



Übersicht

Overview

Project

Id 202324-CF-Networkanalysis

Name Visualisierung der Ergebnisse der Netzdatenmodellanalyse

Team

Jürgen Katzenschlager Project manager

Clemens Schlipfinger Backend programmer

Felix Schneider Frontend programmer

Version History

Version	Datum	AutorIn	Änderungen
0.1	22.08.2023	Felix Schneider	Erstellung des Dokuments

Inhaltsverzeichnis

[Table of Contents]

Inhaltsverzeichnis.....	1
Titel: Visualisierung der Netzdatenmodellanalyse.....	2
BetreuerIn.....	2
Thema.....	2
Ausgangslage.....	2
Projektteam.....	2
Projektpartner.....	2
Aufgabenbereiche.....	3
Allgemein.....	3
Backend.....	3
Frontend.....	3
Übersicht der Applikation.....	5
Meilensteine.....	5
Aufgaben - TODOs.....	6

Titel: Visualisierung der Netzdatenmodellanalyse

BetreuerIn

Hauptverantwortlicher: Jürgen Katzenschlager

Thema

Bei einer Zustandsestimation von dem Stromnetz werden die Daten nach verschiedenen Kriterien überprüft und Ergebnisse erstellt. Diese Ergebnisse sollen in einer Webapplikation in einer übersichtlichen Weise dargestellt werden.

Ausgangslage

Es gibt bereits automatisierte Netzmodellanalysen (Static Data Checks), die ihre Ergebnisse (Findings) in ein Logfile schreiben. Dieses ist aber durch Menschen nicht leicht lesbar. Dieses Problem soll durch die Applikation erleichtert werden.

Die bereits im Finding (Java Objekt) vorliegende Informationen müssen in einem zu programmierenden Java Backend einem ebenfalls zu programmierenden Angular Frontend zur Verfügung gestellt werden.

Projektteam

Arbeitsaufwand

Name	Individuelle Themenstellung	Klasse	Arbeitsaufwand
Felix Schneider	Visualisierungsmethoden für stark vernetzte Daten	5AHIT2324	180h
Clemens Schlipfing	Vergleich der verschiedenen Formen der Messagepropagation in Enterprise Service Buses	5AHIT2324	180h

Projektpartner

Siemens AG

Adresse: Siemensstraße 90, 1211 Wien

Webseite: [Siemens Österreich](#)

Aufgabenbereiche

Allgemein [General]

Die Applikation wird durch **Git** verwaltet. Dieses Programm kann die verschiedenen Versionen und Weiterentwicklungen leicht kontrollieren und den Verlust des Codes gegen null minimieren. Voraussichtlich wird ein **Repository** auf GitLab (oder GitHub) erstellt werden, dass den Programmierer:innen einen freien Spielraum zum Entwickeln zur Verfügung stellt.

Schlussendlich soll die Applikation in einem **Docker** Container rennen, damit sichergestellt ist, dass sie auf allen möglichen (Docker unterstützenden) Geräten ausführbar ist. Hierbei können die Programmierer:innen auch Continuous Integration mithilfe von GitLab Pipelines (oder GitHub Actions) umsetzen, sodass automatisch ein Docker Image beim Pushen auf den Main / Master Branch auf ein Docker Repository gepusht wird.

Backend Clemens Schlipfinger

Die Applikation erhält die Daten über **einen Messagebus** wie zum Beispiel Kafka. Das Backend muss mit einem bestehenden Message Bus System integriert werden können. Während des Entwicklungsprozess soll ein **Message Bus System für die Simulation** der Daten entstehen und für das finale Produkt soll durch ein **Module** die interne Application von Siemens mit einem Messagebus verbunden werden. Die Applikation soll jederzeit auf neue Updates reagieren können und diese an Clients propagieren können (**Server-Side PushUpdates**).

Für die **Persistierung** der Daten in einer relationalen Datenbank kann **PostgreSQL** oder die **Oracle** Datenbank verwendet werden.

Damit das Backend mit dem Frontend kommunizieren kann, wird eine **API** mit effizienter Datenübertragung und **Authentifizierung** implementiert. Hierbei soll der bestmögliche API-Type (das Konzept, welches für die API verwendet wird) evaluiert werden. Es gibt verschiedene Möglichkeiten, wie zum Beispiel **REST, GRPC, GraphQL** und vieles mehr.

Das Spring Framework wird verwendet, um die Java Server Applikation aufzusetzen.

Frontend

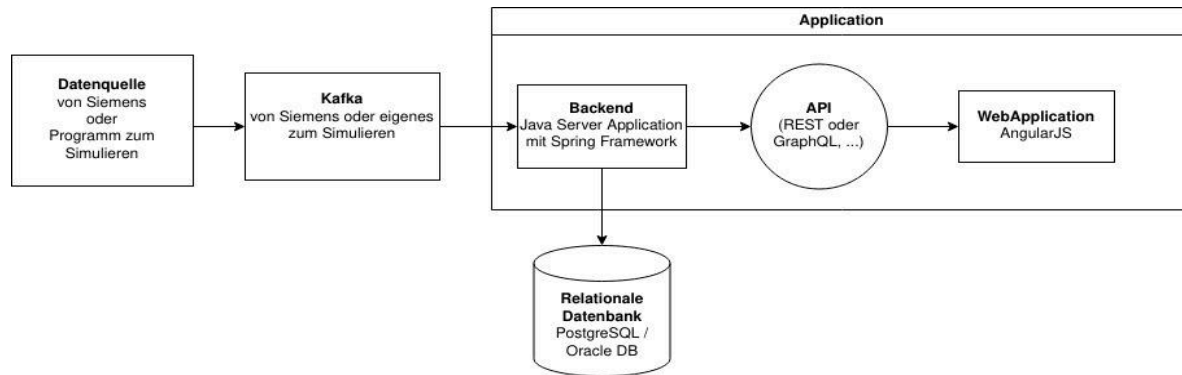
Felix Schneider

Das Frontend, welches die Daten über die **API** erhält, soll diese übersichtlich darstellen und eine einfache Filterung ermöglichen. Hierfür werden optimale **Darstellungsarten**, wie zum Beispiel **Diagramme, Graphen** und **Matrizen**, evaluiert. Es ist wichtig, dass diese Darstellungsarten dem **Benutzer helfen**, die Fehlerquelle zu finden und zu reparieren.

Bei **Datenupdates** vom Server soll das Frontend darauf reagieren und entsprechende Grafiken und Suchergebnisse aktualisieren. Bei allen diesbezüglichen API-Abläufen muss die **Authentifizierung** sichergestellt werden.

AngularJS wird verwendet, um das Frontend effizient aufzusetzen.

Übersicht der Applikation




Meilensteine

Das Projekt enthält einige Meilensteine:

- Projektstart
- Pflichtenheft erstellt
- Recherche
- Empirischer Teil
- Code Freeze
- Bewertung
- Projektpräsentation
- Projektende

Waterfall, Tests, Projektstart, Pflichtenheft erstellen, Systemdesign, Implementierung, Projektende

Aufgaben - TODOs

- Erhalt vom Lastenheft möglicherweise Anfang August - 
- Erstellung von einem Pflichtenheft
- Diplomarbeit in die Diplomarbeitsdatenbank eintragen