

Neudesign einer Prozessleitsystem-Architektur

Studiengang : MAS-IT Software Architecture

Im Rahmen der Digitalisierungs- und Datenstrategie hat sich Endress+Hauser Flowtec AG entschieden, den Life Cycle Prozess und die Technologie des Prozessleitsystems komplett zu erneuern. Im Fokus stehen kürzere Anbindungszeiten von Maschinen und Anlagen, ein umfangreicher Ausbau der Prozess- und Produkt-Datenerfassung für Optimierungs- und Qualitätszwecke sowie ein durchgängiger Entwicklungs- und Betriebsprozess.

Die Arbeit befasst sich neben den Gesamtarchitekturaspekten auch mit einem **durchgängigen Entwicklungs- und Betriebsprozess**, um zukünftige Anforderungen schneller zu ermöglichen, eine optimale Codequalität zu erreichen und letztendlich kostentreibende Störungen zu vermeiden.

Motivation

Endress+Hauser hat eine gruppenweit abgestimmte Initiative lanciert, um Produktionsdaten in einem standardisierten Format mit Hilfe von **Apache Kafka** im Stream zu verarbeiten. Ziel dieses Vorhabens ist es, neben der Standardisierung ein vereinfachtes Anschliessen von weiteren Datenquellen der Produktion zu ermöglichen. In diesem Zusammenhang wurden **Apache Avro Schemas** entwickelt, welche nun als Grundlage für die Datenübertragung und Modellierung dienen. Das Einsetzen von Apache Kafka macht das Aufsetzen einer neuen Gesamtarchitektur notwendig. Die neue Architektur wird neue Technologien aus dem Cloud Umfeld nutzen. Dazu werden die Applikationen containerisiert und von **Kubernetes**

orchestriert betrieben. Die einzelnen Dienste werden nach dem Microservice Ansatz voneinander entkoppelt betrieben.

Architekturmuster

Hierzu existieren diverse etablierte Architekturmuster, die sich mithilfe verschiedener Open Source Frameworks implementieren lassen. Daher wurden verschiedenen Architekturen wie **Service Mesh** oder **API Gateways** verglichen. Ein relativ neues Framework, **Dapr**, würde viele Anforderungen erfüllen. Jedoch gibt es dazu keine direkte Unterstützung für Apache Avro. Daher wurde nach einer Bewertung der etablierten Lösungsvarianten im Rahmen eines Proof of Concepts eine eigene Lösung entwickelt, welche einen generalisierten Sidecar für die Microservices bereitstellt.



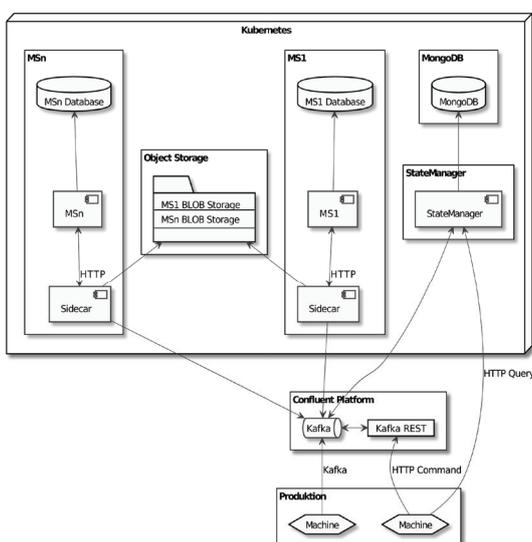
Fabrice Andreis

DevOps

Mithilfe von **API Tests** wird der Code in realitätsnaher Umgebung automatisiert getestet. Sicherheitsscans stellen sicher, dass keine schädlichen Abhängigkeiten existieren und verhindern so Lieferkettenangriffe. Im **Azure DevOps** Prozess wird Code vor der Freigabe in der Testumgebung bereitgestellt und kann somit unter Realbedingungen getestet werden. **Monitoring und Tracing** ermöglichen es jederzeit, den Zustand der Dienste zu überwachen und einzelne Abfragen über alle Punkte nachzuverfolgen. Dokumentationen können aus **Markdown** - inklusive UML Diagrammen - als Webseite veröffentlicht werden.

Ergebniss

Der Proof of Concept zeigt, dass die neuen Technologien aufgrund ihrer Komplexität und der noch geringen Expertise im industriellen Umfeld ein hohes Mass an Wissensaufbau erfordern. Jedoch wird dieser Aufwand nach dem Bereitstellen der Infrastruktur belohnt durch skalierbare, nachverfolgbare Prozesse und einer beschleunigten, modularisierten Entwicklung.



Mit MkDocs und PlantUML generiertes Architekturdiagramm