Planung und Erstellung einer Backend-Microservices-Architektur aus den Anforderungen durch das Spiel Stirnraten.

Michael Rothkegel

28. März 2019

Inhaltsverzeichnis

1 Glossar

Begriff	Erklärung
Deployen	Ein Softwareprodukt updaten.
Hosten	Ein Softwareprodukt auf einem Server bereitstellen. Dafür stehen heutzutage viele Möglichkeiten bereit, vom eigenen Server, gemieteten Servern oder Cloudlösungen (Azure, AWS, Google Cloud)
CI/CD	Continuous Integration/Continuous Delivery beschreibt den kontinuierlichen Prozess ein Softwareprodukt von der Entwicklung bis zur tatsächlichen Veröffentlichung zu begleiten. Dies sollte automatisch funktionieren, schnell gehen und leicht auszuführen sein.
User Interface	Oberfläche für Benutzer
VM	Virtuelle Maschine, Kapselung eines Rechnersystems innerhalb eines anderen
Legacy	Es handelt sich um Altsysteme, die eine Erneuerung benötigen oder abgelöst werden müssen. Das kann verschiedene Gründe haben, wie z.B. schlechte Programmierung, Fehlentscheidungen oder technologische Veraltung
Technologiestack	
Overhead	
Downtime	Zeitpunkt, die eine Anwendung benötigt, bis sie neugestartet ist. Während der Downtime ist eine Anwendung nicht erreichbar.
technische Schuld	
Monolith	monolitsche Struktur
Microservice	Service
Domain Driven Design (DDD	

2 Einleitung

3 Grundlagen

Hier erklären was kommt

3.1 Microservices

Für den Begriff Microservices existiert keine einheitlich anerkannte Definition. Während Wolff unter Microservices unabhängig, deploybare Module versteht, spricht Newman von kleinen, autonomen Services, die zusammenarbeiten. Cockcroft verwendet den Be-

griff Microservice gekoppelt mit einem Architekturbegriff: Eine Microservice Architektur sind gekoppelte Services, welche für einen gewissen Kontextbereich zuständig sind. D.h. jeder Service behandelte gewisse, fachliche Aufgaben und kann genau für diese genutzt werden. Eine Vielzahl von solchen Services bildet dann die gesamte Anwendung.

Amudsen schreibt dem Microservice an sich die Eigenschaft zu, dass er unabhängig zu anderen Microservices sein muss, d.h. ein Microservice kann losgelöst von anderen geupdated (deployed) werden. Weiter ist ein Microservice wie schon bei Cockcroft für einen gewissen Aufgabenbereich zuständig. Eine Microservice-Architektur ist ein zusammenschluss von miteinander kommunizierenden Microservices.

In *Flexible Software Architecture* werden Microservices zu den bisherigen noch weitere, teils technische Eigenschaften zugeschrieben: Microservices sind technologisch unabhängig, d.h. eine Microservice Architektur ist nicht an eine Programmiersprache gebunden. Weiter müssen Microservices einen privaten Datenspeicher haben und sie kommunizieren mit anderen Services über das Netzwerk (z.B. über REST). Ebenfalls werden Microservices verwendet, um große Programme in kleine Teile zu unterteilen. Diese kleine Teile lassen sich automatisch bauen und deployen.

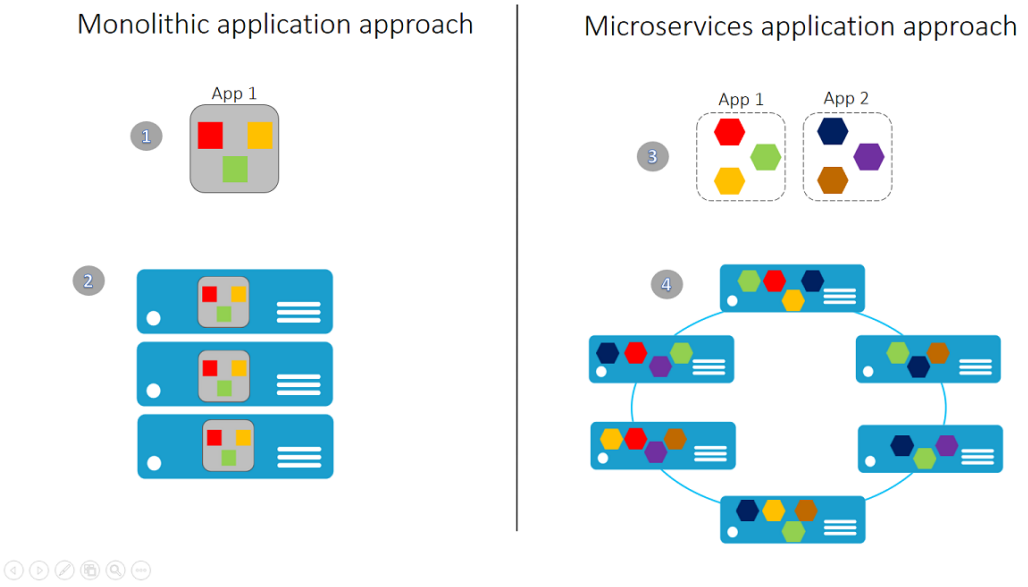

Abbildung 1: Sie sehen Unterschiede.[1]

4 Konzept

Das ist der Grundlagenteil

5 Implementierung

Das ist der Grundlagenteil

6 Fazit

Das ist der Grundlagenteil

Literatur

[1] Matt McLarty Mike Amundsen Irakli Nadareishvili Ronnie Mitra. *Microservice Architecture - Aligning Principels, Practices, and Culture*. O Reilly, 2016.