

MAIN SOFTWARE ARCHITECTURE SYSTEMS

1. Main Software Architecture Systems

00:00 - 00:16

Learning about software architecture is a rewarding process. Understanding different software architectures could make life easier for any development task. DevOps makes the maintenance of most software easier, regardless of its architecture. Let's dive right into it.

2. Software architecture

00:16 - 00:40

Software architecture refers to how the roles and responsibilities of various components of software are defined, how they act, and their relationship with each other. As an analogy, think about the architecture of a building. An architect draws the structure of a building, defines its building blocks, and draws the doors, for example, so people can walk in.

1. ¹ https://en.wikipedia.org/wiki/Software_architecture

3. Application Programming Interfaces (APIs)

00:40 - 01:14

Similarly, software architecture is about defining the structure of software, the main functionality of its systems, their dependencies, and their relationships. Like an architect drawing a door, software architects define application programming interfaces or APIs, so the software can send and receive data. APIs are essential for any software system and architecture and are the gateways for moving data. After all, most software is all about how data flows between systems.

4. Various architectural systems

01:14 - 01:52

There are various architectural systems, all specialized for different use cases. When deciding on the right architectural design, some factors should be considered. First, we must decide what type of software we want to build. Software that runs on a computer differs from an interactive web page that runs on a browser. The second most important factor is the complexity of the software we want to build. Simple software can be handled as a single unit. But as the software gets more complex, it becomes necessary to separate various functionalities from each other.

5. Microservices

01:52 - 02:36

Microservices architecture is arguably the most important architectural system for modern organizations. Organizations love adding new features to their online products. Adding new features makes the software more complex. As the software gets increasingly complex, it becomes impossible to manage it as a single, gigantic unit. Instead, we separate parts of the

software into smaller independent units. These individual units are called microservices. A microservices architecture comprises many smaller software applications separated from each other. Although these units are separate from each other, they can communicate with each other via APIs.

6. Rental example 1

02:36 - 02:52

Let's look at a simple microservices architecture to see it in use. We are running a hotel and rental car booking mobile app. Here, our customer searches for available hotels and rental cars.

7. Rental example 2

02:52 - 03:09

Once the customer searches on the app and presses the submit button, the app on their phone send a message to the backend. The backend is where we host all of our microservices. The app on customers phone needs to learn the available hotels and rental cars from the backend.

8. Rental example 3

03:09 - 03:26

In our backend, we have a separate microservice for each functionality. Microservice hotels return the available hotels. At the same time, the microservice rental cars return the available rental car information. The customer is happy and satisfied.

9. DevOps for microservices architecture

03:26 - 03:51

Microservices architecture is beneficial because it is possible to maintain each microservice independently. In this example, on the left side, both of the microservices are at version one. Microservice hotel goes through the DevOps cycle; it's improved and deployed as version two. In contrast, the microservice rental cars is still at version one.

10. Benefits of DevOps for microservices

03:51 - 04:14

The software is constantly improved and changed countless times daily in large organizations. The microservices architecture allows you to change them one at a time, which makes the changes more secure and flexible. So if anything goes bad with changing the hotels microservice, the rest of the backend will still function well.

11. Let's practice!

04:14 - 04:19

Now let's go and practice what we have learned.