Microservices

1.Why Microservices?

Ans:-

2.)what is Microservices?

Ans:- Essentially, microservice architecture is a method of developing software applications as a suite of independently deployable, small, modular services in which each service runs a unique **process** and communicates through a well-defined,**lightweight** mechanism to serve a business goal.

**Microservices** is a software development technique—a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a **microservices** architecture, services are fine-grained and the protocols are lightweight.

3.Advantages of Microservices?

Ans: There are several benefits to using microservices. For one, because these smaller applications are not dependent on the same coding language, the developers can use the programming language that they are most familiar with. That helps developers come up with a program faster with lower costs and fewer bugs. The agility and low costs can also come from being able to reuse these smaller programs on other projects, making it more efficient.

4. What is the difference between SOA and Microservices?

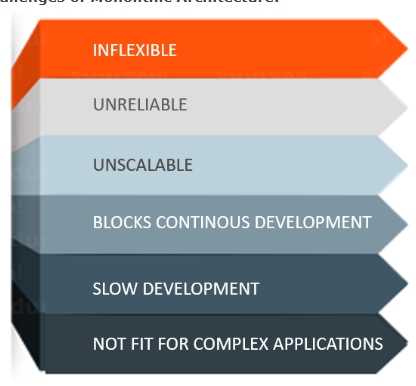
* [Service-oriented architecture](https://www.google.com/url?cad=rja&cd=2&ei=lK2mVebMONbVoATZu5-QAQ&esrc=s&q=&rct=j&sa=t&sig2=_KOGOF7b1rSUs2z2tp-X4Q&source=web&uact=8&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FService-oriented_architecture&usg=AFQjCNFsplk-0iL7kL9ZH1GmrTtkoxYJdw&ved=0CCIQFjABahUKEwjm5pbx6N3GAhXWKogKHdndBxI) (SOA): an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network.
* [Microservices](https://www.google.com/url?cad=rja&cd=2&ei=4q2mVbPNB8PooATrzrbQAw&esrc=s&q=&rct=j&sa=t&sig2=lLIdau8wZ2vfvtjI00vczA&source=web&uact=8&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMicroservices&usg=AFQjCNHQptqemmuXl_mDHLJK2pBkbvozKQ&ved=0CCEQFjABahUKEwizxv6V6d3GAhVDNIgKHWunDTo):  a software architecture style in which complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs

The core **difference between SOA** and **micro services** lies **in the** size and scope. As the word "micro" suggests, it has to be significantly smaller than what **SOA** tends to be. ... **Micro services** must be independently deployable, whereas **SOA** services are often implemented in deployment monoliths.

**1.)What is Monolithic Application?**

Ans:A container which contains the all software components together packed and tightly couple is called Monolithic.

Disadvantages of Monolithic application are given below.



1.Inflexible:-Monolithics application can not be built with different technologies.

2.Unreliable:-Even one component does not work entire application don’t work.

3.Unscalable:-Application can not be scaled easily since each time system need to be updated and built the entire system.

4.Block continous development:-Many features of application can not be built and deployee at the same time.

5.slow development:-Development in Monolithics application take more time to build because need to be build each and every feature after one and other.

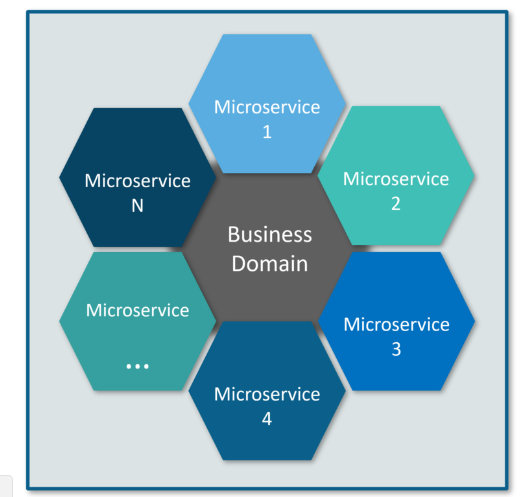
6.Not fit for the complex application:-Feature of complex application is tightly coupled dependency.

The main reasone given above led to evaluation of Microservices.

2**.)What is Microservices?**

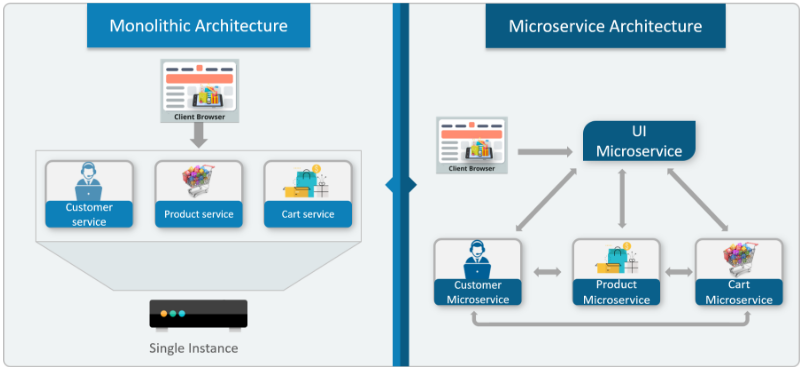
Ans-Microservices is an architectural style which structure the application as collection of small autonomous services ,modeled around the **business domain**.

In Microservices, Each services of an application is self contented and implements sinlge business capability.



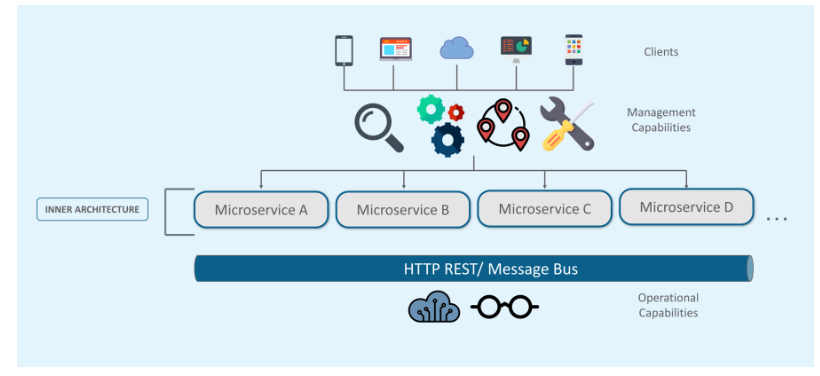
**3.)Differences Between Traditional Architecture and Microservices.**

Consider a E-commerce application as a use-case to understand the both monolithic and Microservices application.



By Looking above diagram ,We can observe that ,Initially all features were at sinlge instance sharing single database. But with Microservices each feature is assigned to each service which has its own business logic to perform the task independently.

**Microservices Architecture**



**In this architecture ,**

All client from different devices want different services like Search, build ,Configure and other management capabilities .

All client are seprated based on their domain and functionalities and further allotted to individual services.

Each microservies has own **load balancer** and **execution environment** to execute its functionality and same time capable to capture the data from its database.

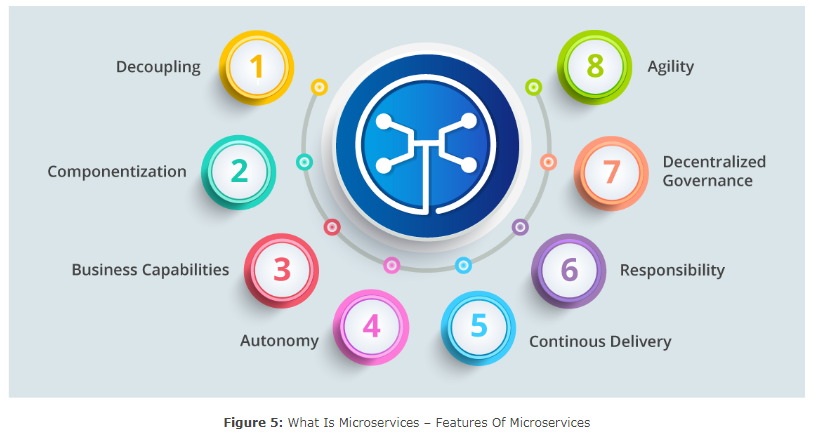
All the services communicate each other through stateless server either using **REST** or **Message bus.**

Microservices know their path of communication with the help of Service Discovery and perform operational capabilities such as automation, monitoring

Then all the functionalities performed by microservices are communicated to clients via API Gateway

All the internal points are connected from the API Gateway. So, anybody who connects to the API Gateway automatically gets connected to the complete system.

## ****Microservices Features****



References:

<https://www.edureka.co/blog/what-is-microservices/>

<https://stackify.com/what-are-microservices/>

<https://www.tutorialspoint.com/microservice_architecture/index.htm>

<https://dzone.com/articles/java-microservices-code-examples-tutorials-and-more>

<https://www.edureka.co/blog/microservices-tutorial-with-example>