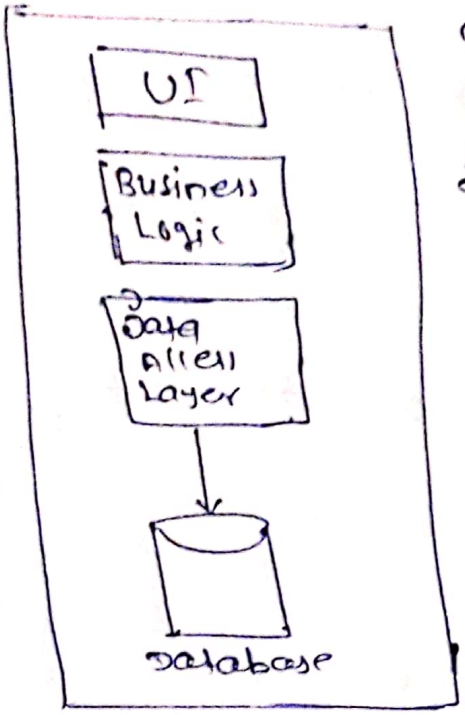
Microservices

Monolithic ⇨ Single Application in which all your different module are smashed into single project & single project is deployed in remote server, such type of Architecture is called monolithic.

mono - single
lithic - stone.



Disadvantage of monolithic Architecture ⇨

1] ~~Difficult to deploy big Application~~.

~~2] Scalability~~ :

1] difficult to ~~deploy~~ maintain deploy process.

e.g → If your project is deployed then you want to add extra functionality then you have to redeploy all your project.

2] Scalability ⇨

Scalability is a process in which number of instances of Application increses & ~~decreses~~ decreases rapidly.

If you have to maintain scalability, then number of instances of application you have to increase when traffic goes high & then traffic goes down

number of instances must decreases.
If you maintain scalability in monolithic then you have to scale all project, i.e. you have to create number of instances of project. you have to scale unrelated code also.
e.g.: If traffic increases on shopping profile pages then you have scale also to user profile & other code that does not need to be scale.

3] Technology not flexible for technology changes -: (Inflexible).
i.e. you can't develop code in different language.

4] monolithic architecture is unreliable.
i.e. when one of module goes down the entire application goes down.

Microservices :
① multiple, indivisual, deployable component called as microservices.
② microservices are an architecture where in all the component of system are put into individual components, which can be built, deployed & scaled individually.

microservice Architecture

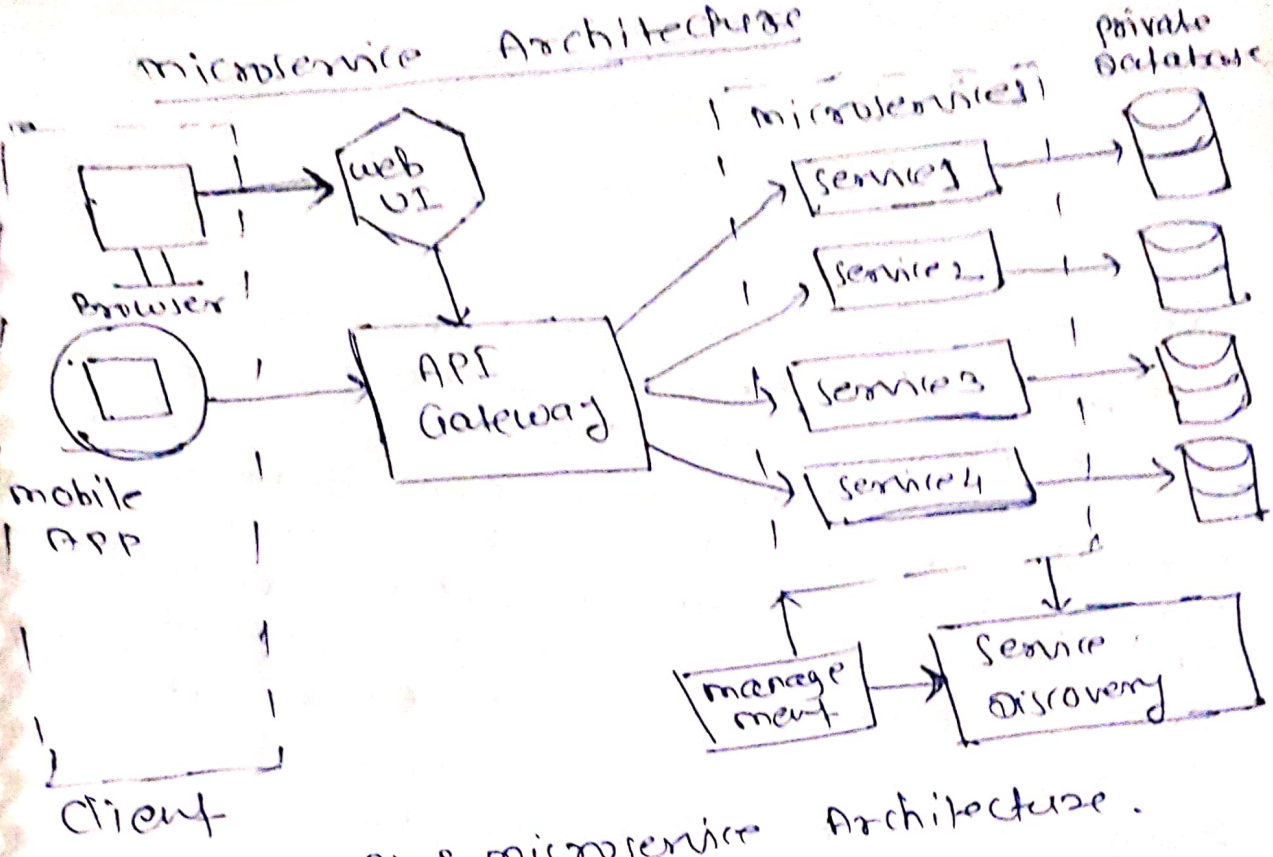


Fig: microservice Architecture.

1] management :
management is a component responsible for placing services on node, identifying failure & rebalancing services.

2] service discovery :
Task of this component is to maintain list of services & which nodes they are located on.

3] API Gateway :
It is entry point for client.
if you use spring cloud to developed microservices then zuul is API Gateway.

How to design microservices ?

→ 1] Seperate data store for each micro-service.
2] seperate build for each service.

TO manage such huge amout of services, we need Registration Server to identify the services.
So Netflix has Server named as Eureka, where all your service can register themself, and can be easily identify.
Eureka means → OK I found it.
& This Server is opensource & Spring has incorporated into Spring cloud.

What is Spring cloud ?

→ It is tool for developer to quickly build some of common pattern in distributed system.
pattern :→ make microservice work well together.
1] Service discovery.
2] configuration management // to put properties file in drive.
3] circuit braker
// When it experience flowness.
4] Intelligent routing //zuul.

Eureka is technology is allowed for above pattern.

Eureka-Server :

1] add dependency "Spring-Cloud-Starter-netflix-eureka-server".
2] use @EnableEurekaServer
3] configure some properties.

Eureka-Client :

1] add dependency "Spring-Cloud-Starter-netflix-eureka-client".
2] use @EnableEurekaClient.
3] configure some properties.

Dynamic-routing

Zuul-Server

Zuul-server is gateway Application that handles all the request & does the dynamic routing of microservice Application.

To create Zuul server:

1] add spring-cloud-starter-zuul dependency
2] @EnableZuulProxy
   @EnableEurekaClient
3] Configure some properties.