

Services

Microservices

Msc. Informatik

13. April 2021

Stand der Dinge

Was bisher geschah:

- Einführung in die Thematik
- Vorstellung der Projektanforderungen

Was geschehen wird:

- Verschiedene Servicetypen
- Aufteilung der Business-Logik

Service-Typen

Verschiedene Arten von Services

- Stateless-Services
- Stateful-Services
- Hybrid-Services

Stateless-Services

Was heißt 'zustandslos' bei Services?

- Daten/Werte werden übergeben und vom Service berechnet
- Ein Request -> Ein Response
- Informationen liegen vollständig vor und Aktionen sind für den Client atomar
- Kein Zustand im Service

Stateless-Services II

Statelessness ist erstrebenswert:

- Kein Zustand muss hergestellt werden
- Services können sofort Arbeit übernehmen
- Benötigen i.d.R. keinen persistenten Speicher
- Sind einfach zu verwalten

Stateful-Services

Das Gegenteil der Stateless-Services

- Arbeiten über einen Zustandsautomaten
- Persistieren Zustand
- Verlust eines Knoten ist kritisch
- Skalierung ist träge

Stateful-Services - Beispiele

Datenbankmanagementsysteme

- Speichern große Mengen von Daten
- DB Constraints können Aktionen verhindern
- Foreign-Keys müssen beachtet werden
- Nebenläufigkeit birgt Schwierigkeiten
- Transaktionen

Weitere Beispiele:

- Spielebackend
- Cache
- IRC

Aufteilung der Domänenlogik

Eine große Problematik bei Microservice-Architekturen liegt in der Aufteilung in einzelne Services.

Die Aufteilung hängt von diversen Aspekten ab:

- Geschäftslogik
- Unternehmensstruktur
- Wahl der Patterns
- Komplexität der Use-Cases

Conway's Law

organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.

– M. Conway

Daraus ergeben sich Bedenken bei dem Entwurf:

- Wie sind die Geschäftsprozesse organisiert?
- Wie ist das Unternehmen gegliedert?
- Welche Verständnisse haben die betroffenen Bereiche?

Scope

Richtwerte für den Scope eines Microservices:

- Implementierung innerhalb von 72 Stunden
- Involviert in mindestens einem Use-Case

Dazu gehören meistens folgende Komponenten:

- Bereitstellung einer API
- Datenhaltung (Datenbank / In-Memory)
- Anbindung an ein Event-System

Das Microservice-Ecosystem

Neben den Microservices existieren in dem Ecosystem andere Komponenten:

- Nachrichten-Bus für asynchronen Austausch
- API-Gateway
- Logging
- Monitoring
- ...

Wir beschäftigen uns zunächst nur mit Domänenaspekten.

Domain-Driven Design

Wir bedienen uns einiger Bausteine aus diesem Entwurfansatz:

- Ubiquitous Language
- Bounded Context
- Aggregate
 - ▶ Zusammenschluss aus mehreren Objekten
 - ▶ Gebündelt durch ein *root entity*
- Domain Event
 - ▶ *Fachliches Ereigniss*
 - ▶ Beschreiben Änderungen am Domänenobjekt
 - ▶ Event-Storming als Möglichkeit der Identifizierung