What is Microservice Architecture?

In this lesson, you will learn about microservice architecture.

We'll cover the following

What is microservices architecture?#

> In a microservices architecture, different features/tasks are split into separate respective modules/codebases that work in conjunction to form a large service as a whole.

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

This particular architecture facilitates easier and cleaner app maintenance, feature development, testing, and deployment compared to a monolithic architecture.

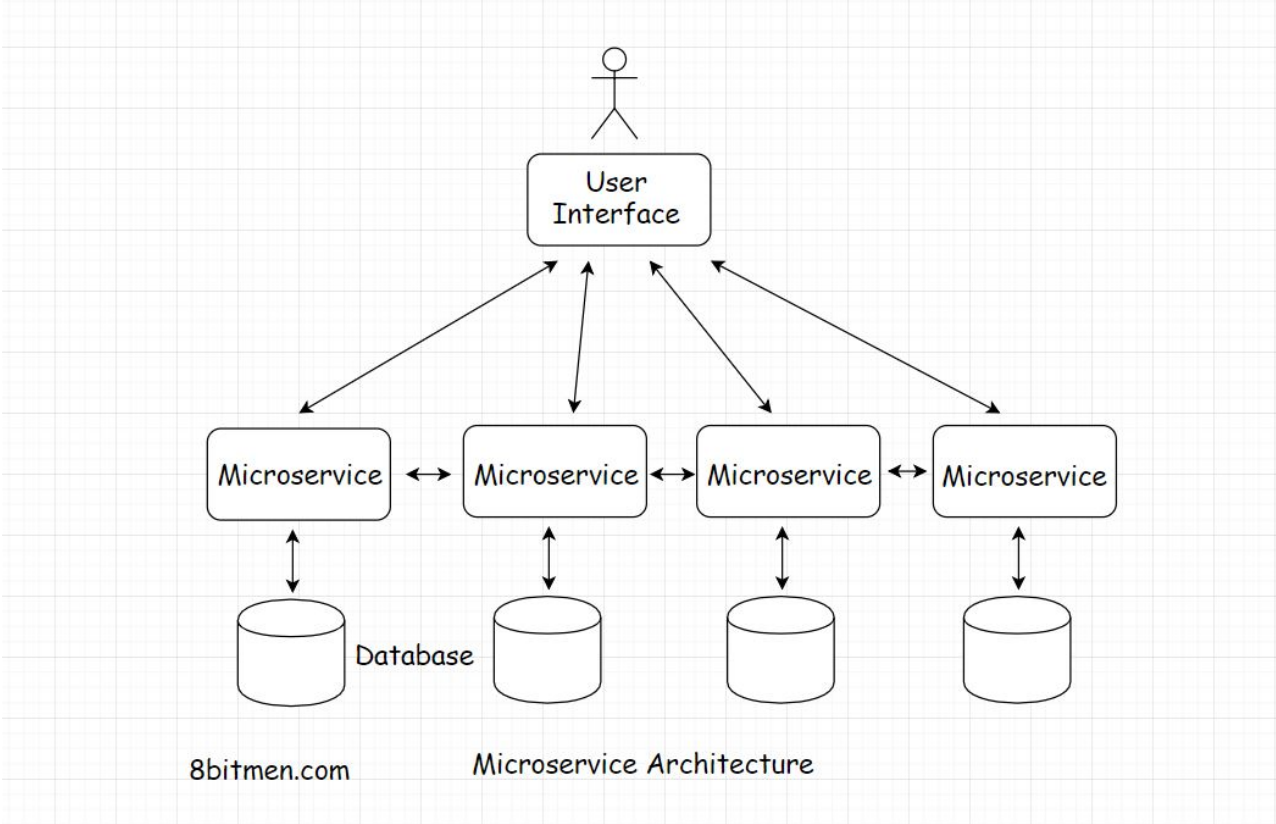
Imagine accommodating every feature in a single repository. **How complex would things be?** It would be a maintenance nightmare.

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 minimal fuss, smoothing out the development

respective teams with minimal fuss, smoothing out the development process.

Did I mention 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, so there are no single points of failure or system bottlenecks.

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

← Back
When should you pick a Monolithic Ar...
Next →
When should you pick Microservices ...
Mark as Completed
Report an Issue