Module 1: What is a Service? What is a Microservice?

- General concept: What is a (customer) service?
  - What is a service / an asset / service management?
  - What a service provider?

- General concept: What is a microservice?
  - What is a microservice? - Explanation of term

What is a service?

A service provided by a service provider is means of

- **delivering value to customers**
- **by facilitating outcomes customers want to achieve (→ customer needs)**
- **without the ownership of specific costs and risks.**

The scope of service ranges from core service, general support to complex service package.

A service that is straight consumed by the end user to do their work and is something they demand and recognize.
(→ pay for service → measure it → SLA)

Examples:

- **Dropbox : online storage service**

What is an asset?

A asset is any resource or capability that could contribute to the delivery of a service.

Examples of assets are:

- **infrastructure, e.g. workstations, router, rooms, locker, …**
- **management + organization,**
- **processes,**
- **knowledge + information,**
- **people, employees ("human resources ")**
- **applications,**
- **financial capital**
- **….**

What is service management?

The set of all technical and organizational capabilities for providing value to customers in the form of services.

These abilities are available by functions and processes that the services throughout their life cycle.

Examples of technical and organizational capabilities:

- **Planning of organizational processes**
- **Customer support by a hotline**
- **Specification and planning of services**

What is a service provider? What is a role?

A service provider is

- **an organizational unit supplying services**
- **to one or more internal customers or external customers.**

We can distinguish three different types of service provider:

- **Type I - Internal Service Provider.**
  A service provider embedded within a business unit. It provides IT services exclusively for this specific business unit.
- **Type II - Shared Services Unit.**
  An internal service provider that provides shared IT services to more than one business unit.
- **Type III – External Service Provider.**
  A service provider that provides IT services to external customers

- **Accordingly, services can be classified into type I (internal service), type II (internal shared service), type III (external service).**

What is a microservice? - Explanation of term

Definition:
A microservice is a small, autonomous service working together with others.

small:

- **Responsible for only one task. “Doing one thing well”.**
- **How small should it be? (Simplicity vs. complex network of small parts)**

autonomous:

- **change independently from others**
- **hidden internal structure (only exposed by APIs)**

working together:

- **by open (technology agnostic) APIs**

Characteristics of Microservices

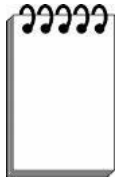
- **A network of multiple components**
- **Built For Business**
  SOA: partitioned along technical layers.
  Service-based architectures: Organized Around Business Capabilities (Domain Driven Design)
- **Life cycle management and DevOps**
  A team will support the service throughout its lifecycle.
- **Simple Routing:**
  Smart REST-APIs and dumb pipes through which the info flows.
- **Decentralized**
- **Failure Resistant:**
  The neighbour service will provide the service.
- **Evolutionary**
  the service architecture is in a constant state of flux. It is dynamic rather than static.
  the development is never finished.

from/compare: What is Microservices Architecture? - https://smartbear.com/learn/api-design/what-are-microservices/

Benefits of Microservices (by Smartbear)

- Microservice architecture gives developers the freedom to independently develop and deploy services
- A microservice can be developed by a fairly small team
- Code for different services can be written in different languages (though many practitioners discourage it)
- Easy integration and automatic deployment (using open-source continuous integration tools such as Jenkins, Hudson, etc.)
- Easy to understand and modify for developers, thus can help a new team member become productive quickly
- The developers can make use of the latest technologies
- The code is organized around business capabilities
- Starts the web container more quickly, so the deployment is also faster
- When change is required in a certain part of the application, only the related service can be modified and redeployed—no need to modify and redeploy the entire application
- Better fault isolation: if one microservice fails, the other will continue to work (although one problematic area of a monolith application can jeopardize the entire system)
- Easy to scale and integrate with third-party services
- No long-term commitment to technology stack

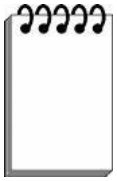
from:: What is Microservices Architecture? - https://smartbear.com/learn/api-design/what-are-microservices/



Drawbacks of Microservices (by Smartbear)

- Due to distributed deployment, testing can become complicated and tedious
- Increasing number of services can result in information barriers
- The architecture brings additional complexity as the developers have to mitigate fault tolerance, network latency, and deal with a variety of message formats as well as load balancing
- Being a distributed system, it can result in duplication of effort
- When number of services increases, integration and managing whole products can become complicated
- In addition to several complexities of monolithic architecture, the developers have to deal with the additional complexity of a distributed system
- Developers have to put additional effort into implementing the mechanism of communication between the services
- Handling use cases that span more than one service without using distributed transactions is not only tough but also requires communication and cooperation between different teams
- The architecture usually results in increased memory consumption
- Partitioning the application into microservices is very much an art

from:: What is Microservices Architecture? - https://smartbear.com/learn/api-design/what-are-microservices/



References

- Sam Newman: Building Microservices - Designing Fine-Grained Systems, O'Reilly Media, 2015.
- Smartbear, What is Microservices Architecture?, https://smartbear.com/learn/api-design/what-are-microservices/ (last accessed: 07.10.2018)