

DOZIERENDER: MAX MUSTERMANN

DATA ENGINEERING

Datensystem-Grundlagen

1

Datenverarbeitung „at Scale“

2

Microservices

3

Governance und Sicherheit

4

Verbreitete Cloud-Plattformen und -Dienste

5

Data Ops

6

LEKTION 3

MICROSERVICES



- erklären, was **monolithische Architekturen** und ihre Nachteile sind
- diskutieren, was **Microservice-Architekturen** sind, wie sie **funktionieren** und warum sie eine Alternative zu klassischen monolithischen Architekturen darstellen
- Microservice-Architekturen **implementieren**
- existierende **monolithische Architekturen** in eine Microservice-Architektur **migrieren**



1. Beschreiben Sie die **Hauptunterschiede** einer sogenannten **monolithischen** und einer **Microservice-Architektur**.
2. Erläutern Sie die **Hauptvorteile** eine **Microservice-Architektur** gegenüber einer monolithischen Architektur
3. Erklären Sie **drei verschiedene Strategien**, um von einer monolithischen zu einer Microservice-Architektur zu **migrieren**.

Einführung in monolithische Architekturen

Einführung in Microservices

- Allgemeine Einführung
- Der Scale-Würfel
- Vorteile & Nachteile von Microservices

Implementierung von Microservices

- Client-Microservices-Kommunikation
- Service Discovery
- Kommunikation zwischen Microservices
- Verteiltes Datenmanagement
- Strategien zur Bereitstellung von Microservices
- Migration von monolithischen zu Microservice-Architekturen

Einführung in monolithische Architekturen

Einführung in Microservices

- Allgemeine Einführung
- Der Scale-Würfel
- Vorteile & Nachteile von Microservices

Implementierung von Microservices

- Client-Microservices-Kommunikation
- Service Discovery
- Kommunikation zwischen Microservices
- Verteiltes Datenmanagement
- Strategien zur Bereitstellung von Microservices
- Migration von monolithischen zu Microservice-Architekturen

DEFINITION MONOLITH

1. ein großer, einzelner, vertikaler **Steinblock**
 2. eine **einzigste, sehr große Organisation**, die sich nur **sehr langsam verändert**
- **vollständige Funktionalität** in **einer einzigen Anwendung/Datei** (.JAR, .EXE, ...)
 - meist in **einer Programmiersprache** gehalten
 - **Schnittstellen** zu **externen Diensten** über APIs

Einführung in monolithische Architekturen

Einführung in Microservices

- Allgemeine Einführung
- Der Scale-Würfel
- Vorteile & Nachteile von Microservices

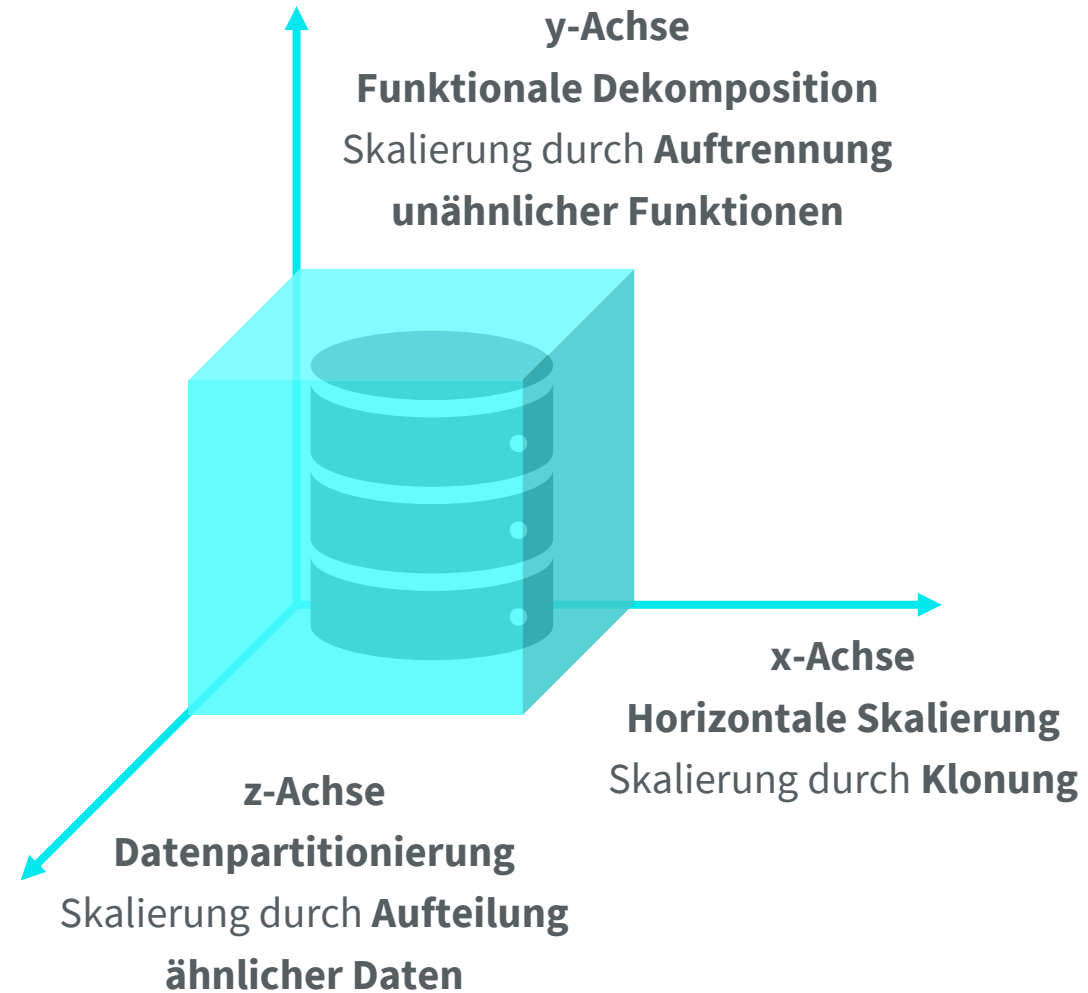
Implementierung von Microservices

- Client-Microservices-Kommunikation
- Service Discovery
- Kommunikation zwischen Microservices
- Verteiltes Datenmanagement
- Strategien zur Bereitstellung von Microservices
- Migration von monolithischen zu Microservice-Architekturen

- sind **lose gekoppelte** Dienste
- lassen sich **unabhängig** voneinander **bereitstellen**
- sind **spezifisch** in ihrer **Funktionalität**
- werden in der Regel von **kleinen Teams entwickelt** und **gewartet**
- sind hoch **skalierbar** (Scalability), **zuverlässig** (Reliability) und **wartbar** (Maintainability)
- **Standard moderner Architekturen**

DER SCALE-WÜRFEL

Abb. 1: Scale-Würfel



- **klein** und **überschaubar**
- leichter **wartbar**, **erweiterbar** und besser **skalierbar**
- **isolierte** Bereitstellung
- erhöhte **Systemzuverlässigkeit** (Reliability)
- **kürzere** Entwicklungszyklen

NACHTEILE VON MICROSERVICES

- **Verwaltung** und **Kommunikation komplexer**
- **verteilte Datenbanken** stellen Herausforderungen dar
- **Service Discovery**

Einführung in monolithische Architekturen

Einführung in Microservices

- Allgemeine Einführung
- Der Scale-Würfel
- Vorteile & Nachteile von Microservices

Implementierung von Microservices

- Client-Microservices-Kommunikation
- Service Discovery
- Kommunikation zwischen Microservices
- Verteiltes Datenmanagement
- Strategien zur Bereitstellung von Microservices
- Migration von monolithischen zu Microservice-Architekturen

- monolithische Architektur: **Kommunikation** zwischen Client und Dienst **einfach**
- **Microservice-Architektur**: Kommunikation nicht mit einem, sondern **mehreren Diensten**
- Lösung: **API-Gateway**

- Dienste **starten dynamisch** und **schalten sich ab**
 - **IP-Adressen** mit erreichbaren **Ports** der einzelnen Dienste **ändern sich**
 - Lösung: **Service Registry**
-
- Service Discovery
 - Client-seitig
 - Server-seitig

- **synchrone** Kommunikation
 - Request/Response
- **asynchrone** Kommunikation
 - Request/Asynchronous Response
 - Notification
 - Pub/Sub
- **domänenspezifische Protokolle**

- eventgetriebe Architektur
 - **Aktionen & Datenflüsse** werden bspw. über **Pub/Sub-Systeme** ausgelöst
 - Eventual Consistency
- Event sourcing
 - Herausforderung: **Atomicity**
 - nicht Status, sondern **Statusänderungen** werden **persistiert**

- **multiple** Dienste pro Host
 - mehrere Dienste in einer Maschine
 - replizierte Maschinen
- **einzelner** Dienst pro Host
 - ein Dienst pro Maschine
 - VM/Container/Serverless

- **neue Funktionalität** als Microservices
- **Aufsplittung** nach:
 - Präsentations-Schicht
 - Geschäfts-Schicht
 - Daten-Schicht
- Ausgliederung von **Modulen** in Microservices



- erklären, was **monolithische Architekturen** und ihre Nachteile sind
- diskutieren, was **Microservice-Architekturen** sind, wie sie **funktionieren** und warum sie eine Alternative zu klassischen monolithischen Architekturen darstellen
- Microservice-Architekturen **implementieren**
- existierende **monolithische Architekturen** in eine Microservice-Architektur **migrieren**

EINHEIT 3

TRANSFERAUFGABE

TRANSFERAUFGABE

Ein Start-Up, das **nachhaltige Produkte in kleineren Geschäften** vertreibt, war in den letzten Jahren sehr erfolgreich. In Folge sollen **weltweit weitere Filialen** eröffnet werden. Als Data Engineer:in sind Sie damit beauftragt, das **Datensystem zu entwerfen**, welches Daten über die **angebotenen Produkte und deren Zulieferer** speichert und verarbeitet.

Das System soll mehrere kleinere Funktionalitäten bieten: Abfrage des Warenbestandes pro Filiale, automatisierte Aktualisierung des Datenbestandes auf Basis der an den Kassen abgerechneten Artikel, Bestellung neuer Ware bei unterschiedlichen Zulieferern, etc.

Sammeln Sie im Team **Argumente** dafür, dass das Datensystem als **Microservice-Architektur** implementiert werden soll. Geben Sie dabei **für jedes Argument** ein **passendes Beispiel** für diesen konkreten Anwendungsfall.

Bitte stellen Sie Ihre
Ergebnisse vor.
Im Plenum werden
die Ergebnisse
diskutiert.





1. Welcher der folgenden Aspekte ist ein Nachteil einer Microservice-Anwendung?
 - a) Überlastung der DIE
 - b) Verwaltung von verteilten Datenbanken
 - c) Anpassung an neue Frameworks
 - d) Skalierbarkeit



2. Welche der folgenden Aufgaben übernimmt das API-Gateway in einer Microservice-Anwendung nicht?
- a) mehrere Zugangspunkte zur Anwendung
 - b) ein einziger Zugangspunkt zur Anwendung
 - c) Protokollübersetzer zwischen Microservices und den Clients
 - d) Anfrage-Router



3. Welcher Aspekt einer monolithischen Anwendung macht ihre Bereitstellung einfach?
- a) Es muss nur eine Datei (JAR oder EXE) bereitgestellt werden.
 - b) Die Größe der Anwendung ist klein.
 - c) Die Bereitstellung von Modulen ist einfacher als die Bereitstellung von Microservices.
 - d) Es ist für jedes Teammitglied einfach, den gesamten Code zu verstehen.

QUELLENVERZEICHNIS

Abbott, M. & Fisher, M. (2015). *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise* (2. Aufl.). Addison-Wesley Professional. Oxford University Press. (o. D.). monolith. In *Oxford Advanced Learner's Dictionary*. Abgerufen am 2. März 2022, von <https://www.oxfordlearnersdictionaries.com/us/definition/english/monolith>

© 2022 IU Internationale Hochschule GmbH

Diese Inhalte sind urheberrechtlich geschützt. Alle Rechte vorbehalten.

Diese Inhalte dürfen in jeglicher Form ohne vorherige schriftliche Genehmigung der IU Internationale Hochschule GmbH nicht reproduziert und/oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.