#RedHatOSD

RED HAT FORUMS

DA ZERO A CENTO CON RED HAT SERVICE MESH

Giuseppe Bonocore - Senior Solution Architect

November 20th 2019 - Roma
December 3rd 2019 - Milano

Red Hat

Cloud Native (CNCF)

/kloud nādiv/ :

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.[1]

1. CNCF Definition

Cloud Native (CNCF)

/kloud nādiv/ :

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, **service meshes**, microservices, immutable infrastructure, and declarative APIs exemplify this approach.[1]

1. CNCF Definition

Red Hat OpenShift

PRODUCTS LEARN COMMUNITY SUPPORT FREE TRIAL LOG IN

Red Hat OpenShift Service Mesh is now available: What you should know

SEPTEMBER 11, 2019 BY BRIAN "REDBEARD" HARRINGTON

Tweet Like Share 24

SPOILER ALERT

Today, Red Hat OpenShift Service Mesh is now available.

As Kubernetes and Linux-based infrastructure take hold in digitally transforming organizations, modern applications frequently run in a microservices architecture and therefore can have complex route requests from one service to another. With Red Hat OpenShift Service Mesh, we've gone beyond routing the requests between services and included tracing and visualization components that make deploying a service mesh more robust. The service mesh layer helps us simplify the connection, observability and ongoing management of every application deployed on Red Hat OpenShift, the industry's most

Search the blog

CATEGORIES

- Containers (120)
- CoreOS (11)
- Educators (14)
- Events (331)
- Istio (12)
- Kubernetes (152)
- News (483)
- Operator Framework (16)
- Products (782)
  - OpenShift Container Platform (488)
  - OpenShift Dedicated (162)
  - OpenShift Online (292)
  - OpenShift Origin (358)
  - OpenStack Platform (3)
  - Red Hat CoreOS (5)
  - Red Hat Quay Registry (5)
- Security (23)
- Serverless (5)
- Technologies (275)
  - .NET (18)
  - Java (68)

Red Hat

Red Hat OpenShift Service Mesh is now available: What you should know

SEPTEMBER 11, 2019 BY BRIAN "REDBEARD" HARRINGTON

Today, Red Hat OpenShift Service Mesh is now available.

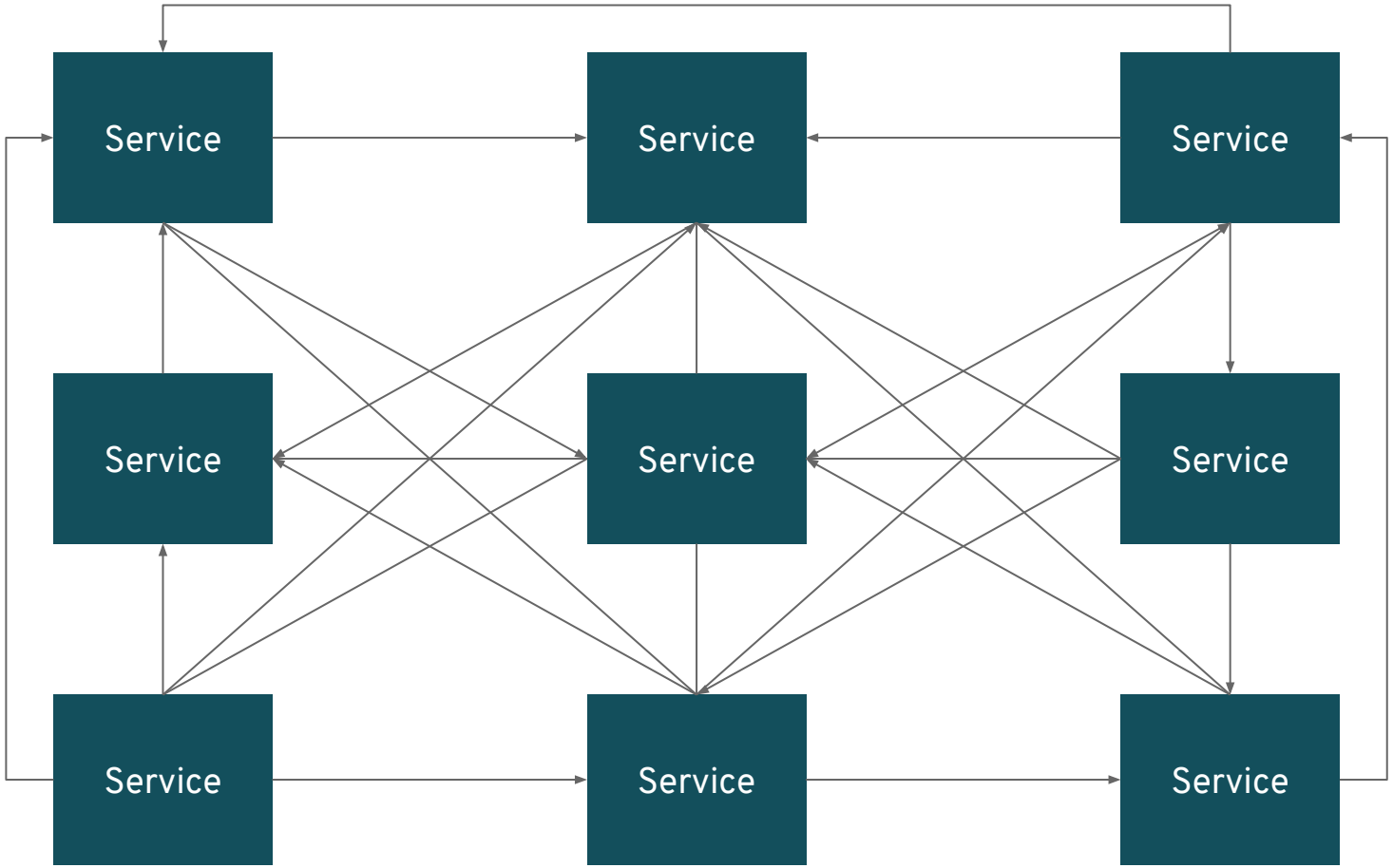
As Kubernetes and Linux-based infrastructure take hold in digitally transforming organizations, modern applications frequently run in a microservices architecture and therefore can have complex route requests from one service to another. With Red Hat OpenShift Service Mesh, we've gone beyond routing the requests between services and included tracing and visualization components that make deploying a service mesh more robust. The service mesh layer helps us simplify the connection, observability and ongoing management of every application deployed on Red Hat OpenShift, the industry's most

Search the blog

CATEGORIES

Containers (120)
CoreOS (11)
Educators (14)
Events (331)
Istio (12)
Kubernetes (152)
News (483)
OpenShift Commons (181)
OpenShift Container Engine (7)
OpenShift Ecosystem (333)
Operator Framework (16)
Products (782)
OpenShift Container Platform (488)
OpenShift Dedicated (162)
OpenShift Online (292)
OpenShift Origin (358)
OpenStack Platform (3)
Red Hat CoreOS (5)
Red Hat Quay Registry (5)
Security (23)
Serverless (5)
Technologies (275)
.NET (18)
Java (68)

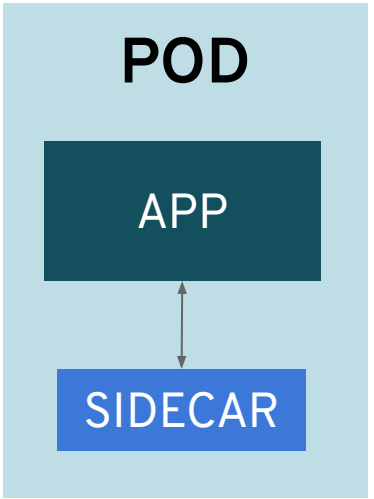
Distributed Architecture



Red Hat

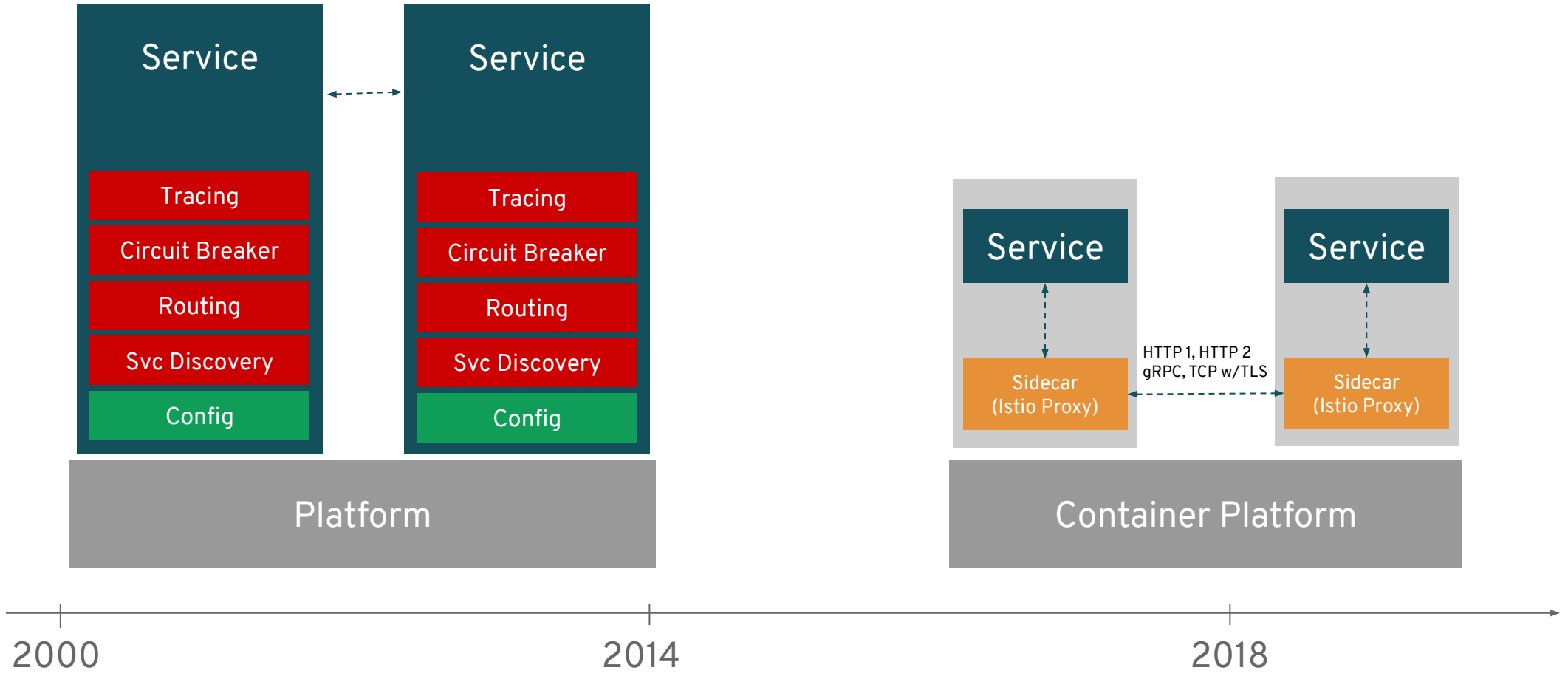
Sidecar Pattern

- 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



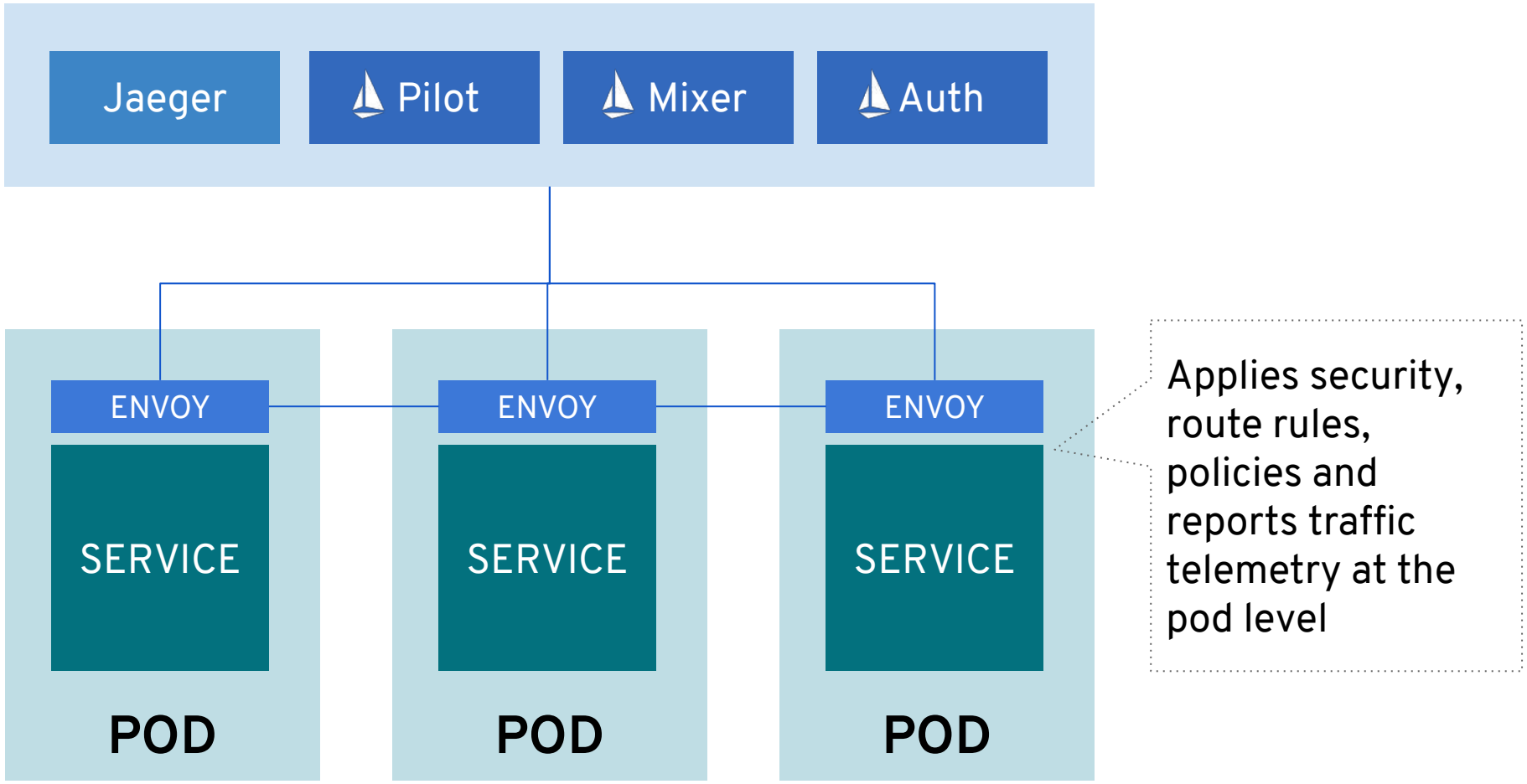
Source: http://blog.kubernetes.io/2015/06/the-distributed-system-toolkit-patterns.html

Evolution Of Services



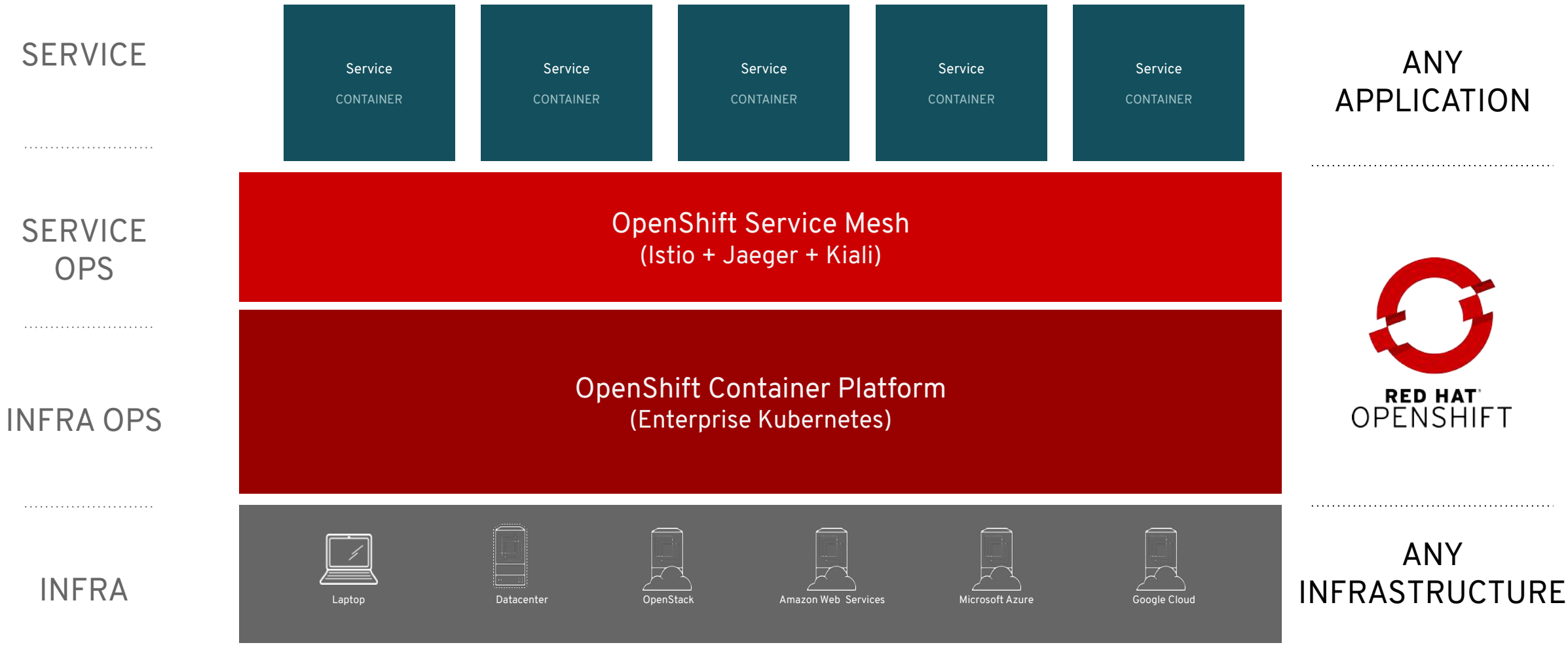
Red Hat

Service Mesh Architecture



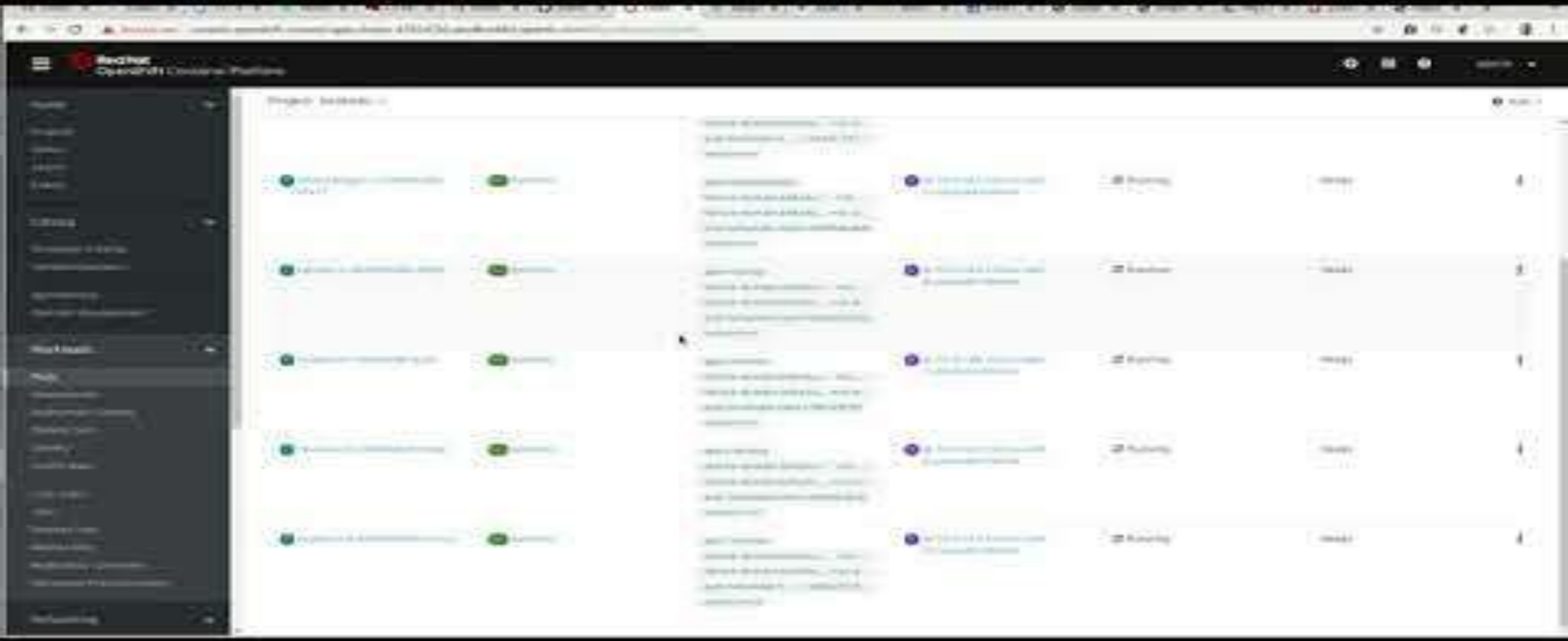
Red Hat

OpenShift Service Mesh



Red Hat

DEMO
ISTIO COMPONENTS

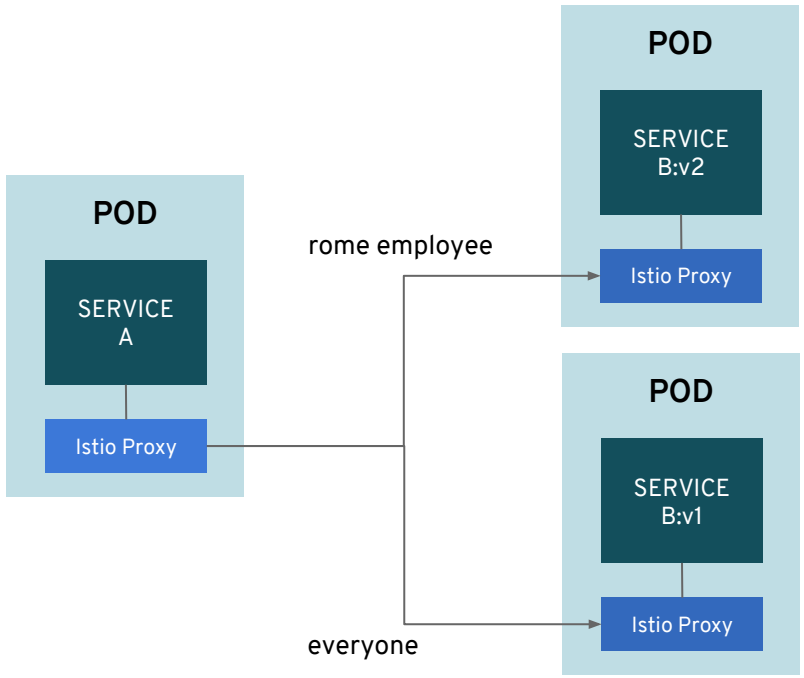


Traffic Management

Wire microservices at runtime, dynamically using blue/green deployment and canary testing.

Picture: https://www.flickr.com/photos/somethingness/

Traffic Management



By wiring software components at runtime, dynamically the Administrator can implement advanced deployment techniques such as blue/green or canary deployments.

The reliability of the overall application can be asserted by using fault injection or delays.

The application reliability can be improved using the Circuit Breaker pattern.

Red Hat

OBSERVABILITY

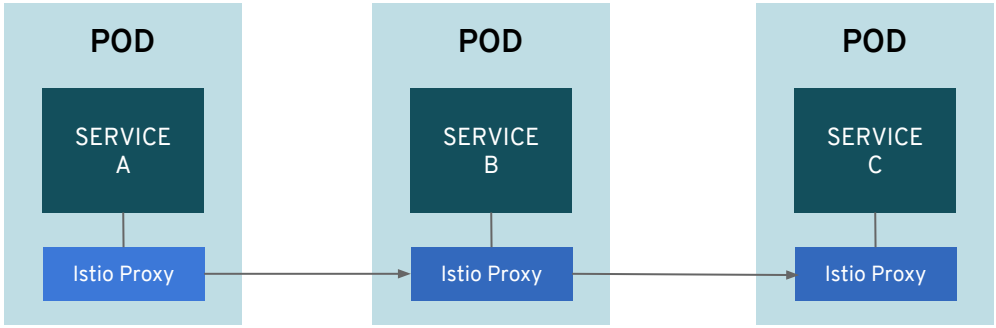
See what's happening, with rich automatic tracing, monitoring, and logging of all your services.

Picture: https://www.flickr.com/photos/goincase/

Observability

By having a proper description of every component (Kubernetes) and of every communication (Istio), any Administrator can jump in a project and immediately discover the application architecture.

By having all logs, traces and performance metrics gathered from all software components, the Administrator can discover the inner working of a running application.



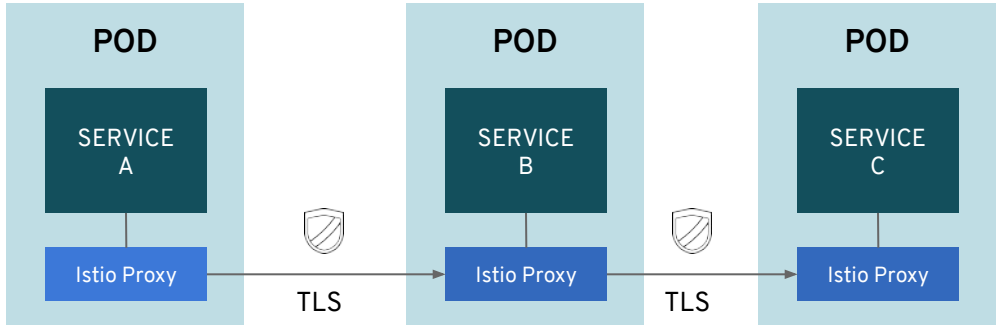
Red Hat

SERVICE IDENTITY & SECURITY

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

Picture: https://www.flickr.com/photos/drbillydude/

Service identity and Security



All the software components of an application can talk to each other securely, thanks to :

- **Security by default:** no changes needed for application code and infrastructure
- **Defense in depth:** integrate with existing security systems to provide multiple layers of defense
- **Zero-trust network:** build security solutions on untrusted networks

Red Hat

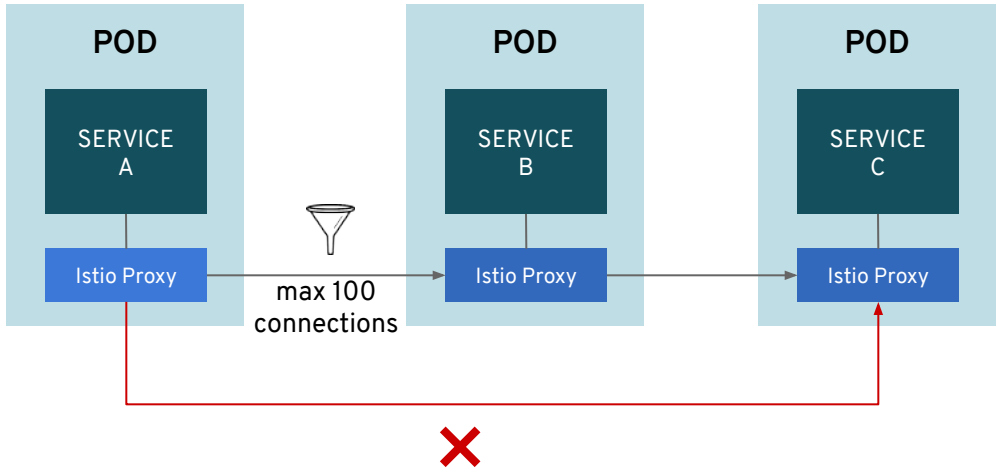
POLICY ENFORCEMENT

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

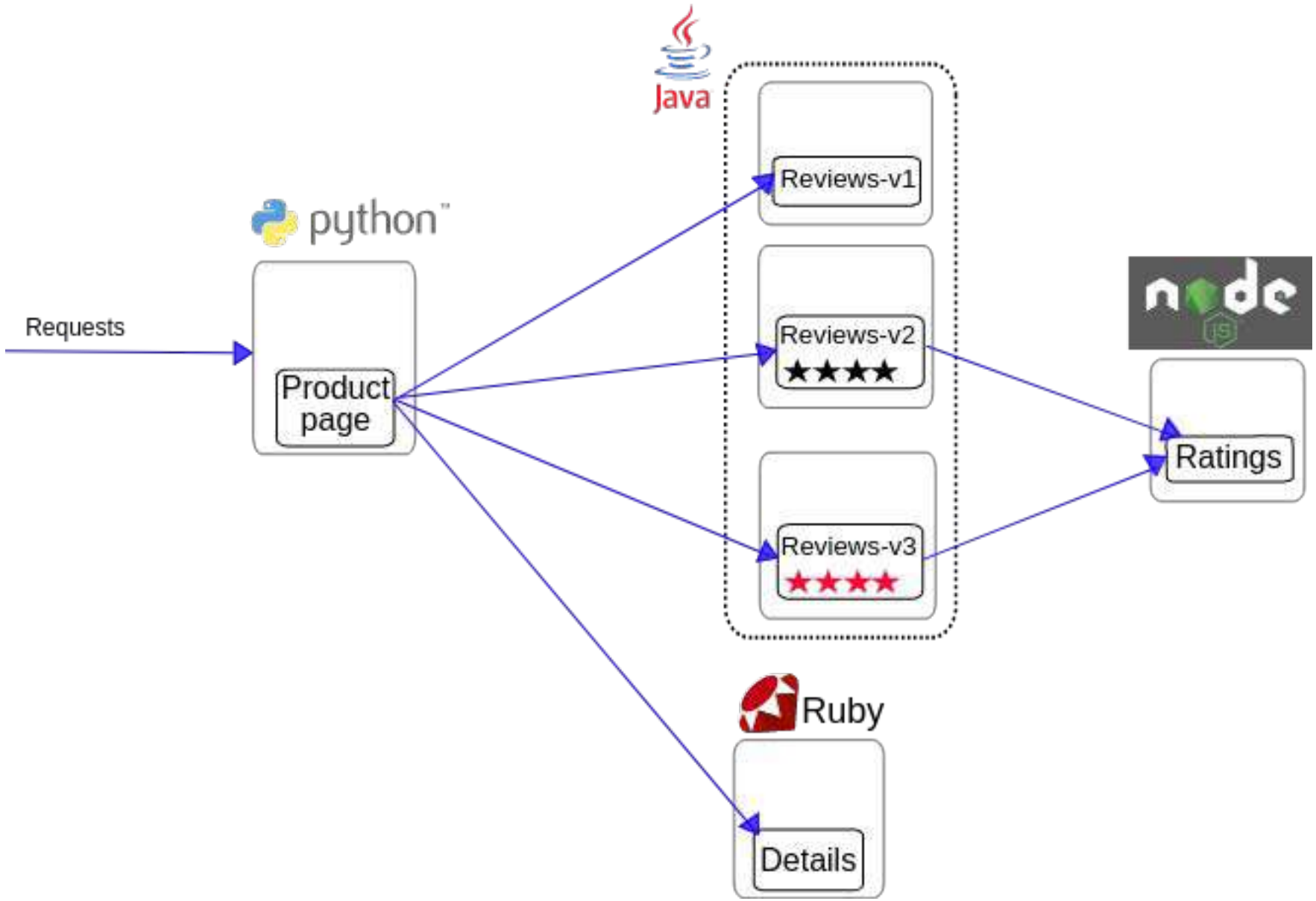
Picture: https://www.flickr.com/photos/velo4it/

Policy Enforcement

By having policies defined and applied dynamically, the Administrator can modify the application security and reliability at runtime.

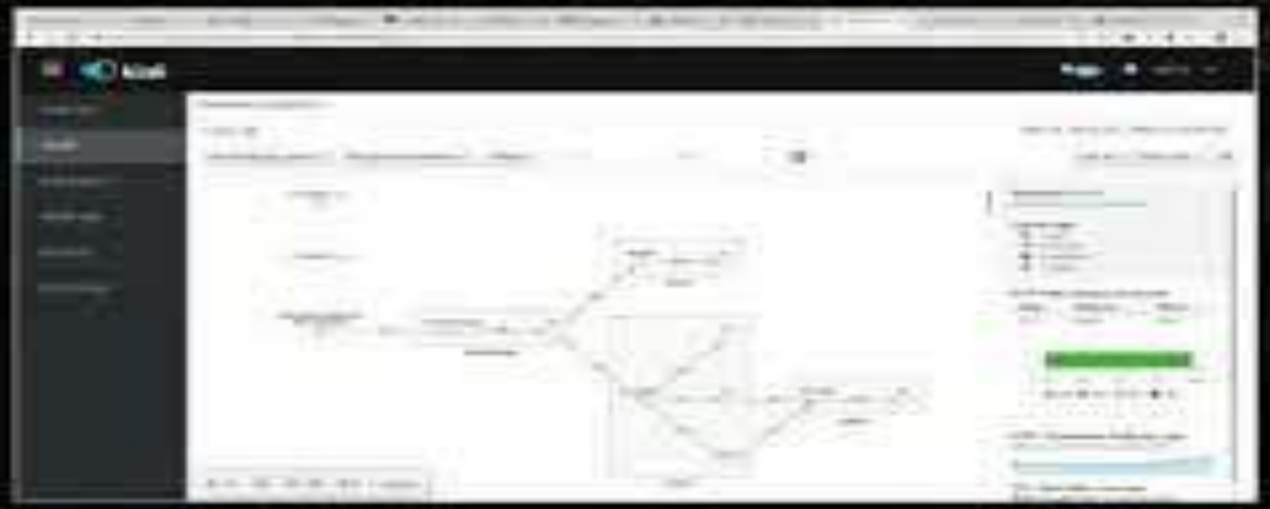


Red Hat


Java
Reviews-v1
python
Reviews-v2
Requests
Product page
Reviews-v3
Ratings
Ruby
Details

DEMO
ISTIO BOOKINFO

```
hosts:
- reviews
http:
- route:
  - destination:
      host: reviews
      subset: v2
    weight: 80
  - destination:
      host: reviews
      subset: v3
    weight: 20
[gloo@gloo@gpl:conda istio]$ oc apply -f ../virtual-service-reviews-80-20.yaml
virtualservice.networking.istio.io/reviews created
[gloo@gloo@gpl:conda istio]$
```



```
black
black
black
black
red
black
black
black
black
black
black
black
black
red
```

#RedHatOSD

RED HAT FORUMS

THANK YOU


linkedin.com/company/Red-Hat


facebook.com/RedHatinc


youtube.com/user/RedHatVideos


twitter.com/RedHat

Red Hat