International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

WWW.IJRPR.COM

Microservices

Allur Shivprasad Rao

AISSMS IOIT

ABSTRACT

The microservice architecture appeared lately as a new paradigm for programming applications by utilizing the formation of small services, each running their processes and connecting via lightweight mechanisms. This method has been developed on the concepts of service-oriented architecture. The term "microservices" was first introduced in 2011 at an architectural workshop as a way to describe the participants' common ideas in software architecture patterns. Until then, this approach had also been known under different names. Microservices now are a new trend in software architecture, many companies and organizations have adopted microservices, and detailed research can be made on the application of the architecture. The popularity of this approach is due to its advantages in terms of maintainability, scalability, and so on.

1.INTRODUCTION

1.1 Introduction to Microservices

Microservices is a new paradigm in programming applications, it is developed by the composition of smaller services to perform the functioning of a major service by communicating and connecting by sending messages. Microservices architecture is developed from service-oriented architecture and distributed systems. This architecture delivers benefits in maintainability, scalability and more.

Microservices architecture is receiving a lot of popularity these days due to the many advantages it delivers. Organizations like Amazon, Netflix, eBay, etc. have adopted this architecture, through these applications, we can study microservices and also learn how to develop this architecture.

1.2 Motivation behind Microservices

The motive behind this topic is to study and learn how to develop the microservices architecture and understand the benefit it provides to the organizations that have decided to adopt this architecture. After studying this architecture and acquiring sufficient knowledge, the next motive is to develop a prototype showing the various benefits and advantages this architecture provides. Although this architecture delivers a lot of benefits, a bunch of problems come along with it. Unless, sufficient knowledge, guidance and resources, the development of this architecture is not simple.

1.3 Aim and Objective(s) of the work

1.3.1 Aim

This seminar topic aims to study and learn the microservices architecture and also to understand how various and popular organizations have implemented this architecture. And then to be capable of developing this architecture to implement and study the various benefits and advantages the architecture gives over the monolithic architecture.

1.3.2 Objectives

- To study monolithic architecture.
- To study the difference between monolithic architecture and microservices.
- To study the working of microservices.
- To study the advantages of microservices.
- To study the applications and usage of microservices by organizations and companies.
- To study the future development of microservices.

1.4 Organization of the report

Chapter 1: Introduction
Chapter 2: Literature Survey
Chapter 3: Monolithic Architecture
Chapter 4: Difference between Monolithic Architecture and Microservices
Chapter 5: Microservices
Chapter 6: Applications of Microservices
Chapter 7: Future Development of Microservices

2.LITERATURE SURVEY

2.1 Introduction to Base Papers

1. **Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina, "Microservices: Yesterday, Today, and Tomorrow", Springer, pp 195-216, Sept. 2017. [1]**

In this paper, the authors have researched the development of the microservices architecture from the service-oriented architecture. They have also given the benefits, advantages and disadvantages of microservices, in addition to that, they have also written about the future of microservices and have pointed out a few solutions to solve existing problems.

2. **Luciano Baresi and Martin Garriga, "Microservices: The Evolution and Extinction of Web Services?", Springer, pp 3-28, Dec. 2019. [2]**

In the early 2000s, when service-oriented architecture got popular in business, e-commerce, enterprise integration, many companies and organizations tried to adopt it. This led to the creation of various definitions of service-oriented architecture and many failed to implement it. In this paper, the authors have defined microservices as "service-oriented architecture done right". The authors have also given about the advantages of microservices in detail and have also mentioned that along with the benefits of this architecture many challenges come along with it. The authors have also reviewed microservices architecture projects from GitHub, the largest open-source repository to date.

3. **Daniel Escobar, Diana Cardenas, Rolando Amarillo, Eddie Castro, ´ Kelly Garces, Carlos Parra, Rubby Casallas, "Towards the Understanding and Evolution of Monolithic Applications as Microservices", IEEE, Jan. 2016. [3]**

Nowadays, developers are facing a lot of problems in maintaining and scaling large applications. This is known as modernization. In this paper, the authors have proposed models to transform these large applications into microservices. The authors have explained the model and dependencies between the business layer and the data layer. They have also proposed modularization into clusters and microservices using 4 different diagrams.

4. **Sam Newman, "Building Microservices: Designing Fine-Grained Systems", O'Reilly Books, Feb. 2016. [4]**

In the past 10 years, distributed systems have become more fine-grained from monolithic architectures to microservices. But developing these architectures brings a lot of problems so the author of this book has given clear solutions to the problems of modelling, integrating, testing, deploying, and monitoring autonomous services. By following the book, the author teaches how to build a microservice architecture and how it affects a single domain.

2.2 Different Approach in Base Papers

Table 2.1: Literature Survey: Differences and Shortcomings

Sr. No.	Name	Difference	Shortcomings
1.	Microservices: Yesterday, Today, and Tomorrow	In this research paper, the authors have focused on how microservices were used in the past, today and how can it be improved for the future. Along with some current issues with the technology, they have explained some betterments that can be done in the future.	Although this research paper has explained the past, present and future of the microservices architecture, the authors haven't given detailed information on the application and how organizations have implemented it, as it is a crucial part of today.
2.	Microservices: The Evolution and Extinction of Web	In this research paper, the authors have explained microservices and what challenges and problems developers face to design, develop and operate a	In this research paper, the authors have also made a qualitative analysis of the microservices architecture, along with the analysis the lack of

	Services?	microservice architecture. Also, the authors have reviewed and explained some microservices-based projects from GitHub.	explanation of applications of the microservices architecture can be considered as a shortcoming.
3.	Towards the Understanding and Evolution of Monolithic Applications as Microservices	In this research paper, the authors have explained microservices and how developers can transform existing monolithic architectures into microservices architecture. This transformation is explained by clustering the services and using various diagrams.	In this research paper, the authors have given detailed information on their evaluation. The shortcoming is that the authors should have also given a practical explanation and implementation of the conversion from monolithic to microservices.
4.	Building Microservices: Designing Fine-Grained Systems	In this book, the author has explained how to make a microservices architecture from scratch. Unlike the other papers, this book uses a fictional organization to explain and implement the architecture. The book also explains various problems the developer faces while developing this architecture.	In this book, the author has given detailed information on developing a microservices architecture. The book focuses on developing a microservice, but the author should have given more information on methods used by organizations to implement microservices.

2.3 Problem Statement

To study monolithic architecture and comparison between the microservices architecture. Also, to study the microservices architecture, from the advantages to the problems that arise. Finally, to study the application of microservices architecture done by companies and the benefit they gained from it.

3.Monolithic Architecture

Monolithic architecture is an architecture that is adopted for server-side applications. This architecture cannot execute its modules independently[1]. According to the Cambridge dictionary, the adjective monolithic also means both too large and unable to change. It is a traditional unified model and its modules are inter-connected and inter-dependent. In this tightly-coupled model, all the modules must be present and inter-connected for the code to be compiled and executed.

As shown in figure 3.1, after the various requests are balanced, the entire load is transferred to the server-side application, which is handling the modules of various services and it is then transferred to the RDBMS. This model has a lot of disadvantages which led to the development of microservices.

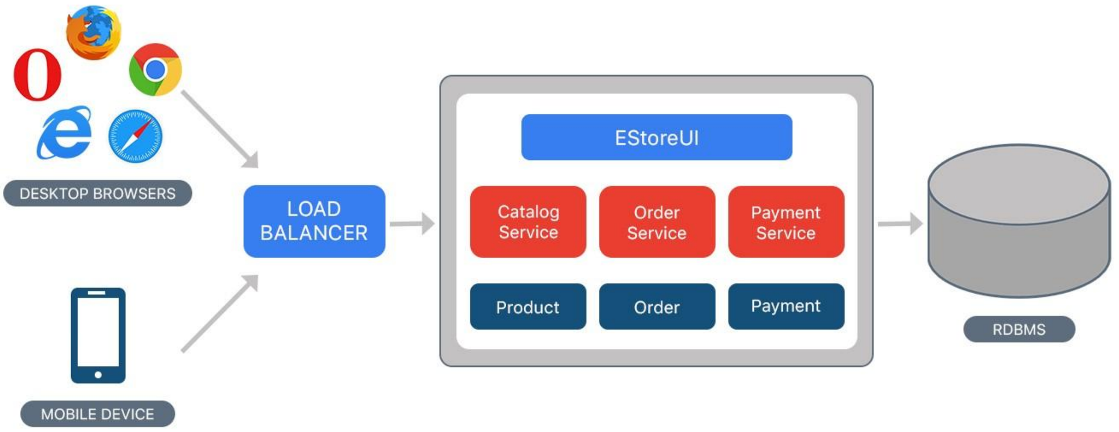


Figure 3.1: Example of Monolithic Architecture

However, there are also benefits to monolithic architectures, monolithic programs typically have better throughput than other models. They can be easier to test and debug because with the smaller number of elements there are fewer variables that come into use.

4.Difference Between Monolithic and Microservice Architectures

As there were a lot of disadvantages in monolithic architecture, a new architecture was developed to keep up with the constantly changing market and technologies[3]. Microservices brought in a lot of benefits which is useful and efficient for organizations and companies, which indirectly helps the people. As shown in figure 4.1, in monolithic architecture one server-based system handles the entire service and in the microservices architecture, the server-based system is distributed among various services.

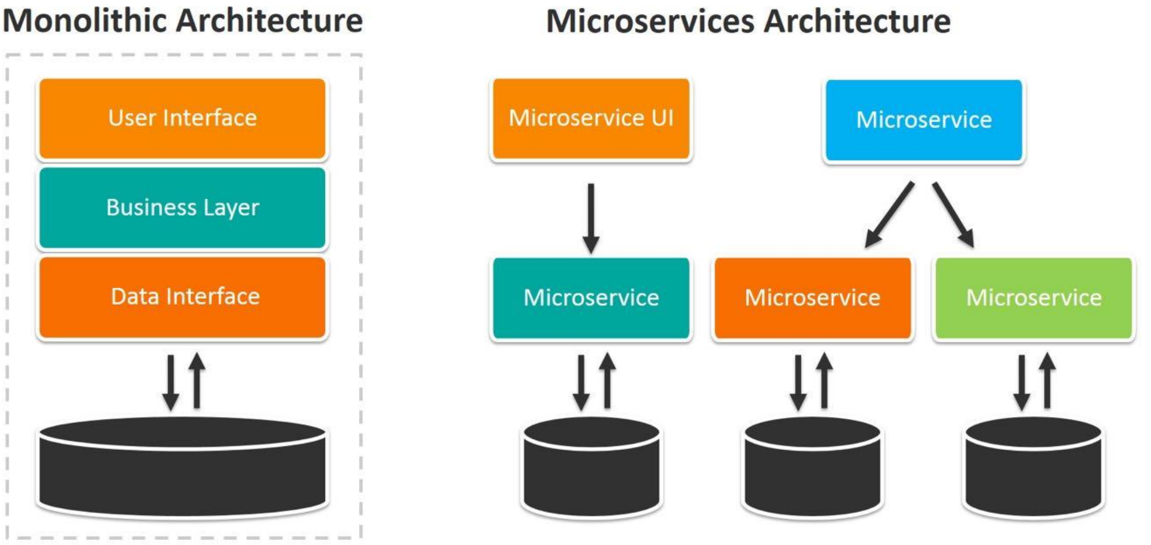


Figure 4.1: Monolithic Architecture Vs Microservice Architecture

The monolithic architecture is vulnerable because, if the system goes down or crashes the entire server-based application will not be functional. This may not seem like an issue for web applications that do not receive a lot of traffic, but for organizations and companies which receive a lot of traffic the entire day, it may lead to cause losses and also may cause discomfort to their customers and employees. This is one of the reasons companies and organizations have adopted microservices architecture.

Some of the differences between monolithic and microservices are given below.

Table 4.1: Difference Between Monolithic and Microservices

Sr. No.	Monolithic Architecture	Microservices Architecture
1.	Monolithic architectures are built as one code base which is tightly coupled and interconnected with all the modules of the code.	Microservices architecture is built by using different code bases and smaller modules that are loosely coupled, based on business functionality[2].
2.	The scaling is not easy as the entire code has to be compiled and deployed after each update[1].	The scaling is easy as only the smaller modules have to be compiled and deployed and the remaining service can still be functional and active.
3.	In this architecture, the database is shared with the entire service.	In this architecture, each process and module can have a different database.
4.	During deployment, the time taken by the IDE increases immensely because of the vast code of the entire server-side application.	In the case of microservices, since the services are separate the time taken for the deployment is greatly reduced.
5.	Since the code base of the architecture is tightly coupled and interconnected, the technology, frameworks, and languages, are extremely complicated to change as they highly depend on each other.	As the services are independent, it is easy to change the framework, technology or language depending on the functionality.

Through the figure and table, we can conclude that microservice architecture is better and capable of handling the rapidly changing market and technological advancements.

5.Microservices

A microservices system is a system that has distributed services that are independent and communicate with other services via messages. Microservice should be independent components deployed separately and equipped with their dedicated memory. As all the components are microservices the behaviour is derived from the composition and the communication via messages.

As shown in figure 5.1, the requests sent by the client is transferred to the distributed services by the API. These microservices communicate among themselves via messages and each microservice has its dedicated memory. Each service is running its process and communicates by using lightweight mechanisms. Using this architecture a vast server-based application having many modules is distributed into smaller services that are loosely coupled.

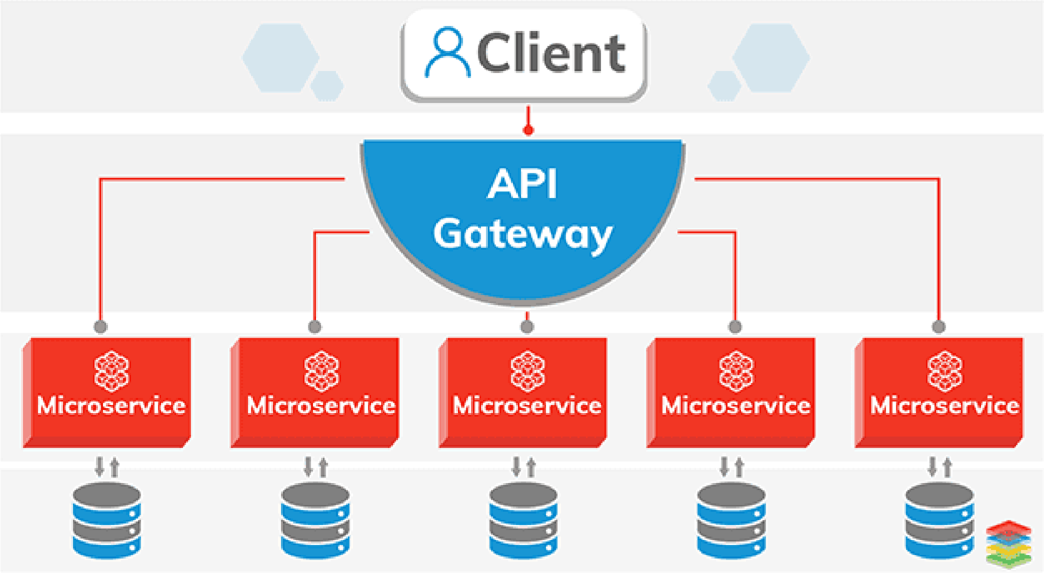


Figure 5.1: Microservices Architecture

5.1 Key Characteristics of Microservices

5.1.1 Flexibility

In today's world, systems have to keep up with the rapidly changing and growing technologies and market. For companies and organizations, they have to always stay active and competitive in the market[4]. To achieve this, microservices architecture is beneficial to adopt as it can easily support modifications and upgrades.

5.1.2 Modularity

Unlike monolithic architecture, where the system behaviour is dependent on a single component, in microservices architecture, the system behaviour depends on the functioning of all the modules[4].

5.1.3 Evolution

As the systems have to be able to evolve to the constantly growing technologies and markets, this characteristic of microservices architecture is beneficial[4]. In this architecture, systems can be modified or upgraded while keeping the system maintainable.

5.2 Advantages of Microservices

5.2.1 Independently Deployable

The services in the microservice architecture can be independently deployed, using this advantage, upgrades or modifications can be done with ease. In a scenario, where the microservices are distributed among teams and each team can focus solely on their respective microservice making it easier to develop.

5.2.2 Scalable

As the microservices are independent, the services can be scaled depending on the requirement[1]. The resources can be used to scale a service rather than scaling the entire system. This makes the process of scaling faster and also makes the process efficient.

5.2.3 Fault Isolation

If a service fails or crashes, rather than the entire system crashing, only a certain service will go down. In this case, that certain service can be isolated from the system and the service can still be rectified while the critical services are still active. Once the problem is solved, the service can be independently redeployed.

5.2.4 Maintainability

Since the services have smaller codes, it is easy to maintain the programs and the development teams can keep the codes clean. This helps the system to deploy quickly during maintenance checks as it's easier to maintain and deploy smaller codes.

5.3 Disadvantages of Microservices

Although microservices have a lot of benefits they come at a cost. Microservices systems are complex, costly, required skilled developers and also require security. Some of the disadvantages are stated below.

5.3.1 Complexity

As the microservices system scale, the system gets more complicated. The services have to communicate with each other via messages, the developers have to ensure that the services are well connected and the communication is secure.

While it is easy to test using microservices, it gets complicated to test the entire system as developers can't test the various components from individual machines.

5.3.2 Critical Interface Control

Each service has its respective API and the services are communicating with each other, if any microservice is affected, it may affect the entire system. And the APIs have to be consistent to make the system easier to maintain. Hence, controlling the APIs are critical.

5.3.3 Expensive

For microservices architecture to work, sufficient infrastructure with good security and maintenance is required[3]. Along with infrastructure skilled developers who understand the system and architecture are required. Thus, one of the disadvantages of microservice architecture is the resources and cost required for the smooth functioning of the architecture.

6.Applications of Microservices

As microservices architecture is extremely popular these days for the benefits it provides to huge organizations and companies. Below are some of the applications and usage of microservices.

6.1 Amazon

In the early 2000s, Amazon retail store was functioning on a very large monolithic architecture and as Amazon was scaling the architecture became more complicated and the software development cycle began to slow down.

In 2001, Amazon developers started to face development delays, coding challenges and the inability to scale because of their rapidly increasing customer base, Amazon broke down their monolithic architecture into smaller and independent services.

The developer analyzed the codes and sorted out the codes that are providing the same service and added these modules in a web-service interface. One of the examples is, the developers developed a service for the "Buy" button. This service sums all the items in the cart and calculates the taxes for the final cost of the products. Small teams of developers were assigned individual services so that they could resolve the challenge and solely focus on one service and work efficiently. Later on, to make the services communicate with the rest of the services it had to be passed through their web APIs. Amazon's service-oriented architecture is now known as microservices architecture.

Through microservices, Amazon developed solutions to support the microservices architecture which are AWS(Amazon Web Service) and Apollo that sell to enterprises all over the world. Unless Amazon hadn't made the transition to the microservices architecture it couldn't have become the most valuable company capped at $941.19 billion on Feb. 28, 2020[5].

The graphic below shows Amazon's microservices architecture in 2008, which is known as Death Star is shown in figure 6.1[5].

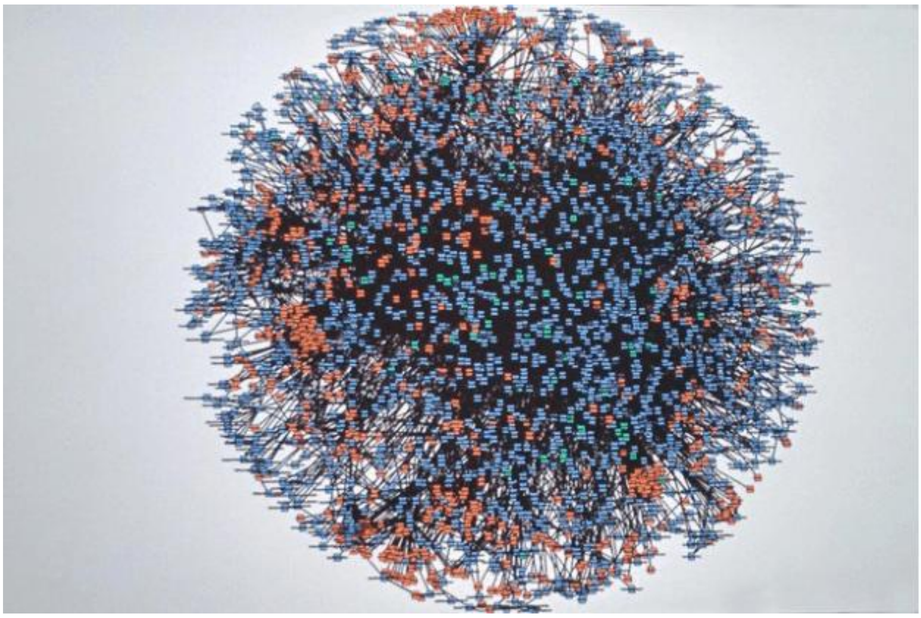


Figure 6.1: Death Star

6.2 Netflix

Netflix started to transition to the cloud in 2008 when it experienced a major corruption in the database and could not supply DVDs for 3 days. In 2009, Netflix began the gradual process of refactoring its monolithic architecture into smaller services i.e. microservices. Netflix decided to migrate the non-customer facing, movie coding platform on AWS cloud as a microservice. Netflix finalized this process in 2012.

Figure 6.2 shows the gradual transition of Netflix from a monolithic architecture to microservices architecture[5].

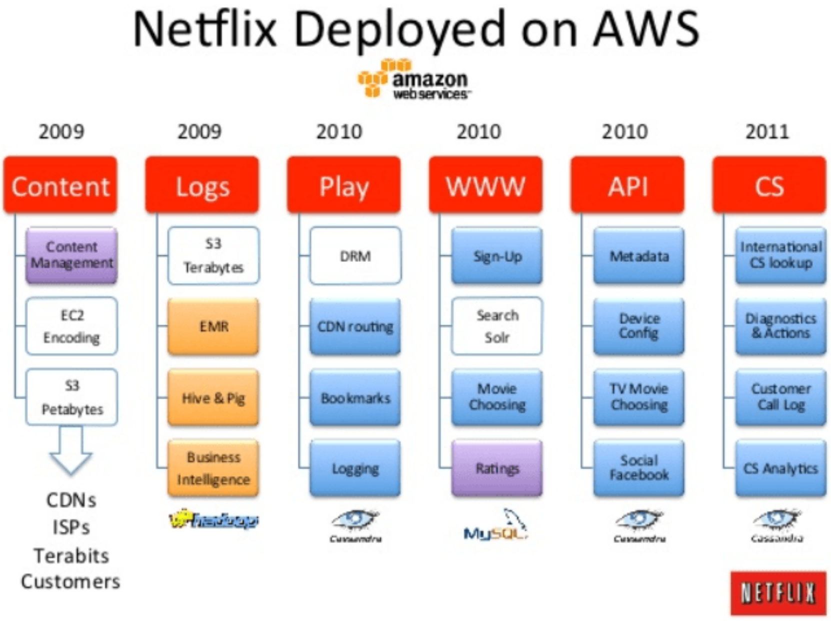
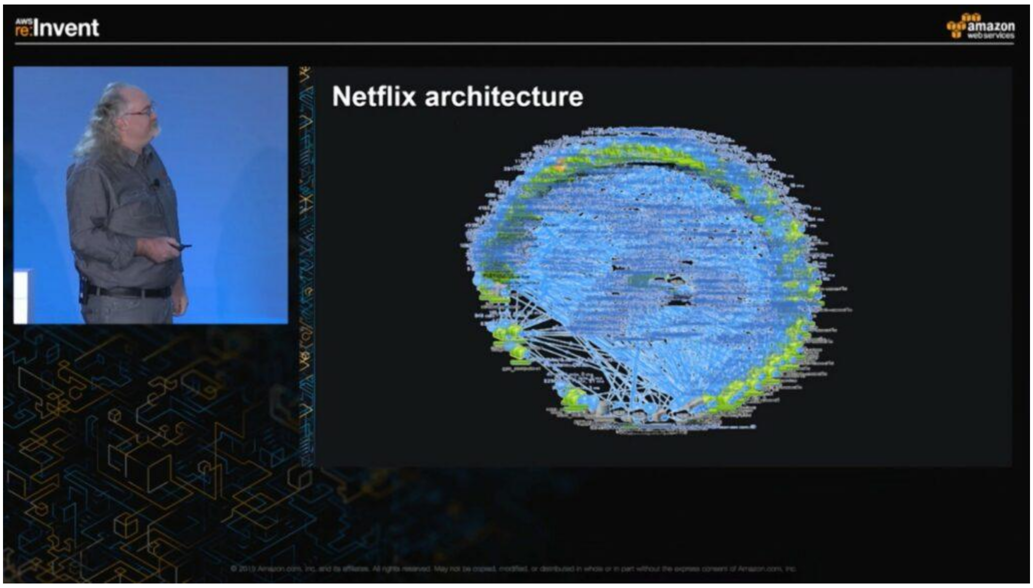


Figure 6.2: Netflix Transition

The transition of Netflix solved their scaling and service outage problems. By 2015, Netflix's API gateway handled 2 billion daily API requests that were managed by 500 cloud-hosted microservices[5].

Now, Netflix streams 250+ million hours of content to more than 139 million subscribers and is still evolving rapidly. Figure 6.3 is an image of Netflix's microservice architecture[5].



Netflix architecture

Figure 6.3: Graphic of Netflix's Microservices Architecture

6.3 Uber

Microservices was developed soon after Uber was founded. Uber used the monolithic architecture as the developers had to work on the existing systems, to make minor bug fixes. In the architecture, the driver and the passenger were connected using REST APIs. The embedded APIs were used for billing, payments and text messages, and it used a SQL Database. All these features were in the monolithic architecture. Figure 6.4 shows Uber's monolithic architecture[5].

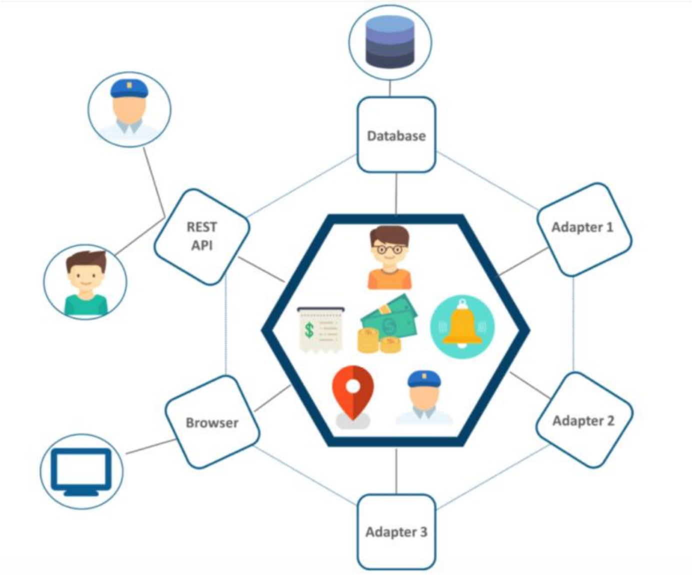


Figure 6.4: Uber's Monolithic Architecture

Later on, Uber struggled to develop and integrate new features, fixing bugs and scaling for global operations. Similar to Netflix, Uber broke down their monolithic architecture and turned to cloud-based microservices. Developers distributed these services as trip management, passenger management, etc. And again similar to Netflix, connected the services using API Gateways[5].

Figure 6.5 shows Uber's microservices architecture[5].

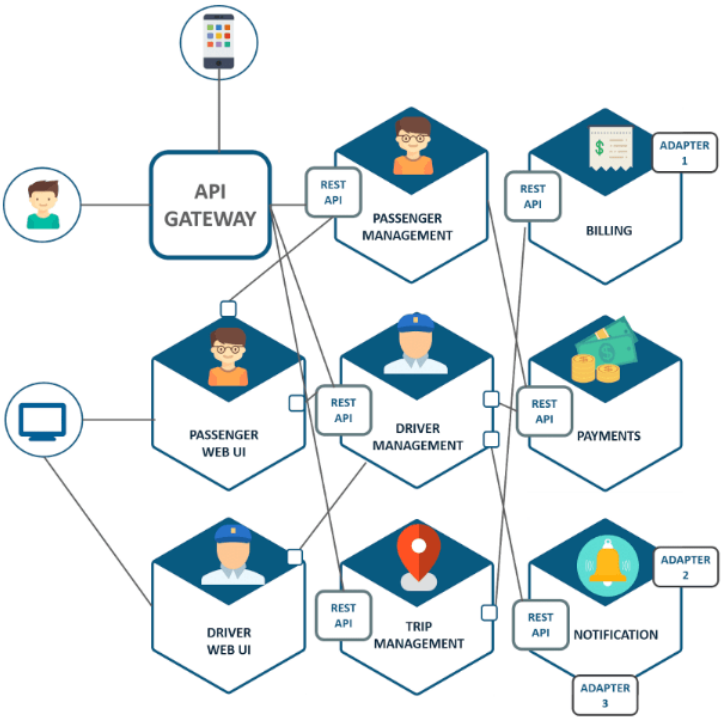


Figure 6.5: Uber's Microservices Architecture

The transition of Uber to the microservices architecture boosted the speed, quality and manageability of the scaling. Uber also got the ability to scale selected services without disrupting the other services and achieved better fault tolerance. During the scaling Uber faced a lot of problems, so they came up with global standardization of microservices to get better reliability and scalability.

To develop global standards, first, they analyzed documentation, performance, reliability, stability and scalability. Then, they established standards that can be measured by business metrics i.e. website views. Finally, they converted the metrics into "requests per second on a microservice".

Figure 6.6 shows an image of Uber's microservices architecture in 2019[5].

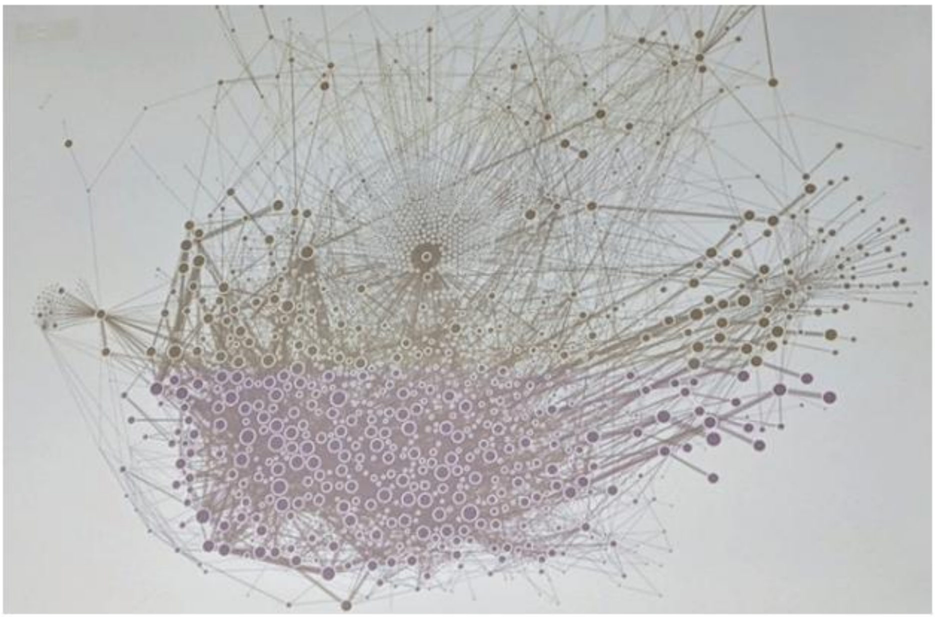


Figure 6.6: Graphic of Uber's Microservices Architecture

7.Future Development of Microservices

Microservices architecture is still a recent technology that has changed server-based applications drastically. Although with all the advantages and disadvantages it has brought to the table, there is still a huge scope for development and betterment. Some of the changes like network complexity require technical advancements in their respective field which can then be beneficial to the microservices architecture. Some of the fields of development of microservices are given below:

7.1 Security

In monolithic architectures, the services communicate with each other using internal communications and the entire system is constrained to one operating system. But in the microservices architecture, since the server-side application is broken down into smaller services and the communication is done by using web APIs which are exposed to the network[2].

7.2 Network

Due to multiple services communicating with each other, the network activity becomes complicated. With the increase in network complexity, it gets difficult to enforce security and applications on the microservices architecture. And the more complicated the service becomes, it gets more difficult to debug and make changes in the service[1]. Also, to have a reliable, fast and efficient system there should be a good network and the message size used for communicating should also be as small as possible.

8.conclusion

To evaluate the seminar topic, we have gone through the problem statement i.e. to study microservices, the advantages of microservices and how industries, companies and organizations have used microservices architecture to grow and solve the problems they were facing. Microservices have played a huge role in the growth of companies like Amazon, Netflix, eBay, Uber, and many more.

Looking back at this topic, microservices architecture is surely beneficial for industries, markets and technologies these days, but on the other hand, it is expensive and resource-heavy to apply, maintain and ensure the proper functioning of the server-based application. Since this architecture is still recent, there are many more developments to be made for its security, network and complication problems. Some of these problems can be solved but the major issues of network and security can be solved if technical advancements are made in their respective fields. Apart from future development, according to this study, companies and organizations have implemented this architecture when there was a rise in clients or customers, when they wanted to scale their systems or when they faced technical issues that had to be resolved by changing the architecture for better use in the future. And as the market and technology grow, the functionality and application of microservices will also grow.

REFERENCES

[1] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina, “Microservices: Yesterday, Today, and Tomorrow”, Springer, pp 195-216, Sept. 2017.

[2] Luciano Baresi and Martin Garriga, “Microservices: The Evolution and Extinction of Web Services?”, Springer, pp 3-28, Dec. 2019.

[3] Daniel Escobar, Diana Cardenas, Rolando Amarillo, Eddie Castro, ´ Kelly Garces, Carlos Parra, Rubby Casallas, “Towards the Understanding and Evolution of Monolithic Applications as Microservices”, IEEE, Jan. 2016.

[4] Sam Newman, “Building Microservices: Designing Fine-Grained Systems”, O'Reilly Books, Feb. 2016.

[5] Jeremy Hillpot, 14th July 2021, 4 Microservices Examples, https://blog.dreamfactory.com/microservices-examples/