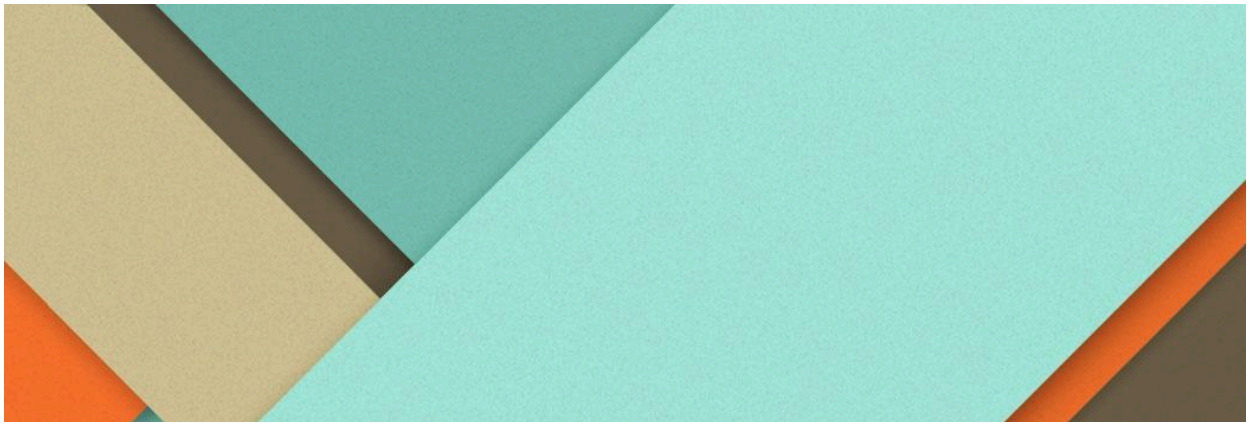


Pflichtenheft

Requirements Specification

—

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	17.08.2023	Felix Schneider	Erstellung des Dokuments
0.2	22.08.2023	Felix Schneider	Hinzufügen einiger Ziele
0.3	23.08.2023	Felix Schneider	Ziele schreiben [überarbeitungswürdig]
0.4	10.10.2023	Felix Schneider	Funktionale Anforderungen Grafik
0.5	11.10.2023	Clemens Schlipfinger	Systemarchitektur Grafik hinzufügen
0.5	25.10.2023	Felix Schneider	Mockup hinzufügen
0.6	25.10.2023	Felix Schneider	Nicht Funktionale Anforderungen anfangen
0.7	26.10.2023	Felix Schneider	Überarbeitung Ziele Frontend, technische Anforderungen
0.8	27.10.2023	Clemens Schlipfinger	Hinzufügen Backend Ziele und Anforderungen
0.9	30.10.2023	Felix & Clemens	Überarbeitung mit Siemens

Inhaltsverzeichnis

[Table of Contents]

Ziele

[Goals]

MUSS-Ziele

Die Muss-Ziele definieren klar, welche Anforderungen unbedingt erfüllt sein müssen. Alle Ziele sind bis **12.04.2024** zu erfüllen.

- ☐ Backend
  - ☑ ~~Der Backend-Server soll die Daten (Findings) persistieren und für die Webanwendung in einer sinnvollen Art zur Verfügung stellen.~~
  - ☐ Der Backend-Server soll in der Lage sein, gleichzeitig Daten von verschiedenen Netzmodellanalysen zu verarbeiten.
  - ☐ Der Backend-Server soll sich leicht in das bestehende System integrieren und als optionale Erweiterung des Systems gelten.
  - ☐ Der Backend-Server soll asynchron funktionieren, welches eine gleichzeitige Verarbeitung und Bereitstellung der Daten ermöglicht.
- ☐ Frontend
  - ☑ ~~Die Webanwendung bietet umfangreiche Suchmöglichkeiten in tabellarischer Form, nach den Kriterien Schweregrad, Kategorie, Spannungsebene, ID des Findings und der technischen Adresse des Findings..~~
  - ☐ Die Webanwendung visualisiert Relationen im Stromnetzwerk mittels Graphen.
  - ☐ Die Webanwendung stellt Informationen übersichtlich in einem Dashboard dar.

KANN-Ziele

Die Kann-Ziele sind optional.

- ☐ Backend
  - ☐ Das Backend soll auch den Client über neue Daten informieren.
  - ☐ Die API verfügt über eine Authentifizierung mittels Siemens Integration.
- ☐ Frontend
  - ☐ Die Webanwendung stellt Zusammenhänge in Diagrammen dar.

NICHT-Ziele

Die Nicht-Ziele sind explizit nicht zu erfüllen. Um Verneinungen zu vermeiden, sind diese trotzdem so formuliert, als würden sie erfüllt werden müssen.

- ☐ Die Fehler des Netzmodells sollen erkannt und in die Findings (Java Objekt) gespeichert werden.
- ☐ Es wird im bestehenden Repository von Siemens gearbeitet.
- ☐ Die Applikation wird auch für mobile Geräte entwickelt.

Funktionale Anforderungen

[Functional Requirements]

Anforderungen an das Backend

Das Backend soll die Daten der Netzmodellfehleranalyse verarbeiten. Die Daten sind die Findings, welche die Ergebnisse der Netzmodellfehleranalyse sind.

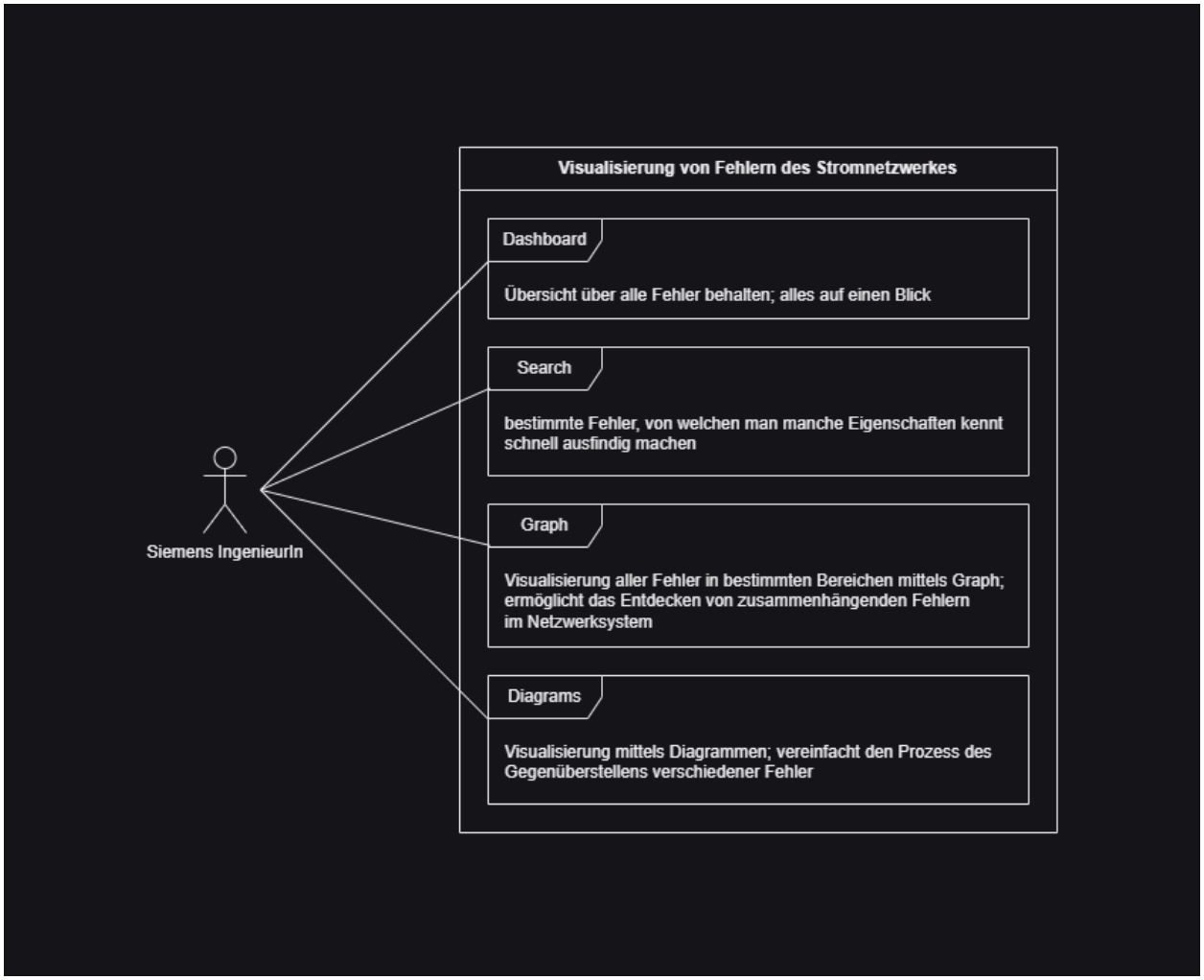
- Es müssen die Daten mit Hilfe einer Datenbank **persistiert** werden und bis zur nächsten Netzmodellfehleranalyse gehalten werden.
- Die Findings müssen durch eine **Schnittstelle (GraphQL) dem Frontend** angeboten werden.
- Der Backend soll sich als **optionale Erweiterung** leicht in das bisherige System integrieren lassen. Zusätzlich soll es nur kleine Veränderungen am Haupt-System erzwingen.
    - Für den Entwicklungsprozess soll eine **Simulationssoftware** verwendet werden.
    - Es kann davon ausgegangen werden, dass ein Kafka-System und eine Datenbank schon bestehen. Das Programm, welches die Daten in das Kafka Topic schreibt, wird von Siemens bereitgestellt.
- Das Backend soll **verschiedene Netzmodellfehleranalyse** verarbeiten können, das heißt, es muss Findings von mehreren Exportmodulen gleichzeitig entgegennehmen können.

Optionale Ziele

- Es kann eine Authorization für das Einschränken des Zugriffs auf die Schnittstelle zum Frontend geben.
- Der Backend-Server soll über die Funktionalität verfügen, den Client mit **Server-Side Push Updates** über den Eingang von neuen Daten zu informieren.

Anforderungen des Frontends

Dieses Use Case Diagramm veranschaulicht die Funktionalitäten des Frontends:



Nicht funktionale Anforderungen

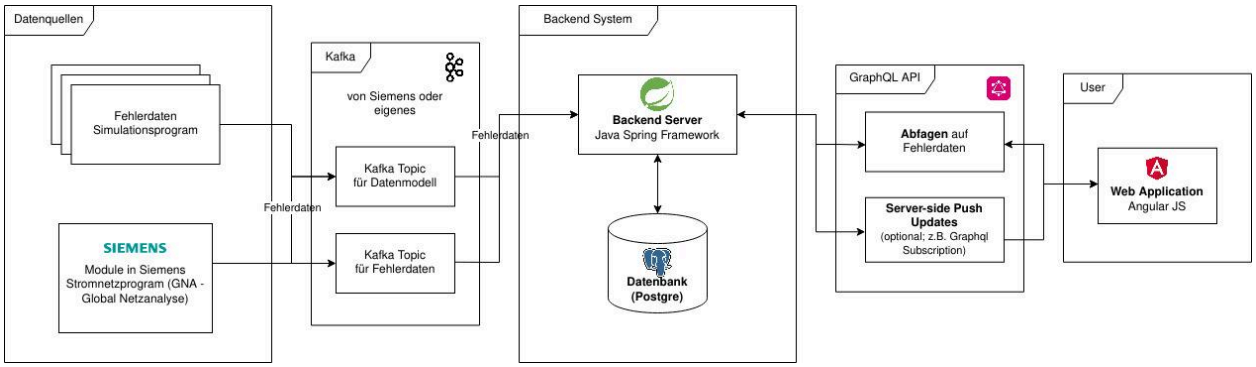
[Non-Functional Requirements]

Aspekt \ Projektergebnis	Webanwendung	Kafka-System
Usability	Intuitive Benutzbarkeit	-
Design	Nicht responsive für mobile Geräte	-
Navigation	Einfache Menüs, max. drei Klicks notwendig, um irgendwohin zu kommen	-
Look & Feel	Anlehnung an Siemens Applikationen; Intuitiv	Integration Siemens Kafka System
Barriere-freiheit	Kein TTS, Zoom oder Augensteuerung	-
Mehrsprachig-keit	Prinzipiell alles English, Icons ermöglichen eingeschränkte multilinguale Nutzung; Möglichkeit auf Erweiterung der Sprachen mittels Konfigurationsfile	-
Customisation	Dark / Light Mode	-
Zeitverhalten	-	Schnelle Änderungen

Technische Anforderungen

In diesem Kapitel wird der Technologie-Stack detaillierter beschrieben.

Übersicht



Backend

Die **Daten**, welche über das Kafka System übertragen werden können in **zwei Arten** unterteilt werden:

- Findings - Informationen über die gefundenen Fehler
- Datenmodell Objekte - Informationen über den Aufbau des Netzes

Diese zwei Arten von Daten werden über **zwei verschiedene Kafka Topics** erhalten.

Der Backend-Server, welcher mit dem **Java Spring Framework** implementiert ist, soll die Daten über die Event Streaming Plattform **Kafka** erhalten. Weiters muss es erhaltene Daten in einer **Datenbank** speichern und als **GraphQL API** den Frontend zur Verfügung stellen.

Frontend

Die Webanwendung verwendet das AngularJS Framework. Genau wie React, Vue und viele andere JS Frameworks basiert dieses auf Komponenten.

Über eine GraphQL API können Abfragen an das Backend gestellt werden.

Mockup

Das Mockup für die Web-Applikation wurde mit Figma erstellt. Hier ist ein Link zu der Datei: https://www.figma.com/file/PpiE28AJCmFWBdhD66wYX5/Siemens?type=design&mode=design&t=pvAVxSDRdFlCOXGr-1

Folgende Bilder veranschaulichen grob das Design: