MICROSERVICES ARCHITECTURE

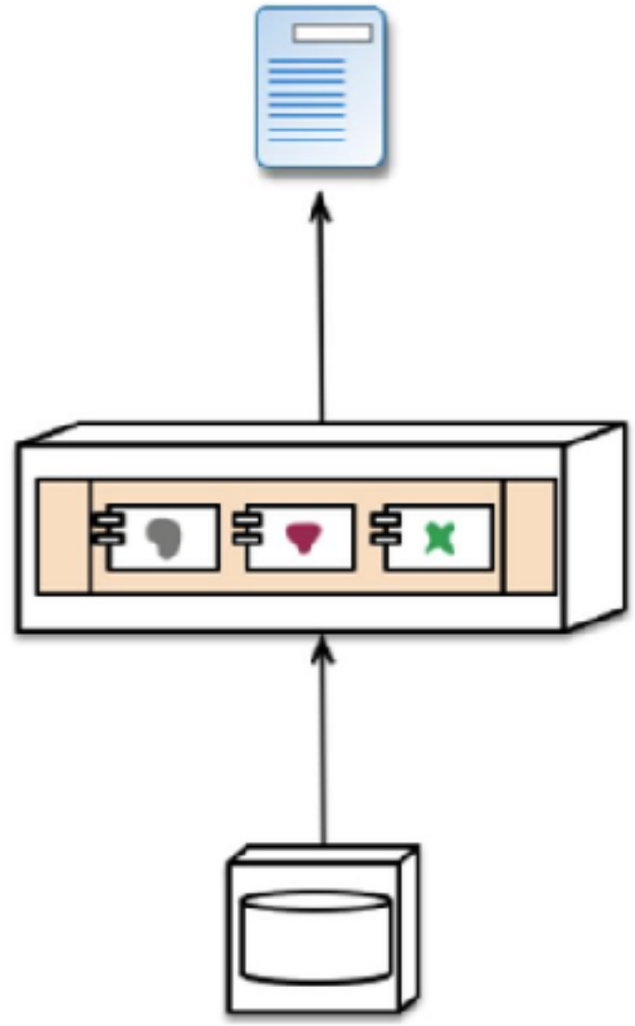
DIGITAL LAB ACADEMY

@edumar111

Aplicaciones Monolíticas

Arquitectura 3 capas

- UI
- App
- DataBase



Manejo de una sola aplicación:

- Desarrollo
- Testing
- Despliegue
- Escalado

http://martinfowler.com/articles/microservices.html

Aplicaciones Monolíticas - Problemas

Funcionales :

- Incremento de Posibles Errores
- Mayores Dependencias -> código más inestable y menos mantenible
- Re-desplegar toda la aplicación.
- Bajo “TIME TO MARKET”

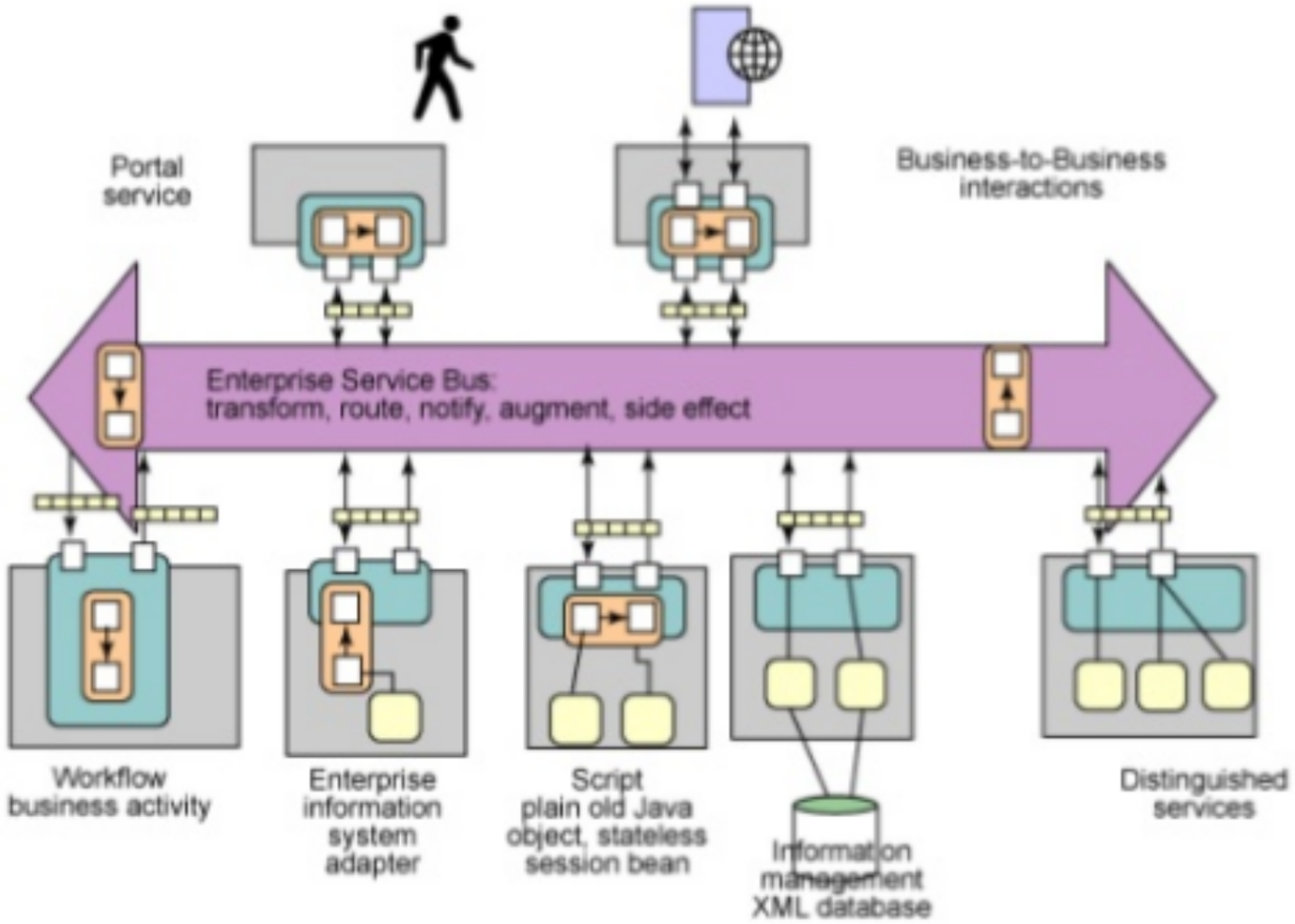
Aplicaciones Monolíticas - Problemas

No Funcionales :

- Escalabilidad -> escalado en su totalidad
- Respuesta frente a fallos -> un error o caída afecta a toda la aplicación

SOA

La arquitectura orientada a servicios. Un servicio es una representación lógica de una actividad de negocio. (Registrar un cliente. Obtener catalogo de Productos.)



Los componentes en SOA se desarrollan de manera más autónoma pero los equipos deben coordinarse con otros para que los cambien encajen en el diseño general.

¿Que son los Microservicios?

La **Arquitectura de microservicios**, es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP). Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos

Microservicios: Ventajas

- **Responsabilidad Unica**: Cada micro servicio hará una cosa y lo hará bien.

- **Eficiencia y Simplicidad**: Servicios mas pequeño y más especializados

- **Heterogeneos**: Cada micro servicio puede ser desarrollado en un tecnología diferente.

- **Alta disponibilidad**: Escalado eficiente, elástico y horizontal en función de la demanda.

- **Independencia**: Cada micro servicio se despliega independientemente, máximo desacople.

- **Time to market**: integración y despliegue Continuo.

Microservicios: Cuando implementarlos

- Cuando hay la necesidad de **reducir el time-to-market**.

- Cuando queremos publicar servicios en **alta disponibilidad y rendimiento**.

- Cuando queramos **centralizar la funcionalidad.**

- Cuando tu **sistema es grande** en:
    - **Funcionalidad**- Difícil implementar mejoras o mantenimiento.

    - **Usuarios**

Casos de Exito

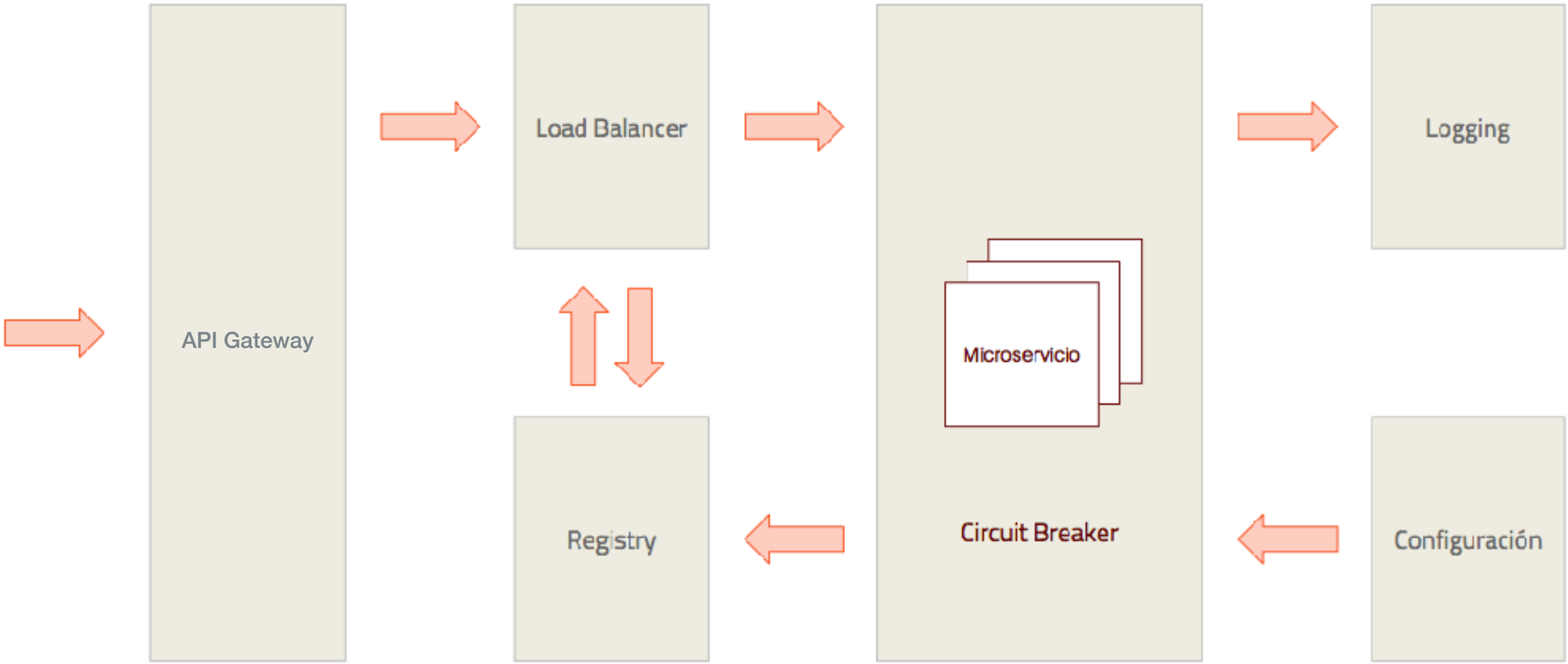
Desafíos

- ¿ Dónde se encuentran desplegados los servicios?
- ¿ Qué pasa si falla un servicio?
- ¿ Cómo se configuran los servicio al desplegarse?
- ¿ Dónde se guardan los logs?
- ¿ Cómo los monitorizamos?
- ¿ Cómo los desplegamos?

¿Qué necesitamos?

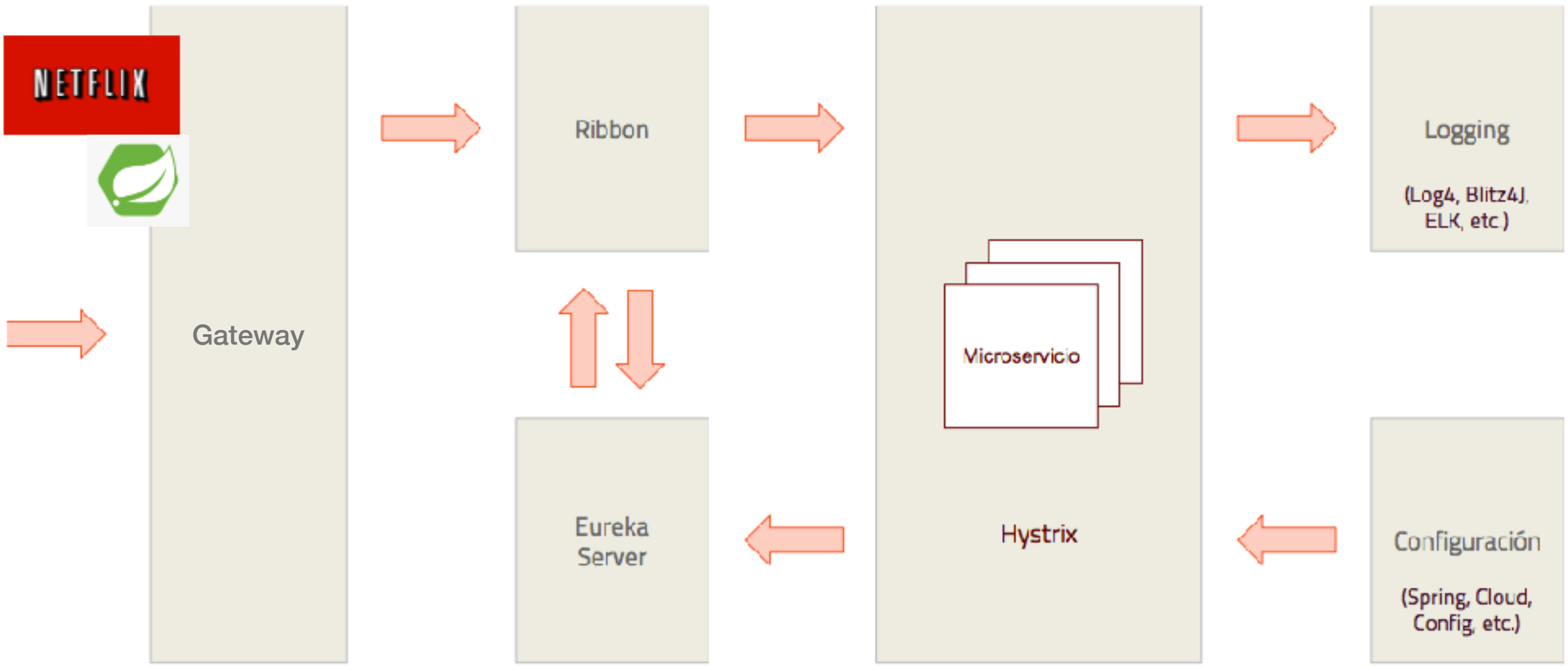
- Registry
- Config
- Load Balancer
- Circuit Breaker
- Api GateWay
- Log Center

Arquitectura



Spring Boot - Spring Cloud

https://spring.io/



https://start.spring.io/

Cloud-Config-Server

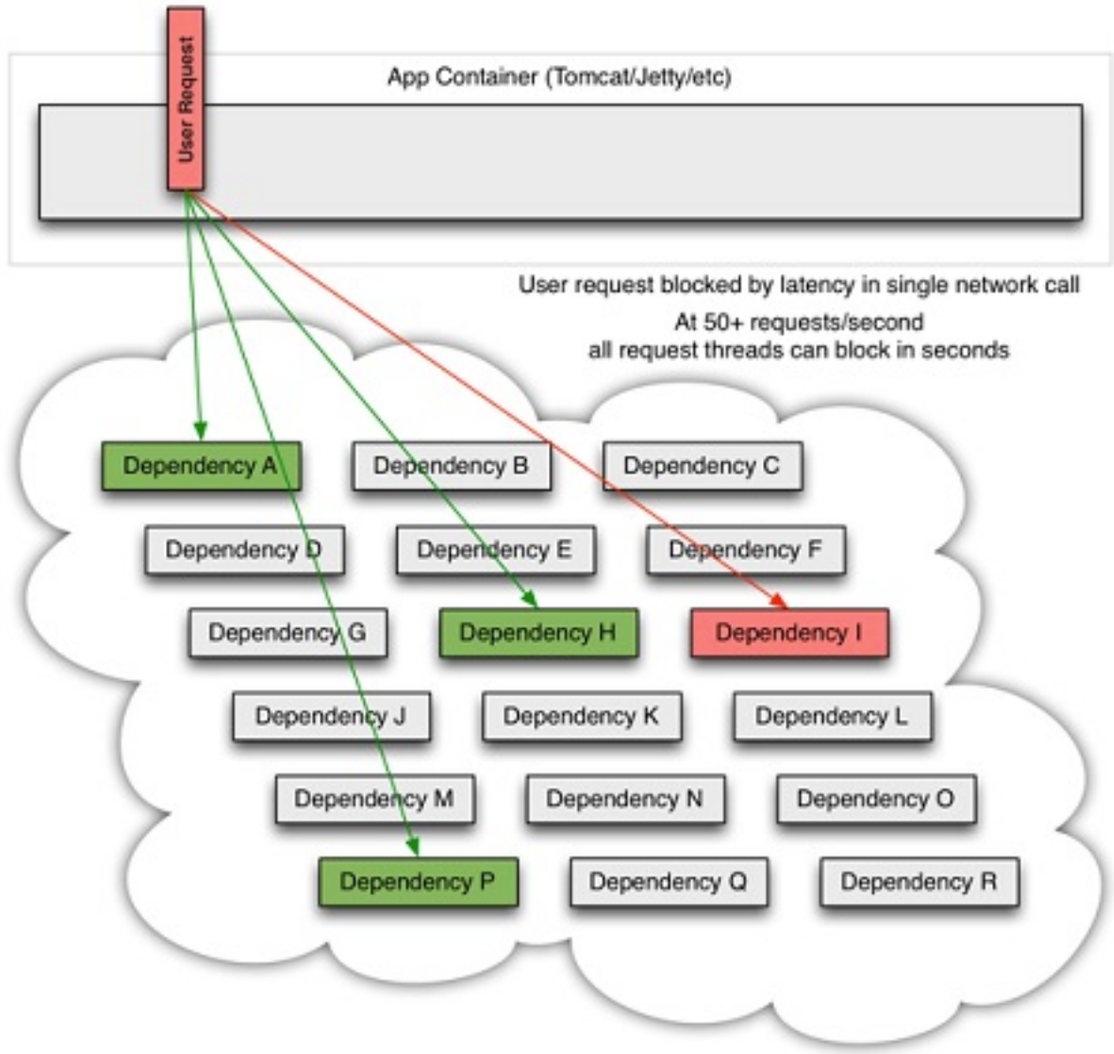
- Servidor de configuración centralizado
- Jerarquia de configuración
- Histórico de configuración
- Centralizar configuración de varios entornos
- Recarga de valores de propiedades en caliente

EUREKA SERVER

- Servidor para registro y localización de instancias de Microservicios
- Cada instancia de MS(eureka-client) notifica ‘heartbeats’ cada 30s
- Cada MS tiene cacheada una copia del registro de Eureka
- Funcionamiento en modo cluster
- Modo Selft-Preservation

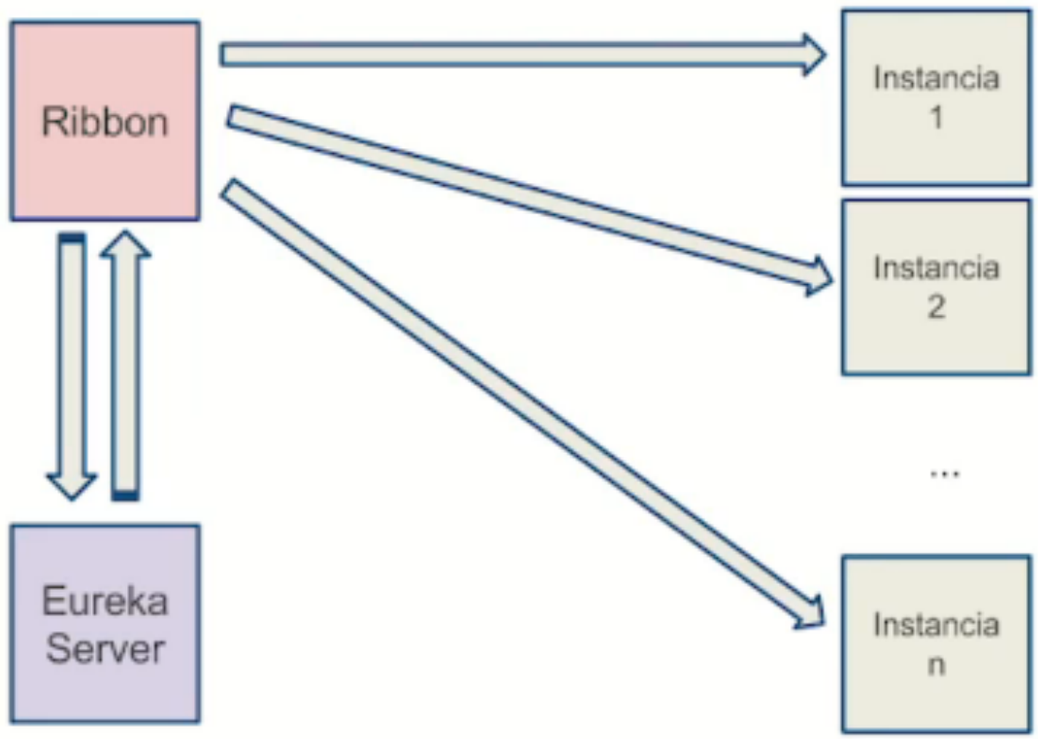
Hystrix

- Patrón circuit-breaker
- Finalidad: mejorar la fiabilidad del sistema
- Control de latency y tolerancia a fallos.



Ribbon

- Balanceo de carga en cliente
- Integración con Eureka
- Integración con Hystrix



Gateway

- Proporciona una puerta de entrada única a todo nuestro ecosistema de micro servicios para peticiones desde dispositivos móviles, sitios web u otros sistemas externos.

- Enrutamiento dinámico, monitorización, seguridad.

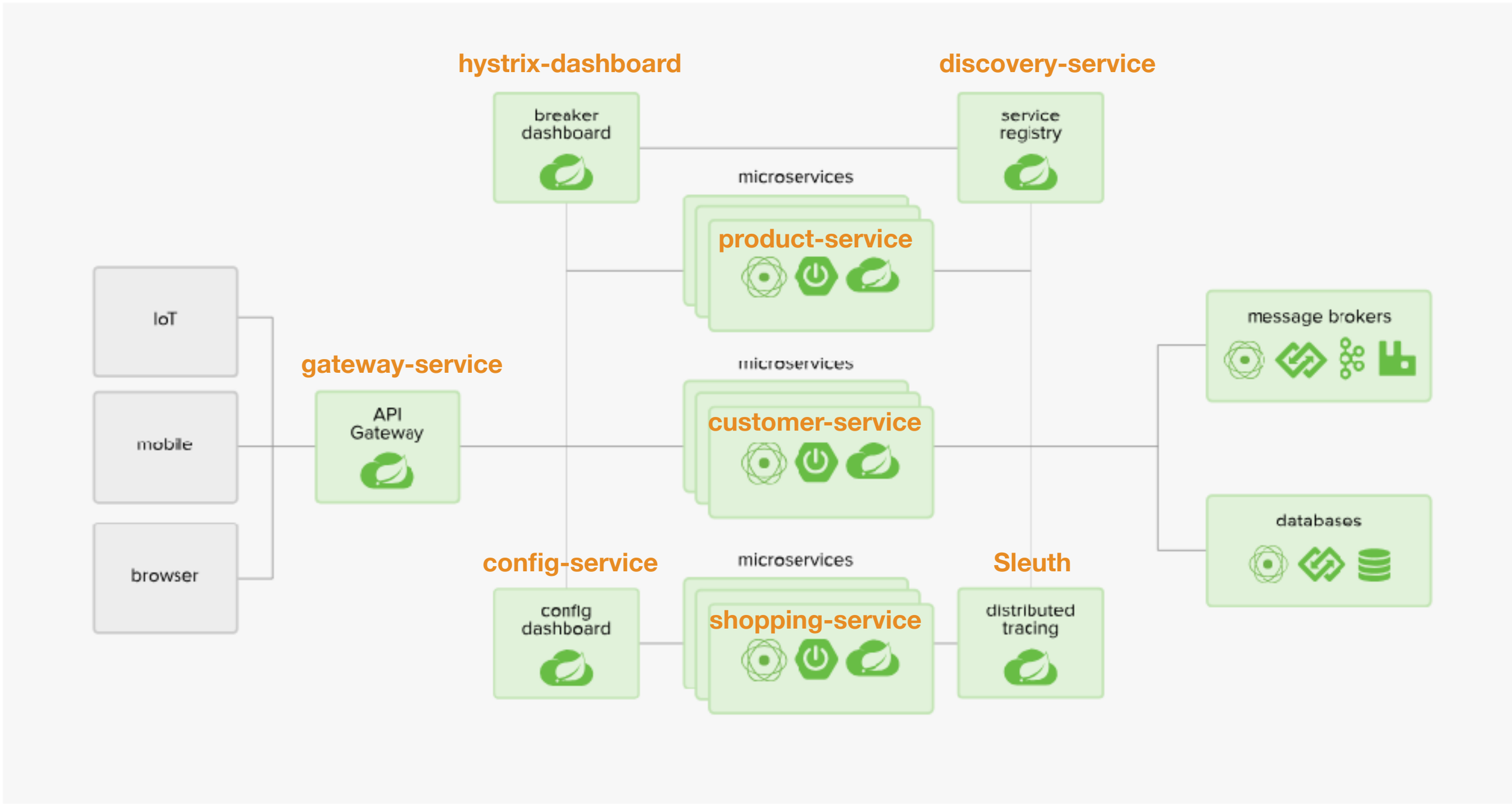
- Carga de Filtros en caliente.

Demo

Vamos por la Demo!!!

https://github.com/edumar111/workshop-microservices-spring.git

Demo - Arquitectura



@edumar111

facebook.com/digitallab.academy