

Einführung in Microservices

Sebastian Heuer

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg

03.03.2022

Zielsetzung

- Einführung in das Thema Microservices
- Überblick über Herausforderungen und Technologien
- Nachvollziehen eines Definitionsversuchs
- Wertung

Zielsetzung

Folgende Punkte werden (fast) nicht behandelt:

- Vor- und Nachteile
- Service-Komposition
- Frameworks, Docker und Kubernetes

Geschichte

- 1999 - "resource oriented computing" - Dexter research project, Hewlett Packard Labs - REST als Teil von ROC abstraction
- 2005 - "Software components are Micro-Web-Services" - Dr. Peter Rodgers, Web Services Edge conference
- 2009 - Netflix spaltet den Monolithen und geht in die Cloud
- 2011 - "Microservices" - erste Erwähnung, workshop for software architects
- 2012 - "Microservices" - Namensgebung, workshop for software architects
- 2014 - "Microservices, a definition of this new architectural term" - Lewis Fowler, Fachartikel für Thoughtworks
- 2015 - "Building Microservices" - Buch von Sam Neil, Nachschlagewerk für design, development, deployment, testing und maintenance

Grundlagen

Service-Oriented Architecture

Softwarekomponenten sind eigene Applikationen in einem verteilten System.

Agile Entwicklung

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Grundlagen

DevOps

- Methodensammlung zum Kombinieren von Entwicklung (Dev) und Organisation(Ops)
- Effektive Prozesse und Werkzeuge zur Steigerung der Qualität und Geschwindigkeit
- Betrifft: Entwicklung, Qualitätsmanagement, Auslieferung, Wartung, Kommunikation, ...
- Beispiele: Virtualisierung, Versionsverwaltung, Infrastructure as Code, Continuous Integration

CI/CD

Continuous Integration, Continuous Deployment
Durchgehende Auslieferung von kleinen Änderungen anstatt
Sammeln von Änderungen in größeren Updates

Motivation

Wir haben eine Sammlung von Anforderungen an die Softwareentwicklung

Organisation

- Wir betrachten große, komplexe Anwendungen.
- Wir wollen kleine, autonome Teams.

Prozesse

- Agile Methoden
- CI/CD
- DevOps
- ...

Architektur

- Welcher Stil passt zu unseren Anforderungen?

Microservices

Microservices

Microservices

Kritik

Bewertung

Versuch 1: Komponententrennung durch Services

Versuch 2: Zerlegen nach technischen Schichten

Versuch 3: Strukturierung nach Geschäftseinheiten

Bewertung der Definition

Zusammenfassung

Microservices

- Beschreibung eines Architekturstils
- Keine formale Definition
- Definitionsversuch über gemeinsame Merkmale
- Aufbrechen von vorhandener Software entlang von geschäftsrelevanten Grenzen

Zusammenfassung

Microservices

- Geringe Kosten für Erweiterung, Skalierung
- Automatisierung von Deployment
- Hoher Einarbeitungsaufwand

Danke für Ihre Aufmerksamkeit

Fragen?



Wann sollte man keine Microservices verwenden?

- Geschäftsprozesse müssen bekannt sein.
Änderungen in Service-Service Beziehungen sind teuer!
- Bei Neuentwicklungen sind die Abhängigkeiten zwischen Geschäftsprozessen nicht ausreichend bekannt.
Man sollte mit einem Monolithen starten!
- Probleme die durch Microservices verursacht werden verschlimmern sich mit der Skalierung.
Implementierung von Microservices sollte graduell passieren, je nach Änderungsvermögen des Systems!

Besondere Services

- Ingress
- API Gateway
- Eventbus
- Dynamic Service Registry
- Management/Orchestration

Aufbau eines Microservices

- Cluster IP
- Cache
- Security
- Safety