



# # 1 CUSTOMER STORY

[WWW.ARACOM.DE](http://WWW.ARACOM.DE)

## VERTRIEBS- UND VERWALTUNGS- PLATTFORM FÜR SOFTWARE UND PRODUKTE

---

MAI 12, 2020

AUTOR: DR. MATTHIAS SOMMER

Die Anforderung unseres Kunden war die Umsetzung einer zentralen Anlaufstelle für die gesamte Kundensoftware in Form eines Internetportals bereitzustellen.

Der Kunde findet - nach der Bestellung eines Produktes - alle Unterlagen, Tutorials, sowie benötigte Software & Tools an einem Ort. Die Anwendung muss einfach zu bedienen und zielgerichtet zugeschnitten sein.

Besonderer Fokus lag dabei auf der Nutzbarkeit, sowie einer zuverlässigen und zukunftsreichen Architektur- und Technologieauswahl.

Darüber hinaus stellt das Onlineportal eine zentrale Quelle für Dokumente, Produkte, Seriennummern und Tools dar. Diese Daten werden anderen internen Anwendungen über Schnittstellen bereitgestellt.

## INHALTSVERZEICHNIS

1. ANFORDERUNGEN
2. WIE HELFEN MICROSERVICES BEI VERTRIEBS- UND VERWALTUNGSPLATTFORMEN
3. AUSFALLSICHERHEIT UND SKALIERBARKEIT DER VERWALTUNGSPLATTFORM DURCH CLOUD INFRASTRUKTUR
4. HOHE CODEQUALITÄT DURCH CONTINUOUS INTEGRATION SICHERSTELLEN
5. FLEXIBILITÄT UND WIEDERVERWENDBARKEIT IM FRONTEND
6. IMPORT UND VALIDIERUNG VON DATENQUELLEN ZU VERTRIEBS- UND PRODUKTINFORMATION
7. ÜBERWACHUNG UND ANALYSE DER MICROSERVICES IM LAUFENDEN BETRIEB
8. FAZIT: VORTEILE UND HERAUSFORDERUNGEN VON MICROSERVICES

### 1. DIE VERTRIEBS- UND VERWALTUNGSPLATTFORM MUSS DABEI FOLGENDEN ANFORDERUNGEN GENÜGEN:

- Unterstützung verschiedener Authentifizierungsverfahren, z.B. Single Sign On mit PingFederate
- Nutzer-Autorisierung und Rollenmanagement (u.a. IT-Administrator, Projekt-Administrator, Projektmitglied)
- Projektverwaltung
- Schnittstelle zu einer Desktopanwendung für die Installation von Tools
- Geschützter Administrationsbereich zum Verwalten von Produkten, Ressourcen und Nutzern
- Bereitstellung von Schnittstellen zu anderen firmeninternen Anwendungen
- Täglicher Datei- und Datenbank-Import aus anderen Anwendungen

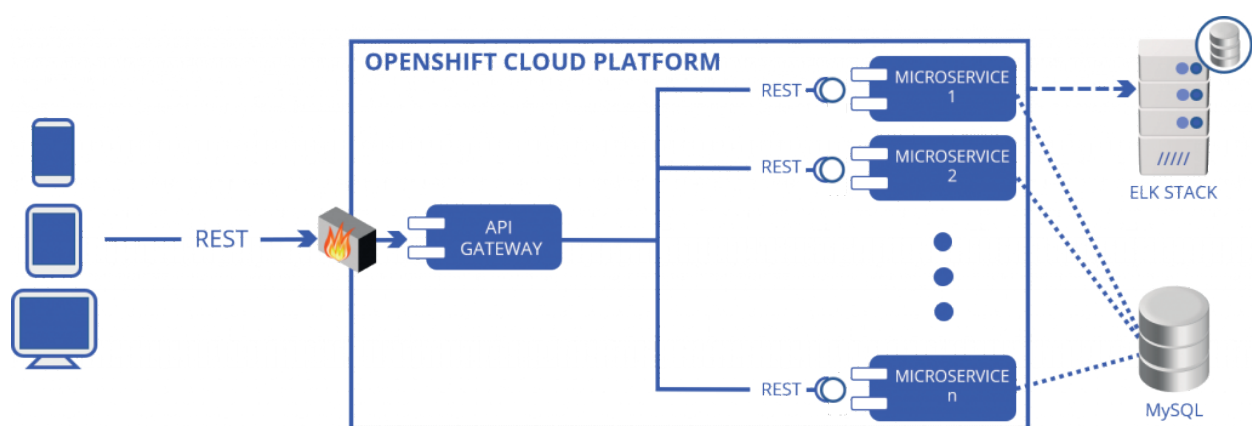
### 2. WIE HELFEN MICROSERVICES BEI VERTRIEBS- UND VERWALTUNGSPLATTFORMEN?

Die Microservice-Architektur ermöglicht die schnelle, häufige und zuverlässige Bereitstellung großer, komplexer Anwendungen.

Die Aufteilung einer großen Anwendung wie dieser umfangreichen Verwaltungsplattformen in lose zusammenhängende, spezialisierte Dienste (die Microservices) bietet folgende Vorteile:

- Schnellere Reaktion auf sich ändernde Geschäftsanforderungen
- Dienste können in verschiedenen Programmiersprachen entwickelt werden
- Dienste können unabhängig voneinander entwickelt und ausgeliefert werden
- Jeder Dienst ist auf eine Aufgabe spezialisiert und kann einfach durch eine andere Variante ersetzt werden
- Häufiges Ausliefern neuer Versionen der Software

Die Microservices wurden von unserem Team mit der Programmiersprache Golang programmiert. Golang wird seit 2007 von Google entwickelt, um die Probleme der Unterstützung von Skalierbarkeit und Effektivität auf Cloud-Infrastrukturen zu lösen. Die Sprache bietet eine einfache Syntax und eine sehr performante Leistung, da der Code direkt als Binärdatei kompiliert wird. Dazu kommen der geringe Speicherbedarf (5-15 MB) und die integrierte Parallelisierung. Der Programmcode ist plattformunabhängig und somit auf verschiedenen Betriebssystemen einsetzbar.



Der modulare Aufbau unserer Microservices nach dem Domain Driven Design erlaubt es uns in kürzester Zeit weitere Dienste zu implementieren.

Der Erfolg dieses Ansatzes zeigt sich in einer Folgebeauftragung bei der dieser Architekturansatz von uns erneut erfolgreich angewandt und weiter verbessert wird.

### 3. AUSFALLSICHERHEIT UND SKALIERBARKEIT DER VERWALTUNGSPLATTFORM DURCH CLOUD INFRASTRUKTUR

Die Microservices werden als ausführbare Datei in einem Docker Abbild (Image) abgelegt. Diese Images werden in der firmenintern bereitgestellten Enterprise Container Plattform „Openshift“ hinterlegt und dort in Containern ausgeführt.

Auch hier verfolgen wir den Ansatz „Infrastructure as code“. Das bedeutet, dass die Verwaltung und Bereitstellung von Containern und der Build-Pipeline durch maschinenlesbare Definitionsdateien geschehen. Diese werden wie normaler Programmcode in unserer Versionsverwaltung gespeichert und versioniert und somit problemlos wiederholbar und testbar.

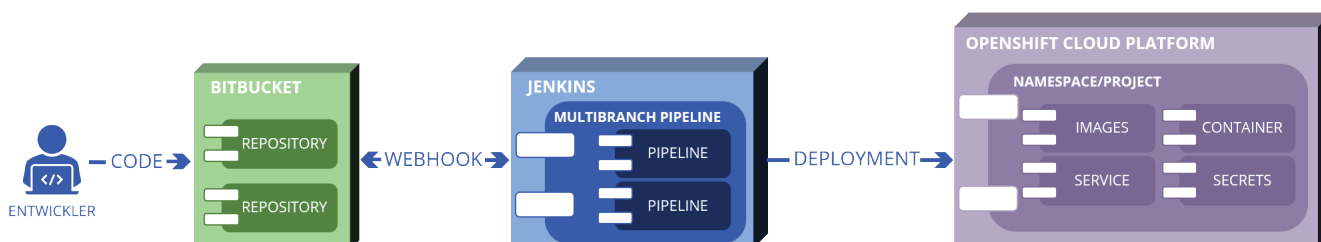
Durch die parallele Verwendung von redundanten Replikationen und der dynamischen Lastverteilung stellen wir sicher, dass die Produktivinstanzen zu jeder Zeit erreichbar sind.

Auch beim Ausrollen neuer Software-Releases entstehen durch das Blue-Green Deployment keine Ausfallzeiten.

### 4. HOHE CODEQUALITÄT DURCH CONTINUOUS INTEGRATION SICHERSTELLEN

Durch die konsequente Umsetzung von „Continuous Integration“, d.h. durch das Ausführen einer automatisierten Pipeline, vom Einpflegen des Programmiercodes eines Entwicklers bis hin zum automatischen Bereitstellen der Software in der Openshift Cloud Plattform, wird die Entwicklungszeit drastisch beschleunigt.

Für jeden Feature-Branch in Git erzeugt der Build-Server Jenkins automatisch eine neue Pipeline auf Basis eines vordefinierten Schemas. Bei deren Ausführung werden in Openshift, ebenfalls automatisiert, Images und andere Ressourcen nach vordefinierten Vorlagen erzeugt und abgelegt.



Die Jenkins-Pipelines beinhalten zum einen das automatische Ausführen von funktionalen Tests (z.B. Komponenten-Tests, End-to-end-Tests) in Frontend und Backend. Zum anderen stellen wir durch das automatisierte Evaluieren von syntaktischen und stilistischen Regeln sicher, dass nur lauffähiger und sauberer Code eingepflegt werden kann.

Durch Codereviews (Begutachtung durch Kollegen) in Bitbucket reduzieren wir die Wahrscheinlichkeit von Fehlern und nicht umgesetzten Anforderungen.

## 5. FLEXIBILITÄT UND WIEDERVERWENDBARKEIT IM FRONTEND

Auch im Frontend legen wir Wert auf moderne Technologien und Wiederverwendbarkeit.

Bei der Entwicklung des Webportals wurde von Beginn an auf „Responsive Web Design“ geachtet, d.h. auf Nutzbarkeit unabhängig von Bildschirmgröße und Gerät (PC, Smartphone, etc.). Dafür verwenden wir die mobile-first Frontend-Frameworks Bootstrap und Bootstrap Vue.

Mit dem „Vue Design System“ stellen wir eine wachsende Bibliothek an modularen Vue.js-Komponenten bereit, die von anderen Anwendungen genutzt werden. Somit ist sichergestellt, dass das Firmenbranding in allen auf Vue.js basierten Projekten korrekt umgesetzt wird.

Schließlich stellen wir durch End-to-end Tests die funktionale Korrektheit der grafischen Oberfläche in verschiedenen Browsern sicher.

## 6. IMPORT UND VALIDIERUNG VON DATENQUELLEN ZU VERTRIEBS- UND PRODUKT-INFORMATIONEN

Zur Bereitstellung von Dokumenten, Tools und anderen Ressourcen muss die Onlineplattform einige andere Dienste und Datenbanken ansprechen. Über automatisierte Importer in OpenShift werden Datenbankinhalte und Dateiformate in die Datenbank importiert und abgeglichen. Durch die Validierung stellen wir die Konsistenz der Daten sicher. Andere Systeme können diese Daten über REST-Schnittstellen konsumieren.

## 7. ÜBERWACHUNG UND ANALYSE DER MICROSERVICES IM LAUFENDEN BETRIEB

Natürlich können in jedem technischen System Fehler auftreten. Deswegen haben wir Warnverfahren im Einsatz, die regelmäßig alle Microservices und die Webseite überprüfen. Wird ein Problem detektiert, wird das Entwicklerteam per Email in Kenntnis gesetzt.

Mit Hilfe des Elastic Stacks (Elasticsearch, Logstash, Kibana) können wir alle Vorgänge innerhalb der Microservices an einer Stelle sammeln und analysieren. Das erlaubt uns eine schnelle Fehlerfindung und -behebung.

Für den Geschäftsbereich sind Kennzahlen wie die Anzahl der Nutzer und der registrierten Produkte von Relevanz. Die in Kibana erhobenen Zahlen visualisieren wir mit der Open Source Visualisierungssoftware Grafana.

## 8. FAZIT: VORTEILE UND HERAUSFORDERUNGEN VON MICROSERVICES

Natürlich bringen Microservices durch ihre indirekte Abhängigkeit zueinander auch neue Herausforderungen mit sich. So muss zum Beispiel sichergestellt werden, dass deren Schnittstellen zueinander konsistent bleiben. Das zentrale Logging und Monitoring der gesamten Microservice-Plattform hat sich bei der Fehlersuche und -behebung ausgezahlt.

Durch die Verwendung von Microservices und deren Bereitstellung in der Openshift Cloud konnten wir im Projekt schnell Hotfixes und dringliche Anforderungen ausrollen. Die konsequente Automatisierung der gesamten Pipeline trug maßgeblich zu dieser Geschwindigkeit und dem Erfolg der Vertriebs- und Verwaltungsplattformen bei.