Java Techie Follow

Oct 7, 2019 . 5 min read . Listen

Save

Micros Services Architecture

Micro services based architecture has got a lot of media traction during the last few years. Plenty of blogs and articles have already been written about the benefits of micro services based architecture. It's nothing but a unique way of designing applications as different

Contents:

What is Micro Services?

Micro Service is Architectural way to design our application uniquely with several small modules developed individually, packaged individually, and deployed individually, in individual processes.

Why Micro service came into picture where previously we are developing and deploying our application in efficient way in Monolithic approach

Let’s go through previous architecture which people are following I.e.(monolithic) and find the dis advantages



Normally using monolithic approach we are developing our every service individually and at end of developing we are packaging all services as single war file and deploying in Server

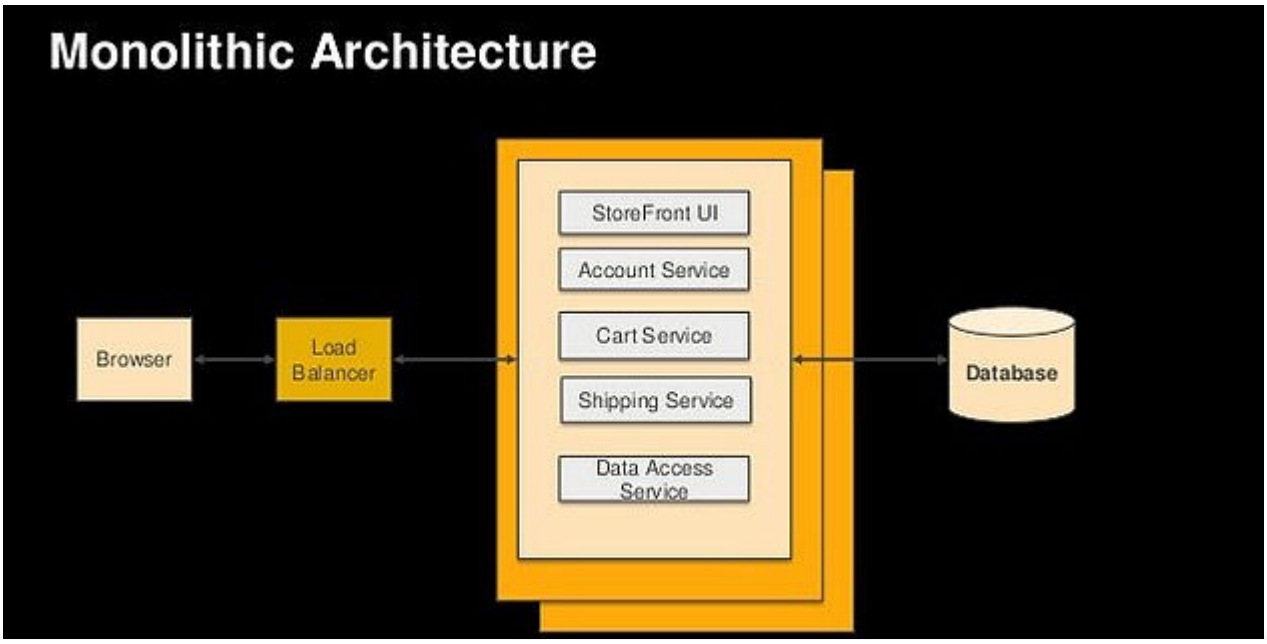
So assume I have 10 services and all depend with each other, suppose am changing business in one of service so what happened for single service change I need to repackage again and need to re deploy which is big dis advantages.

Let's go through architectural diagram for better understand

Am going to explain one of the Real world example that is Any online shopping site like Flip-Kart, Amazon, E Bay which developed before with monolithic approach

Assume I have few services which is

Account management Services
Inventory/Cart Services
Payment Service
Shipping Services
Notification Service



See the above Diagram I develop all 4 services and then packaged all and deployed all as single war, and accessing services using Rest API call

> ***Find the drawbacks from as per above diagram***

1. As I deployed all service as single entry point (WAR), performance became slow because huge data present in war. Once the application itself becomes big, quick development and delivery becomes too much difficult. Bug tracking and fixing also becomes too much difficult to handle

2. Continues deployment will suffer a lot. This is because, let's say you want to deploy a bug fix in one of the component in the application. Now as the application is monolithic, the entire application needs to be re-deployed

3. Adopting new technologies or languages will be difficult. As you cannot implement one language for one module of the application easily

4. Overloaded IDE — the larger the code base the slower the IDE and the less productive developers are.

5. Overloaded web container — the larger the application the longer it takes to start up. This had have a huge impact on developer productivity because of time wasted waiting for the container to start. It also impacts deployment too.

6. Exception propagation not proper suppose assume my payment service failing then it impact on my Entire application before

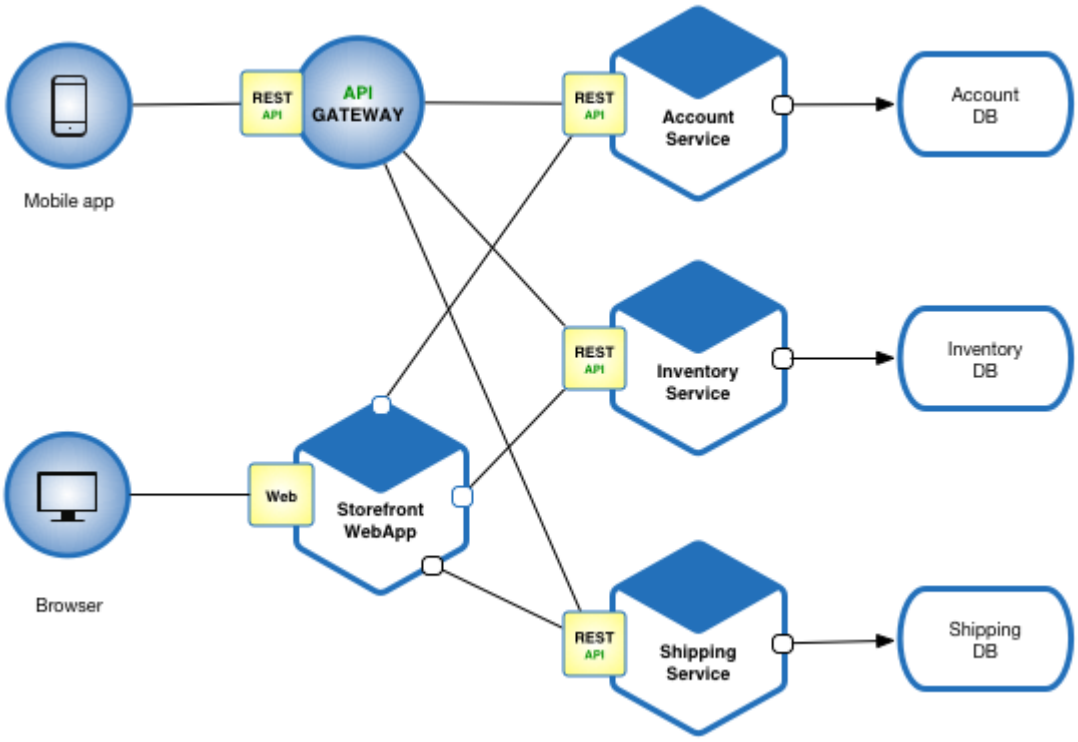
7. Code Readable is not there , like assume one new resource join in your project and got some task so how can he/she analyze , they need to understand complete flow to know the exact business of specific module which seems lees productivity

Now Flip-Kart and Amazon migrate their architecture from monolithic to Micro services

To overcome these above drawbacks Micro-services concept came in to picture

As I already mention sort definitio we know every module we are developing, packaging and deploying individually containers so even though all depends on each other one service failing is not impact to other service where user will not suffer



Microservice Architecture

See as per diagram, each service are interlink with each other but all modules are separate here means independent war

All above mentioned 5 services have their own development and own package and deployed we can say each service is one one micro services

There is one API gateway who manage all micro services and it act like front entry point

Development procedure and deployment procedure of micro services

As per above diagram I developed 5 micro services, as am not going to deploy all as single war then think how can I achieve features from one service means how to communicate from one service to another

Netflix is one of them who provide us great features to register N number of micro services, so he provided one embedded server that is Eureka where we can register our micro services to communicate with each other

So here API-Gateway is nothing it can be any Cloud environment or Eureka or Zuul proxy server

SO from UI we are not communicating directly to our micro services,

First Request will come to API-Gateway then based on incoming URL it will find mapping controller from Service Registry then it will delegate Request to corresponding Controller and process the business

So think how the process and performance is faster in micro services as compare monolithic application it seems we can achieve more scalability

Advantages:

1. As I deployed each service individually in different container and assume registry in Eureka Service Registry performance won't be slow. As each are one micro service so quick development and delivery becomes faster and easy to track Bug and fixing

2. Continues deployment will not suffer a lot. This is because, let's say you want to deploy a bug fix in one of the component in the application. Now as this module is independent just do the changes on that specific module and deploy only that module , no need to bother about other services as they are already register in Service registry so they are UP .Think how easy this .

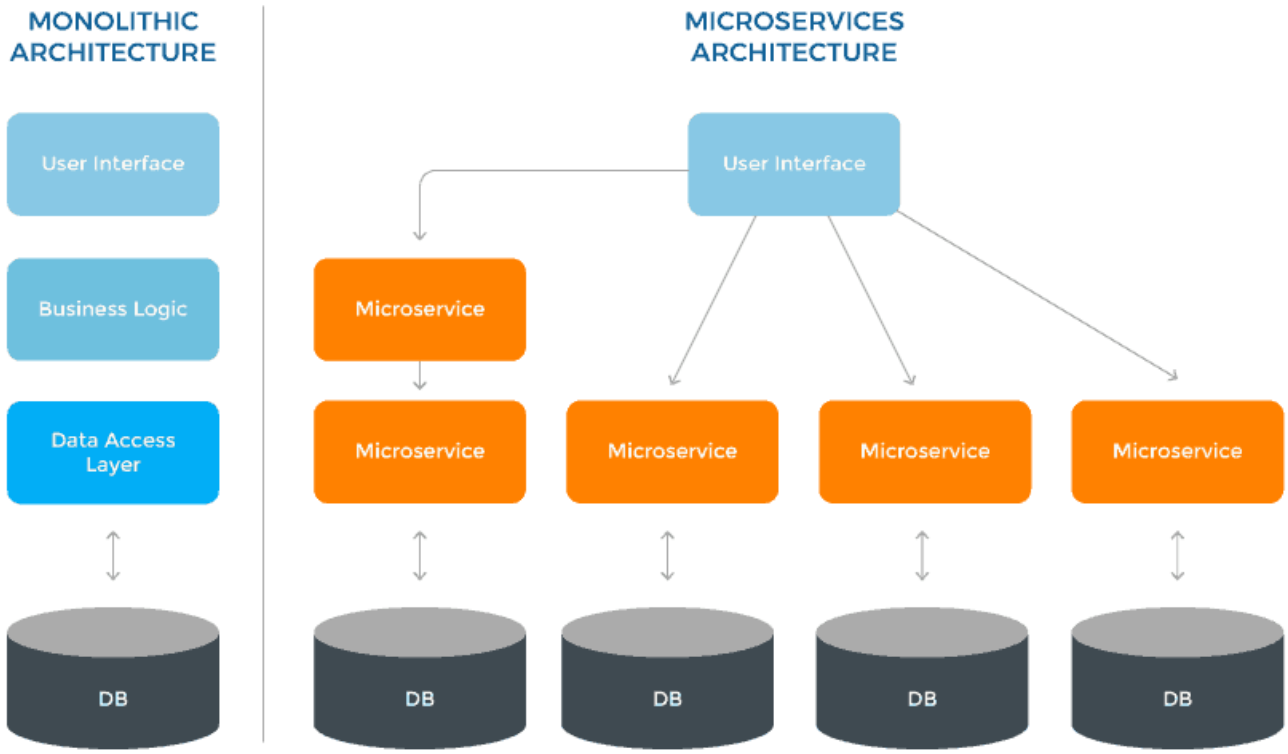
3. Adopting new technologies or languages will be very easy. As you can implement one language for one module of the application easily

4. As each service are separate war or jar with less data specific to particular module so server start up won't take much time, we can up server in multiple port and auto restart in every logic change so for that we need to use either dev tools or spring loader

6. Code Readable is very clear , suppose one new resource join in your project and got task to work on Shipping micro service so he/she can easily open that service and can check the flow as it is individual service

Let's have a look on combined Architectural diagram of both Monolithic and Micro service



So this is All about Microservices