What Is A Monolithic Architecture?

In this lesson, we will discuss the Monolithic Architecture.

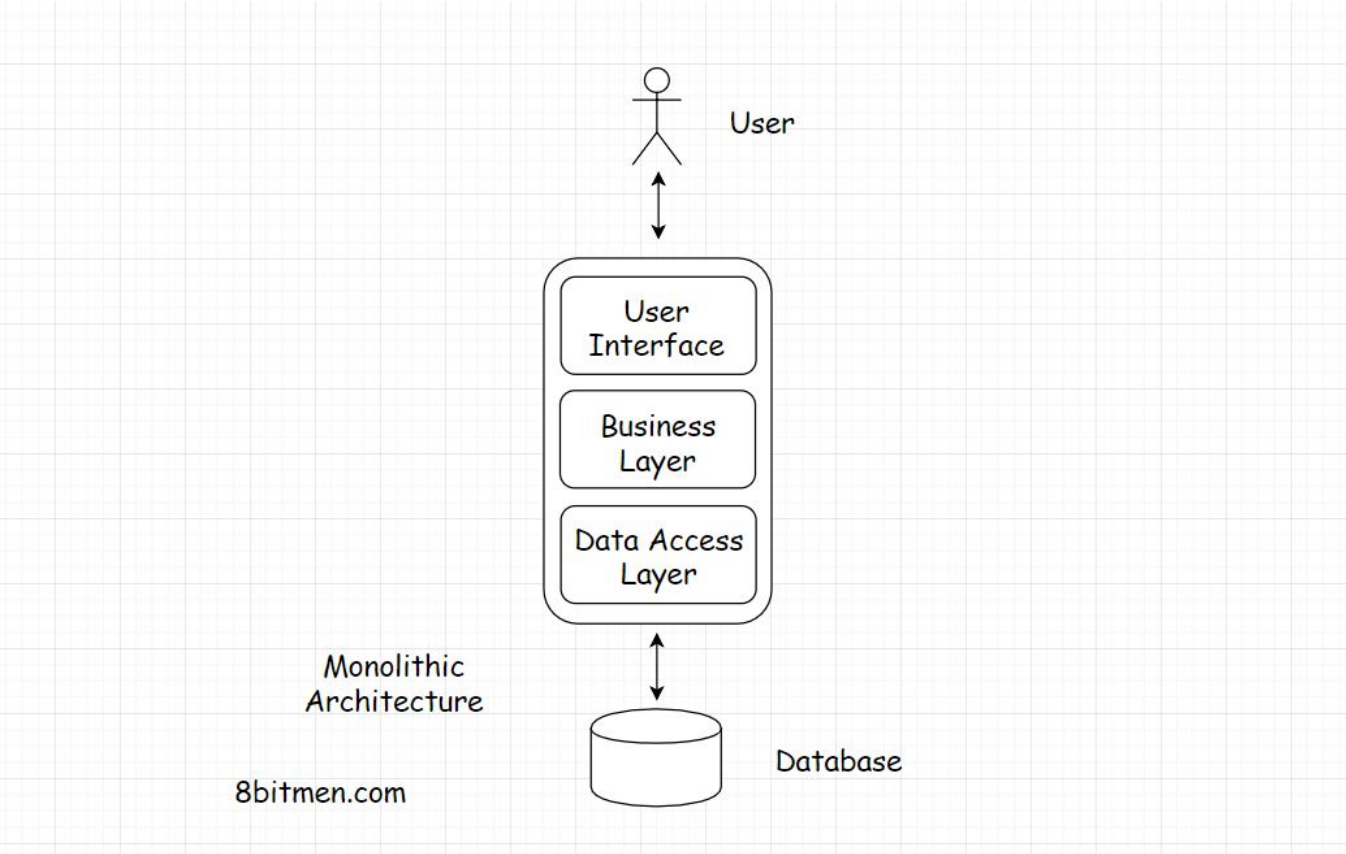
We'll cover the following

What Is A Monolithic Architecture?

> An application has a monolithic architecture if it contains the entire application code in a single codebase.

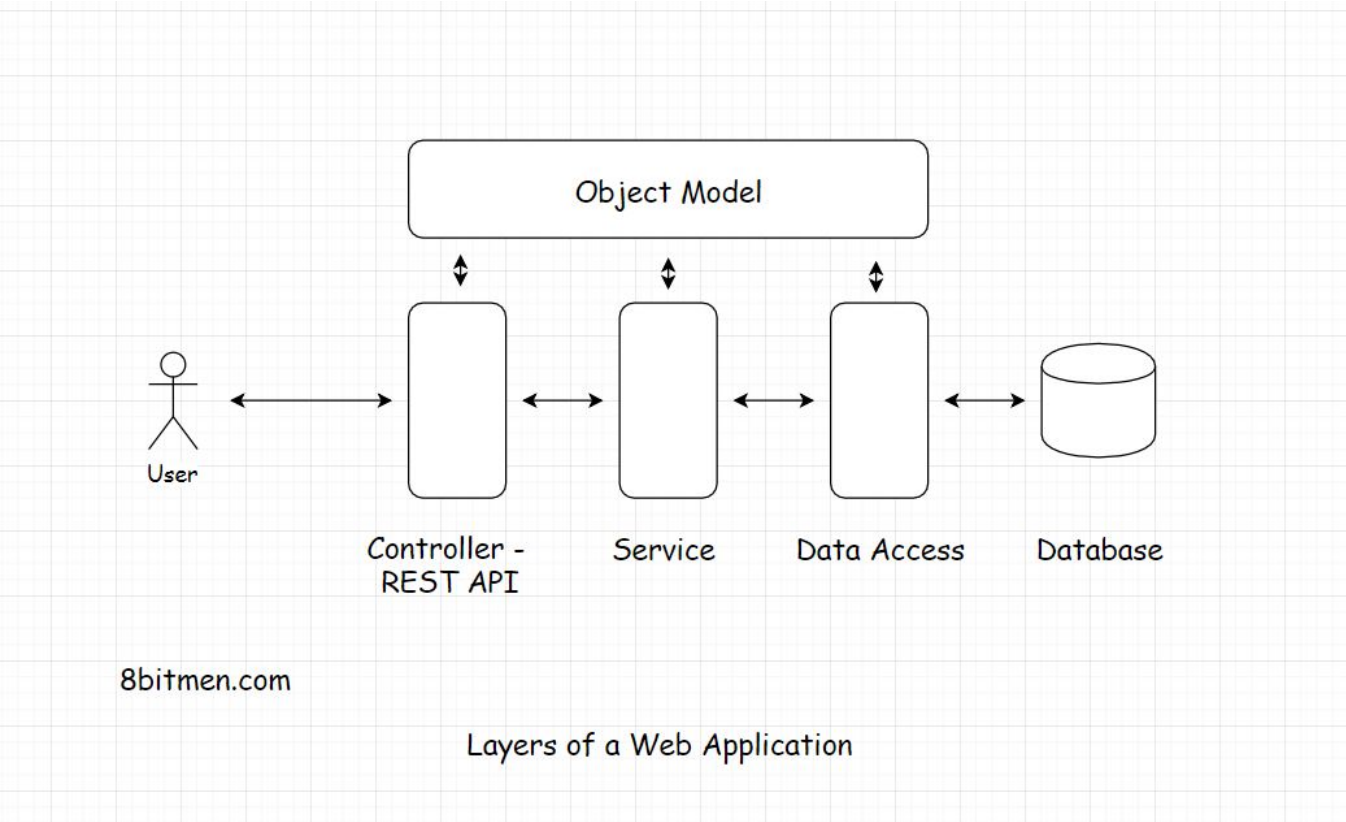
A monolithic application is a self-contained, single-tiered software application unlike the microservices architecture, where different modules are responsible for running respective tasks and features of an app.

The diagram below represents a monolithic architecture:



In a monolithic web-app all the different layers of the app, UI, business, data access etc. are in the same codebase.

We have the *Controller*, then the *Service Layer interface*, *Class* implementations of the interface, the *business logic* goes in the *Object Domain model*, a bit in the Service, Business and the *Repository/DAO [Data Access Object]* classes.



Monolithic apps are simple to build, test & deploy in comparison to a microservices architecture.

There are times during the initial stages of the business when teams chose to move forward with the monolithic architecture & then later intend to branch out into the distributed, microservices architecture.

Well, this decision has several trade-offs. And there is no standard solution to this.

In the present computing landscape, the applications are being built & deployed on the cloud. A wise decision would be to pick the loosely coupled stateless microservices architecture right from the start if you expect things to grow at quite a pace in the future.

Because re-writing stuff has its costs. Stripping down things in a tightly coupled architecture & re-writing stuff demands a lot of resources & time.

On the flip side, if your requirements are simple why bother writing a microservices architecture? Running different modules in conjunction with each other isn't a walk in the park.

Let's go through some of the pros and cons of monolithic architecture.