

Optimierung der myGüdel Microservice-Architektur

Studiengang: MAS-IT Software Engineering

Das Kunden- und Serviceportal myGüdel begleitet Anwendende von der ersten Kontaktaufnahme bis zum Betrieb ihrer Maschinen. Angesichts steigender Anforderungen in der Industrie soll die Microservice-Architektur optimiert werden. Ein Proof of Concept soll zeigen, wie der Einsatz eines Message Brokers und moderner Architekturmuster eine entkoppelte und skalierbare Weiterentwicklung ermöglicht.

Ausgangslage

Die Güdel Group AG ist ein international tätiges Unternehmen, das auf die Entwicklung, Herstellung und Integration von Automationslösungen spezialisiert ist. Mit dem Kunden- und Serviceportal myGüdel steht eine zentrale Plattform bereit, die Anwendenden vom Erstkontakt bis zum Betrieb und zur Ersatzteilbestellung begleitet. Güdel optimiert Applikationen fortlaufend und integriert internes wie externes Feedback systematisch in den Entwicklungsprozess. So wird sichergestellt, dass die Anwendungen aktuellen Standards entsprechen und die Bedürfnisse der Anwendenden erfüllen.

Zielsetzung

Die Microservice-Architektur von myGüdel soll analysiert und optimiert werden. Im Mittelpunkt steht ein Proof of Concept, der zeigt, wie der Einsatz eines Message Brokers und moderner Architekturmuster eine entkoppelte, skalierbare Weiterentwicklung ermöglicht. Dieses Konzept soll die Grundlage für den langfristigen Ausbau und die nachhaltige Entwicklung der Plattform bilden.

Vorgehen

Die Analyse der Microservice-Architektur und ihrer Applikationen wurde durch eine Literaturrecherche ergänzt, um bewährte Methoden in die Entwicklung des Proof of Concept zu übertragen und fundierte Ansätze für die Weiterentwicklung der bestehenden Systemarchitektur aufzuzeigen. Auf dieser Basis

wurde unter Berücksichtigung des Nutzerverhaltens ein Teilbereich definiert, der als Grundlage für die Umsetzung diente.

Um einen geeigneten Message Broker zu bestimmen, wurde eine Nutzwertanalyse auf Basis definierter Kriterien durchgeführt. Dabei schnitt RabbitMQ im Gesamtergebnis am besten ab und wurde als bevorzugte Lösung definiert.

Der Broker wurde im Rahmen einer exemplarischen Architektur unter Verwendung des Outbox-Patterns implementiert und in einer Demoumgebung bereitgestellt, die analog zur Produktivumgebung aufgebaut ist.

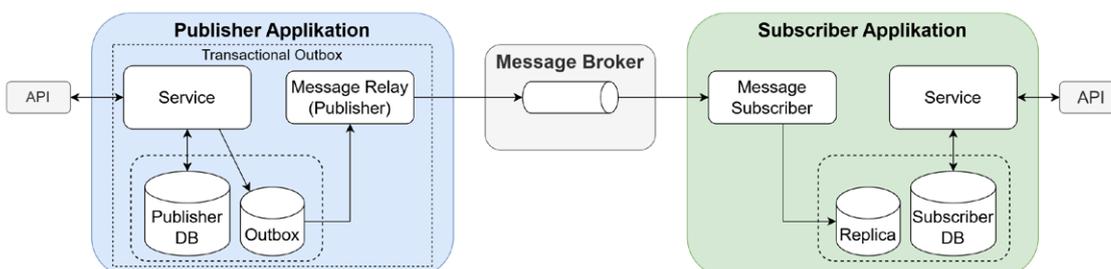


Patrick Stern

Ergebnis

Die Implementierung zeigt, wie RabbitMQ in Kombination mit einem Outbox-Pattern die Kommunikation zwischen den Applikationen resilient gestaltet. Das Outbox-Pattern stellt sicher, dass Nachrichten zuverlässig verarbeitet werden, auch wenn temporäre Ausfälle einzelner Komponenten auftreten.

Im Proof of Concept wird die Entkopplung der Applikationen exemplarisch umgesetzt. Dabei bleiben anwendungsrelevante Daten weiterhin zentral verwaltet, wobei das Outbox-Pattern sicherstellt, dass die Microservices stets über den aktuellen Stand verfügen. Auf diese Weise wird die Eigenständigkeit der Applikationen gestärkt und Abhängigkeiten innerhalb der Architektur reduziert. Der Proof of Concept bietet somit eine anwendungsorientierte Basis für die Weiterentwicklung von myGüdel.



Architektur des Proof of Concept mit implementiertem Outbox-Pattern und Message Broker zur Entkopplung von Publisher- und Subscriber-Applikation.