

# Cloud Native Java Microservices

Kenny Bastani  
Spring Developer Advocate

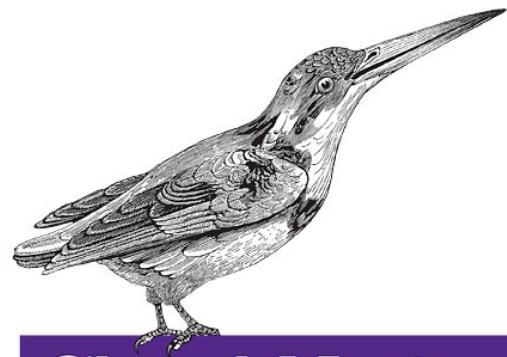


# Kenny Bastani

Spring Developer Advocate



O'REILLY®



Cloud Native Java

DESIGNING RESILIENT SYSTEMS WITH SPRING BOOT,  
SPRING CLOUD, AND CLOUD FOUNDRY

Josh Long & Kenny Bastani

# Agenda

## Agenda

1 Microservices & Cloud Native

2 Spring Boot

3 Spring Cloud

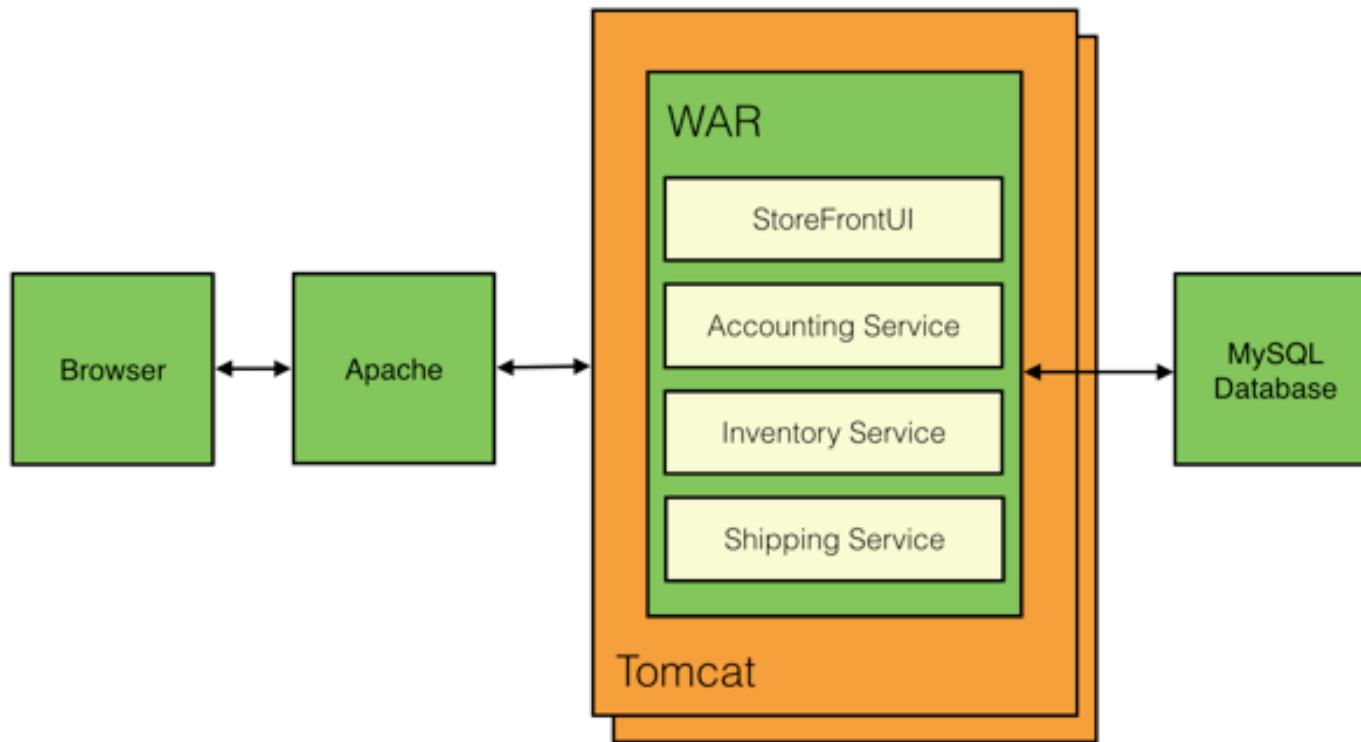
4 RESTful Microservices

5 Push to Cloud Foundry

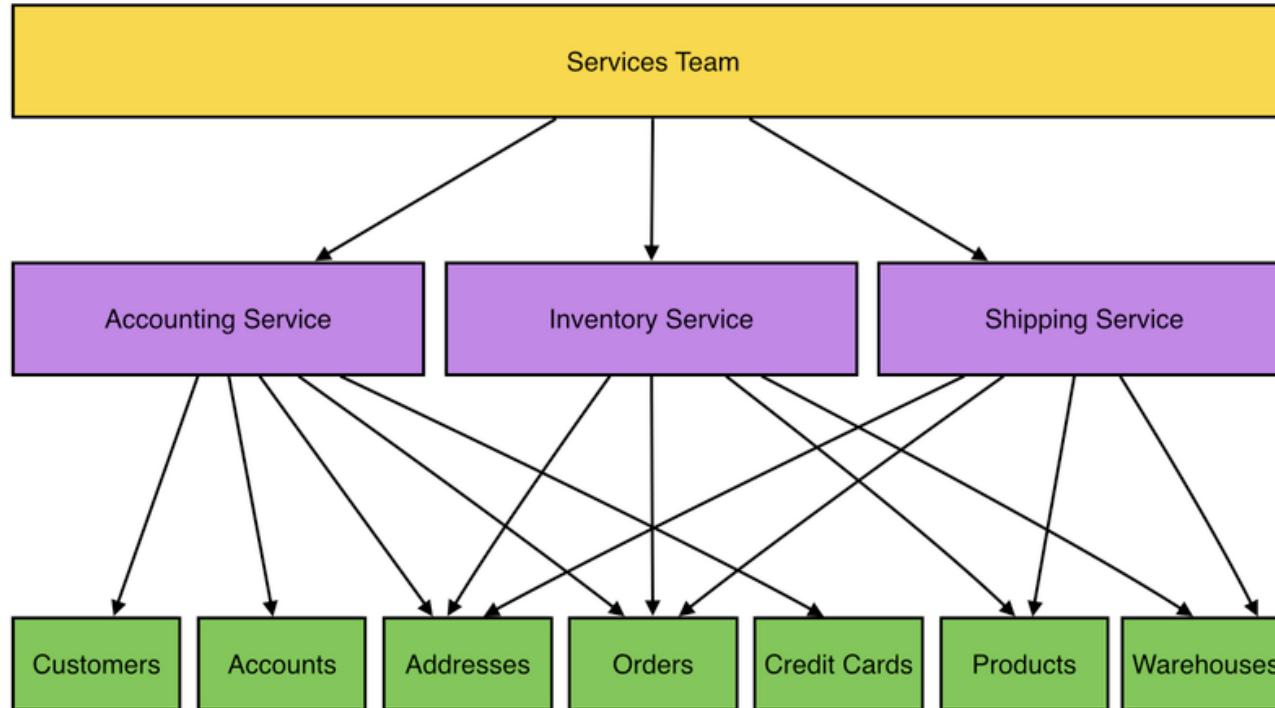
# Microservices

microservices

# Monolithic Architecture

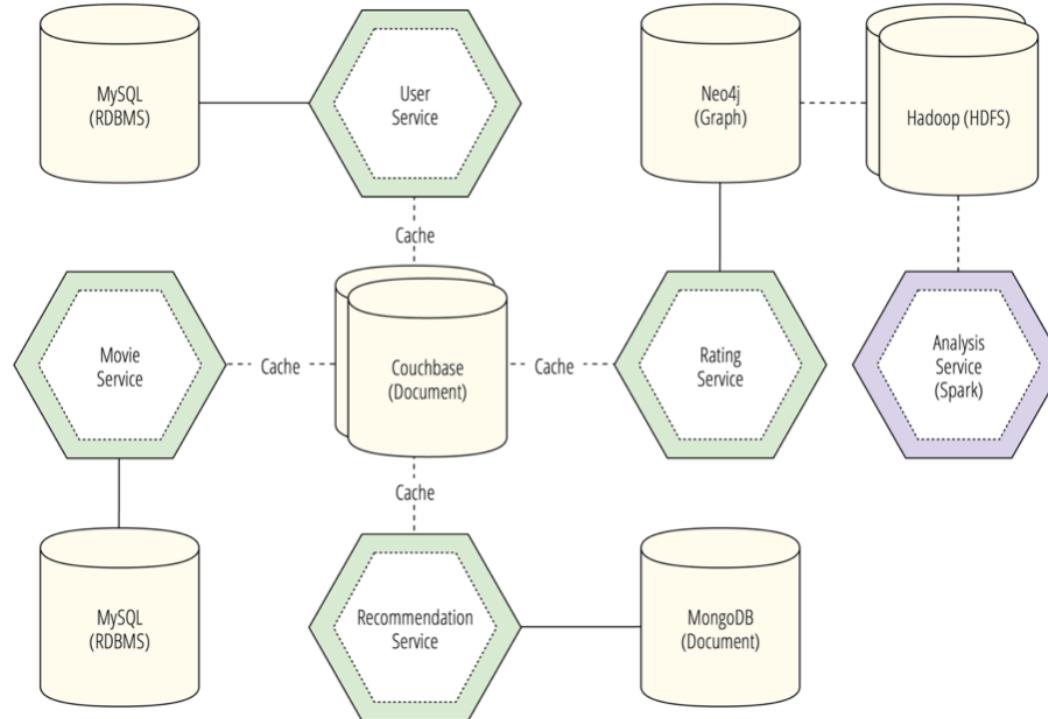


# SOA - Service Oriented Architecture

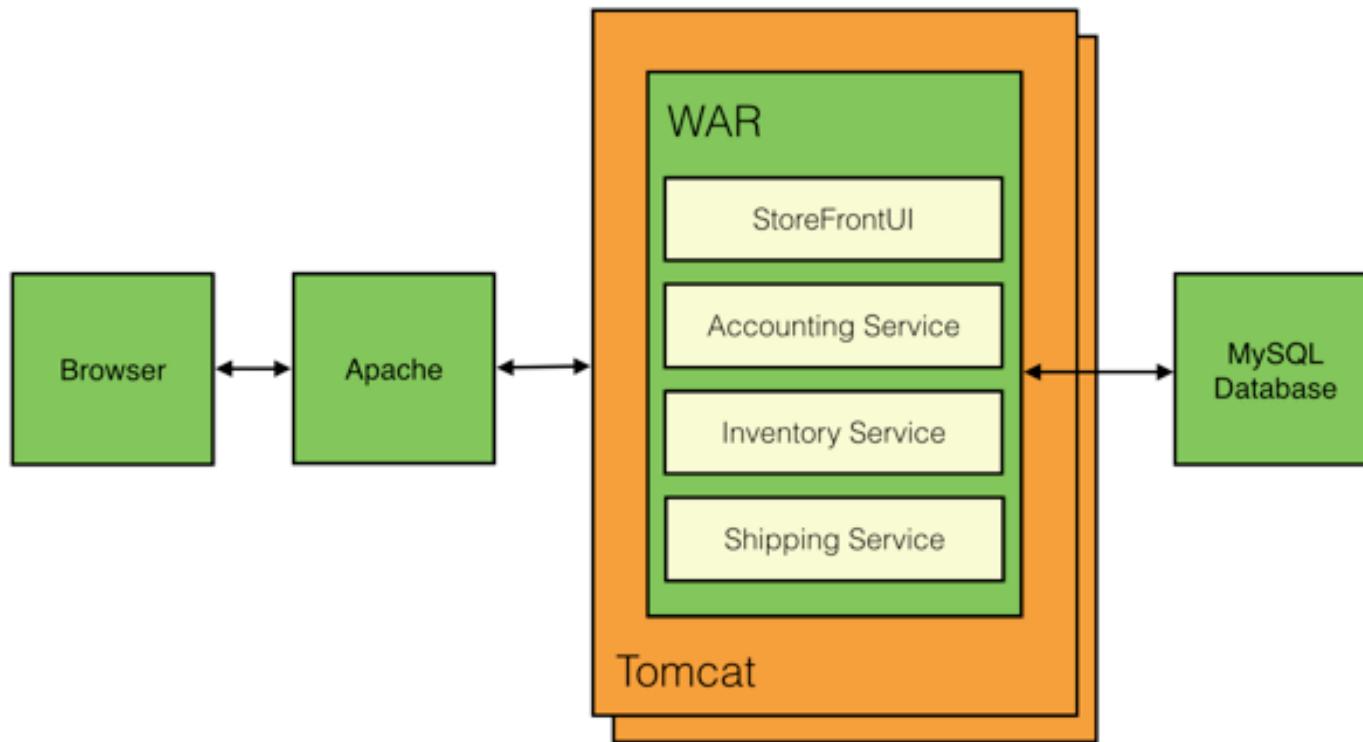


# Microservice Architecture

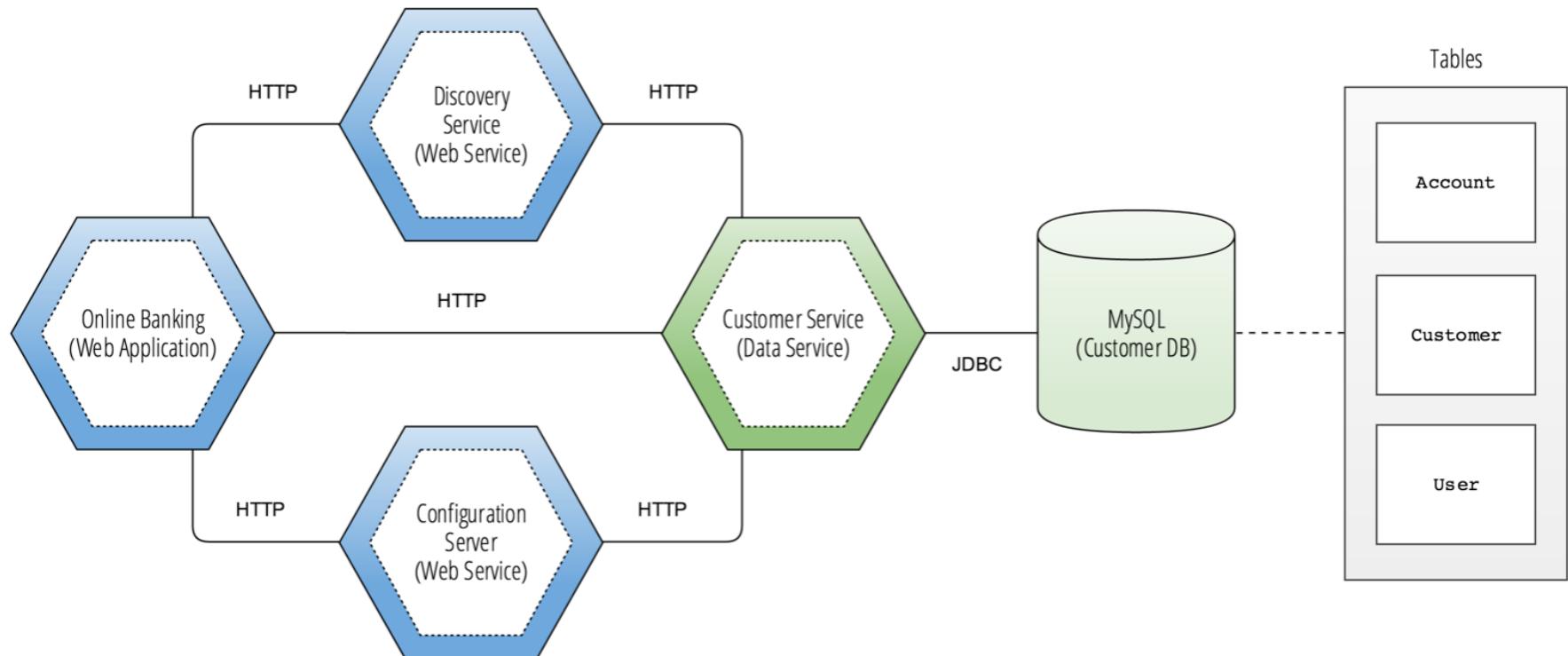
- Each team gets one database and one service
- Shared caches are platform provided services that are shared for consistency



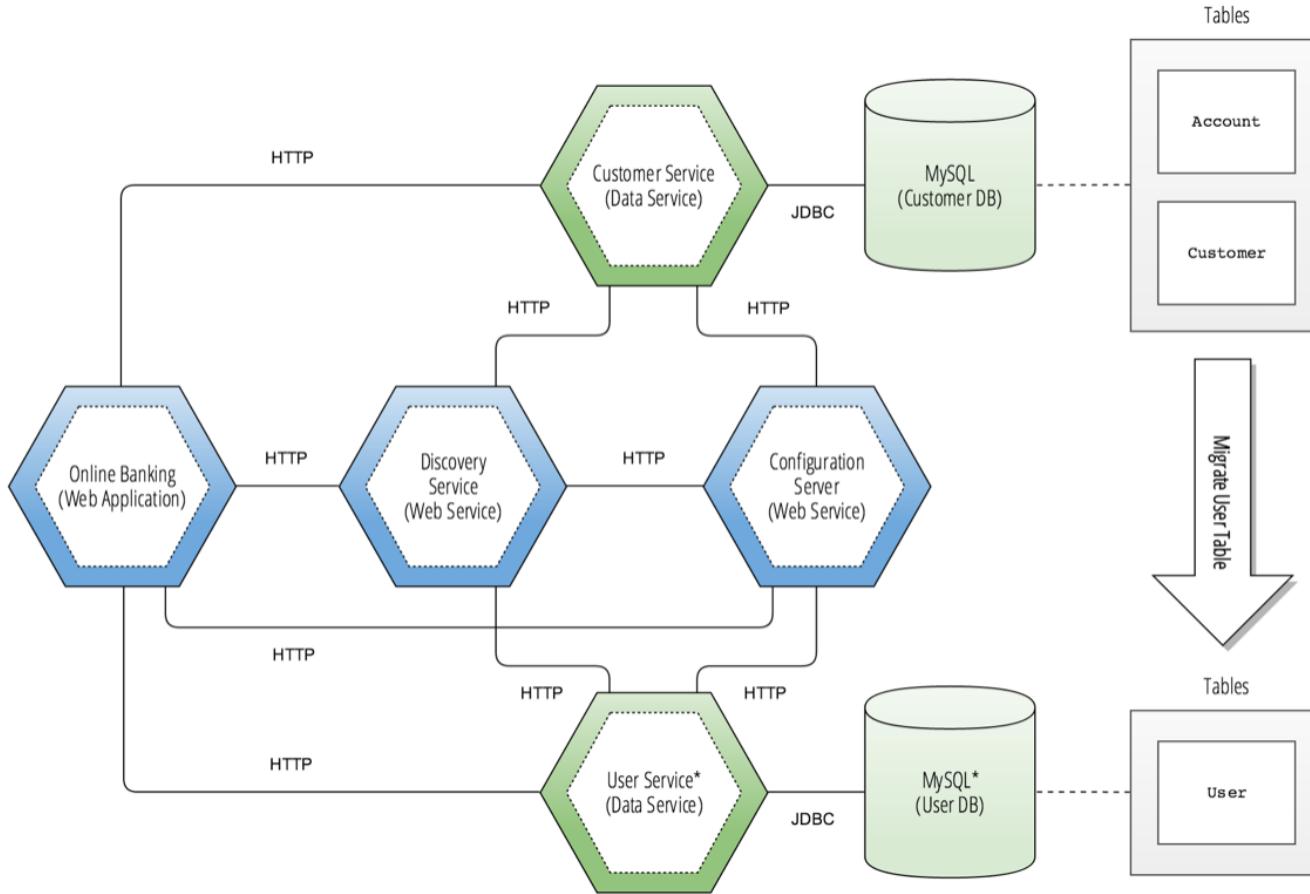
# Monolithic Architecture



# Monolith to Microservice



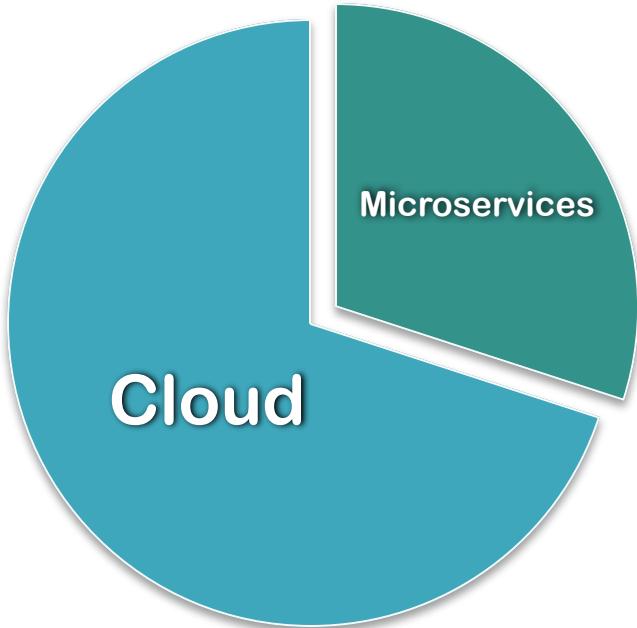
# Monolith to Microservice



# Cloud Native Apps

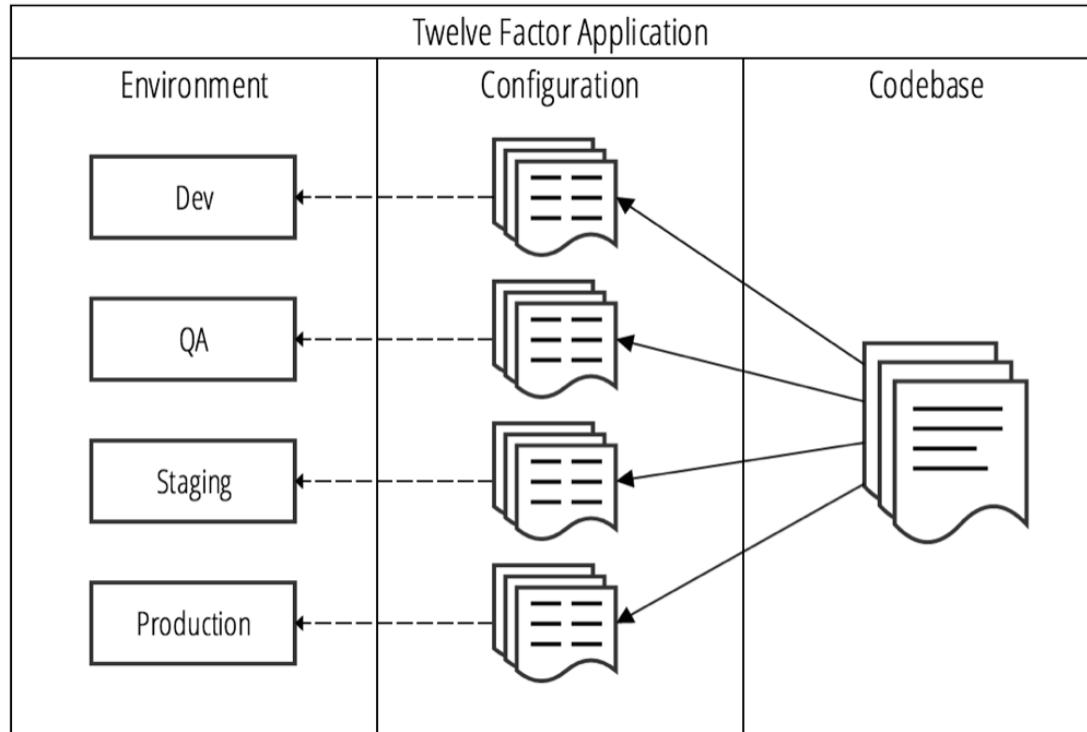


# Cloud Native & Microservices

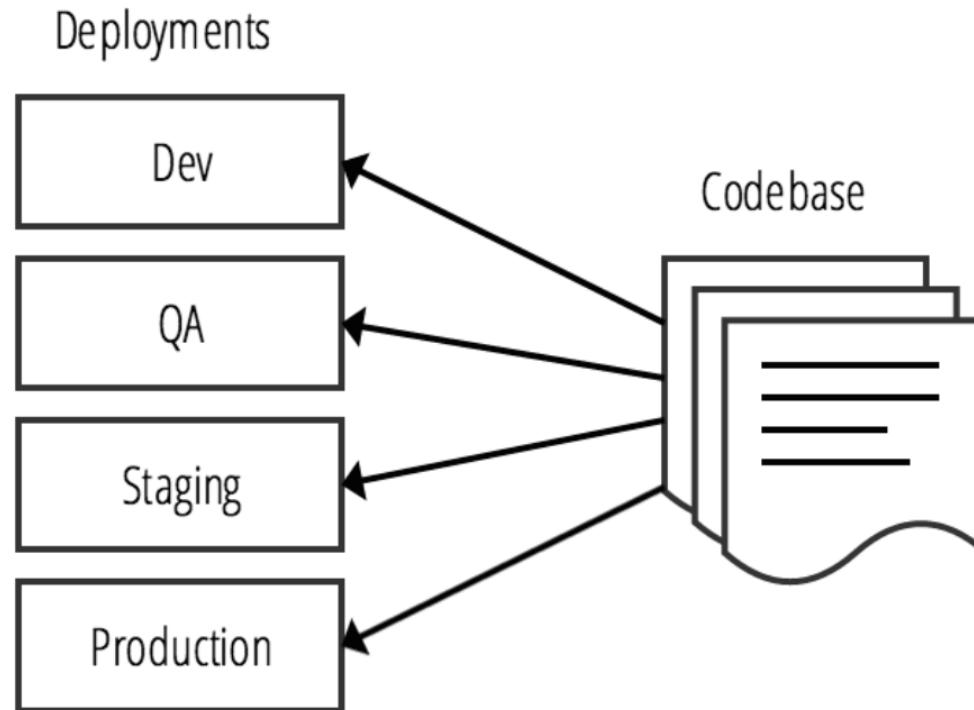


- Microservices are only a small part of a larger picture

# Twelve-factor Application Configuration



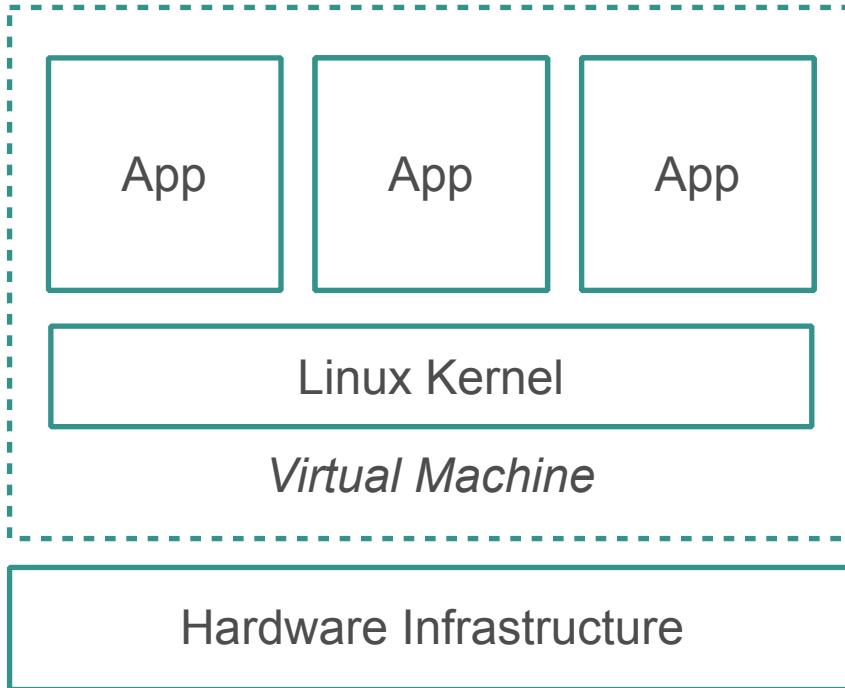
# Twelve-factor Application Deployment



# Cloud Native Ops

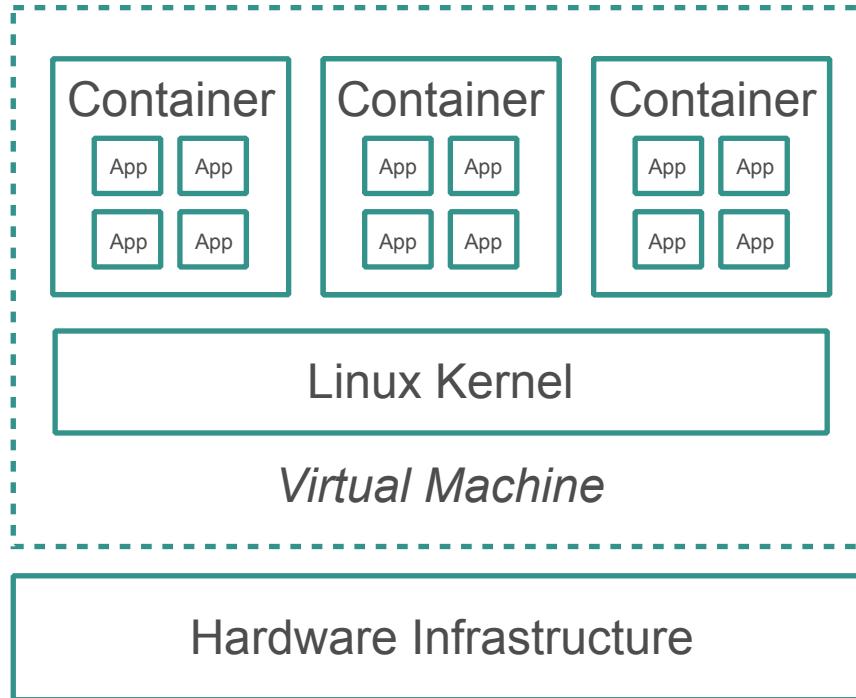


# Application Server Deployment - Monolith



- Load balancing requires provisioning of new VMs and app server installations
- Poor resource isolation; memory leaks can cause other applications to become unavailable
- Runtime environment is driven by the operator

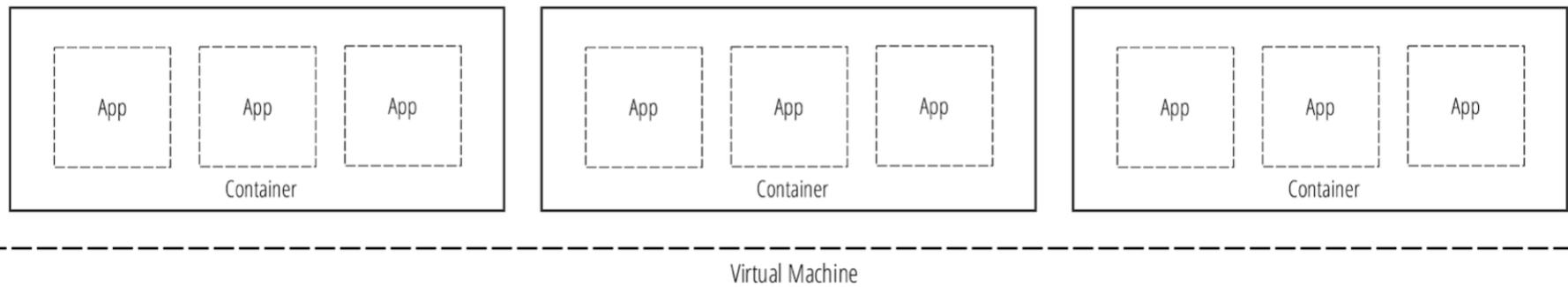
# Linux Container Deployment - Microservice



- Development team drives the application runtime of a container
- Containers are resource isolated, allowing efficient scheduling onto a grid of VMs
- Containers take seconds to start, VMs take minutes
- It's not a rule that there is one microservice per container

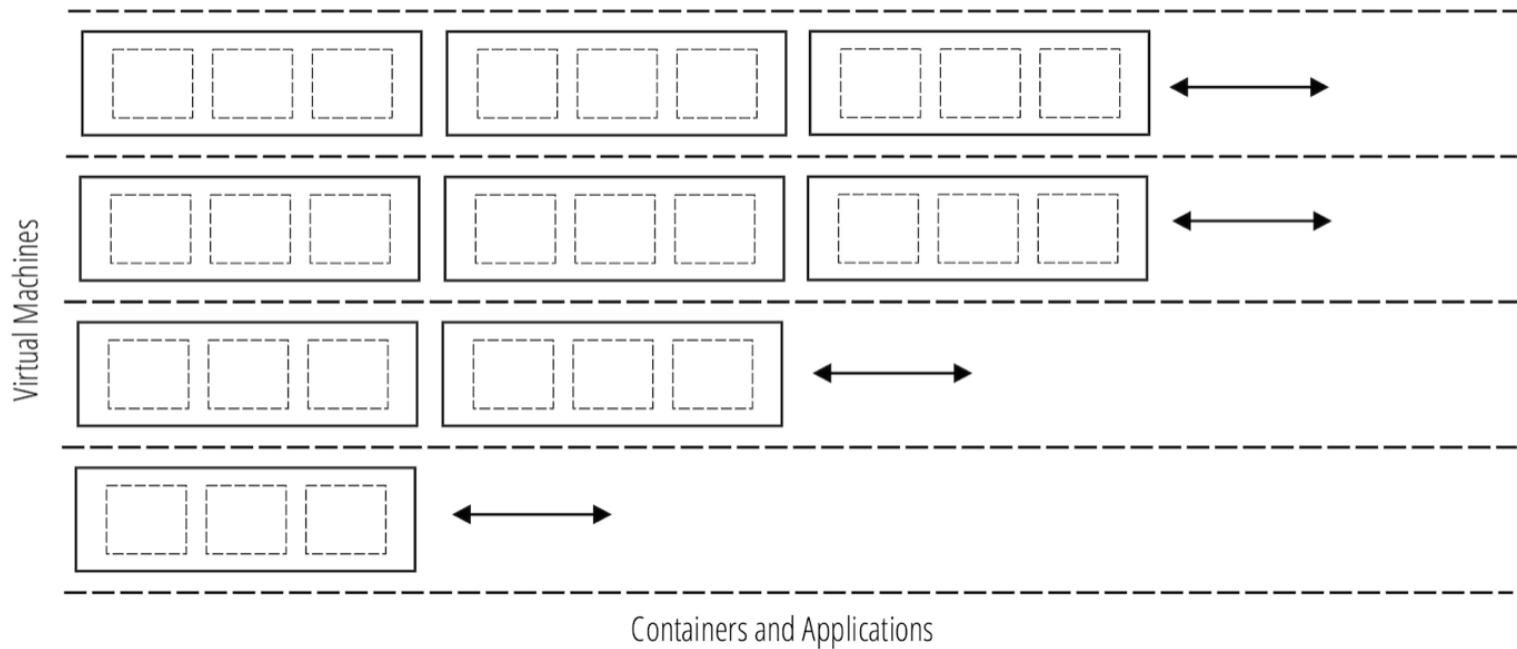
# Microservices - Container Deployment

- Each microservice can be containerized with their application dependencies
- Containers get scheduled on virtual machines with an allotted resource policy



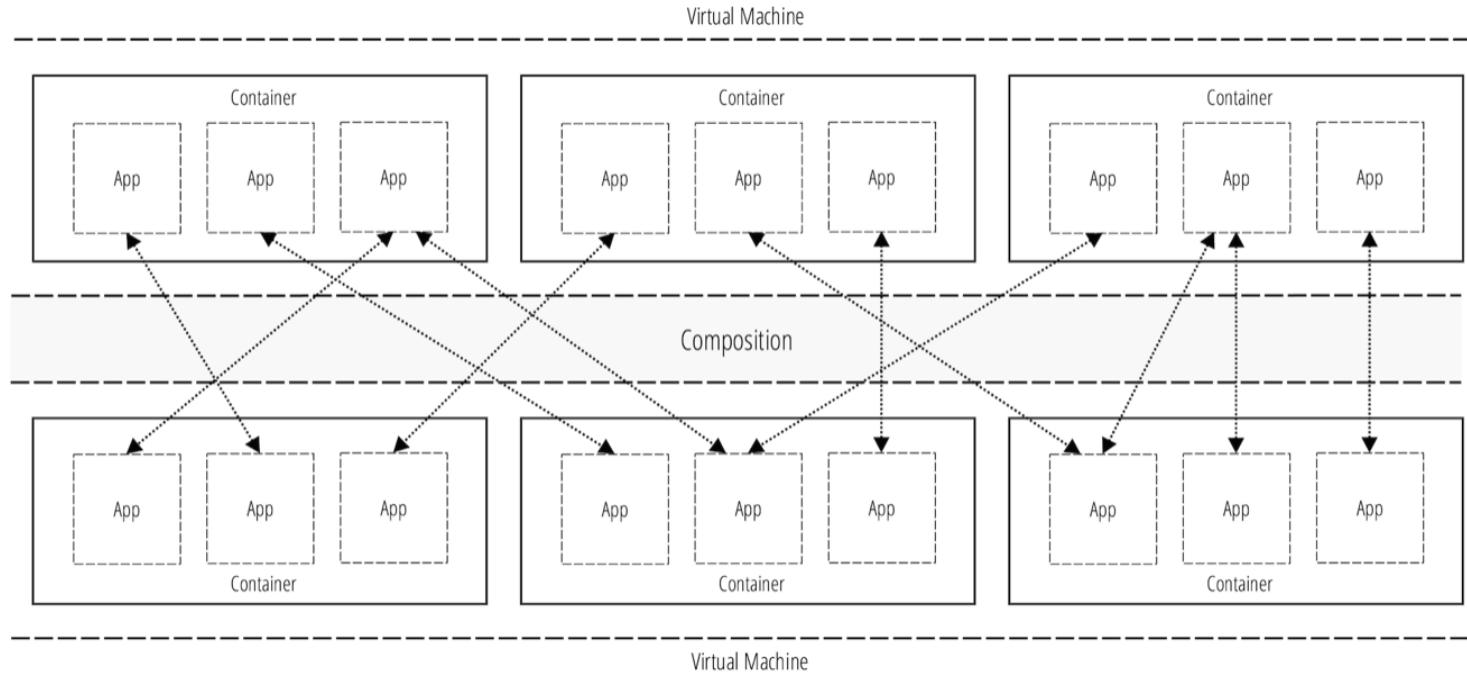
# Auto-scaling

- Minutes to start a VM, but seconds to start a container
- An elastic runtime handles auto-scaling of VMs with cloud providers



# Orchestration

- Each microservice needs to communicate outside containers
- Service discovery provides an automatic method for finding other service dependencies



# Cloud Native Java

Cloud Native Java

# Spring Boot

about Spring

A JVM micro-framework for building  
microservices

# What is Spring Boot?



**Phil Webb**  
@phillip\_webb



Following

For those on [reddit.com/r/java/](https://www.reddit.com/r/java/) saying it's Spring Boot is "the framework for a framework" here's a diagram:

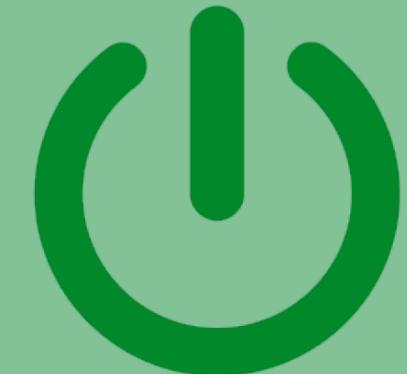


RETWEETS  
**111**

FAVORITES  
**68**



7:54 PM - 8 Sep 2015



# spring boot

supports rapid development of production-ready applications and services

## Spring Initializr for bootstrapping your applications

The screenshot shows the Spring Initializr web application at [start.spring.io](http://start.spring.io). The page title is "SPRING INITIALIZR bootstrap your application now". It features a "Generate a [Maven Project] with Spring Boot 1.3.2" button. The left side has a "Project Metadata" section with fields for Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), Package Name (com.example), Packaging (Jar), Java Version (1.8), and Language (Java). The right side has a "Dependencies" section with a search bar ("Search for dependencies: Web, Security, JPA, Actuator, Devtools...") and a list of selected starters: Web, JPA, Security, Actuator, and Rest Repositories.

SPRING INITIALIZR bootstrap your application now

Generate a [Maven Project] with Spring Boot 1.3.2

Project Metadata

Artifact coordinates

Group  
com.example

Artifact  
demo

Name  
demo

Description  
Demo project for Spring Boot

Package Name  
com.example

Packaging  
Jar

Java Version  
1.8

Language  
Java

Dependencies

Add Spring Boot Starters and dependencies to your application

Search for dependencies  
Web, Security, JPA, Actuator, Devtools...

Selected Starters

Web, JPA, Security, Actuator, Rest Repositories

Too many options? [Switch back to the simple version.](#)

Generate Project

# Spring Boot Roles



# Automatic Configuration

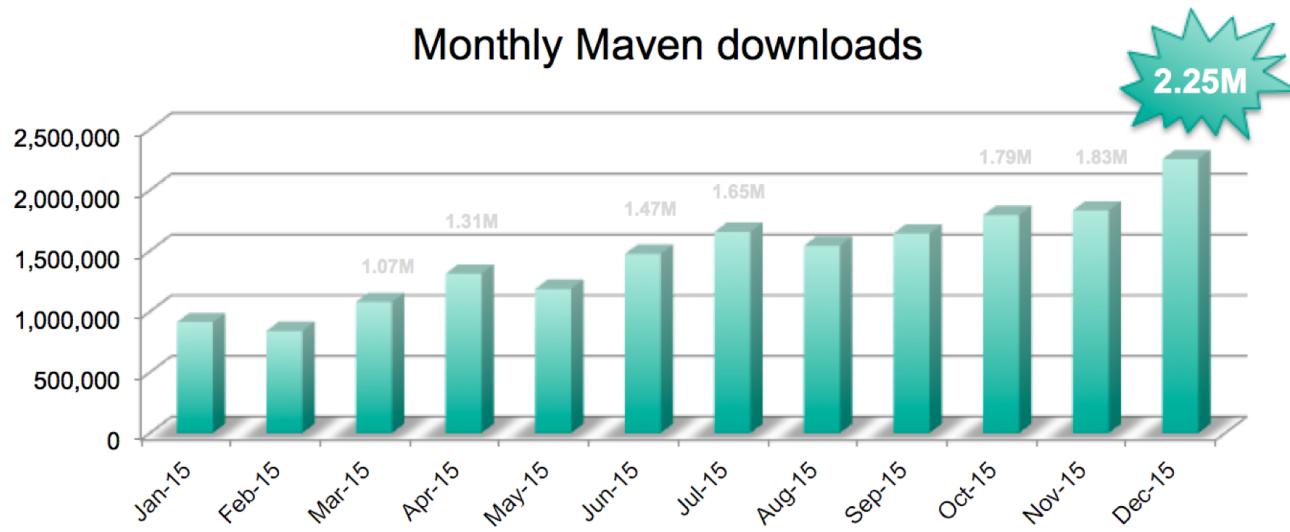
- An application class is annotated with `@SpringBootApplication`
- Additional annotations are added to indicate the role of the Spring Boot application

```
@SpringBootApplication
@EnableSidecar
public class Application {
    @Autowired
    private RepositoryRestMvcConfiguration restConfiguration;

    public static void main(String[] args) {
        new SpringApplicationBuilder(Application.class).web(true).run(args);
    }

    @PostConstruct
    public void postConstructConfiguration() {
        restConfiguration.objectMapper().registerModule(new Jackson2HalModule());
    }
}
```

# Spring Boot for Microservices



# Spring Cloud



A toolset designed for building distributed systems



spring cloud



Apache Zookeeper



\* these logos are all **trademark/copyright** their respective owners (T-B, L-R):

Netflix, [amazon.com](http://amazon.com), Apache Software Foundation, Cloud Foundry, Hashicorp  
they are ALL great organizations and we love their open-source and their APIs!!



# What is Spring Cloud?

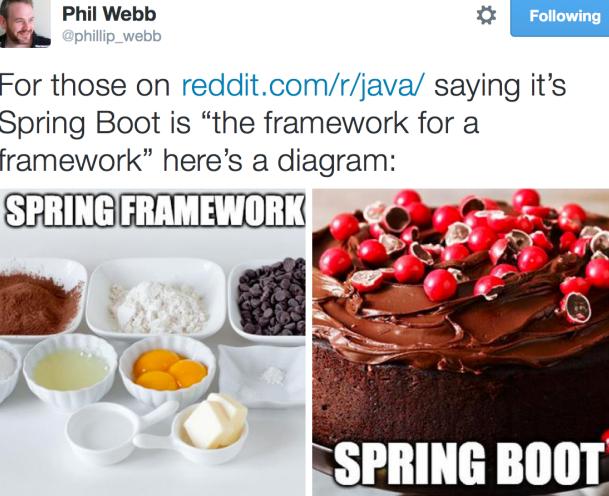
- \* Service Discovery
- \* API Gateway
- \* Config Server
- \* Circuit Breakers
- \* Distributed Tracing

# What is Spring Cloud?

- Spring Cloud provides a way to turn Spring Boot microservices into distributed applications

 **Phil Webb**  
@phillip\_webb

For those on [reddit.com/r/java/](https://www.reddit.com/r/java/) saying it's Spring Boot is "the framework for a framework" here's a diagram:



The diagram shows a comparison between the Spring Framework and Spring Boot. On the left, under 'SPRING FRAMEWORK', there are several small bowls containing ingredients like flour, sugar, eggs, and butter. Next to them is a chocolate cake topped with cherries. An arrow points from this image to the right side of the slide. On the right, under 'SPRING BOOT', there is a large image of many colorful, decorated cupcakes.

**SPRING FRAMEWORK**

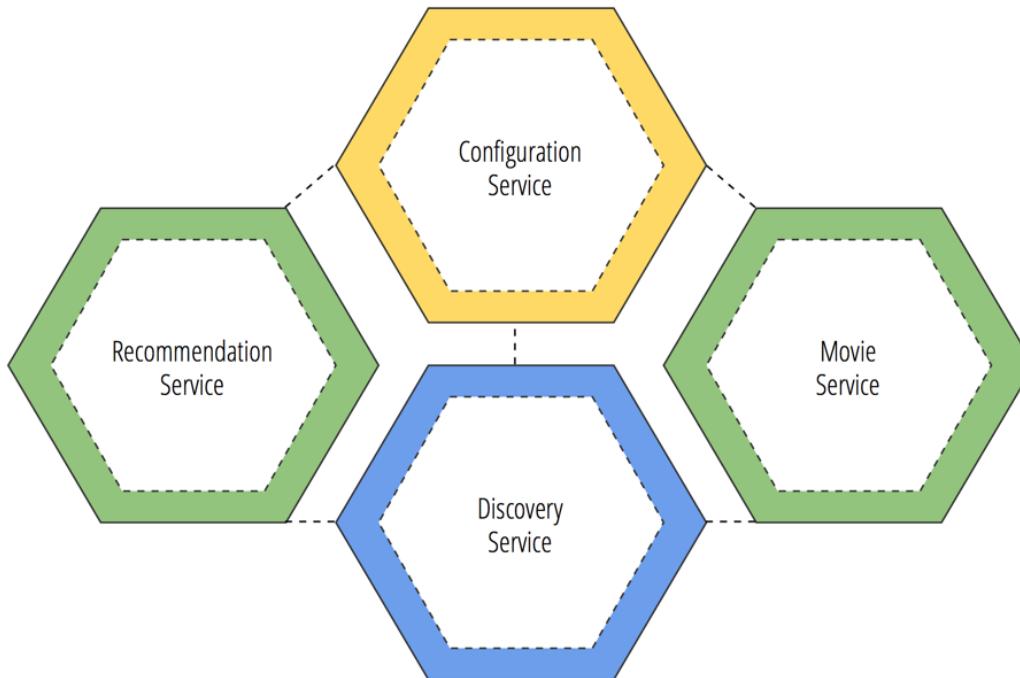
**SPRING BOOT**

RETWEETS 111 FAVORITES 68

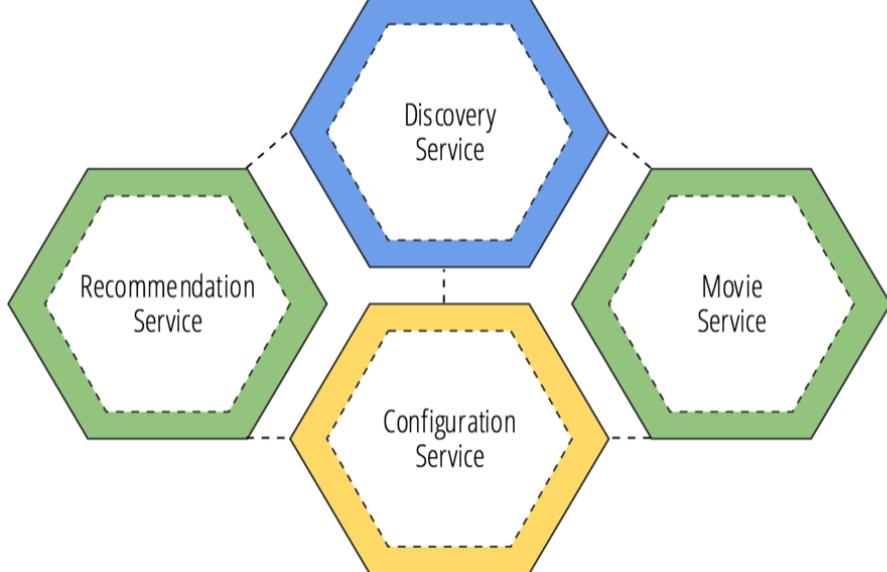
7:54 PM - 8 Sep 2015



# Service Discovery & Intelligent Routing



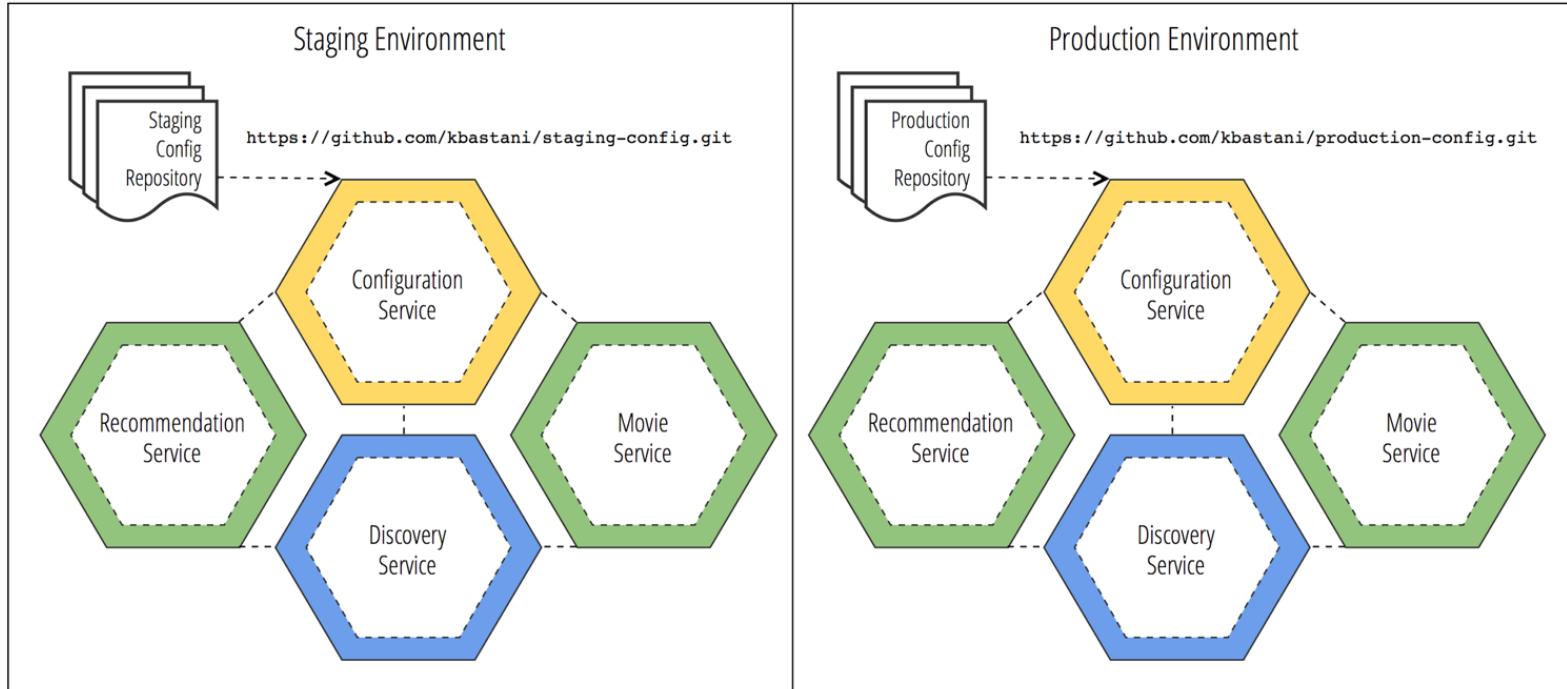
# Client-side load balancing



```
Service Registry

<application>
  <name>MOVIE</name>
  <instance>
    <instanceId>192.168.99.1:movie:9004</instanceId>
    <hostName>192.168.99.1</hostName>
    <app>MOVIE</app>
    <ipAddr>192.168.99.1</ipAddr>
    <status>UP</status>
    <port enabled="true">9004</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
<application>
  <name>RECOMMENDATION</name>
  <instance>
    <instanceId>192.168.99.2:recommendation:9712</instanceId>
    <hostName>192.168.99.2</hostName>
    <app>RECOMMENDATION</app>
    <ipAddr>192.168.99.2</ipAddr>
    <status>UP</status>
    <port enabled="true">9712</port>
    <securePort enabled="false">443</securePort>
  </instance>
</application>
```

# Configuration Service



# API Gateway



# RESTful Microservices

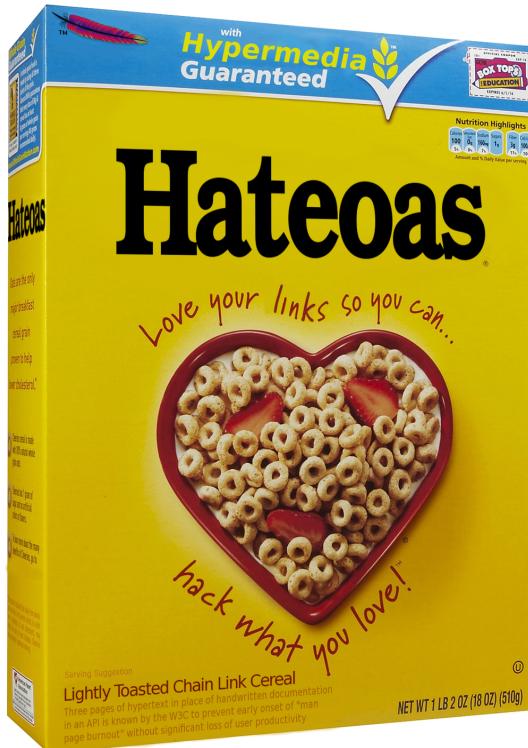
RESTFUL MICROSERVICES

# HATEOAS

Hypermedia as the Engine of Application State

(REST APIs that self-describe)

# It's pronounced hawt-ee-oh-as



# Richardson Maturity Model

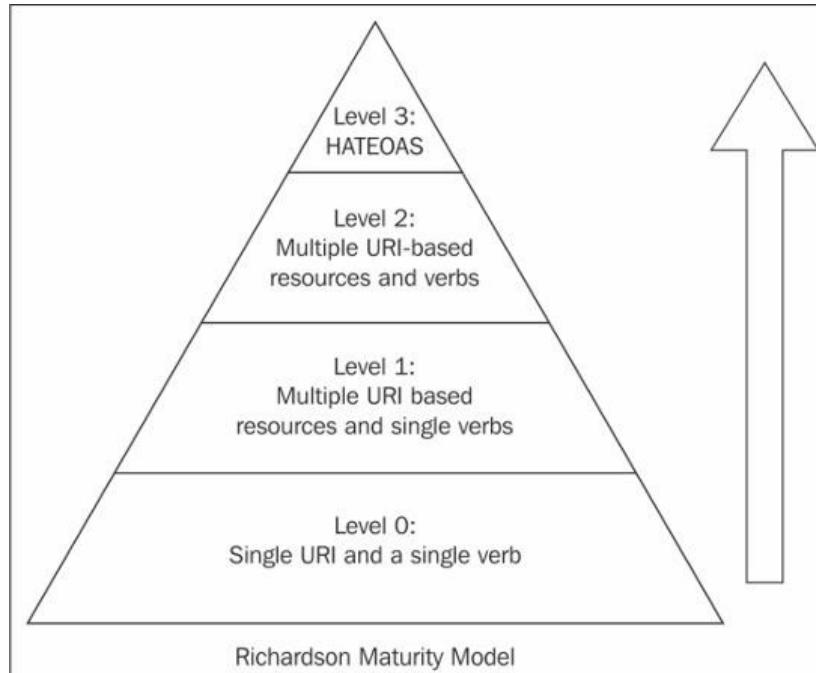


Photo credit: Packt Publishing

# REST API Gateway



- API gateway downloads routes from services exposing REST APIs
- Each route of other microservices will be hosted on the API gateway
- API gateway will automatically reverse proxy to backend services
- Hypermedia allows traversing entire REST API description of microservices

# Browsing REST APIs

```
{  
  "_links": {  
    "ratings": {  
      "href": "http://rating-77.cfapps.io/ratings{?page,size,sort}",  
      "templated": true  
    },  
    "products": {  
      "href": "http://rating-77.cfapps.io/products{?page,size,sort}",  
      "templated": true  
    },  
    "users": {  
      "href": "http://rating-77.cfapps.io/users{?page,size,sort}",  
      "templated": true  
    },  
    "profile": {  
      "href": "http://rating-77.cfapps.io/alps",  
      "templated": false  
    }  
  }  
}
```

# Each API call provides links

```
{  
  "_links": {  
    "first": {  
      "href": "http://rating-77.cfapps.io/users?page=0&size=20",  
      "templated": false  
    },  
    "self": {  
      "href": "http://rating-77.cfapps.io/users",  
      "templated": false  
    },  
    "next": {  
      "href": "http://rating-77.cfapps.io/users?page=1&size=20",  
      "templated": false  
    },  
    "last": {  
      "href": "http://rating-77.cfapps.io/users?page=47&size=20",  
      "templated": false  
    },  
    "search": {  
      "href": "http://rating-77.cfapps.io/users/search",  
      "templated": false  
    }  
  },  
  "_embedded": {  
    "users": [  
      {  
        "id": 0,  
        "knownId": "196",  
        "_links": {  
          "self": {  
            "href": "http://rating-77.cfapps.io/users/0",  
            "templated": false  
          },  
          "user": {  
            "href": "http://rating-77.cfapps.io/users/0",  
            "templated": false  
          }  
        }  
      }  
    ]  
  }  
}
```

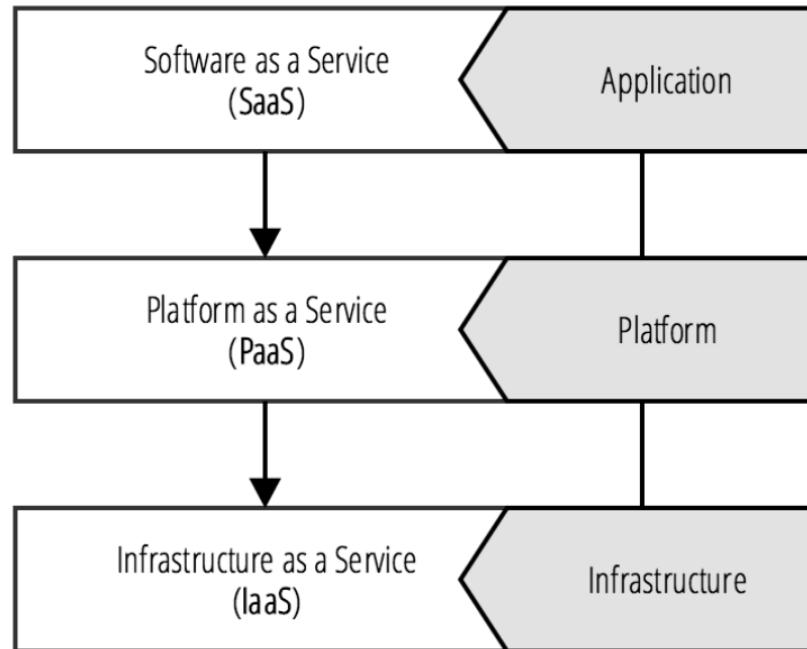


# Cloud Foundry

“To always be shipping, you need a shipyard.”

# Cloud Foundry

Application-centric opinionated platform



# Push to Cloud Foundry

- <http://run.pivotal.io>
- Go to Spring Initializr, select **Web**
- Name the artifactId **hello-world**
- Add a **@RestController** named Hello
- Return “Hello World” at the root endpoint
- **mvn clean install & cf push**

Thanks!

@kennybastani