Real-Time Production Monitoring

Von

Roland Bauer
Dominik Prüll
Sebastian Weidele
Oliver Tomondy

in
Big Data Infrastructure

am
FH Technikum Wien

Sommersemester 2021

16.6.2021

Inhaltsverzeichnis

Einleitung

Ein Betrieb, der der Automobilindustrie zuliefert, steht unter ständigem Lieferdruck.
Alles muss just in time produziert und geliefert werden. Unvorhergesehene Maschinen-Stillstände müssen bestmöglich vermieden werden.
Bis jetzt wurde dies durch präventive Wartung bzw. Tausch von Komponenten erreicht.
Dies ist allerdings nicht zuverlässig und außerdem sehr teuer.
Des Weiteren hat das Unternehmen derzeit keine Möglichkeit, den Status der Geräte zentral zu überwachen.
Nun sollen alle relevanten Prozessdaten aller Elektroantriebe lückenlos aufgezeichnet und auf Ausreißer überwacht werden. Die Prozessdaten werden übersichtlich auf einem Dashboard dargestellt.

Tritt gegebenenfalls ein Ausreißer auf, so wird dies durch einen E-Mail-Alert signalisiert.
Dadurch hat das Wartungspersonal die Möglichkeit, Wartungen und den Tausch von Komponenten gezielt und nur dann, wenn es tatsächlich notwendig ist, durchzuführen.

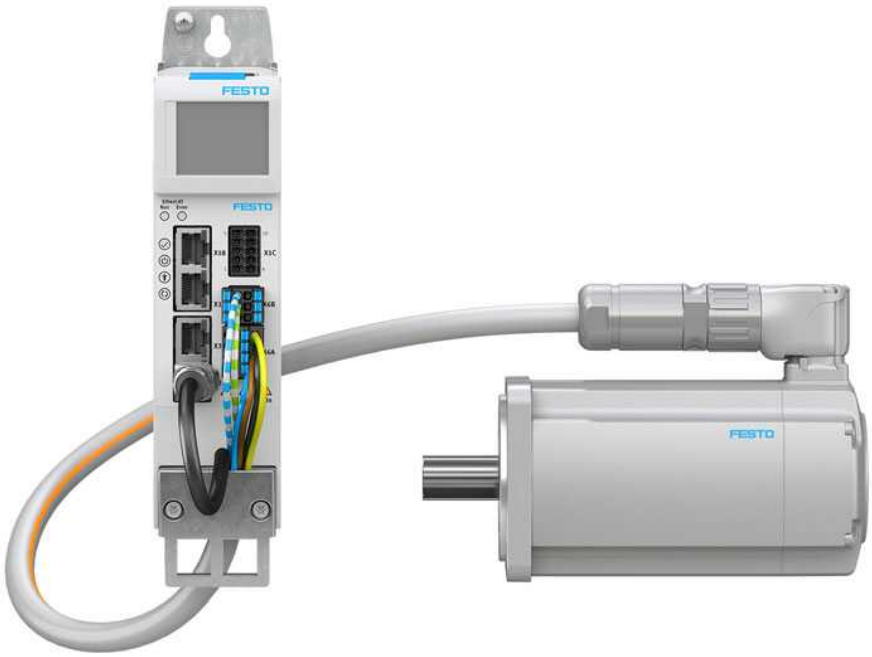

Abbildung 1 Elektromotor

Use Cases

Die kritischen Metriken der Motoren sollen übersichtlich auf einem Dashboard für den Benutzer dargestellt werden, um ein frühzeitiges Eingreifen im Fehlerfall zu ermöglichen. Dieses wird zusätzlich über ein Alerting bei Unter-/Überschreiten von Schwellwerten unterstützt.
Es soll Filtermöglichkeiten geben, um auf die einzelnen Ebenen herunterbrechen zu können. Die Ebenen spiegeln die räumliche Verteilung der Motoren wider:
Motor, Halle, Produktions-Unternehmen.

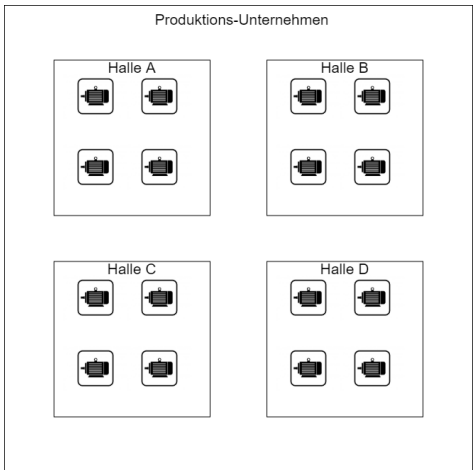

Abbildung 2 Schematische Darstellung der Produktionsanlage

Visualisierung in Superset

Im Folgenden werden die Anforderungen an das Dashboard in Superset aufgelistet:

- Überblick über Temperatur/Voltage der Motoren global, pro Halle und pro Motor
- Alerting: Bei Überschreitung der Grenzwerte soll eine E-Mail geschickt werden, die das aktuelle Dashboard beinhaltet
- Auslastung der Hallen (State der Motoren)

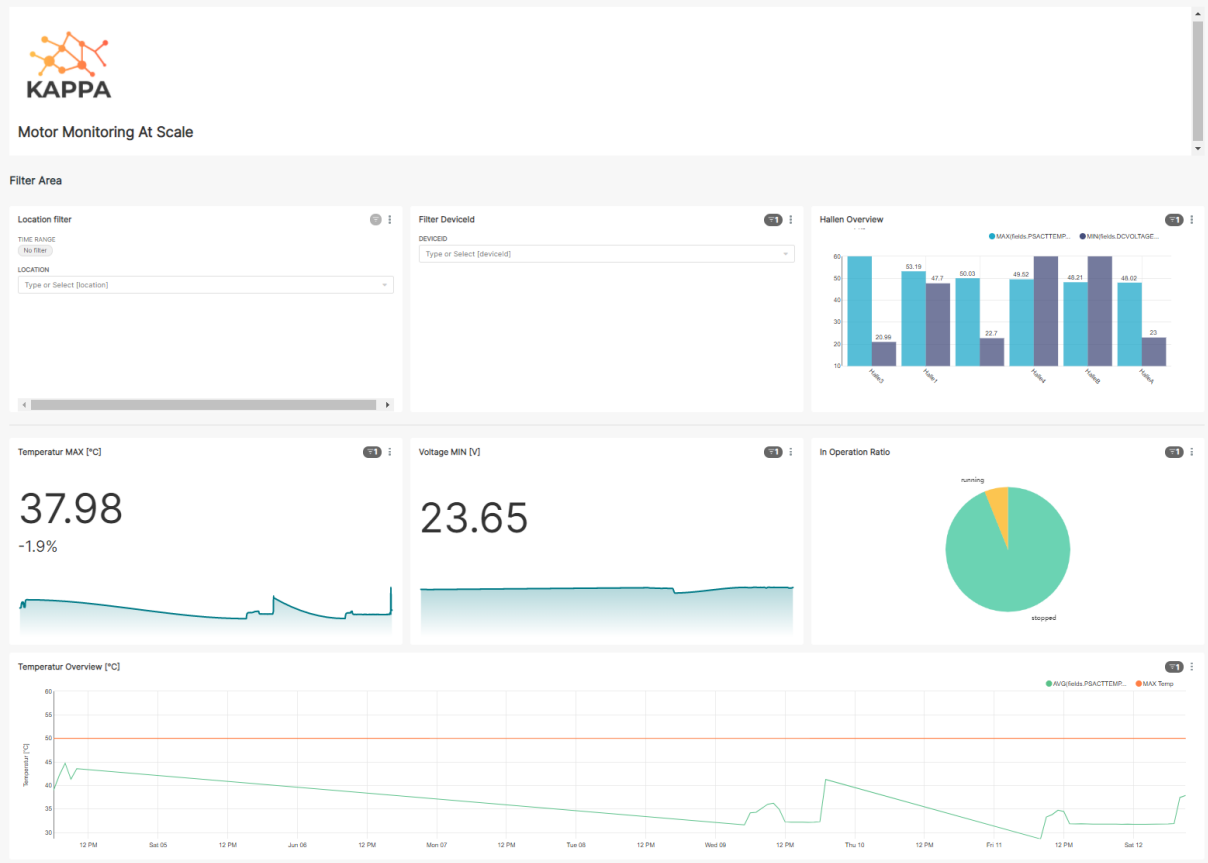

Abbildung 3 Superset Dashboard

Detailbeschreibungen der Charts

Alle folgenden Charts können dynamisch auf folgende Kriterien gefiltert werden:

- Zeitraum
- Standort (Location) - entspricht der jeweiligen Halle
- Motor (DeviceId)

Hallen Overview

Dieses Chart zeigt eine Übersicht über die minimale Spannung und die maximale Temperatur, gruppiert nach Standort bzw. nach Filtereinstellung.

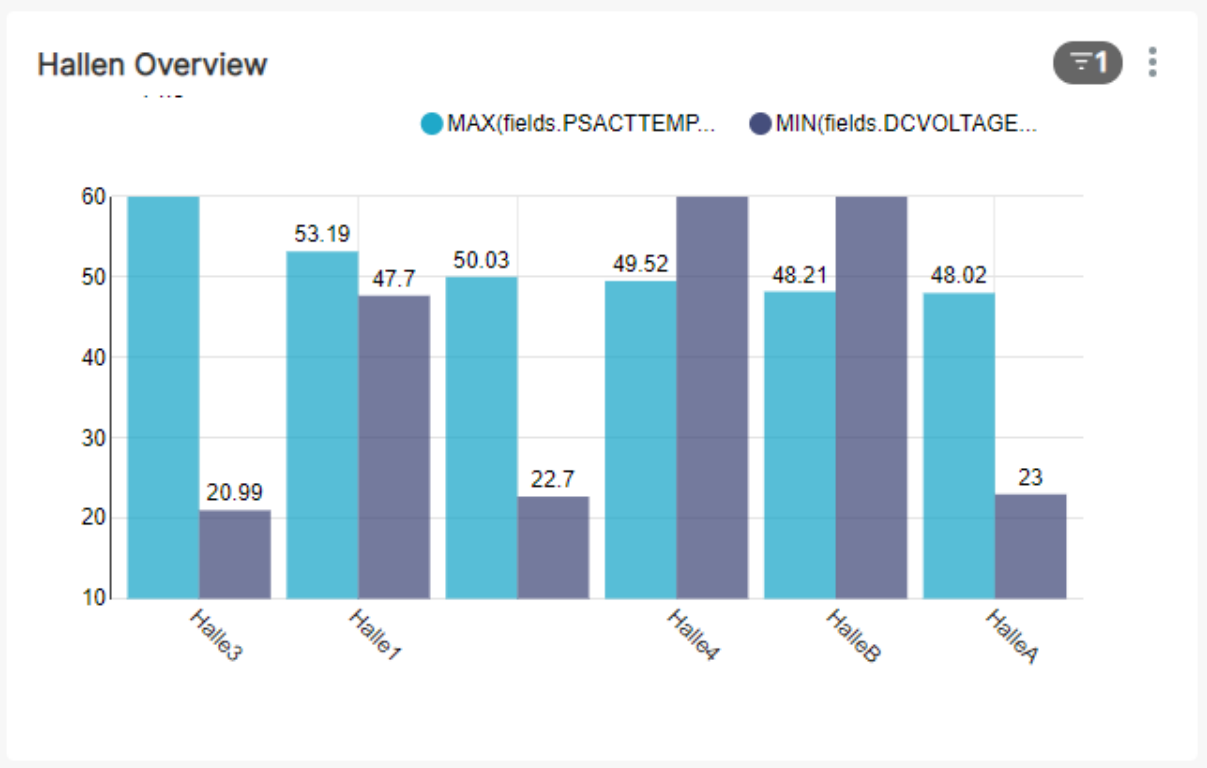

Abbildung 4 Chart Hallen Overview

Chart Spannung

Dieses Chart zeigt die minimale Spannung aller Motoren sowie den zeitlichen Verlauf, in Abhängigkeit der gesetzten Filter. Damit können Netzschwankungen erkannt werden. Zu große Netzschwankungen können sich negativ auf die Motoren und deren Leistung auswirken.

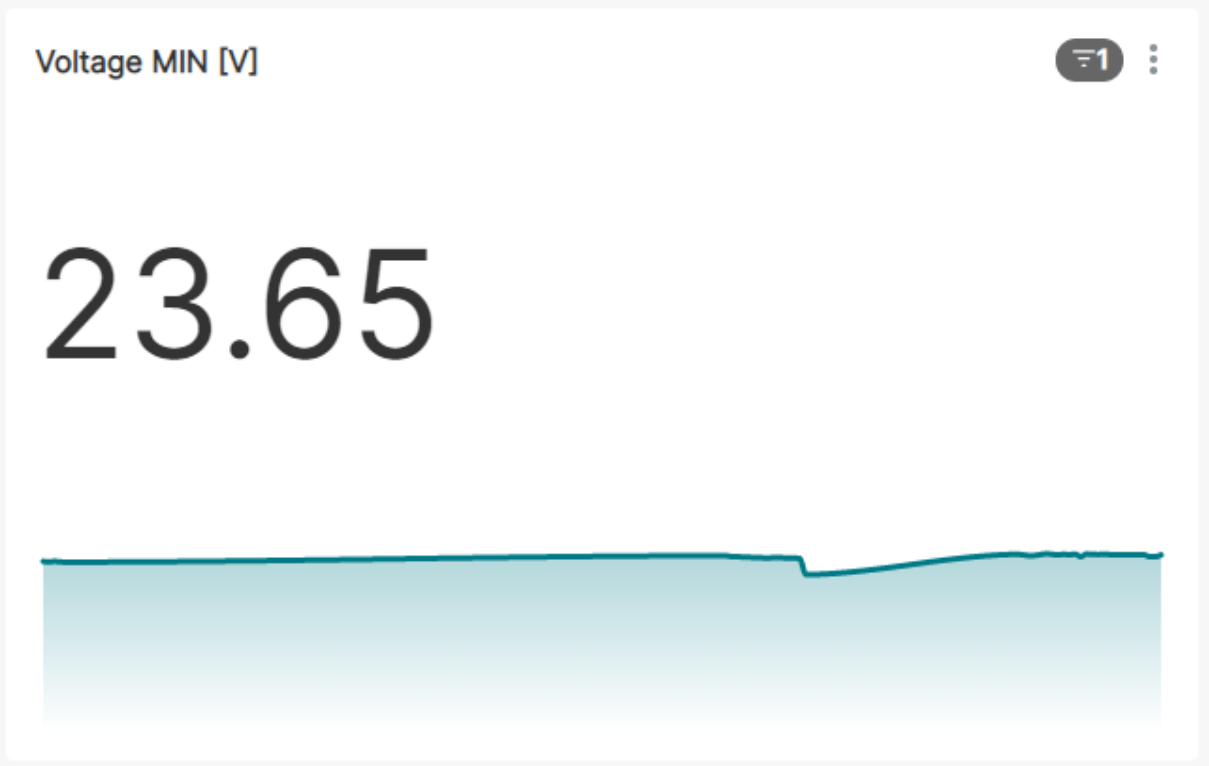

Abbildung 5 Chart Voltage MIN

In Operation Ratio

Dieses Chart zeigt die Auslastung der Motoren, in Abhängigkeit der gesetzten Filter und ermöglicht damit einen raschen Überblick über die Produktivität aller Motoren. Bei zu geringem “In Operation Ratio” kann auf zu viele Anlagenstillstände rückgeschlossen und entsprechende Gegenmaßnahmen können eingeleitet werden.

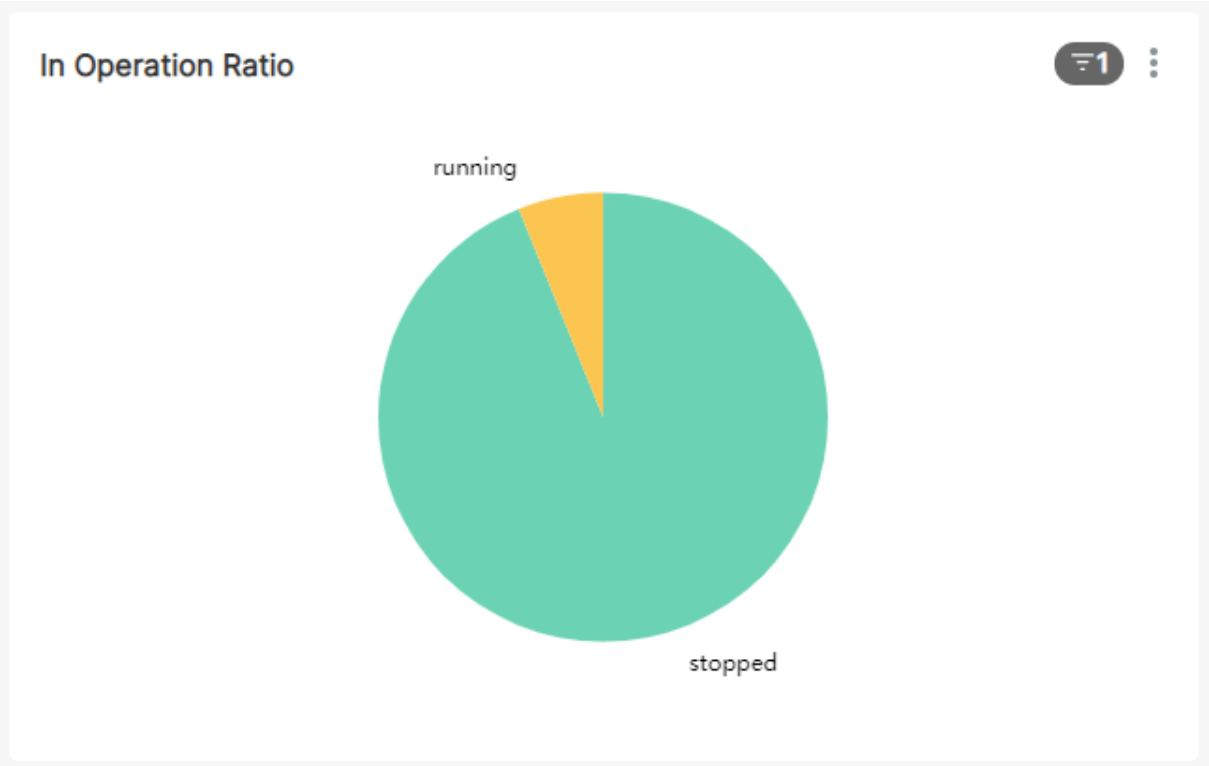

Abbildung 6 Chart In Operation Ratio

Temperature Overview

Dieses Chart zeigt einen Überblick über die durchschnittliche Temperatur der Motoren, in Abhängigkeit der gesetzten Filter. Zusätzlich wird der obere Temperaturgrenzwert dargestellt.
Damit erhält der Bediener einen guten Überblick über die Leistungsreserven und mögliche Überlastungs-Szenarien der Motoren. Bei Bedarf können wiederum geeignete Gegenmaßnahmen eingeleitet werden.

Abbildung 7 Chart Temperature Overview

Architektur

Nachfolgend werden die Daten und die Architektur beschrieben.

Daten

Alle Motoren haben standardmäßig Sensoren, die Informationen über Temperatur, Drehmoment, Spannung und Status erfassen und diese über die integrierte Ethernet Schnittstelle bereitstellen. Übergeordnet besitzt jede Maschine ein IOT-Gateway, welches die Daten sammelt und an Kafka weiterleitet. Die Abtastrate der Daten beträgt 500 ms je Motor.

Im Folgenden sind die einzelnen Metriken der JSON-Message tabellarisch dargestellt:

Metrik	**Type**	**Beschreibung**
time	DATETIME	Der genaue Zeitpunkt der Messung
device_id	STRING	Die ID des Geräts
location	STRING	Standort des Motors
messageType	STRING	Gibt an, ob es sich um Prozess-, Asset- oder Diagnosedaten handelt. Es wurden nur die Prozessdaten übertragen.
fields	OBJECT	Enthält Informationen aus den einzelnen Sensoren
↳ DCVOLTAGE	FLOAT	Spannung des internen Last-Zwischenkreises.
↳ PSACTEMP	FLOAT	Die Temperatur des Geräts
↳ STATE	INTEGER	Der Status des Geräts
↳ TORQUEMOTOR	FLOAT	Das Drehmoment des Geräts

Tabelle 1 Daten

Diese Daten müssen gespeichert werden, um rückblickend die Leistung der Motoren analysieren zu können und in Echtzeit visualisiert werden, um der Production-Monitoring Abteilung die Überwachung der Antriebe zu vereinfachen.

Data Flow im IOT-Gateway

Die Daten werden vom Motor an einen MQTT-Broker geschickt. Um die Daten an den Kafka-Broker weiterzuleiten, wurde in Node-Red folgender Flow erstellt:

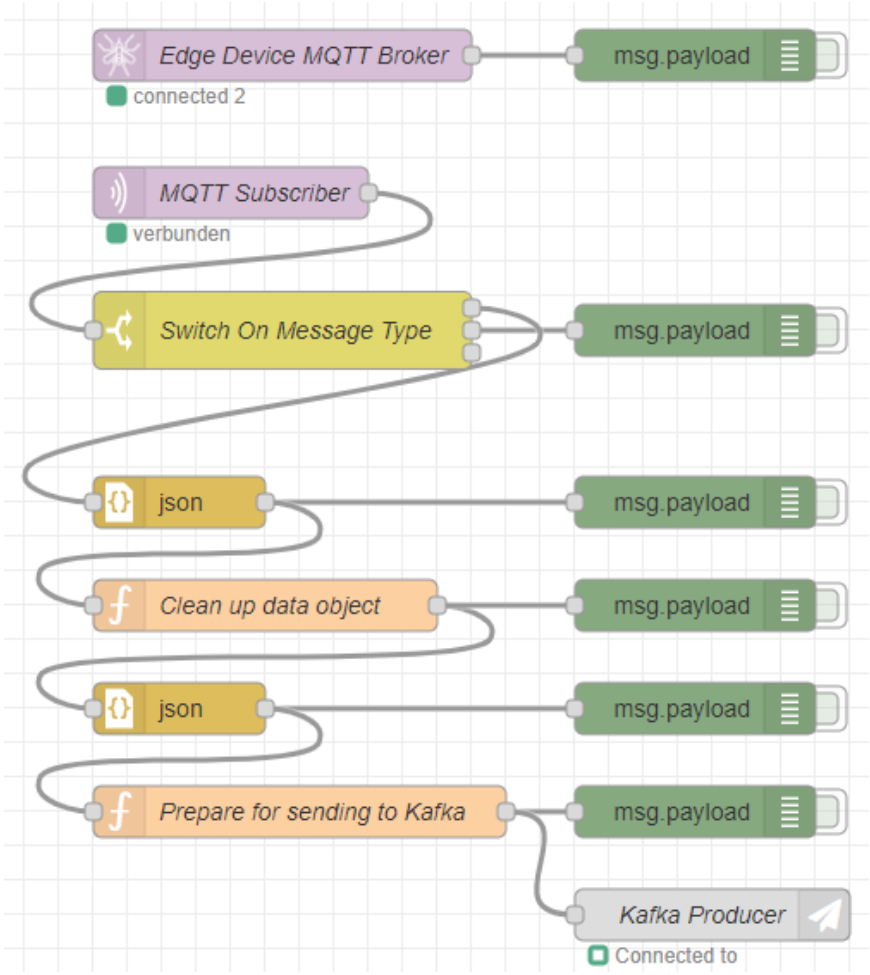

Abbildung 8 Node-Red Dataflow

Wahl der Technologien und Architektur

Für diese Anforderungen haben wir die nachfolgende Architektur entworfen.
Die Daten aller Motoren aus der gesamten Produktion werden an den Kafka Broker gesendet. Mittels Druid werden die Daten aus Kafka abgegriffen und persistiert. Abschließend stellt Druid Superset die Daten zur Visualisierung zur Verfügung. In Superset wurden die entsprechenden Charts, das Dashboard, die Alerts und Reports konfiguriert, um sie dem Benutzer bereitzustellen.

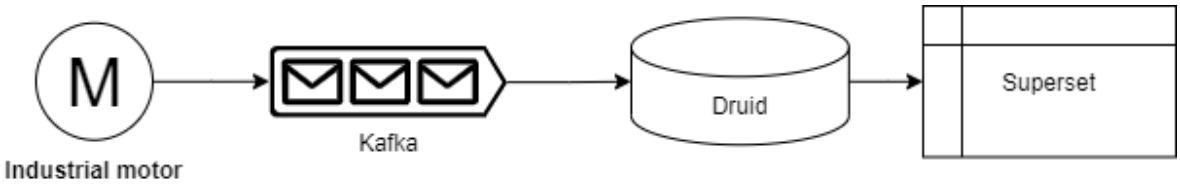

Abbildung 9 Architektur

Bei der Wahl der Technologien wurden ausschließlich Open Source Lösungen in Betracht gezogen deren Komponenten miteinander gut integrierbar sind. Nachfolgende Kriterien waren generell für die Auswahl auschlaggebend:

- Open Source
- Kostenlos
- Skalierbarkeit

Im Folgenden werden die Hauptauswahlgründe für die einzelnen Technologien dargestellt.

Apache Kafka

- Der Datendurchsatz ist größer als bei Pulsar (2x) oder RabbitMQ (15x)
- Geringste Latenz beim größten Datendurchsatz
- Empfohlene max. Message Größe (1 MB) absolut ausreichend
- Größte Community und beste Dokumentation

Apache Druid

- Real Time Aggregations
- Geringe Latenz
- Möglichkeit kostengünstig Daten in Data Lakes (HDFS etc.) zu persistieren
- Bietet OLAP für zukünftige Weiterentwicklung im BI-Bereich

Apache Superset

- Gute Anbindung an Druid
- Möglichkeiten für Alerts + Reports (E-Mail und Slack)
- Einfaches Setup
- Viele mitgelieferte Chart-Templates
- Lightweight semantic layer (z.B. calculated columns und Metriken)

Umsetzung

Google Cloud Computing

Um einen funktionierenden Prototypen bereitzustellen, wurde auf Google Cloud Computing gesetzt. Dies hat Vorteile bzgl. Performance, da Apache Druid sehr ressourcenintensiv ist. Außerdem erleichterte es die Arbeitsverteilung im Team.

Ubuntu VM

Für die Umsetzung wurde eine Ubuntu VM mit folgenden Performance Parametern verwendet:

- 6 vCPUs
- 32 GB RAM
- 50 GB SSD
-

Docker

Die einzelnen Komponenten, Kafka, Druid und Superset, wurden in einzelnen Docker Containern aufgesetzt. Dieser Ansatz unterstützt die Portierbarkeit in ein Produktivsystem und etwaige Skalierung.

Netzwerk

Um Konfigurationsänderungen minimal zu halten, wurde für die VM außerdem eine statische IP-Adresse gebucht.

Lessons Learned

- Weniger ist oft mehr:
    - Der ursprünglich geplante Einsatz von Spark wurde aufgrund von Features anderer Komponenten verworfen. So können bspw. direkt in Superset “Calculated Columns” implementiert werden.
    - Dies führt zu einer schlanken Architektur und damit weniger Wartungsaufwand.

- Richtige Auswahl der Technologie ist entscheidend und ein Umstieg auf eine andere Technologie sehr ressourcenintensiv. Dadurch wird die Konfiguration der einzelnen Komponenten minimal gehalten.
- Trotz der zahlreich enthaltenen Charts ist Customizing in Superset schwierig umsetzbar
- Druid ist ressourcenintensiv, was ein lokales Docker Setup erschwert.

Abbildungsverzeichnis

Tabellenverzeichnis