Microservices

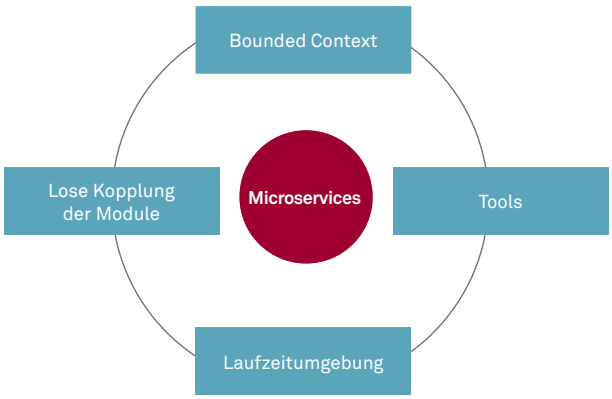
Softwarearchitektur für skalierbare Anwendungen und Entwicklungsprozesse

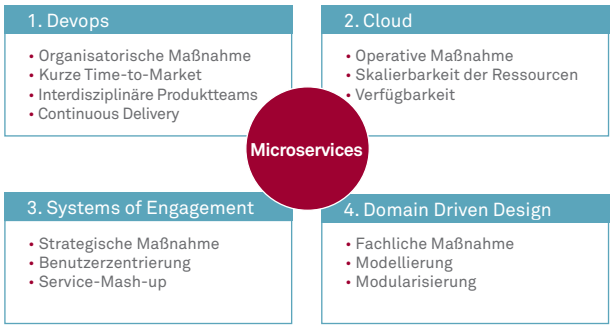
Eine kurze Vorlaufzeit (engl. time to market) ist im Zeitalter der Digitalen Transformation für viele Unternehmen ein strategischer Vorteil. Neben Systems of Engagement auf strategischer und Devops auf organisatorischer Seite, lösen Microservices dieses Dilemma auf architektureller Seite. Sie erlauben es, Lastspitzen zur Laufzeit jederzeit mit zusätzlichen Ressourcen auszugleichen.

Definition

Microservice bezeichnet eine lose gekoppelte und verteilte Architektur. Jeder Microservice hat dabei seine eigene Ablaufumgebung oder Plattform, die ihn ausführt. Aufgrund der verteilten Architektur skalieren Microservices sehr stark. Ein einzelner Microservice kann bei Bedarf automatisch mehrfach ausgerollt werden, um Auslastungsschwanken zu kompensieren.

Es gibt keine Größenfestlegung für Microservices, so dass sie beliebig klein ausfallen dürfen. Entscheidend ist, dass sie fachlich abgeschlossen sind und keine direkten Abhängigkeiten aufweisen. Nur so lassen sich Microservices unabhängig voneinander entwickeln, produktiv stellen und ersetzen. Microservices sind isoliert von anderen Microservices, beinhalten ihre eigene Datenbank und eine eigene Bedienoberfläche und werden von dem Team betrieben, das ihn entwickelt hat.





Referenzszenario

Ein Unternehmen möchte ein Systems of Engagement aufbauen, um einerseits Dienste mit hohem Kundennutzen anzubieten und sich andererseits in Wertschöpfungsketten von Partner- und Drittunternehmen integrieren zu können. Dazu muss das Unternehmen zunächst die organisatorischen Voraussetzungen durch Devops schaffen. Die durch Devops etablierten interdisziplinären Produktteams entwickeln und betreiben fortan die Microservices.

Die Microservices werden nach Fachlichkeit modularisiert, um Abhängigkeiten untereinander zu minimieren. Dazu setzt das Unternehmen auf Domain Driven Design. Durch die fachliche Trennung kann das Unternehmen einzelne Microservices beliebig durch andere Microservices ersetzen und so sehr schnell auf Veränderungen im Anwenderverwalten oder dem Markt zu reagieren.

Dabei erklärt sich das Unternehmen bereit, den Entwicklungsteams große Entscheidungsfreiheit bei der Ausgestaltung der Microservices zu lassen. Denn nur so entstehen in kürzester Zeit die Microservices, die die gestellten Anforderungen maximal erfüllen.

Amazon und Netflix haben das Prinzip von Devops und Microservices erfolgreich umgesetzt.

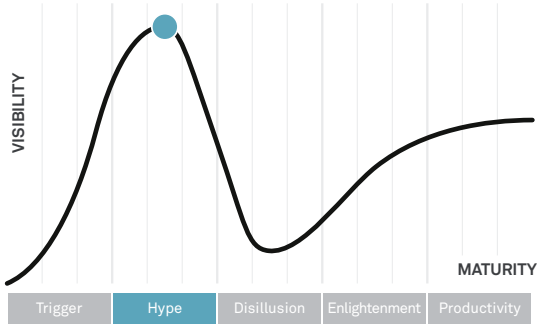
Business Impact

Mit Microservices lässt sich spontan auf neue Trends am Markt reagieren. Die Architektur erlaubt schnelle Anpassungen und kurzfristige Deployments von neuen Funktionalitäten, so dass Microsevices zudem als Wegbereiter für Systems of Engagement wirken. Die entstandenen Services lassen sich jederzeit zu neuen Systemen und Anwendungen zusammenstellen, so dass ein Service-Mash-up entsteht. Die hohe Skalierbarkeit erlaubt ein starkes Wachstum des Geschäfts, das einen direkten Vorteil im Wettbewerb bedeutet. Klassische, monolithische Architekturen können diese Vorteile nicht bieten.

Reifegrad

Grundsätzlich befinden sich Microservice-Architekturen derzeit auf dem Höhepunkt der Hype-Phase. Viele Unternehmen erkennen die Architektur als den Schlüssel zu ihrem Digitalisierungserfolg. Sie übersehen aber mitunter die strategischen, organisatorischen und fachlichen Voraussetzungen, die sie zuvor schaffen müssen. Außerdem schätzen sie ihren Bedarf oft zu hoch ein, weil ihr Unternehmen gar nicht in dem erwarteten Maße wächst.

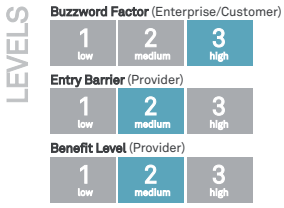
Unternehmen wie Amazon und Netflix sind Vorreiter der Digitalisierung und Early Adopter der Microservice-Architektur. Sie haben die Architektur maßgeblich durch eigene Entwicklungen geprägt und stellen etwa schlüsselfertige Technologie-Stacks und Infrastruktur bereit. Sie setzen Microservices im großen Stil produktiv und erfolgreich ein.

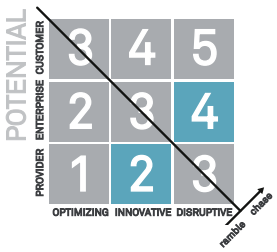


Marktübersicht

Microservices sind unabhängig von Technologien. Sie schreiben im Gegenteil sogar vor, jede Technologie zu verwenden, die die spezifischen Anforderungen am besten abdeckt. Die Umsetzung kann deshalb in beliebigen Programmiersprachen erfolgen und die Sprache sogar zwischen Microservices desselben Unternehmens wechseln.

Besonders häufig zitiert und verwendet wird der so genannte Netflix-Stack in der Kombination mit dem Spring-Framework und Java. Er adressiert bereits viele technische Aspekte einer verteilten Microservice-Architektur. Sämtliche Komponenten des Netflix-Stacks sind Open Source.





Alternativen

In der Praxis werden neben reinen Microservices auch Hybride-Lösungen eingesetzt, die das Prinzip von Shared Assets aufgreifen. Dabei werden Datenbanken, UI-Komponenten oder Bibliotheken zwischen den Microservices geteilt. In der Java-EE-Welt wird oft ein einzelner Applikationsserver für mehrere Microservices geteilt, um Ressourcen zu sparen. Jeder Microservice ist dabei ein Java-EE-Webmodul.

In den alternativen Szenarien entstehen allerdings technische Abhängigkeiten zwischen den Microservices, zu ihren Ressourcen oder zu einer Laufzeitumgebung respektive Cloud-Plattformen. Dadurch verlieren Microservices an Zugkraft.

Pro	Contra
Kurze Implementierungs- und Veröffentlichungsphasen einzelner Microservices (time to market)	Hohe Komplexität aufgrund der Entkopplung und Verteilung der Schnittstellen
Hohe Skalierbarkeit der Microservices und der Entwicklungsprozesse	Komplexe Testszenarien aufgrund der hohen Flexibilität
Hohe Ausfallsicherheit durch Verteilung der Microservices	Nur mit Framework- oder Plattform-Unterstützung praktikabel einsetzbar

https://msg.direct/techrefresh

Stand September 2017

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München | Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113 | www.msg.group | info@msg.group