

Benjamin Swarovsky

Technologische Ansätze zur
Umsetzung einer
Microservice-Architektur






Prototypische Implementierung einer
Anwendung zur Verwaltung der IT-
Kontaktmesse an der Fachhochschule Erfurt

Gliederung

- Einleitung
- Architektur
- API Gateway
- Service Discovery
- Load Balancer
- Zusammenspiel der Technologien
- Auswertung
- Demonstration des Prototypen

- **Einleitung**
 - **Problemstellung**
 - **Ziele**
- Architektur
- API Gateway
- Service Discovery
- Load Balancer
- Zusammenspiel der Technologien
- Auswertung
- Demonstration des Prototypen

Problemstellung

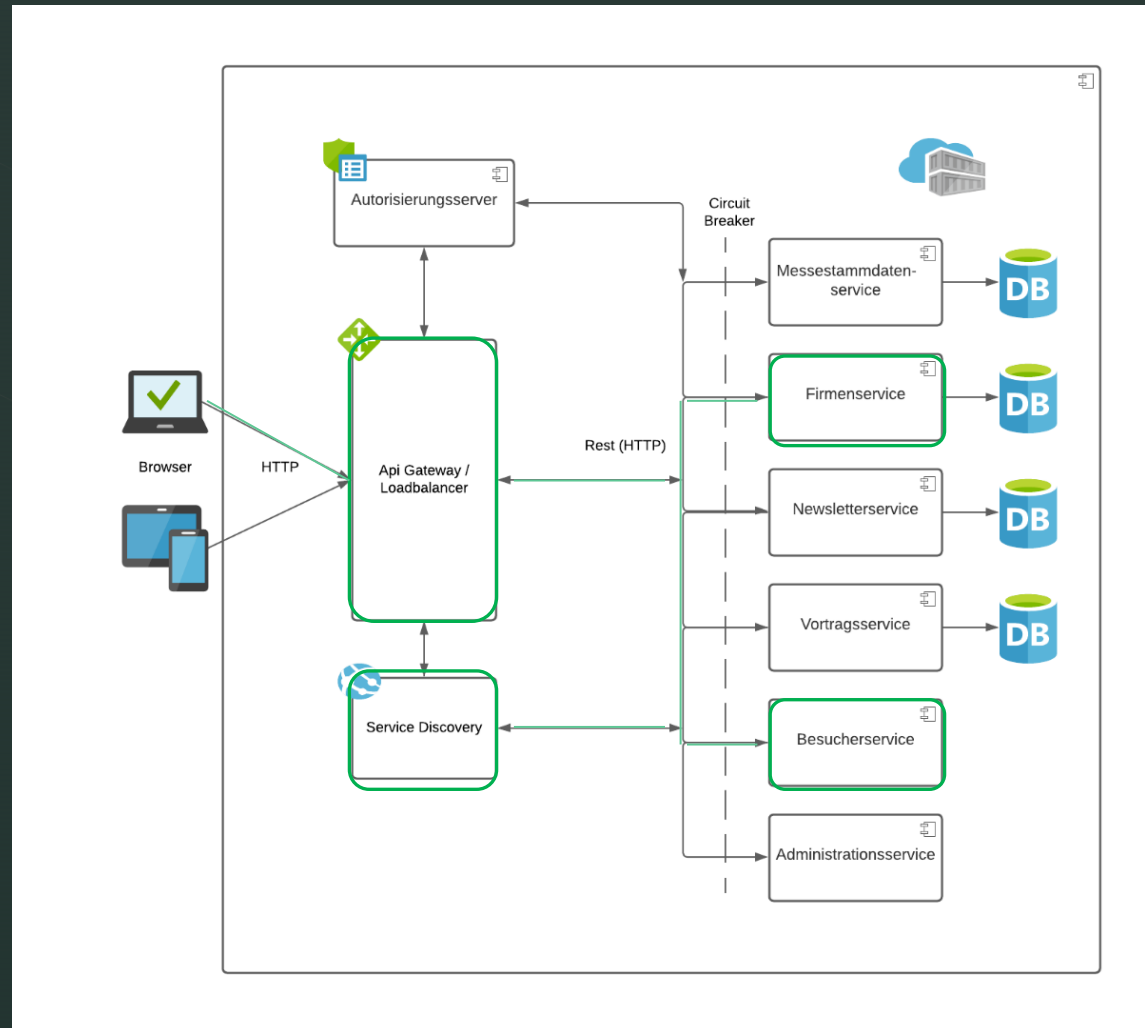
- Monolith:
 - Enge Kopplung 
 - Schlecht skalierbar 
 - Big Ball of Mudd 
- Microservices 
- Anspruchsvolle Architektur 
- Herausforderungen beim Einsatz mehrerer Services:
 - Konsistenz
 - Kommunikation
 - Fehlerbehandlung

Ziele

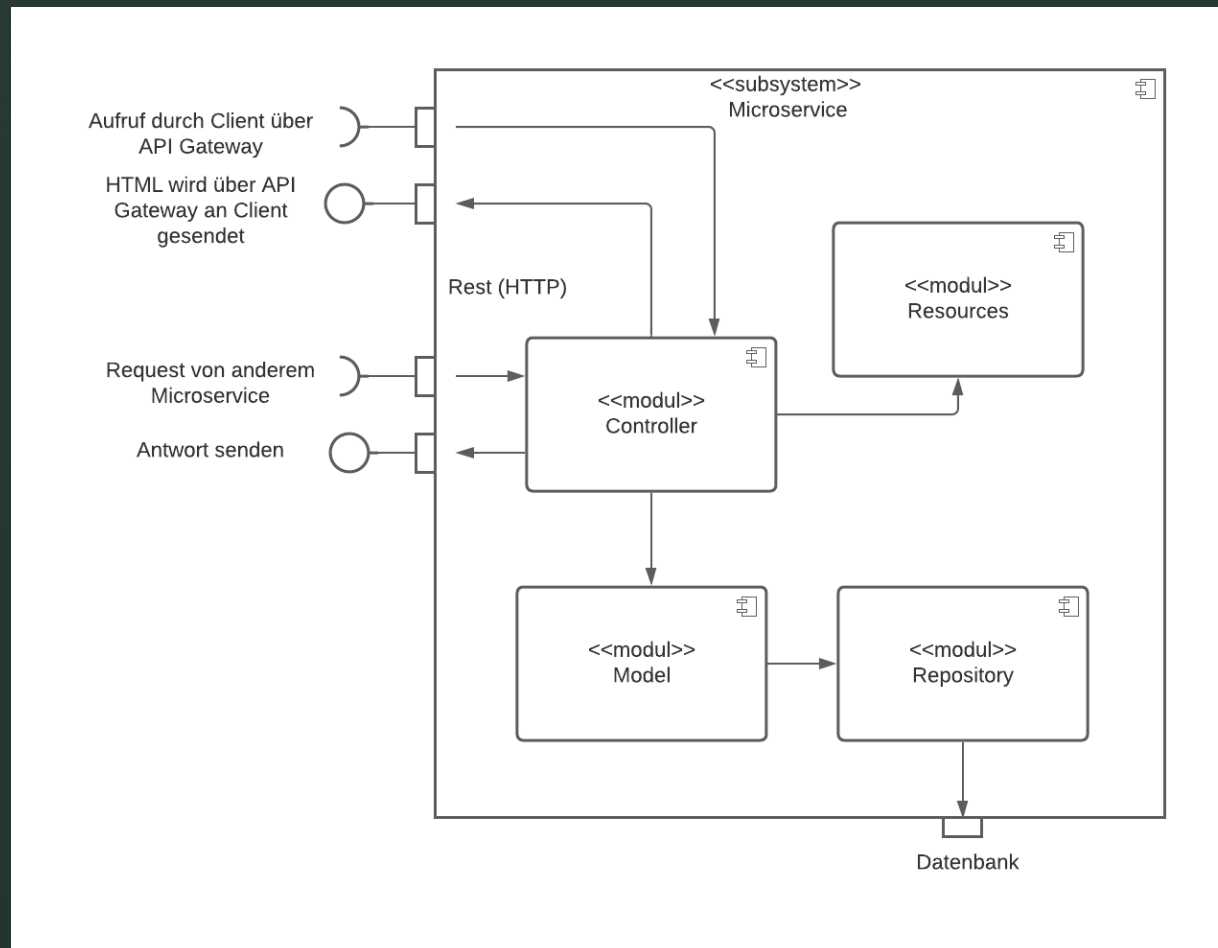
- Technologische Ansätze zur Umsetzung von Microservices (Frameworks, Bibliotheken, Algorithmen, ...)
- Beispielanwendung (Verwaltungsprogramm für die IT-Kontakmesse)
- Implementierung der Technologischen Ansätze (Springboot, Eureka, Jaeger, ...)
- Auswertung
 - Komplexität
 - Welche Probleme traten auf
 - Wie wichtig sind einzelne Technologien für die Umsetzung von Microservices

- Einleitung
- **Architektur**
 - **Bausteinsicht Ebene 1**
 - **Bausteinsicht Ebene 2**
- API Gateway
- Service Discovery
- Load Balancer
- Zusammenspiel der Technologien
- Auswertung
- Demonstration des Prototypen

Bausteinsicht Ebene 1



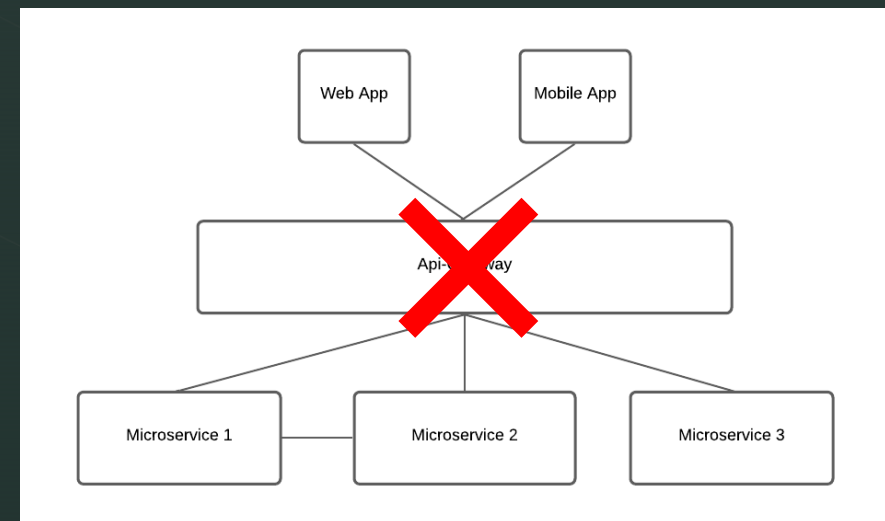
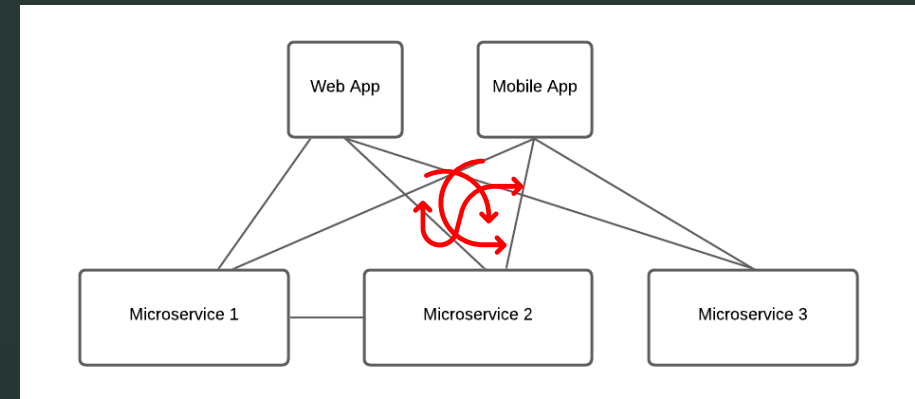
Bausteinsicht Ebene 2



- Einleitung
- Architektur
- **API Gateway / Load Balancer**
 - **API Gateway**
 - **Load Balancer**
 - **Spring Cloud API Gateway**
 - **Implementierung**
- Service Discovery
- Zusammenspiel der Technologien
- Auswertung
- Demonstration des Prototypen

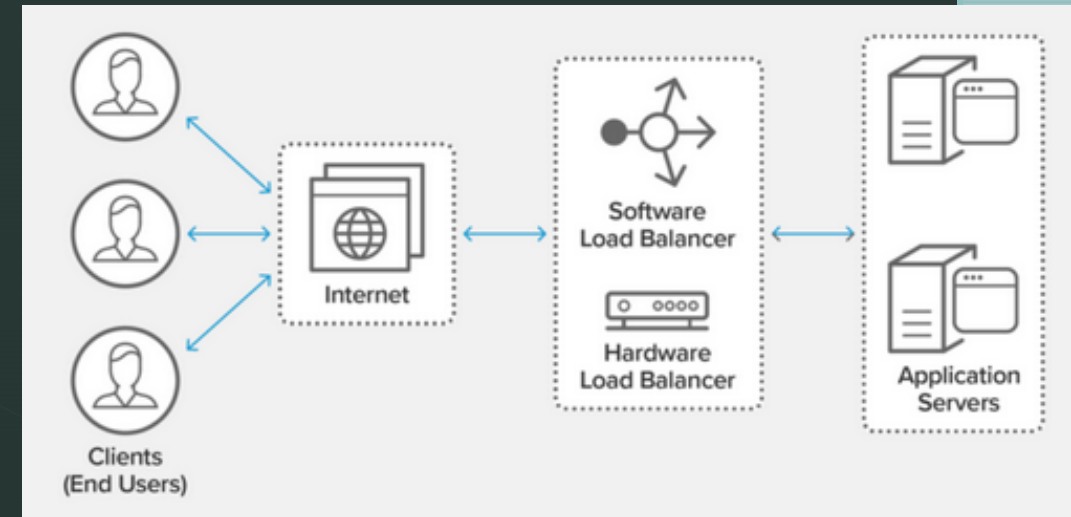
API Gateway - Grundlagen

- Direkte Kommunikation (Client - Services)
 - Sicherheitsprobleme
 - Enge Kopplung
- Kommunikation über Gateway
 - Kontaktpunkt für Ein/-Ausgehenden Netzverkehr
 - Autorisierung & Authentifizierung
 - Zentrales Logging
 - Nachteil: Ausfall des Gateways zieht den Ausfall des gesamten Systems nach sich



Load Balancer

- Setzt Lastverteilung in einem Netzwerk um
- Ermittelt welche Rechenressource die Clientanforderung erfüllen kann
- Realisiert als Software- oder Hardware-Load Balancer
- Nutzt Algorithmen wie zum Beispiel: Round Robin



Spring Cloud API Gateway

- Basiert auf asynchronen eventgetriebenen Framework Netty
- Features:
 - Ribbon Load Balancer
 - Sicherheitskonfigurationsmöglichkeiten mit Spring Security
 - Einbindung eigener Filter
- Einfache Integration in Spring Anwendung
- Alternativen: Ocelot, KrakenD, Kong

API Gateway - Implementierung

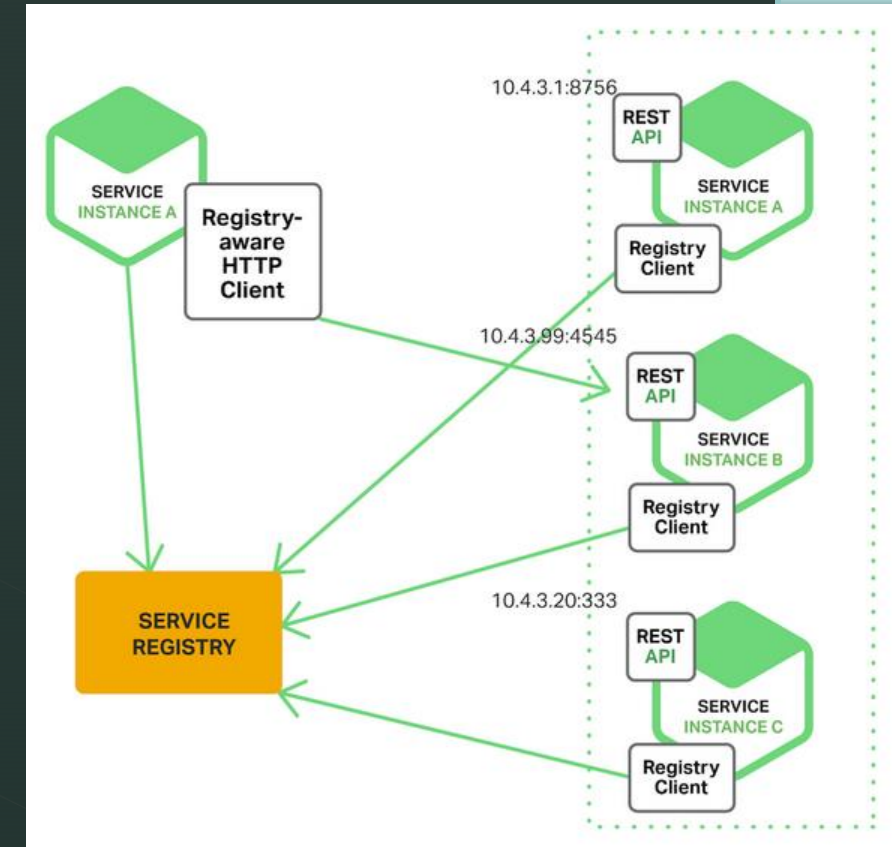
```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-gateway</artifactId>  
</dependency>
```

```
server.port=8081  
spring.application.name=ApiGateway  
  
spring.cloud.gateway.routes[2].id=lbFirmenservice    //id der Route  
  
spring.cloud.gateway.routes[2].uri=lb://firmenservice /* Name des aufzurufenden -  
                                           Services  
                                           (für Loadbalancing) */  
  
spring.cloud.gateway.routes[2].predicates[0]=Path=/** /* Pfad unter dem  
                                           Loadbalancing  
                                           Erfolgen soll */
```

- Einleitung
- Architektur
- API Gateway / Load Balancer
- **Service Discovery**
 - Grundlagen
 - Eureka
 - Implementierung
- Zusammenspiel der Technologien
- Auswertung
- Demonstration des Prototypen

Service Discovery - Grundlagen

- Services mit dynamisch zugewiesenen Netzwerkstandorten können sich gegenseitig finden
- Services registrieren sich an einer Registry
- Adressauflösung kann über Namen des Services erfolgen
- Unter anderem Clientseitige und Serverseitige Discovery



Eureka Discovery Service

- Clientseitige Service Discovery
- Alternativen: NGINX (serverseitig), Zookeeper (clientseitig), ...

Service Discovery Implementierung

- Einleitung
- Architektur
- API Gateway / Load Balancer
- **Service Discovery**
- **Zusammenspiel der Technologien**
- Auswertung
- Demonstration des Prototypen

- Einleitung
- Architektur
- API Gateway / Load Balancer
- **Service Discovery**
- Zusammenspiel der Technologien
- **Auswertung**
- Demonstration des Prototypen

Demonstration des Prototypen

Ende