What Is A Microservice Architecture?

In this lesson, we will learn about the Microservice Architecture.

We'll cover the following

What Is A Microservices Architecture?

> In a microservices architecture, different features/tasks are split into separate respective modules/codebases which work in conjunction with each other forming a large service as a whole.

Remember the *Single Responsibility* & the *Separation of Concerns* principles? Both the principles are applied in a microservices architecture.

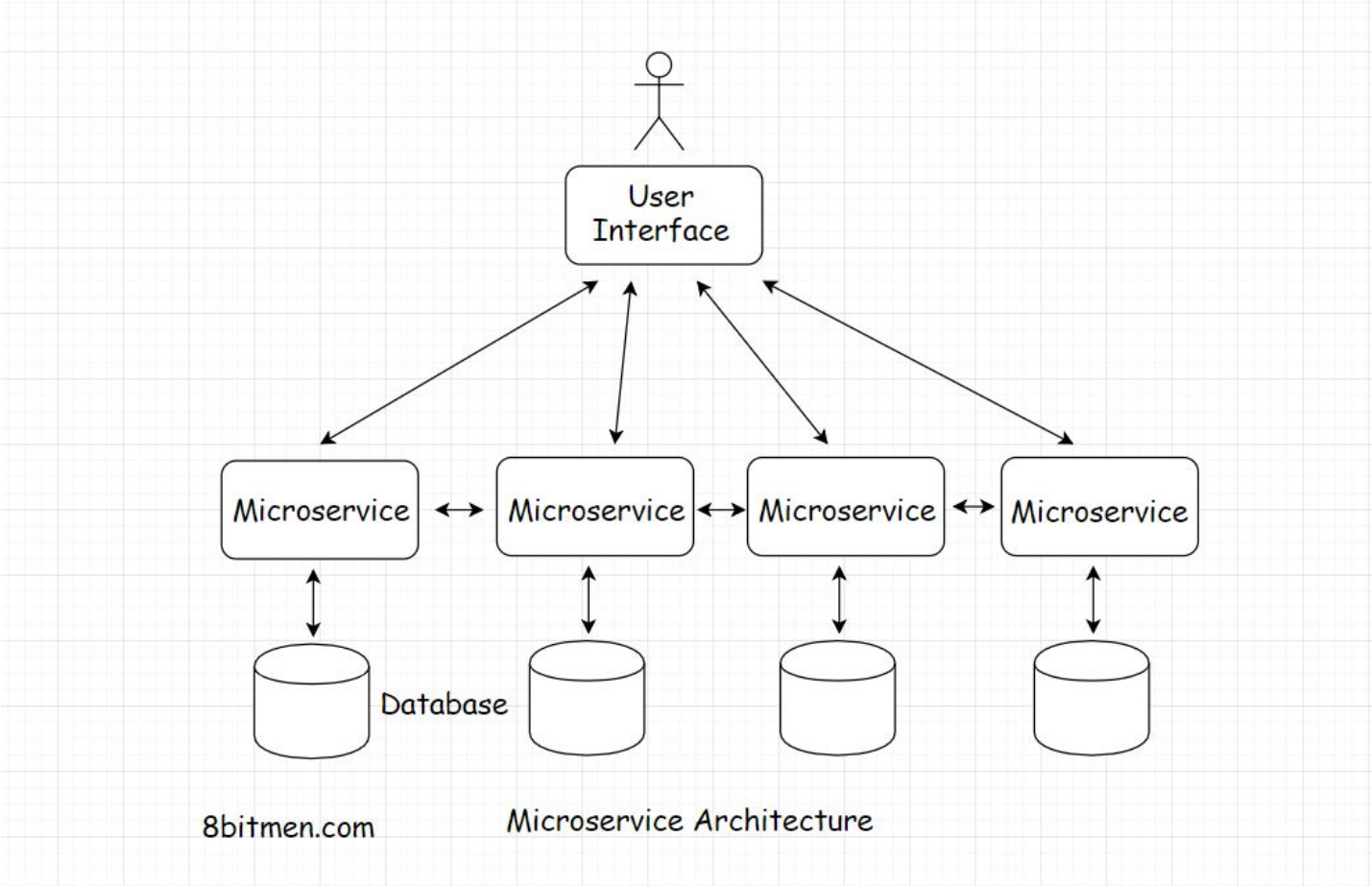
This particular architecture facilitates easier & cleaner app maintenance, feature development, testing & deployment in comparison to a monolithic architecture.

Imagine accommodating every feature in a single repository. **How complex things would be?** It would be a maintenance nightmare.

Also, since the project is large, it is expected to be managed by several different teams. When modules are separate, they can be assigned to respective teams with minimum fuss, smoothening out the development process.

And did I bring up scalability? To scale, we need to split things up. We need to scale out when we can’t scale up further. Microservices architecture is inherently designed to scale.

The diagram below represents a microservices architecture:



Every service ideally has a separate database, there are no single points of failure & system bottlenecks.

Let’s go through some of the pros and cons of using a microservices architecture.