

1 Microservices - Interaktion und Circuit-Breaker Pattern

1.1 Topics

Bei einer Microservice Architektur sind die einzelnen Komponenten des Systems als eigenständige Services implementiert. Um die Anfragen zu bearbeiten bzw. Geschäftsprozesse umzusetzen, ist eine Interaktion der Services über das Netzwerk notwendig. Man spricht in diesem Zusammenhang von einem *verteilten System*.

Ein Problem, dass sich daraus ergibt ist, dass bei Ausfall oder Leistungsengpässen eines Services die Requests der anderen Services fehlschlagen oder verzögert werden. Dadurch ergibt sich eine Art Schneeballeffekt, sodass ein Microservice nach dem anderen ausfallen kann. Um diesem Verhalten vorzubeugen, wird das sog. *Circuit Breaker Pattern* angewandt.

1.2 Reading Tasks

- [Microservices - Tradeoffs - Distribution](#)
- [Circuit breaker pattern in Microservices](#)
- [Circuit breaker pattern in Node.js](#)

2 Questions

2.1 Microservices - Distribution

Als Nachteil einer Microservice Architektur wird Verteiltheit (*distribution*) genannt. Diskutieren Sie, welche *neuen Probleme* man sich in dieser Hinsicht stellen muss, wenn man von einer monolithischen Architektur auf eine Microservice Architektur migriert.

2.2 Circuit Breaker Pattern

Beschreiben Sie in eigenen Worten das Circuit Breaker Pattern. Woher kommt der Name dieses Patterns?

Welche Zustände kennt der Circuit Breaker und unter welchen Bedingungen kommt es zu den Übergängen zwischen den Zuständen?