

1 Kong - Microservices & API Gateway

1.1 Topics

Durch das Aufsplitten einer monolithischen Architektur in Microservices ergeben sich weitere Herausforderungen an das Handling der Requests. Statt einem reinen Load Balancing müssen die einzelnen Requests auch an die jeweiligen Microservices weitergeleitet werden, sodass die interne Architektur aus der Sicht der Nutzer der API verborgen bleibt. Auch ergibt sich durch die Einführung von Microservices die Anforderung, dass bestimmte Dienste wie Logging, Rate Limiting etc. an zentraler Stelle erfolgen. Die dafür zuständige Komponente wird *API Gateway* genannt.

1.2 Tutorial

- [Install Kong using Docker with a database](#)
- [Services and Routes](#)
- [Rate Limiting](#)
- [Key Authentication](#)

1.3 Reading Tasks

- [Microservices](#)
- [Microservice Trade-offs](#)
- [What is an API Gateway](#)
- [API Gateway vs. Load Balancer](#)
- [Kong Gateway Tutorial - Abschnitt Why Use Kong Gateway?](#)

2 Questions

2.1 Microservice Architektur vs. Monolith

Diskutieren Sie die Unterschiede einer Microservice Architektur und einem Monolithen. Welche Auswirkung haben die beiden Ansätze auf die Skalierbarkeit und die Software-Entwicklung?

2.2 Conway's Law

Was versteht man unter *Conway's Law*? Welche Auswirkung hat dies auf die Entwicklung von Softwaresystemen im speziellen auf Microservices?

2.3 Microservices - Vor- und Nachteile

Nennen Sie jeweils drei Vor- und Nachteile einer Microservice Architektur und diskutieren Sie diese anhand des im Unterricht verwendeten Applikation.

2.4 Microservices - Technology Diversity

Als Vorteil einer Microservice Architektur wird *technology diversity* genannt. Was bedeutet dies und wie könnte diese im vorliegenden Beispiel zur Anwendung gelangen?