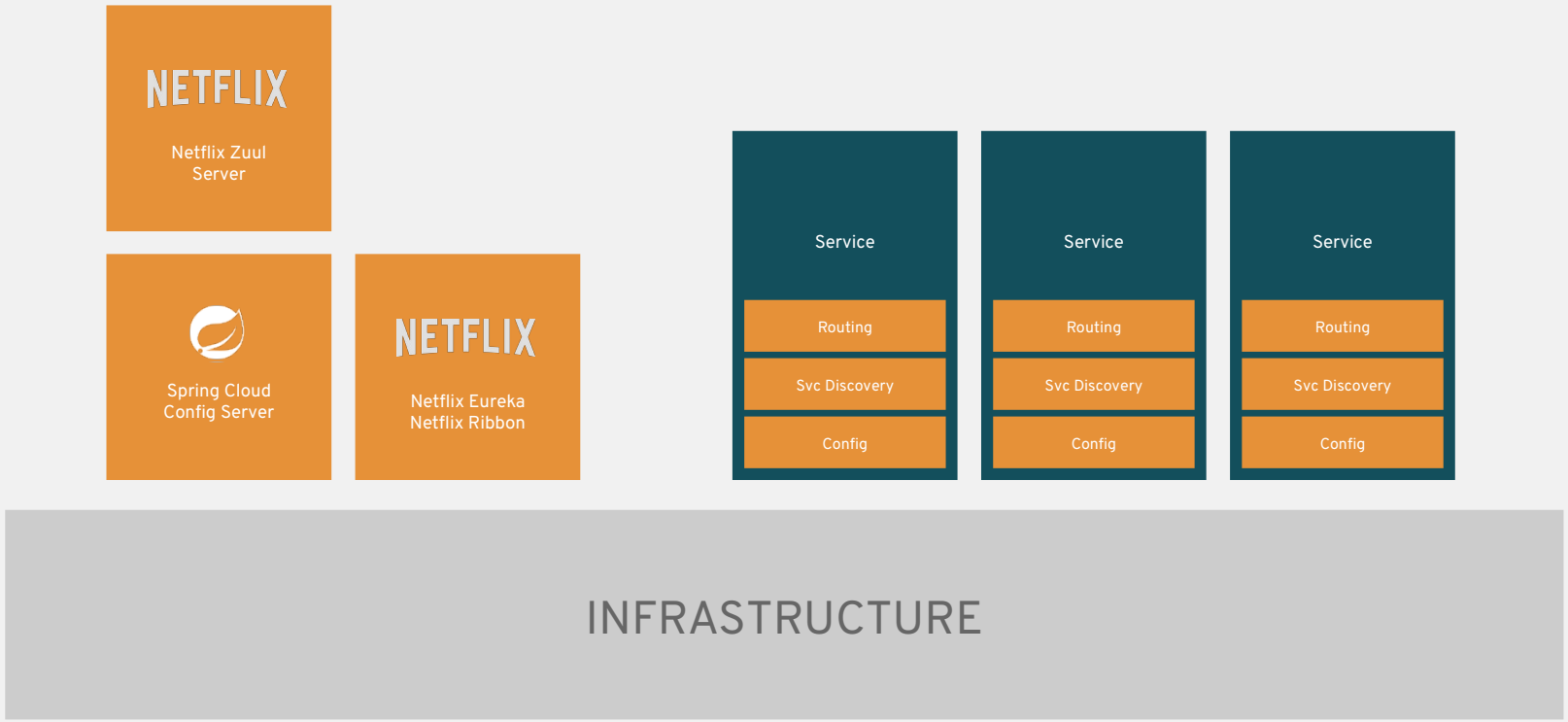
CONFIGURATION



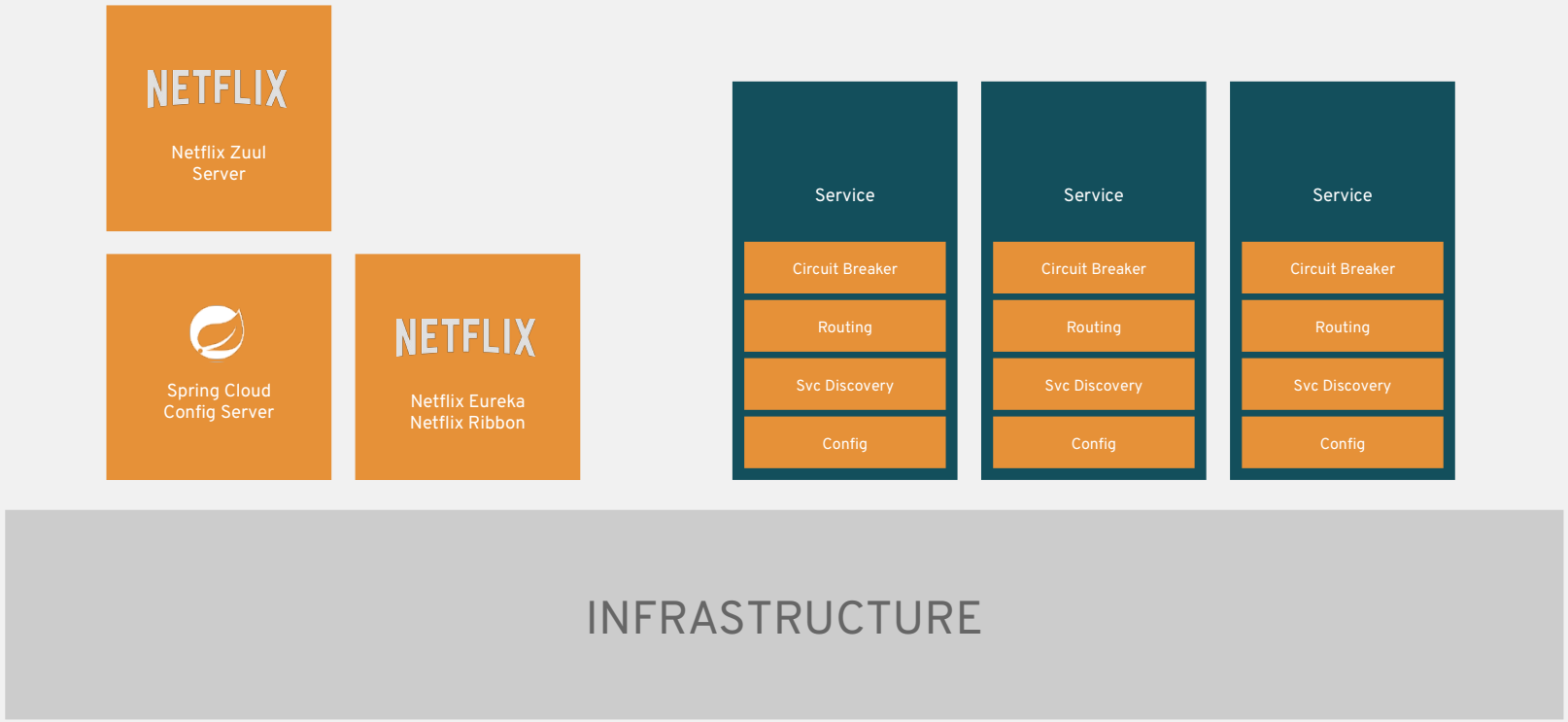
SERVICE DISCOVERY



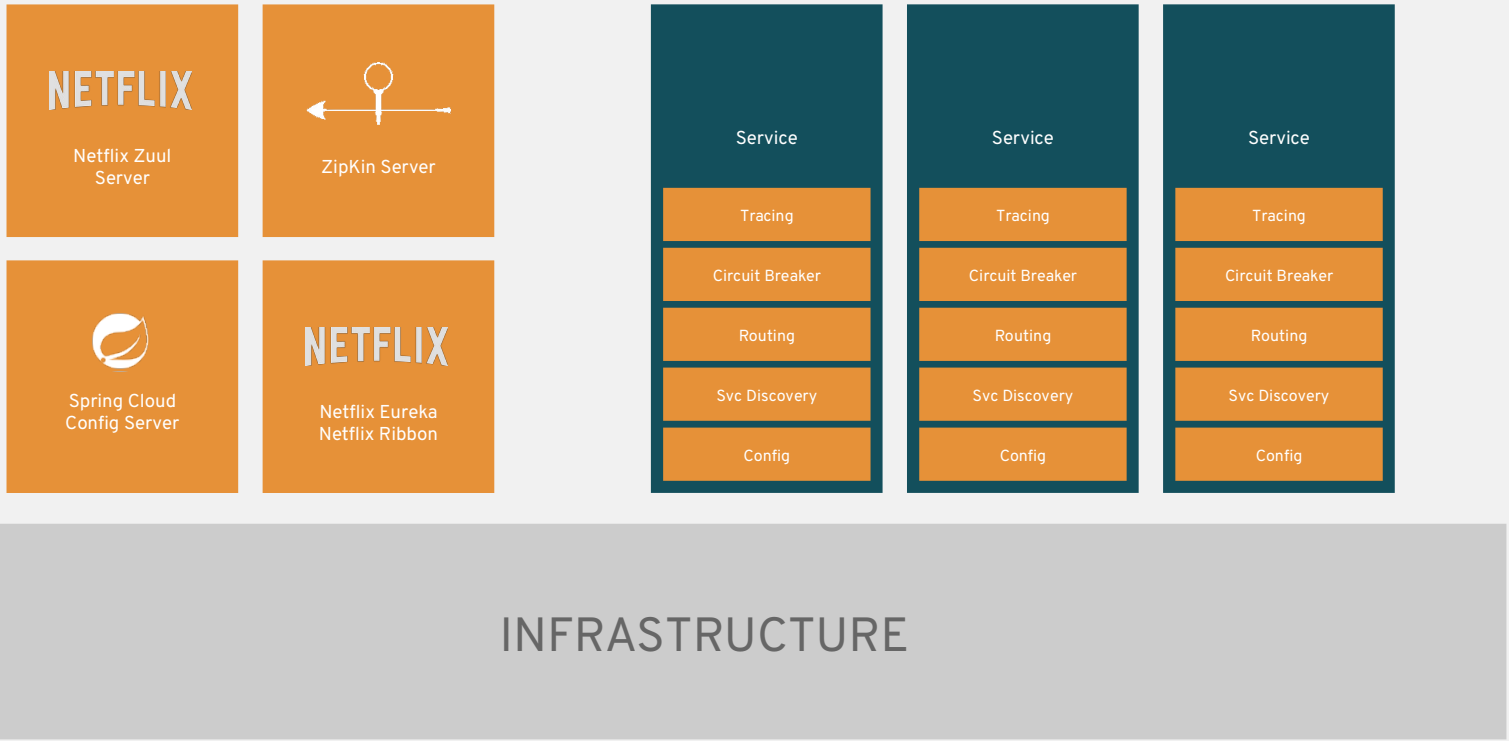
DYNAMIC ROUTING



FAULT TOLERANCE



TRACING AND VISIBILITY



WHAT ABOUT...?

POLYGLOT APPS

EXISTING APPS

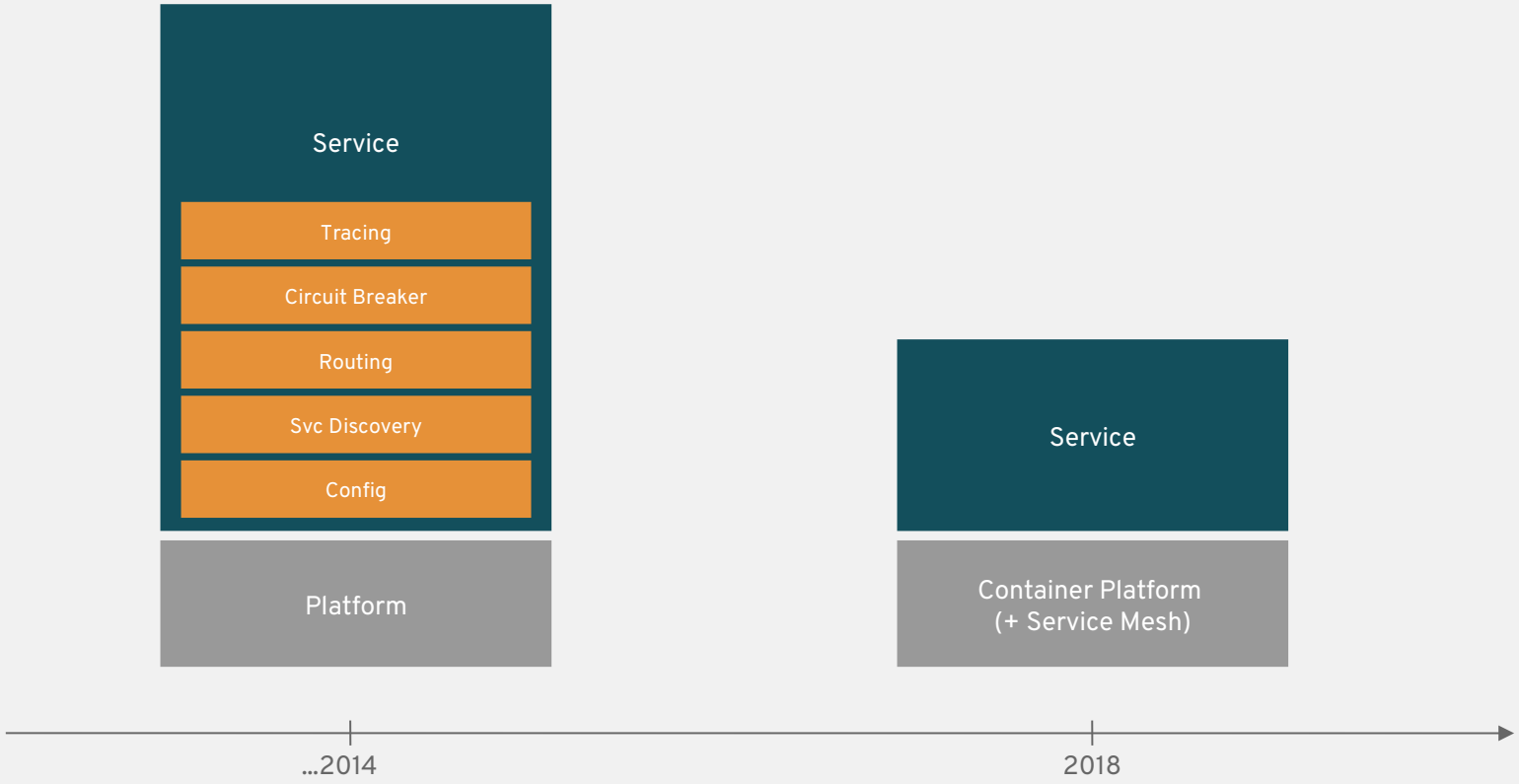
THERE SHOULD BE A BETTER WAY

ADDRESS THE COMPLEXITY IN THE INFRASTRUCTURE

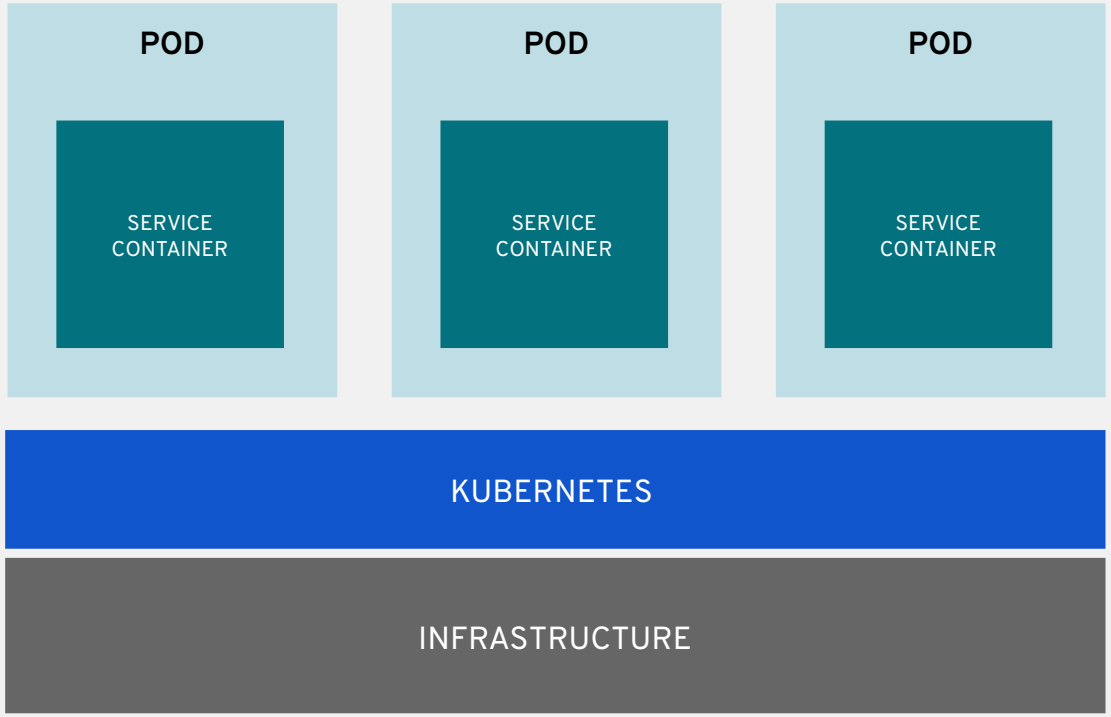
SERVICE MESH

A dedicated infrastructure layer for service-to-service communications

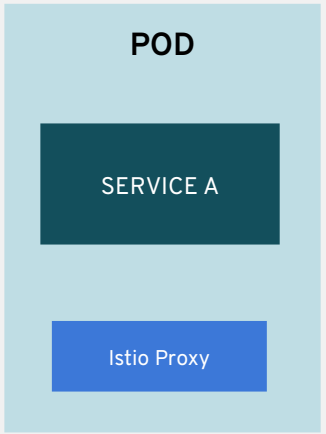
MICROSERVICES EVOLUTION



AUTOMATING CONTAINER DEPLOYMENT



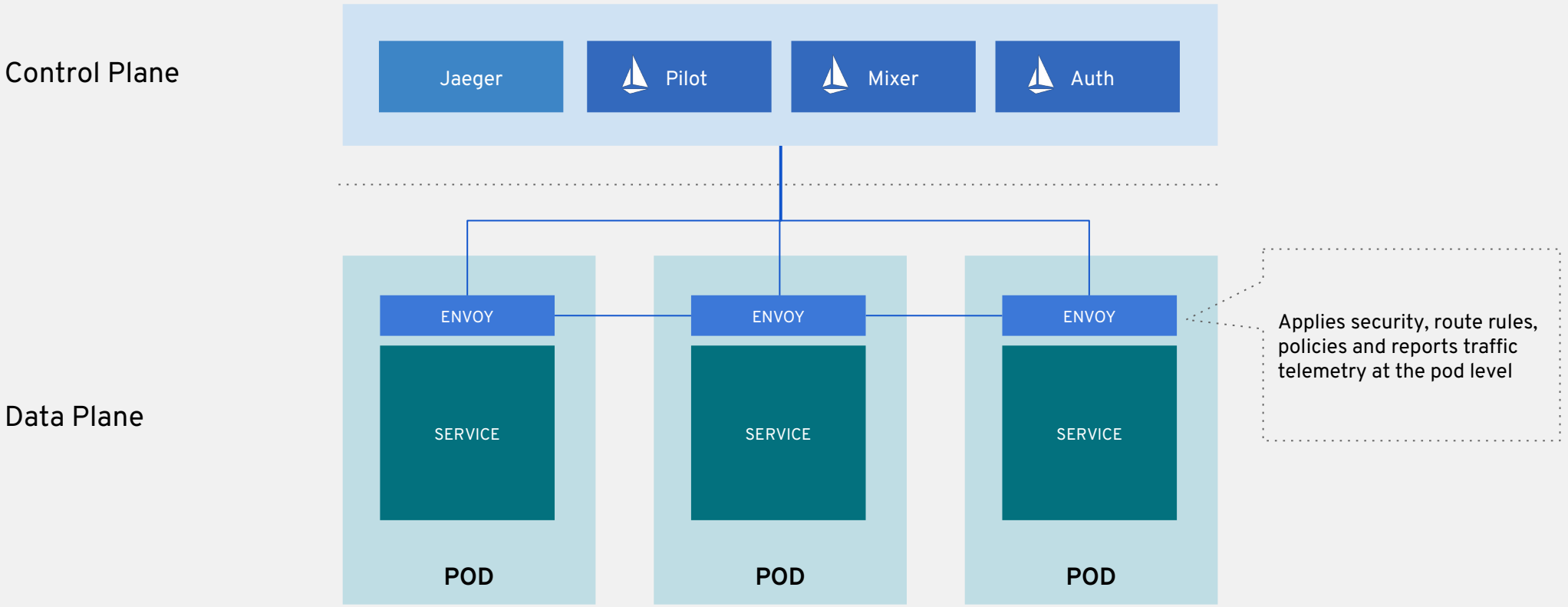
SIDECARS



- 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

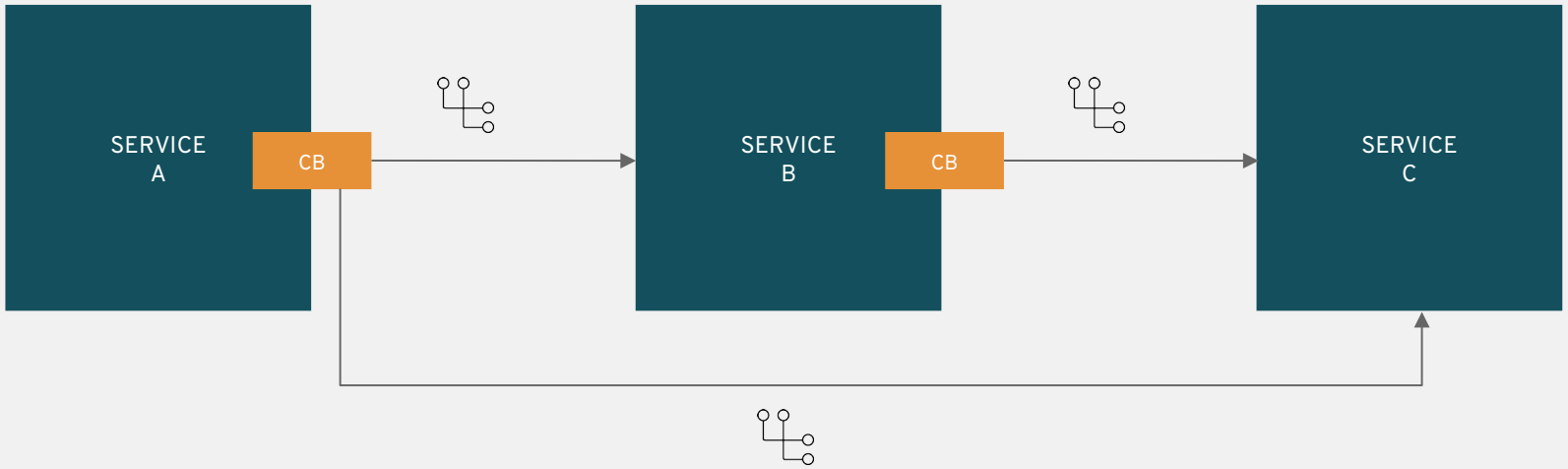
SERVICE MESH ARCHITECTURE



MAJOR FUNCTIONALITY

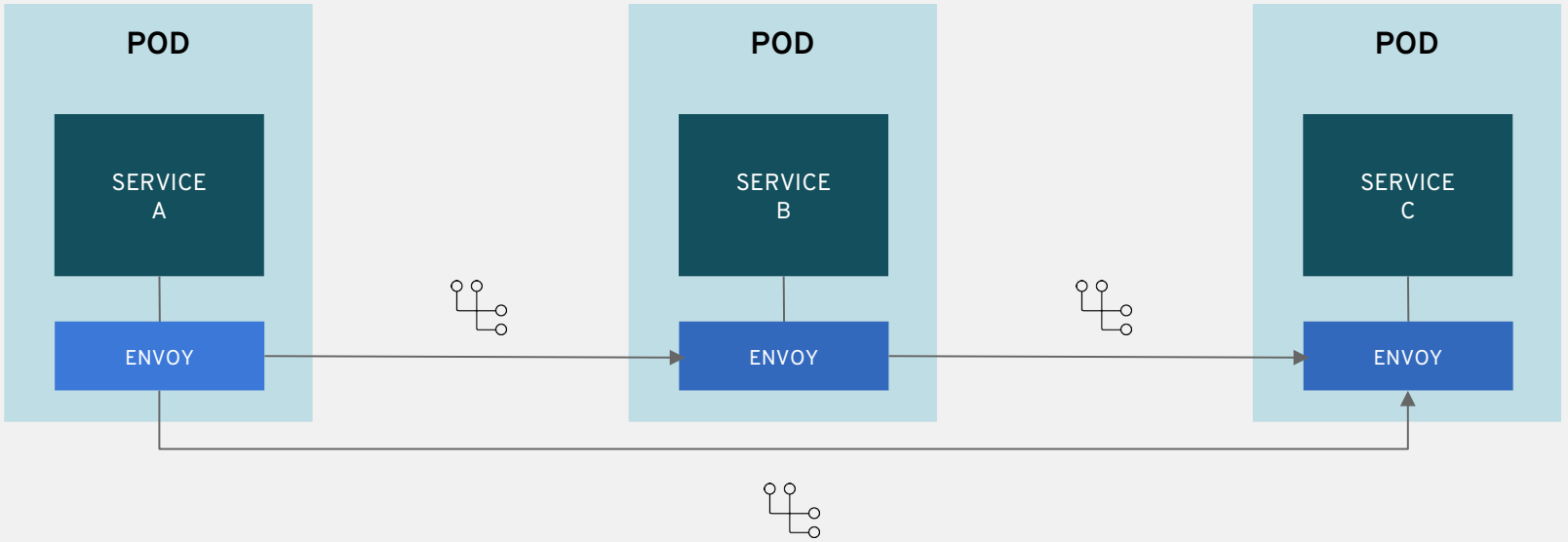
FAULT TOLERANCE

CIRCUIT BREAKERS WITHOUT ISTIO



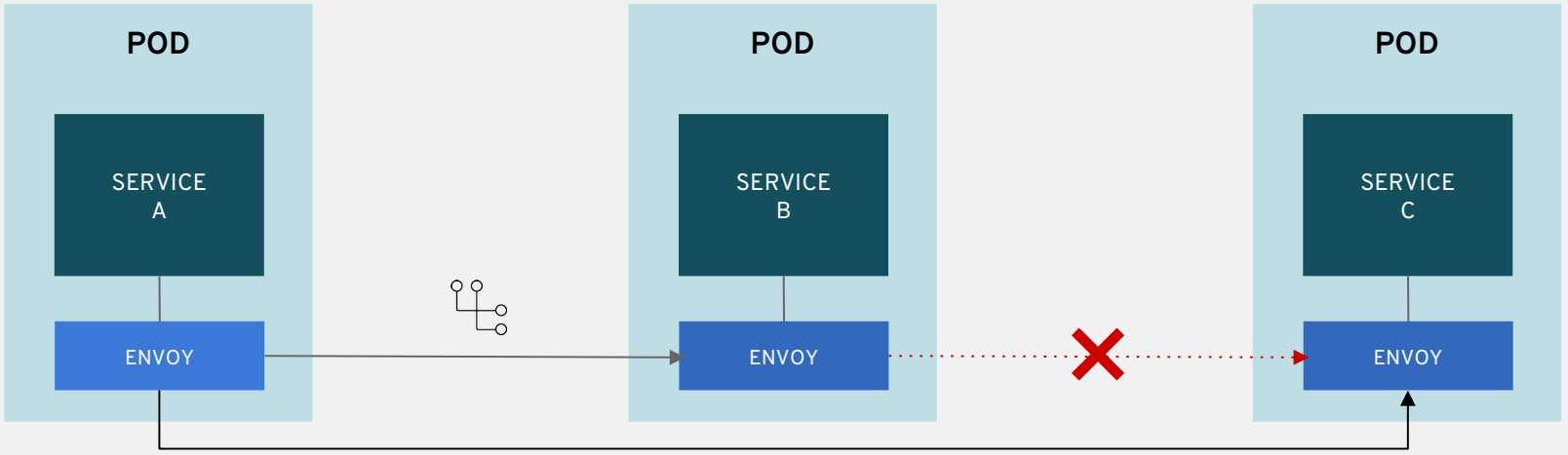
coupled to the service code

CIRCUIT BREAKERS WITH ISTIO



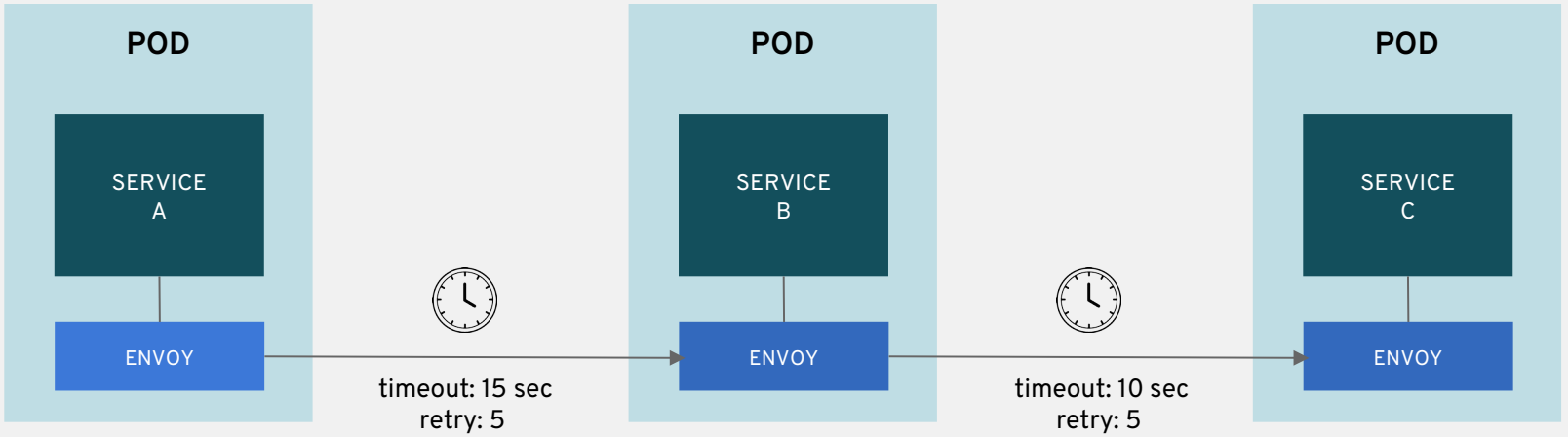
transparent to the services

CIRCUIT BREAKERS WITH ISTIO



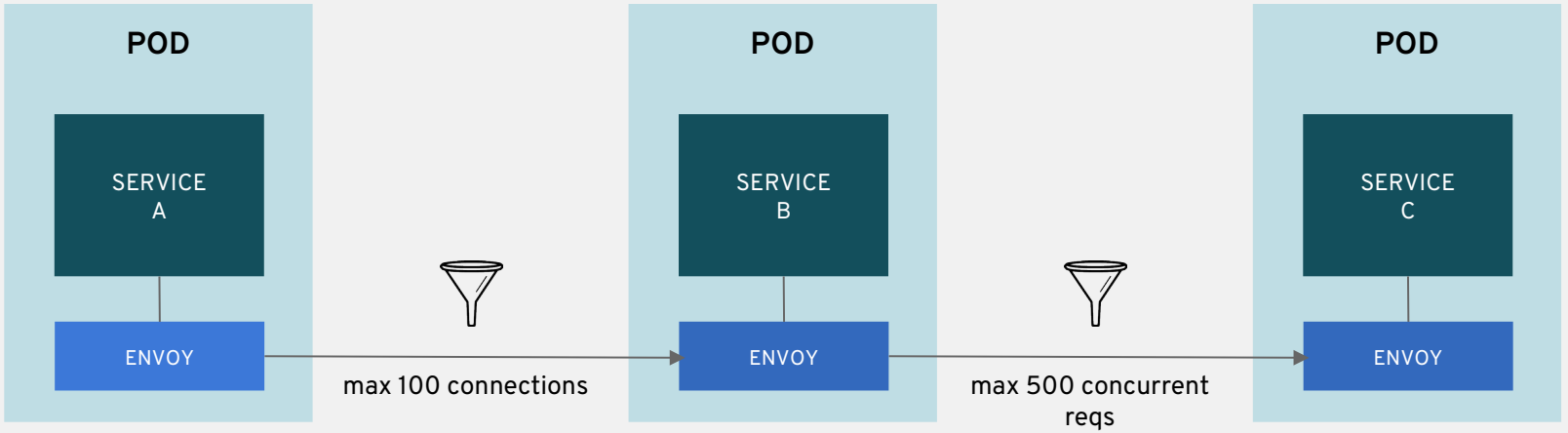
improved response time with global circuit status

TIMEOUTS AND RETRIES WITH ISTIO



configure timeouts and retries, transparent to the services

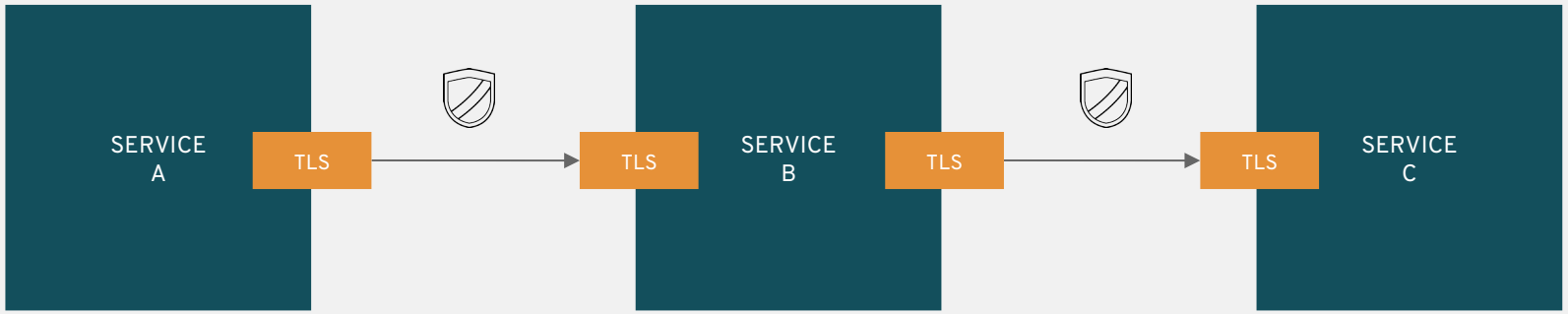
RATE LIMITING WITH ISTIO



limit invocation rates, transparent to the services

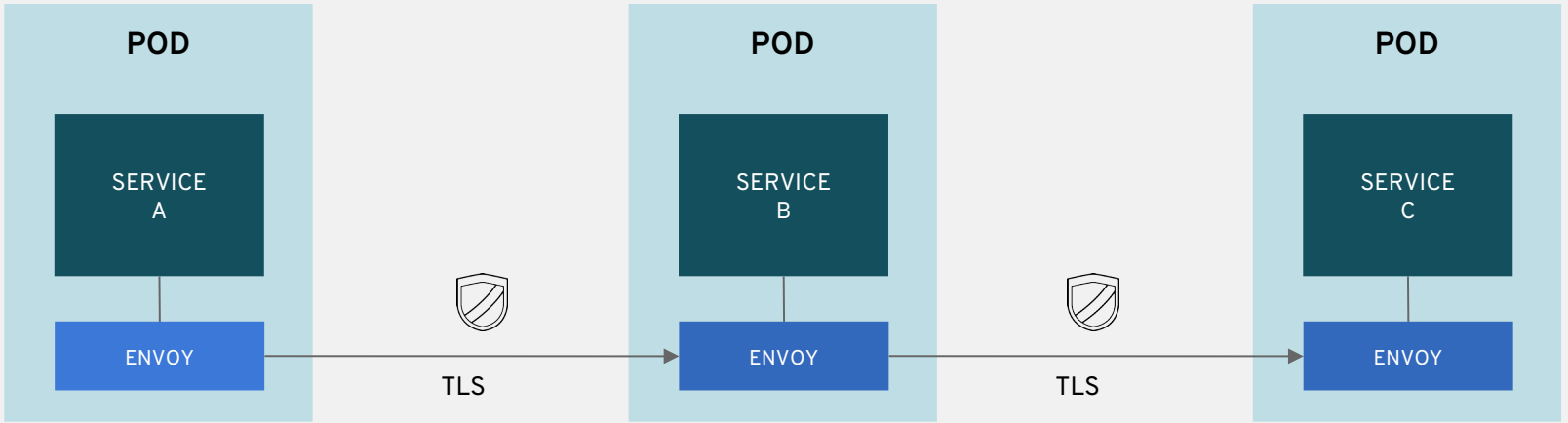
SERVICE SECURITY

SECURE COMMUNICATION WITHOUT ISTIO



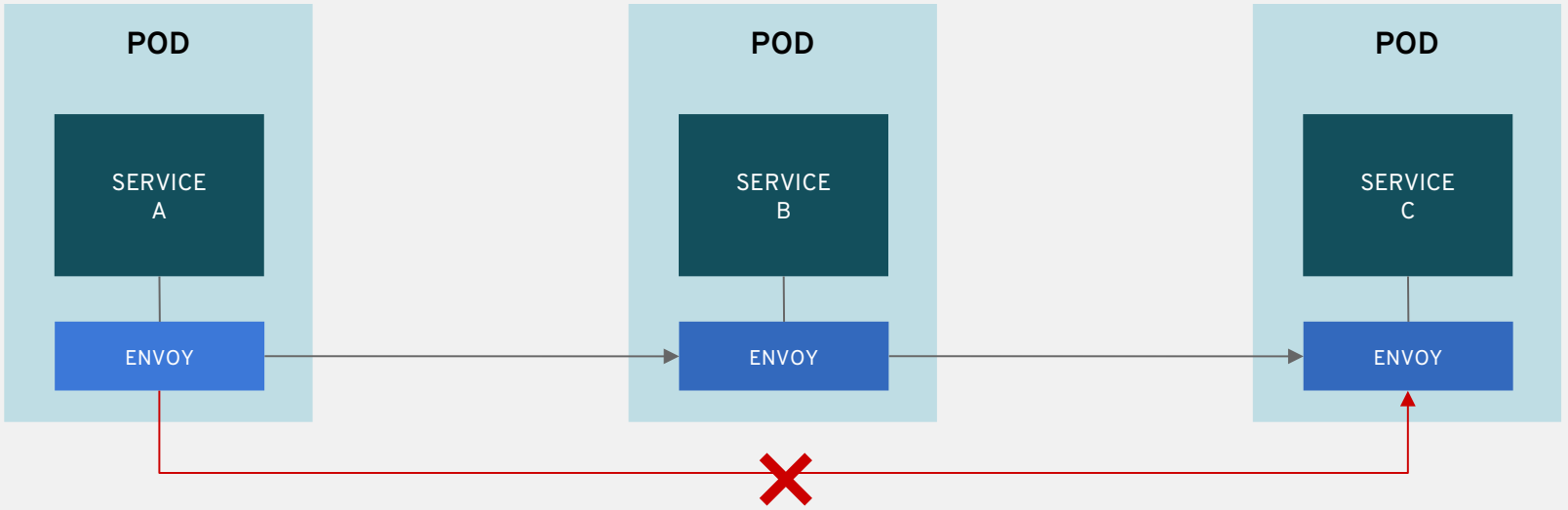
coupled to the service code

SECURE COMMUNICATION WITH ISTIO



mutual TLS authentication, transparent to the services

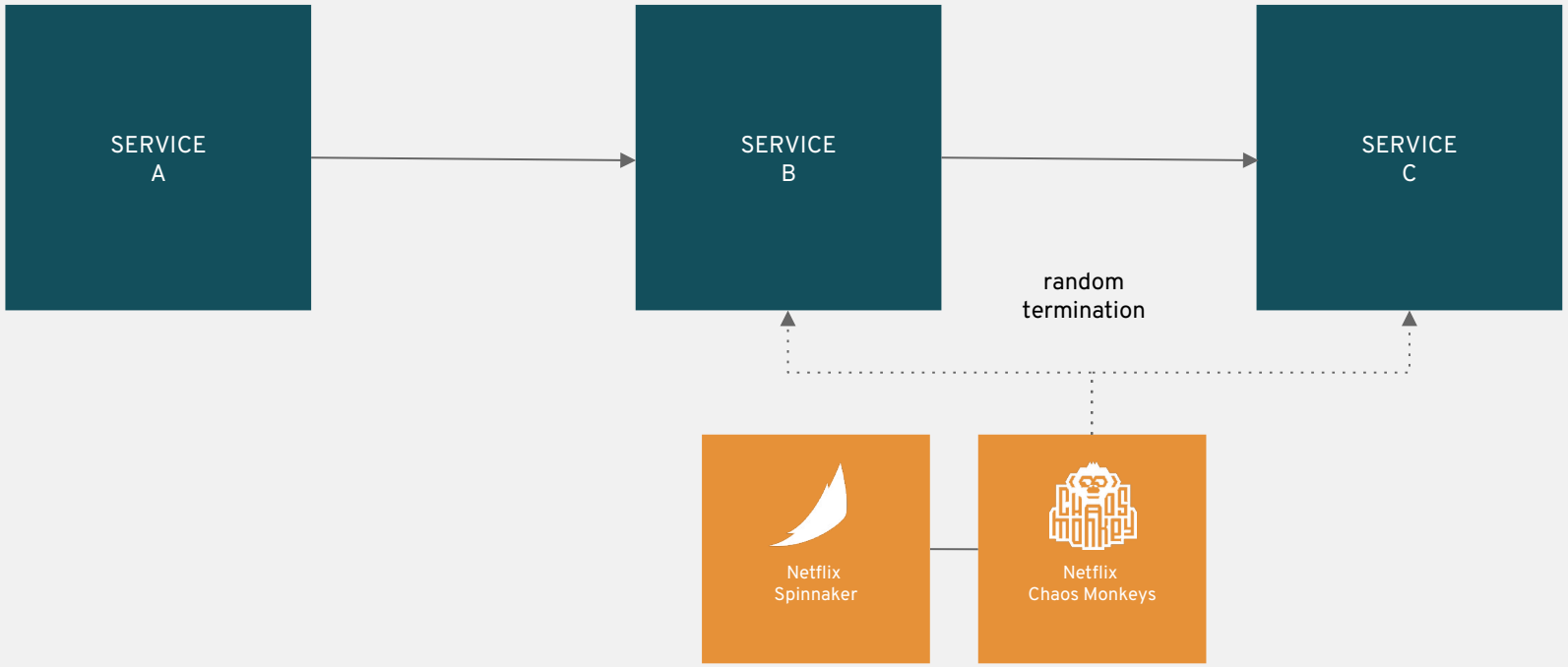
CONTROL SERVICE ACCESS WITH ISTIO



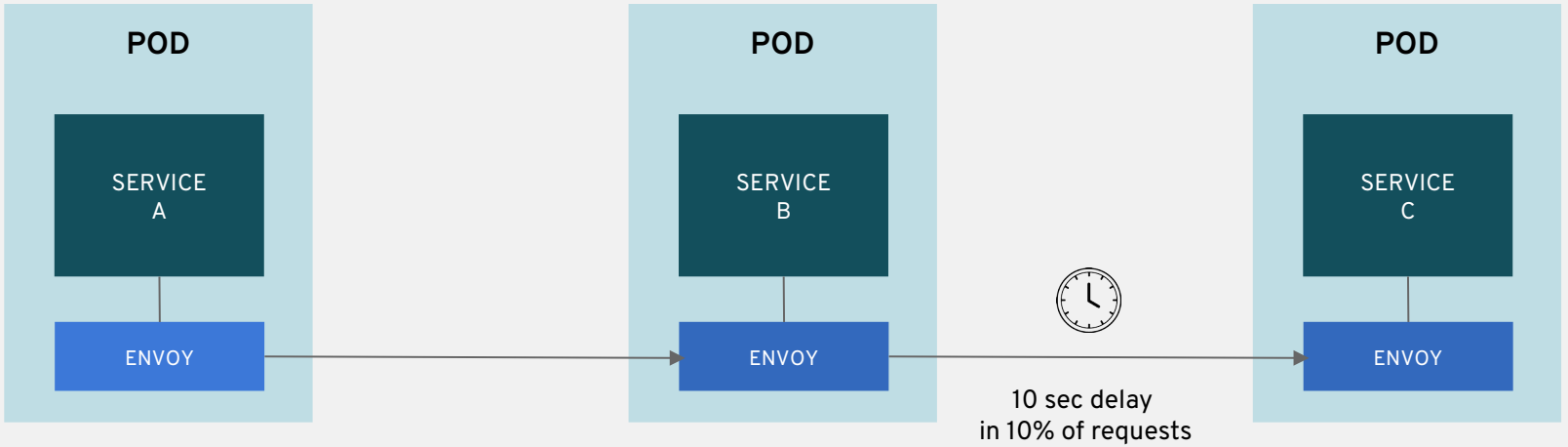
control the service access flow, transparent to the services

CHAOS ENGINEERING

CHAOS ENGINEERING WITHOUT ISTIO

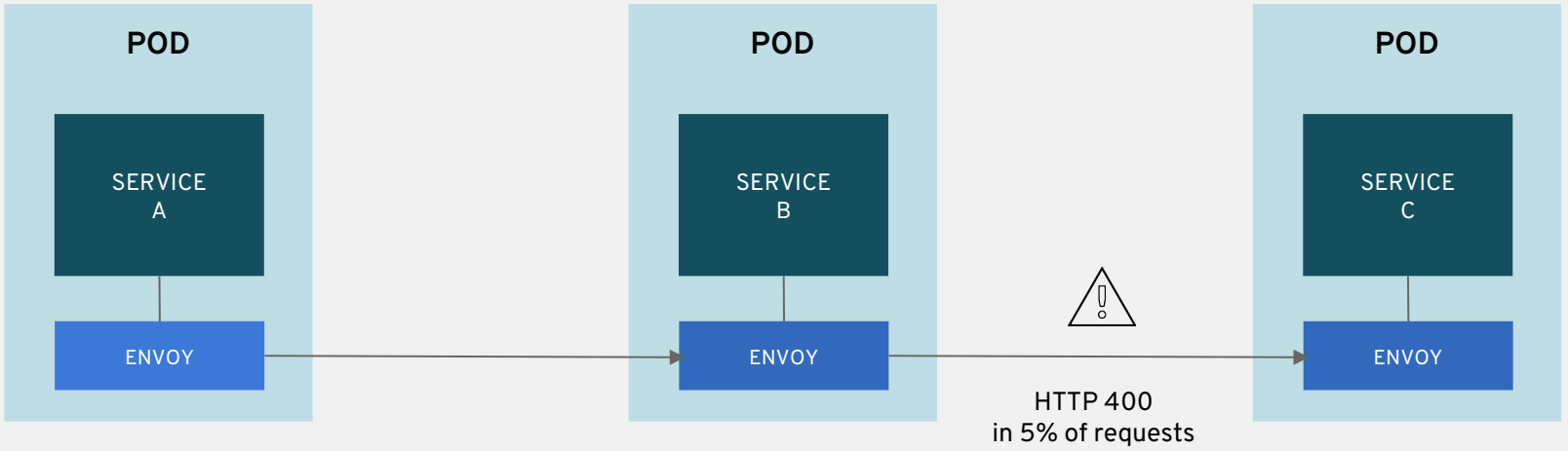


CHAOS ENGINEERING WITH ISTIO



inject delays, transparent to the services

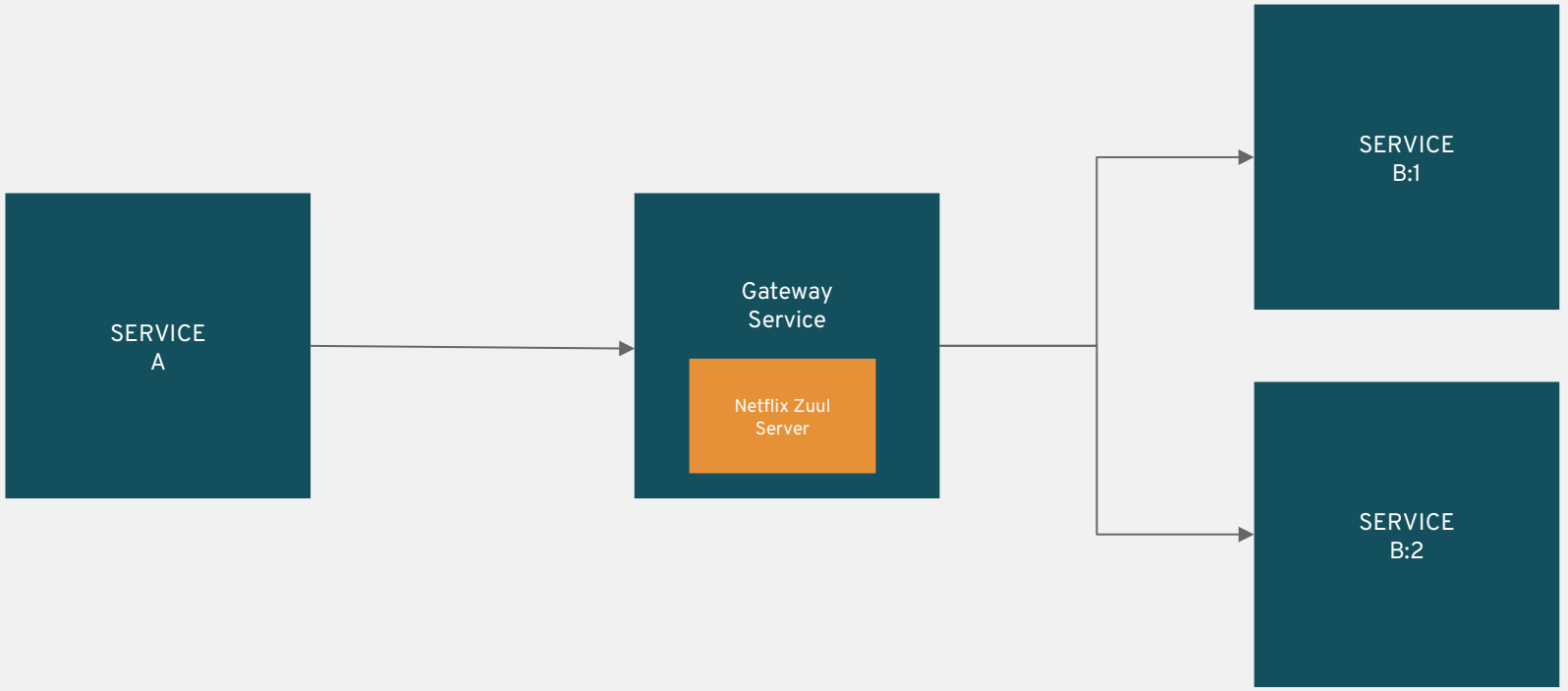
CHAOS ENGINEERING WITH ISTIO



inject protocol-specific errors, transparent to the services

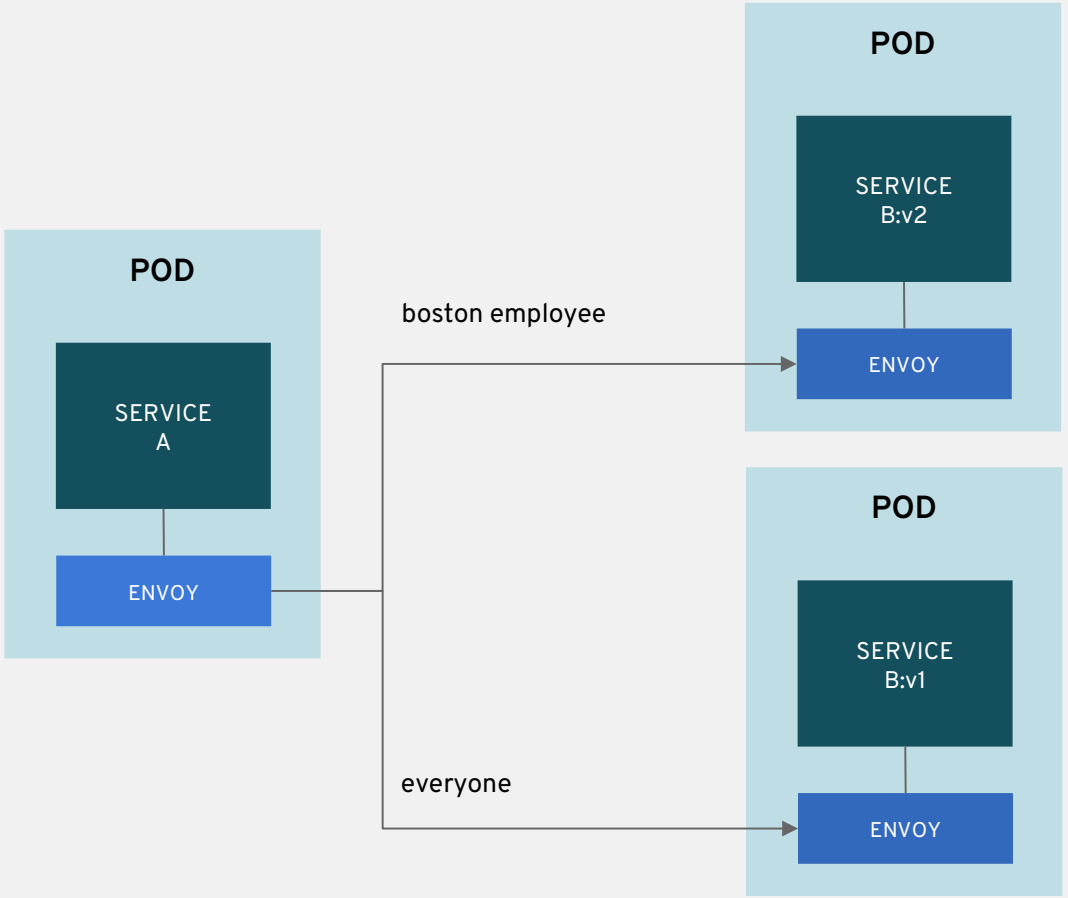
DYNAMIC ROUTING

DYNAMIC ROUTING WITHOUT ISTIO

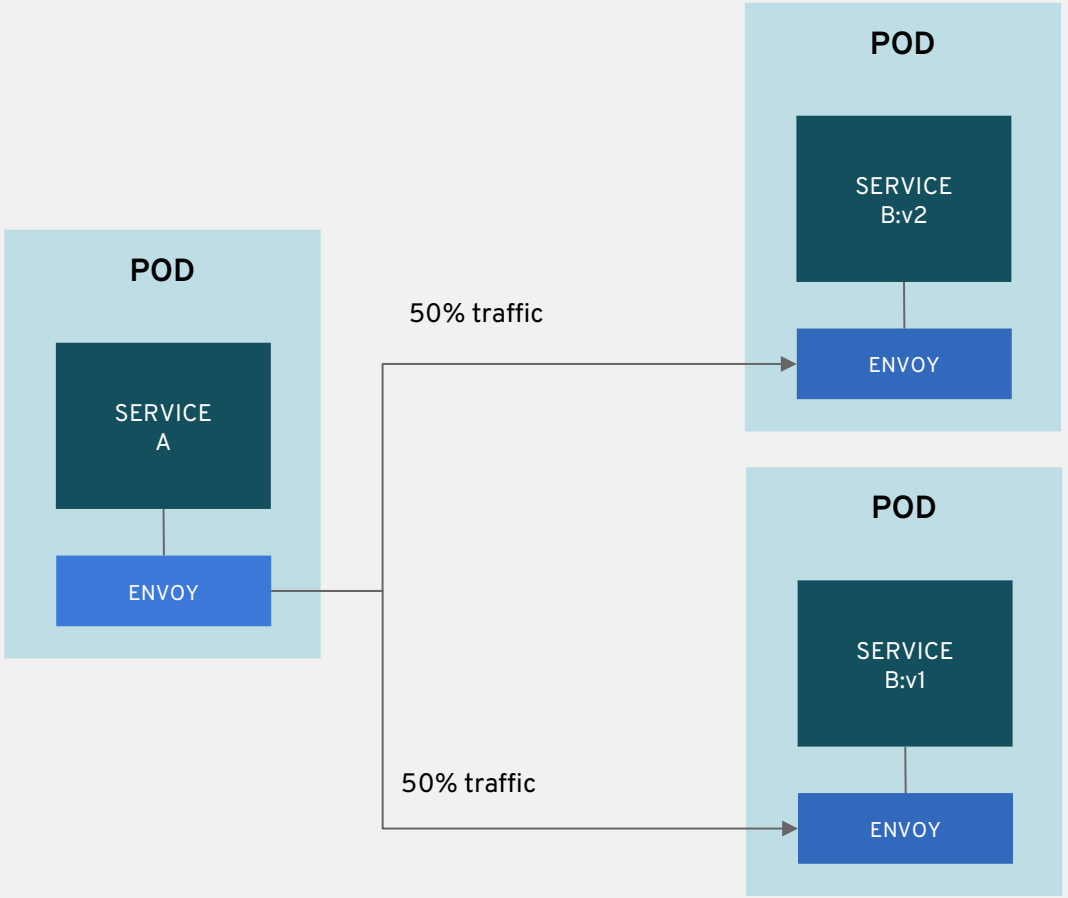


custom code to enable dynamic routing

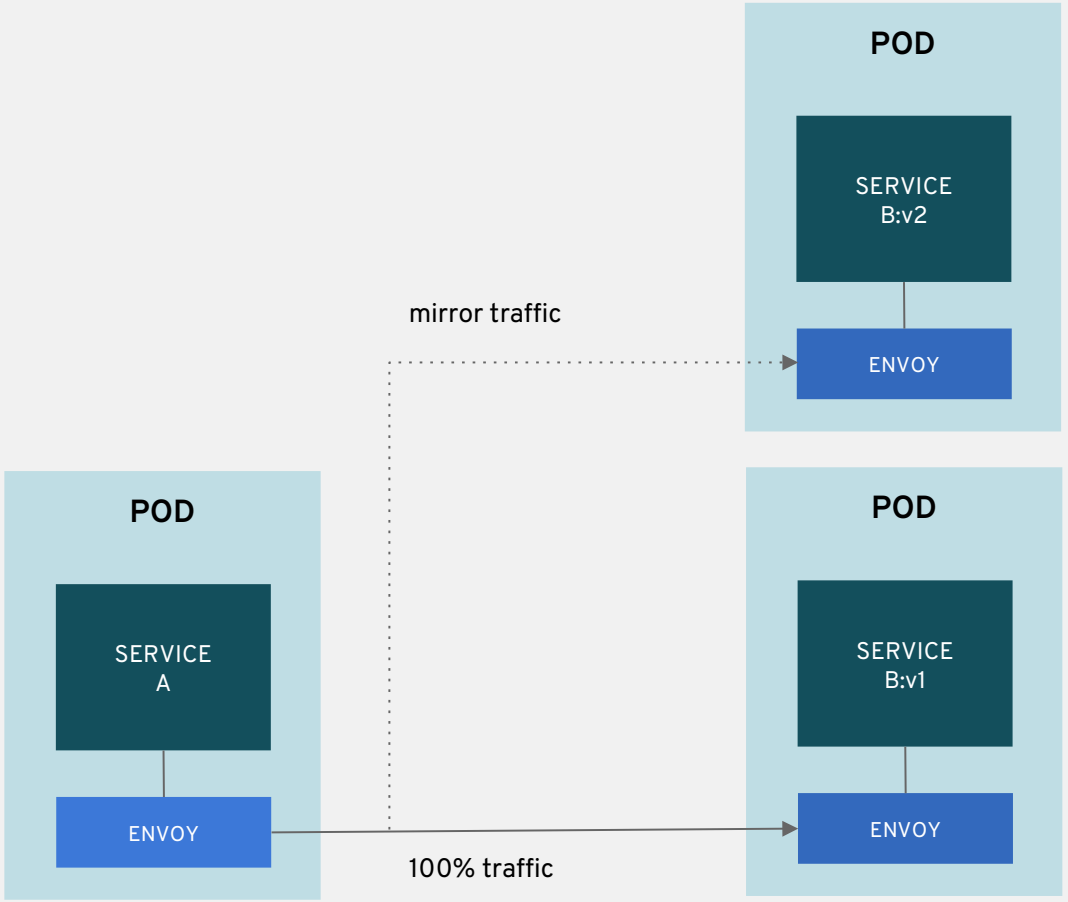
CANARY DEPLOYMENT WITH ISTIO



A/B DEPLOYMENT WITH ISTIO

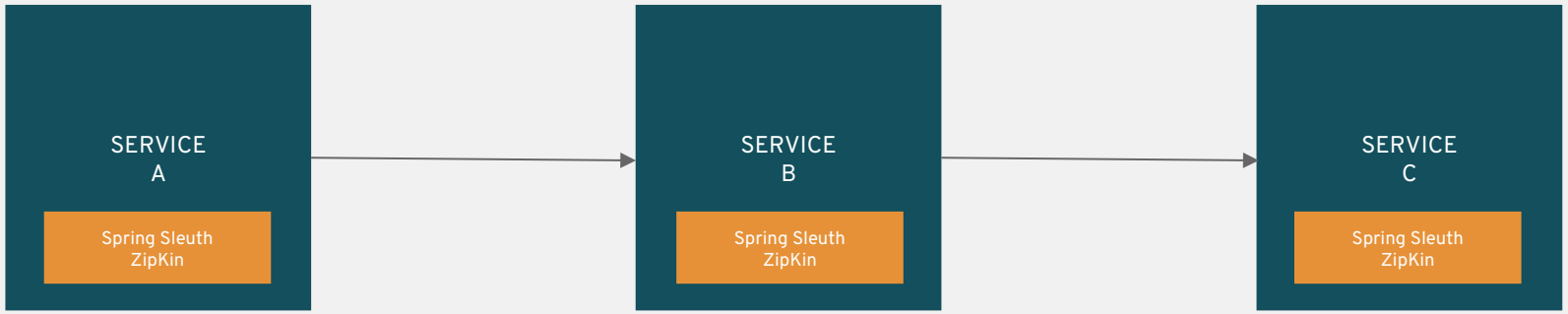


DARK LAUNCHES WITH ISTIO



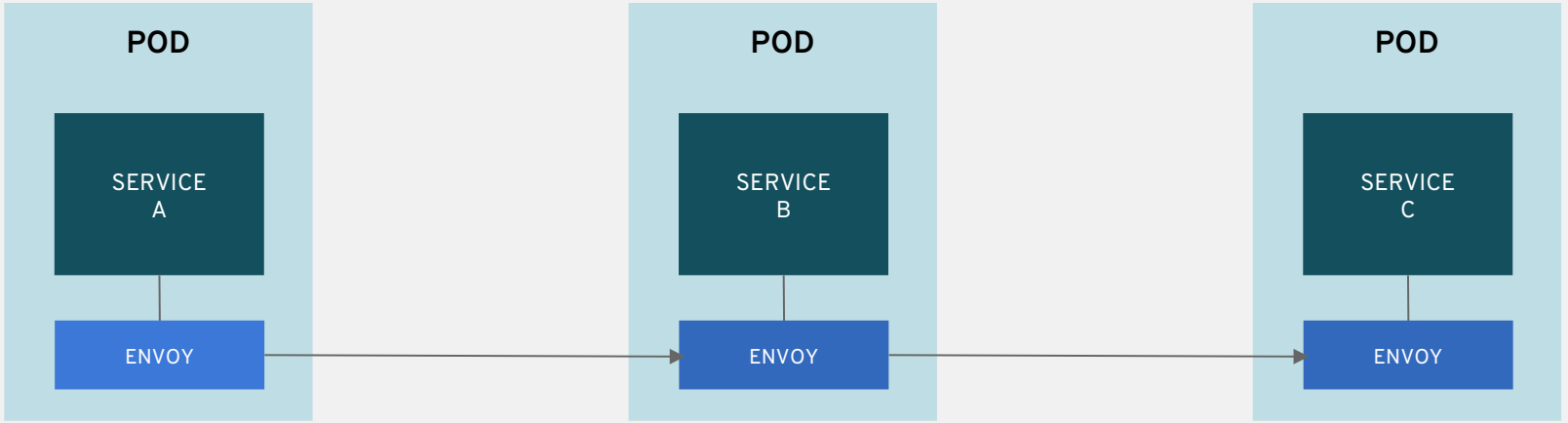
DISTRIBUTED TRACING (JAEGER)

DISTRIBUTED TRACING WITHOUT ISTIO

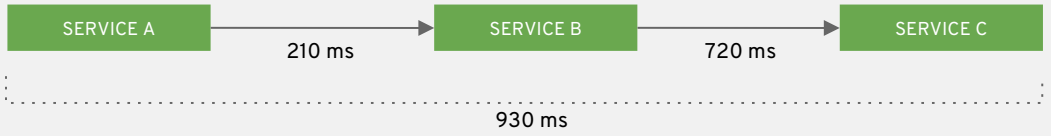


code to enable dynamic tracing

DISTRIBUTED TRACING WITH ISTIO & JAEGER

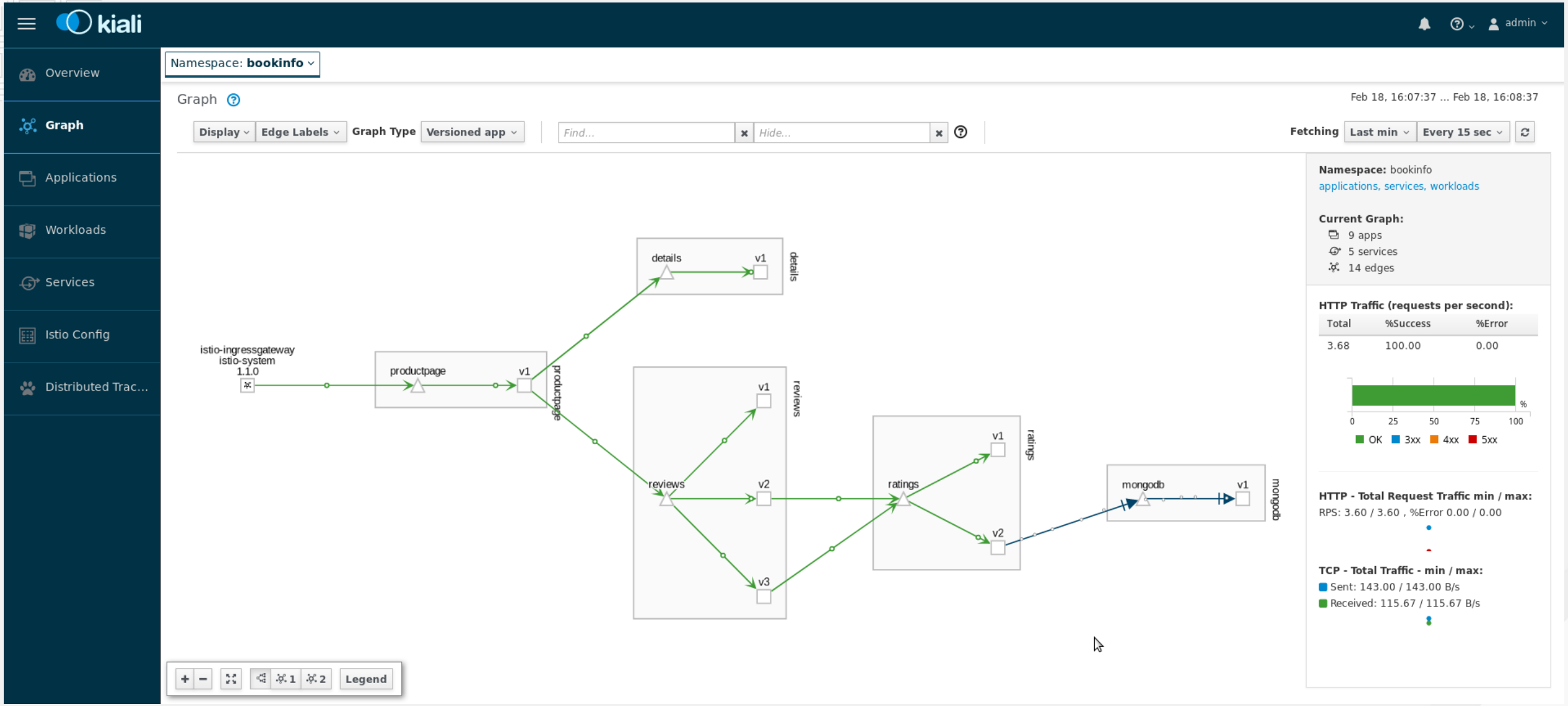


discovers service relationships and process times, transparent to the services

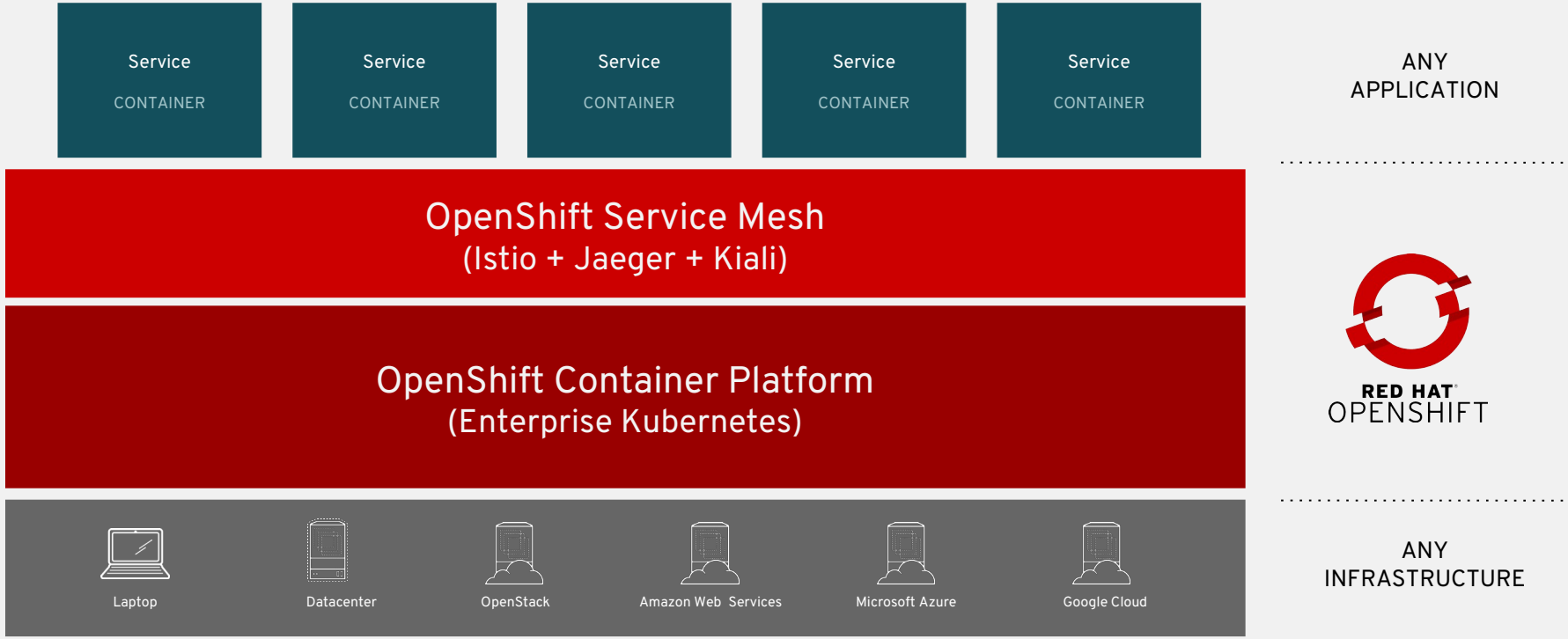


SERVICE MESH OBSERVABILITY (KIALI)

Red Hat


kiali
admin
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
0
25
50
75
100
%
OK
3xx
4xx
5xx
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
istio-ingressgateway
istio-system
1.1.0
productpage
v1
details
reviews
v2
v3
ratings
mongodb
Legend

DISTRIBUTED SERVICES PLATFORM



References

How to explain service mesh in plain English
https://enterprisersproject.com/article/2019/6/service-mesh-plain-english

OpenShift Commons 2019
https://blog.openshift.com/wp-content/uploads/State-of-the-Platform-Services-Integrated-1.pdf

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat