Variations

The ideas for migration can easily combine with many other approaches. Let's study a few.

WE'LL COVER THE FOLLOWING

Combining approaches to migration

- The ideas concerning typical migration strategies from the lesson on typical migration strategies fit very well to the concept of **self-contained systems (SCS)**. The migration can therefore simply separate a part of the legacy system into an SCS.

- Rules for authentication or communication between microservices as well as between microservices and the legacy system can be the starting point of a **macro architecture** (see chapter 3). A domain macro architecture is very useful, which can also include the legacy system in addition to microservices.

- **Frontend integration** can make sense for the integration between the legacy system and microservices.

- **Asynchronous microservices** fit very well with migration because they allow for a loose coupling. Especially for a migration, it can be sensible to continue to use an existing messaging technology for asynchronous communication to minimize the effort.

- **Synchronous microservices** should be used cautiously because this creates a tight coupling and resilience becomes difficult.

- **Kubernetes, PaaS, or Docker** are also interesting in a migration scenario. However, they represent a *new environment* that needs to be

operated. It may, therefore, make sense to use a classical deployment and operation environment at least at the beginning to reduce the initial migration effort. In the long term, however, such environments have many advantages. In addition, of course, the old system can be operated in such an environment.

Experiments

The migration strategy must match the respective scenario. The following questions are important in order to design your own strategy.

- What are the **goals of the migration** to microservices?
  - Which are especially important?
  - What impact does this have on the migration strategy?

In principle, **migration should take place gradually**. The selection of the parts to be migrated to microservices can be made according to technical or domain criteria. However, domain criteria are better suited, at least in the long term.

The **following approach is suitable for a migration based on domain criteria**:

- Split the system into bounded contexts.
- Which of the bounded contexts will you migrate first? Why? Reasons can be the simple migration of the bounded context or many planned changes in the bounded context. Consider the different scenarios.

In the next lesson, we'll look at a quick chapter conclusion.