**The Best Architecture to choose for developing your Business | Monolithic Or Microservice?**

For last couple of years there has been a trend of comparison between two developing architecture –monolithic and microservice in tech industry. It’s not like that these things were never before but it’s the fact that after the Giants (ex-Google,Amazon,facebook,Netflix etc) has taken microservice architecture for their products due to its advantages of scalability agility and flexibility, microservice became the new trend whereas due to same issues the monolithic architecture is going down. So, to get a better idea about what is this all about and why someone should choose which architecture, I am dividing my overview into three parts below.

What is Monolithic and Microservice Architecture?

We need to know first that what is Monolithic and what is Microservice architecture by theory. Monolithic architecture is the typical way of building applications. It builds the application as a single and indivisible unit. Usually such results in comprises a client-side interface a server side application and a database. On the contrary, Microservice architecture breaks down an application into a collection of smaller units where these units carry out each application process as a separate service. Within a microservice architecture the entire program is splited up into independently deployable modules, which communicate with each other through defined methods, called APIs.

Robust and Fragile sides of Monolithic and Microservice Architecture

The main robust points of monolithic architecture are its Less cross-cutting concerns, easier debugging, simple to deploy and simple to develop opportunities. For instance, in monolithic application cross cutting concerns such as logging, handling, caching and monitoring performance is less hectic since it concerns only one application. Besides Since it concerns only one application as a whole so it is easy to deploy and develop the whole application with the small team with minimal required knowledge. On the other hand, the robust points of microservices is it’s independent component, easier to understand and it’s better scalability and flexibility in choosing tech and agility. For instance, if some company has to build a big application, which has to maintain huge, pressure of user load, transaction, and what not. In this case, they choose microservice architecture to break its services into independent components .since its divided into small units so it’s easier to understand as a component and in terms of scalability it’s more cost and time effective since each element can be scaled independently. Most importantly microservices architecture provides flexibility to choose new tech if needs in future since it does not require to change the application as a whole.

The fragile side of monolithic architecture is, it becomes hard to understand as it scales up. Besides, making changes is difficult here and as it grows adopting new technology becomes hard. Whereas in microservice architecture, since it’s a distributed system it creates extra complexity to set up connection between them and distribute the system. Besides, Testing the application becomes tough in microservice applications.

So,which Architecture to choose for your Business?

It will be a bad decision to choose the microservice architecture for your application since it is a buzzword and trending now. If you understand your business demand and resources first then it is handy to go for one of the architecture. For example, if your business is fresh and you have a small team with not so microservice expertise and engineering skills and if your business is not a scalable complex application then you should go for monolithic architecture. whereas, if you have all exactly the opposite of above and you have enough resources with enough engineering skills and if your business covers a vast amount of user interactions then you can go for microservice architecture to validate your business. That would be beneficial for you in that case.