………..MicroServices……

# **What are Monolith Applications?**

* A monolithic application is **built as a single unit** as a single component.
* So Basically Such Applications built in three layers:

1. database — consisting of many tables usually in a relational database management system.

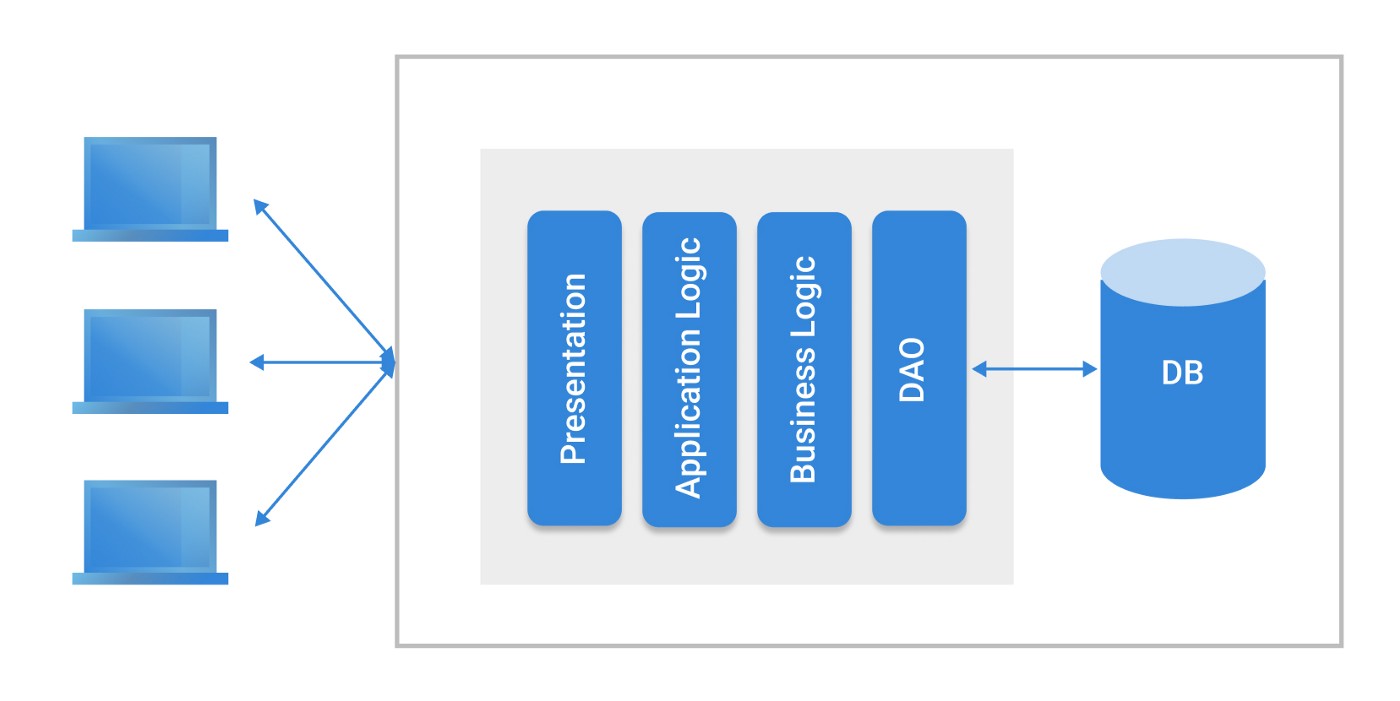
2) client-side user interface — consisting of HTML pages and/or

JavaScript running in a browser (usually done with angular ,react etc )

1. Server Side Applications like for Ex: we have springboot , springmvc.

* Normally a monolith application have one large code base and it lack modularity.

# **Monolith Application Architecture**



Clients

Repository Layer

Business Layer

Controller layer

Presentation layer

# **Disadvantages of Monolith Application**

* Large code base; tough for developers and QA to understand the code and business knowledge.
* Tight coupling between components, as everything is in one application.
* Less reusability.
* It is very difficult to introduce new technology as it affects the whole application.
* A single bug in any module can bring down the whole application
* It is very difficult to scale single module. One has to scale the whole application.
* Continuous deployment is extremely difficult. Large monolith applications are actually an obstacle to frequent deployments. In order to update one component, we have to redeploy the entire application.

**What are Microservices?**

While Monolith Application works a single component, a Microservice Architecture breaks it down to **Independent** standalone small application,

Microservices are **an architectural style that develops a single application as a set of small services**. Each service runs in its own process. The services communicate with clients, and often each other, using lightweight protocols, often over messaging or HTTP.

Within this microservices architecture, the entire functionality is split in independent deployeable module which communicate with each other through API’s (RESTful web services)

# Microservice Application Architecture

Microservices

Services

Mysql Db

Oracle

UI

Mango Db

Services

Services

**Advantages of Microservices**

* All the services are independent of each other. Therefore testing and deployment is easy as compare to monolith application.
* If there is bug in one microservice it has an impact only on a particular services and does not affect the entire application.
* With microservices architecture, it’s easy to build complex application.
* It will give flexibility to choose technologies and framework for each microservices independently.

**How to start with Micro services**

Steps :

1. If u have a monolith application, Identify all possible standalone functionalities and modules.
2. Once you have identify them, you need to create standalone projects, we are taking spring boot to create these microservices.
3. You need them to interact with each other through some ways, it can be Rest Api or Messaging. we are going to use restful architecture for the same.
4. But just doing this does not make sure you have implemented microservices architecture. These are till now just 2 Restful web services. You need **load balancer** , **Eureka** for service discovery(useful during load balancing and cloud deployments ). API gateways and many more stuff.

Client/browser/

swagger

Register

Register

Eureka

Vaccination center

service

Citizen services

**Eureka Server :**

Eureka Server is **an application that holds the information about all client-service applications**. Every Micro service will register into the Eureka server and Eureka server knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.