

Software Architektur für Continuous Deployment: Merchant Onboarding

Studiengang: MAS Information Technology

Neue Anforderungen an Applikationen, welche zeitnah in Betrieb genommen werden müssen, gehört bei SIX zum Alltag. Viele der Aufgaben werden aufgrund von Richtlinien und Altsystemen manuell oder halb-automatisch durchgeführt. Anhand der Anwendung «Merchant Onboarding» wurde eine Software Architektur entwickelt, welche es erlaubt die Anwendung ohne Unterbruch des Dienstes vollautomatisch zu aktualisieren.

Ausgangslage

Als Finanzdienstleister ist SIX durch die aktuellen technologischen Fortschritte und die Anforderungen der Kunden in einem Spannungsfeld, welches eine kontinuierliche Aktualisierung von Applikationen unumgänglich macht. Das Ausrollen neuer Versionen nach drei Monaten entspricht nicht mehr den Kundenanforderungen.

Ziele

Das Ziel der Thesis war eine Software Architektur zu entwickeln welche den Anwendungsfall nicht ändert aber eine kontinuierliche Aktualisierung der Anwendung erlaubt. Die neue Architektur sollte in einem Software Architektur Dokument auf Basis der Arc42 Vorlage definiert und mit Prototypen verifiziert werden.

Vorgehen

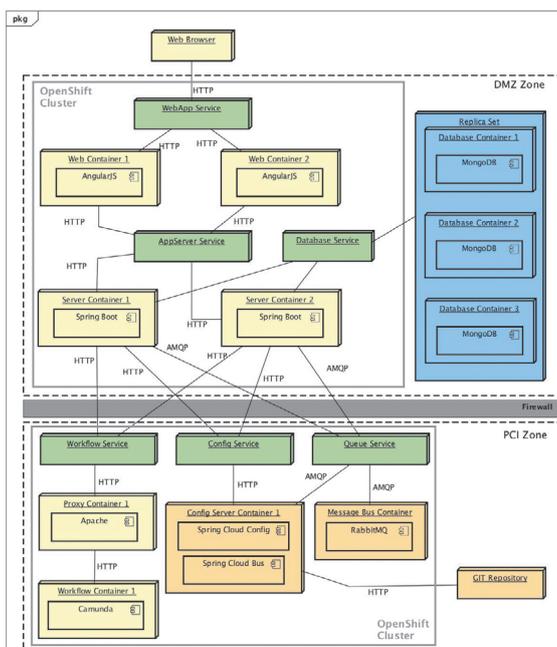
Zu Beginn wurden die neuen Anforderungen an die Software Architektur erfasst und in Qualitätsziele und passende Szenarien umgewandelt. Die dadurch entstandene Gesamtaufgabe wurde in Teilprobleme aufgeteilt und jeweils passende Lösungsvarianten gesucht. Dabei galt es eine grosse Vielfalt an Varianten zu suchen um bisher verwendete Praktiken zu überprüfen und gegebenenfalls zu ersetzen. Mittels einer Bewertungsmatrix wurden die einzelnen Teillösungen bewertet und bei Unklarheiten Prototypen erstellt um die Eignung zu verifizieren. Die gewonnenen Erkenntnisse konnten schlussendlich mit der bestehenden Architektur verknüpft und im Software Architektur Dokument definiert werden. Das Dokument gilt als Basis für einen vereinfachten finalen Prototyp, welcher die neuen Technologien und Komponenten enthält und mit welchem schlussendlich die Qualitätsziele verifiziert wurden.

Ergebnis

Durch die strukturiertere Herangehensweise und mittels der Arc42 Vorlage konnte das Problem optimal erfasst werden. Mithilfe moderner Technologien wie MongoDB, Docker, OpenShift und Spring Cloud Config konnte im Anschluss eine Architektur entworfen werden welche die gesetzten Ziele und Anforderungen erfüllt. Durch die generelle Problemlösung kann die Architektur oder Teile davon ebenfalls als Blaupause für andere Anwendungen verwendet werden.



Andreas Heubeck
Heuby@gmx.ch



Verteilungssicht der Applikation auf der OpenShift Container Plattform.