

The background features a dark space with a constellation of bright blue stars connected by thin lines. A prominent pinkish-purple nebula is visible in the upper left quadrant. The text is centered over the constellation.

# Spring Cloud

В эру Kubernetes

# Привет!

- Алексей Нестеров, VMware
- Команда Spring Cloud Commercial
  - Spring Cloud for Tanzu Application Service (PCF)
  - Spring Cloud Gateway for Kubernetes
- Часто захожу в подкаст Radio-T



# О чем буду говорить

- Жизнь до Kubernetes и немного истории
- Эра Kubernetes и как он решил все проблемы
- Что может дать Spring Cloud
- Демо
- Итоги
- Q&A

# Зачем?

Приложения распределенные

- **Discovery**
- **Fault tolerance**
- **Resilience**
- **Configuration**
- **API Management**

А значит нужны:

- **Service Discovery**
- **Circuit Breakers**
- **Load Balancing**
- **Configuration**
- **API gateways**

# Что было до Kubernetes?

Одна кинокомпания **N** сделала свое решение

- Service Discovery: **Eureka**
- Client-side load balancing: **Ribbon**
- Circuit breaker: **Hystrix**
- App configuration: **Archaius**
- API Management: **Zuul**

# Что было до Kubernetes?

Одна софтверная компания  адаптировала для Spring

- Service Discovery: **Eureka**
- Client-side load balancing: **Ribbon**
- Circuit breaker: **Hystrix**
- App configuration: Archaius
- API Management: Zuul

# Что было до Kubernetes?

И постепенно переписала

- Service Discovery: **Eureka** (or Consul / etcd / zookeeper)
- Client-side load balancing: **Spring Cloud Load Balancer**
- App configuration: **Spring Cloud Config Server**
- Circuit breaker: **Spring Cloud Circuit Breaker**
- API Management: **Spring Cloud Gateway**

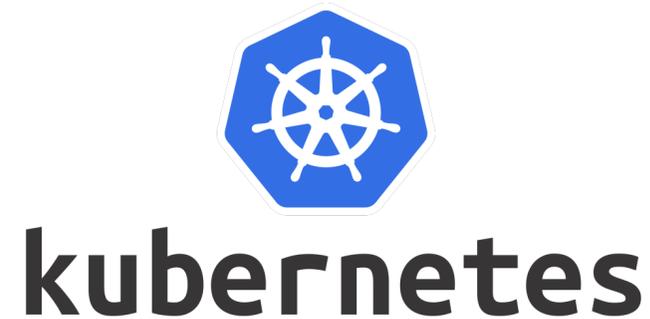
# Spring Cloud: what's in the box

Еще там много всего не привязанного к платформе:

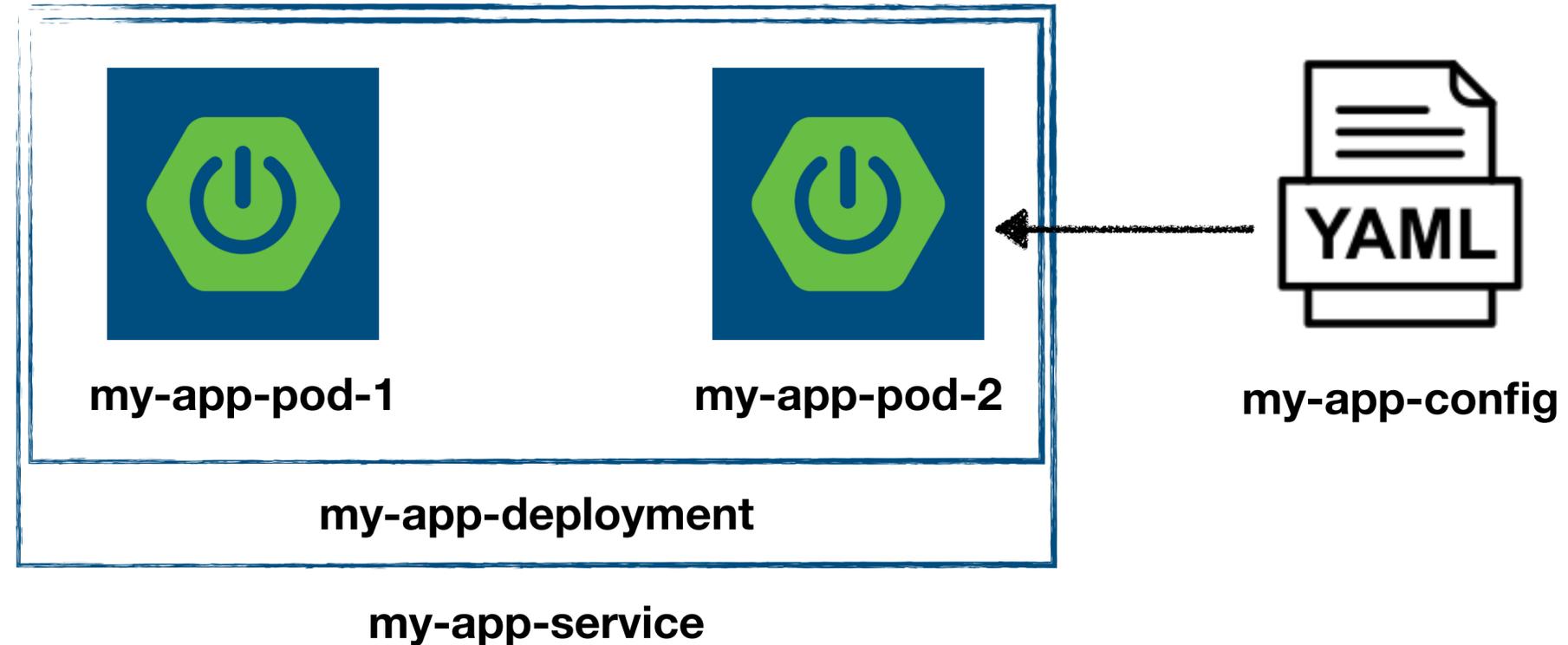
- Tracing: **Sleuth**
- Event-based microservices: **Spring Cloud Stream**
- Consumer Driven Contracts: **Spring Cloud Contract**
- FaaS: **Spring Cloud Function**
- REST clients: **OpenFeign**

# Spring Cloud: what's in the box

И поддержка интеграция со многими облачными провайдерами



# Kubernetes за 2 минуты



`my-app-service.my-namespace.svc.cluster-domain.example`

# Что там в Kubernetes?

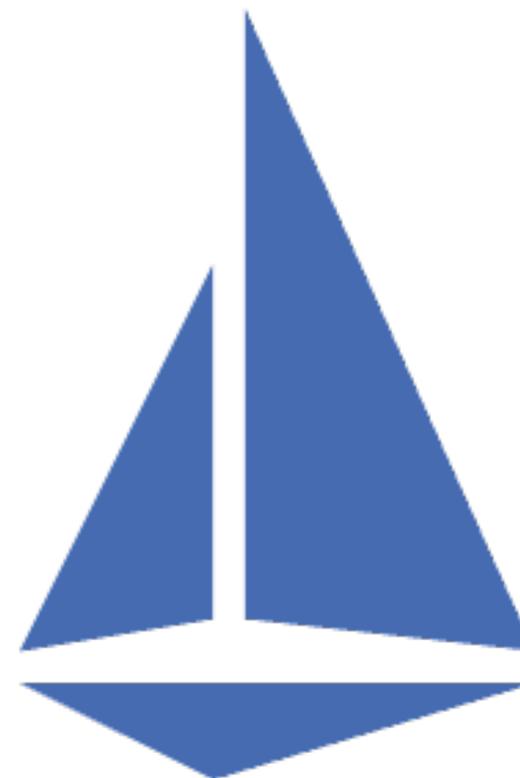
K8S stack

- **Service Discovery:** K8S DNS
- **Fail tolerance:** services + liveness / readiness
- **App configuration:** ConfigMap + Secrets
- Circuit breaker: ?
- API Management: ?

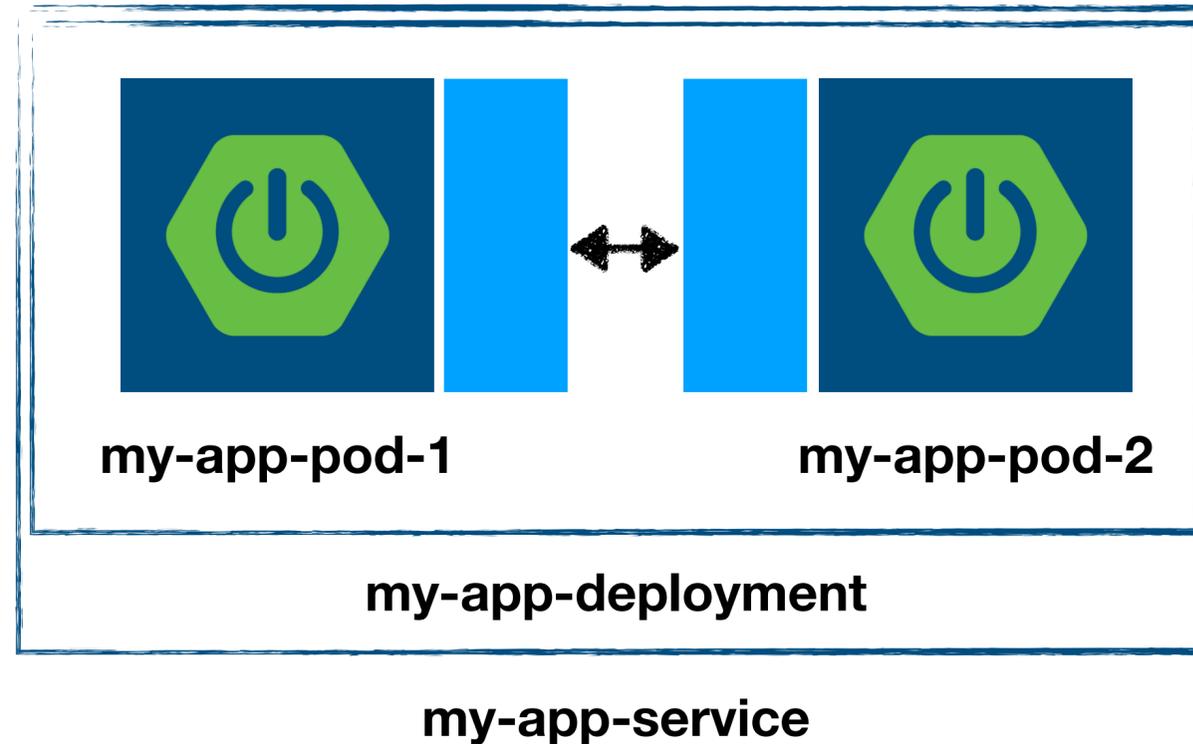
# Так Service Mesh же!



+



# Service Mesh за 1 минуту



`my-app-service.my-namespace.svc.cluster-domain.example`

# Tak Service Mesh же!

K8S+Istio stack

- **Service Discovery:** K8S DNS
- **Fault tolerance:** services + liveness / readiness
- **App configuration:** ConfigMap + Secrets
- **Circuit breaker:** Istio
- **API Management:** Istio

# Spring Cloud не нужен? 🤔

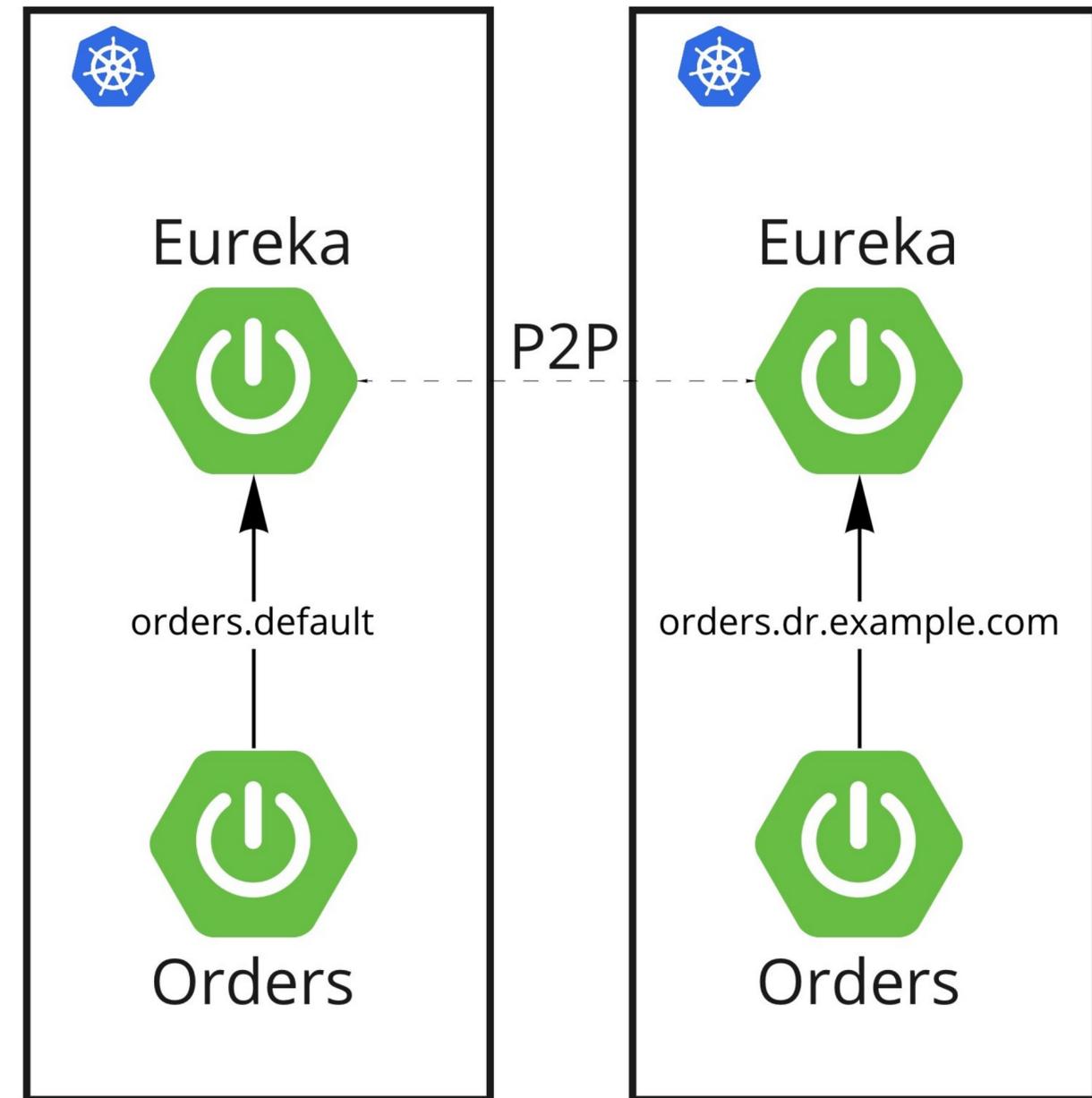
K8S+Istio

- Имеют далеко не все фичи, которые могут быть нужны
- Внешняя сущность по отношению к приложению
- Часто Ops, не Dev. Может привести к Issue-tracker deployment
- Для сложных случаев приходится программировать на YAML

# Service Discovery

## Eureka + Spring Cloud Discovery Client

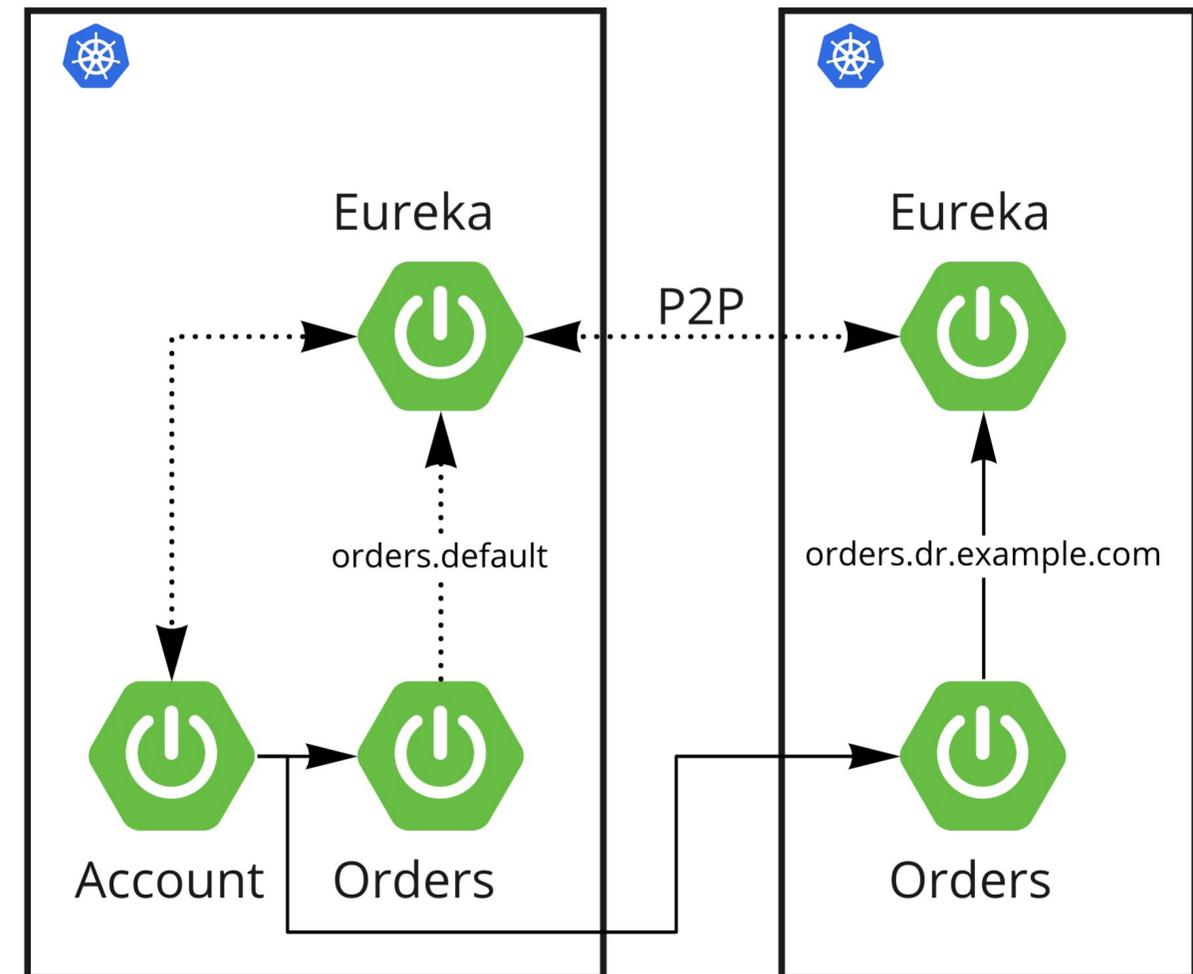
- Универсальный клиент
- Разные бэкенды - Eureka, Consul, etcd
- Несколько регионов (датацентров)
- Несколько зон с zone affinity
- Peer-to-peer репликация
- Сервисы вне кластера
- Spring Boot приложение



# Resilience

## Spring Cloud Load Balancer

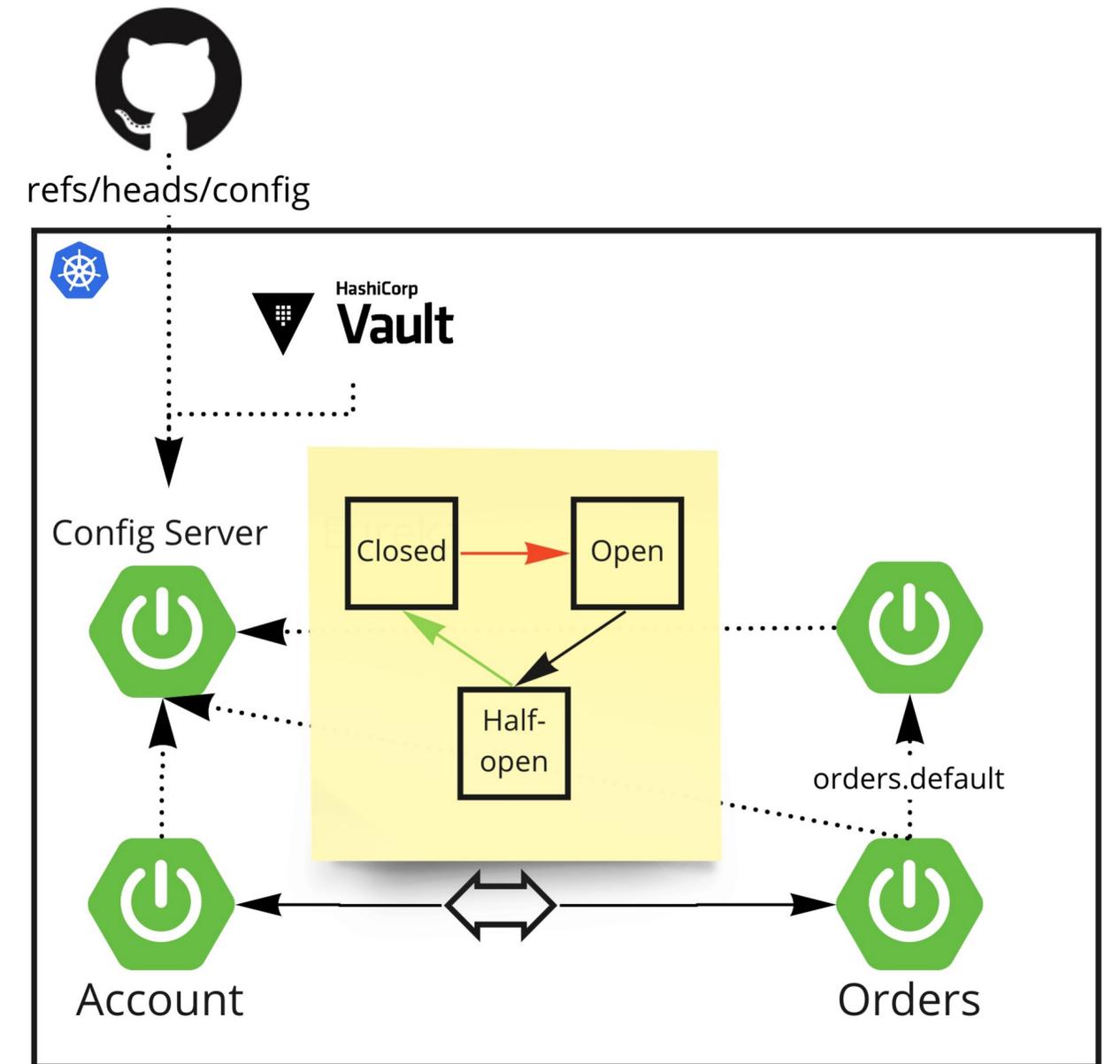
- Интеграция с Service Discovery
  - Multi-zone
  - Zone affinity
- Часть приложения, можно тонко настроить



# Fault tolerance

## Spring Cloud Circuit Breaker

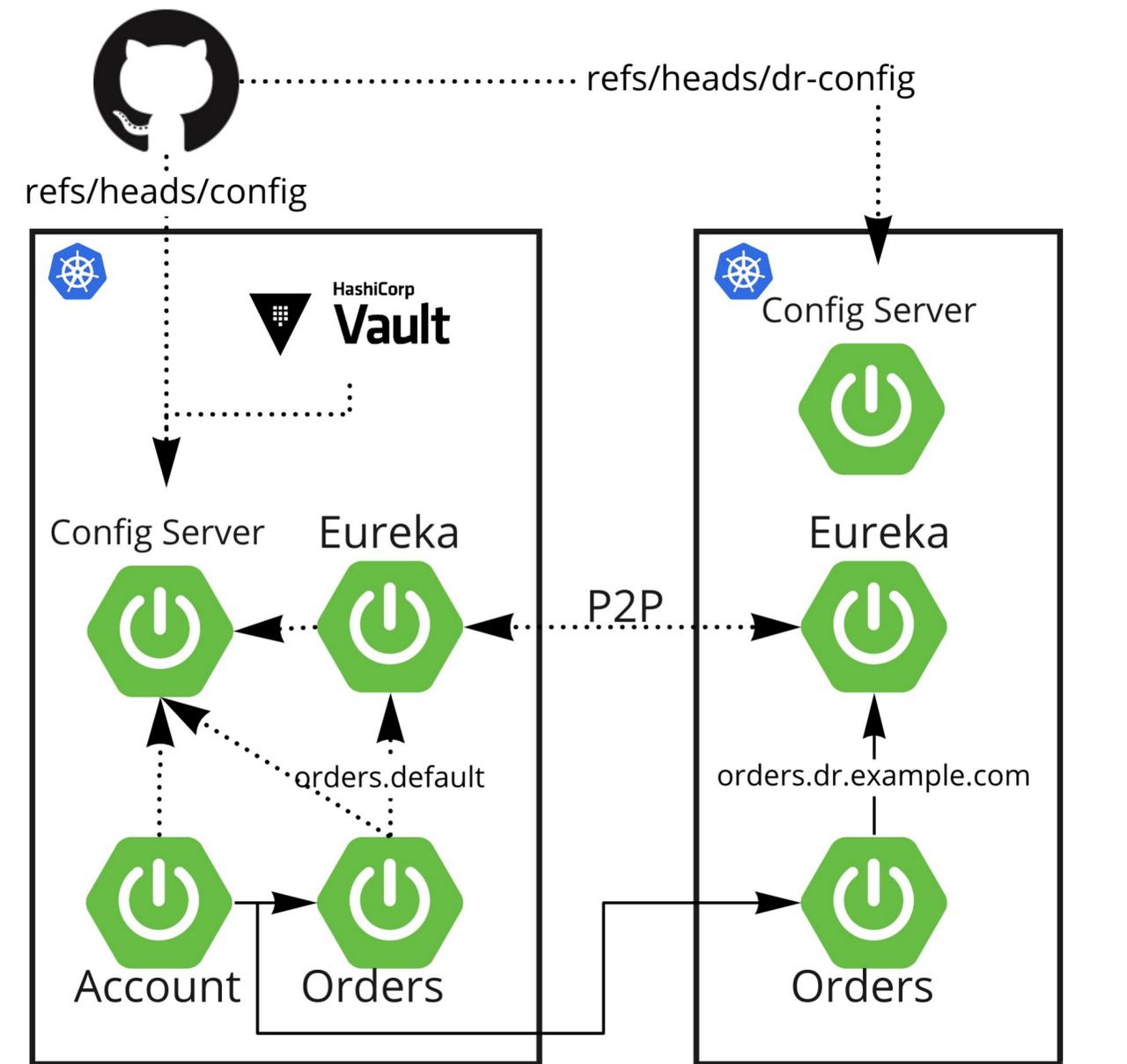
- Разные реализации Resilience4j, Hystrix
- Часть приложения, можно тонко настроить
- Метрики через Micrometer



# Configuration

## Spring Cloud Config Server + Client

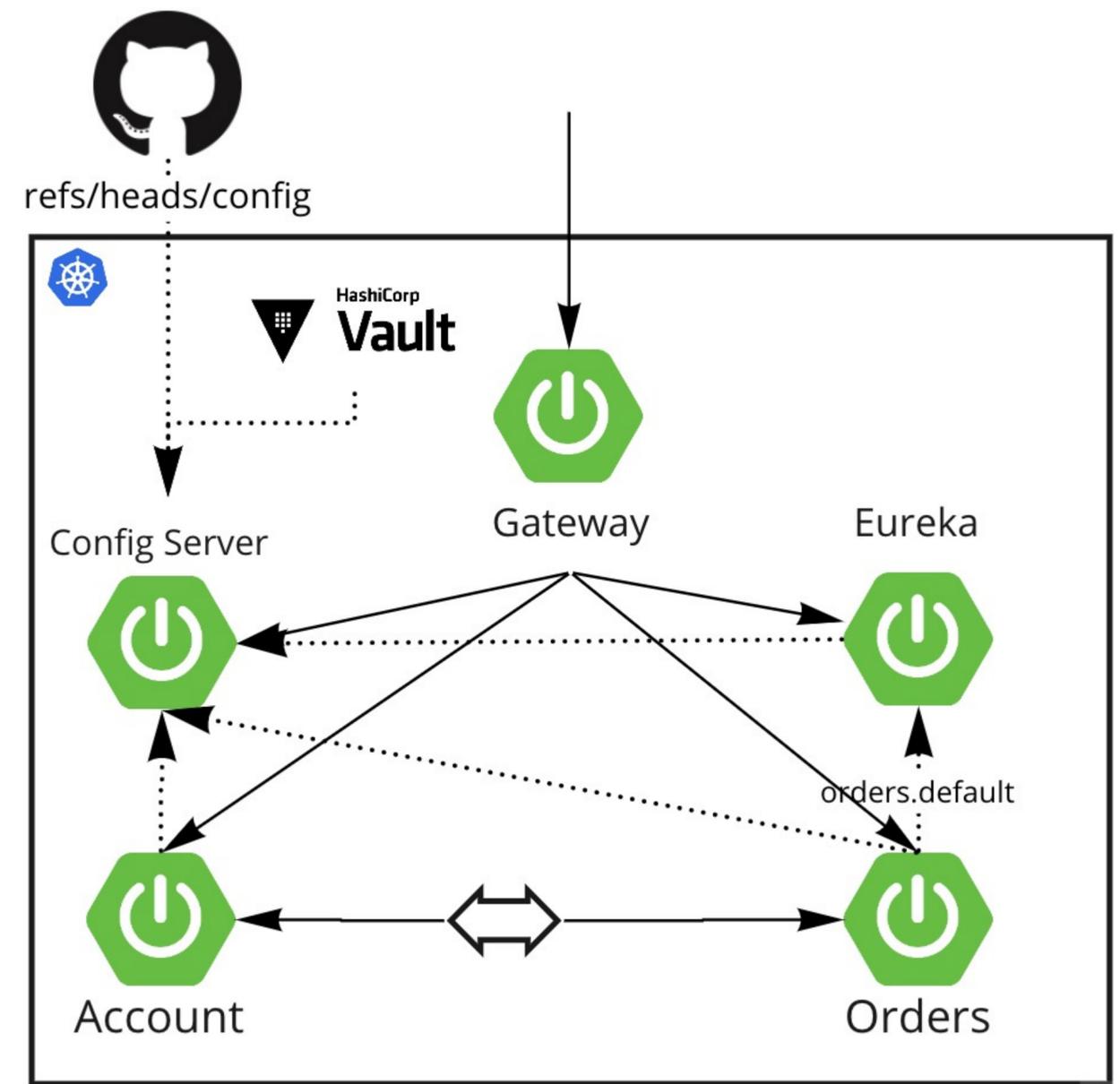
- Централизованное управление конфигураций
- Централизованное и распределенное
- Разные бэкенды - Git, Hg, Vault, Consul
- Профили
- Обновление без рестарта со Spring Cloud Bus + Message Queue

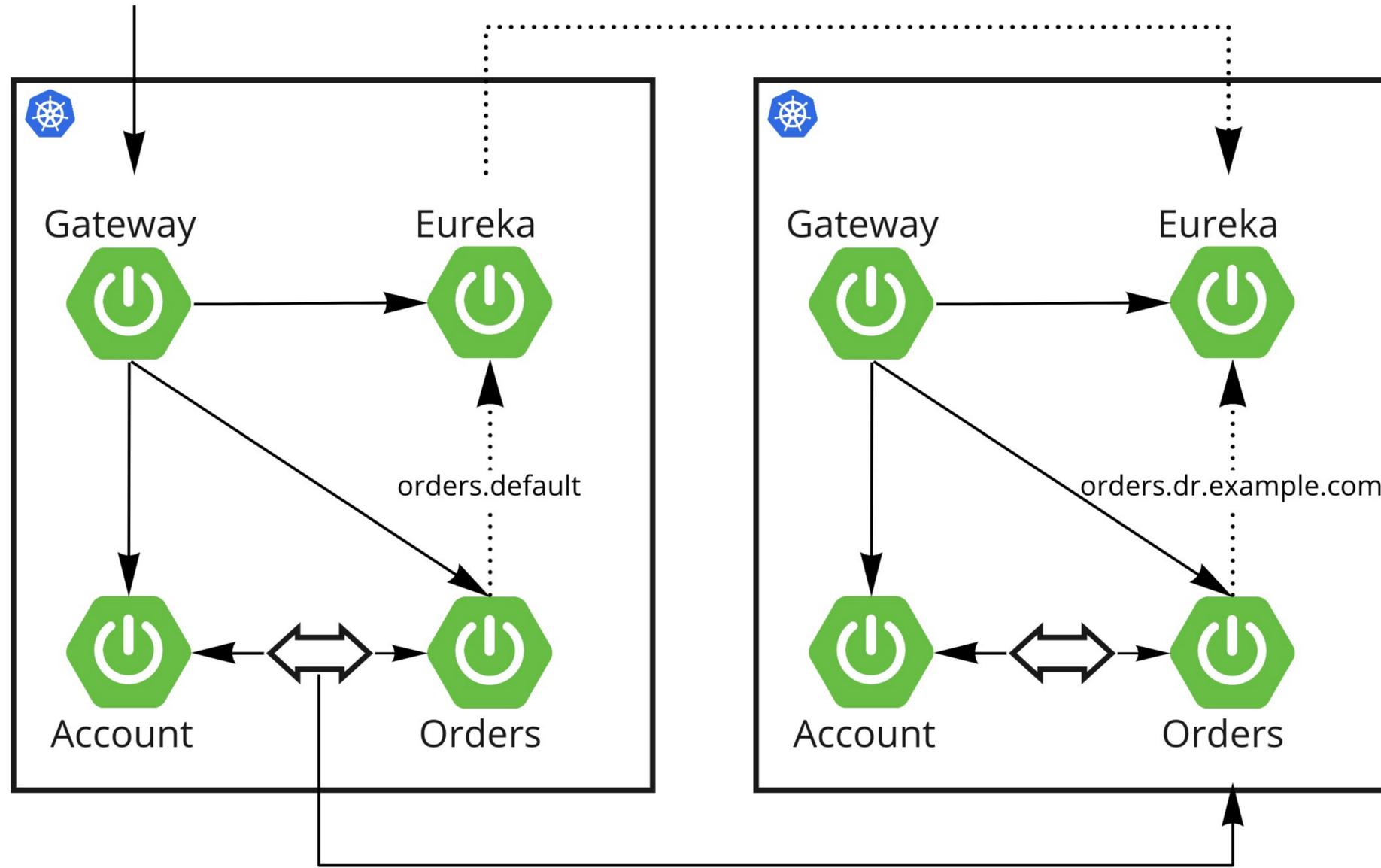


# API Management

## Spring Cloud Gateway

- API Gateway на Java
- Spring Boot приложение
- Non-blocking, reactive приложение
- Интегрирован с Discovery, Load Balancer, Circuit Breaker, etc
- Гибкий роутинг с конфигурацией через YAML, Discovery, Java, REST
- Можно писать сложную логику и трансформации на Java





miro

# Demo

# Официальная поддержка?

- Spring Boot и Spring Cloud уже имеют поддержку K8S
- Для серверных продуктов (**Eureka, Config Server, Gateway**) официальных OCI образов нет
- ...бесплатных. Но есть коммерческие предложения от замечательной, полной талантливых людей команды Spring Cloud Commercial

VMware Spring Cloud Gateway for Kubernetes 1.0 🥳 🎉 🍾

# И что использовать?



**kubernetes**

Для независимых от языка и стека  
базовых функций



**Spring Cloud**

Для тонкой настройки и полного  
контроля микросервисов на Spring Boot

Для надежных, отказоустойчивых и  
масштабируемых приложений для  
счастливых пользователей



**Спасибо!**