

Duale Hochschule Baden-Württemberg

Mannheim

**Portfolio**

**Studiengang Wirtschaftsinformatik**

Studienrichtung Software Engineering

Verfasser: Nele Ecker, Lars Langhammer, Mireille Puschmann, Jan Vögeli, Philip Wagner

Gruppe:

Kurs: WWI14 SE A

Studiengangsleiter: Prof. Dr.-Ing. Jörg Baumgart

Modul: Projekt

Lehrveranstaltung: Projekt I

Dozent: Tarek Auel

Abgabedatum: 20. November 2016

**Inhaltsverzeichnis**

[**1 Git Workflow** 1](#_Toc467337889)

[**2 Pipeline** 1](#_Toc467337890)

[2.1 Docker 1](#_Toc467337891)

[2.2 Taktstraße 1](#_Toc467337892)

[2.3 ActiveMQ 1](#_Toc467337893)

[2.4 Kafka 2](#_Toc467337894)

[2.5 Consumer 2](#_Toc467337895)

[2.6 Webserver 2](#_Toc467337896)

[2.7 Datenbank 2](#_Toc467337897)

[2.8 UI 4](#_Toc467337898)

[2.9 Spark 4](#_Toc467337899)

[**3 Anleitung für die Pipeline** 4](#_Toc467337900)

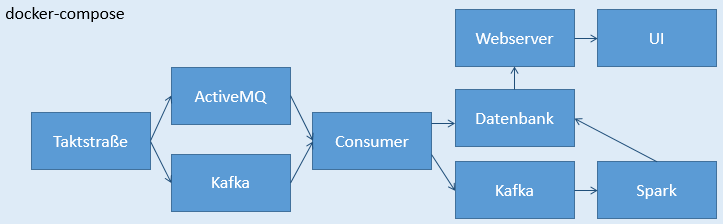
# **1 Git Workflow**

Es wurde sich entschieden GitHub zu verwenden. Das Branching Modell für GitHub ist das Feature Branching. Für das Feature Branching sprach, dass durch dieses Modell eine Trennung der einzelnen Feature vorgenommen wird. Zudem bietet dieses Modell eine bessere Übersichtlichkeit. Fertige Komponenten sollen in den Master Branch commitet werden.

Ein weiteres Modell wäre die Aufteilung der Branches nach Gruppenmitgliedern gewesen. Hierbei kommt das Problem auf, dass Gruppenmitgliedern an den gleichen Features wie zum Beispiel den Docker oder der Datenbank gearbeitet haben.

Link zum Git-Repository:

# **2 Pipeline**



## 2.1 Docker

Für die Pipeline gibt es Docker für die Taktstraße, für ActiveMQ, für die Datenbank, für das UI, für Kafka, für den Consumer, für Spark und für den WebServer.

## 2.2 Taktstraße

De Taktstraße ist eine Applikation, welche die Produktionsdaten für Kafka liefert. Diese Applikation wurde von Professor Reichwald zur Verfügung gestellt.

## 2.3 ActiveMQ

## 2.4 Kafka

## 2.5 Consumer

## 2.6 Webserver

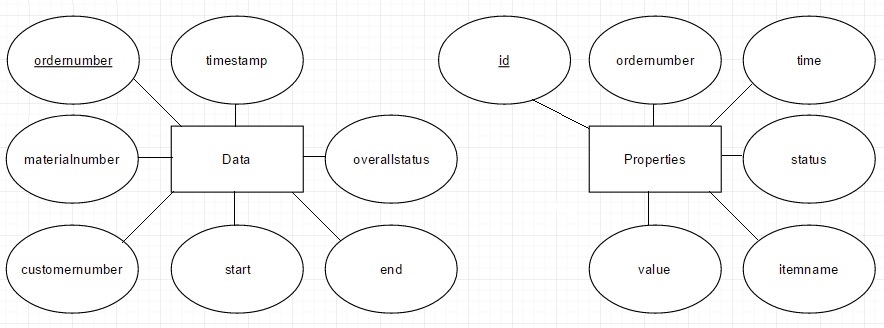
## 2.7 Datenbank

Die Datenbank basiert aus PostgreSQL. Für die Verwendung von PostgreSQL sprach, dass PostgreSQL eine relationale Datenbank ist. Durch die relationale Datenbank ist eine leichte Aggregation der Daten möglich. Zusätzlich werden Datenredundanzen verhindert. Ein weiteres Argument für die relationalen Datenbank war, dass diese bereits in den Vorlesungen besprochen wurden. Analysen werden über SQL Abfragen vorgenommen.

In der Datenbank werden zwei Datenbanktabellen erzeugt.

Die Tabelle Data enthält Daten von den ERP-Daten und den ActiveMQ-Daten.

Die Tabelle Properties enthält Daten aus Kafka. Im Anhang



## 2.8 UI

Die grafische Oberfläche basiert auf OpenUI 5. Für OpenUI 5 sprach, dass es ein einfaches Grundgerüst besitzt. Zudem sind leichte Anpassungen an dem UI möglich. Zusätzlich wurde die c3.js-Bibliothek, die auf d3.js basiert eingebunden. Probleme bei OpenUI 5 waren, dass viele Eigenkonfigurationen vorgenommen werden mussten, welche einen hohen Aufwand darstellten.

## 2.9 Spark

# **3 Anleitung für die Pipeline**