Technologische Ansätze zur Umsetzung einer Microservice-Architektur

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

Betreuer: Prof. Dr. Steffen Avemarg, Dipl.-Inf. Steffen Späthe

Studiengang Angewandte Informatik, Altonaer Str. 25, 99085 Erfurt, Tel. 0361 6700 642, e-mail: informatik@fh-erfurt.de

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik

Benjamin Swarovsky

1990 Geboren in Erfurt

2007-2009 Andreas Gordon Schule Erfurt (Fachhochschulreife)

2010-2014 Elektroniker für Energie und Gebäudetechnik (Ausbildung)

2018-2021 Studium FH-Erfurt Angewandte Informatik (Bachelor)

Kurzfassung

Ein weit verbreitetes Software-Architekturmuster stellt heutzutage die Microservice-Architektur dar. Folgende wissenschaftliche Arbeit beschreibt Technologien wie zum Beispiel Frameworks und Bibliotheken, welche für die erfolgreiche Verwirklichung einer solchen Architektur eingesetzt werden können. Unter anderem werden Grundlagen wie Algorithmen, Protokolle und Entwurfsmuster genannt, auf denen diese Technologien basieren. Beispielhaft wird der Einsatz der Technologien anhand eines Softwareprototyps demonstriert. Dieser dient zur Umsetzung einer Anwendung, zur Verwaltung der IT-Kontaktmesse an der Fachhochschule Erfurt. Der Prototyp wird anhand einer Microservice-Architektur realisiert. Es wird dargestellt, wie diese Architektur auf Grundlagen der Anforderungen, welche an das Verwaltungssystem gestellt werden, angefertigt wird. Am Ende erfolgt eine Auswertung, inwieweit die vorgestellten Technologien implementiert werden konnten. Zusätzlich wird die Bedeutsamkeit dieser Technologien, für die erfolgreiche Verwirklichung einer vollständigen Anwendung, welche über den Funktionsumfang des Prototyps hinausgehen würde, beurteilt.

Autorisierungsserver

Der Einsatz eines Autorisierungsservers bietet unter Verwendung des OAuth2 Protokolls eine Lösung zur Umsetzung von Autorisierung in einem verteilten System. Dadurch kann eine Authentifizierung für die Verwendung mehrerer Microservices mit nur einem Login ermöglicht werden.

API Gateway

Ein API Gateway schafft dem gegenüber Lösungsmöglichkeiten. Dieses gleicht bezüglich seiner Funktionalitäten einem Reverse Proxy. Es stellt jeweils einen Kontaktpunkt für ein- und ausgehenden Netzwerkverkehr bereit. Es bietet dazu ein vereinheitlichtes Interface, welches mit dem Client interagiert. Dementsprechend werden Gruppen interner Microservices unter einer einzigen URL bereitgestellt. Eine einzelne Clientanfrage kann mehrere Microservices aggregieren. Dadurch kann der Datenaustausch zwischen Backend und Client reduziert werden.

Load Balancer

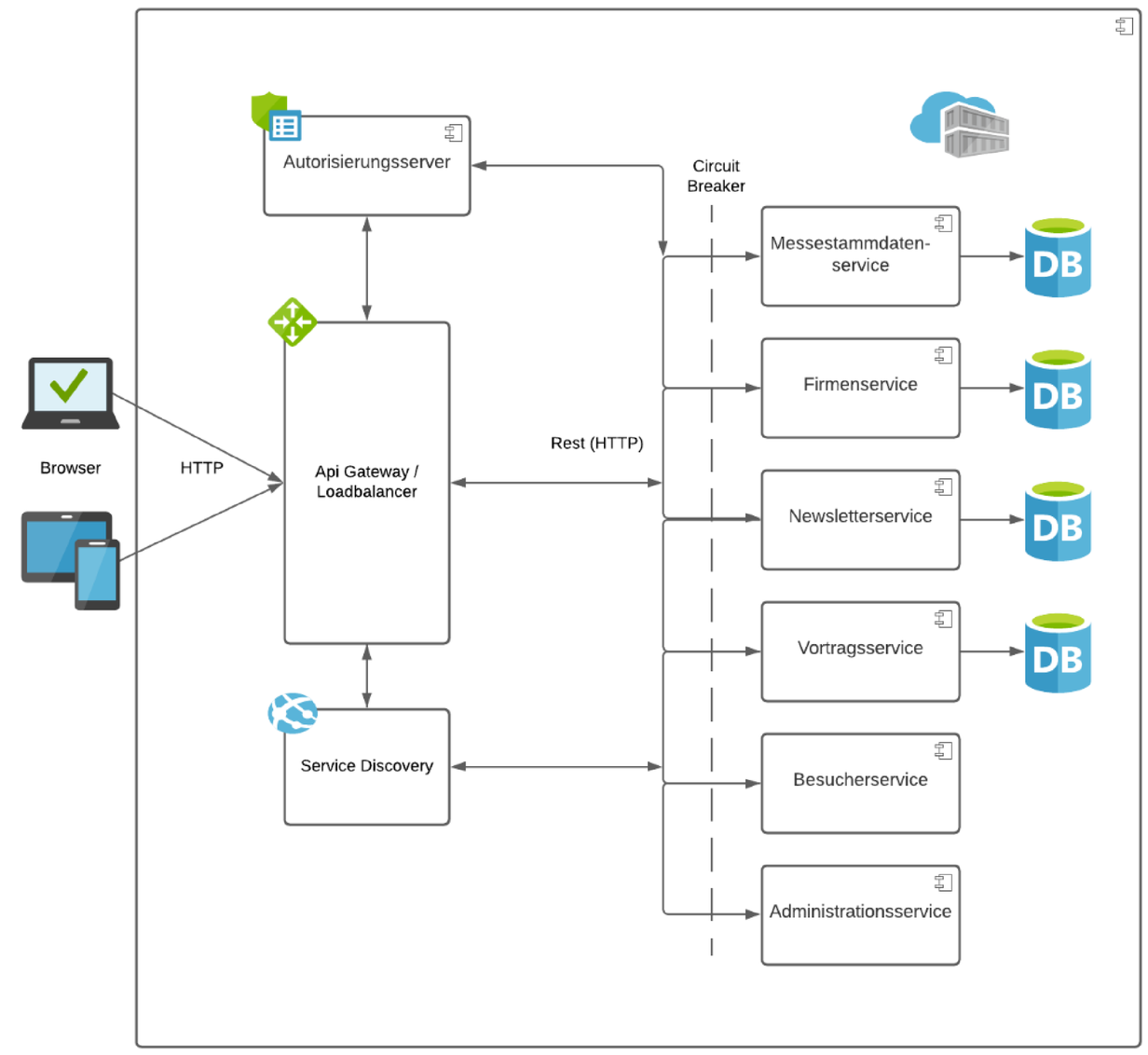
Ein Load Balancer welcher als Software- oder Hardware-Load Balancer angeboten wird, setzt die Lastverteilung in einem Netzwerk um. Ziel ist es Arbeitsbelastung auf Rechenressourcen wie zum Beispiel Servern gleichmäßig zu verteilen, um dadurch die Zuverlässigkeit, Effizienz und Kapazität des Netzwerkes zu optimieren.

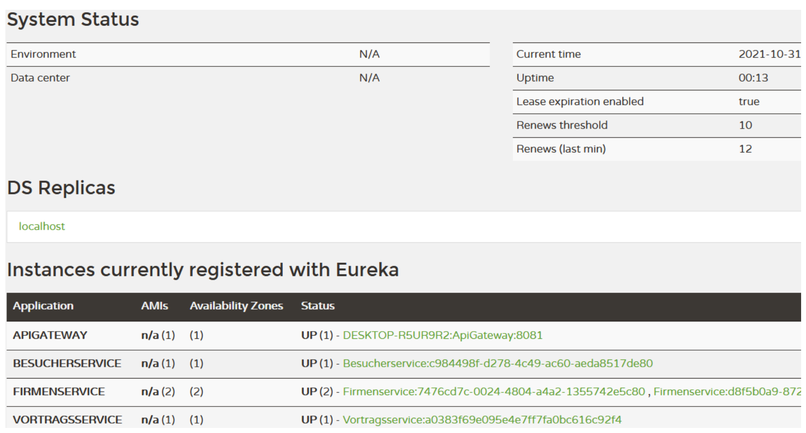
Service Discovery

Service Discovery wird als Software realisiert, welche die Registrierung von Service-Instanzen in einem System ermöglicht. Bei Anfragen werden alle verfügbaren Ziele aus einer Liste mit den registrierten Instanzen abgerufen. Diese Liste, welche Adressen und Ports der Services beinhaltet, wird als Service Registry bezeichnet. Service Discovery ermöglicht es, Service Adressen über den Namen des aufgerufenen Service für den Client aufzulösen. Service-Adresse und Port werden dazu aus der Service Registry übermittelt.

Circuit Breaker

In einer Microservice-Architektur sind in der Regel unter der synchronen Kommunikation mehrere Services voneinander abhängig. Fällt einer dieser Services aus, dann warten abhängige Microservices nach Anfragen an diesen Service auf dessen Antwort. Während der Wartezeit können weitere Anfragen in das System eintreffen. Dieses Verhalten kann dazu führen, dass sich die Anfragen anstauen. Im schlimmsten Fall kann dadurch das gesamte System lahmgelegt werden. Das Circuit Breaker Pattern dient zur Vermeidung einer solchen Problemstellung. Die Funktionalität kann, mit der einer Sicherung in einem elektrischen Stromkreis verglichen werden.

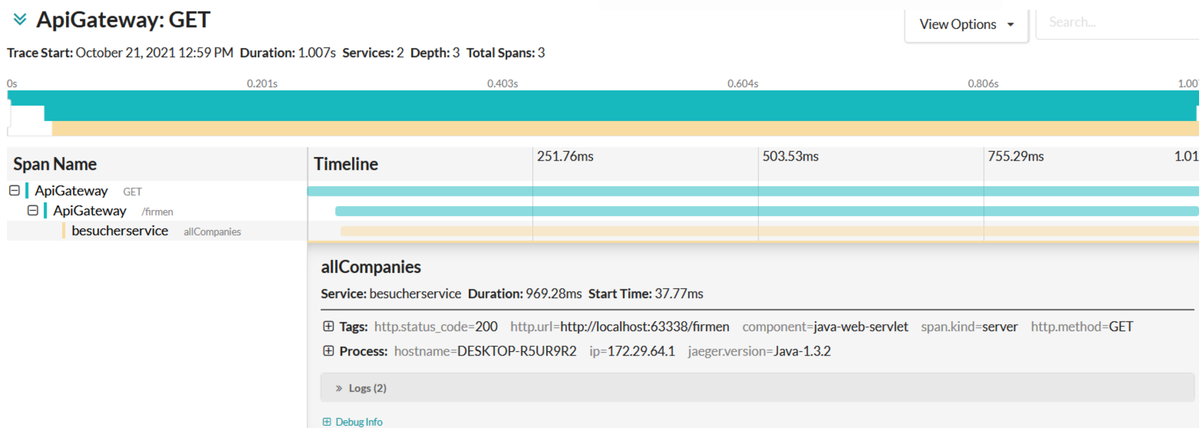




Eureka Service Discovery

Die Service Discovery wird von dem Netflix Tool Eureka umgesetzt. Es handelt sich um eine clientseitige Service Discovery, welche sich relativ einfach über Spring implementieren lässt und daher gut für einen Prototypen geeignet ist.

Über den festgelegten Port des Eureka-Servers (unter der URL *http://localhost:8010/*) erhält man Zugriff zum Eureka Dashboard im Browser. Man erhält von dort aus unter anderem Informationen über alle registrierten Eureka-Clients.



Distributed Tracing mit Jaeger

Jaeger ist ein open source Distributed Tracing System von der Firma Uber. Es dient zur Fehlerbehebung und Überwachung in einer Microservice-Anwendung. Es visualisiert den gesamten Prozessfluss einer Anfrage durch verschiedene Microservices. Folgende Inhalte werden dabei geboten:

- Transaktionsüberwachung
- Ursachenanalyse
- Analyse von Dienstabhängigkeiten
- Latenz- und Performanceoptimierung