

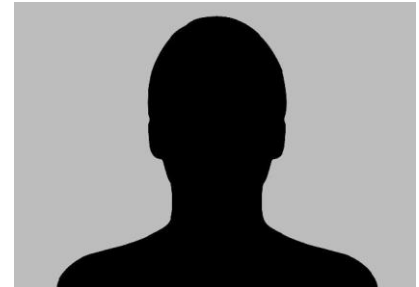
Modernisierung einer Wertschriftenhandelsplattform

Die AXA Schweiz betreibt eine Applikation namens «Vesba», eine Bankverwaltungssoftware für Freizügigkeits- und Vorsorgekonten. «Vesba» bietet die Möglichkeit, auf Konten automatisiert Wertschriften zu handeln. Bei «Vesba» handelt es sich um eine Applikation mit monolithischer Architektur, in welcher der Wertschriftenhandel direkt verbaut wurde und welche aus grösstenteils veralteten Technologien besteht. Mit diesen Voraussetzungen kann der Wertschriftenhandel nicht für andere Applikationen zur Verfügung gestellt werden. Ziel dieser Arbeit war das Herauslösen dieser Wertschriftenhandelslogik vom Monolithen «Vesba» zu einer eigenständigen Komponente, welche dadurch von weiteren Applikationen benutzt werden kann. Im Rahmen dieser Arbeit wurde die Machbarkeit einer solchen Lösung bestätigt. Durch die Neuimplementierung wurde eine Modernisierung der Software vorgenommen. So wurden aktuelle Technologien eingesetzt und auf begründete Designentscheide abgestimmt. Durch das gewählte Design können die geforderte Installation und der Betrieb in einer Cloud erfüllt werden.

Als Lösungsansatz wurden verschiedene unabhängige Softwarekomponenten neu implementiert. Mithilfe einer eventbasierten Architektur wurden diese unabhängigen Softwarekomponenten miteinander verbunden, was zu einer losen Kopplung führt, wodurch sie ohne Abhängigkeiten voneinander angepasst und verteilt werden können. Die Komponenten können in der Cloud einfach installiert und über Umgebungsvariablen konfiguriert werden, was einen Einsatz in diversen Cloud-Umgebungen erlaubt. Die Lösung dieser Arbeit wurde in einer Cloud-Plattform namens Kubernetes installiert und ist somit bereits gut vorbereitet, um in Zukunft direkt in der Cloud-Umgebung der AXA integriert zu werden.

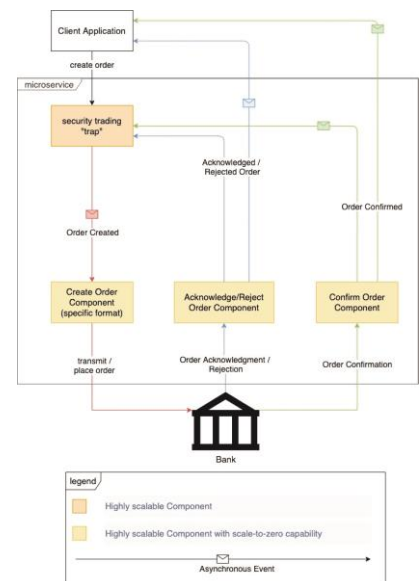
Die Installation auf dem Kubernetes-basierten OpenShift war eine Kernanforderung. Zusätzlich wurde auf die Serverless Technologie Knative und die Service-Mesh Technologie Istio gesetzt, zusammen mit Kafka als Eventbroker und Quarkus als Native kompilierbare Java Technologie.

Die verschiedenen Komponenten wurden als Serverless-Services implementiert. Der Einsatz von Serverless spart Ressourcen, indem Applikationen bei Nichtgebrauch heruntergefahren werden – sogenanntes «scale-to-zero». Quarkus ermöglicht den Start einer Java-Applikation innert weniger Millisekunden, um schnelle Antwortzeiten zu garantieren. Der Einsatz von Quarkus hat aufgrund von diversen Problemen des Kompilervorgangs und Testbarkeit nur bedingt überzeugt.



Diplomierende
Daniele Colombo
Matthias Gubler

Dozent
Walter Eich



Vereinfachter Aufbau der
Wertschriftenplattform