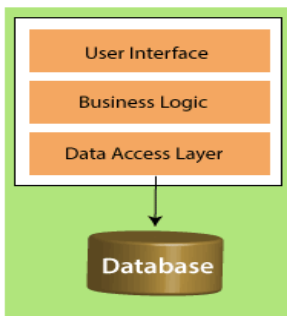


Microservices

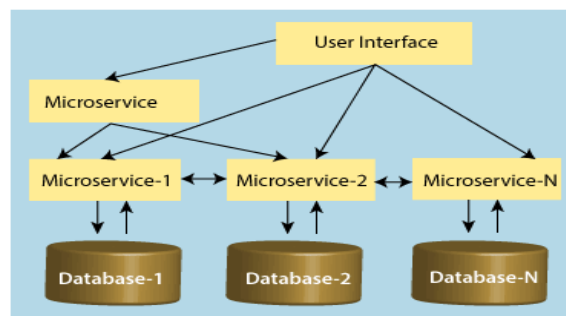


Monolithic Architecture



Monolithic Architecture

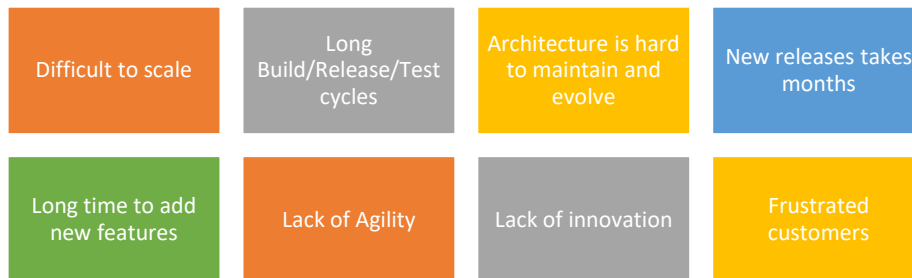
Microservices Architecture



Microservice Architecture

Monolithic vs Microservice Architecture

Challenges of Monolithic Architecture –



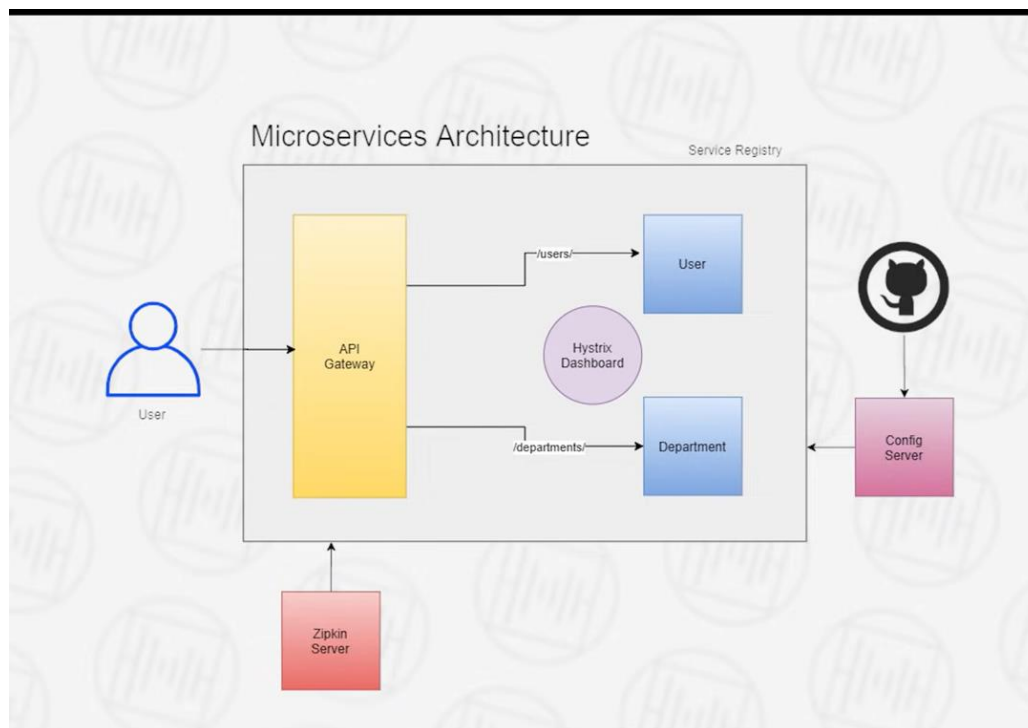
Definition – Microservices are the small independently deployable services that work together.

Principles of Microservices

There are the following principles of Microservices:

- **Single Responsibility principle** -- a class or a module in a program should have only one responsibility.
- **Modelled around business domain** -- never restrict itself from accepting appropriate technology stack or database.
- **Isolate Failure** – We can scale, upgrade fix failures without damage to other services.
- **Deploy independently** – Fix resources, cost based on requirements.

Components of Microservices



API Gateway

An API gateway is **an API management tool that sits between a client and a collection of backend services**. An API gateway acts as a reverse proxy to accept all application programming interface (API) calls, aggregate the various services required to fulfill them, and return the appropriate result.

The Service Registry is a **database that contains the network locations of service instances**. The service registry needs to be highly available and up to date so clients can go through network locations obtained from the service registry.

Netflix Eureka Naming Server

Netflix Eureka Server is a discovery server. It provides the REST interface to the outside for communicating with it. A microservice after coming up, register itself as a discovery client. The Eureka server also has another software module called **Eureka Client**. Eureka client

interacts with the Eureka server for service discovery. The Eureka client also balances the client requests.

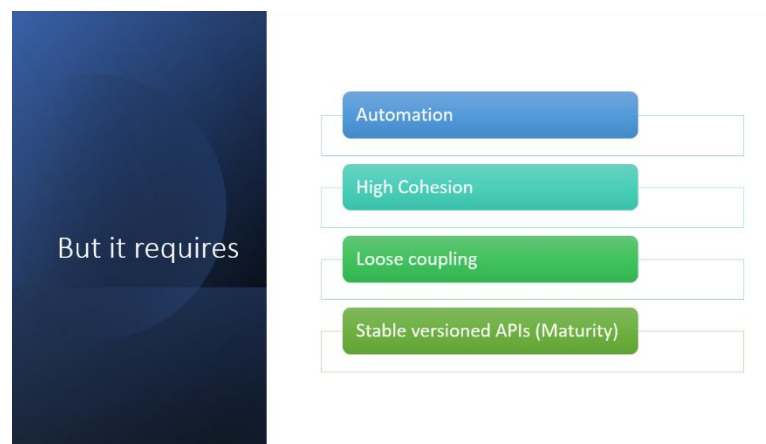
Hystrix Server

Hystrix server acts as a fault-tolerance robust system. It is used to avoid complete failure of an application. It does this by using the **Circuit Breaker mechanism**. If the application is running without any issue, the circuit remains closed. If there is an error encountered in the application, the Hystrix Server opens the circuit.

Zipkin Distributed Server

Zipkin is an open-source project m project. That provides a mechanism for sending, receiving, and visualization traces. Use for Distributed Login and logging traces.

But, Following things are important for microservice architecture -





Loose coupling is **an approach to interconnecting the components in a system or network** so that those components, also called elements, depend on each other to the least extent practicable.

References –

- <https://www.youtube.com/watch?v=BnknNTN8icw>
- <https://deloittedevelopment.udemy.com/course/microservices-with-spring-boot-and-spring-cloud/learn/lecture/8004660?start=15#overview>
- <https://github.com/shabbirdwd53/Springboot-Microservice/tree/main/user-service>