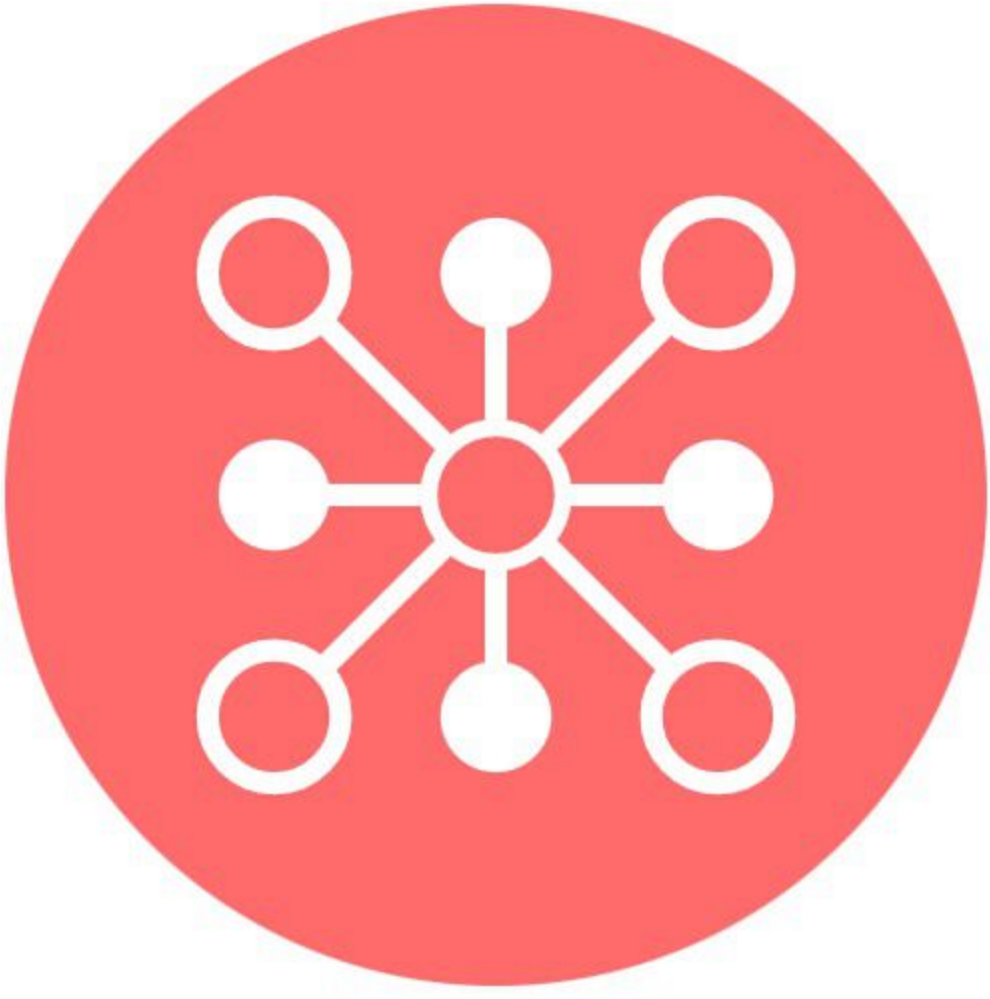
What Is

MICROSERVICES

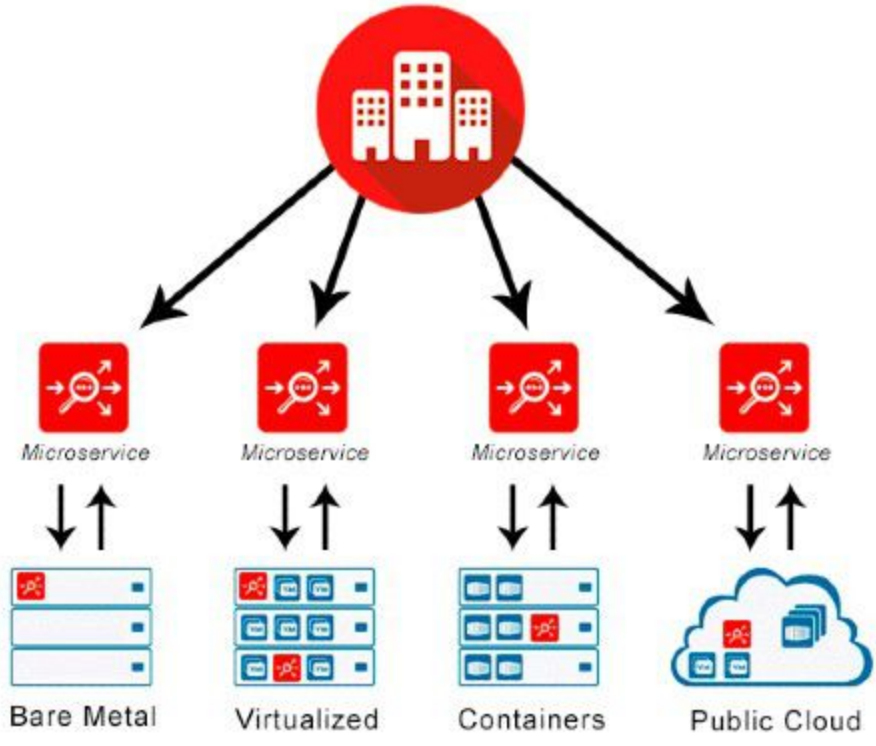


1- Microservices Definition

Microservices is an architectural design for building a distributed application using containers. They get their name because each function of the application operates as an independent service.

This architecture allows for each service to scale or update without disrupting other services in the application.

Microservices Architecture

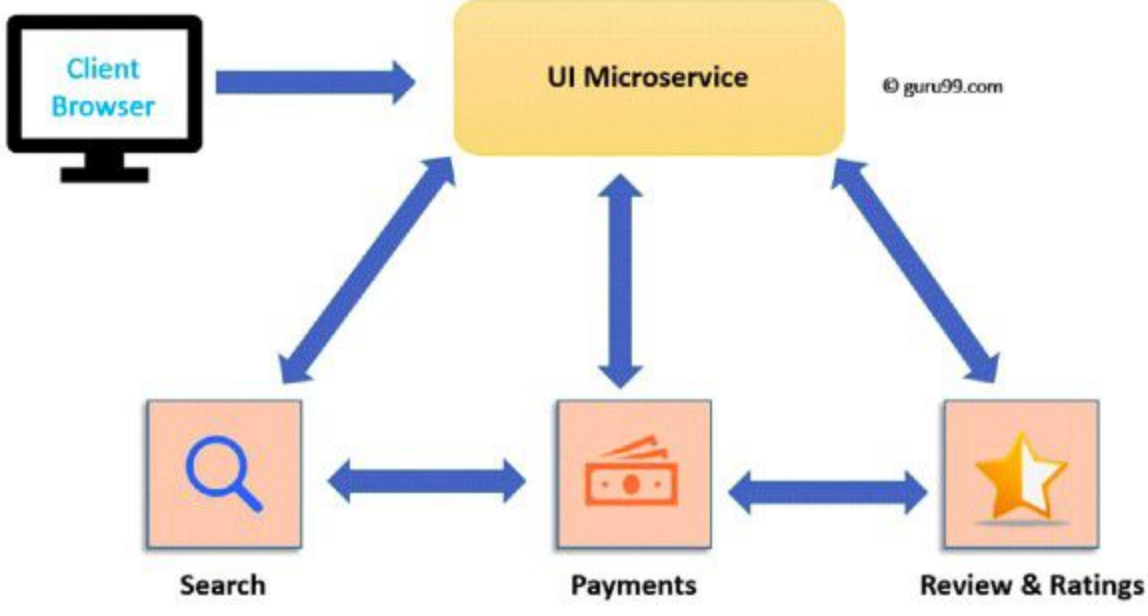


2- What is Microservice Architecture?

Microservice Architecture is an architectural development style that allows building applications as a collection of small autonomous services developed for a business domain. It is a variant of structural style architecture that helps arrange applications as a loosely coupled service collection. The Microservice Architecture contains fine-grained services and lightweight protocols.

Let's take an example of e-commerce application developed with microservice architecture. In this Microservices architecture example, each microservice is focused on single business capability. Search, Rating & Review and Payment each have their instance (server) and communicate with each other.

Microservices architecture E-Commerce Application



3- When to Use Microservices?

Ultimately, any size company can benefit from the use of a microservices architecture if they have applications that need frequent updates, experience dynamic traffic patterns, or require near real-time communication.

4- Who Uses Microservices?

Social media companies like Facebook and Twitter, retailers like Amazon, media provider like Netflix, ride-sharing services like Uber and Lyft, and many of the world's largest financial services companies all use microservices.

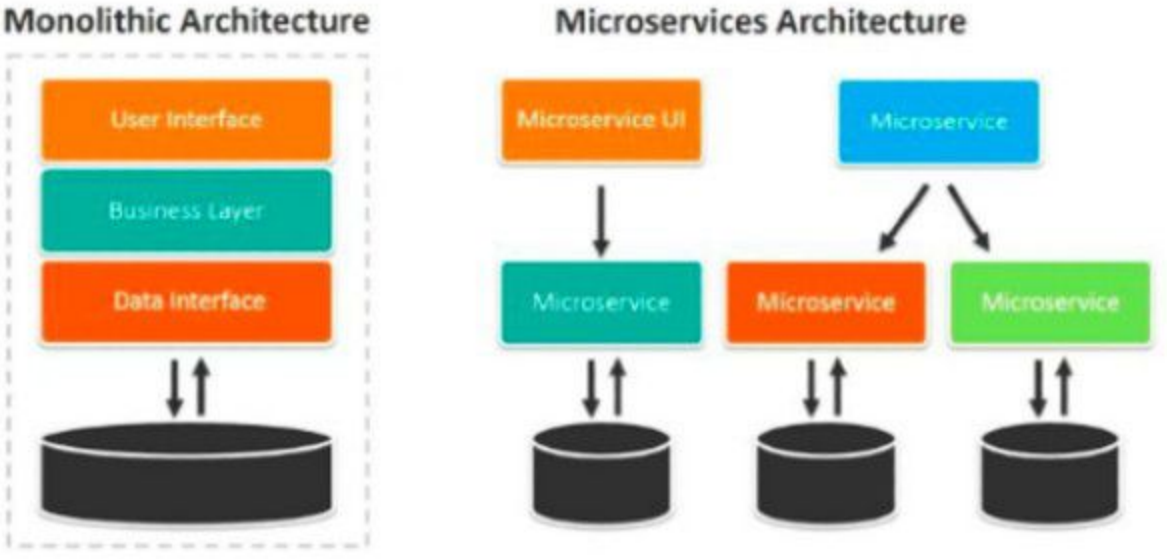
The trend has seen enterprises moving from a monolithic architecture to microservices applications, setting new standards for container technology and proving the benefits of using this architectural design.

5- Monolithic architecture vs microservice architecture

The monolithic architecture pattern has been the architectural style used in the past, pre-Kubernetes and cloud services days.

In a monolithic architecture, the software is a single application distributed on a CD-ROM, released once a year with the newest updates.

Examples are Photoshop CS6 or Microsoft 2008.



That style was the standard way of building software. But as tech has evolved, so too the architectural style must advance. In an age of Kubernetes, and CI/CD workflows, the monolithic architecture encounters many limitations—companies need to push to microservices.

5-1- Characteristics of a monolithic architecture:

- Changes are slow
- Changes are costly
- Hard to adapt to a specific, or changing, product line

Monolithic structures make changes to the application extremely slow. Modifying just a small section of code can require a completely rebuilt and deployed version of software.

If developers wish to scale certain functions of an application, they must scale the entire application, further complicating changes and updates. Microservices help to solve these challenges.

6- What are the Benefits of Microservices?

- Increase agility
- Improve workflows
- Decrease the amount of time it takes to improve production

Microservices are small applications that your development teams create independently. Since they communicate via messaging if at all, they're not dependent on the same coding language.

Developers can use the programming language that they're most familiar with. This helps them come work faster, with lower costs and fewer bugs.

Microservices improve your architecture's scalability, too.

Also, as more and more enterprises embrace the cloud, you're probably looking that way, too. Microservices are a great way to get there.

7- Disadvantages of microservices

Of course, the microservice architecture comes with a learning curve. First-time users might struggle to determine:

- Each microservice's size.
- Optimal boundaries and connection points between microservices.
- The right framework to integrate services.

More broadly, microservices have these drawbacks:

- Increased complexity
- More expensive
- Greater security risks

8- How Are Microservices Deployed?

Deployment of microservices requires the following:

- Ability to scale simultaneously among many applications, even when each service has different amounts of traffic
- Quickly building microservices which are independently deployable from others
- Failure in one microservice must not affect any of the other services

Docker is a standard way to deploy microservices using the following steps:

- Package the microservice as a container image
- Deploy each service instance as a container
- Scaling is done based on changing the number of container instances

9- Best Practices for Microservices

You should be able to tell by now that making the shift to microservices creates a lot of benefits for development, operations, and the business.

They create opportunities for increased scalability, greater reliability, and cost savings.

Here are some best practices that will help your migration.

- Separate data store for each Microservice
- Keep code of a similar level of maturity.
- Separate build for each Micro service.
- Always treat – severe as stateless.