When should you pick a Monolithic Architecture?

In this lesson, you will learn about the pros and cons of monolithic architecture and when to choose it for your project.

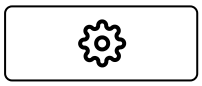
We'll cover the following

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, and what not. Things are simple.

Cons of monolithic architecture#

Continuous deployment#

Continuous deployment is a pain in monolithic applications because even a minor code change in a layer necessitates 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 because the layers are tightly coupled with each other. A change in one layer impacts the other layers significantly.

Single points of failure#

Monolithic applications have a *single point of failure*. If any of the layers has a bug, it can take down the entire application.

Scalability issues#

Flexibility and scalability are a challenge in monolith apps because a change in one layer often necessitates a change and testing in all the

change in one layer often necessitates 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 are tricky because using heterogeneous technologies is difficult in a single codebase due to compatibility issues.

It is tricky to use *Java* and *NodeJS* together in a single codebase, and when I say tricky, I am being generous. I am not sure if it is even possible.

Not cloud-ready, hold state#

Generally, monolithic applications are not cloud-ready because they hold state in the static variables. An application to be cloud-native, to work smoothly, and 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, and the app is expected to handle a limited amount of traffic. One example of this is an organization's internal tax calculation app or a similar open public tool.

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

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 when required. This is exactly what LinkedIn did. (https://engineering.linkedin.com/architecture/brief-history-scaling-linkedin)

In the next lesson, you will learn about microservice architecture.

← Back

What is Monolithic Architecture?

Next →

What is Microservice Architecture?

Mark as Completed

Report an Issue