When Should You Pick a Monolithic Architecture?

In this lesson, we will learn about the pros and cons of a Monolithic Architecture & when to choose it for our project.

We'll cover the following

Pros Of Monolithic Architecture

Simplicity

Monolithic applications are simple to develop, test, deploy, monitor and manage since everything resides in one repository.

There is no complexity of handling different components, making them work in conjunction with each other, monitoring several different components & stuff. Things are simple.

Cons Of Monolithic Architecture

Continuous Deployment

Continuous deployment is a pain in case of monolithic applications as even a minor code change in a layer needs a re-deployment of the entire application.

Regression Testing

The downside of this is that we need a thorough *regression testing* of the entire application after the deployment is done as the layers are tightly coupled with each other. A change in one layer impacts other layers significantly.

Single Points Of Failure

Monolithic applications have a *single point of failure*. In case any of the layers has a bug, it has the potential to take down the entire application.

Scalability Issues

Flexibility and scalability are a challenge in monolith apps as a change in one layer often needs a change and testing in all the layers. As the code size increases, things might get a bit tricky to manage.

Cannot Leverage Heterogeneous Technologies

Building complex applications with a monolithic architecture is tricky as using heterogeneous technologies is difficult in a single codebase due to the compatibility issues.

It's tricky to use *Java* & *NodeJS* together in a single codebase, & when I say tricky, I am being generous. I am not sure if it's even possible to do that.

Not Cloud-Ready, Hold State

Generally, monolithic applications are not cloud-ready as they hold state in the static variables. An application to be cloud-native, to work smoothly & to be consistent on the cloud has to be distributed and stateless.

When Should You Pick A Monolithic Architecture?

Monolithic applications fit best for use cases where the requirements are pretty simple, the app is expected to handle a limited amount of traffic. One example of this is an internal tax calculation app of an organization or a similar open public tool.

These are the use cases where the business is certain that there won't be an exponential growth in the user base and the traffic over time.

There are also instances where the dev teams decide to start with a monolithic

There are also instances where the dev teams decide to start with a monolithic architecture and later scale out to a distributed microservices architecture.

This helps them deal with the complexity of the application step by step as and when required. This is exactly what LinkedIn did.

In the next lesson, we will learn about the Microservice architecture.