

Microservice Architektur in .NET Core

Ein Kunde der Abraxas AG nutzt diverse derer cloudbasierten Produkte. Dabei fallen Kosten an, für die der Kunde eine Rechnung erhält. Um diese Kosten nachvollziehen zu können, muss er Kontakt mit der Abraxas AG aufnehmen, diese wiederum müssen die Daten des Kunden auslesen und ihm eine Antwort zukommen lassen.

Das ist ein unnötig komplizierter Ablauf, der nun mit einer Webapplikation vereinfacht werden soll. Genauer soll der Kunde über das normale Kundenportal der Abraxas AG auf eine Webapplikation gelangen, die seine genutzten Produkte und deren aktuellen Kosten aufzeigt. Er möchte sowohl eine Übersicht über alle seine Kosten haben als auch die Möglichkeit, für jedes seiner Produkte eine Auflistung der genutzten Daten und Dienstleistungen einzusehen. Dies soll ihm vereinfachen, die Kosten nachzuvollziehen.

Um diese Möglichkeit einem Kunden anzubieten, muss eine Applikation erstellt werden, die Daten in einem beliebigen Format von den Produkten entgegennimmt, verarbeitet und ausgibt. Dabei wird eine ansprechende und übersichtliche Benutzeroberfläche benötigt, die alle Daten angemessen ausgibt. Die Daten müssen über eine geeignete Schnittstelle im Backend des Programms verarbeitet und geholt werden; für das Backend soll eine Microservice-Architektur genutzt werden.

Die Microservices sollen auf Probleme und deren mögliche Lösungen hin untersucht werden. Hierbei spielen die Unabhängigkeit, Datenkonsistenz und Performance eine fundamentale Rolle. Durch Recherche sollen geeignete Technologien gefunden und angewendet werden. Ausserdem soll dadurch eine Dokumentation und Untersuchung der Probleme und Gefahren stattfinden.

Um Unabhängigkeit zu gewährleisten, wurde für jeden Microservice eine eigene Datenbank implementiert. Doch dies führt gleichzeitig dazu, dass die Gefahr von Dateninkonsistenzen ansteigt. Diese Gefahr kann durch diverse Ansätze verringert, jedoch nicht vollständig beseitigt werden. Um eine schnelle Kommunikation zu gewährleisten, wurde ein neuer Ansatz der Kommunikation untersucht und angewendet.

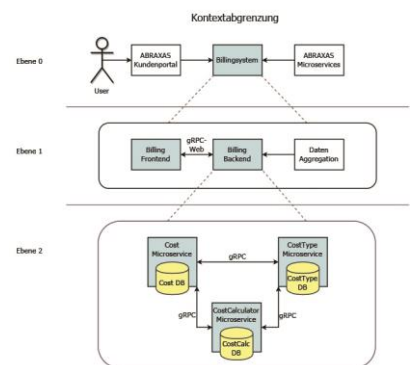


Diplomierende

Armend Bytyqi
Dominik Tripet

Dozent

Karl Rege



Microservice Architektur mit gRPC-Kommunikation