

Review of Framework of Microservices, Understanding Its Purpose, Benefits and Limitations

Dr. Jatin Gupta

Assistant Professor, Chitkara University, India
jatin.gupta.1988@ieee.org

Abstract: - Microservices is the methodology which is used to develop applications by dividing the huge application into small components where each component will provide service which are called as microservices. This concept was introduced when it was discovered that by dividing the large application into small services will be easy to develop and customer will also have trust that the application is getting ready as it will be delivered in small microservice components and feedback will be accepted from the client. The advantage of developing the application in the form of microservices is that each service can be written in any language using any tool. Once a microservices is developed it can also be utilised for another project which will have similar functionality. Microservices with the help of cloud computing as well as containerization helps to develop the microservices faster than the development of the complete application development process. Each microservice will be independent of other components of the application being developed. For the communication between various interfaces of the application, microservices uses the concept of API. The paper will discuss the architecture of microservices along with the working mechanism. It will also discuss the benefit and challenges of microservices concept.

Keywords: Introduction to Microservices, Microservices framework/architecture, Types of microservices framework, Purpose of microservices, Pros of Microservices, Cons of Microservices.

Introduction: - [1]

The innovation of the microservices was done to facilitate better and improved communication between various platforms. IT was discovered to provide user friendly interfaces which can be used to develop the application by using small serviceable components which are independent from each other. It was observed that if the application is huge and has complex features then the development of the application takes a great amount of effort and time which can otherwise be used to develop many small applications. So, the concept of microservices was introduced. It is a service-based application development strategy which is used to develop large application by breaking into small components where each component is able to provide unique service. The advantage of developing an application using this concept is that these microservices are independent from each other and can be used for developing other application which will save time and effort. The microservices are developed and delivered faster as compared to other techniques. Each microservice has the capability to deliver specific business goals. In service- oriented architecture, the whole application is divided into small components which are loosely connected to each other and uses different protocols to communicate between themselves in order to serve the aims and goals of the organisation. Service-oriented architecture is a pattern which is used to develop the application whereas microservices is the process of using service-based architecture to develop the application.

Purpose/Importance of Microservices: - [2]

Following are the reasons which serves the purpose of microservices: -

- **Monolithic architecture: -**
In this type of architecture, the code is present in one single file. Due to this in case of bugs identification, it is very difficult to locate and fix the bug as it can be anywhere in the application. It is difficult to make any changes or update in this type of architecture. Using this architecture for application development takes longer period of time to develop the software application. In order to avoid these issues, microservices architecture was introduced where the whole application can be divided into different microservices which are independent of each other and has the capability to serve the business goals. They can be developed at much faster rate as compared to the monolithic architecture.
- **Scaling issues: -**
It is the main challenge of the monolithic architecture in which it is difficult to scale as when there is overload of data, load balancers are required to manage the traffic and the only way is to develop complete monolithic application. This issue is resolved in microservices architecture where containers can be used to add into the microservices which makes the efficient use of the resources.
- **Reduces time and cost: -**
Monolithic architecture takes a lot of time and effort to develop the application whereas microservices are able

to develop the application faster and reduces the overall cost and time to develop the application.

Microservices Framework: -

- Microservices is the approach or process which will implement service-based architecture to develop an application by diving it into different components known as microservices. Each microservice is capable to serving the business goal as they have their own services to provide. They are independent of each other and can be easily utilised for other application development as well.
- Each microservice has its own database and has its own process to run the services. Each microservice is capable of performing various tasks like user identification and authorisation, user interfaces, log data etc.
- Microservices provide decentralised system where if any changes or bugs are identified, they easily be located and fixed. Each microservice can easily be isolated, modified, can be built again, and delivered again and all this can happen independently.
- Following are the three main points which should be kept in mind while developing the microservices: -
 - a. Business goals: - It is necessary that each microservice is capable of serving atleast one business goal.
 - b. Coupling: - All the microservices of the application should be connected toe ach other in such a way that they do not depend on each other but still be connected.
 - c. Independent: - Another important point of developing the microservices is that they should be independent of each other.

Types of Microservices Framework/Design strategies: - [3]

Microservices have different design patterns and frameworks. Following are few points which should be kept in mind while dividing the application into microservices: -

- Different assistance examples per have
- Administration occasion per have
- Administration occasion per VM
- Administration occasion per Container
- Serverless sending
- Administration sending stage

Various Microservices Design patterns are: -

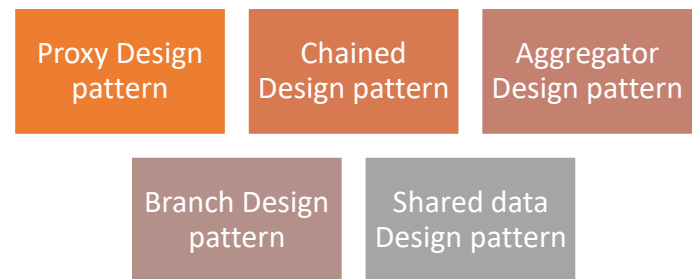


Figure 1 Types of Design pattern of microservices.

1. Proxy Design pattern: -
This pattern is similar to aggregator pattern but it is not used to put aggregations on the end user. When the services are presented through interface and not directly to the client then this design pattern is used. It can scale both in x-axis and z-axis. This is further divided into two parts: -
 - a. Dumb proxy: - This type of proxy will make the request to single service available.
 - b. Smart proxy: - This will use data transformation technique and then sends the respond to the client.
2. Chained Design pattern: -
This type of pattern will create only single response to the whenever the user makes the request. To communicate with other services, they will form a chain like structure. In order to complete the chain, the client will be blocked by the services whether or not they have different requests etc. Also, it should keep in mind the chain formed should not be very long.
3. Aggregator design pattern: -
It is one of the most commonly used pattern which is a web page which uses a number of services to complete the task. The web page is able to identify the data as it uses the mechanism called as REST. If nothing is required to be displayed than this pattern will take data from each service and then applies business logic to it.
4. Branch Design pattern: -
It is extended form of the aggregator method where it takes request from various chains of microservices and provides response simultaneously.
5. Shared data Design Pattern: -
For this design pattern, the service needs to be full stacked and should have complete control over all the parts of the application, user interfaces, transaction etc. It can be more efficient as it will be able to select the right tool.
6. Asynchronous design pattern: -
There are a constraint inside the REST configuration design because of synchronicity. Some microservices will rather utilize informing lines to assist with mitigating a portion of these impediments. This will permit one help to call one more while conveying through an informing line

- and subsequently, have the option to do more intricate exercises.

Three C's of Microservices: - [4]

Following are the three main points which are important for the implementation of the microservices: -

1. Collaborate: -
 - Collaboration has two important aspects to notice which are : - API contract and technology standardisation. The first one is a contract which will brief about how a service will be used and also make sure that everybody in the team stick to it.
 - The goal of technology standardisation is to make sure that each microservice can be deployed into the infrastructure.
2. Componentize: -
 - For this element, first of all a pilot project is identified which can be turned into microservices.
 - First of all, a component of the selected application is identified which can be converted into microservice. For this selected component, APIs will be defined, then a plan is created using the development language and platform for which everybody in the team is comfortable to work with for the implementation process.
3. Connect: -
 - Once the components are ready, now it is time to connect them using presentation layer which can then be delivered to the end users to use it.
 - A reliable, stable proxy should be used and also a reverse proxy which will serve as the server.

Advantages of Microservices: - [5]

Following are the advantages of microservices: -

1. Size: - Since the application is divided into small components called microservices where each service performs specific tasks, they are smaller in size and easy to maintain them.
2. Deployment process is easy: - Due to the small size of each microservice, they can be deployed in any environment with little effort and less time and are faster to be delivered.
3. Independent: - Each microservice serves specific business goal and each component is independent of each other.
4. Reusability: - Since each component or microservice has its own unique function, it can also be reused in another application project serving the same functions.
5. Easy to identify bugs and fix them: - Since the microservices have their own code used for its development, it is easy to identify and locate the bugs and fix them. It is also very easy to make any modifications, and changes.

Limitations of microservices: - [6]

Following are some of the challenges of microservices: -

1. Distributed system: - As each microservice is developed using different language and skills it gets difficult to manage large number of databases along with the need of having number of skilled professionals.
2. Higher costs: - Each service will perform specific business goals and therefore need number of databases and servers to maintain it. Hence, cost of implementation of microservice is higher.
3. Complex interface control: - If the interfaces of the microservices are connected to each other then it is difficult to make changes to one of the microservices as it will also impact other services connected to it.
4. Security: - In critical scenarios, the security of each microservice is difficult to maintain as there will be number of microservices of an application which will be challenging to maintain security in each microservice.

Conclusion: - Microservices is the approach which is utilized to foster applications by partitioning the enormous application into little parts where every part will offer support which are called as microservices. This idea was presented when it was found that by separating the huge application into little administrations will be not difficult to create and client will likewise have believe that the application is preparing as it will be conveyed in little microservice parts and criticism will be acknowledged from the client. The upside of fostering the application as microservices is that each help can be written in any language utilizing any apparatus. When a microservices is created it can likewise be used for another task which will have comparative usefulness. Microservices with the assistance of distributed computing as well as containerization assists with fostering the microservices quicker than the advancement of the total application improvement process. Every microservice will be autonomous of different parts of the application being created. For the correspondence between different connection points of the application, microservices utilizes the idea of API. Microservice is heterogeneous in nature as precise advances are utilized to play out a particular undertaking. Microservices can be thought of as a combination of numerous innovations.

References: -

- [1]. <https://www.dataversity.net/a-brief-history-of-microservices/#>
- [2]. <https://blog.dreamfactory.com/when-to-use-microservices-sam-newman-and-martin-fowler-share-their-knowledge/>
- [3]. <https://www.castsoftware.com/glossary/microservice-s-framework-best-definition-designs-patterns>
- [4]. <https://www.infoworld.com/article/2936148/three-keys-to-successful-microservices.html>

- [5]. https://www.tutorialspoint.com/microservice_architecture/microservice_architecture_introduction.htm