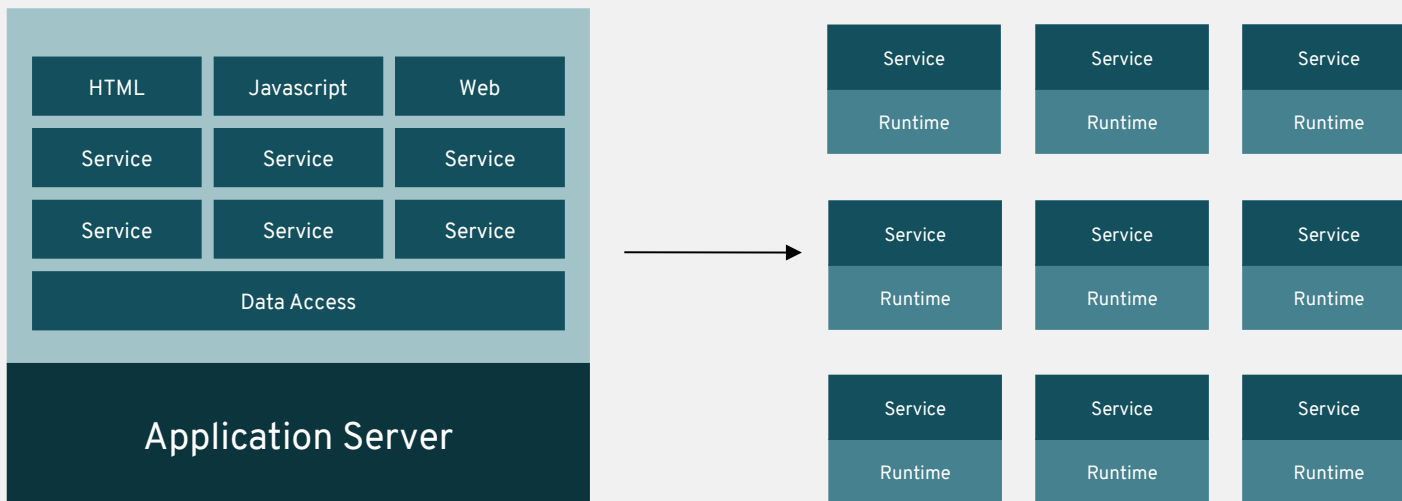




Bereitstellung hybrider Cloud- Anwendungen mit OpenShift und Istio

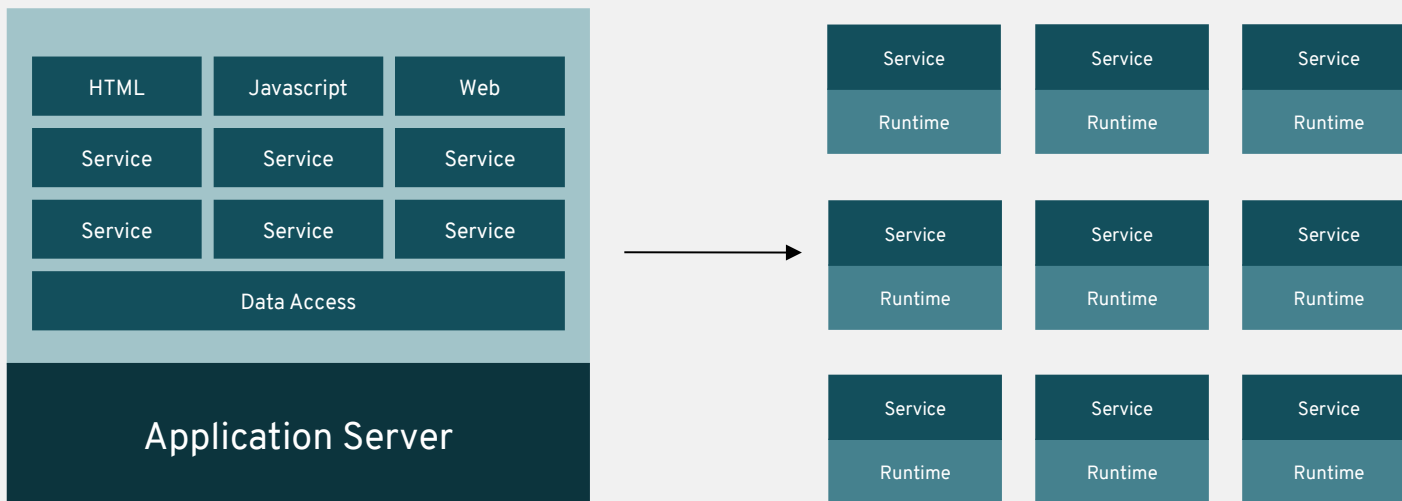
Robert Baumgartner
Senior Solution Architect OpenShift & Middleware
Red Hat Austria

MICROSERVICES ARCHITECTURE



MICROSERVICES ARCHITECTURE

DISTRIBUTED



EIGHT FALLACIES OF DISTRIBUTED COMPUTING

1. The **network** is reliable
2. **Latency** is zero
3. **Bandwidth** is infinite
4. The network is **secure**
5. **Topology** doesn't change
6. There is one **administrator**
7. Transport cost is zero
8. The network is homogeneous



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

Photo: Icon made by Freepik from www.flaticon.com

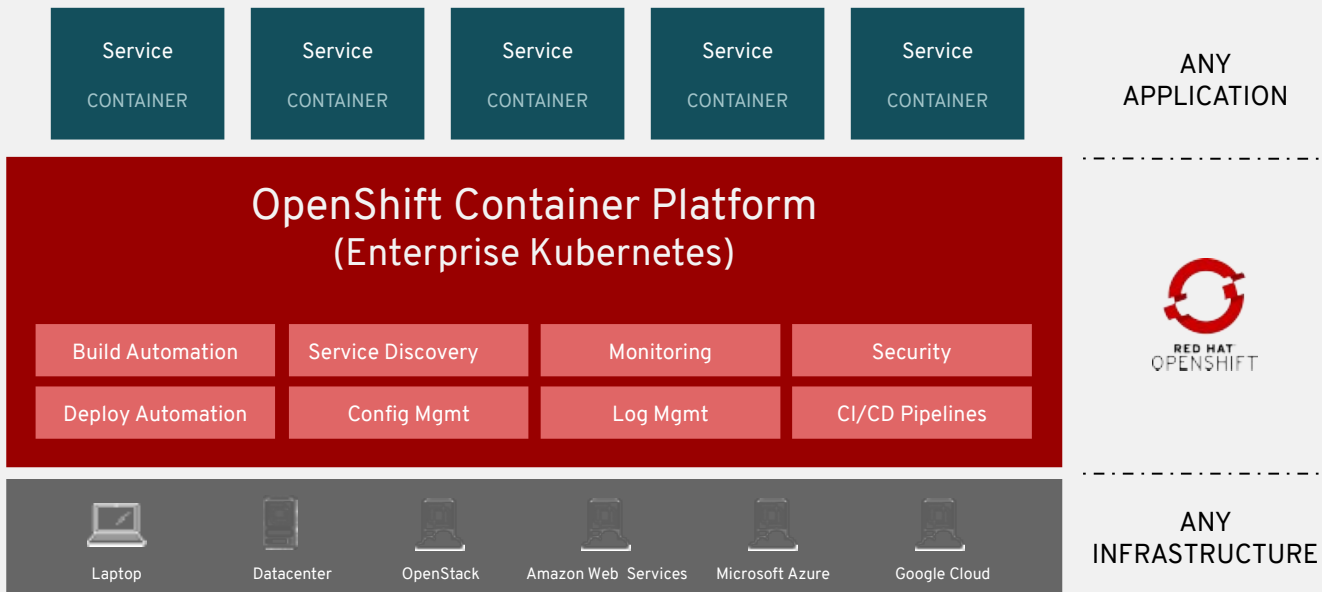


HOW TO DEAL WITH THE COMPLEXITY?

ADDRESS THE COMPLEXITY
IN THE INFRASTRUCTURE



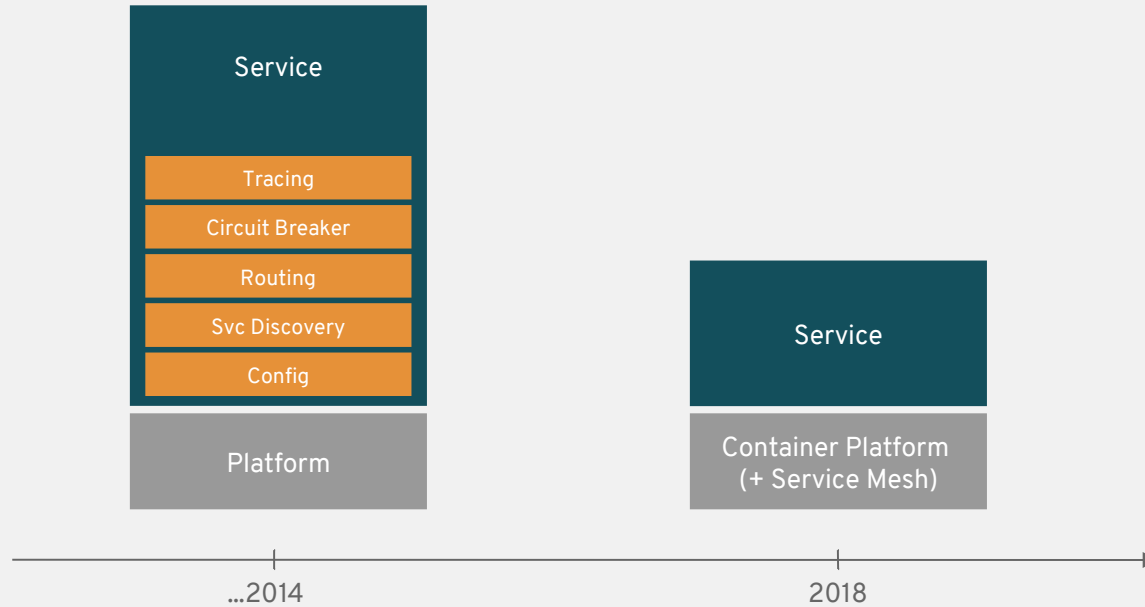
BUILD AND DEPLOY CLOUD-NATIVE APPS WITH RED HAT OPENS SHIFT



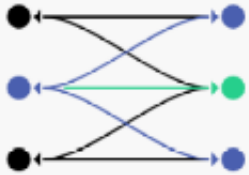
SERVICE MESH

A dedicated network for service-to-service communications

MICROSERVICES EVOLUTION



SERVICE MESH



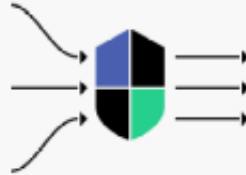
Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



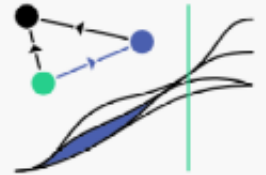
Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



Control

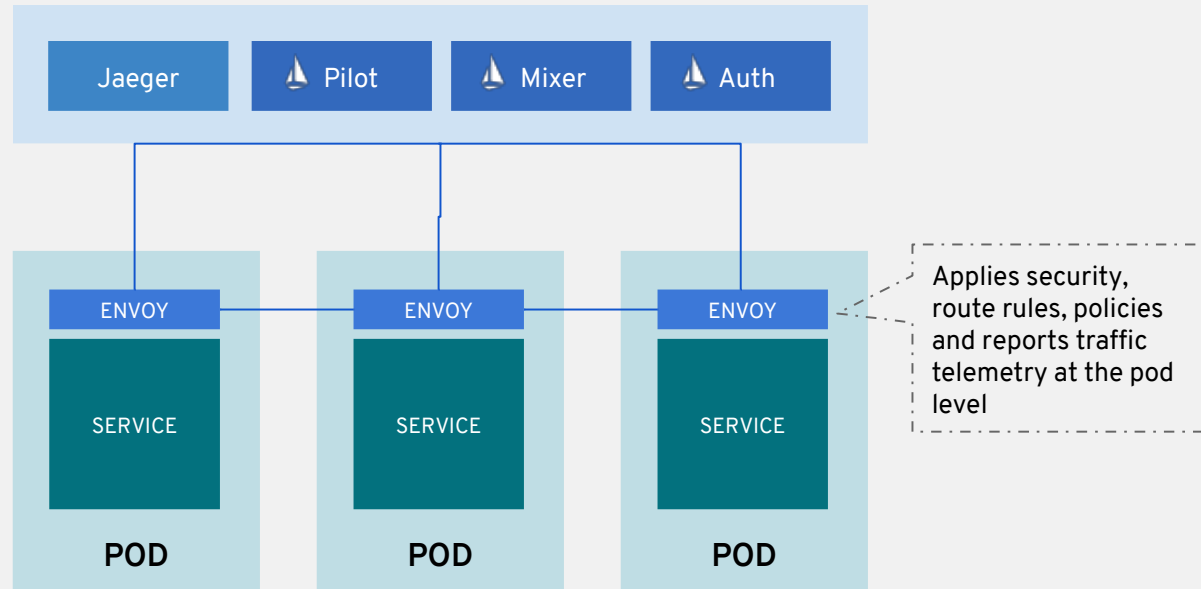
Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.



Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.

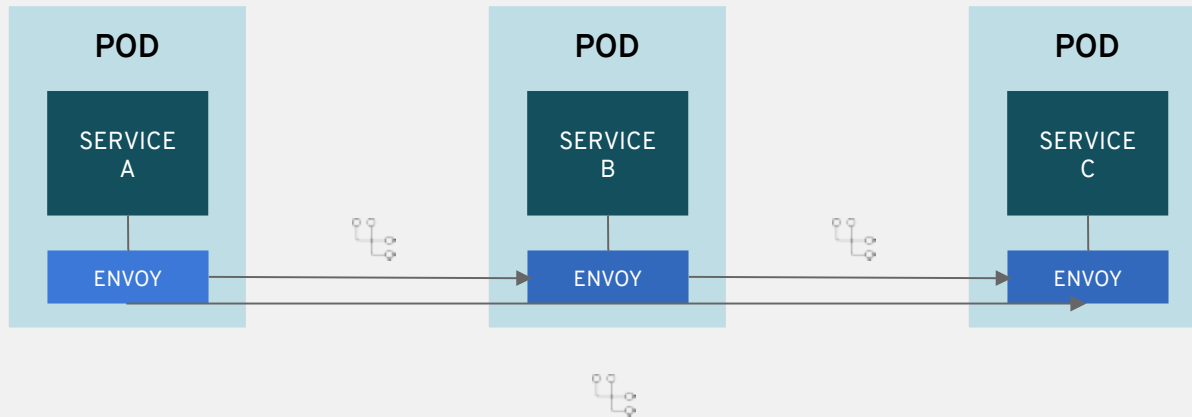
SERVICE MESH ARCHITECTURE



FAULT TOLERANCE

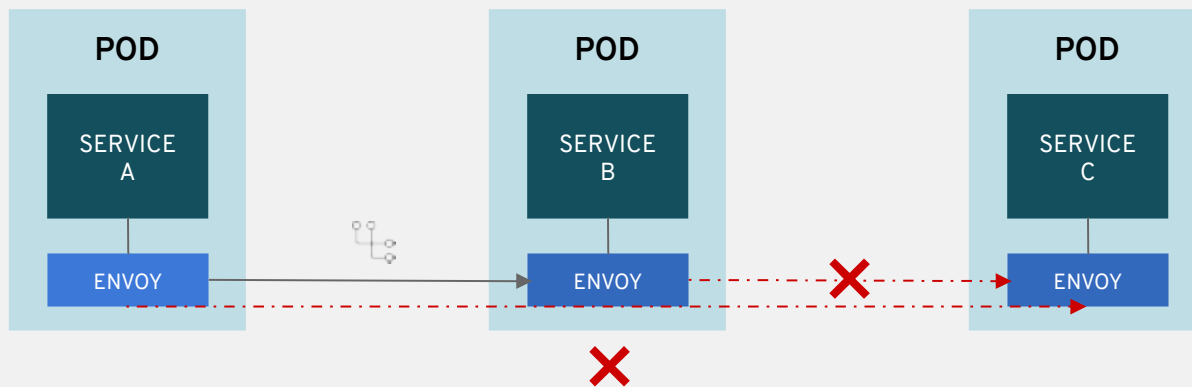


CIRCUIT BREAKERS WITH ISTIO



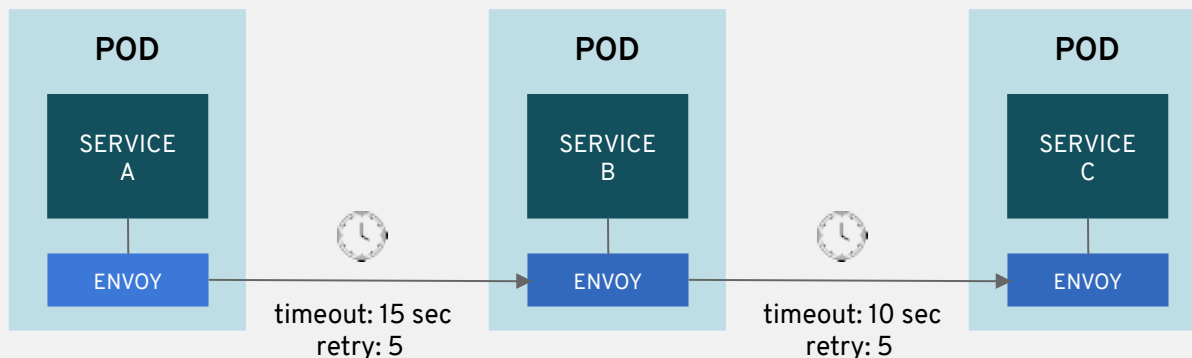
transparent to the services

CIRCUIT BREAKERS WITH ISTIO



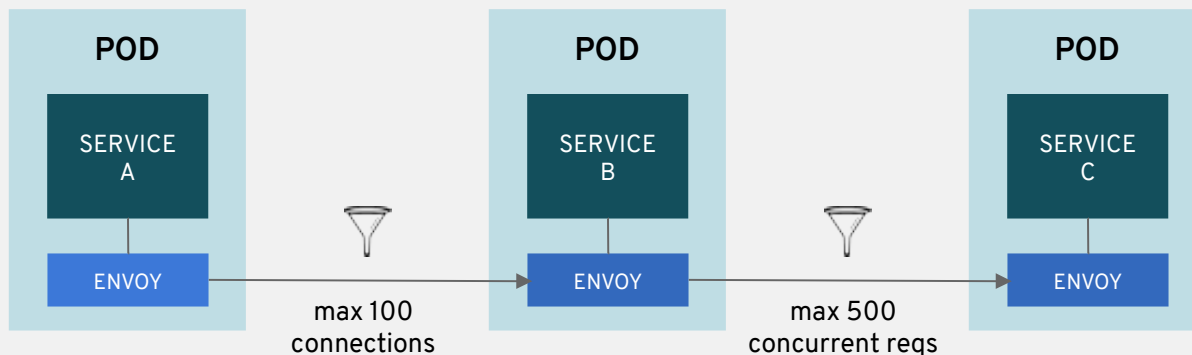
improved response time with global circuit status

TIMEOUTS AND RETRIES WITH ISTIO



configure timeouts and retries, transparent to the services

RATE LIMITING WITH ISTIO

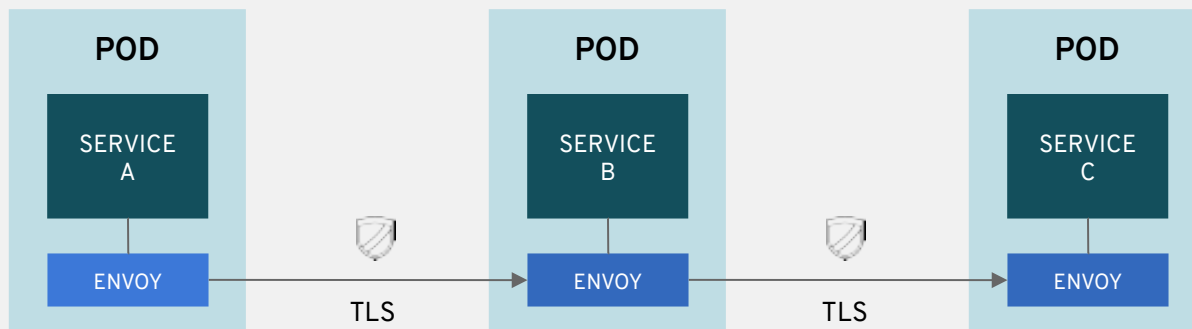


limit invocation rates, transparent to the services

SERVICE SECURITY

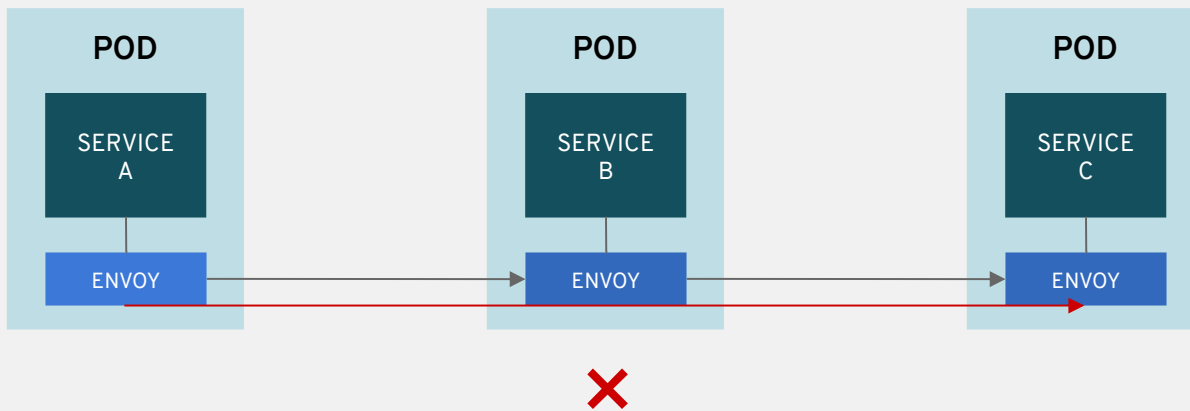


SECURE COMMUNICATION WITH ISTIO



mutual TLS authentication, transparent to the services

CONTROL SERVICE ACCESS WITH ISTIO

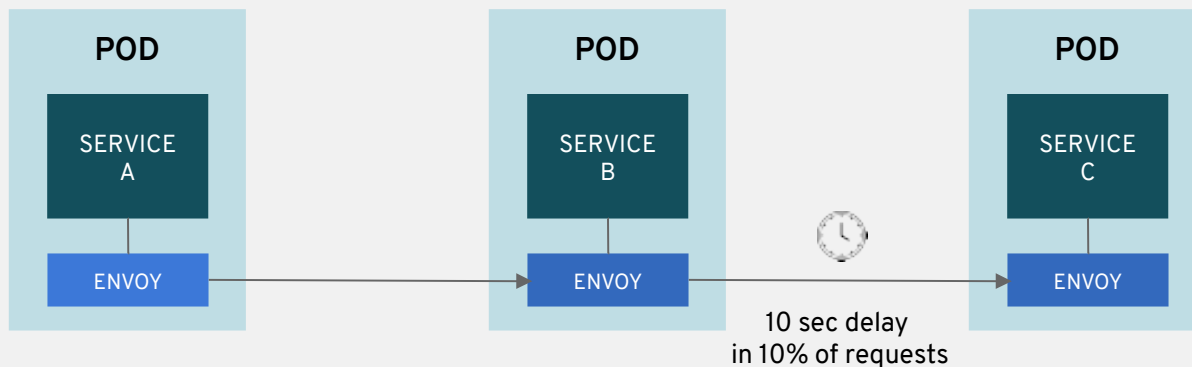


control the service access flow, transparent to the services

CHAOS ENGINEERING

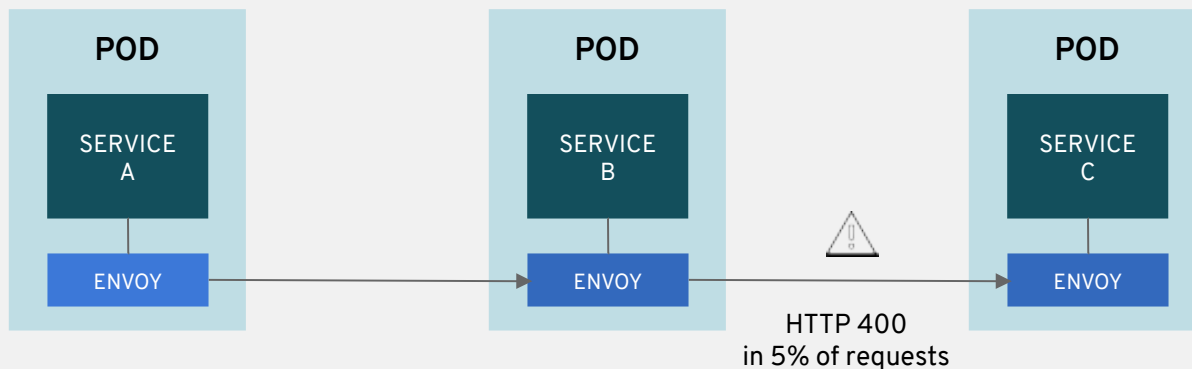


CHAOS ENGINEERING WITH ISTIO



inject delays, transparent to the services

CHAOS ENGINEERING WITH ISTIO

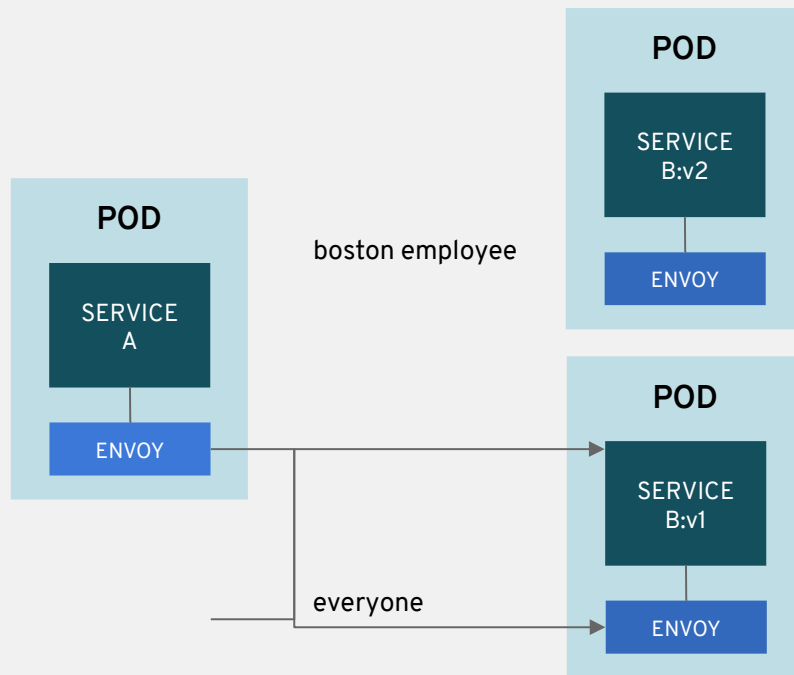


inject protocol-specific errors, transparent to the services

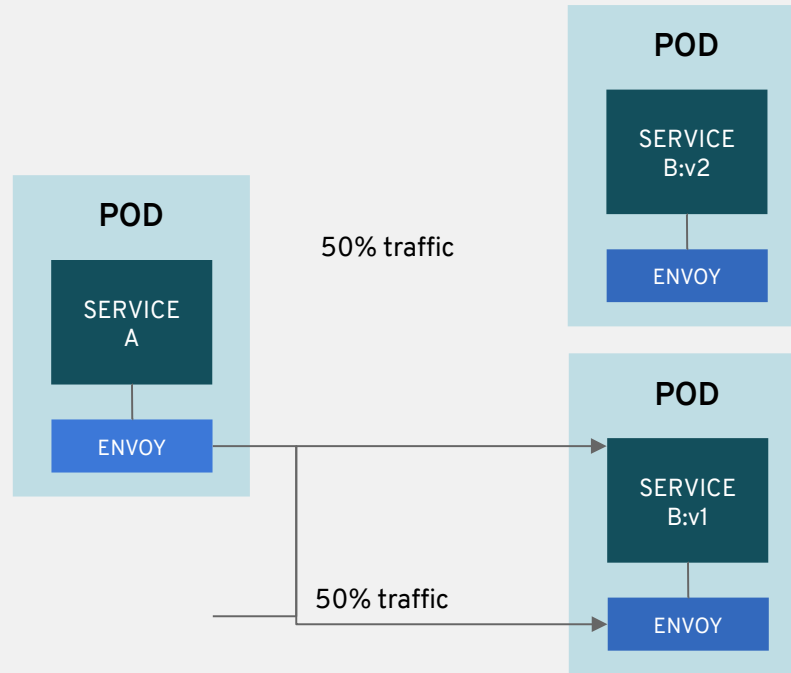
DYNAMIC ROUTING



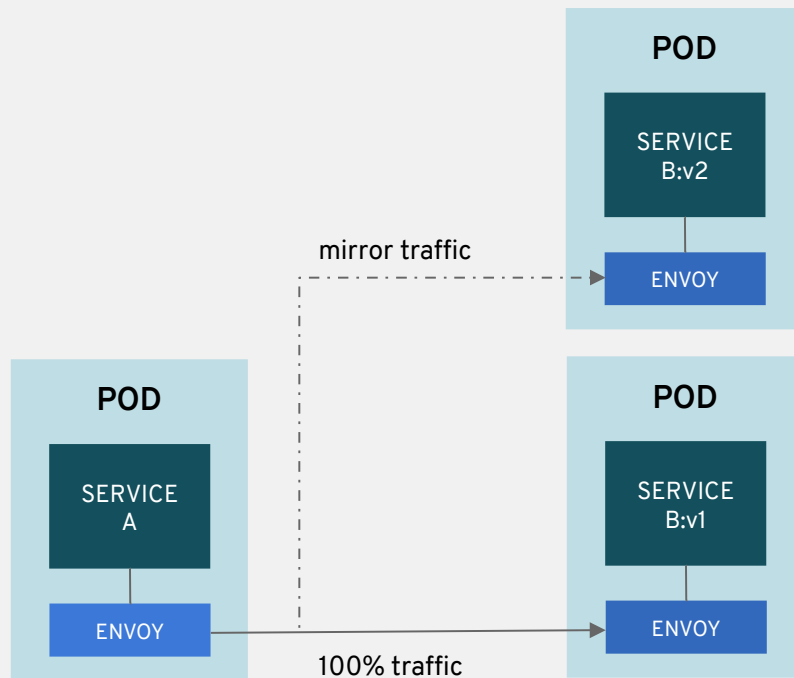
CANARY DEPLOYMENT WITH ISTIO



A/B DEPLOYMENT WITH ISTIO



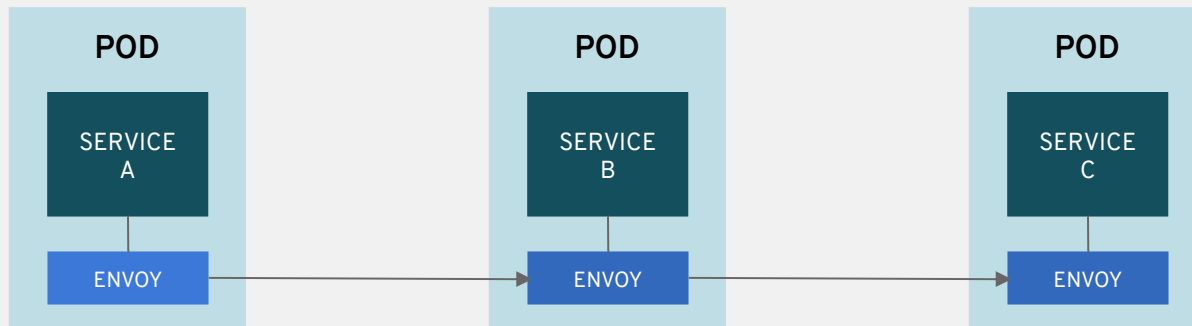
DARK LAUNCHES WITH ISTIO



DISTRIBUTED TRACING



DISTRIBUTED TRACING WITH ISTIO & JAEGER



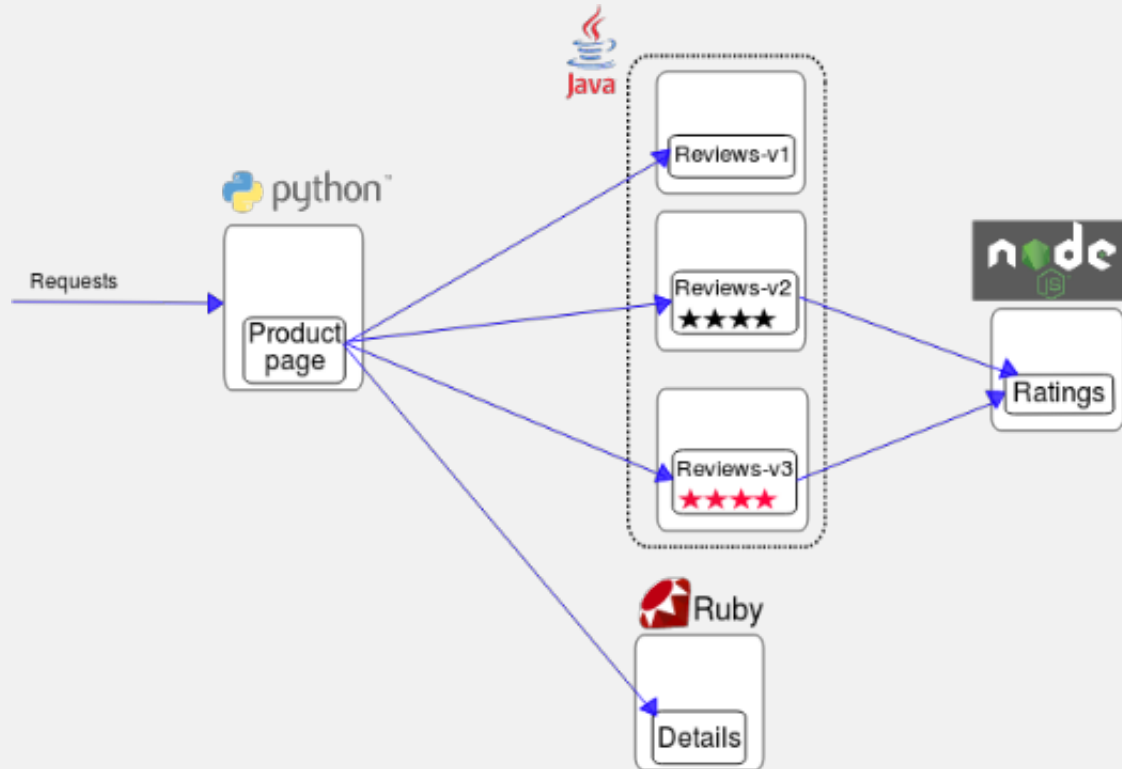
discovers service relationships and process times,
transparent to the services



Demo



Application: Bookinfo



Environment

OpenShift Container Platform 3.11
OpenShift Service Mesh 0.4 Technical Preview
(based on Istio 1.0.3)

- <https://docs.openshift.com/container-platform/3.11/servicemesh-install/servicemesh-install.html>

Istio

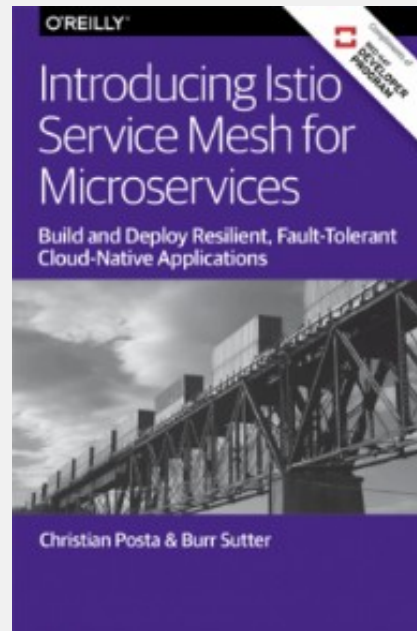
- <https://istio.io/>

Istio on Openshift Tutorial

- <https://developers.redhat.com/books/introducing-istio-service-mesh-microservices/>

Book

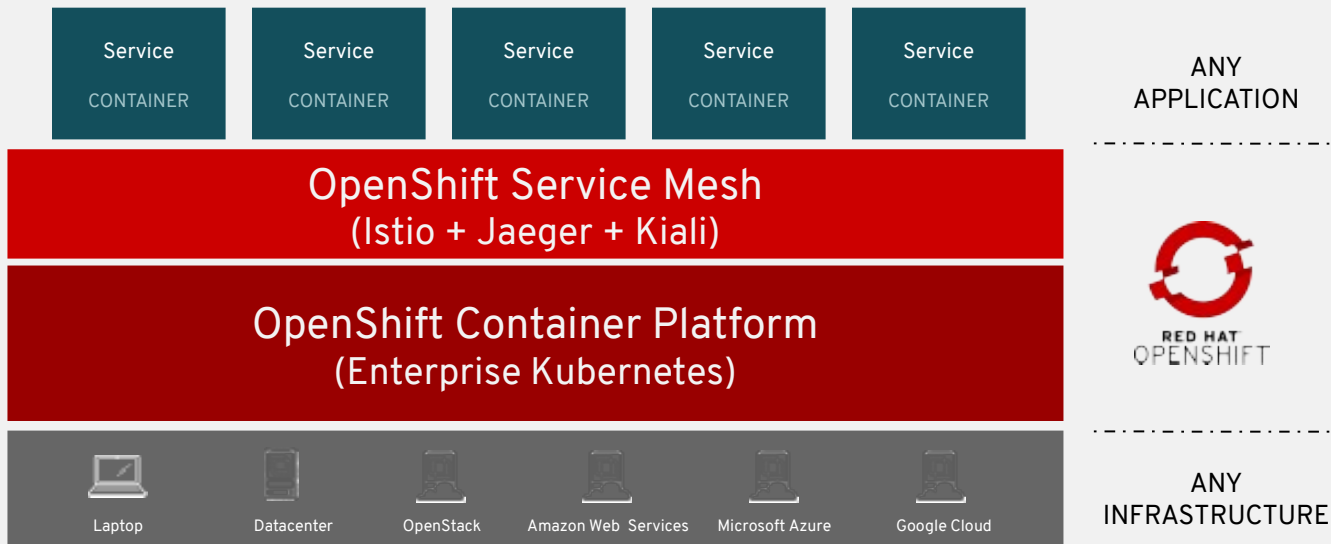
- <https://developers.redhat.com/books/introducing-istio-service-mesh-microservices/>



TAKEWAY



DISTRIBUTED SERVICES PLATFORM



try it at <https://learn.openshift.com/servicemesh/>

Questions?





THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHat



youtube.com/user/RedHatVideos