



# Microservices Tutorial

With Spring Boot



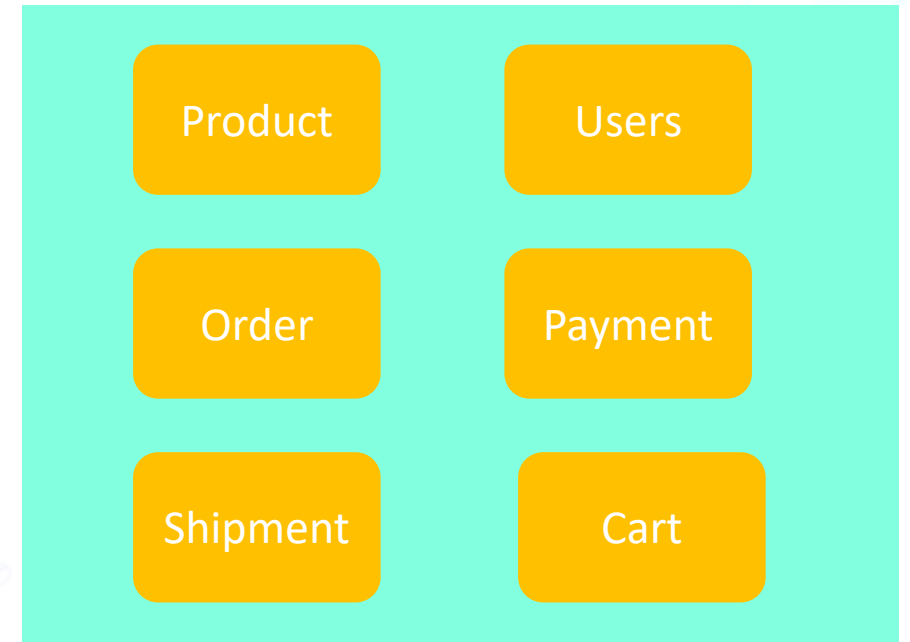
# Prerequisite for Microservices Tutorial

- Spring Boot
- Project using Monolithic Architecture
  - Smart Contact manager
  - Exam Portal
  - Blog Application



# Before Microservices

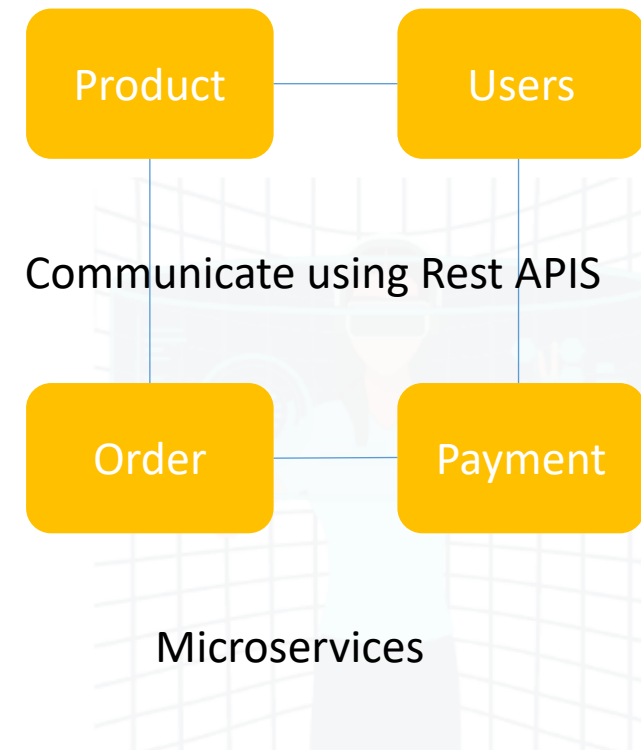
- Monolithic architecture: multiple components are combined in single large app.
- Single Code Base
- Deployed in single bundle
- Change in one service then whole app is redeployed
- Building problem: developers has to communicate
- Problem in scale
- Cumbersome over time

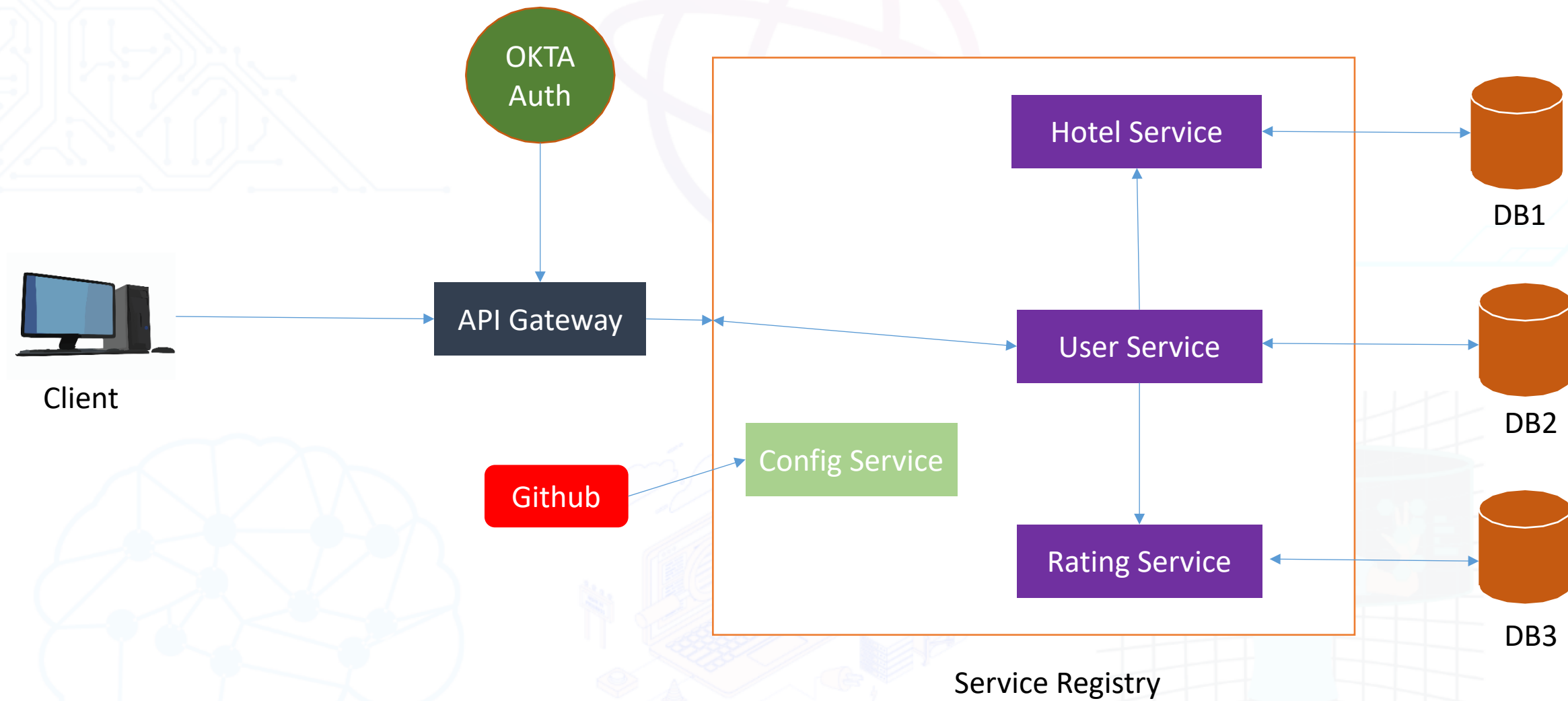


Application using Monolithic

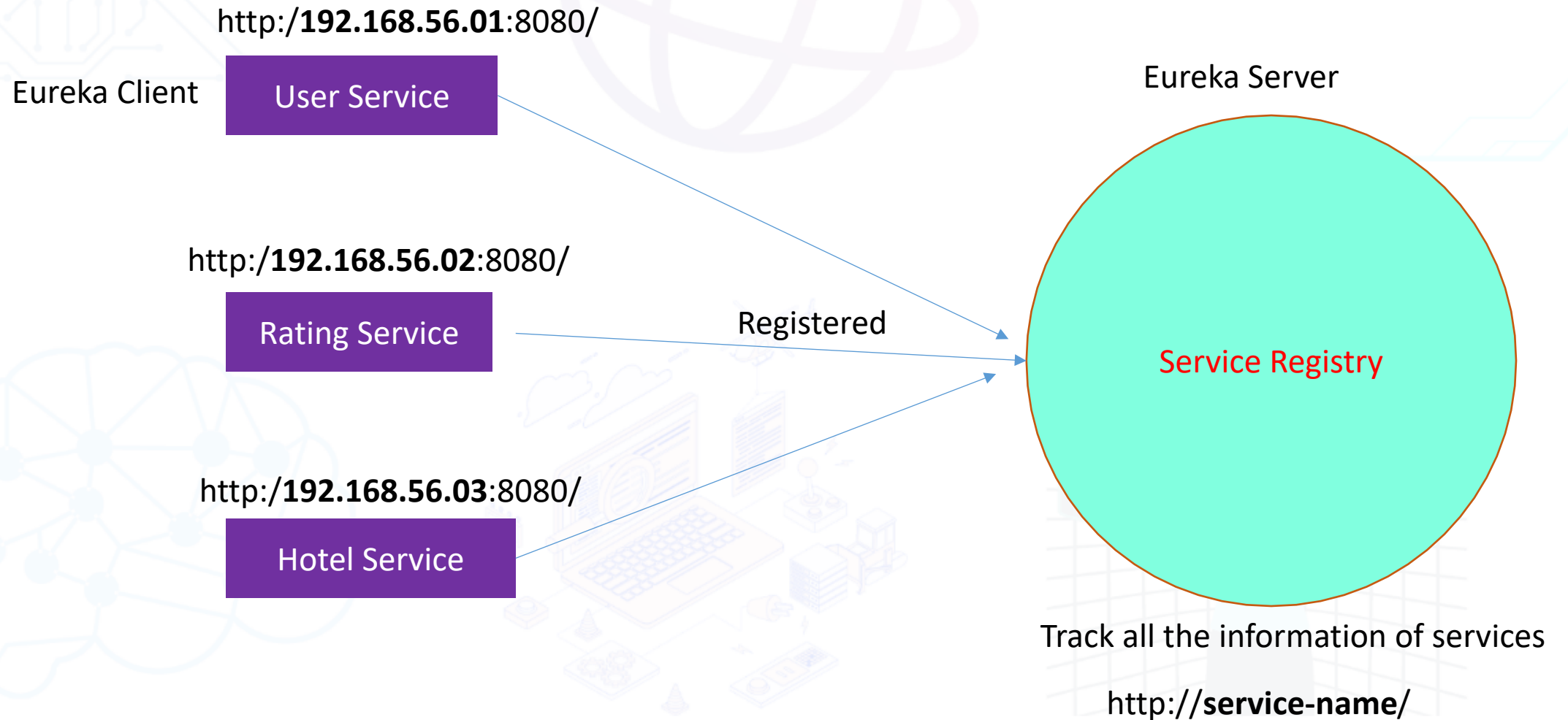
# Microservices

- Large apps are divide into small parts
- Different codebase
- Each module managed independently
- Different tech stack
- **Handling microservices is complex**

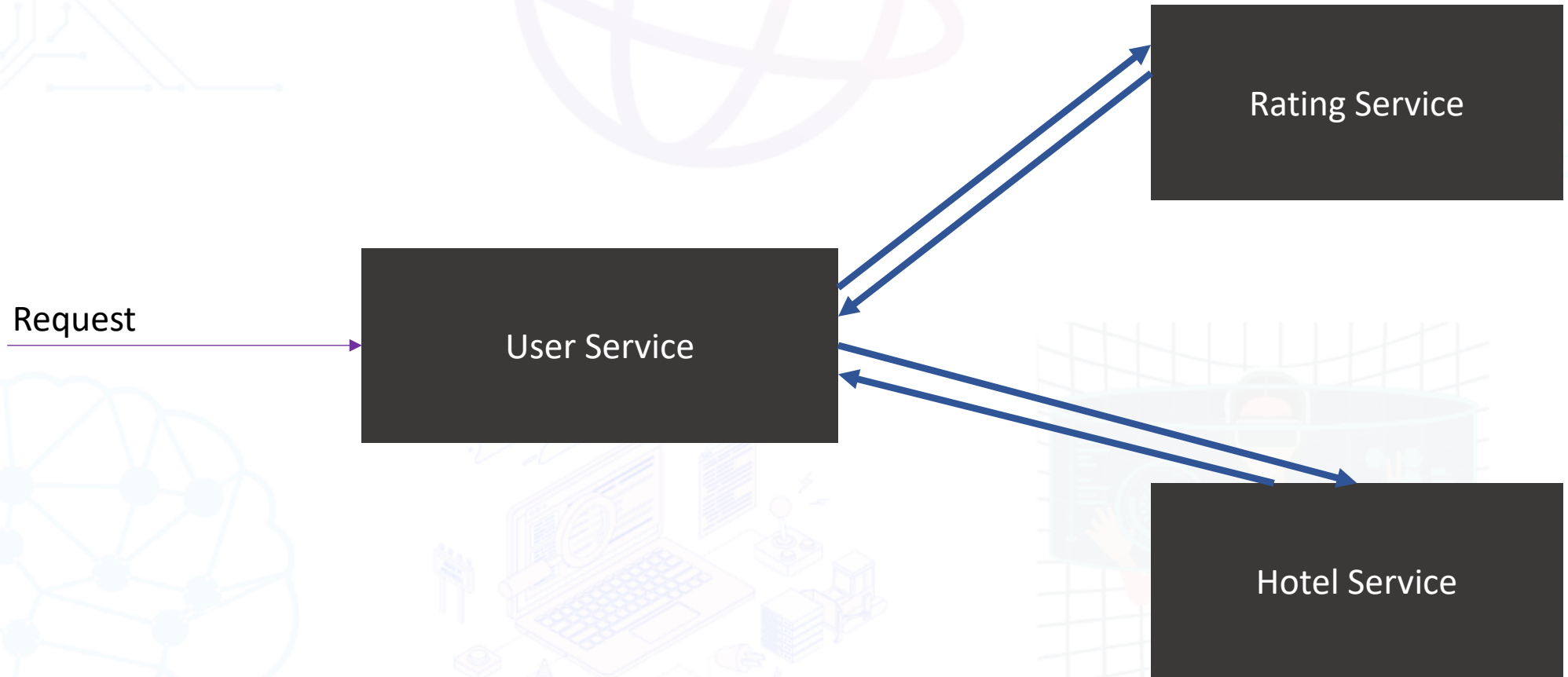


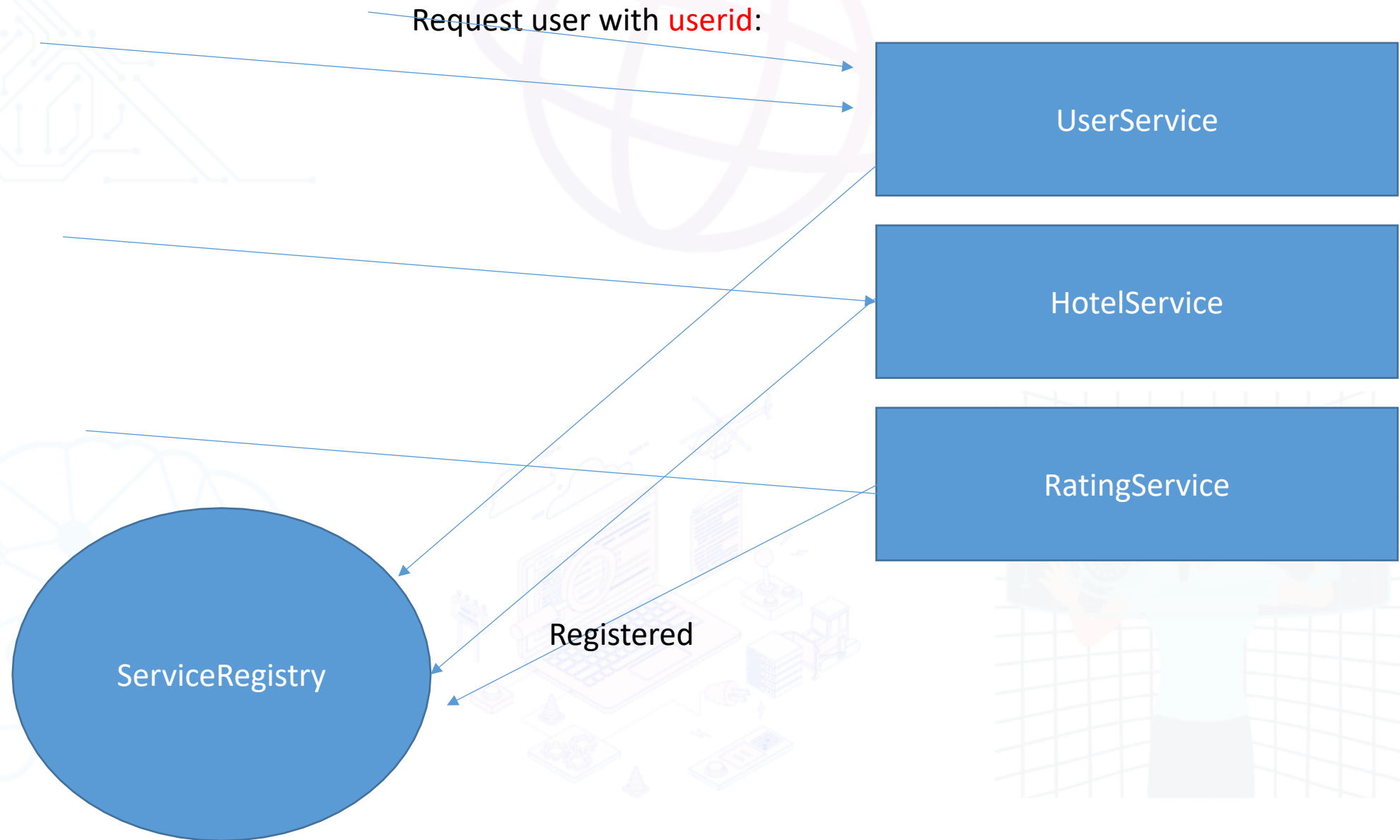


# Service Registry



# Lets start building microservices



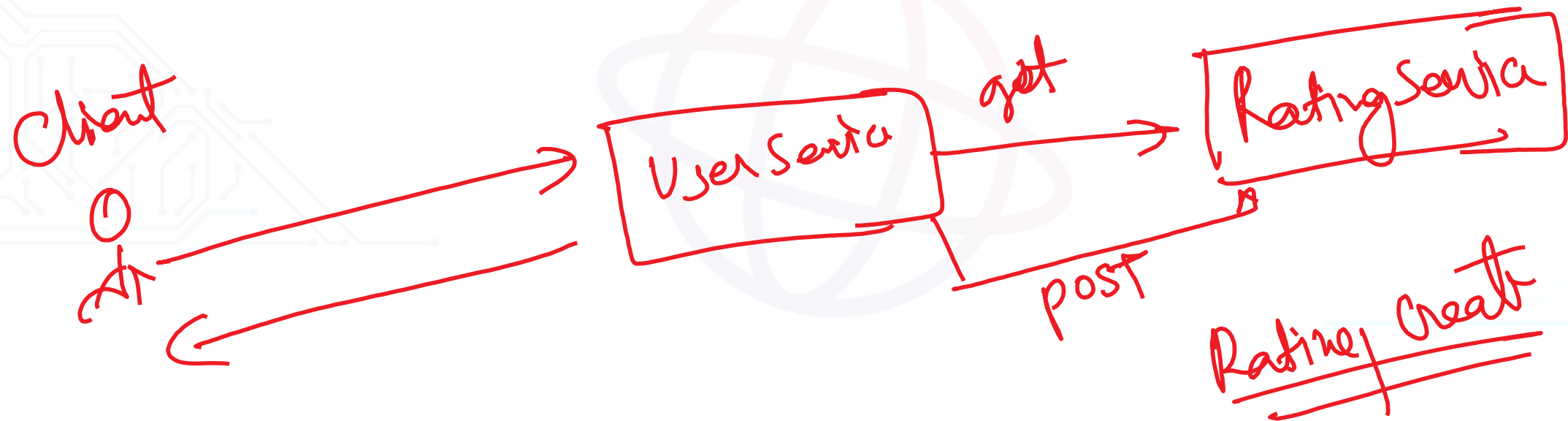


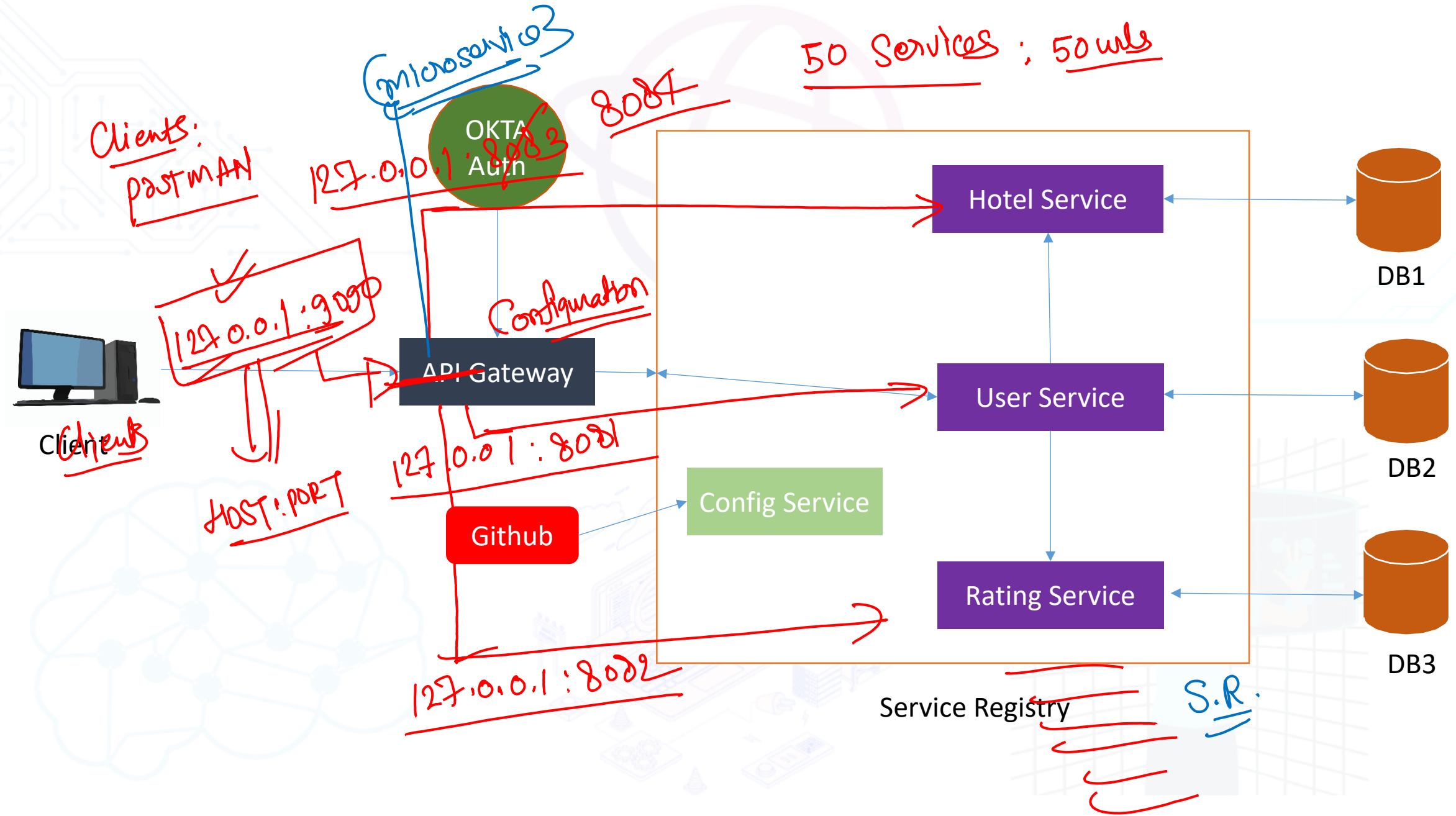


# Feign Client

- The Feign is a **declarative HTTP web client** developed by **Netflix**.
- If you want to use Feign, **create an interface**, and **annotate it**.



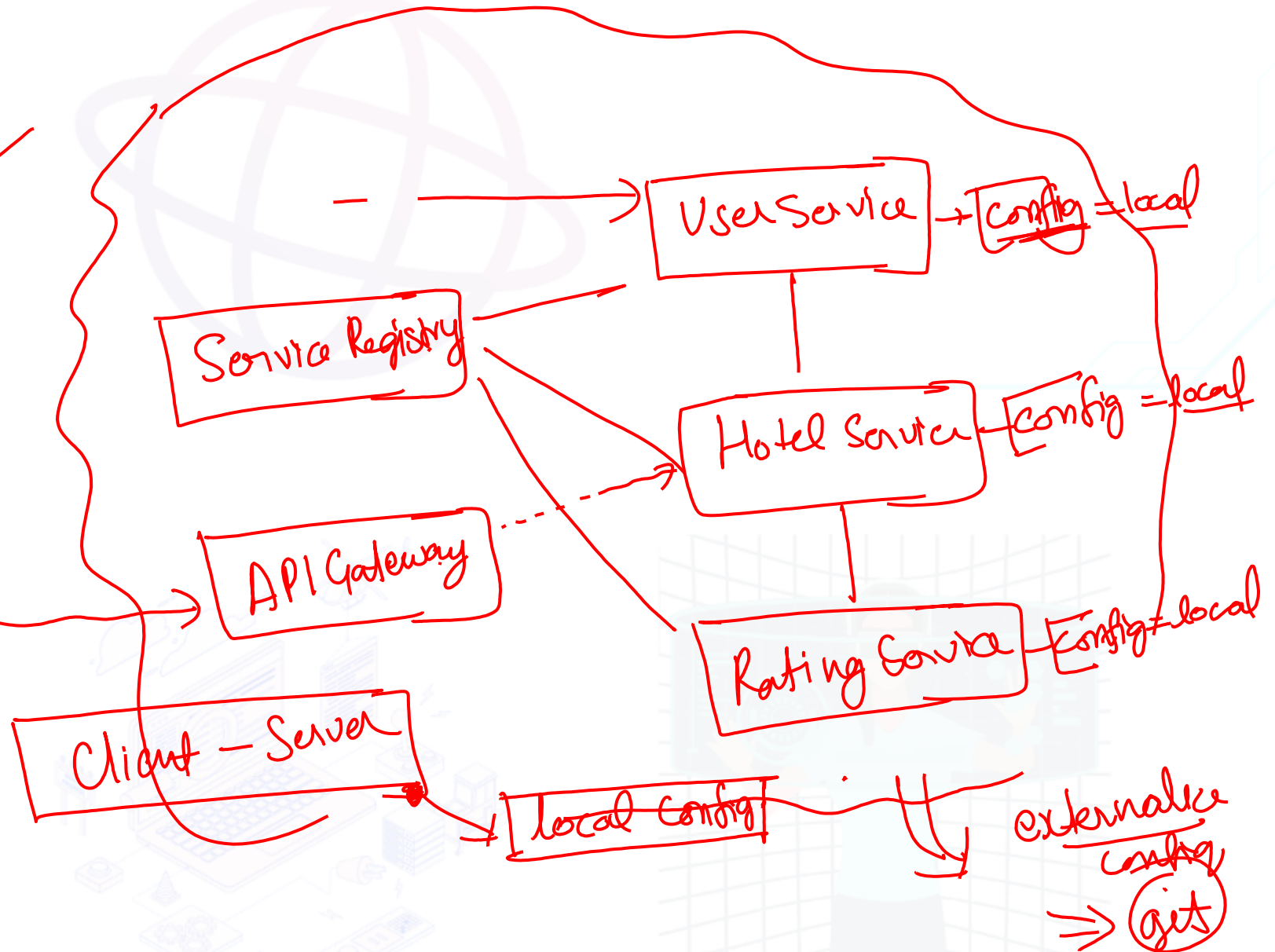




# Config Server

Client

Request



# Fault Tolerance

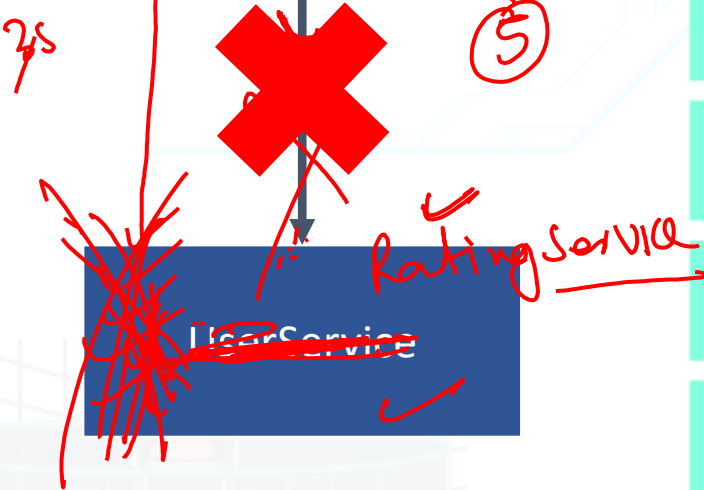
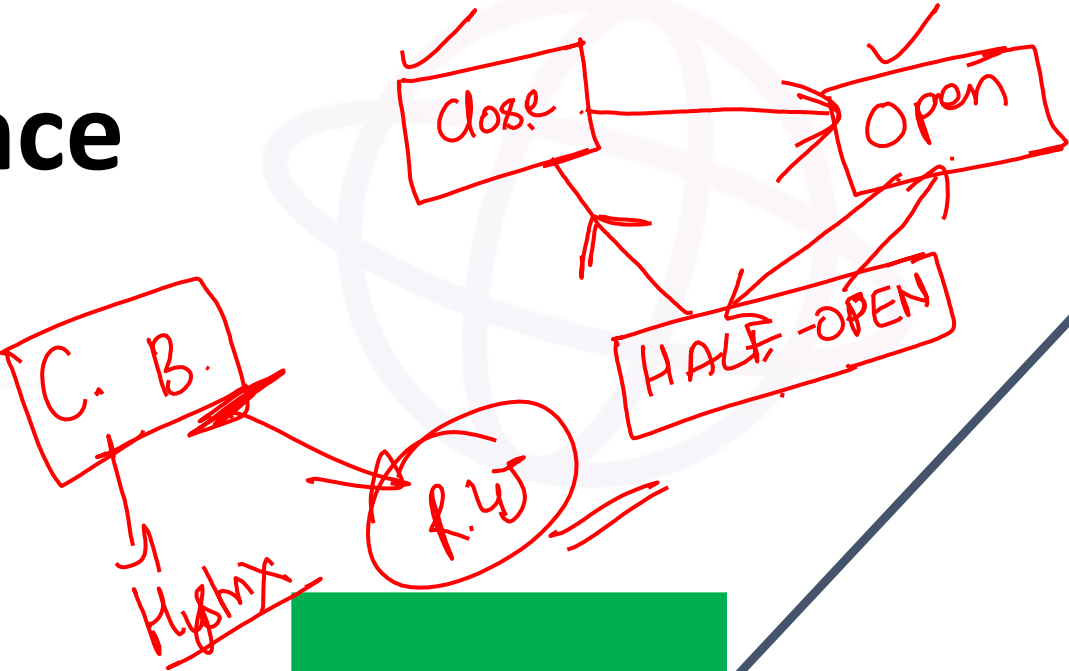
Client

API Gateway

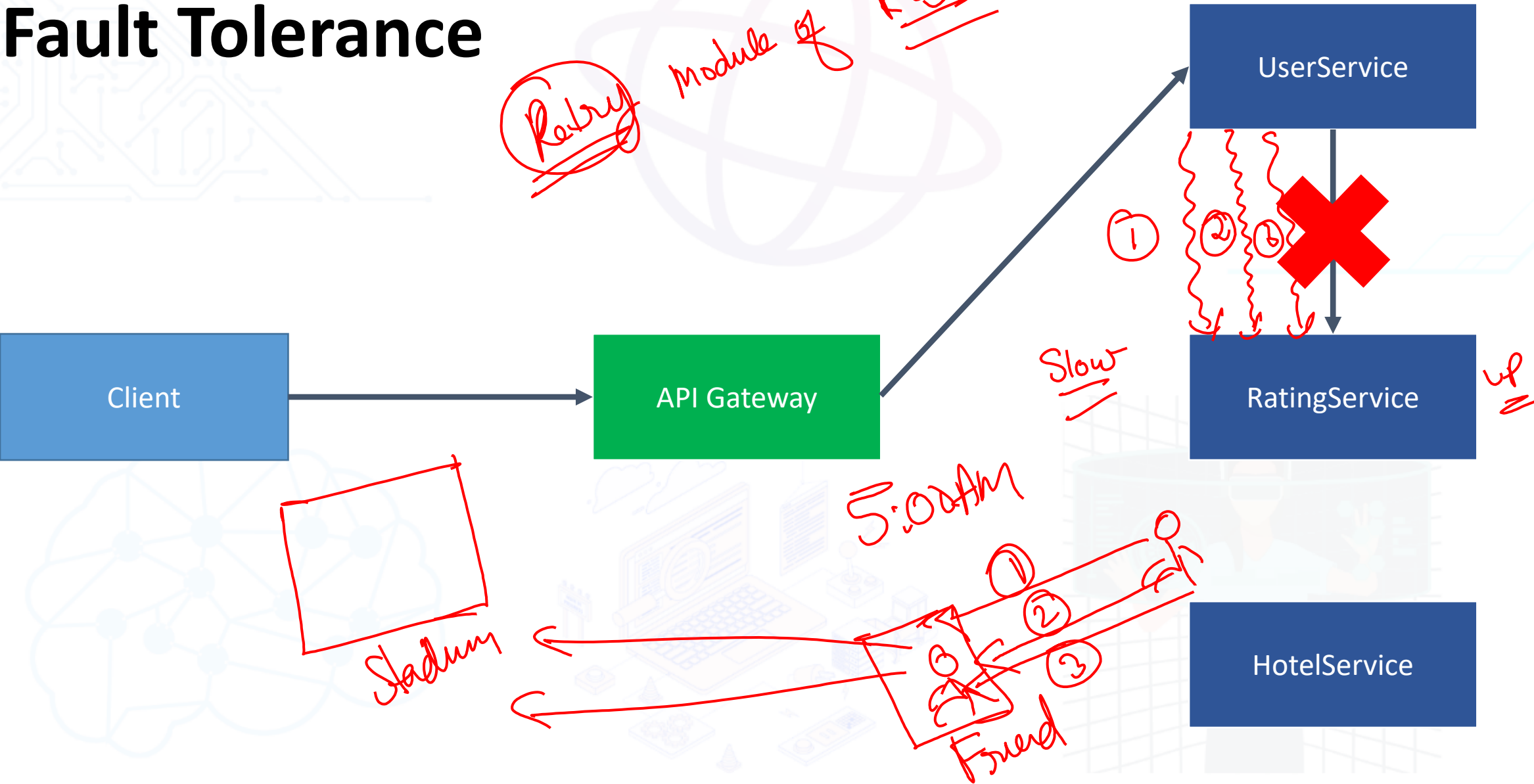
UserService  
~~UserService~~

~~UserService~~  
RatingService

HotelService



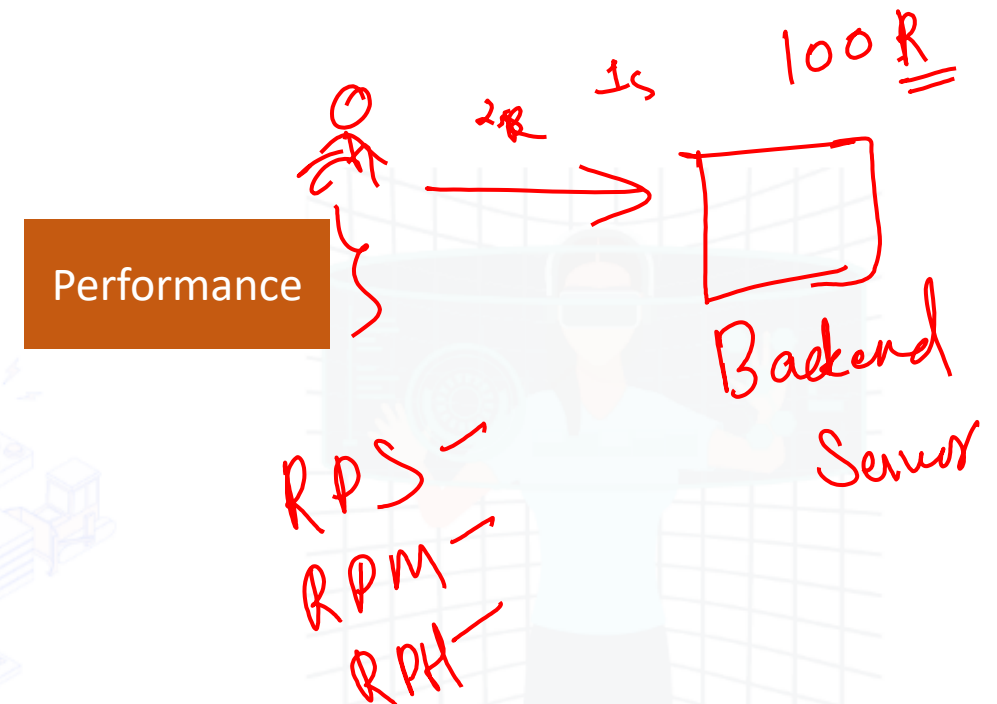
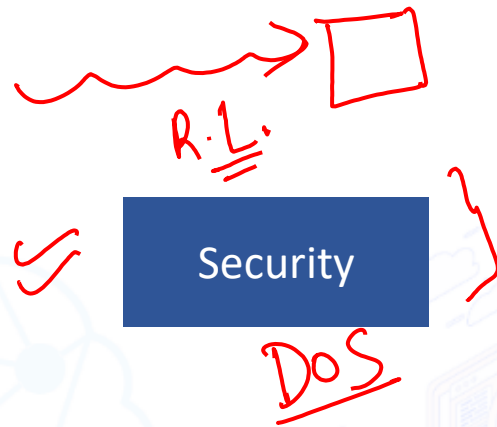
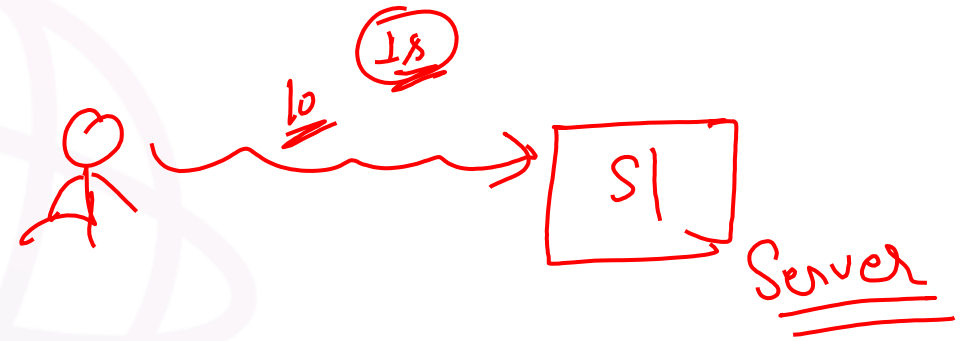
# Fault Tolerance



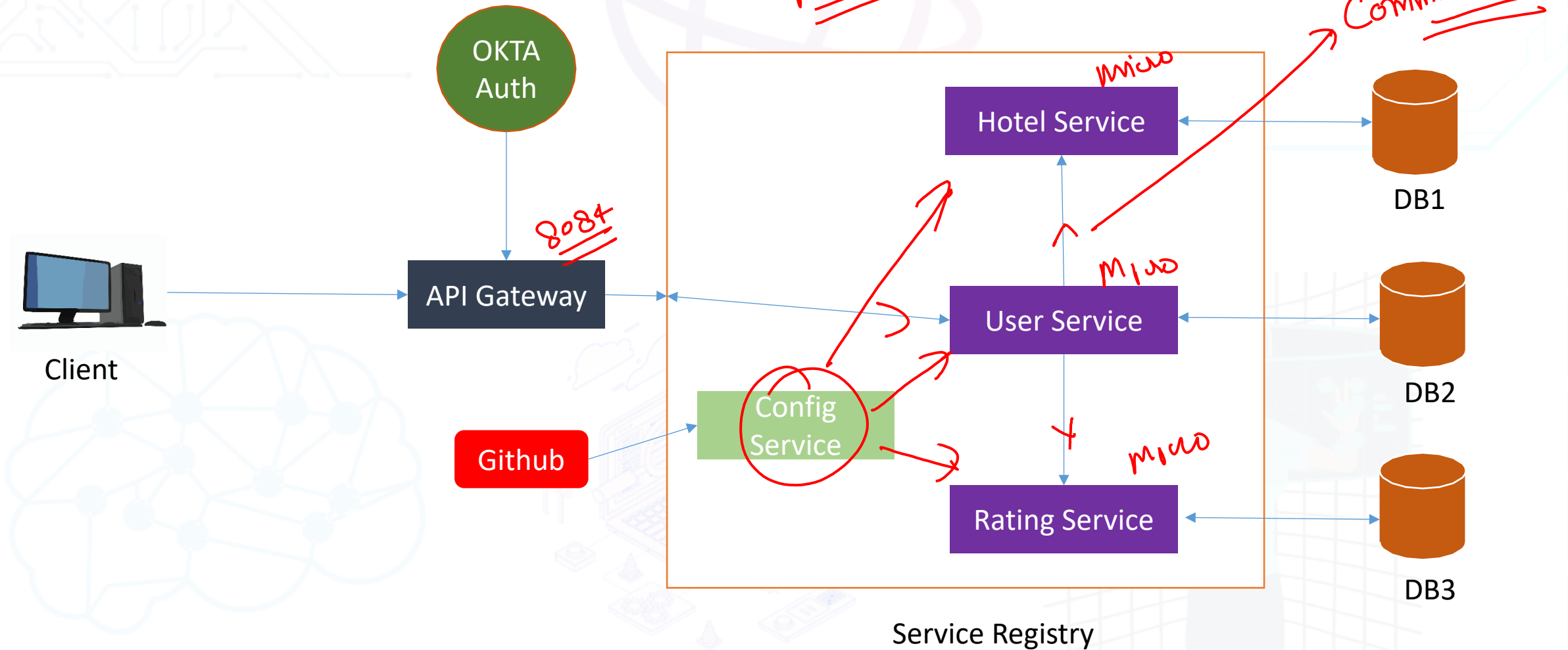
# Rate Limiter

This functionality allows limiting access to some service.

Rate Limiter make services highly available by limiting the number of calls we could process in specific windows.



# Security with OKTA





# Security with OKTA

