

PRESSEMITTEILUNG

Der „tanzende Monolith“: Consol modernisiert das klassische Architekturmodell

München, 31. Juli 2019 – Trotz ihres schlechten Images sind monolithische Applikationen in einigen Fällen die bessere Wahl gegenüber einer Microservices-Architektur. Oliver Weise, Software-Architekt bei Consol, erklärt, wie Programmierer sich das Beste aus beiden Welten zunutze machen und flexible, ressourcenschonende und cloudfähige Software „am Stück“ entwickeln.

Microservices-Architekturen gelten als die Crème de la Crème der Software-Entwicklung. Nicht zu Unrecht: Microservices sind sauber strukturiert, granular skalierbar, ressourcenschonend, leicht erweiterbar und unterstützen die agile Software-Entwicklung in Teams. Mit einem lose gekoppelten Microservices-Zoo handeln sich IT-Abteilungen in diesem stark verteilten System aber auch einen gewissen Overhead hinsichtlich Verwaltung und Kommunikation ein, der den großen Mehraufwand nicht immer rechtfertigt. Gerade bei kleineren Software-Projekten mit überschaubarem Skalierungsbedarf und kleineren Entwicklungs-Teams, einer Situation, wie man sie oft in mittelständischen Unternehmen vorfindet, sind Microservices eher ungeeignet. Dort empfiehlt Consol eine Software-Entwicklungsmethodik, die monolithisch startet, jedoch bei Bedarf auch die Vorteile von Microservices nutzen kann. Oliver Weise, Software-Architekt bei Consol, erklärt die Grundprinzipien:

1. Prinzip: Sourcecode sauber nach Funktionalitäten strukturieren

Sinnvoll ist, auch im Monolithen eine klare vertikale Trennung nach Funktionalitäten vorzunehmen, die über Schnittstellen lose miteinander gekoppelt sind. Modularisierungs-Frameworks wie OSGi oder Jigsaw können Programmierer dabei unterstützen, diese Trennung im Sourcecode einzuhalten. Die Komponenten werden gemeinsam bereitgestellt (Deployment), in einem gemeinsamen Prozess betrieben und skalieren auch nur „am Stück“, das heißt über weitere Instanzen des Gesamt-Deployments. Für sehr viele Anwendungsfälle ist diese Skalierbarkeit durchaus ausreichend. Sollte sich später im Betrieb herausstellen, dass eine Software-Komponente zeitweilig einen besonders hohen Skalierungsbedarf hat, lässt sie sich dank der internen Modularisierung immer noch extrahieren und als separaten Microservice betreiben.

2. Prinzip: Ressourcen-Exzesse vermeiden

Die berüchtigten Ressourcen-Exzesse existierender monolithischer Applikationen sind eher auf veraltete Plattformen und Legacy-Konzepte als auf die Architektur zurückzuführen. In vielen Fällen können Monolithen ebenfalls sparsam im Ressourcenverbrauch sein, vorausgesetzt, sie bedienen sich moderner Entwicklungsplattformen und Prinzipien, etwa durch die Verwendung externer Caching-Dienste wie zum Beispiel Redis. So lassen sie sich auch auf Cloud-Plattformen wie Kubernetes ausrollen und problemlos zwischen den Cluster-Knoten bewegen. Vorausgesetzt, der Fokus der Software-Entwickler liegt von Anfang an auf bewusster Ressourcenverwaltung und schnellem Startup sowie Shutdown. Entpuppt sich eine Komponente im Nachhinein trotzdem als Ressourcenfresser, kann sie relativ einfach extrahiert und der Ressourcenverbrauch reduziert werden.

3. Prinzip: Die Best Practices der 12-Faktor-App-Methodik berücksichtigen

Die 12-Faktor-App-Methodik ist eine Sammlung von Best Practices zur Entwicklung von Software-as-a-Service-Applikationen für die Private und die Public Cloud. Wesentliche Bestandteile der Empfehlungen sind eine saubere Kapselung der einzelnen Sourcecode-Komponenten und klare Schnittstellen zu Betriebssystemen und Services. Die Methodik empfiehlt, implizite Abhängigkeiten zu Bibliotheken und System-Tools zu vermeiden und alle vorhandenen Abhängigkeiten explizit zu deklarieren. Orientieren sich Software-Entwickler konsequent an den Best Practices der 12-Faktor-App-Methodik, werden auch monolithische, in Containern betriebene Anwendungen flexibel, agil, cloud-fähig und ressourcenschonend.

4. Prinzip: Den Monolithen zum Tanzen bringen

Moderne Release-Prozesse mit vollautomatisierten Tests, unveränderlichen Deployments und Infrastruktur-Automation eignen sich auch für monolithische Architekturen. Da man nur einen Prozess braucht, im Vergleich zu einer Microservices-Umgebung, die viele Prozesse benötigt, spart ein Monolith sogar Aufwand. Zwar gibt es an einem Monolithen mehr zu testen als an einem Microservice, was die Laufzeit des Prozesses in die Länge treibt. Durch Parallelisierung ist dies in der Regel aber in den Griff zu bekommen.

Eine weitere Voraussetzung dafür, flexible Monolithen zu entwickeln, ist neben der Berücksichtigung der technischen Best Practices auch der Abbau von organisatorischen Hindernissen im Unternehmen. Träge Freigabeprozesse für Releases oder für den Einsatz moderner

Software-Entwicklungsplattformen, mangelnde Kommunikation und fehlendes Vertrauen zwischen Development und Operations haben in der Vergangenheit monolithischen Anwendungen oft zu schaffen gemacht. Ein Monolith darf niemals die Entschuldigung dafür sein, DevOps-Prinzipien zu vernachlässigen oder ganz zu verhindern.

„Ein flexibler, beweglicher Monolith kann durchaus heute noch in der modernen IT bestehen und ist bei kleineren Projektvorhaben oft noch immer die beste Wahl“, betont Oliver Weise, Software-Architekt bei Consol.

**Diese Presseinformation und Bildmaterial sind im
Internet abrufbar unter www.pr-com.de/consol**

Über Consol

Die Consol Consulting & Solutions Software GmbH betreut seit mehr als 30 Jahren Kunden aller Branchen bei nationalen und internationalen IT-Projekten. „Wir unternehmen IT“ ist dabei das Credo, auf dessen Basis die Spezialisten, Umsetzer und Innovationstreiber bei Consol passgenaue IT-Lösungen für den gesamten Software-Lifecycle erarbeiten: High-End IT-Beratung, Software Engineering, IT Operations und DevOps sind die Kernkompetenzen des 1984 gegründeten Unternehmens mit Hauptsitz in München.

Die technologischen Schwerpunkte liegen unter anderem auf Software-Architektur, Cloud-native, CI/CD, Testautomatisierung und Monitoring. Consol verfolgt hierbei einen agilen Arbeitsansatz und nutzt unter anderem Open-Source-Lösungen. Darüber hinaus entwickelt und vertreibt das Unternehmen die Software Consol CM, eine Plattform zur Digitalisierung von Geschäftsprozessen.

Consol ist Red Hat Premier Partner und unterhält strategische Partnerschaften zu AWS oder Microsoft Azure. Zu den Kunden zählen Großunternehmen wie Haribo, Daimler oder Telefónica.

Die Faszination der Consol-Mitarbeiter für technologische Herausforderungen bildet die Basis des Unternehmenserfolgs. Aktuell beschäftigt Consol rund 260 Mitarbeiter an seinen Standorten München, Düsseldorf, Wien, Krakau, Dubai sowie San Francisco.

Weitere Informationen unter <https://www.consol.de> und <https://cm.consol.de> sowie auf Twitter unter https://twitter.com/consol_de.

Pressekontakt

ConSol Consulting & Solutions Software GmbH
Isabel Baum
St.-Cajetan-Straße 43
D-81669 München

Fon: +49-89-45841-101

Fax: +49-89 45841-111

E-Mail: Isabel.Baum@consol.de

Web: <https://www.consol.de> und <https://cm.consol.de>

PR-COM GmbH

Nicole Oehl

Sendlinger-Tor-Platz 6

D-80336 München

Fon: +49-89-59997-758

Fax: +49-89-59997-999

E-Mail: nicole.oehl@pr-com.de

Web: www.pr-com.de