



ISTIO: Service mesh e microservizi

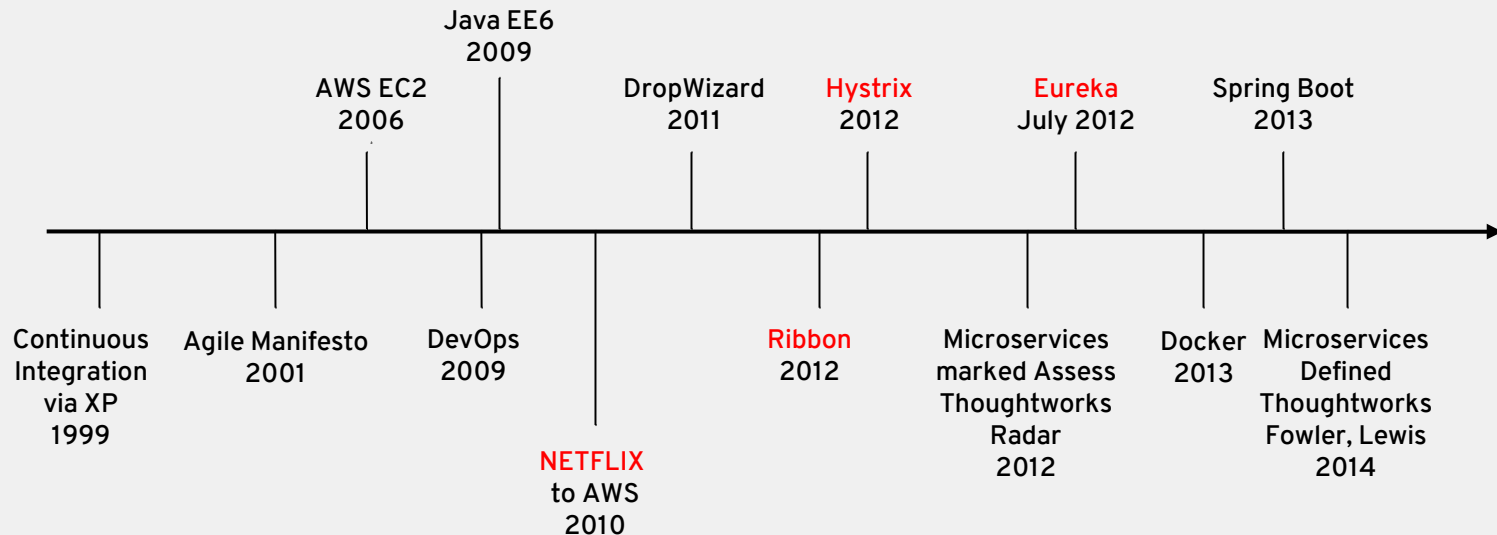
Giuseppe Bonocore - Solution Architect

Ugo Landini - Solution Architect



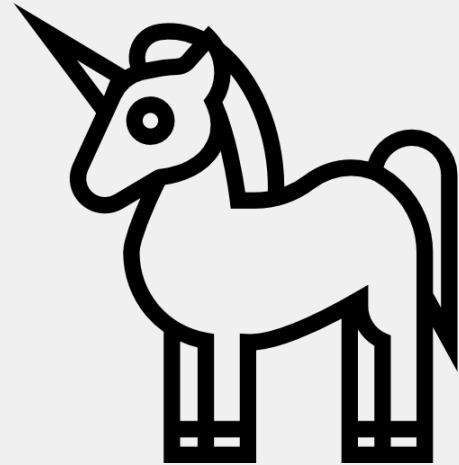
#RedHatOSD

SHORT HISTORY OF MICROSERVICES



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



Java Microservices Platform circa 2014



NETFLIX Ribbon



Why these components?



Eureka is the Service Registry where the clients lookup for service locations a.k.a Service Discovery



Config Server externalized the Configuration



Ribbon is the client side Load Balancer



Hystrix is the Circuit Breaker



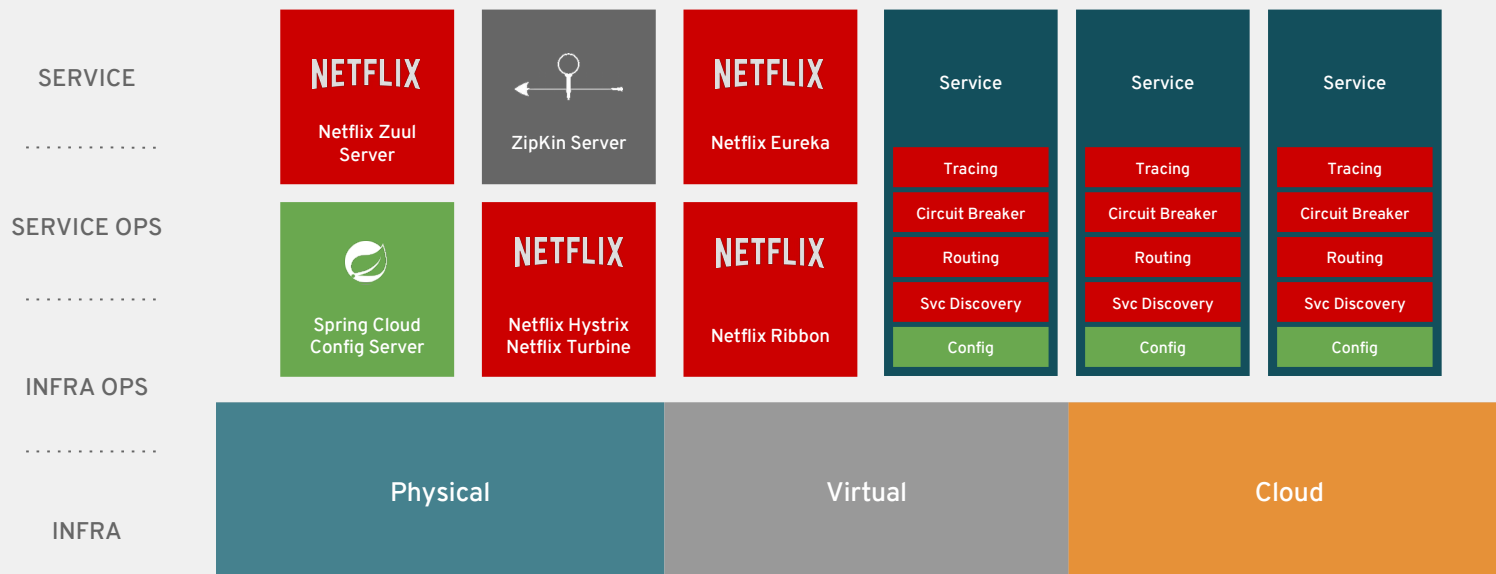
Zipkin is the Distributed Tracer



Zuul is the smart proxy purely based on Java



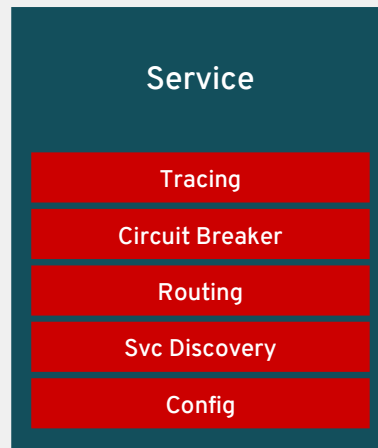
MICROSERVICES **EMBEDDING** CAPABILITIES



AN OPERATION NIGHTMARE!

Infra capabilities are tightly coupled with applications and services

- ✘ Incompatible across languages and frameworks
- ✘ Existing apps require refactoring
- ✘ Upgrades needs tight coordinations libraries



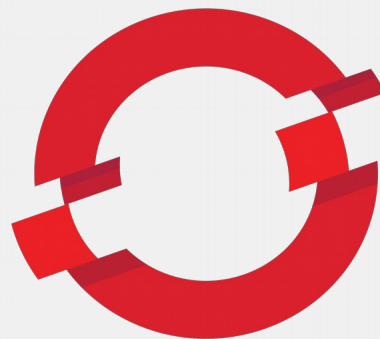
Infrastructure cluttering your code?

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-config</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-zuul</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-hystrix</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
```





kubernetes



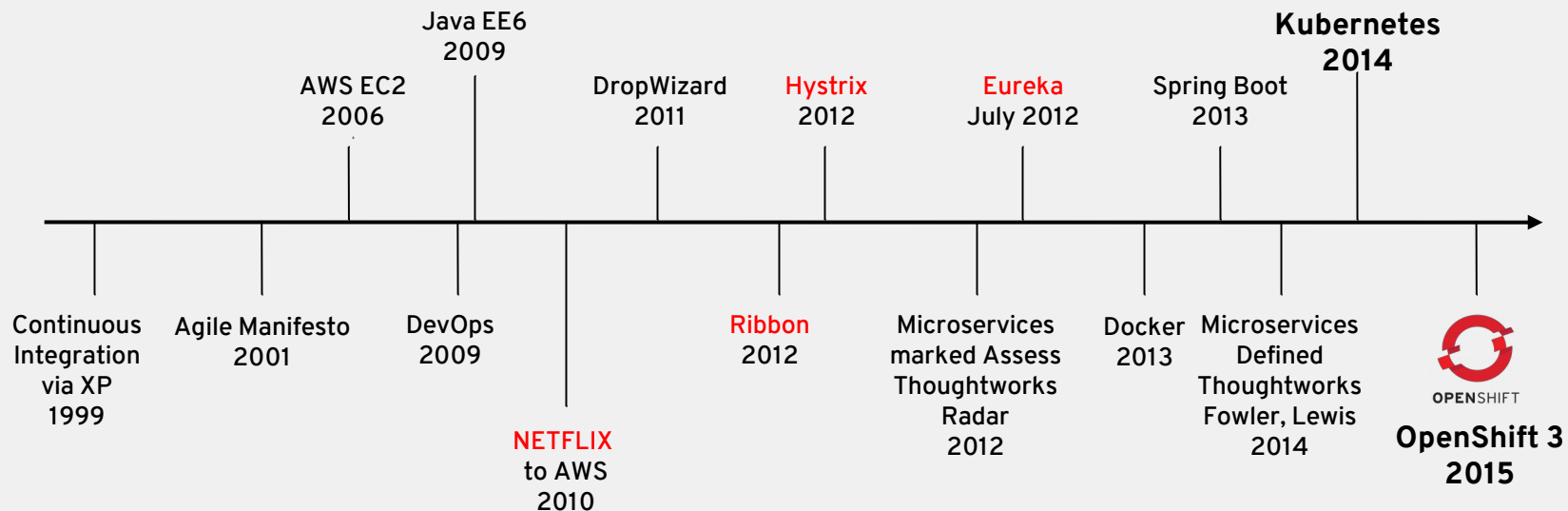
OPENS SHIFT



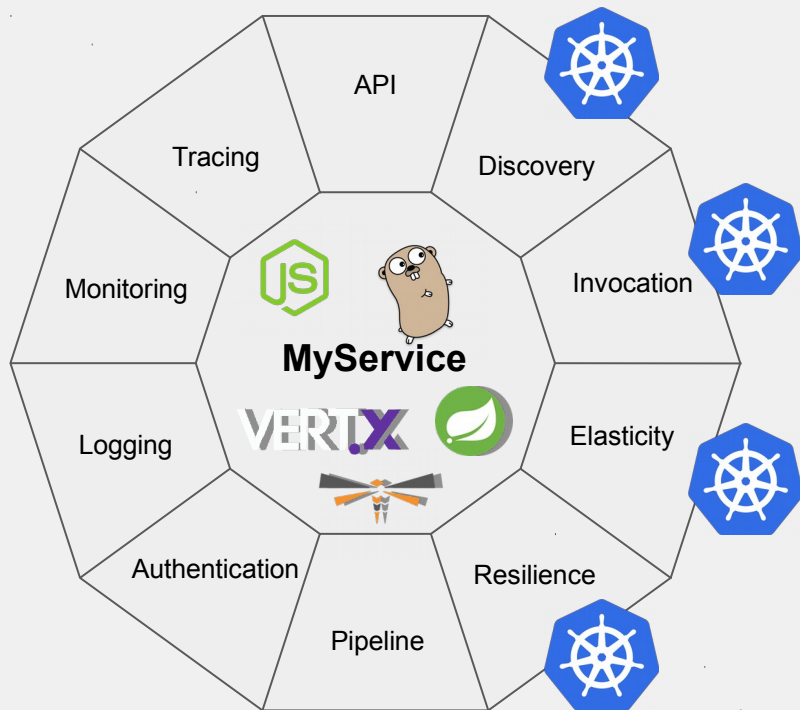
#RedHatOSD



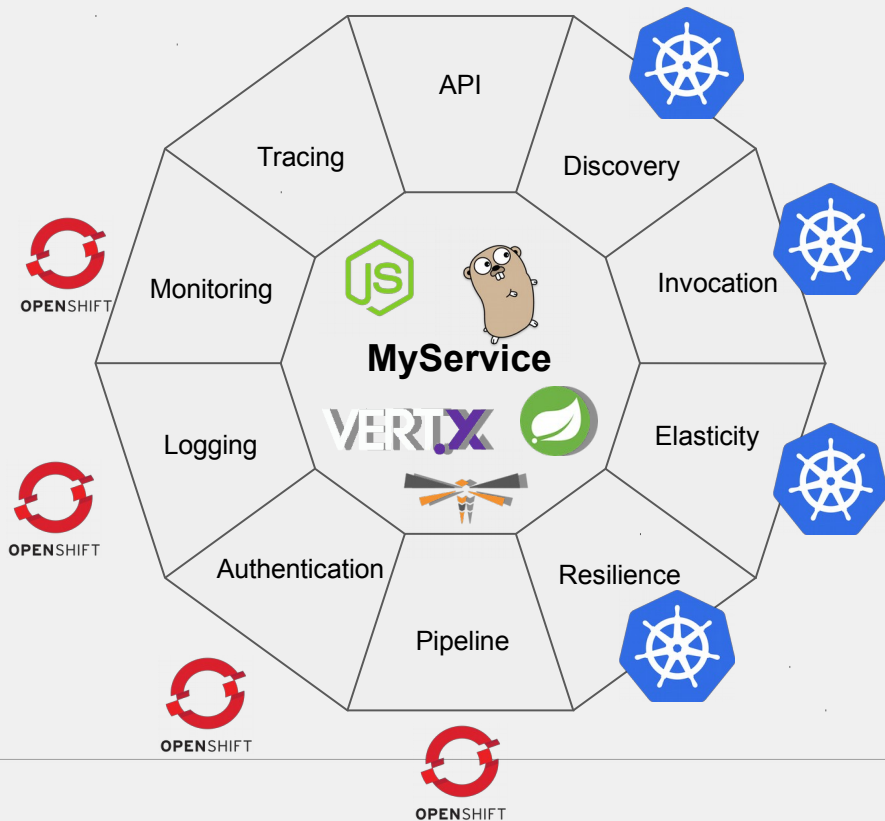
SHORT HISTORY OF MICROSERVICES



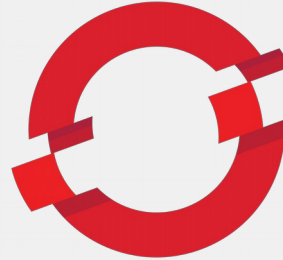
MICROSERVICES'ILITIES + KUBERNETES



MICROSERVICES'ILITIES + OPENSHIFT



Better Microservices Platform circa 2016



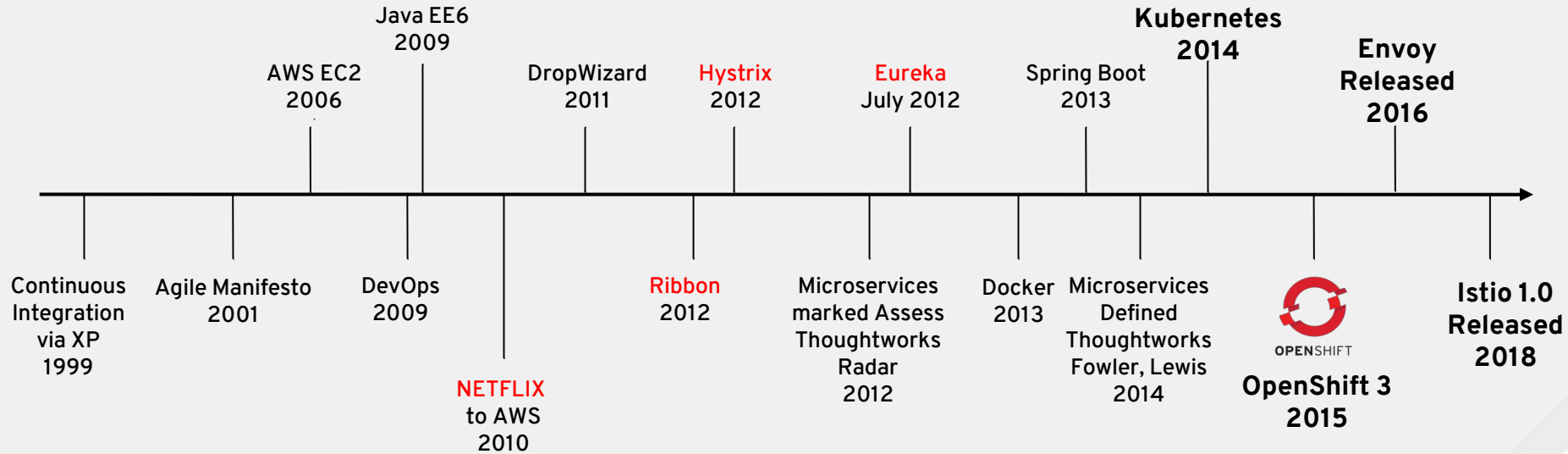
kubernetes OPENSIFT



HYSTRIX
DEFEND YOUR APP



SHORT HISTORY OF MICROSERVICES





Istio - Sail

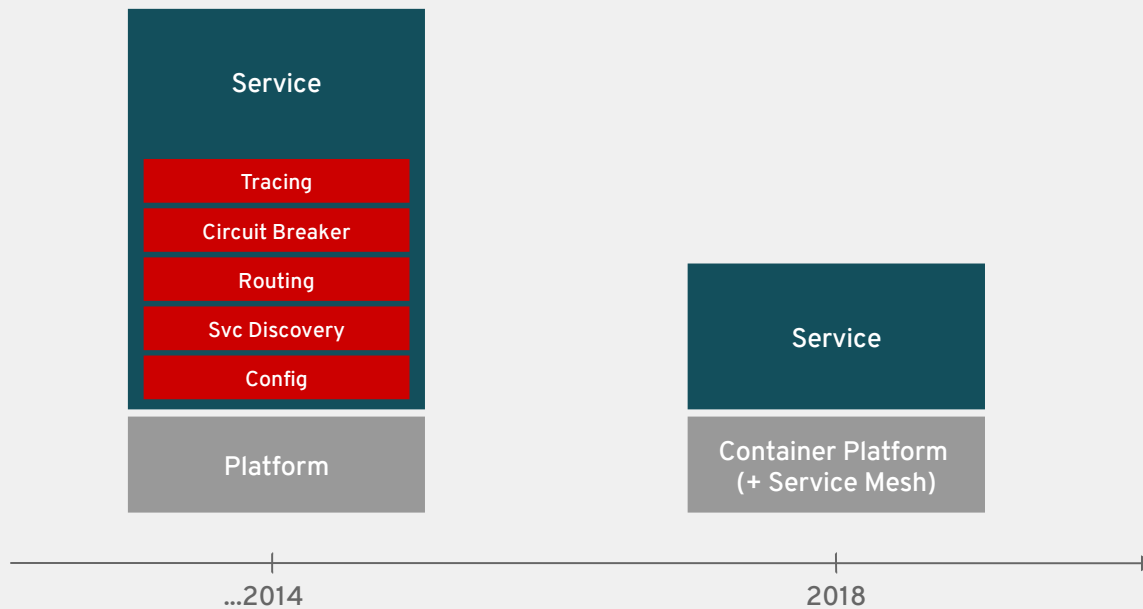
(Kubernetes - Helmsman or ship's pilot)



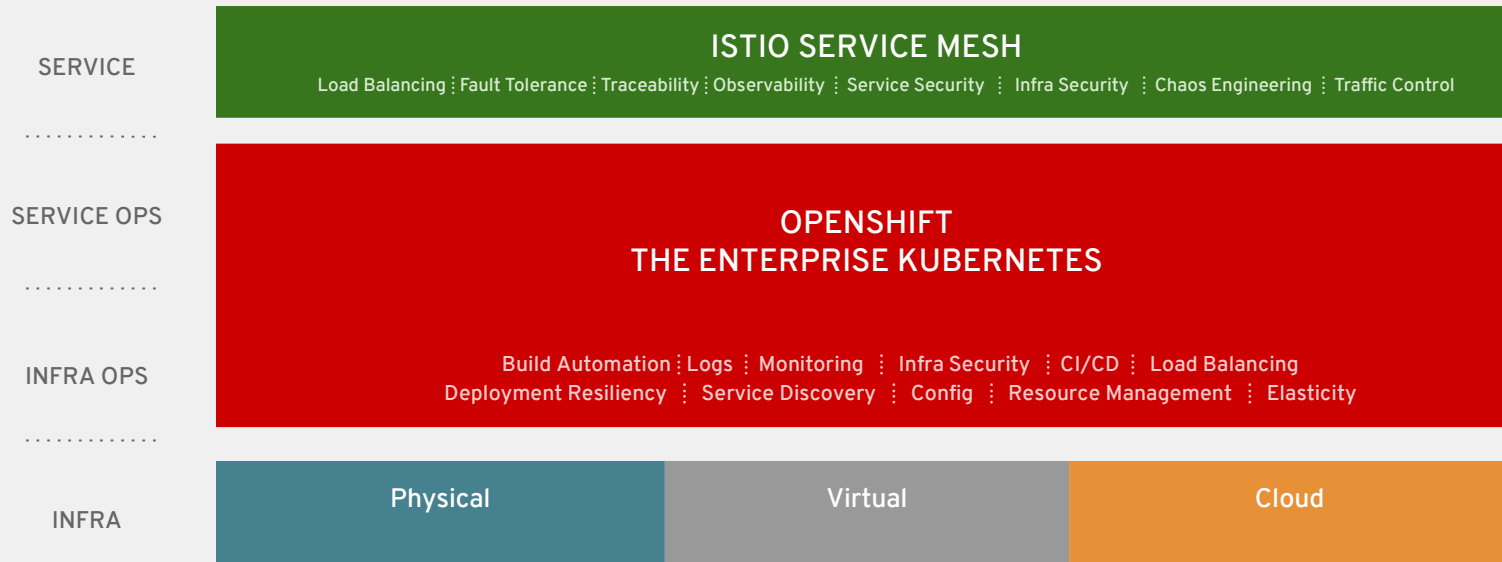
#RedHatOSD



EVOLUTION OF SERVICES



DISTRIBUTED SERVICES WITH ISTIO SERVICE MESH

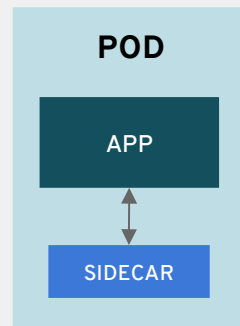


Sidecars



SIDECAR PATTERN

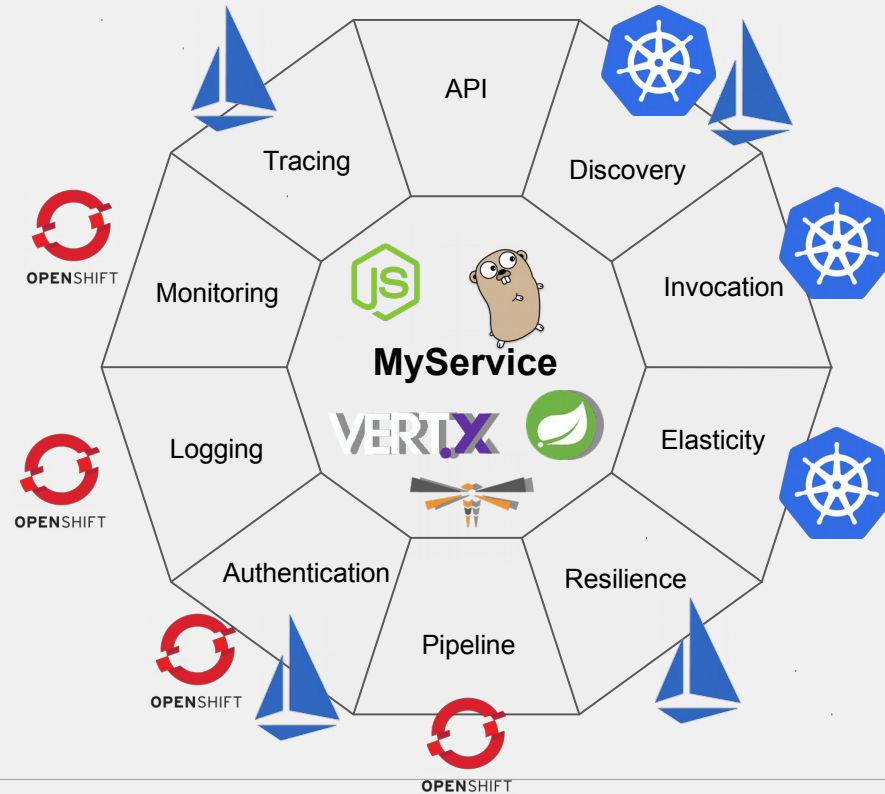
- A utility container in the same pod to enhance the main container's functionality
- Share the same network and lifecycle
- Istio uses an Istio Proxy (L7 Proxy) sidecar to proxy all network traffic between apps



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



MICROSERVICES'ILITIES + ISTIO



Polyglot Microservices Platform circa 2018



tutorial

Search: 10/19 Applications

Overview

Applications

Pods

Resources

Storage

Monitoring

Config

customer <http://customer.tutorial.192.168.99.100.nginx.io/>

customer/customer-#1 🔄 ⌵ ⌶

preference

preference/v1-#1 🔄 ⌵ ⌶

recommendation

recommendation/v1-#1 🔄 ⌵ ⌶

recommendation/v2-#1 🔄 ⌵ ⌶



OPENSIFT SERVICE MESH

- Supported distribution of
 - Istio
 - Jaeger
 - Kiali
 - Prometheus
 - Grafana
- Upstream project called Maistra
<https://github.com/Maistra>
- Integrated with Red Hat OpenShift Application Runtimes (RHOAR)

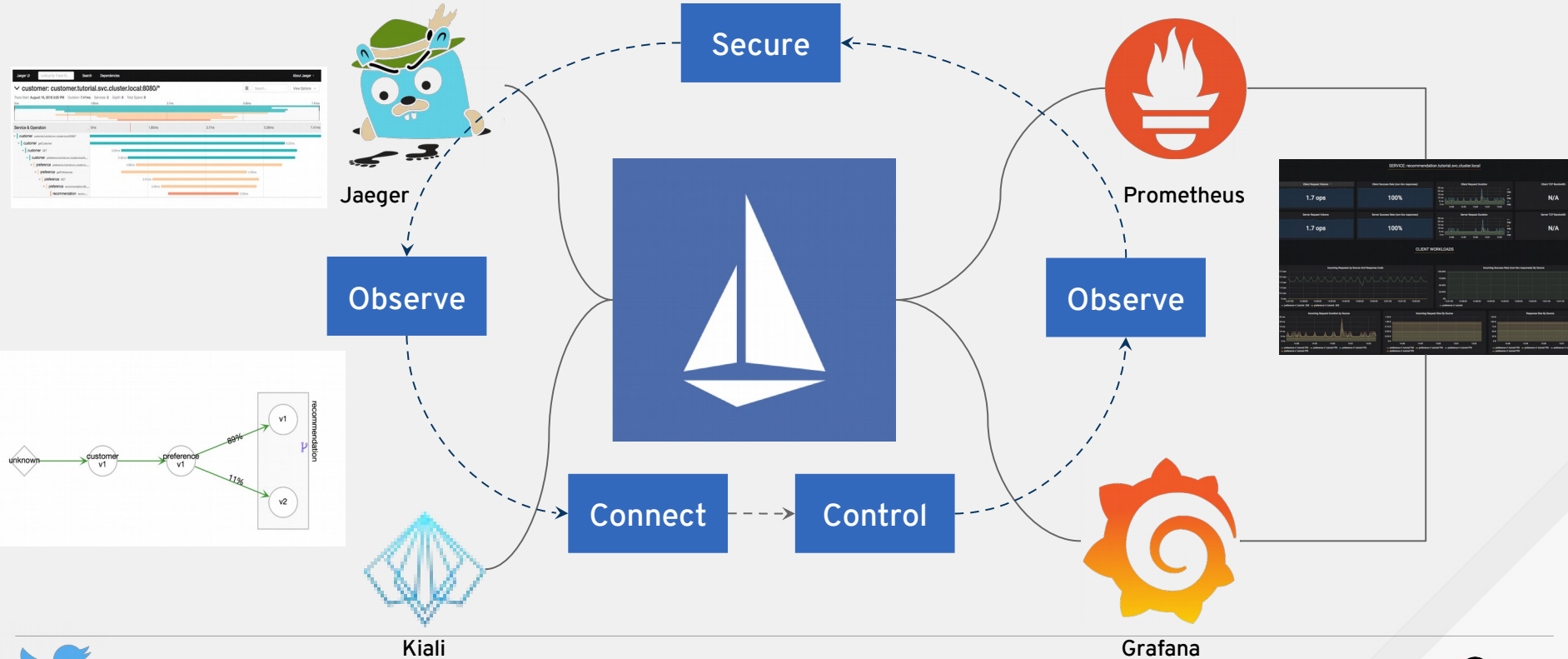
Select a mission

A specification that describes what your application will do.

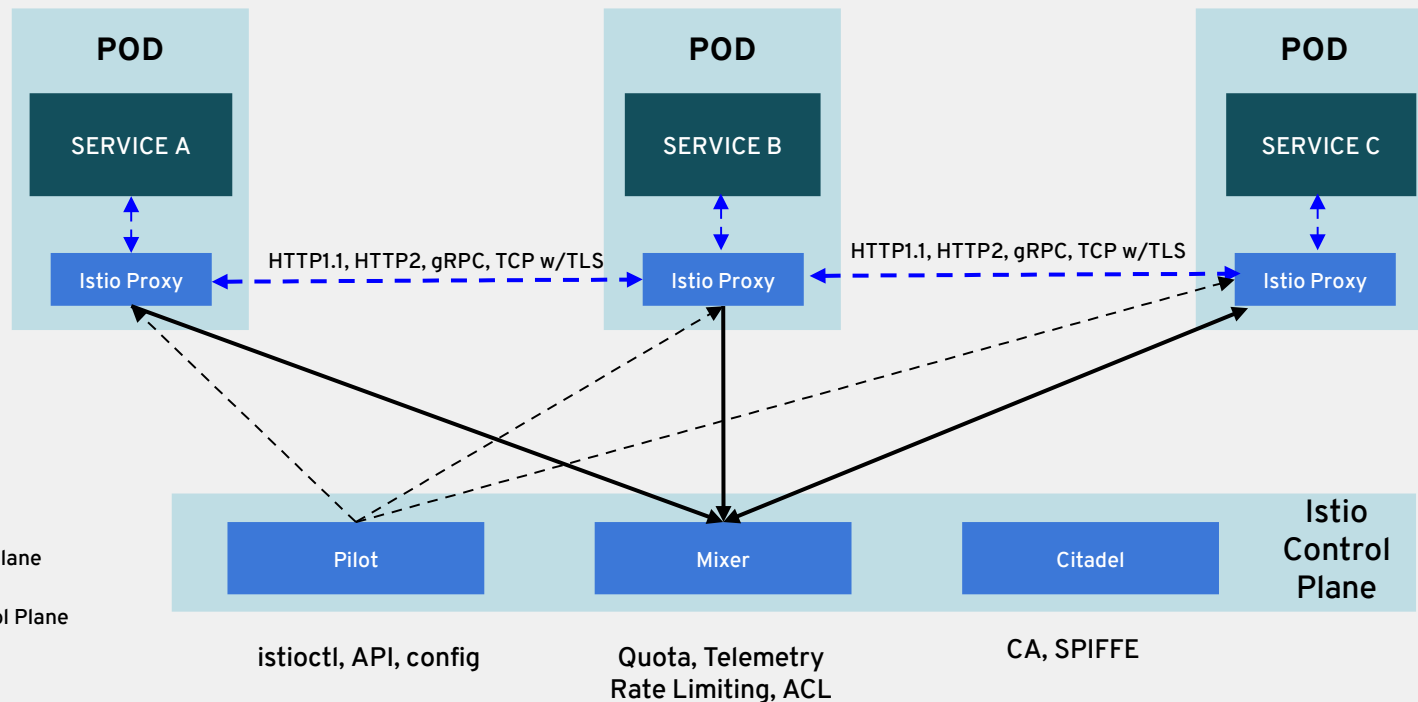
- Istio - Circuit Breaker**
The Istio Circuit Breaker mission demonstrates limiting access to... [More](#)
- Istio - Security**
The Istio Security mission demonstrates how Istio secures communi... [More](#)
- Istio - Distributed Tracing**
The Istio Distributed Tracing mission demonstrates how simple dis... [More](#)



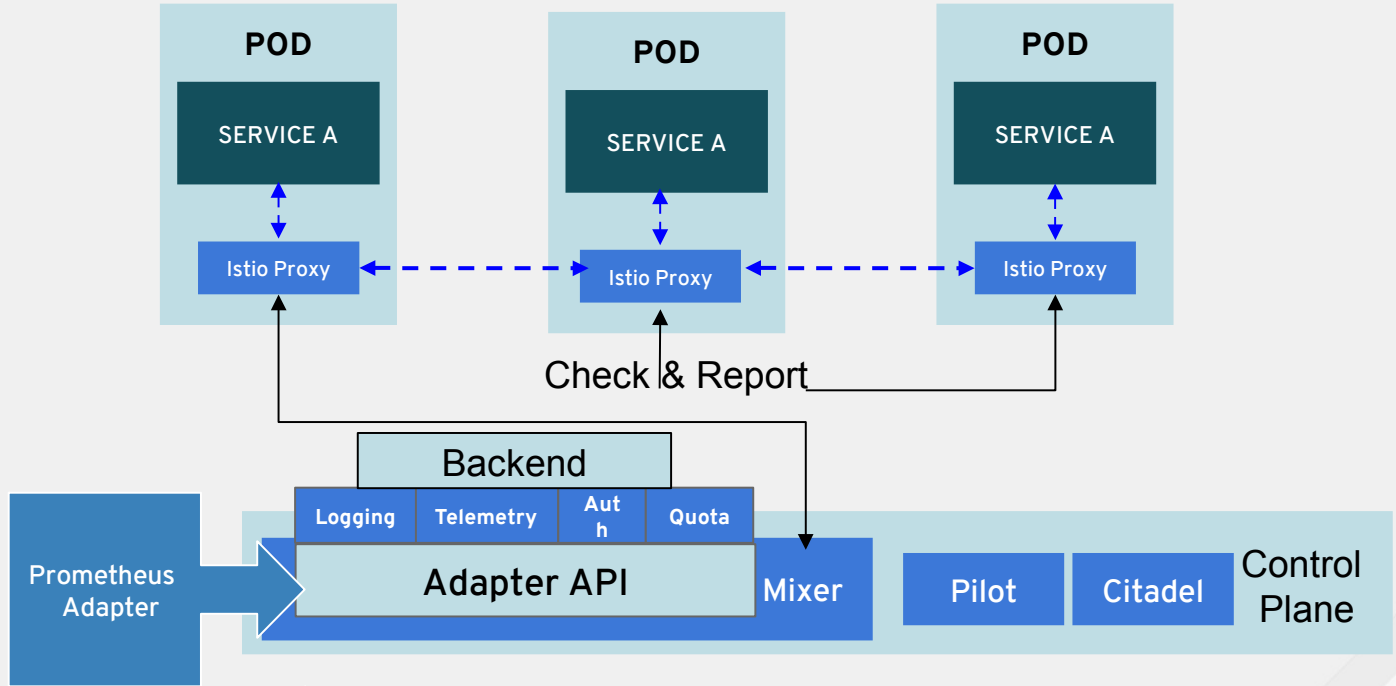
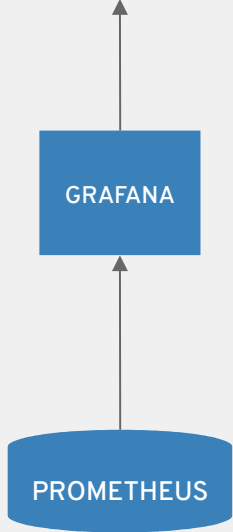
SERVICE MESH ECOSYSTEM



Istio Service Mesh



ISTIO MONITORING



OPENSIFT SERVICE MESH TECH PREVIEW

TECH PREVIEW 1

- Available for OCP 3.10
- Renders the whole cluster as tech preview
- Installation with Service Mesh Operator

LIMITATIONS

- Works on a single cluster where all deployments are on OpenShift
- Works with OpenShift SDN with no external providers

HOW TO GET ACCESS

- Install Istio Operator
- Create an Istio Installation object
- Scheduled to be available on CDK at the end of the year



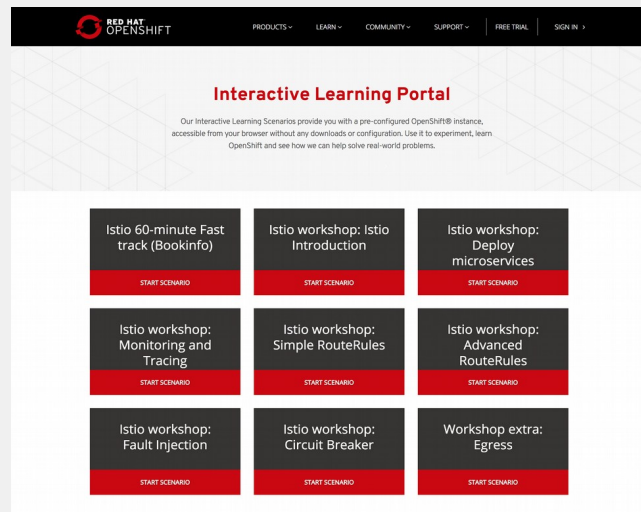
Homeworks

Self-paced Labs

learn.openshift.com/servicemesh

GitHub

bit.ly/istio-tutorial



The screenshot displays the Red Hat OpenShift Interactive Learning Portal. At the top, there is a navigation bar with the OpenShift logo and links for PRODUCTS, LEARN, COMMUNITY, SUPPORT, FREE TRIAL, and SIGN IN. The main heading is "Interactive Learning Portal". Below this, a paragraph states: "Our Interactive Learning Scenarios provide you with a pre-configured OpenShift® Instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems." The main content area features a 3x3 grid of scenario cards, each with a title and a "START SCENARIO" button. The scenarios are:

- Istio 60-minute Fast track (Bookinfo)
- Istio workshop: Istio Introduction
- Istio workshop: Deploy microservices
- Istio workshop: Monitoring and Tracing
- Istio workshop: Simple RouteRules
- Istio workshop: Advanced RouteRules
- Istio workshop: Fault Injection
- Istio workshop: Circuit Breaker
- Workshop extra: Egress



OpenShift Roadmap

OpenShift Container Platform 3.10 (July)

- Kubernetes 1.10 and CRI-O option
- **Istio (Tech Preview)**

OpenShift Container Platform 4.0 (Dec/Jan)

- Kubernetes 1.12 and CRI-O default

Q2
CY2018

Q3
CY2018

Q4
CY2018

Q1
CY2019

OpenShift Container Platform 3.11 (Sept/Oct)

- Kubernetes 1.11 and CRI-O option

OpenShift Container Platform 4.1 (March)

- Kubernetes 1.13 and CRI-O default
- **Istio GA**





GRAZIE PER L'ATTENZIONE

Giuseppe Bonocore - Solution Architect

Ugo Landini - Solution Architect



#RedHatOSD