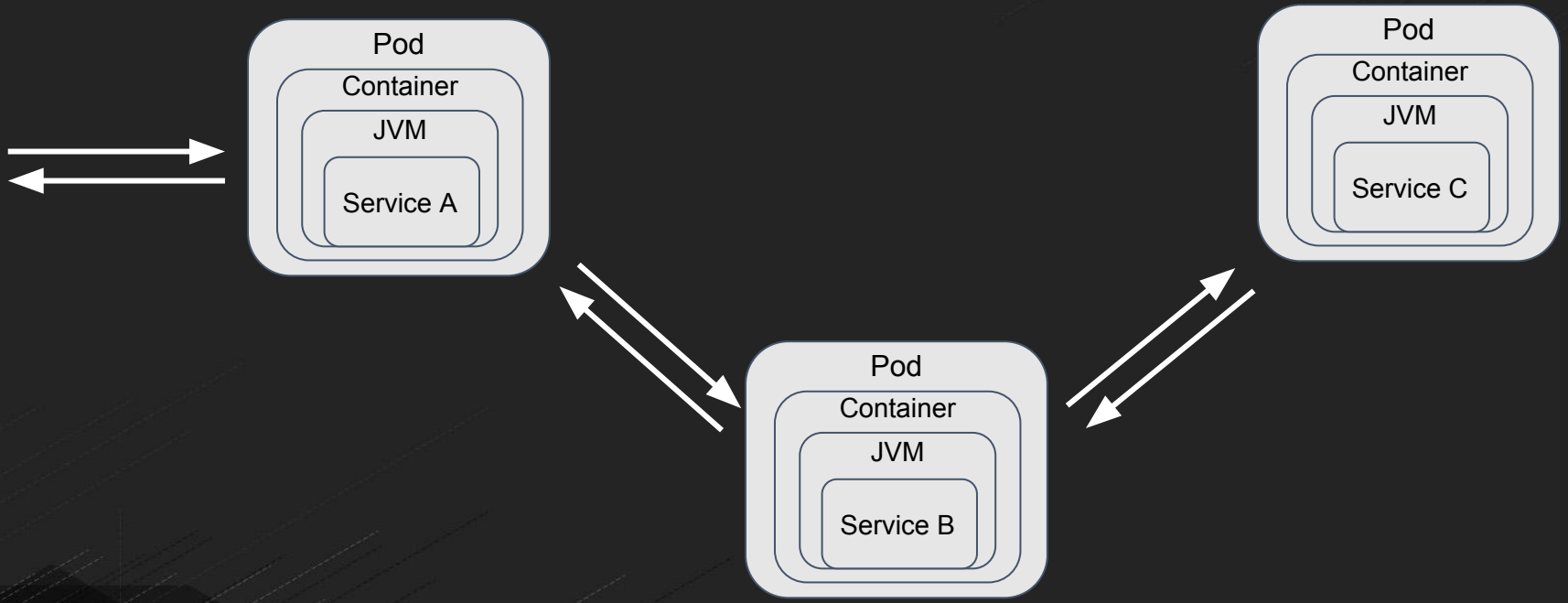
RED HAT®
DEVELOPER
PROGRAM

Microservices in repeatable demos

@roelhodzelmans roel@redhat.com

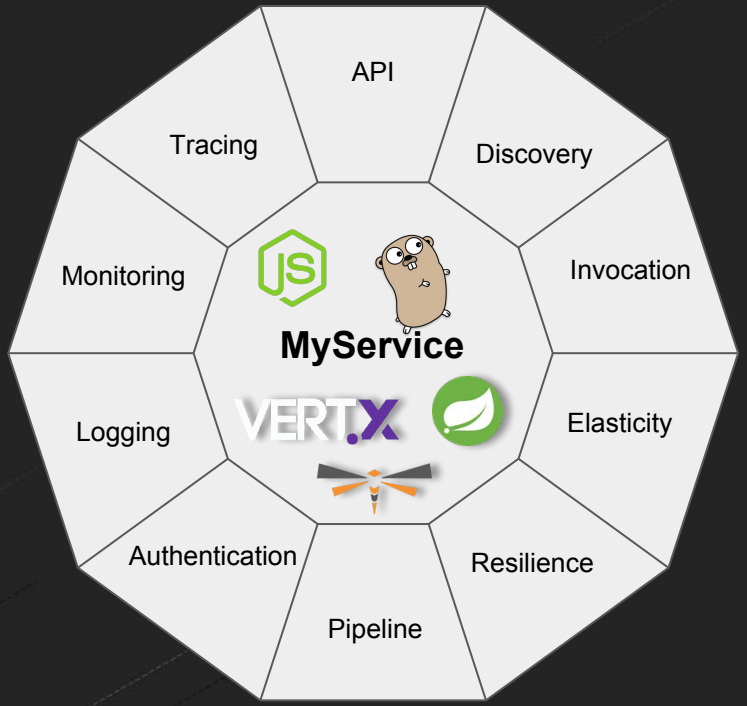
Microservices == Distributed Computing

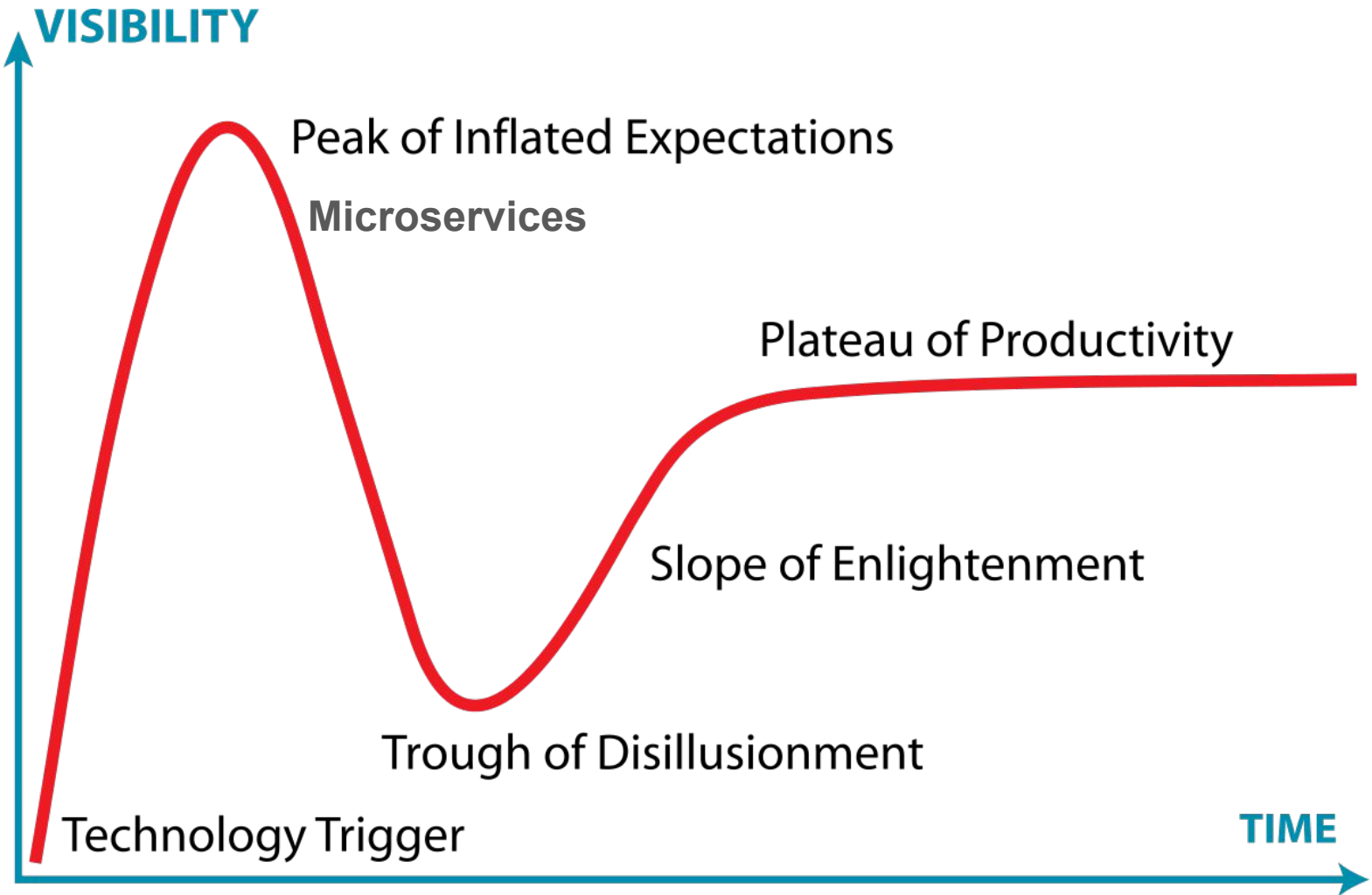


Microservice Principles/Characteristics

1. Deployment **Independence** - updates to an individual microservice have no negative impact to any other component of the system. Optimized for **Replacement**
2. Organized around **business** capabilities
3. **Products** not Projects
4. **API** Focused
5. **Smart** endpoints and dumb pipes
6. Decentralized Governance
7. Decentralized Data Management
8. Infrastructure Automation (infrastructure as code)
9. Design for failure
10. Evolutionary Design

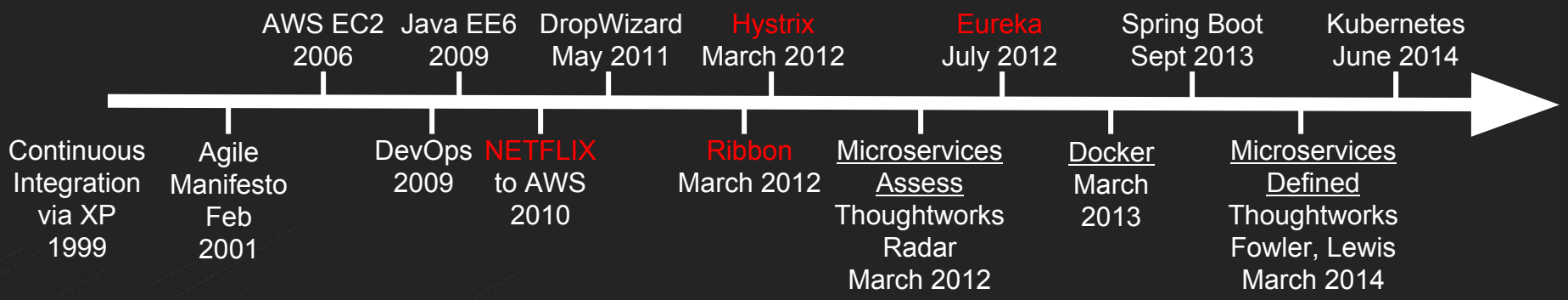
Responsibilities are hard




VISIBILITY
Peak of Inflated Expectations
Microservices
Plateau of Productivity
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
TIME

Overnight Sensation?

Short History of Microservices



NETFLIX | OSS

RED HAT DEVELOPERS

Gee thanks Kevin!

NETFLIX | OSS

February 2013

RED HAT DEVELOPERS

Awesomeness of 2012

Indoor Clouds

http://techland.time.com/2012/11/01/best-inventions-of-the-year-2012/

Google Glass

http://mashable.com/2012/11/30/top-25-tech-of-2012/#WVjQ0a_HBsqL

Windows Phone

http://mashable.com/2012/11/30/top-25-tech-of-2012/#WVjQ0a_HBsqL

Windows 8

http://mashable.com/2012/11/30/top-25-tech-of-2012/#WVjQ0a_HBsqL

Java Microservices (1.0) Platform circa 2014

CLOUDFOUNDRY

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

But progress didn’t stop in 2014

Kubernetes - Helmsman or ship’s pilot

Better Microservices (2.0) Platform circa 2016

Better Java/Node.js/Python/Go/etc. Platform

Running the demo

https://github.com/jbossdemocentral/coolstore-microservice

But progress didn’t stop in 2016 either!

The feedback was:

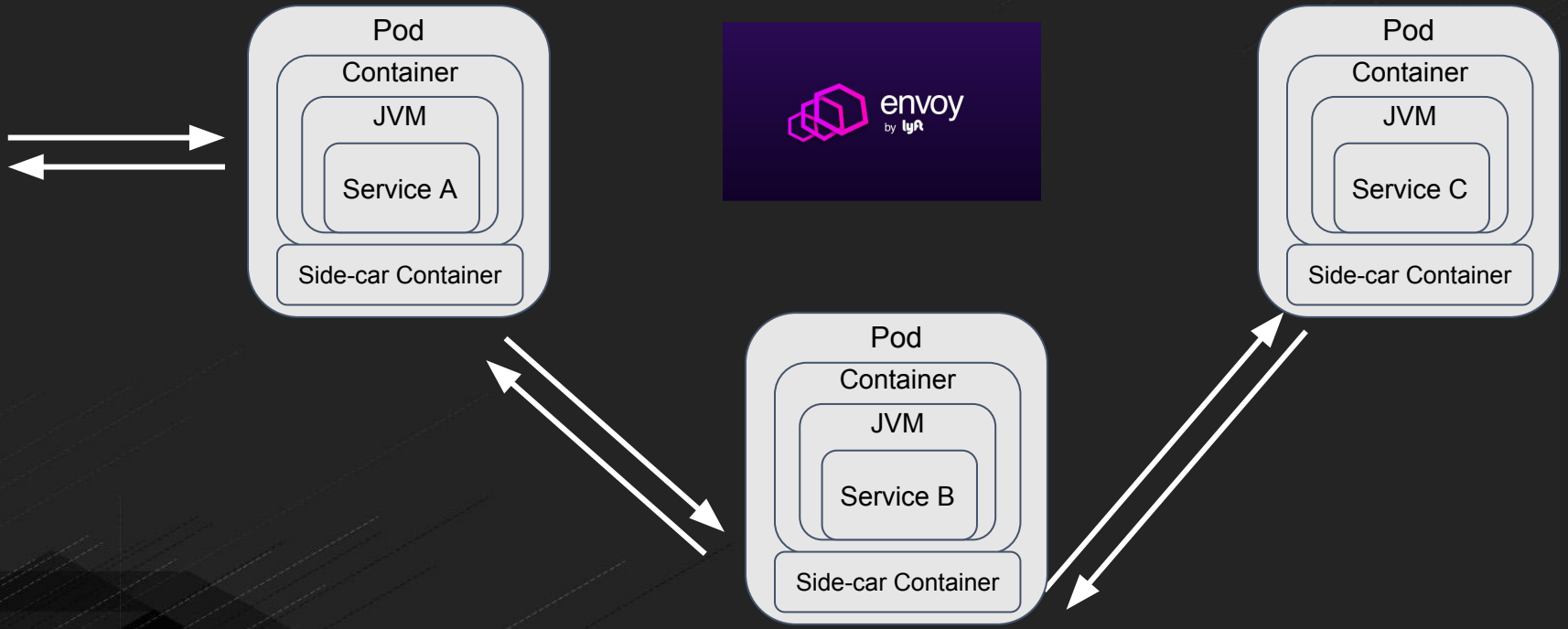
Still too much infrastructure in my business logic

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>
```

Sidecars

Pods with 2 Containers



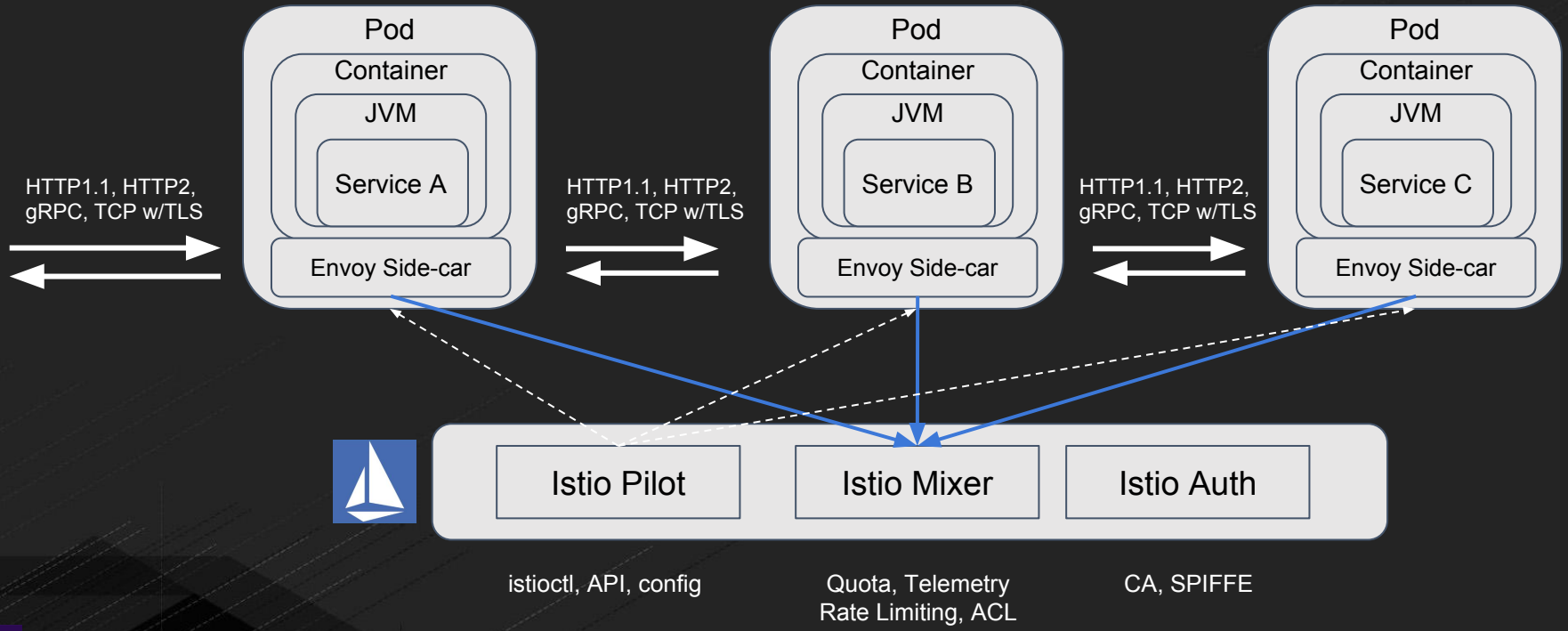
Istio - Sail

Microservices 3.0 - Service Mesh

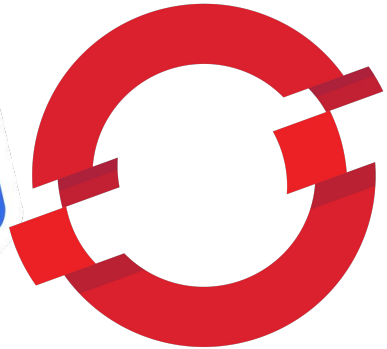
Code Independent

- Intelligent Routing and Load-Balancing
  - A/B Tests
  - Canary Releases
  - Dark Launches
- Distributed Tracing
- Circuit Breakers
- Fine grained Access Control
- Telemetry, metrics and Logs
- Fleet wide policy enforcement

Istio Control Plane



Better Microservices Platform circa 2018



OPENSHIFT

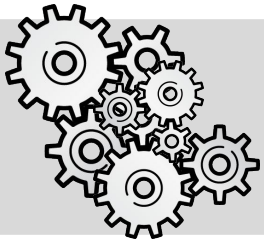
Running the demo

https://github.com/RHodzelmans/istio-lab
https://blog.openshift.com/istio-traffic-management-diving-deeper/

Microservices 4.0?

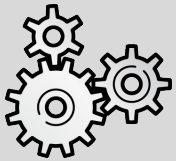
Always Evolving

Service



> Autonomous
> Loosely-coupled

Microservice



> Single Purpose
> Stateless
> Independently Scalable
> Automated

Function

> Single Action
> Event-sourced
> Ephemeral

SERVERLESS PROJECTS / SERVICES

twilio

webtask

APACHE OpenWhisk™

Iron.io

APEX | SERVERLESS INFRASTRUCTURE

Microsoft Azure

f(x)

Back&

SERVERLESS

http://funcatron.org

fission

CLOUD FUNCTIONS BETA

«stdlib»

serverless-docker

syncano

Running the demo

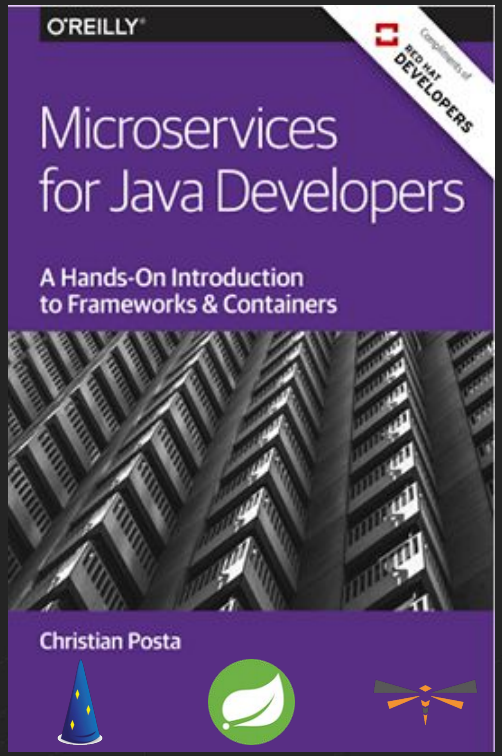
https://gist.github.com/bbrowning/713ed5355324c1a65ad37dded503c4c1
Or better: http://tinyurl.com/openwhisk

So what next?

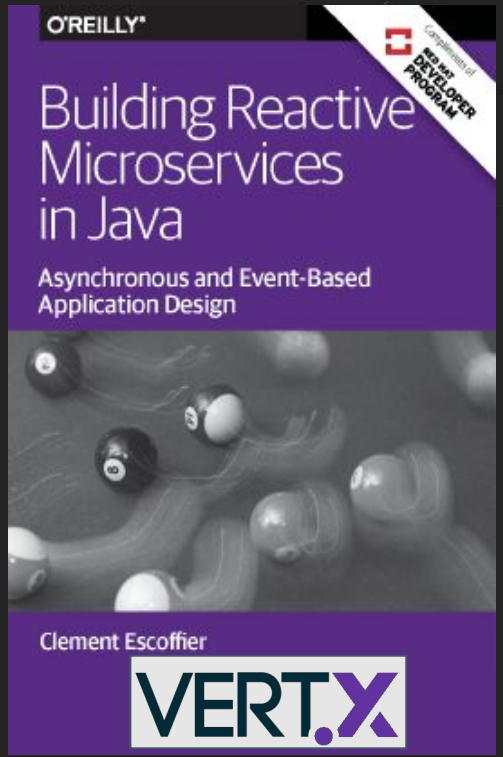
So where can I find this stuff?

https://developer.redhat.com

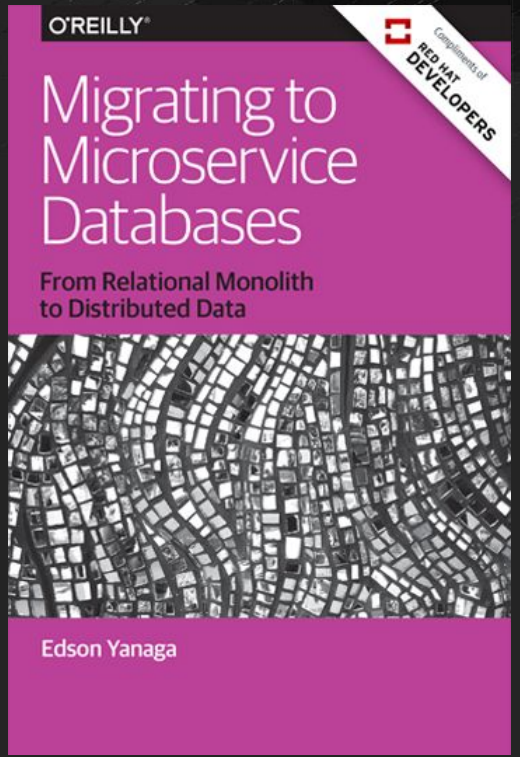
- Blog:
  - https://developers.redhat.com/blog
  - https://blog.openshift.com
- OpenShift:
  - Via Red Hat: CDK or Online
  - Via the public cloud:
    - Google - https://redhat-google.orbitera.com/c2m/trials/signup?testDrive=977
    - Azure - https://testdrive.azure.com/#/test-drive/redhat.openshift-test-drive
- And…...



bit.ly/javamicroservicesbook



bit.ly/reactivemicroservicesbook



bit.ly/mono2microdb

RED HAT®
DEVELOPER
PROGRAM

Thank you, happy to answer questions

@roelhodzelmans roel@redhat.com