SCSs and Microservices

In this lesson, we'll study the relationship between SCSs and microservices.

We'll cover the following

- One SCS can be split into several microservices
- Summary

SCSs stand out from deployment monoliths in the same way as microservices. A deployment monolith would implement the entire application in a single deployable artifact while **SCSs divide the system into several independent web applications**.

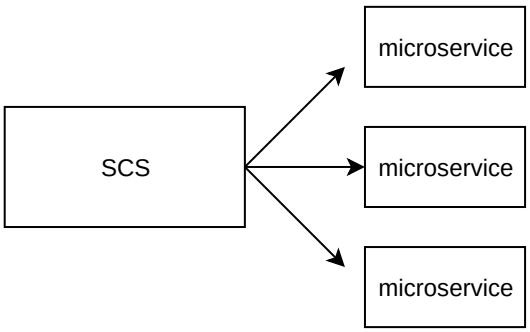
Because an SCS is a separate web application, it can be deployed independently of the other SCSs. SCSs are also modules of an overall system. Therefore, **SCSs are independently deployable modules** and consequently correspond to the definition of microservices.

One SCS can be split into several microservices

However, an SCS may be split into several microservices. If the payment in the *check out* SCS of an e-commerce system causes a particularly high load, then it can be implemented as a separate microservice which can be **scaled separately** from the rest of the *check out* SCS. So, the *check out* SCS consists of a microservice for payment and contains the rest of the

functionality in another microservice.

Another reason for separating a microservice from an SCS is the **security advantage offered by greater isolation**. Additionally, domain services that calculate the price of a product or quotation for all SCSs can be a useful shared microservice. The microservice should be assigned to a team like an SCS in order to avoid too much coordination.



An SCS can be split into microservices

Summary

In summary, microservices and SCSs differ in the following ways:

- Typically, **microservices are smaller than SCSs**. An SCS can be so large that an entire team is busy working on it. Microservices can potentially have only a few hundred lines of code.
- **SCSs focus on loose coupling**. There is no such rule for microservices, although tightly coupled microservices have many disadvantages and therefore should be avoided.
- In any case, **an SCS must have a UI**. Many microservices offer only a technical interface for other microservices but no user interface.
- **SCSs recommend UI integration or asynchronous communication**; synchronous communication is allowed but not recommended. Large microservices systems such as Netflix also

focus on *synchronous communication*.

QUIZ

1 If an SCS can be separated into several microservices, it is still a good idea for the original team to work on all the microservices.

A) True

B) False

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

In the next lesson, we'll look at some challenges pertaining to using SCSs.

← Back

Next →

An Example

Challenges

Mark as Completed

Report an Issue