Microservices Text

1.What are microservices?

Micro service is architectural way to design our application uniquely with several small modules

developed individually, packaged individually, and deployed individually, in individual processes

(It is not framework, no tightly coupled between modules)

2.Monolithic Architecture

We are Developing Our every service individually and at end of developing we are packaging all

services as single war file and deploying in serverDeveloping

3.Drawbacks of MonolithicArchitecture

->Continuous deployment not possible -If issue in one service after fix in 1 service ,need to redeploy entire code

-> Adopting to new technologies will be difficult(for all modules need to use same technology)

->Exception propagation not proper– If driver manager exception occur it impact others

->code redeable not there –(new person joined proj if) issue in billing service one person need to understand entire application to fix only billing service module

->Performance issue due to huge data

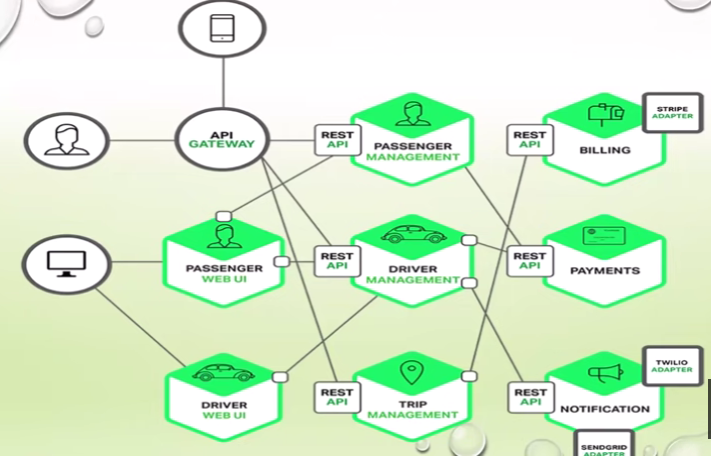
4.Advantages to use microservices

3 services like Passenger management uses PM Database,

DriverManagement uses DM Database,

TripManagement uses TripManagement Database

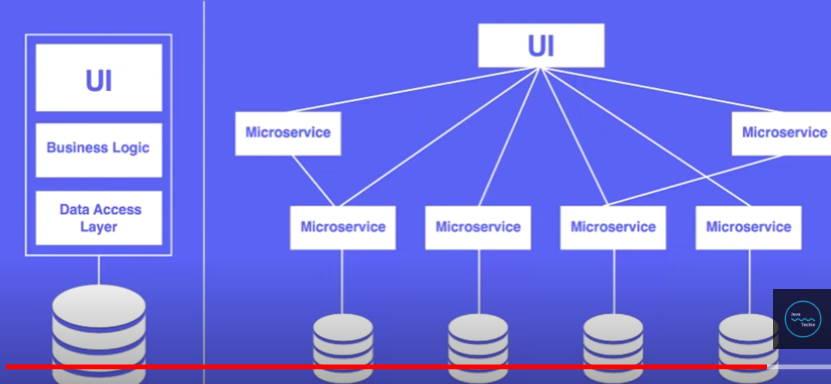
1. Each module deployment is independent
2. Load will not be huge as we are deploying each ms as war file
3. Continuous deployment is easy(Only changed module redeploy can be done)
4. Adopt Technology
5. Code Readable is easy(since single module work upon , if issue occurs)



**In Monolithic Architecture request comes from UI. From UI to Business Logic ,BL will have all module data bind to specific war.There is only 1 DB(common db, data access layer) for all modules.This is monolithic architecture.**

So bcz of drawbacks of MA , Microservices came into picture

**For Microservice Architecture - From UI request comes to each Microservice fromthere own data base instance**



**Spring Boot** technology provide complete support for microservice architecture

Histrix,Eureks Server,Zuul, Console are some components supported by Spring Boot

https://javagyanmantra.wixsite.com/website/single-post/2018/02/13/Micros-Services-Architecture

Eureka Server – Instaed of using many end point for different mapping we use Eureka server to save all end points here and access from here (@EnableEurekaServer)

Insurance Config Server – Many clients having end point if some method is renamed by client, the end point wont work without renaming..so we use Insurance config server…

From Insurance Provider changes done to github..IT loads all values and change method names by using insurance config server

@RefreshScope – change notification at client side

3.Load Balancing – distribute incoming request to multiple server(deploy application instance in multiple server..so incoming traffic is redirected to particular server)

How to perform Load Balancing using Spring cloud ribbon –(Sequential Order or Round Robin)

1st req to chatbook1 ,2nd req to chatbook 2 ,3rd req to cb1,4th req to chbook2..So it is called Sequential or Round Robin

Least Connections – 1st 100 request goes to ex: chatbook 1 next 100 req to chatbook2

IP address – Based on IP address it needs to send request

4.Feign Client – Declarative HTTP client developed by Netflix.

When we are performing load balancing in spring cloud and when integrating spring cloud eureka service registry we should not use Rest Template(clien)t then we go for feign client where it internally balances load

No need to write unit test case for rest client..only developer need to declare and annotate with interface while actual implementation provided at run time

Jsonschema2pojo

5.Distributed Tracing in Microservice with Spring Clouth Sleuth & Zipkin

**Sleuth** : Generate unique Trace id(same for all microservices in application) and span id(diff for each microservice)– used for tracking purpose

**Zipkin** : Efficient for distributed tracing microsystem. @EnableZipkinServer(Instanceid,relativetime,get)

1applica -4 mics -which component took how much time , when there is lag so that we can check all transaction statement like which component working slow

6.API Gateway:

Zuul : JVM based server(Based on url it will go to multiple microservices)and router

Instead of ui know abt microservices details we can provide unified proxy interface that will forward request to ms based on url patterns

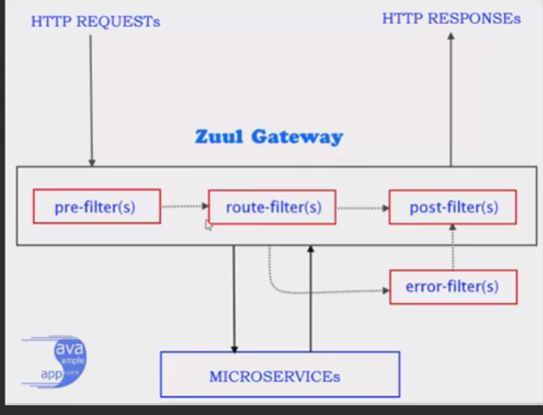
Zuul uses filter/Routing to find microservice based on url pattern

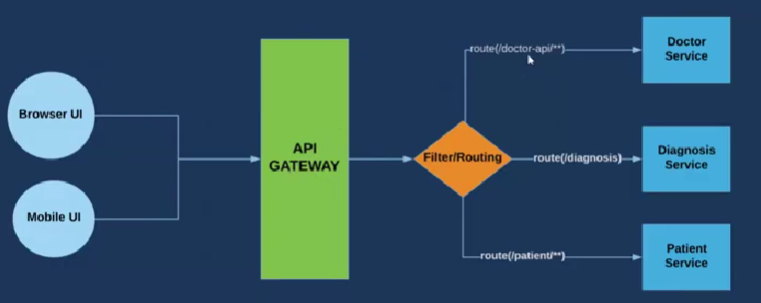
4 filters to trace request flow – which microservice 1.**prefilter** – It’s called before request route to microservice(ms)

2.route to ms – then routefilter() is called

3.postfilter() – once ms endpoint done(request completed) it gives response..that’s y postfilter

4.error-filter() -if any error errorfilter() method called





7. Consul(Centralized Configuration) : Hashicorp is owner of consul

Instead of hardcoding confidential properties we use consul

If we want confi details we need to get access to consul

8. Single Sign On – One time login(If we login in one time it should not ask whenwe login other time after few days)

**Okta server used for – sign up**

**@EnableOAuth2Sso**

8. @functional

1.Promote implementation of business logic via functions

2.Enables Spring boot features (auto configuration,DI,Metrics)

**2 dependencies specified for @functional :**

**->spring- cloud- starter -function -web**

**->spring- cloud -function- compiler**

Cloud version :Greenwich sr2