

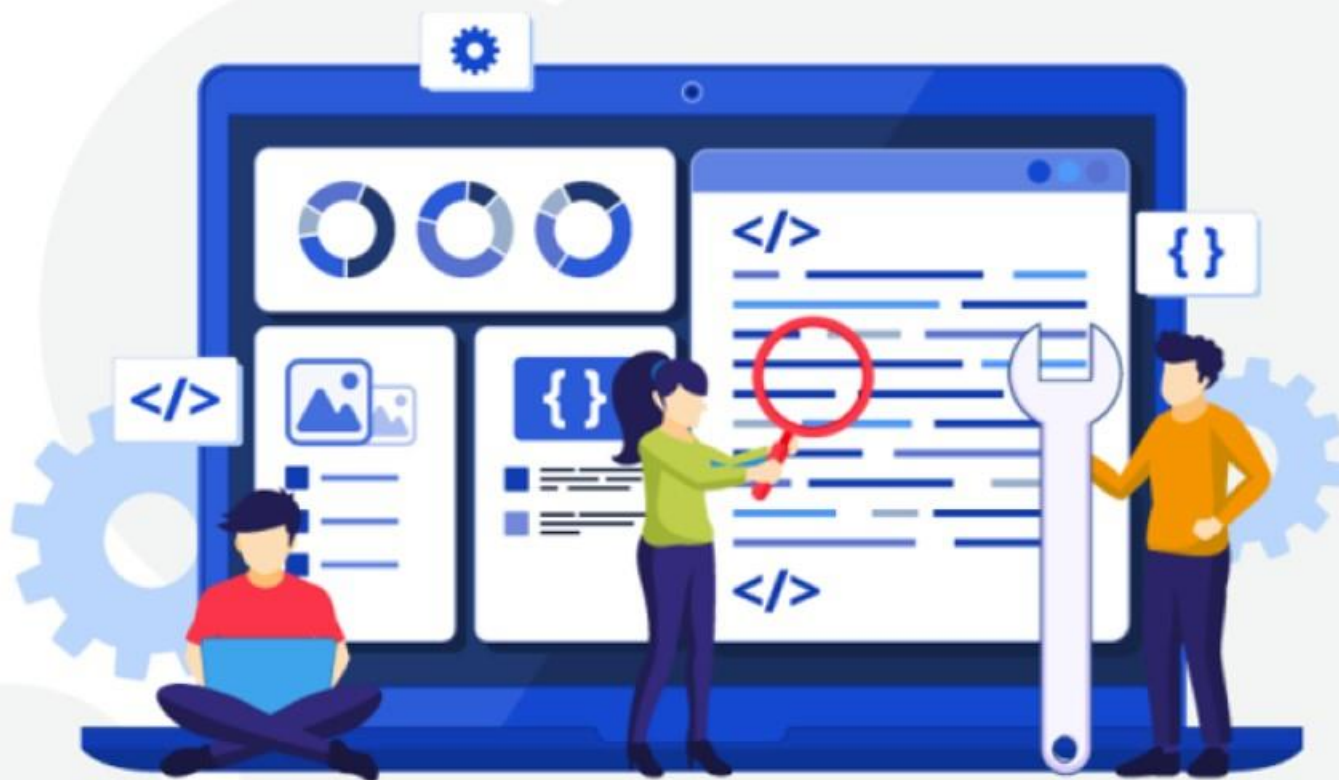


CICLO 4a

[FORMACIÓN POR CICLOS]

## Desarrollo de **APLICACIONES WEB**

Introducción a  
microservicios  
con Netflix OSS



**UNIVERSIDAD  
DE ANTIOQUIA**

Facultad de Ingeniería

# Introducción a microservicios con Netflix OSS

---

Diego Iván Oliveros Acosta

# Agenda

**Generalidades**

**Microservicios con Netflix OSS**

**Components**

- **Eureka**
- **Zuul**
- **Hystrix**
- **Feign**
- **Ribbon**



**Referencias**

# Microservicios Netflix OSS

Netflix OSS es un conjunto de herramientas y componentes de software que permiten desarrollar servicios de manera fácil y rápida.

La arquitectura de Netflix está compuesta por más de 500 microservicios y cuenta con más de 50 millones de suscriptores.

Upstream		Downstream		Aggregate	
BitTorrent	18.37	Netflix	35.15%	Netflix	32.72%
YouTube	13.13	YouTube	17.53%	YouTube	17.31%
Netflix	10.33	Amazon Video	4.26%	HTTP - OTHER	4.14%
SSL - OTHER	8.55			Amazon Video	3.96%
Google Cloud	6.98%	iTunes	2.91%	SSL - OTHER	3.12%
iCloud	5.98%	Hulu	2.68%	BitTorrent	2.85%
HTTP - OTHER	3.70%	SSL - OTHER	2.53%	iTunes	2.67%
Facebook	3.04%	Xbox One Games Download	2.18%	Hulu	2.47%
FaceTime	2.50%	Facebook	1.89%	Xbox One Games Download	2.15%
Skype	1.75%	BitTorrent	1.73%	Facebook	2.01%
	69.32%		74.33%		72.72%



# Algunas opiniones:

- "All teams will henceforth expose their data and functionality through service interfaces" Jeff Bezos
- "que tan grande o pequeño debe ser un microservicio?" y "cual es la diferencia entre microservicios y Arquitectura Orientada a Servicios (SOA)" Martin Fowler y James Lewis



# ¿Por qué microservicios?

- Motivos para el cambio:
  - Disponibilidad (24/7), escalado, velocidad
  - Data center vs AWS
  - Agilizar desarrollo y despliegue
- Timing:
  - 2009 comienzan a migrar de una arquitectura monolítica a una arquitectura cloud basada en MS
  - Para finales 2010 - el front se había movido a AWS
  - Para finales 2011 - todo su monolito en AWS descompuesto en cientos de microservicios
  - 2012 - empiezan a liberar código como Open Source



# Microservicios Netflix OSS

- Separación de responsabilidades
  - Modularidad, encapsulamiento,
  - Escalabilidad
- Escalamiento horizontal
  - Partición de cargas de trabajo
- Virtualización y elasticidad
  - Automatización de operaciones
  - Aprovisionamiento bajo demanda

# NETFLIX

# OSS



# Patterns in Microservices Architecture

## API Gateway

1. Choose to build the application as a set of micro-services.
2. Decide how the application client's will interact with the micro services.
3. With a monolithic application there is just one set of (typically replicated, load-balanced) endpoints.
4. In a micro services architecture, however each micro services exposes a set of what are typically fine-grained endpoints.

## Service Registry

1. Service registry helps to determine the location of service instances to send request to the corresponding service
2. Here, we have used the Netflix Eureka to register a service that are available to be registered in service registry server and it can be identified through the router.

## Service Discovery

1. In a monolithic application, services invoke one another through language-level method or procedure calls.
2. But, in a modern micro services based application typically runs in a virtualized environments where the number of instances of a service and their locations changes dynamically.
3. Each service can be identified using router that are registered with service registry server.



# Micro servicios Netflix OSS



EUREKA



ARCHAIUS



HYSTRIX



TURBINE



ZUUL



BLITZ4J



RIBBON

Operations Component	Netflix, Spring, ELK
Service Discovery server	Netflix Eureka
Dynamic Routing and Load Balancer	Netflix Ribbon
Circuit Breaker	Netflix Hystrix
Monitoring	Netflix Hystrix dashboard and Turbine
Edge Server	Netflix Zuul
Central Configuration server	Spring Cloud Config Server
OAuth 2.0 protected API's	Spring Cloud + Spring Security OAuth2
Centralised log analyses	Logstash, Elasticsearch, Kibana (ELK)

<https://netflix.github.io/>



Eureka



SERVICE  
DISCOVERY

EUREKA

RIBBON

CLIENT IPC

USES

LOAD  
BALANCING

NETFLIX

USES

USES

ZUUL

GATEKEEPER

HYSTRIX

SECURITY

ROUTING

BULK HEAD  
PATTERN

CIRCUIT  
BREAKER

MONITORING



HYSTRIX



# Microservicios Netflix OSS

- **Eureka**
  - Servidor para el registro y la localización de los microservicios.
  - Balanceo de carga y tolerancia a fallos.
  - Su función principal es registrar las diferentes instancias de microservicios existentes.
- **Cómo funciona Eureka**
  - Cada microservicio, durante su arranque, se comunica con el servidor Eureka para notificar que está disponible y luego continuará notificando su estado cada 30 segundos.



## Eureka

# Microservicios Netflix OSS

- **Qué aporta Eureka**

- Abstracción de la localización física de los microservicios
- Conocimiento del estado del ecosistema de microservicios actualizado en todo momento mediante un dashboard
- Configuración como clúster para aumentar la tolerancia a fallos.
- Soporte a multirregión
- Integración con los servicios de AWS



## Eureka

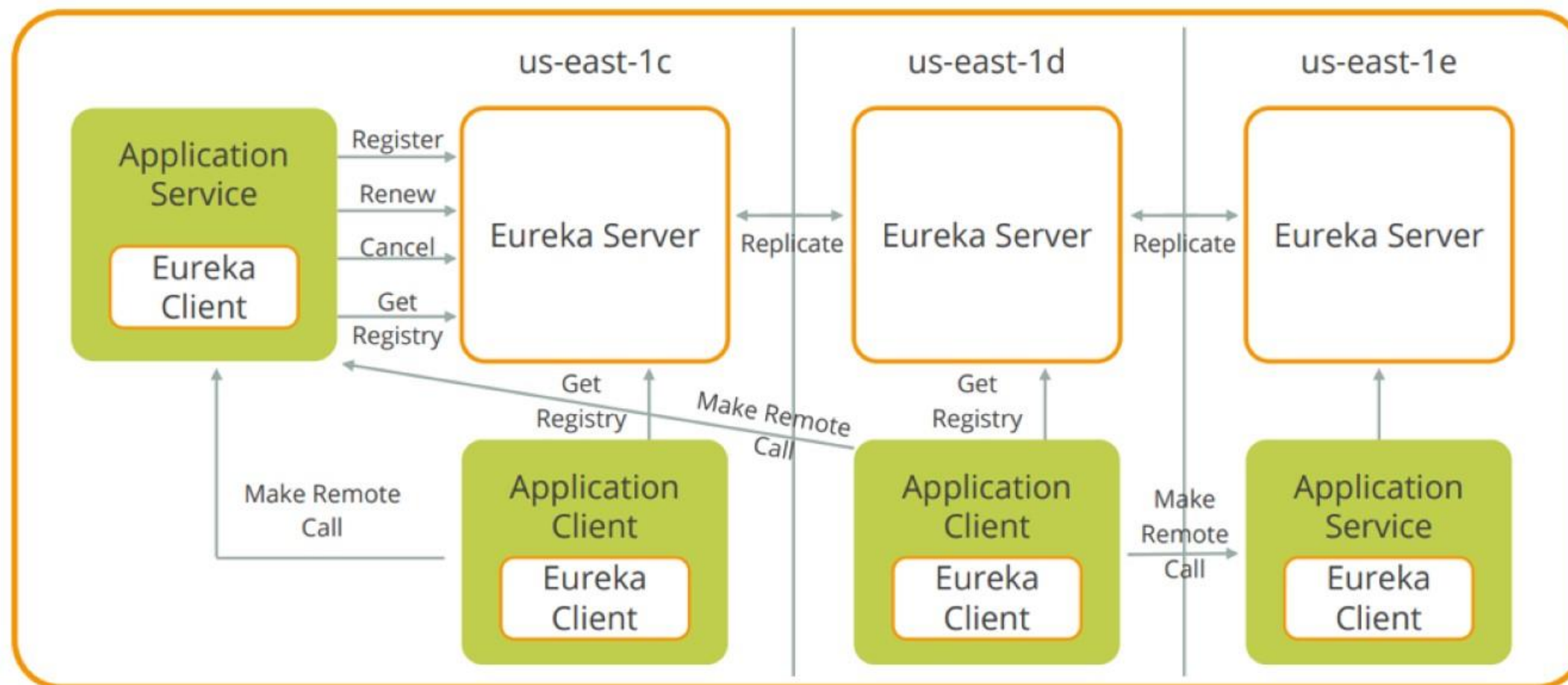


# Microservicios Netflix OSS



Eureka

Qué aporta Eureka



# Microservicios Netflix OSS

- **Zuul**
  - Se puede definir como un proxy inverso.
  - Permite enrutar y filtrar nuestras peticiones.
  - Actúa como un punto de entrada a nuestros servicios.
  - Se encarga de solicitar una instancia de un microservicio concreto de Eureka.
  - Enruta hacia el servicio que consumimos.



# Microservicios Netflix OSS

- **Cómo funciona Zuul**
  - Será configurado como el punto de entrada al ecosistema de microservicios.
  - Es el encargado de enrutar y balancear las peticiones que se reciban de los microservicios.



# Microservicios Netflix OSS

## Qué aporta Zuul

- Proporciona un sistema que reacciona rápidamente cambiando el comportamiento ante diferentes situaciones.
- Proporciona filtros para gestionar diferentes situaciones:
  - Filtros de autenticación
  - Filtros de seguridad
  - Filtros de monitoreo
  - Filtros para enrutados dinámico
  - Filtros para test de carga
  - Filtros para gestión de recursos
  - Filtros para la gestión de multirregión

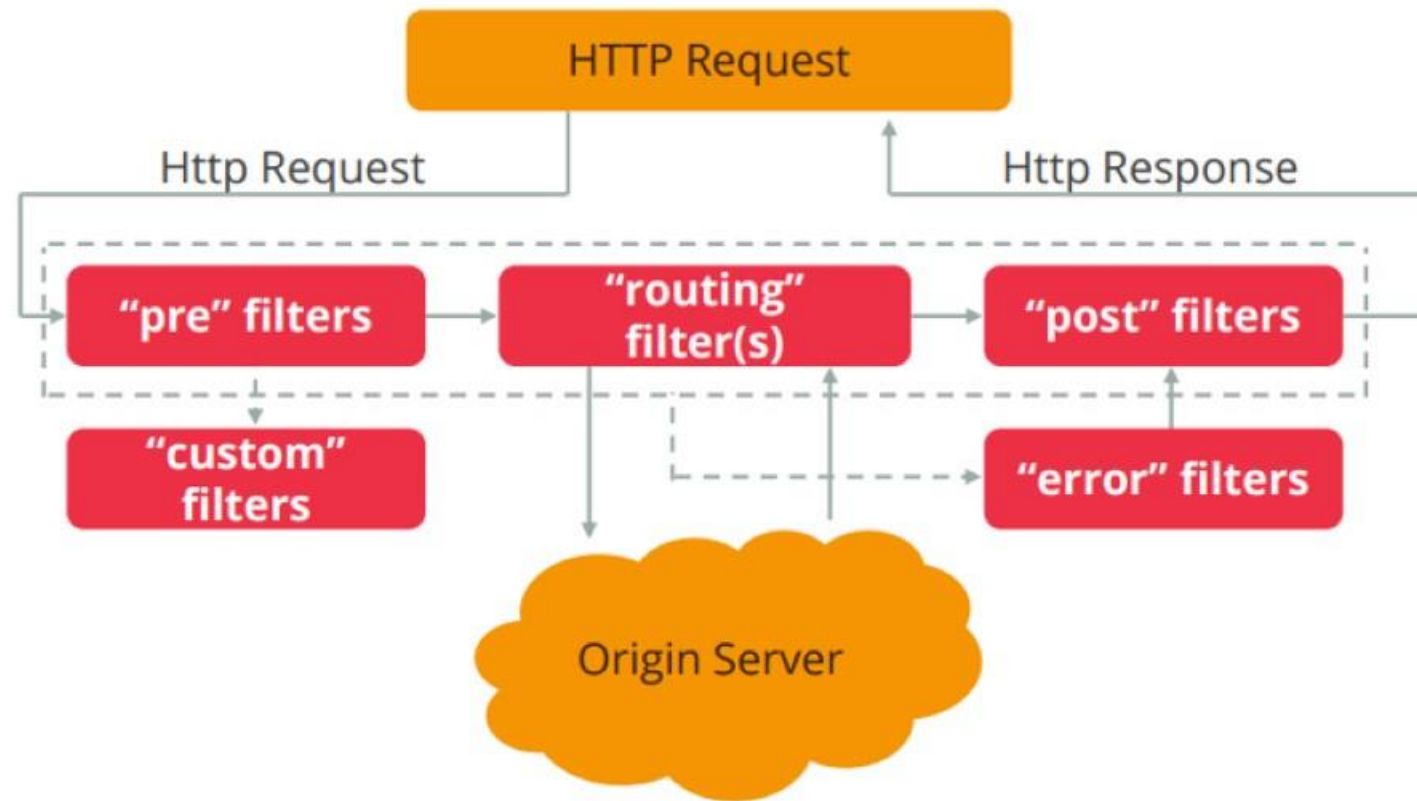




# Microservicios Netflix OSS



Qué aporta Zuul



# Microservicios Netflix OSS

- **Hystrix**
  - Implementa el patrón CircuitBreaker.
  - Permite gestionar las interacciones entre servicios en sistemas distribuidos con lógica de latencia y tolerancia a fallos.
  - Ofrece librería para aislar puntos de acceso a sistemas remotos.
  - Mejora la fiabilidad global del sistema.
- **Cómo funciona Hystrix**
  - Encapsula las peticiones a sistemas “externos” para gestionar aspectos tales como timeout, estadísticos y propagación de errores.



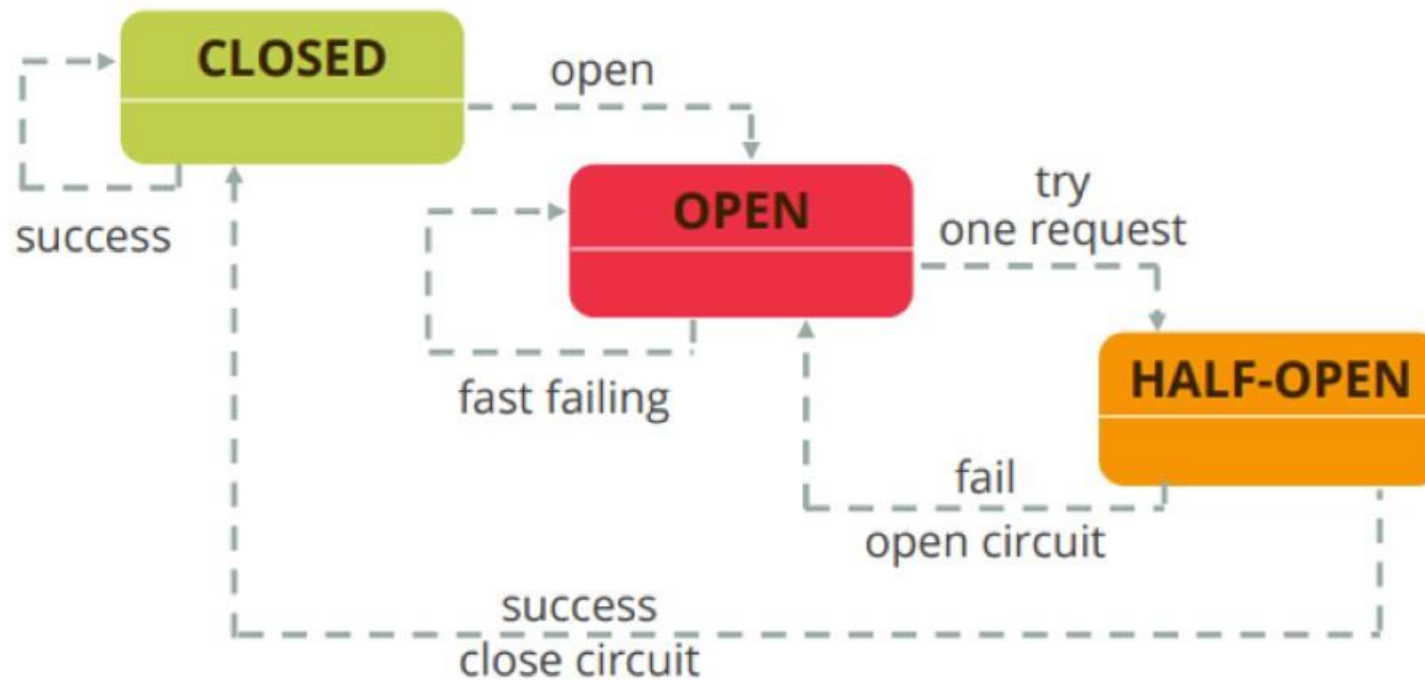
# Microservicios Netflix OSS

- **Qué aporta Hystrix**
  - Encapsula las peticiones.
  - Cancela las peticiones que exceden el timeout.
  - Gestiona pool de hilos para cada petición a sistema “externo”.
  - Gestiona la propagación de errores en cascada.
  - Proporciona un dashboard que integra las métricas capturadas.



# Microservicios Netflix OSS

Qué aporta Hystrix



**Circuit Breaker State Diagram**



# Microservicios Netflix OSS

- **Ribbon**
- Es una librería diseñada para la comunicación entre procesos en la nube que realiza balanceo de carga en el lado del cliente.
- Tiene funcionamiento integrado con Eureka para el descubrimiento de las diferentes instancias de un microservicio.



# Microservicios Netflix OSS

## Cómo funciona Ribbon

- Identifica el microservicio por el nombre con que se registra en Eureka sin ser necesario identificar la máquina, o ip o puerto donde está el microservicio.
- Identifica cuántas instancias existen de un mismo microservicio y en qué máquinas.
- Ejecuta el algoritmo de balanceo de carga Round Robin para determinar qué instancia de microservicio invocar.



# Microservicios Netflix OSS

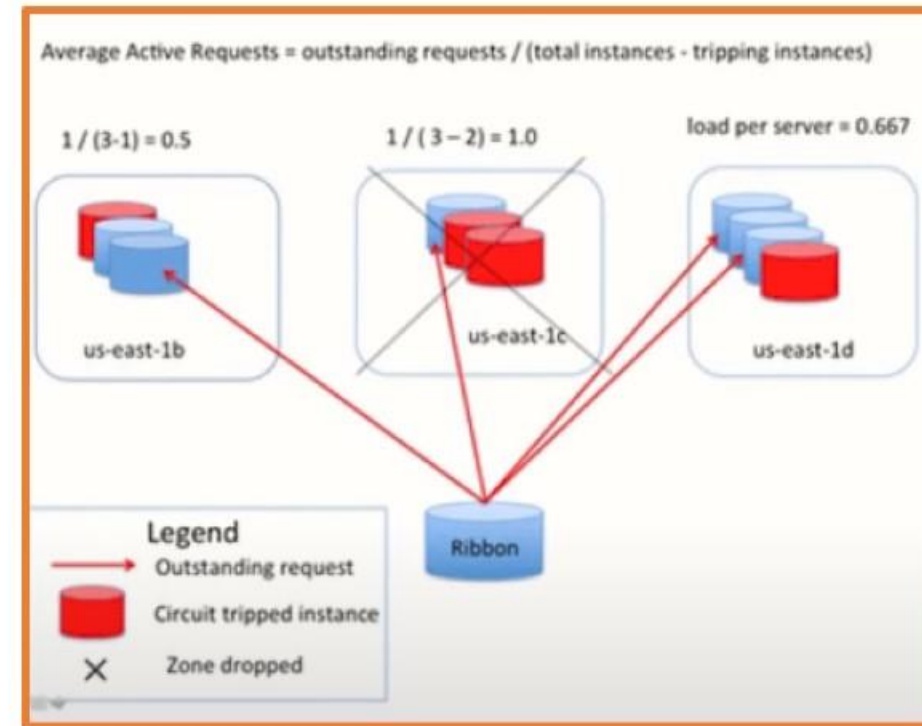
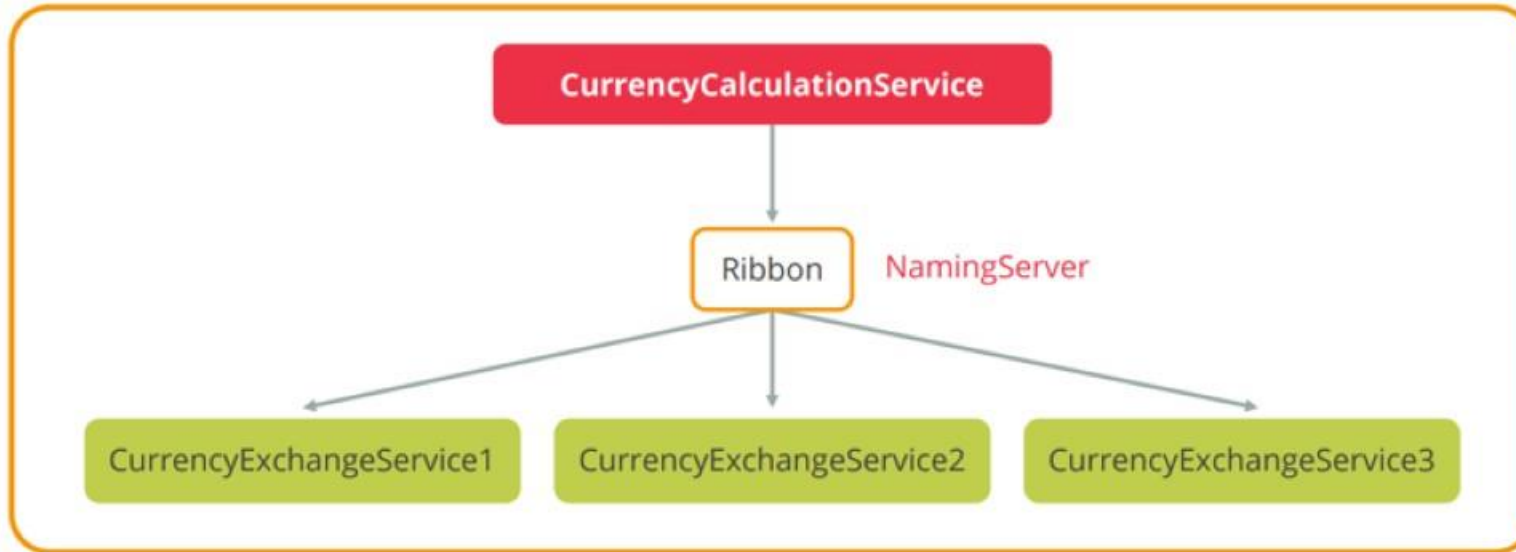
## Qué aporta Ribbon

- Proporciona abstracción del número de instancias existentes.
- Posee diferentes implementaciones de algoritmos de balanceo.
- Permite la gestión de zonas en entornos Cloud.
- Distingue diversas zonas.
- Mediante la integración con Eureka detecta qué instancias de microservicios están caídas.
- Se integra con Hystrix para encapsular las peticiones.



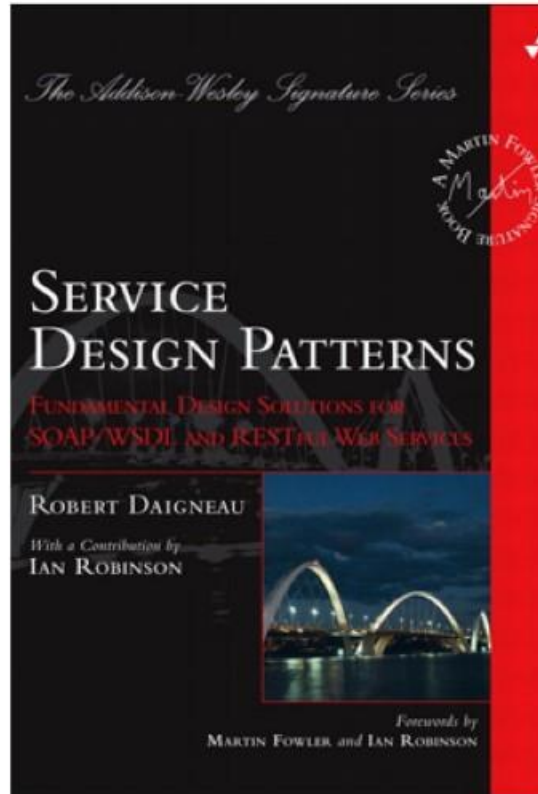
# Microservicios Netflix OSS

## Qué aporta Ribbon





# Referencias



- <https://spring.io/projects/spring-cloud-netflix>
- <https://www.youtube.com/watch?v=uWWKQhpGWPw&t=167s>
- Service Design Patterns
- Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services
- **by Robert Daigneau, with Ian Robinson**
- <https://www.youtube.com/watch?v=wgdBVIX9ifA> 2014

## Referencias

- **Building Microservices: Designing Fine-Grained Systems 1st Edición**
- **Microservices Patterns: With examples in Java**
- **Microservice Patterns and Best Practices: Explore patterns like CQRS and event sourcing to create scalable, maintainable, and testable microservices**
- <https://www.youtube.com/watch?v=uWWKQhpGWPw&t=167s>

