**Introduction to Micro-Service Architecture:**

Have you ever heard the word Micro-service architecture and got over-whelmed by the fancy word. Don’t worry in this article we will get you through the basics of micro service architecture in quite a simple way.

**Micro-Service Architecture:**

Before we get to technical side lets understand what word “Micro” ,”service” and Architecture means on their own so that when we hear this word we can intuitively tell what it tells about itself.

1. Micro: This word just means “small” like used in micro-computer (a “small” computer) ,micro-processor(a “small” processor) ,etc..
2. Service: "The action of doing work for someone”
3. Architecture: “The complex or carefully designed structure of something”

Now that you have got what each word means individually now let’s look at their collective meaning:

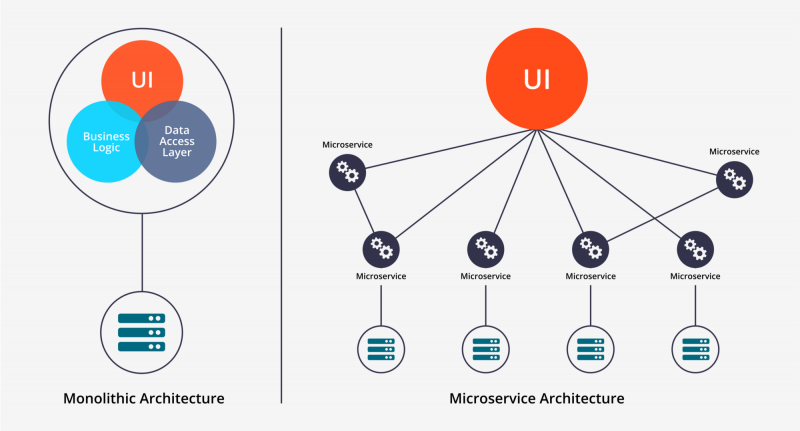
**“Micro-Service architecture is an architectural style that structures an application as a collection of services”**

It is just a concept of arranging and designing your components in such a way that its each component can work independent of other components of that system or in other words are decoupled from other components. This architecture is based on ancient and proven technique **“Divide and conquer”**

Now that you have got fair idea of what micro-service architecture is let’s move on to why the need of this architecture was felt.

**Why Micro-Service Architecture?**

Before the Micro-service architecture only choice architects had was monolithic architecture in which all the components are quite dependent on each other or in other words are tightly coupled i.e. changing one functionality might break down the entire software application, therefore code was harder to maintain and harder to scale up the software. The micro-service provided a way to escape through this bottle-neck.but as all good things come with a cost there are also some downside of this architecture which now will be discussed



**Downside of Micro-service Architecture:**

Since all components works independently it is more vulnerable to hacking and data breaches since there are many more end points which can be exploited as compare to monolithic

There is a higher chance of failure during communication between different serviced

**Advantages of Microservices:**

1. Isolated services have better fault tolerance
2. Adding and removing of functionality Is easier
3. Shorter release cycles
4. Microservices are platform agnostic. It means we can design and deploy them independently without affecting the other services.

**Future of Monolithic :**

Since the Micro-services are gaining popularity it doesn’t mean monolithic is going to die but it will continue to be used for small and medium size projects because applying micro-service architecture can cause excessive complexity for these kinds of projects. The micro services also shifts complexity from developer side to operations which require extra resources such as dev-Ops. So usually Micro-services is not considered to be a good choice for these kind of projects