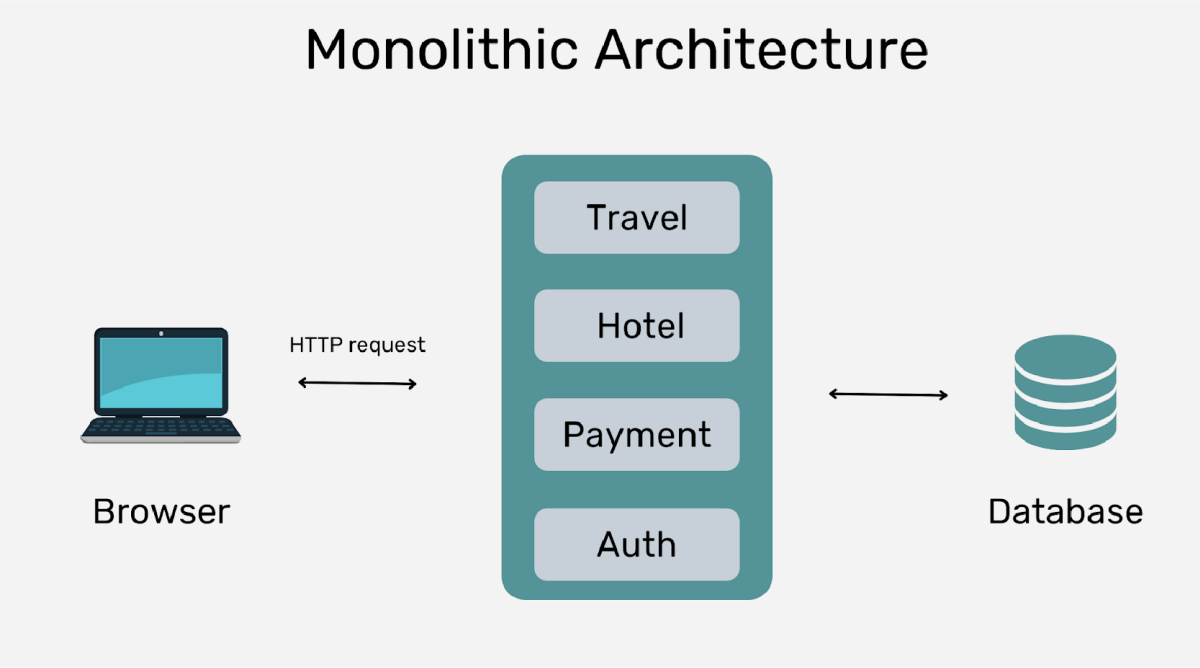
Micro services

Microservices

- What is a Microservice?

- Small autonomous services that work together

A monolithic architecture will most likely have a single codebase or repository for development



Monolithic Architecture

- It’s the conventional and most commonly used architecture in software engineering.

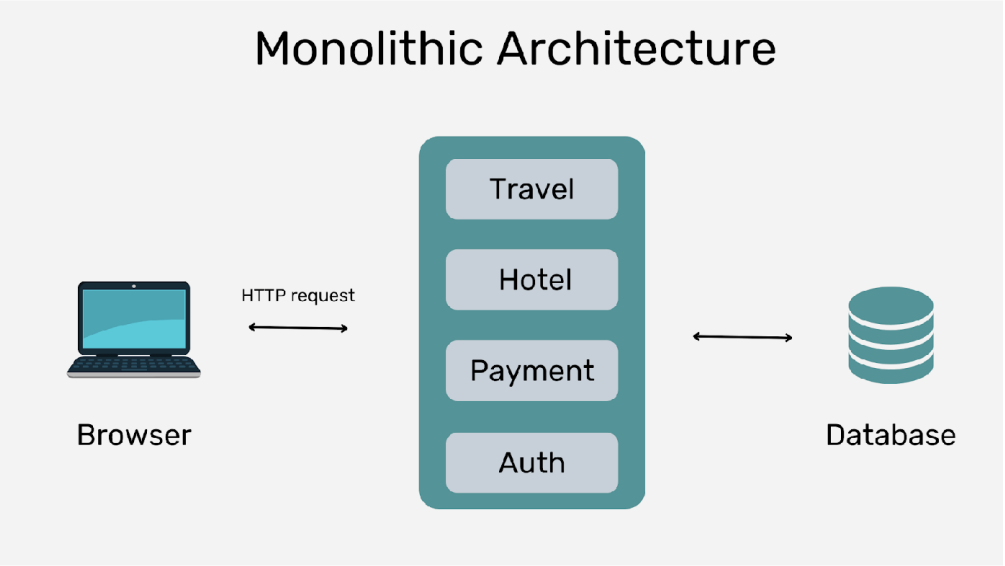
- It advocates that all modules of your software should coexist in one place.

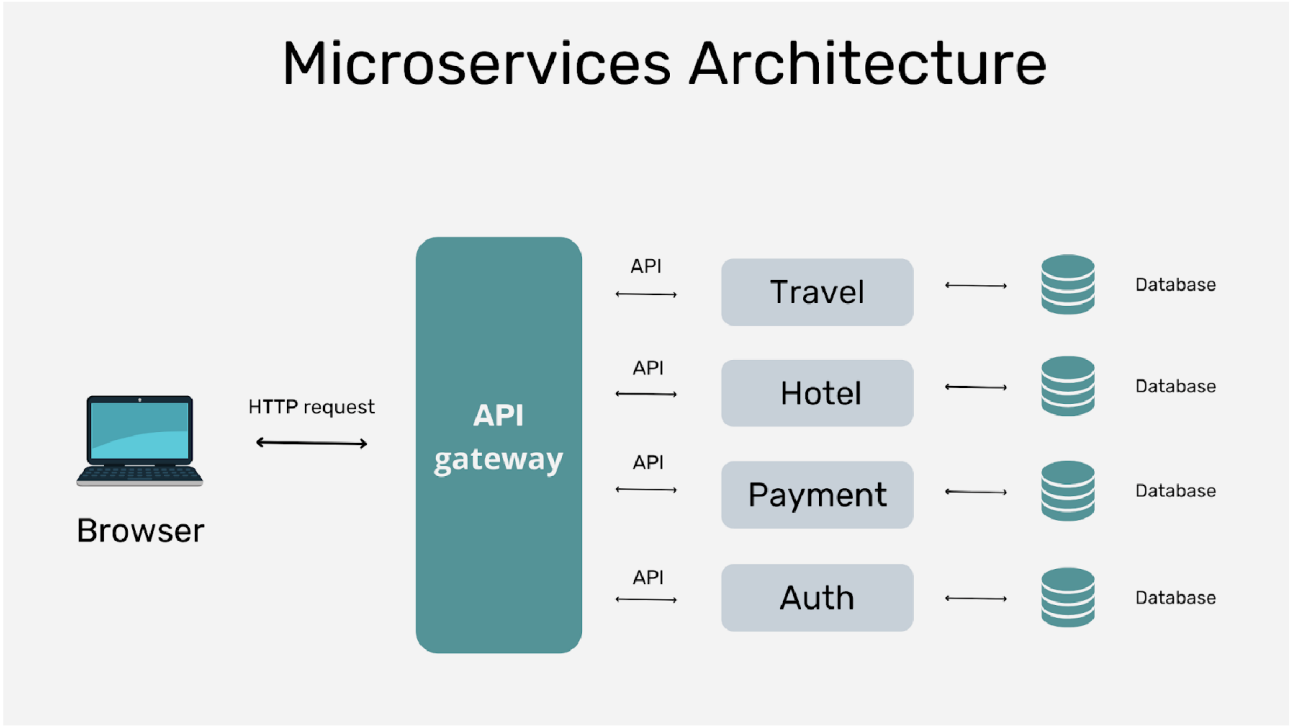
- If a new feature added needs to redeploy and test entire App

Cannot scale certain modules

We need to scale entire app

Which is more cost effective




Microservices Architecture
Browser
HTTP request
API gateway
API
Travel
Database
API
Hotel
Database
API
Payment
Database
API
Auth
Database

What is service-oriented architecture?
Service-oriented architecture was largely created as a response to traditional, monolithic approaches to building applications.

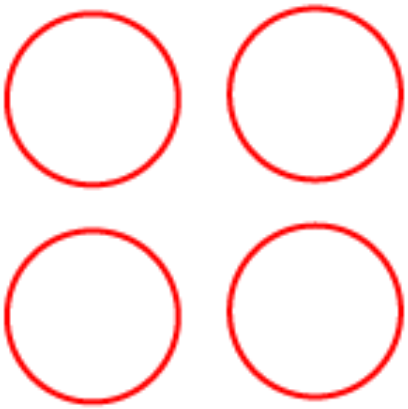
SOA breaks up the components required for applications into separate service modules that communicate with one another to meet specific business objectives.

What is a microservice?
Microservice architecture is generally considered an evolution of SOA as its services are more fine-grained, and function independently of each other.

Monolithic Vs SOA Vs Microservices

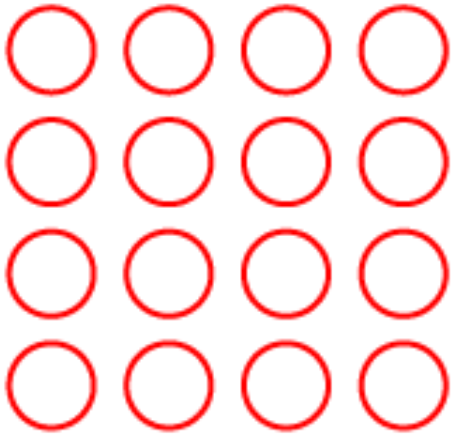
Monolithic

Single Unit



SOA

Coarse-grained

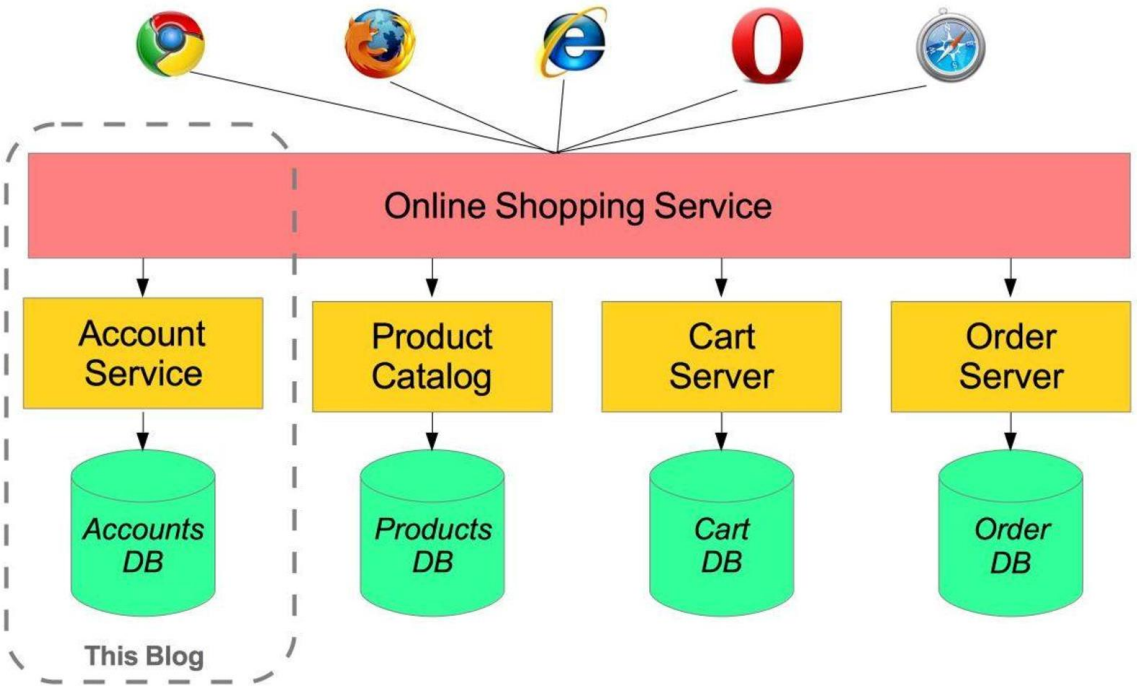


Microservices

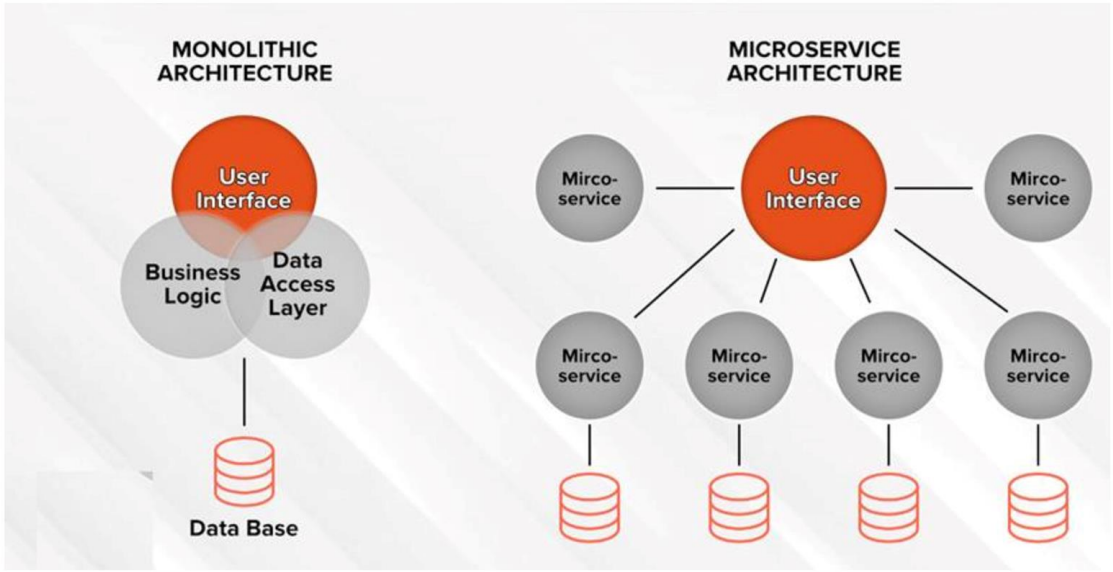
Fine-grained

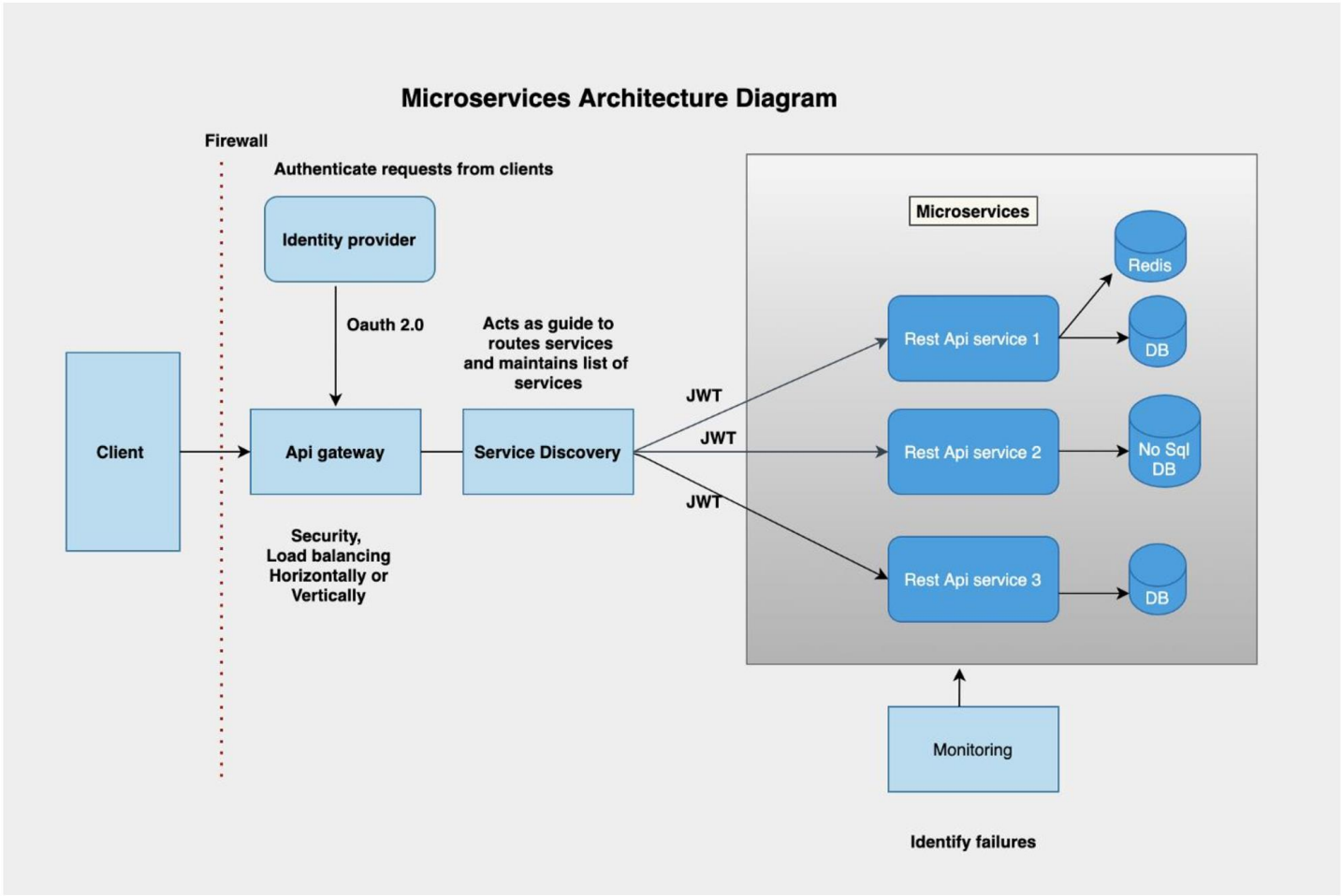
- Microservices advocate breaking the monolithic application into small pieces or components.

- It is an architectural development style in which the application is made up of smaller services that handle a small portion of the functionality and data by communicating with each other directly using lightweight protocols like HTTP


Online Shopping Service
Account Service
Product Catalog
Cart Server
Order Server
Accounts DB
Products DB
Cart DB
Order DB
This Blog

Molithic vs Microservices




Microservices Architecture Diagram
Firewall
Authenticate requests from clients
Identity provider
Oauth 2.0
Acts as guide to routes services and maintains list of services
Client
Api gateway
Service Discovery
Security, Load balancing Horizontally or Vertically
JWT
JWT
JWT
Microservices
Redis
Rest Api service 1
DB
Rest Api service 2
No Sql DB
Rest Api service 3
DB
Monitoring
Identify failures

Microservices - Challenges

- Configuration Management
- Dynamic Scale Up and Scale Down
- Visibility
- Pack of Cards
- Zero Downtime Deployments

Microservice - Solutions

- **Spring Cloud Umbrella Projects**
- Centralized Configuration Management (Spring Cloud Config Server)
- Location Transparency - Naming Server (Eureka)
- Load Distribution (Ribbon, Spring Cloud Load Balancer)
- Visibility and Monitoring (Zipkin)
- API Gateway (Zuul, Spring Cloud Gateway)
- Fault Tolerance (Hystrix, Resilience4j

Ports Standardization

- Limits Microservice 8080, 8081, ...
- Spring Cloud Config Server 8888
- Product Catalouge Service 8000, 8001, 8002, ..
- Order Service 8100, 8101, 8102, ...
- Netflix Eureka Naming Server 8761
- API Gateway 8765
- Zipkin Distributed Tracing Server 9411

Need for Centralized Configuration

- Lot of configuration:
- External Services
- Database
- Queue
- Typical Application Configuration

Feign Client

- Feign is a declarative web service client.

- Microservices talk each other
- Developers need not bother about REST internal calls

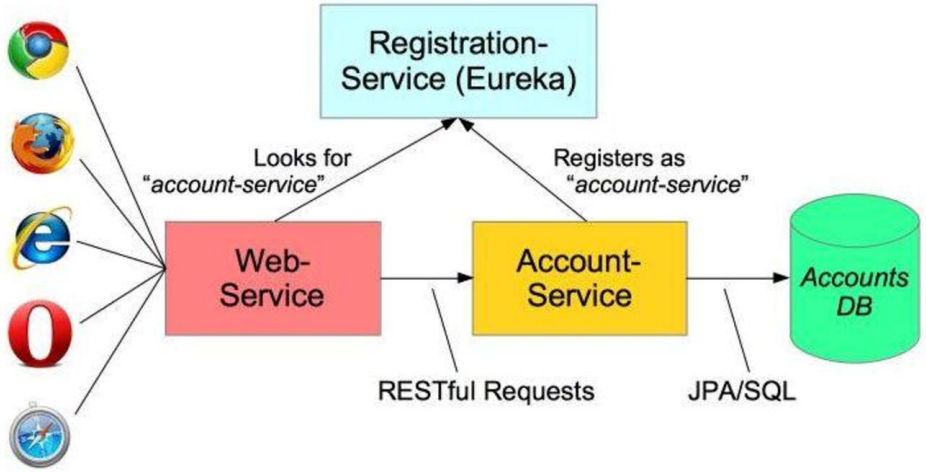
```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-openfeign</artifactId>
</dependency>
```

Eureka Naming Server

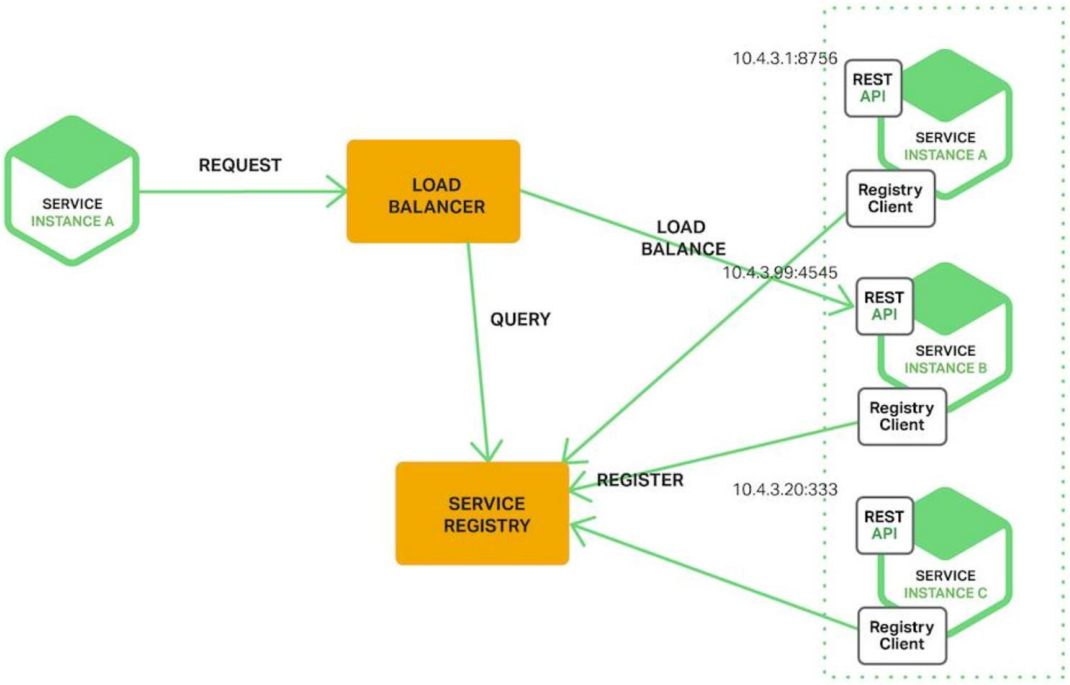
- **Eureka naming server** is a REST-based server that is used in the **AWS Cloud** services for load balancing and failover of middle-tier services.

Important features of naming server

- Service registration
- Service discovery



Eureka Naming server load balancing



Naming server maven dependency

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>

@SpringBootApplication
@EnableEurekaServer
public class Application {

public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}

}
```

To avoid naming server register itself

eureka.client.registerWithEureka=false
eureka.client.fetchRegistry=false

Eureka client maven dependency

```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

Application.properties

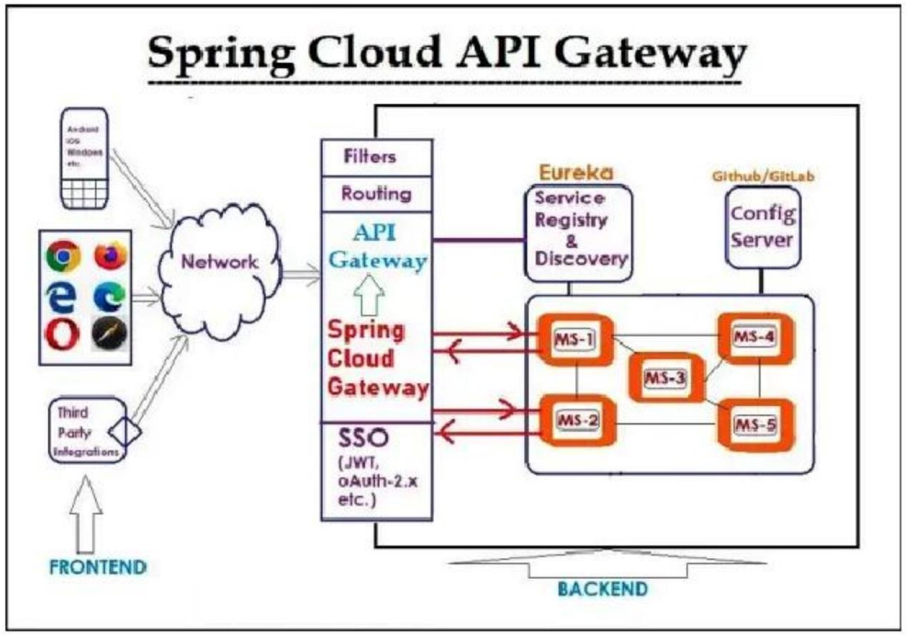
```
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
#default is 8761

spring.cloud.loadbalancer.ribbon.enabled=false
#to disable default client load balancer (ribbon)
```

Spring Cloud Gateway

- Simple, yet effective way to route to APIs
- Provide cross cutting concerns:
- Security Monitoring/metrics
- Built on top of Spring WebFlux (Reactive Approach)
- Features:
- Match routes on any request attribute
- Define Predicates and Filters
- Integrates with Spring Cloud Discovery Client (Load Balancing)
- Path Rewritin

Spring Cloud Gateway



Gateway Maven dependency

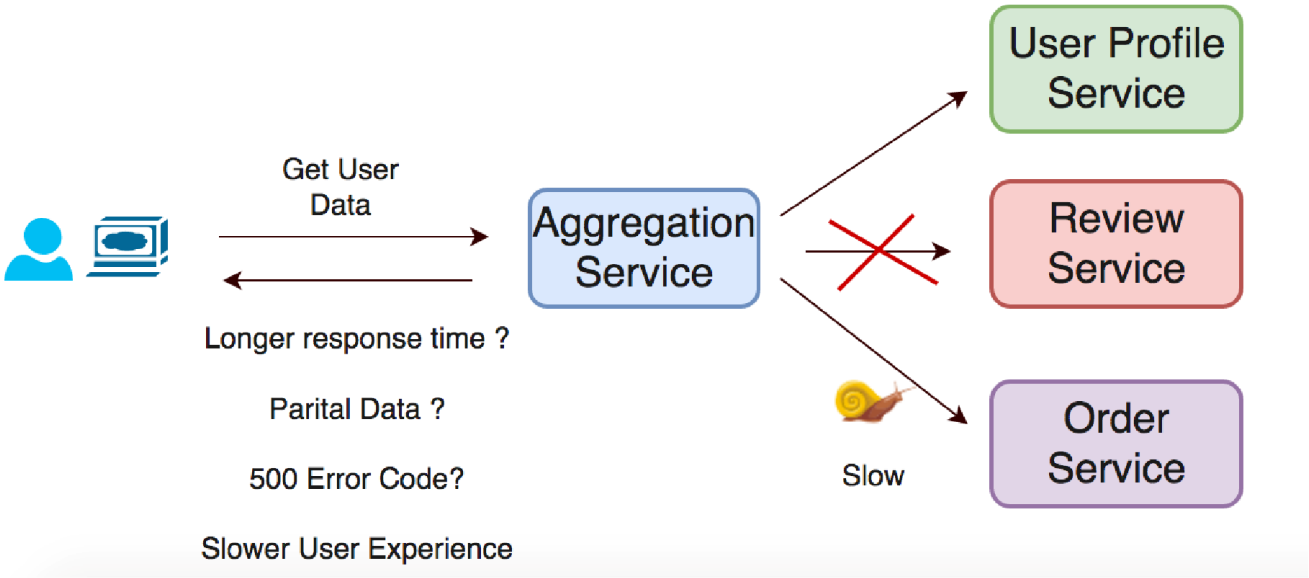
```
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-gateway</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

Application.properties

```
spring.cloud.gateway.discovery.locator.enabled=true
spring.cloud.gateway.discovery.locator.lower-case-service-id=true
```

Circuit Breaker

- What if one of the services is down or is slow?
- Impacts entire chain!



Circuit Breaker Framework - Resilience4

- Solution:
- Circuit Breaker Framework - Resilience4

- Circuit breakers allow the system to handle a few of these failures gracefully. It helps in preventing cascading failures in a complex distributed system and enables resilience in systems where failure is inevitable by enabling to fail fast and rapid recovery.

Circuit Breaker

- There are several open-source libraries available for integrating the circuit breaker :
- Resilience4j
- Netflix Hystrix
- Sentinel by Alibaba
- Failsafe
- Service Mesh like Istio, Linkerd, Cilium etc.