

# Effective Servicemesh with Istio



Kamesh Sampath, Director of Developer Experience Red Hat

[bit.ly/sail-into-cloud](https://bit.ly/sail-into-cloud)



[kamesh.sampath@hotmail.com](mailto:kamesh.sampath@hotmail.com)



@kamesh\_sampath



kameshsampath

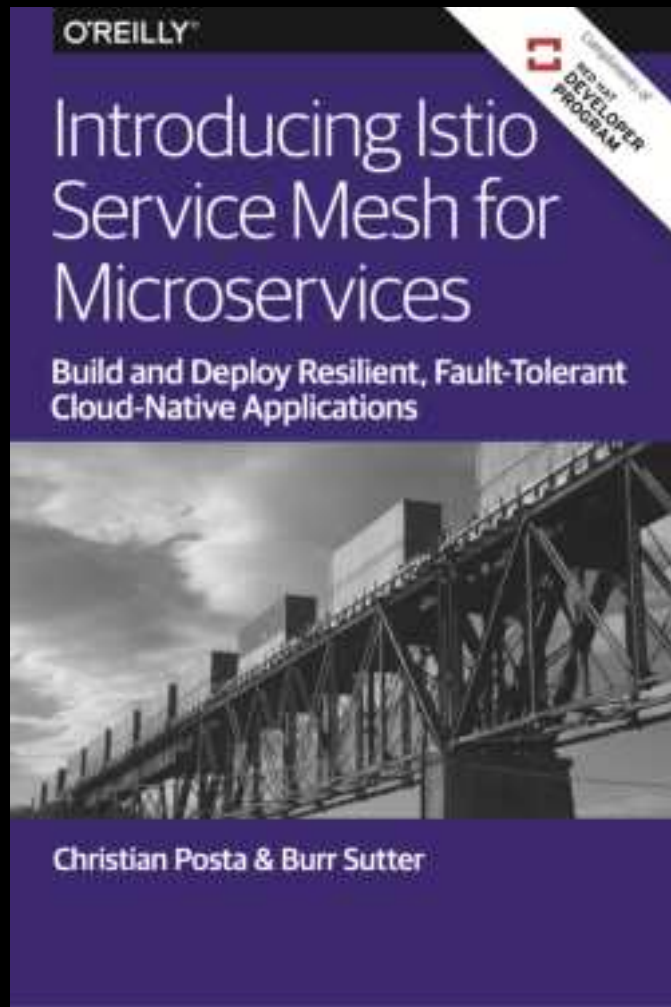


A photograph of a stone path winding through a dense forest of evergreen trees. The path is made of flat, light-colored stones and is surrounded by moss and fallen leaves. The trees are tall and dark green, creating a canopy overhead. The lighting is soft, suggesting a slightly overcast day.

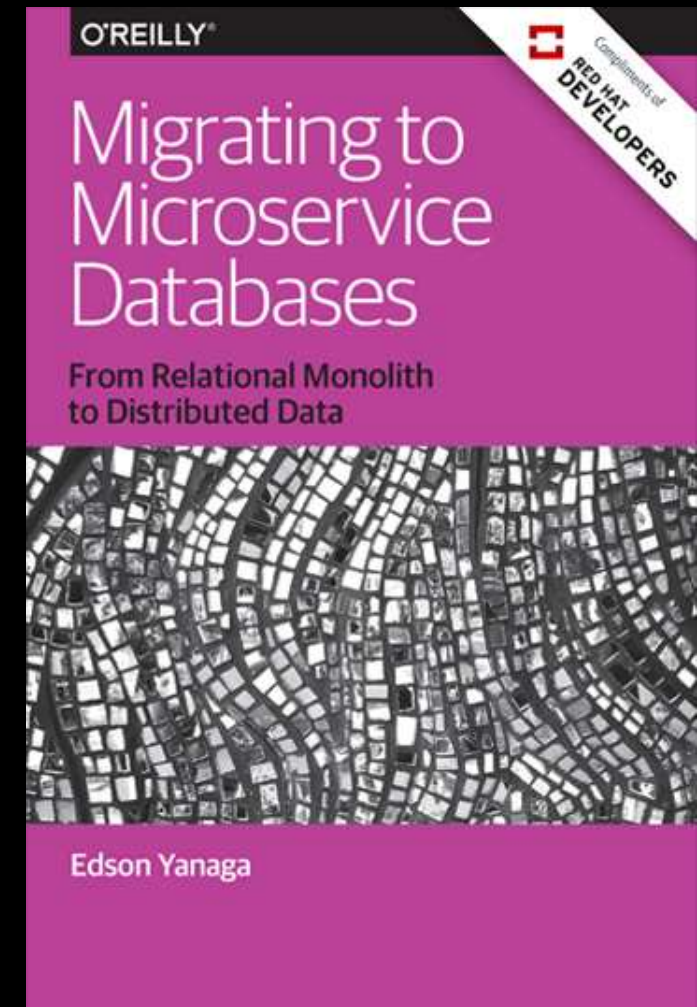
For developers  
who know it's  
about **the journey**,  
and the destination

[developers.redhat.com](https://developers.redhat.com)

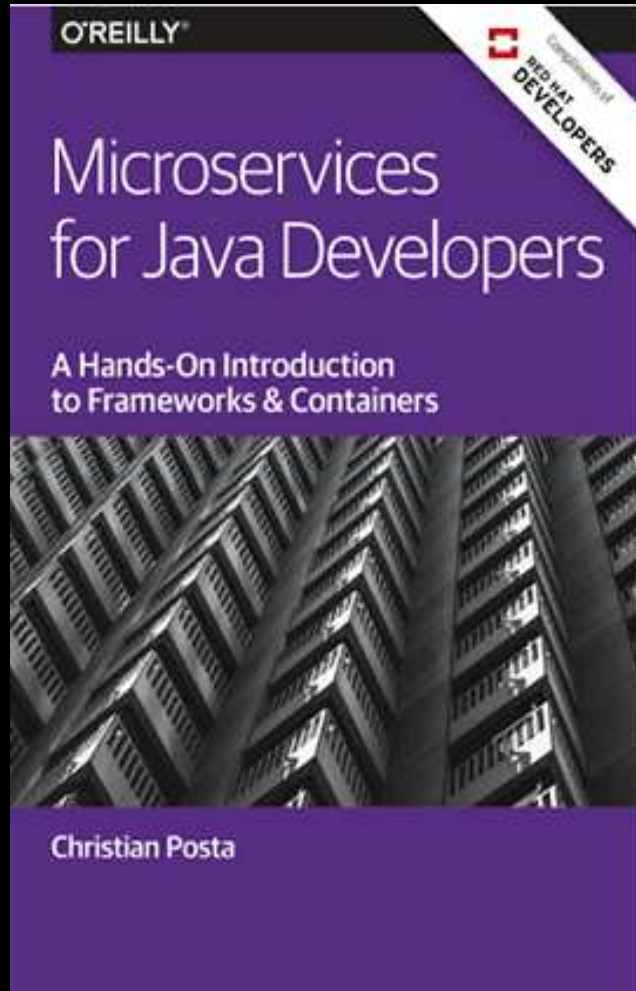
[bit.ly/istiobook](https://bit.ly/istiobook)



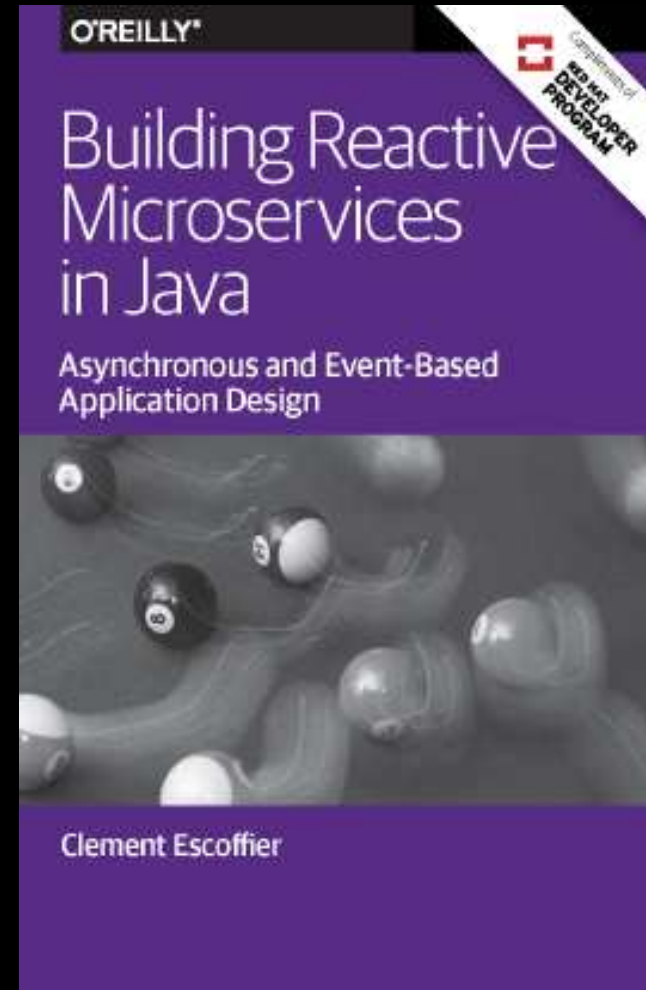
[bit.ly/mono2microdb](https://bit.ly/mono2microdb)



[bit.ly/javamicroservicesbook](https://bit.ly/javamicroservicesbook)



[bit.ly/reactivemicroservicesbook](https://bit.ly/reactivemicroservicesbook)



@kamesh\_sampath



kameshsampath

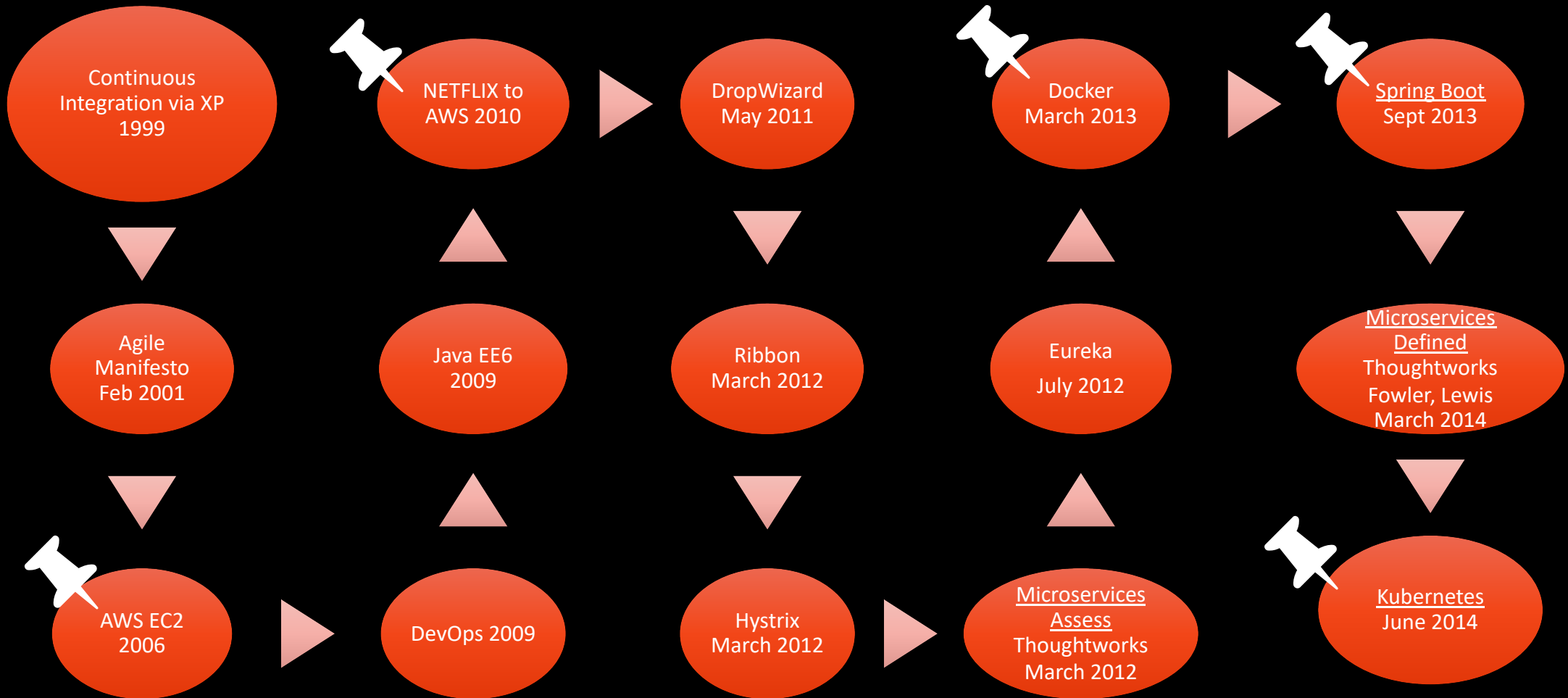


# What is a microservice?

The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.

These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

# Short History of Microservices



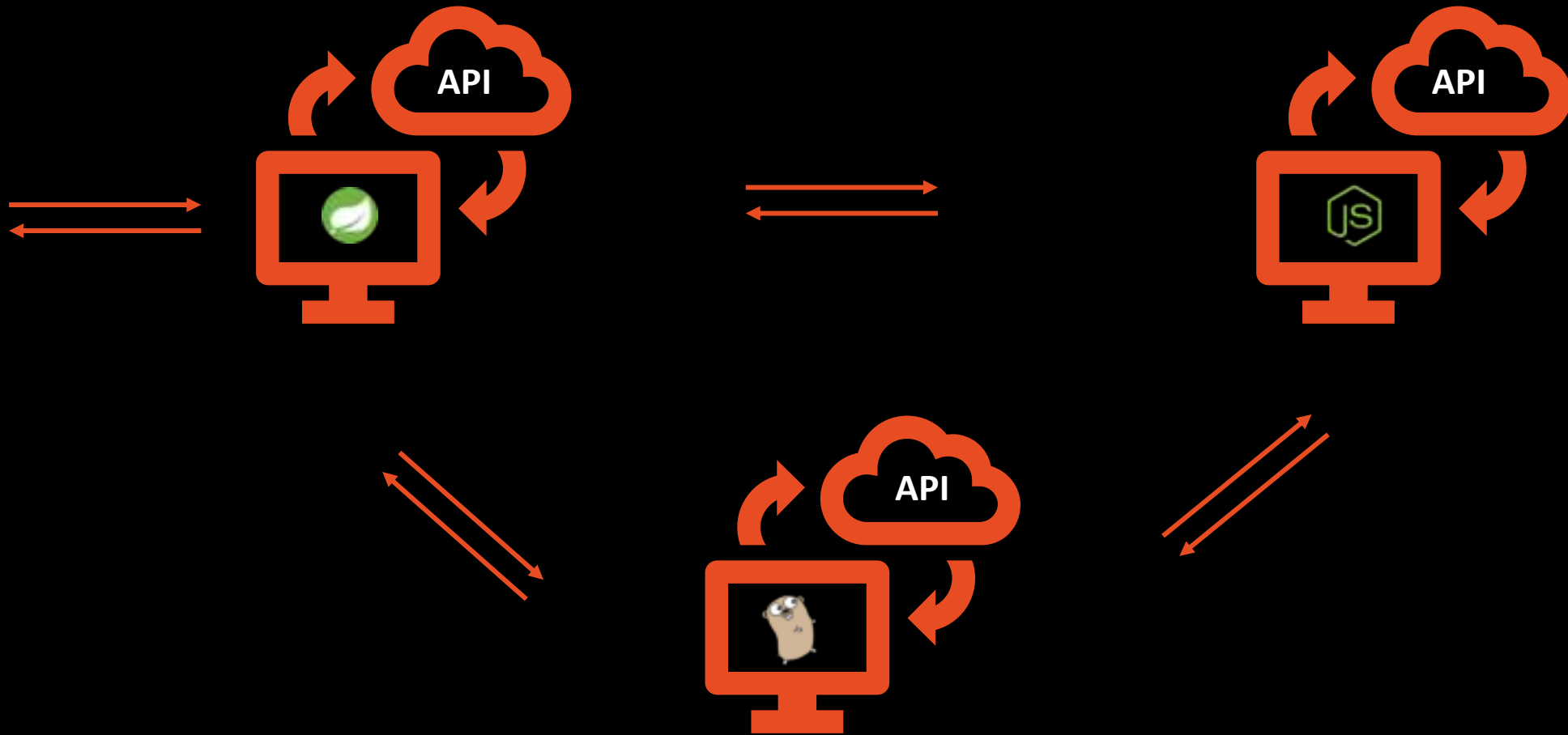
@kamesh\_sampath



kameshsampath



# Microservices == Distributed Computing



@kamesh\_sampath



kameshsampath







# Microservices'ilities



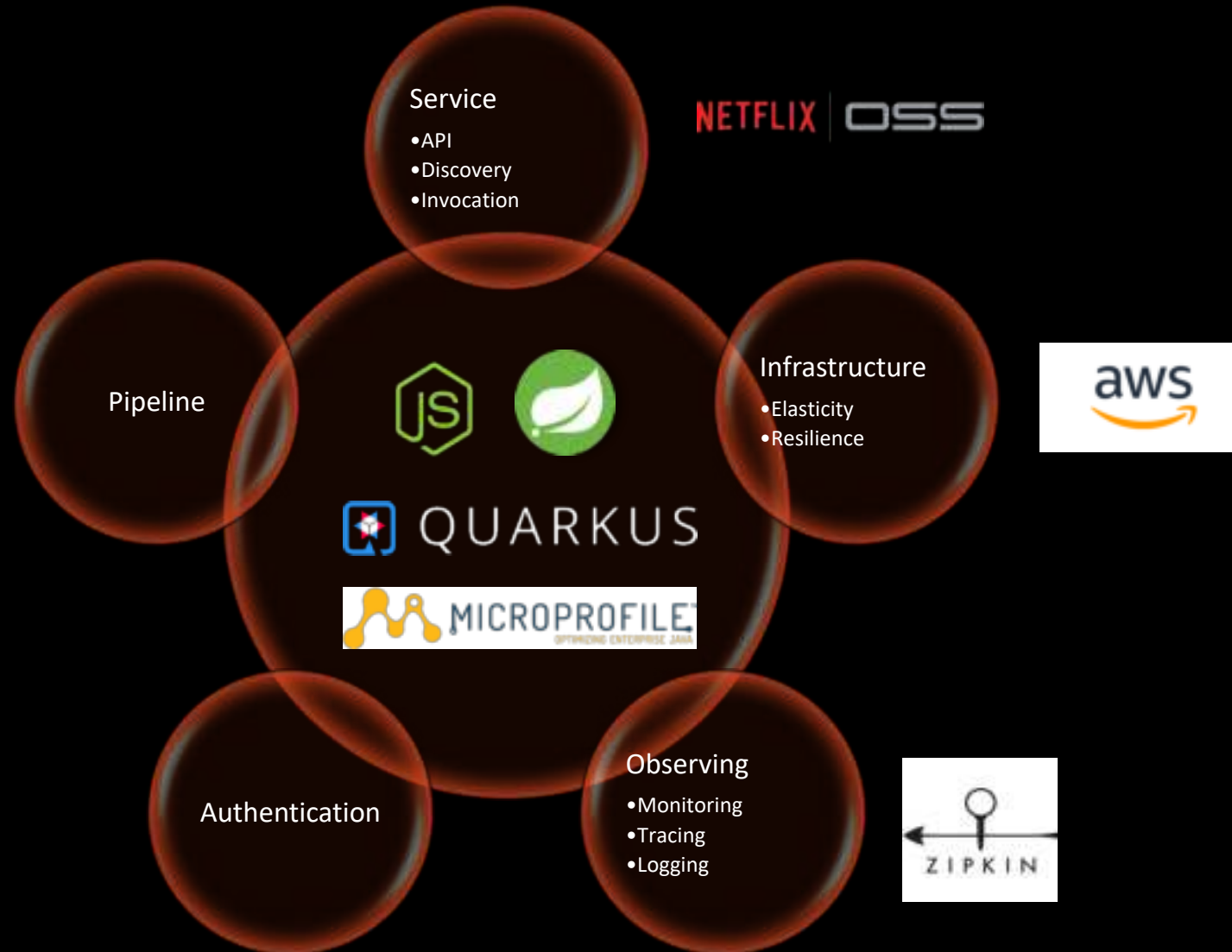
@kamesh\_sampath



kameshsampath



# Microservices'ilities



@kamesh\_sampath



kameshsampath



# Better Microservices Platform 2014-



Config Server



NETFLIX Ribbon



HYSTRIX  
DEFEND YOUR APP



@kamesh\_sampath



kameshsampath

# DevOps Challenges with Microservices

- How to scale?
- How to avoid port conflicts?
- How to manage them on multiple hosts?
- What happens if a host has trouble?
- How to keep them running?
- How to update them?
- Where are my containers?



**OPENSIFT**



@kamesh\_sampath

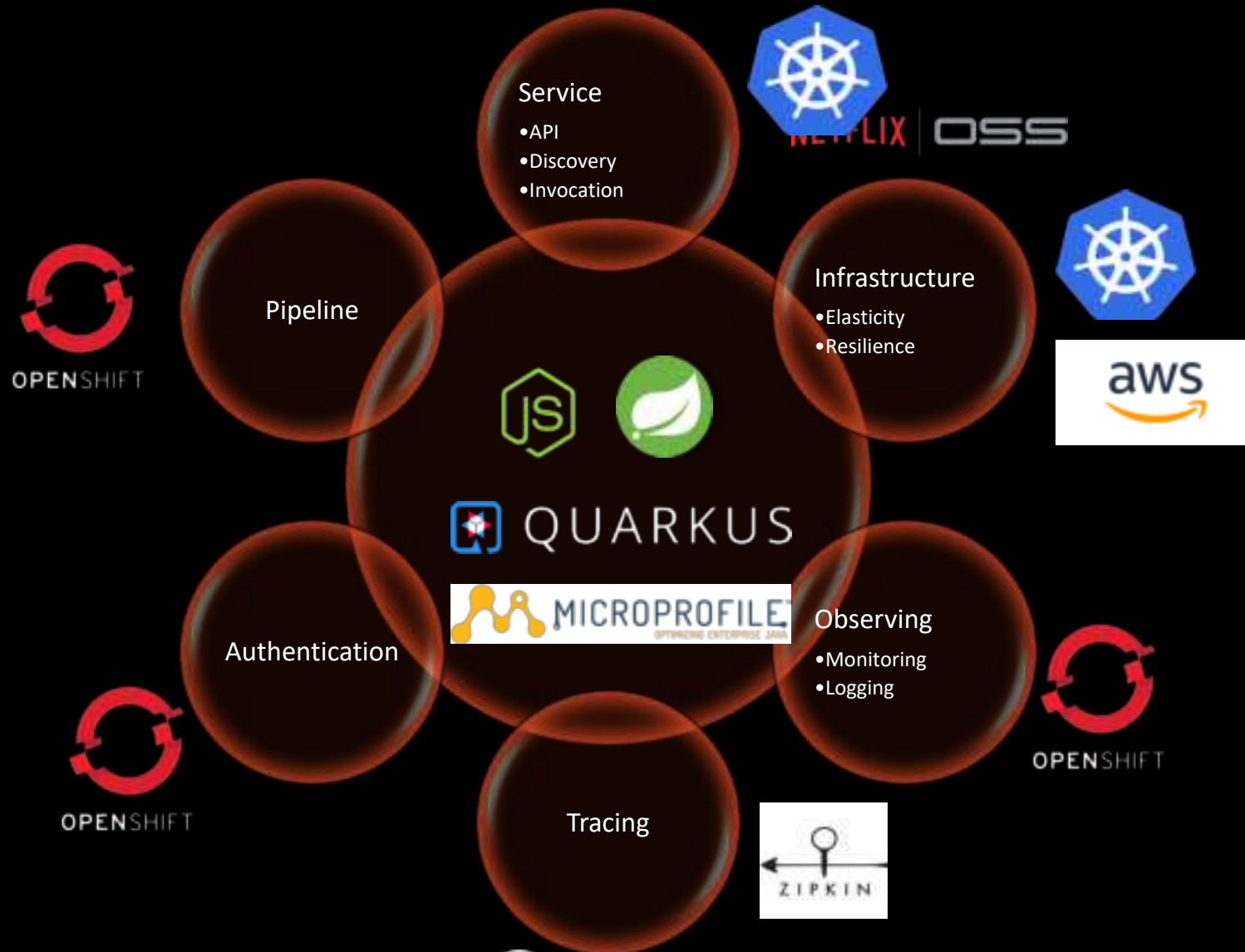


kameshsampath



**RED HAT**  
DEVELOPER

# Microservices'ilities



@kamesh\_sampath



kameshsampath



# Microservice(Yes) Pain Points

- Discovery
- Distributed Tracing
- Circuit Breakers
- Metrics and Monitoring
- Operational Requirements
  - A/B Testing
  - Canary Release
  - Rate Limiting
  - Access Policies



# Istio - Sail

(Kubernetes - Helmsman or ship's pilot)



@kamesh\_sampath



kameshsampath





# Microservices'ilities



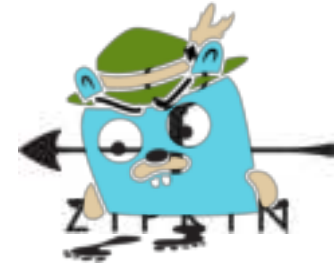
@kamesh\_sampath



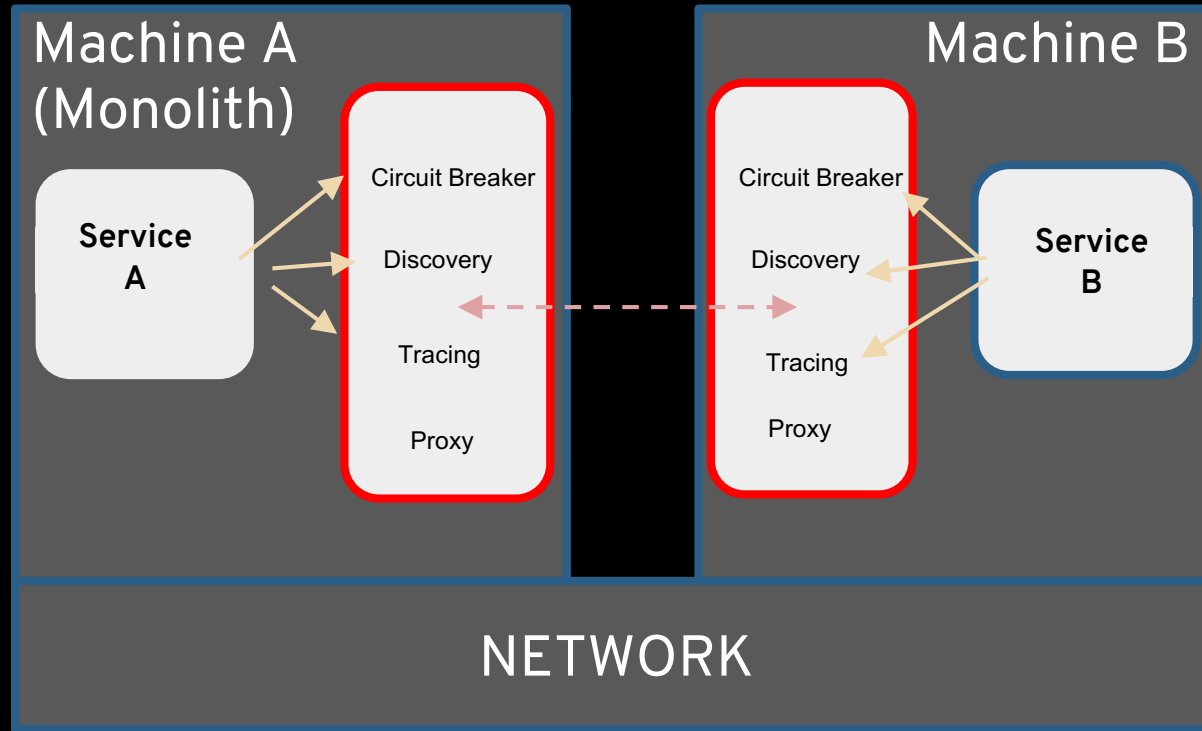
kameshsampath



# Better Microservices Platform 2018-



# What is a Servicemesh ?



A service mesh is a dedicated infrastructure layer for handling service-to-service communication. It's responsible for the reliable delivery of requests through the complex topology of services that comprise a modern, cloud native application. In practice, the service mesh is typically implemented as an array of lightweight network proxies that are deployed alongside application code, without the application needing to be aware.

<https://www.cncf.io/blog/2017/04/26/service-mesh-critical-component-cloud-native-stack/>



@kamesh\_sampath



kameshsampath





What's Wrong with Netflix OSS?

# Java Only

Adds a lot of libraries to YOUR code



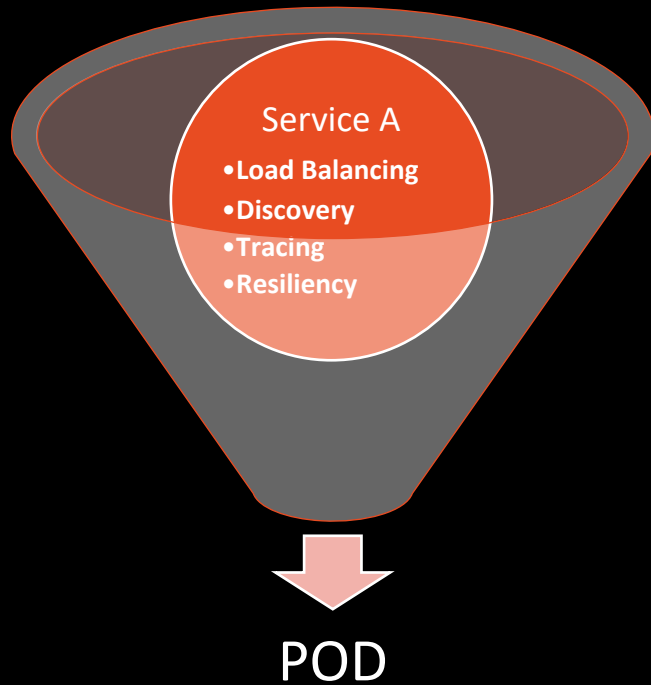
@kamesh\_sampath



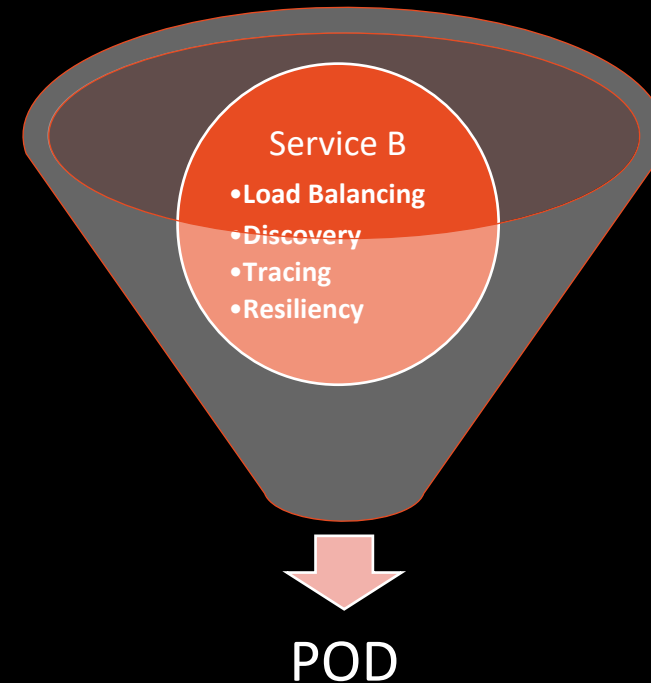
kameshsampath



# Microservices embedding Capabilities



Before **Istio**



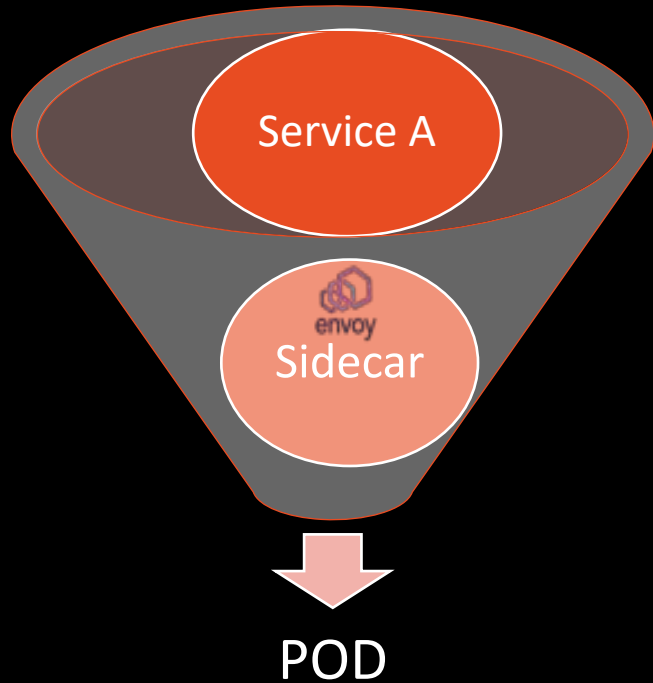
@kamesh\_sampath



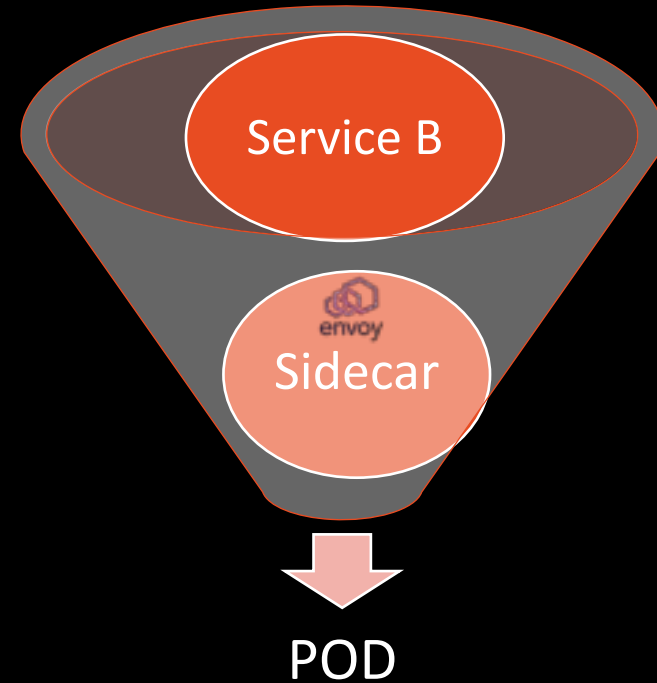
kameshsampath



# Microservices embedding Capabilities



After Istio

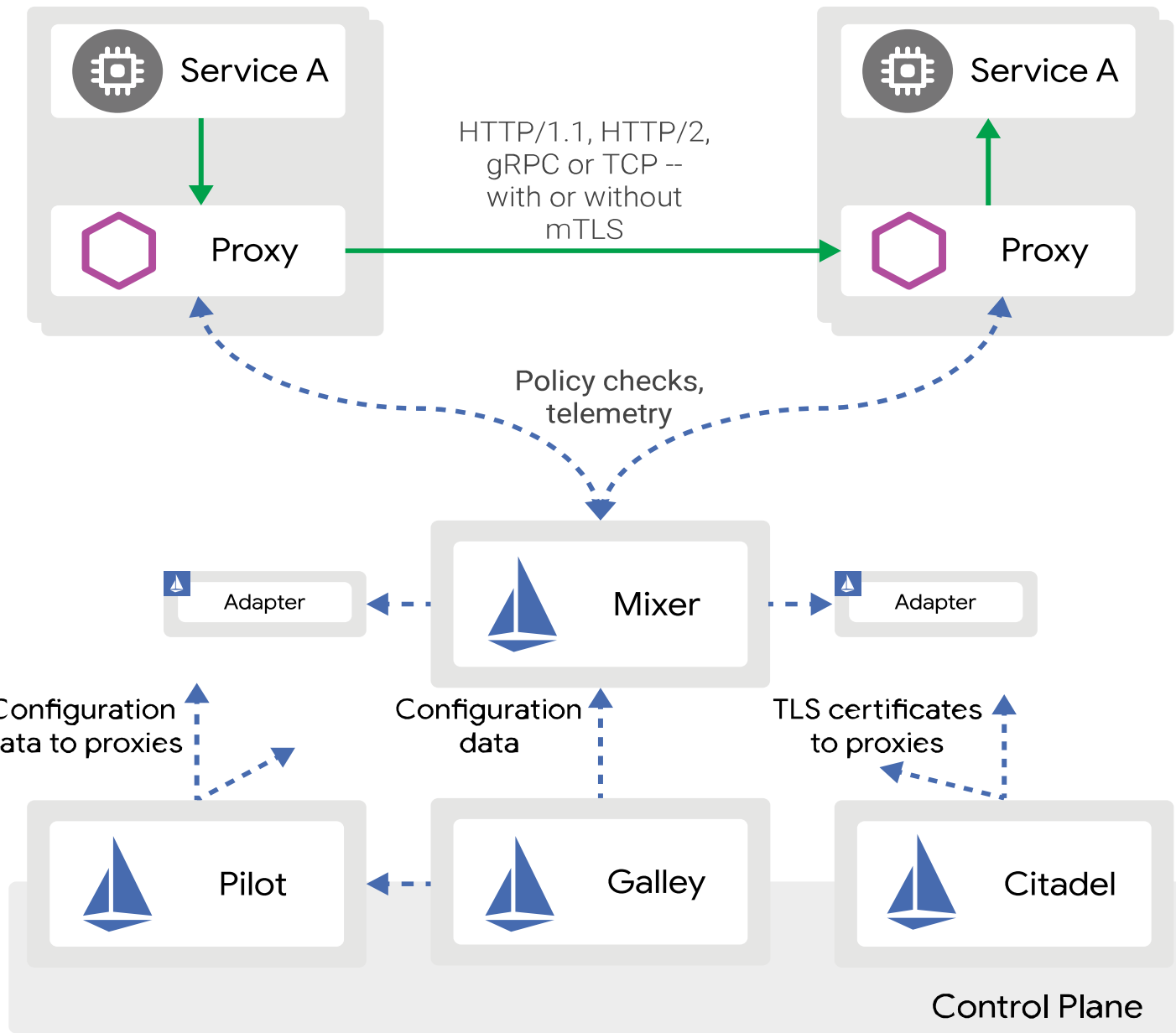


@kamesh\_sampath



kameshsampath





<https://istio.io/docs/concepts/what-is-istio/>

# Next Generation Microservices - Service Mesh

- Code Independent (Polyglot)
- Intelligent Routing and Load-Balancing
- A/B Tests
- Smarter Canary Releases
- Chaos: Fault Injection
- Resilience: Circuit Breakers
- Observability: Metrics and Tracing
- Fleet wide policy enforcement



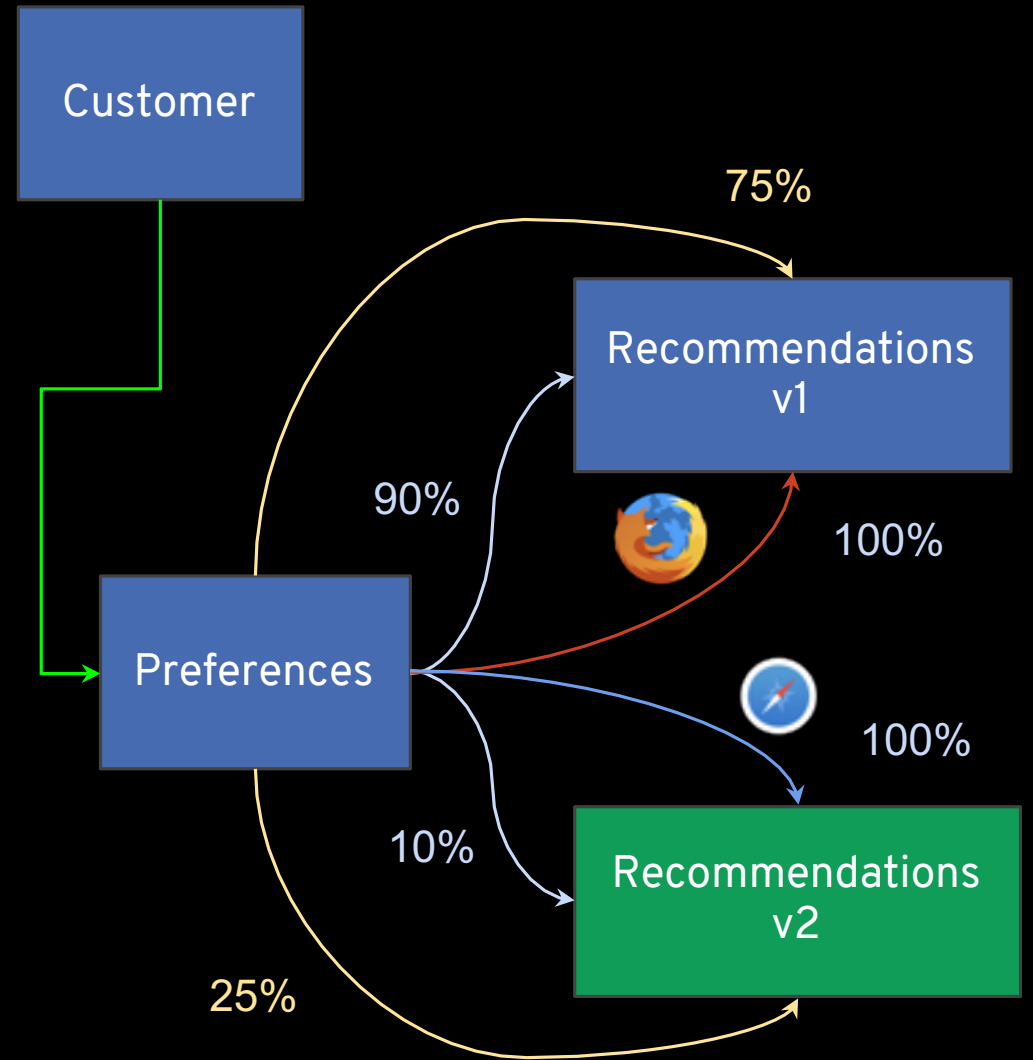
@kamesh\_sampath



kameshsampath



# Demo



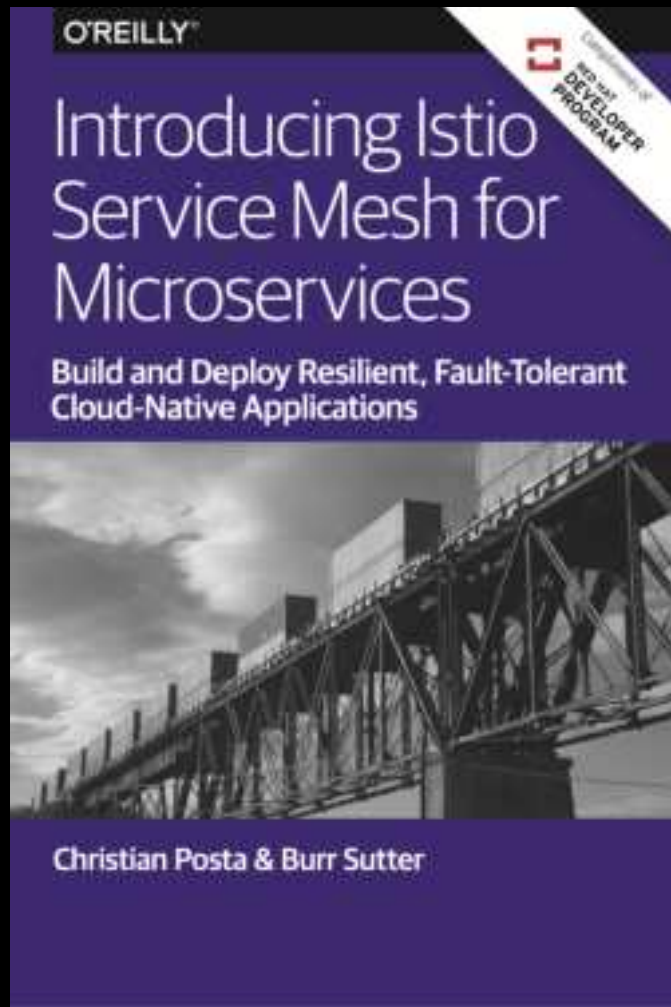
@kamesh\_sampath



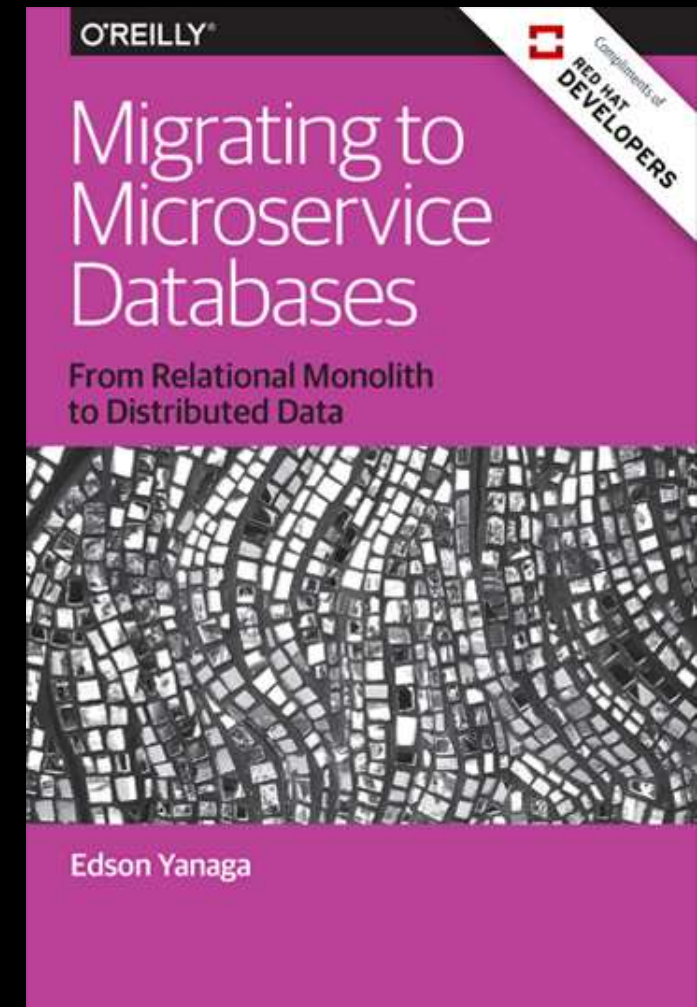
kameshsampath



[bit.ly/istiobook](https://bit.ly/istiobook)



[bit.ly/mono2microdb](https://bit.ly/mono2microdb)



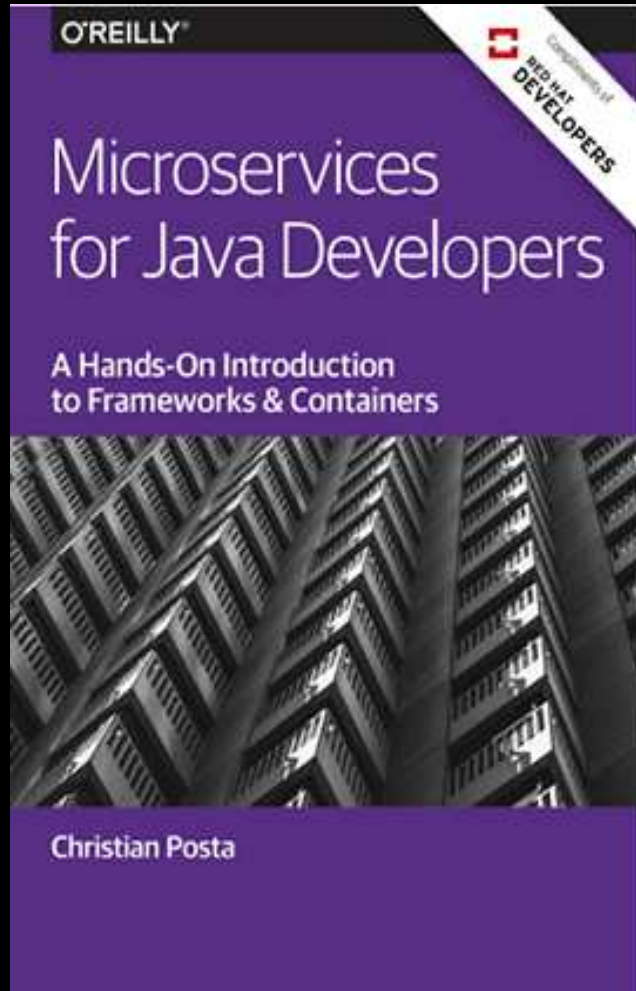
@kamesh\_sampath



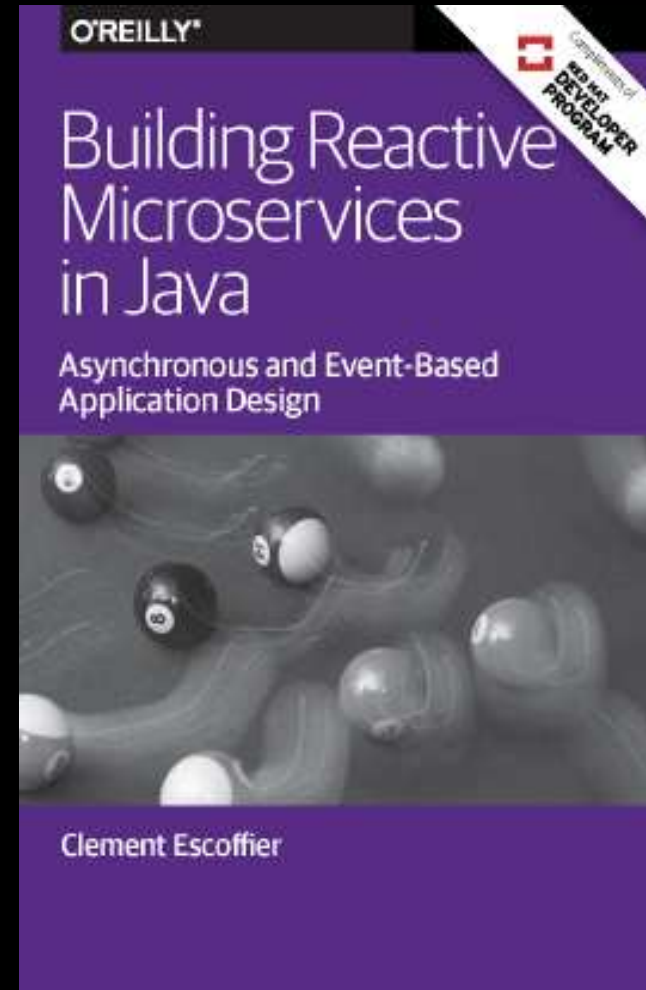
kameshsampath



[bit.ly/javamicroservicesbook](https://bit.ly/javamicroservicesbook)



[bit.ly/reactivemicroservicesbook](https://bit.ly/reactivemicroservicesbook)



@kamesh\_sampath



kameshsampath



# Resources

- Slides -- [bit.ly/sail-into-cloud](https://bit.ly/sail-into-cloud)
- Tutorials
  - Istio Tutorial - [bit.ly/istio-tutorial](https://bit.ly/istio-tutorial)
  - Knative Tutorial – [bit.ly/knative-tutorial](https://bit.ly/knative-tutorial)
  - Demo: [bit.ly/msa-instructions](https://bit.ly/msa-instructions)
  - Slides: [bit.ly/microservicesdeepdive](https://bit.ly/microservicesdeepdive)
- Video Training:
  - [bit.ly/microservicesvideo](https://bit.ly/microservicesvideo)
  - [Kubernetes for Java Developers](#)
  - [9 Steps to Awesome with Kubernetes](#)



@kamesh\_sampath



kameshsampath