Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Informatik

Dokument-Tracking im Unternehmensumfeld

Themenbereiche: Softwareentwicklung, Monitoring

Studierender: Matthias Kafka

Dozent: Roland Gisler

Experte: Urs Zumstein

Wirtschaftspartner: Ajila AG

Keywords: Apache Kafka, Spring, Monitoring, Web-Application

1. Aufgabenstellung

Ziel dieser Arbeit ist es auf Basis von Java/Spring einen Service zu entwickeln, welcher es erlaubt den gesamten Ablauf der Dokumentbewirtschaftung möglichst automatisiert zu überwachen. Dazu sollen alle involvierten Zuliefersysteme über eine zu definierende Schnittstelle Statusmeldungen an den Service absetzen können. Diese werden vom Service analysiert, korreliert und persistiert. Werden Fehler oder Unregelmässigkeiten festgestellt, sollen automatisiert Fehlermeldungen erzeugt und an die zuständigen Personen (per Mail) oder direkt an/in das Ticket System übermittelt werden. Als grobe Priorisierung ist folgender Projektablauf vorgegeben:

1. Implementation des Service zum Entgegennehmen der Statusmeldungen via SOAP und REST inkl. Persistierung.
2. Erstellen einer simplen Abfrage-Schnittstelle via SOAP oder REST (z.B. "Liefere alle in den letzten 24h aufgetretenen Fehler").
3. Aufbau eines simplen Alertings bei Meldung von Fehlern durch die Zuliefersysteme.
4. Integration der Abfrage-Schnittstelle (2.) in ein simples Web-Frontend (mit Angular 2).
5. Optional: Ausbau der Benutzeroberfläche zur Anzeige eines graphischen Prozess-Schaubildes.

Sofern möglich und sinnvoll, kommt das firmeneigene «ajila Application Framework» zum Einsatz. Eine wichtige Anforderung an die Lösung ist eine maximale Performance. Dazu ist ein(e) geeignete(s) Architektur/Design zu wählen und die Performance mit Messungen in Form von automatisierten Tests zu verifizieren.

2. Ergebnisse

- Webapplikation zur Visualisierung von Fehlermeldungen und Dokumenterstellungs-prozessen.
- Spring Back-End zur Verarbeitung der Statusmeldungen.
- Testapplikation zur Simulation von Umsystemen (SAP, StreamServe, ...) für Performance Tests.

3. Lösungskonzept

Das Lösungskonzept basiert auf dem Technologiestack des Ajila Application Framework (Java/Spring/Angular). Um die Perfomanceansprüche zu erfüllen, wird Apache Kafka als Message Bus eingesetzt. Umsysteme können Statusmeldungen via REST oder SOAP an das Back-End absetzen.

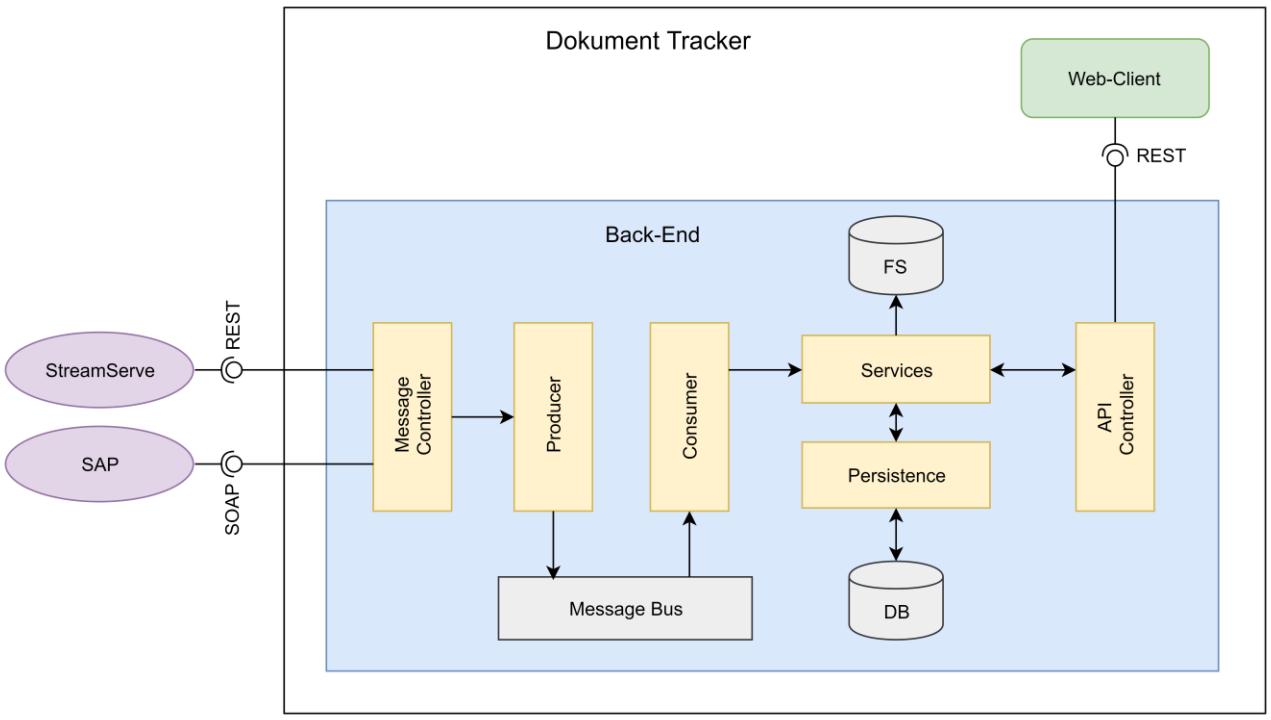

Abbildung 1: Softwarearchitektur

4. Spezielle Herausforderungen

Die Applikation stellt hohe Ansprüche an die Performance. Um das grosse Volumen von Statusmeldungen zuverlässig verarbeiten zu können wurde auf den Message Bus Apache Kafka gesetzt. Dies ermöglicht Statusmeldungen unabhängig von der Persistierung mittels Datenbank zu Verarbeiten. Die Anforderung 28 Messpunkte pro Sekunde verarbeiten zu können wurde mit einer rund 10-fachen Sicherheit erreicht. In Performance Messungen dauerte es Durchschnittlich 106 ms um 30 Messpunkte zu Verarbeiten.

Um einen beliebigen Prozess zu überwachen basiert das Dokument Tracking auf einer Prozess Definitionsdatei welche ein Prozess abbildet. Die Definition enthält alle Messpunkte eines Prozesses und definiert deren Reihenfolge.

5. Ausblick

Obwohl das Back-End ein grosses Volumen an Messpunkten verarbeiten kann, macht es Sinn das Back-End in ein verteiltes System aufzuteilen. Dies erfordert ein minimaler Eingriff in die Architektur welcher das alternative Deployment (Abbildung 2) ermöglicht. Neben der horizontalen Skalierung bringt dieses Deployment auch Vorteile bezüglich Wartung und Ausfallsicherheit mit sich. Eine weitere Verbesserungsmöglichkeit besteht bezüglich der Prozessdefinition mittels XML Datei. Anstatt sie als Datei zu hinterlegen könnte sie direkt über den Web-Client erstellt werden und so die Benutzerfreundlichkeit steigern. Bezüglich Ajila Application Framework besteht die Möglichkeit nachträglich die Registrierungs- und Authentifizierungskomponente einzubinden um die Applikation als Ajila Application Framework Addon in ein gemeinsames User-Interface mit anderen Addons zu

integrieren. Der Web-Client hat am meisten Ausbau Potenzial, da dieser bewusst einfach gehalten wurde um den Zeitrahmen dieser Arbeit nicht zu überschreiten.

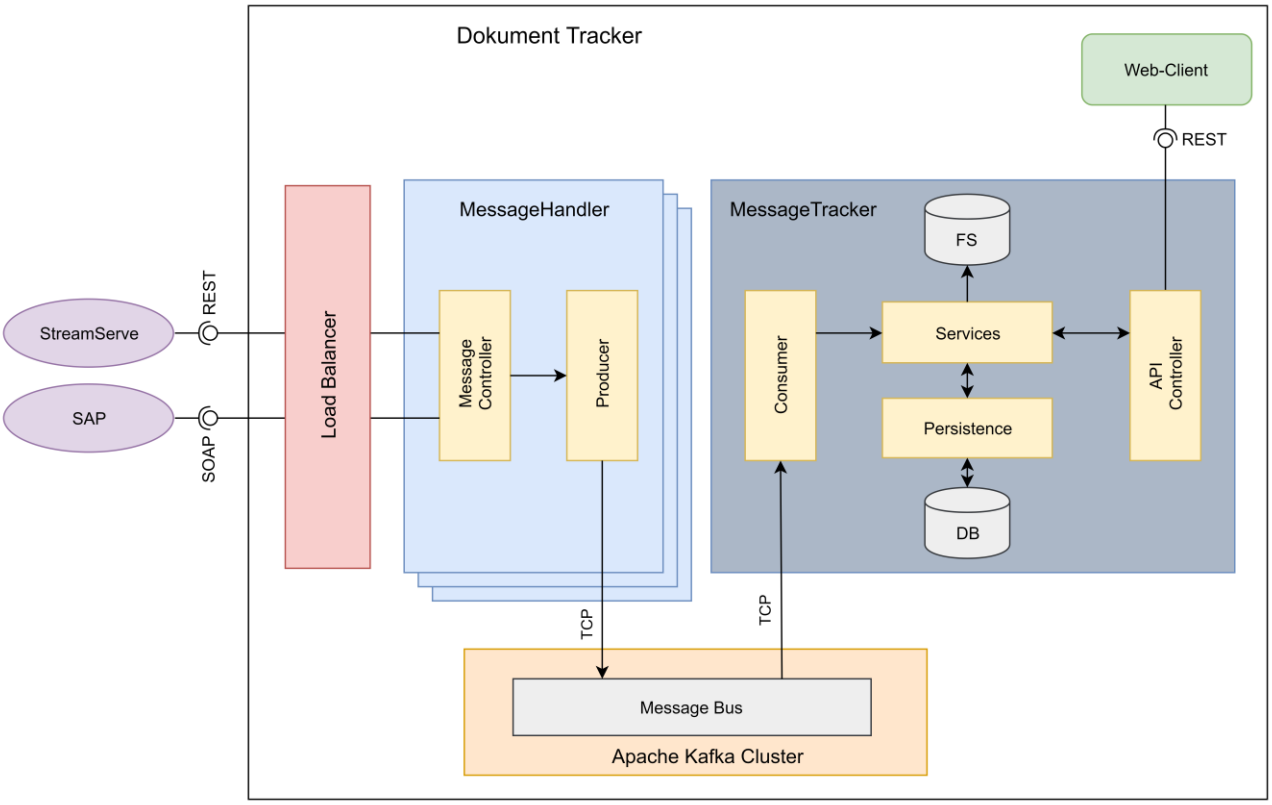

Abbildung 2: Alternatives Deployment