Variations

We'll look at a few quick variations of what we've learned in this chapter now.

We'll cover the following

- Frontend integration
- Asynchronous communication
- Combination
- Future chapters

Frontend integration

Frontend integration (chapter 3 (https://www.educative.io/collection/page/10370001/5441945024331776/6644855302258688)) can be a good addition to synchronous communication.

Asynchronous communication

Asynchronous communication (see chapter 6 (https://www.educative.io/collection/page/10370001/5441945024331776/4516786718375936)) is another alternative.

Synchronous and asynchronous communication are possibilities for microservices to communicate with each other on the logic level.

One of these options should be enough to build a microservices system.

Combination

Of course, a **combination** of synchronous and asynchronous is also possible.

The asynchronous communication with Atom and the synchronous communication with REST use the same infrastructure so that these two communication mechanisms can be used together very easily.

Future chapters

The following chapters show concrete implementations for synchronous communication.

The examples all use REST for communication. Today REST is the preferred architecture for synchronous communication.

In principle, other approaches such as SOAP (https://www.w3.org/TR/soap/) or Thrift (https://thrift.apache.org/) are also conceivable.

QUIZ

Q In what situation would a combination of asynchronous and synchronous communication be a good idea?

□ **A)** When at least the underlyung infrastructure supports both communication methods.

□ **B)** When some microservices are suited to asynchronous communication and others to synchronous.

□ **C)** It is always a good idea.

Submit Answer

Reset Quiz

We'll conclude this chapter with the next lesson.

← Back

Benefits & Challenges

Next →

Introduction

☑ Mark as Completed

Report an Issue

⚙ 📋