



TECO Research Group

Marcel Köpke
Matthias Budde
Till Riedel



PaVoS Logo

PFLICHTENHEFT

Version 0.1

Visualizing & Mining of Geospatial Sensorstreams with Apache Kafka

Jean Baumgarten
Oliver Liu
Patrick Ries
Erik Wessel
Thomas Frank

27. Mai 2018

Inhaltsverzeichnis

1	Zielbestimmung	4
1.1	Musskriterien	4
1.1.1	Backend (Server)	4
1.1.2	Frontend (Webinterface)	4
1.2	Wunschkriterien	5
1.2.1	Backend (Server)	5
1.2.2	Frontend (Webinterface)	5
1.3	Abgrenzungskriterien	6
2	Produkteinsatz	7
2.1	Anwendungsbereiche	7
2.2	Zielgruppe	7
3	Produktumgebung	8
3.1	Software	8
3.1.1	Server	8
3.1.2	Client	8
3.2	Hardware	8
3.2.1	Server	8
3.2.2	Client	8
4	Funktionale Anforderungen	9
4.1	Pflicht Funktionale Anforderungen	9
4.1.1	Server	9
4.1.2	Webinterface	12
4.2	Wunsch Funktionale Anforderungen	13
4.2.1	Server	13
4.2.2	Webinterface	14
5	Produktdaten	15
5.1	Server	15
5.2	Webinterface	15
6	Nichtfunktionale Anforderungen	16
7	Qualitätssicherung	18
7.1	Globale Testszenarien	18
7.1.1	Server	18

7.1.2	Webinterface	20
8	Systemmodelle	23
8.1	Anwendungsfälle	23
8.1.1	Bedienung des Webinterface	23
8.1.2	Bedienung der Konfigurations-GUI	24
8.2	Aktivitätsdiagramm	26
9	Entwicklungsumgebung	28
9.1	Entwicklungstools	28
9.2	Verwendete Technologien	28
10	Benutzeroberfläche	29
10.1	Konfigurations-GUI	29
10.2	Webinterface	31
11	Anhang	36

1 Zielbestimmung

Das Produkt dient der Verarbeitung und Darstellung von Sensordatenstreams. Durch die übersichtliche Visualisierung der Daten auf einer Karte wird die schnelle Analyse von großen Datenmengen ermöglicht und der Zeitaufwand wird minimiert.

Ein Hauptmerkmal unseres Produktes ist die Fähigkeit, zusätzlich zu Echtzeitdaten auch historische Datenbestände zu verarbeiten und zu exportieren.

1.1 Musskriterien

1.1.1 Backend (Server)

- MK1000** Der Server kann Sensordaten empfangen
- MK1010** Eingeführte Sensordaten werden gesichert
- MK1020** Neue Sensordaten werden zeitnah an alle Instanzen des Webinterfaces weitergeleitet und dargestellt
- MK1030** Der Dienst ist logisch modular aufgebaut und erlaubt das Ergänzen und Ersetzen von einzelnen funktionalen Modulen, wie z.B. verschiedene Exportformate, Zwischenmodule
- MK1040** Der Server verarbeitet und speichert Daten für spätere Verwendung
- MK1050** Der Server kann vorverarbeitete und gespeicherte Daten abrufen
- MK1060** Der Server unterstützt skalar- und vektorwertige Sensortypen
- MK1070** Der Server kann durch eine Admin-GUI gesteuert werden
- MK1080** Der Server unterstützt das Filtern von ausgegebenen und angezeigten Daten

1.1.2 Frontend (Webinterface)

- MK2000** Das Webinterface unterstützt die rasterisierte Darstellung der Sensordaten auf einer Weltkarte in Form von vordefinierten Shapes
- MK2010** Das Webinterface unterstützt die Darstellung einer auf Deutschland beschränkten Ansicht

- MK2020** Der Nutzer kann die Karte in vordefinierten Detaillierungsgraden darstellen lassen
- MK2030** Der Nutzer kann aktuelle und historische Sensordaten über das Webinterface darstellen lassen
- MK2040** Der Nutzer kann die Sensordaten über das Webinterface herunterladen
- MK2050** Der Nutzer kann kürzlich beobachtete Daten als Wiederholung anzeigen lassen
- MK2060** Das Webinterface unterstützt die Darstellung von erweiterten Informationen bzgl. der Sensordaten in Form von Graphen
- MK2070** Die Standardsprache des Webinterfaces ist Englisch
- MK2080** Das Webinterface kann parallel von mehreren Nutzern aufgerufen und benutzt werden

1.2 Wunschkriterien

1.2.1 Backend (Server)

- WK1000** Der Server skaliert mit unterschiedlich großen Datenmengen
- WK1010** Der Server läuft auch mit fehlerhaften Daten stabil
- WK1020** Der Server überarbeitet im Leerlauf fehlerhafte Daten aus der Datenbank
- WK1030** Der Server unterstützt das Hinzufügen von neuen Anzeigesprachen für das Webinterface
- WK1040** Der Server unterstützt den Import von historischen Daten im NetCDF-Format und kann diese Daten problemlos verarbeiten

1.2.2 Frontend (Webinterface)

- WK2000** Die Genauigkeit der Darstellung von Clustern wird entsprechend der Zoom-Stufe angepasst
- WK2010** Die Approximation von Clustern führt nicht zu größeren Diskrepanzen oder Wartezeiten
- WK2020** Der Nutzer kann verschiedene Sensordatentypen an- bzw. ausschalten
- WK2030** Der Nutzer kann seine Ansicht als Favoriten abspeichern
- WK2040** Der Nutzer kann favorisierte Ansichten wiederherstellen
- WK2050** Der Nutzer kann Sensoren melden

- WK2060** Dem Nutzer wird darauf hingewiesen, wenn das Abrufen von Sensordaten nicht möglich ist
- WK2070** Enthält ein Cluster keine Sensoren, wird der Cluster nicht angezeigt
- WK2080** Der Nutzer kann zwischen einer automatischen und einer manuellen Aktualisierung von Echtzeitsensordaten wechseln
- WK2090** Der Nutzer kann eine Anzeigesprache auswählen
- WK2100** Die Anwendung speichert automatisch die Einstellungen und Favoriten des Nutzers über Browsersessions hinweg

1.3 Abgrenzungskriterien

- AK1000** Der Server speichert Sensordaten nicht auf unbegrenzte Zeit und in unbegrenzter Menge
- AK1010** Der Server speichert keine Daten von Nutzern und deren Aktivitäten/Interaktionen mit dem Webinterface
- AK1020** Der Datendurchsatz des Servers wird durch lokale Netzwerkgeschwindigkeiten beschränkt
- AK1030** Der Server ist nicht in der Lage, korrekte Vorhersagen zu erstellen
- AK1040** Der Server ist nicht in der Lage, Sensordaten von fehlerhaften Sensoren auszuwerten
- AK1050** Der Server ist nicht in der Lage, unbegrenzt viele Daten anzuzeigen und zu aktualisieren
- AK1060** Der Server ist nicht in der Lage, durch Störungseinflüsse veränderte Sensordaten zu erkennen
- AK1070** Der Server unterstützt keine weiteren Eingabeformate als Apache Kafka Streams (und, falls **WK1040** erfüllt ist, NetCDF-Dateien)
- AK1080** Der Server unterstützt nicht das manuelle Hinzufügen von neuen Sensordaten

2 Produkteinsatz

Das Produkt ermöglicht es seinen Nutzern, Echtzeitdaten sowie archivierte Daten vieler Sensoren von unterschiedlichen Messgrößen abzurufen und darzustellen. Es bietet dem Nutzer hierfür eine moderne und intuitive webbasierte Bedienoberfläche. Weiterhin ermöglicht das Produkt dem Nutzer über eine öffentliche Schnittstelle den direkten Zugriff auf die Daten. Unter anderem können diese Daten als Archivdatei exportiert werden. Durch die modulare Architektur unseres Produkts werden vielseitige Einsatzmöglichkeiten außerhalb der Luftqualitätsmessung ermöglicht.

2.1 Anwendungsbereiche

- Analyse und Visualisierung von Sensordaten
- Archivierung von Sensordaten

2.2 Zielgruppe

- Personen, die Sensordaten auswerten und visuell darstellen wollen. Dazu gehören:
 - Wissenschaftler
 - Privatpersonen, die mit Sensoren arbeiten
- Personen, die Sensordaten sammeln und archivieren wollen
- generelle Bevölkerung, um Informationen über aktuelle Werte von Sensoren zu erhalten, z.B. über Luftverschmutzung, Temperatur usw.

3 Produktumgebung

Das Produkt ist intern in einer Master-Worker Rechnerarchitektur aufgebaut. Clients verbinden sich mit dem Server über einen Webbrowser.

3.1 Software

3.1.1 Server

- FROST-Server liefert Sensordaten
- Kafka speichert und verteilt Sensordaten
- Docker für die Containerumgebung
- Java-Anwendung mit Java 8
- Apache Tomcat als Webserver

3.1.2 Client

- Webbrowser, Referenzstandard Mozilla Firefox 57

3.2 Hardware

3.2.1 Server

- Containerumgebung (min. 1GB RAM, 10GB Festplattenspeicher, 2 CPU Kerne)

3.2.2 Client

- Standardrechner mit genügend Leistung für Firefox (min. 512MB RAM, 512MB Festplattenspeicher, 1.5GHz CPU)

4 Funktionale Anforderungen

4.1 Pflicht Funktionale Anforderungen

4.1.1 Server

PF1000 Initialisierendes Starten

Beim ersten Aufruf wird der Nutzer aufgefordert, einen Kafka Stream in die Admin-GUI einzufügen und dessen Gültigkeit wird überprüft. **PF1010**

Vorbedingung: Programm wird gestartet.

Nachbedingung: Das Programm ist einsatzbereit.

Bei Scheitern: Kann das Programm nicht weiter genutzt werden und man wird aufgefordert einen Stream einzufügen.

PF1010 Kontrollieren der Gültigkeit des Streams

Es wird eine Verbindung zu Kafka hergestellt und seine Struktur wird für das Programm überprüft.

Vorbedingung: Kafka-Stream wurde hinzugefügt **PF1020**

Nachbedingung: Der Kafka Stream wird vom akzeptiert.

Beim Scheitern: Der Benutzer wird aufgefordert einen anderen Kafka Stream einzufügen.

PF1020 Einfügen eines Kafka Streams

Ein Kafka Stream kann in der Admin-GUI eingefügt werden und dessen Gültigkeit wird überprüft **PF1010**

Vorbedingung: Das Programm ist gestartet

Nachbedingung: Der Kafka Stream wird dem Programm hinzugefügt.

Beim Scheitern: Der Kafka Stream wird nicht hinzugefügt und dem Nutzer wird die Möglichkeit geboten einen anderen Stream hinzuzufügen.

PF1030 Entfernen von Stream

Ein Kafka Stream kann mit Hilfe der Admin-GUI entfernt werden und falls der Initial-Stream entfernt wird, wird das Programm gestoppt.

Vorbedingung: Es sind Kafka Streams vorhanden

Nachbedingung: Der Kafka Stream dem Programm entfernt.

Beim Scheitern: Der Kafka Stream wird nicht entfernt.

PF1040 Ein und Ausschalten des Programms

Über die Admin-GUI kann das Programm gestoppt und gestartet werden.

Vorbedingung: Das Programm läuft

Nachbedingung: Das Programm wird gestoppt oder gestartet.
Beim Scheitern: Nichts passiert und es wird eine Fehlermeldung ausgegeben.

PF1050 Importieren von NetCDF

Eine NetCDF kann importiert werden und daraus wird ein Kafka Stream erstellt. Dieser wird dann automatisch hinzugefügt **PF1020**

Vorbedingung: Das Programm ist gestartet

Nachbedingung: Die Sensordaten werden verarbeitet.

Beim Scheitern: Nichts passiert und es wird eine Fehlermeldung ausgegeben.

PF1060 Importieren von JSON

Eine JSON kann importiert werden und daraus wird ein Kafka Stream erstellt. Dieser wird dann automatisch hinzugefügt **PF1020**

Vorbedingung: Das Programm ist gestartet

Nachbedingung: Die Sensordaten werden verarbeitet.

Beim Scheitern: Nichts passiert und es wird eine Fehlermeldung ausgegeben.

PF1070 Gruppierung der Daten

Die Sensordaten werden nach Vektor und Skalare Daten gruppiert.

Vorbedingung: Ein Kafka Stream vorhanden.

Nachbedingung: Die Sensordaten werden weiter verarbeitet.

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1080 Berechnung des Mittelwerts

Von den gruppierten Sensordaten werden Mittelwerte berechnet. **PF1070**

Vorbedingung: Sensordaten sind gruppiert.

Nachbedingung: Die Mittelwerte können auf der View angezeigt werden

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1090 Kafka Streams werden aufgeteilt

Die Kafka Streams werden auf verschiedene Worker-Dienste aufgeteilt und dort verarbeitet zu werden **PF1080**

Vorbedingung: Ein Kafka Stream vorhanden

Nachbedingung: Die Daten werden erfolgreich verarbeitet.

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1100 Clusterposition berechnen

Aus den Sensordaten werden Cluster berechnet.

Vorbedingung: Mittelwert wurde berechnet

Nachbedingung: Diese Kacheln können in der View angezeigt werden

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1110 Clustergröße berechnen

Die Größe der Cluster kann in der Konfiguration-GUI eingestellt werden

Vorbedingung: Clusterposition wurde berechnet

Nachbedingung: Die Cluster werden angepasst

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1120 Clusterform auswählen

Die Form der Cluster kann in der Konfiguration-GUI eingestellt werden

Vorbedingung: Clusterposition wurde berechnet

Nachbedingung: Die Cluster werden angepasst

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1130 Cluster werden an View gesendet

Die Cluster werden an die View gesendet um dort angezeigt zu werden

Vorbedingung: Clusterposition, Clustergröße und Clusterform wurde berechnet.

Nachbedingung: Die Cluster werden von der View entgegengenommen.

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1140 Sensoren in Tabellenansicht

Die einzelnen Sensoren werden in einer Tabelle angezeigt

Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Sensoren können angezeigt werden

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1150 Sensoren sortieren nach ihren Parameter

Man kann die Sensoren in der Tabellenansicht sortieren nach ihren Parameter

Vorbedingung: Sensordaten in der Tabelle vorhanden

Nachbedingung: Die Sensoren können angezeigt werden

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1160 Intervallspeicherung der Daten der Sensoren

In einem fest definierbaren Intervall werden Daten für Sensoren in einem Datenerhaltungssystem gespeichert.

Vorbedingung: Kafka Stream muss vorhanden sein

Nachbedingung: Die Daten werden für Statistiken weiterverarbeitet.

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1170 Berechnung von Zeitstatistiken für Sensoren

Aus der Intervallspeicherung **PF1160** einzelne Sensoren werden die Daten in eine Zeitstatistik umformatiert.

Vorbedingung: Intervallspeicherung vorhanden

Nachbedingung: Die Daten werden an die View weiter gegeben

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1180 Berechnung von Zeitstatistiken für Cluster

Aus den Zeitstatistiken der einzelne Sensoren **PF1160** werden in Cluster

zusammen formatiert.

Vorbedingung: Intervallspeicherung vorhanden

Nachbedingung: Die Daten werden an die View weiter gegeben

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1190 Export der Daten als NetCDF

Die Sensordaten können in einem Zeitleichenintervall in dem Format Net-CDF exportiert werden

Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Daten werden exportiert

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1200 Export der Daten als JSON

Die Sensordaten können in einem Zeitleichenintervall in dem Format JSON exportiert werden

Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Daten werden exportiert

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

4.1.2 Webinterface

PF1210 Mehrere Instanzen des Webinterface

Mehre Instanzen des Webinterfaces werden aufrufbar sein, so dass mehrere Nutzer gleichzeitig mit den Sensordaten arbeiten können. Vorbedingung:

Kafka Stream vorhanden

Nachbedingung: View wird angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1220 Sensordaten werden auf der Karte dargestellt

Nach dem die Daten an die View gesendet wurden **PF1130**, werden sie auf der Karte angezeigt.

Vorbedingung: Daten wurden an die View gesendet

Nachbedingung: Die Daten werden angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1230 Verschiedene Sensor darstellen

Man kann die dargestellten Sensoren einteilen von deren Typ. Vorbedingung: Es werden Sensordaten angezeigt

Nachbedingung: Die Einteilung wird angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1240 Verschiedene Cluster

Man kann die Karte in verschiedene Cluster einteilen. Vorbedingung: Es werden Sensordaten angezeigt

Nachbedingung: Die Einteilung wird angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1250 Export von Daten

Man kann die Daten auf der View exportieren in Json und netCDF **PF1190 PF1200**

Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Daten werden exportiert

Beim Scheitern: Es wird eine passende Fehlermeldung ausgegeben.

PF1260 Wiederholung anzeigen

Auf der View kann man Daten mit Hilfe von einem Slider als Wiederholung anzeigen lassen.

Vorbedingung: Daten wurden an die View gesendet

Nachbedingung: Die Wiederholung wird angezeigt.

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1270 Detail Ansicht von Sensoren

Auf der View kann man von einzelnen Sensoren eine Detailansicht darstellen um so nähere Informationen über den Sensor zu erhalten. **PF1170**

Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Details werden angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1280 Statisten anzeigen

Auf der View kann man sich vorberechnen Statistiken anzeigen lassen

PF1170PF1180 Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Statistiken werden angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1290 Echtzeitdaten

Es besteht immer die Möglichkeit sich die Echtzeitdaten für ein Cluster oder Sensor anzeigen zu lassen Vorbedingung: Kafka Stream vorhanden

Nachbedingung: Die Echtzeitdaten werden angezeigt

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

PF1300 Zoomen in die Karte

Auf der angezeigten Karte kann man rein und raus zoomen und die Cluster werden entsprechend angepasst Vorbedingung: Karte wird angezeigt.

Nachbedingung: Die Daten werden angezeigt.

Beim Scheitern: Es wird eine Fehlermeldung in den Log geschrieben.

4.2 Wunsch Funktionale Anforderungen

4.2.1 Server

WF1000 Der Dienst sucht nach Fehlerhaften Sensordateien

Der Dienst überprüft im Datenerhaltungssystem ob Sensordaten Outlier

sein können und gibt den Nutzer eine Meldung, was an den Daten ungewöhnlich ist. Vorbedingung: Kafka Stream vorhanden
Nachbedingung: Fehlerhafte Daten werden angezeigt.

WF1010 Mehrere Sprachen

Die Anzeigesprache kann in mehrer Sprachen geändert werden. Vorbedingung: Programm gestartet.
Nachbedingung: Sprache wird umgesetzt

4.2.2 Webinterface**WF1020 Favoriten**

Man kann Elemente auf der View favorisieren um somit schneller zu denen gelangen bei späteren einsehen
Vorbedingung: Favoriten Symbol muss vorhanden sein
Nachbedingung: Favorit wird gespeichert

WF1030 Mehre Ansichten

Man kann 2 Karten mit verschiedenen Datensätzen nebeneinander setzen um diese zu vergleichen
Vorbedingung: Kafka Stream muss vorhanden sein
Nachbedingung: Parallele Ansicht

WF1040 Eigene Filter

Man kann eigene Filter erstellen um seine Analyse genauer durchzuführen
Vorbedingung: Kafka Stream muss vorhanden sein
Nachbedingung: Filter-Ansicht

5 Produktdaten

5.1 Server

PD1000 Sensordaten

Apache Kafka speichert alle eintreffenden Sensordaten in einem persistenten Speicher.

Echtzeitsensordaten werden bis zu 1GB in dem Arbeitsspeicher gespeichert.

5.2 Webinterface

PD2000 Benutzerdefinierte Einstellungen

Beim Setzen von Favoriten und beim Verlassen der Browsersession werden folgende Daten gespeichert:

- Zoomstufe
- Lokalität
- Clustergruppe
- Clustershape
- Sensortyp
- an- bzw. ausgeschaltete Sensordatentypen

Alle anderen Daten, die für eine korrekte Darstellung benötigt werden, werden nicht persistent gespeichert, sondern nach Bedarf generiert.

6 Nichtfunktionale Anforderungen

NF1000 Anzeigesprache

Die Standardanzeigesprache des Webinterfaces ist Englisch.

NF1010 Parallele Zugriffe

Das Webinterface muss von bis zu 100 Nutzern gleichzeitig bedient werden können. Alle folgenden Anforderungen gelten nur für bis zu 100 parallele Nutzer. Für mehr Nutzer sind keine Leistungsgarantien gegeben.

NF1020 Sichtbare Cluster

Das Webinterface muss bis zu 1000 Cluster gleichzeitig anzeigen können. Alle folgenden Anforderungen gelten nur für bis zu 1000 sichtbare Cluster. Für mehr Cluster sind keine Leistungsgarantien gegeben.

NF1030 Zoomstufen

Das Webinterface muss je nach Ansicht 4 bis 5 unterschiedliche Zoomstufen unterstützen.

NF1040 Verzögerung von neuen Daten

Neue Echtzeitsensordaten werden innerhalb einer Minute verarbeitet und auf allen Instanzen der Webinterface angezeigt.

NF1050 Verzögerung von historischen und archivierten Daten

Historische und archivierte Daten, die eine große Datenmenge umfassen und/oder von einem persistenten Speicher geladen werden müssen, benötigen mehr Zeit zur Verarbeitung und Visualisierung. Für diese sind keine Leistungsgarantien gegeben.

NF1060 Speicherung von kürzlichen Echtzeitsensordaten

Kürzliche Echtzeitsensordaten werden bis zu einer Gesamtgröße von 1GB in dem Arbeitsspeicher gespeichert. Diese kürzlichen Daten können schneller abgerufen werden als ältere Daten und werden für die Wiederholungsansicht verwendet.

NF1070 Speicherung von älteren Echtzeitsensordaten

Alle an Apache Kafka angelegte Sensordaten werden von Apache Kafka in einem persistenten Speicher gespeichert. Die Größe des Speichers begrenzt die Menge der gespeicherten Daten.

NF1080 Skalierbarkeit

Der Server startet automatisch neue Worker-Container um Sensordaten

und Anfragen des Servers zu verarbeiten. Mindestens 10 Worker können je nach Bedarf gestartet und beendet werden.

NF1090 Stabilität

Der Server muss im normalen Betrieb problemlos mit unerwarteter Terminierung von einem oder mehreren Worker-Containern umgehen und innerhalb von fünf Minuten die ursprüngliche Rechenleistung wiederherstellen. Fehler in den Eingabedaten wie ungültige Werte oder Syntaxfehler führen nicht zu einem Absturz des Servers.

NF1100 Reaktionszeit

Das Webinterface muss innerhalb von zwei Sekunden auf Interaktionen des Nutzers reagieren. Falls der Webserver bis dahin nicht das erwünschte Ergebnis liefern kann, wird der Nutzer darauf hingewiesen.

NF1110 Ladezeiten der Darstellungselemente

Ein Zugriff auf das Webinterface lädt Darstellungselemente wie Cluster-Shapes, Graphen und Buttons innerhalb von 5 Sekunden.

NF1120 Genauigkeit der Clusterberechnung

Die Approximation der Daten in Clustern darf nicht zu einer Abweichung über 5% von Ergebnissen einer akkuraten Berechnung führen.

NF1130 Geschwindigkeit der Clusterberechnung

Die Approximation der Daten in Clustern darf nicht mehr als 2 Sekunden dauern.

NF1140 Aktualisierungsgeschwindigkeit

Die Karte aktualisiert Echtzeitsensordaten alle 10 Sekunden.

7 Qualitätssicherung

Für die Komponententests unseres Systems verwenden wir grundsätzlich JUnit. Für weiterführenden Tests werden im folgenden Abschnitt Szenarien beschrieben.

7.1 Globale Testszenarien

7.1.1 Server

TI1000 Erstanwendung der Konfigurations-GUI mit Kafka Stream

Die Konfigurations-GUI wurde noch nie gestartet und es steht ein Kafka Stream namens 'TestStream' bereit.

Der Nutzer startet die Konfigurations-GUI, woraufhin diese ihn auffordert, den hinzuzufügenden Kafka Stream auszuwählen.

Der Nutzer schreibt 'TestStream' in das Feld des Popups und bestätigt.

Die Konfigurations-GUI akzeptiert den Stream und fügt ihn hinzu.

TI1010 Erstanwendung der Konfigurations-GUI ohne Kafka Stream

Die Konfigurations-GUI wurde noch nie gestartet und es steht kein Kafka Stream bereit.

Der Nutzer startet die Konfigurations-GUI, woraufhin diese ihn auffordert, den hinzuzufügenden Kafka Stream auszuwählen.

Der Nutzer schreibt 'TestStream' in das Feld des Popups und bestätigt.

Die Konfigurations-GUI akzeptiert den Stream nicht und fordert ihn erneut auf, den hinzuzufügenden Kafka Stream auszuwählen.

TI1020 Hinzufügen von zusätzlichen Streams zur Konfigurations-GUI mit Kafka Stream

Die Konfigurations-GUI ist gestartet, es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden und es steht ein Kafka Stream namens 'TestStream2' bereit.

Der Nutzer wählt die Option 'Stream hinzufügen', woraufhin die Konfigurations-GUI ihn auffordert, den hinzuzufügenden Kafka Stream auszuwählen.

Der Nutzer schreibt 'TestStream2' in das Feld des Popups und bestätigt.

Die Konfigurations-GUI akzeptiert den Stream und fügt ihn hinzu.

TI1030 Hinzufügen von zusätzlichen Streams zur Konfigurations-GUI ohne Kafka Stream

Die Konfigurations-GUI ist gestartet, es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden und es steht kein Kafka Stream bereit.
Der Nutzer wählt die Option 'Stream hinzufügen', woraufhin die Konfigurations-GUI ihn auffordert, den hinzuzufügenden Kafka Stream auszuwählen.

Der Nutzer schreibt 'TestStream' in das Feld des Popups und bestätigt.
Die Konfigurations-GUI akzeptiert den Stream nicht und fordert ihn erneut auf, den hinzuzufügenden Kafka Stream auszuwählen.

TI1040 Entfernen von Streams in der Konfigurations-GUI

Die Konfigurations-GUI ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden.

Der Nutzer wählt die Option 'Stream entfernen', woraufhin die Konfigurations-GUI ihn auffordert, den zu entfernenden Kafka Stream auszuwählen.

Der Nutzer schreibt 'TestStream' in das Feld des Popups und bestätigt.
Die Konfigurations-GUI entfernt den Stream.

TI1050 Export einer Archiv-Datei in der Konfigurations-GUI

Die Konfigurations-GUI ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden.

Der Nutzer wählt die Option 'Export Stream', woraufhin die Konfigurations-GUI ihn auffordert, den Dateityp, das Verzeichnis und den Namen der zu exportierenden Archiv-Datei auszuwählen.

Der Nutzer wählt das Format

- NetCDF
- CSV

, wählt das Verzeichnis 'SensorExport' bzw. erstellt es, falls es noch nicht existiert, legt den Namen auf 'TestSensorExport' fest und bestätigt.

Die Konfigurations-GUI exportiert den Stream.

TI1060 Import einer Archiv-Datei in der Konfigurations-GUI

Die Konfigurations-GUI ist gestartet, es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden und es existiert eine importierbare Archiv-Datei im Verzeichnis 'SensorExport' mit dem Namen 'TestSensorExport'.

Der Nutzer wählt die Option 'Import Stream', woraufhin die Konfigurations-GUI ihn auffordert, die zu importierenden Archiv-Datei auszuwählen.

Der Nutzer wählt die

- NetCDF
- CSV

Datei und bestätigt.

Die Konfigurations-GUI importiert den Stream.

TI1070 Clusterform in der Konfigurations-GUI festlegen

Die Konfigurations-GUI ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden.

Der Nutzer wählt die Option 'Select Clusterform', woraufhin die Konfigurations-GUI ihn auffordert, aus mehreren Möglichkeiten eine Form für die Cluster zu berechnen.

Der Nutzer wählt die

- Quadrat
- Hexagon

Form und bestätigt.

Die Konfigurations-GUI akzeptiert die neue Clusterform und berechnet Mittelwerte und Clusterpositionen neu.

TI1080 Clustergröße in der Konfigurations-GUI festlegen

Die Konfigurations-GUI ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden.

Der Nutzer wählt die Option 'Change Clustersize', woraufhin die Konfigurations-GUI ihn auffordert, einen Wert innerhalb eines Bereiches einzutragen.

Der Nutzer wählt die Größe '2' und bestätigt.

Die Konfigurations-GUI akzeptiert die neue Clusterform und berechnet Mittelwerte und Clusterpositionen neu.

7.1.2 Webinterface**TI2000 Darstellung des berechneten Mittelwertes im Webinterface**

Das Webinterface ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden. Die Ansicht des Webinterfaces ist auf Karlsruhe in Deutschland fokussiert.

Der Nutzer wählt alle Skalierungsstufen der Karte in sich entfernender Reihenfolge aus, woraufhin die berechneten Mittelwerte auf der Karte als Farbe der Cluster zu sehen sind.

TI2010 Darstellung von Clustern ohne Sensoren im Webinterface

Das Webinterface ist gestartet und es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden. Die Ansicht des Webinterfaces ist auf Karlsruhe in Deutschland fokussiert. Es existiert ein Cluster ohne Sensoren in seinem Einflussgebiet, hier auch 'leerer Cluster' genannt.

Der Nutzer wählt alle Skalierungsstufen der Karte in sich entfernender Reihenfolge aus. Die Karte zeigt leere Cluster nicht an.

TI2020 Darstellung der tabellarischen Ansicht im Webinterface

Das Webinterface ist gestartet, es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden. Die Ansicht des Webinterfaces ist auf Karlsruhe in Deutschland fokussiert.

Der Nutzer wählt in der Ansicht den tabellarischen Modus. Daraufhin stellt das Webinterface die Sensordaten in einer geordneten Tabelle dar.

Der Nutzer selektiert dann den Parameter 'Name', woraufhin das Webinterface die Sensoren nach Namen alphabetisch sortiert.

TI2030 Benutzung mehrerer Webinterface-Instanzen

Das Webinterface ist nicht geöffnet, es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden und im Browser sind Daten des Webinterfaces gespeichert. Die Ansicht des Webinterfaces ist auf Karlsruhe in Deutschland fokussiert.

Der Nutzer öffnet das Webinterface für jede mögliche Ansicht einmal mehr in unterschiedlichen Tabs oder Fenstern des Browsers.

Das Webinterface lädt auf allen Instanzen zu Beginn die gleichen Ansichten.

Anschließend wählt der Nutzer pro Instanz verschiedene Ansichten.

Das Webinterface stellt pro Instanz eine Ansicht dar. Die Ansichten der Instanzen werden nach der Auswahl gleichzeitig angezeigt.

TI2040 Auswahl eines Sensortyps im Webinterface

Das Webinterface ist geöffnet und es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden. Die Ansicht des Webinterfaces ist auf Karlsruhe in Deutschland fokussiert.

Der Nutzer wählt im Webinterface die darzustellenden Sensortypen aus. Er aktiviert nur den Sensortyp 'Feinstaub'.

Das Webinterface stellt daraufhin nur noch die Werte für 'Feinstaub'-Sensoren dar.

TI2050 Export einer Archiv-Datei aus dem Webinterface

Das Webinterface ist geöffnet und es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden.

Der Nutzer wählt die Option 'Export Sensordata' und wählt das Format 'NetCDF', woraufhin das Webinterface den browserspezifischen Download-Dialog öffnet.

Der Nutzer wählt das Verzeichnis 'SensorExport' bzw. erstellt es, falls es noch nicht existiert, legt den Namen auf 'TestSensorExport' fest und bestätigt.

Der Nutzer wählt das Format

- NetCDF

, wählt das Verzeichnis 'SensorExport' bzw. erstellt es, falls es noch nicht existiert, legt den Namen auf 'TestSensorExport' fest und bestätigt.

Die Konfigurations-GUI exportiert den Stream.

TI2060 Wiederholungen im Webinterface anzeigen

Das Webinterface ist geöffnet und es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden.

Der Nutzer wählt im Webinterface eine Zeit aus oder verschiebt den Zeitregler 1 Minute zurück.

Das Webinterface stellt daraufhin eine Wiederholung der Sensordaten von der Ausgewählten Zeit bis zum aktuellen Zeitpunkt dar.

Der Nutzer kann mit einem Klick auf den 'Pause' Knopf die Wiederholung anhalten. Drückt er anschließend wieder auf 'Play', läuft die Wiederholung weiter.

TI2070 Echtzeitdarstellung im Webinterface

Das Webinterface ist geöffnet, es ist ein Kafka Stream namens 'TestStream' hinzugefügt worden und das Webinterface zeigt eine Wiederholung an.

Der Nutzer wählt 'Live' aus, woraufhin das Webinterface wieder die Echtzeitdaten darstellt und den Zeitregler wieder zurück auf 'Live' stellt.

TI2080 Detailansicht eines Sensors im Webinterface anzeigen

Das Webinterface ist geöffnet und es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden.

Der Nutzer wählt in der Ansicht einen Sensor aus.

Das Webinterface stellt daraufhin eine detaillierte Auflistung der Sensoreigenschaften dar.

TI2090 Melden eines Sensors im Webinterface

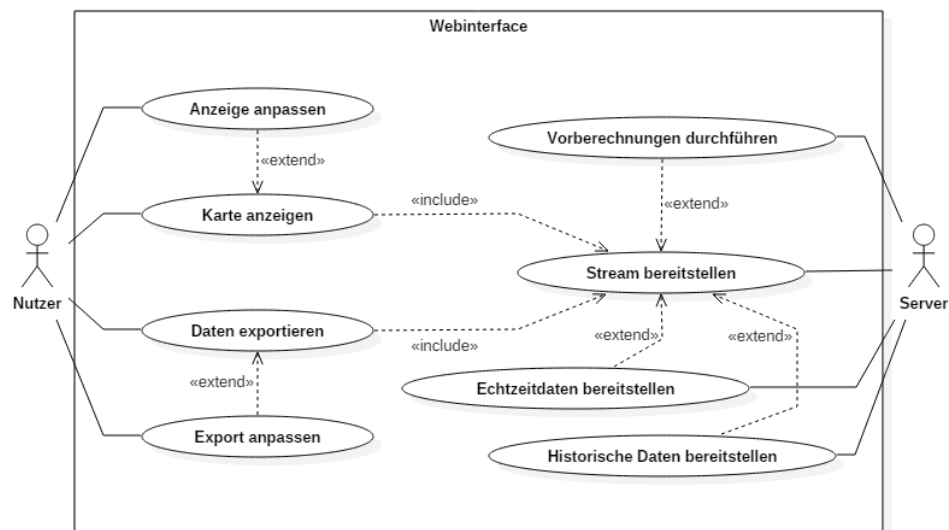
Das Webinterface ist geöffnet und es ist ein Kafka Stream namens 'Test-Stream' hinzugefügt worden.

Der Nutzer wählt in der Ansicht einen Sensor aus und klickt auf die Schaltfläche 'Report Sensor'. Anschließend schreibt der Nutzer in das Feld 'Reason' den Grund des Meldens an. Hier ist dieser Grund 'TestReportReason'. Das Webinterface zeigt dem Nutzer daraufhin an, dass die Meldung des Sensors erfolgt ist und speichert sie ab.

8 Systemmodelle

8.1 Anwendungsfälle

8.1.1 Bedienung des Webinterface



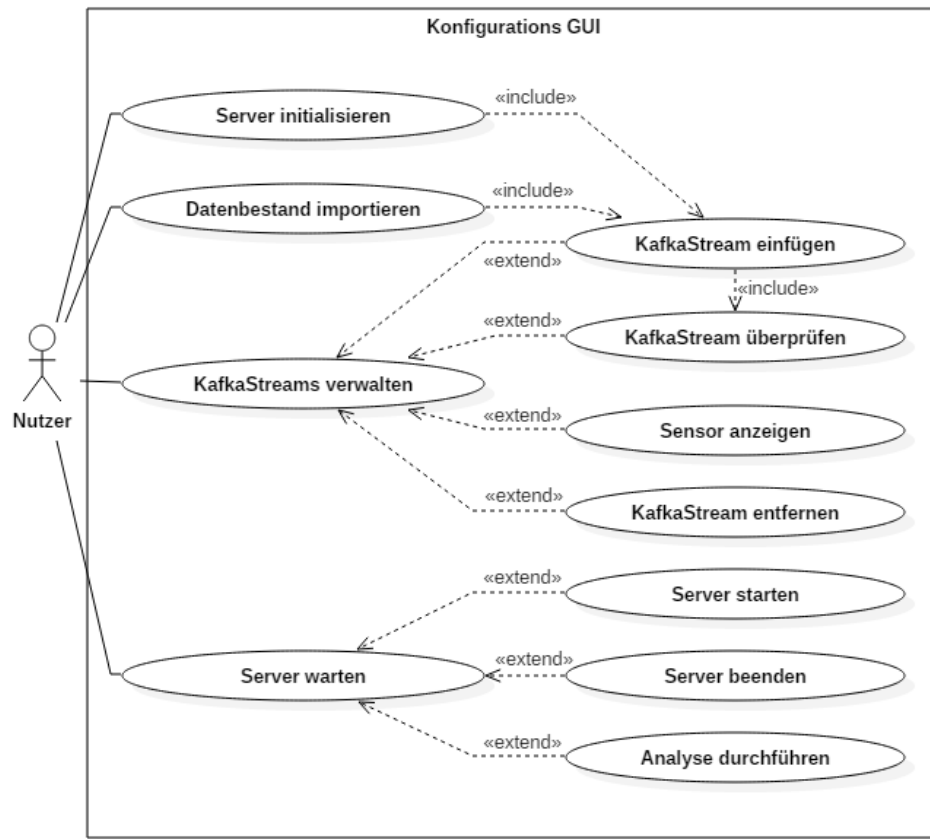
Dieser Anwendungsfall beschreibt die Bedienung des Webinterface. Dem Nutzer stehen folgende Aktionen zur Verfügung:

- Karte anzeigen
- Anzeige anpassen
- Daten exportieren
- Export anpassen

Nachdem der Nutzer auf das Webinterface zugreift, kann er sich die Karte anzeigen lassen. Diese visualisiert die Daten, welche vom Server durch KafkaStreams bereitgestellt werden. Diese können Echtzeit-, sowie historische Daten enthalten. Der Server führt im Hintergrund Vorberechnungen zur schnelleren Visualisierung durch. Die Anzeige der Karte, sowie des gesamten Interfaces lässt sich an die Wünsche des Nutzers anpassen, beispielsweise durch die Einbeziehung von Graphen zur besseren Darstellung. Die Datenbestände lassen sich herunterladen und in das gewünschte Format exportieren. Durch Beschränkung der Datenmengen

auf bestimmte Zeitintervalle, Datentypen und Wertebereiche kann der Export weiter angepasst werden.

8.1.2 Bedienung der Konfigurations-GUI



Dieser Anwendungsfall beschreibt die Bedienung der Konfigurations-GUI. Dem Admin stehen folgende Aktionen zur Verfügung:

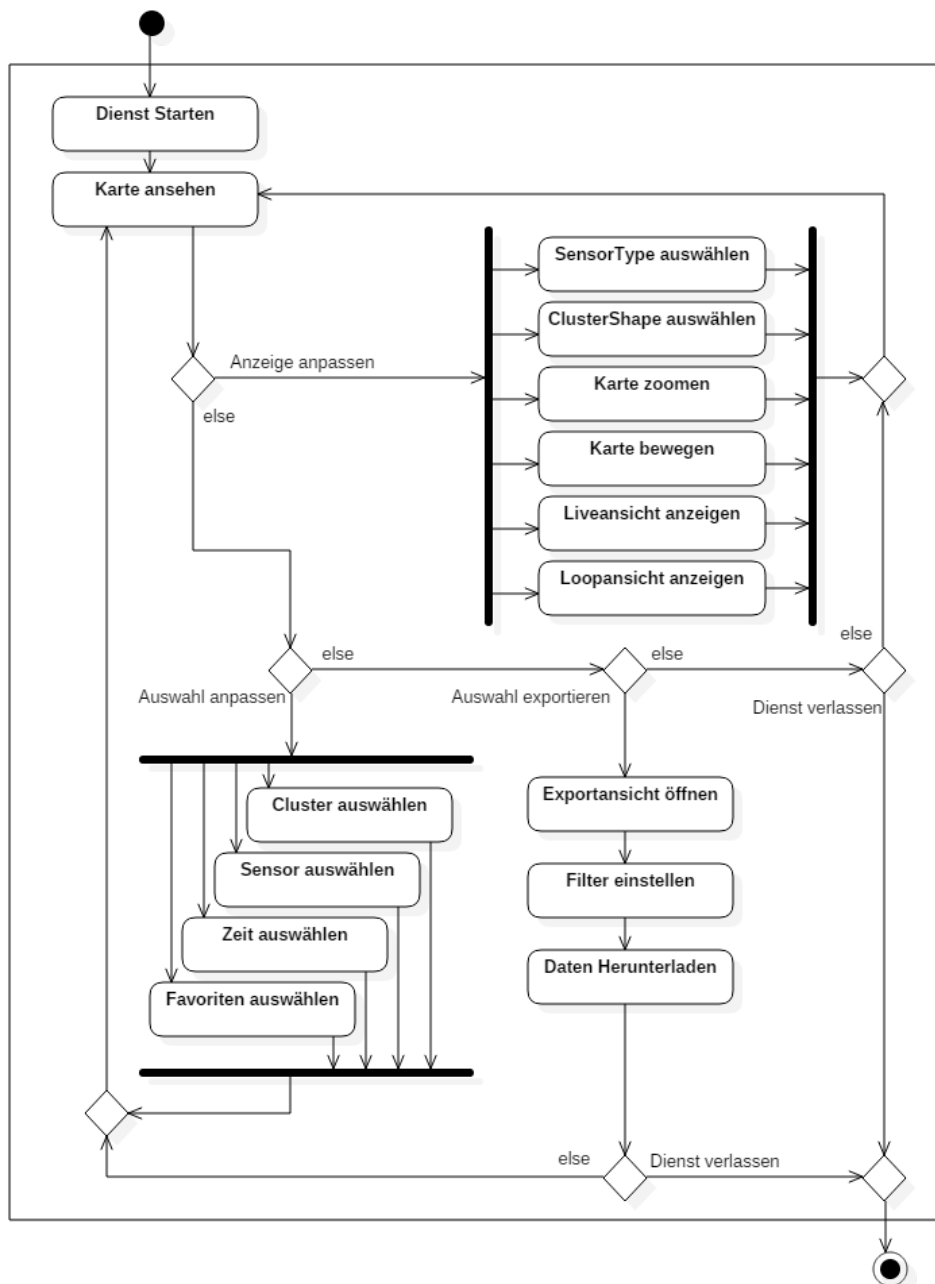
- Server initialisieren
- KafkaStreams verwalten
- Datenbestand importieren
- Server warten

Um den Server und damit die Applikation zu starten, greift der Nutzer auf die Konfigurations-GUI zu. Er initialisiert den gestarteten Server, indem ein KafkaStream, bzw. ein Datenbestand eingefügt wird, welcher in einen Stream umgewandelt wird. Dieser wird daraufhin überprüft. Gibt es keine Probleme, so können die Nutzer des Webinterfaces nun die Streams vom Server empfangen.

Außerdem ist es nun möglich, die KafkaStreams auf dem Server zu verwalten. Man kann neue Streams hinzufügen, Vorhandene entfernen und enthaltene Sensoren und Sensorendaten anzeigen. Weiterhin lässt sich der Server durch die Konfigurations-GUI warten. Man kann ihn starten, sowie beenden um Analysen durchzuführen und eventuelle Fehler zu beheben.

8.2 Aktivitätsdiagramm

8.2.1



Das Aktivitätsdiagramm beschreibt das Verhalten des Webdienstes, der durch das Laden der Dienstwebseite gestartet wird und durch Schließen des Tabs indem er geladen wurde wieder beendet wird.

Jegliche Interaktion mit dem Dienst findet auf derselben Ansicht statt. Es ist keine Reihenfolge für die möglichen Aktionen vorgegeben. So kann unabhängig voneinander der Kartenbereich oder die Auswahl (Selektion) angepasst werden. Das gleiche gilt für das Anfordern eines Downloads allgemeiner oder selektierter Daten.

Die Kartendarstellung kann angepasst werden, sodass die gewünschte Anzeige erreicht wird:

- SensorType auswählen
- ClusterShape auswählen
- Kartenbereich zoomen
- Kartenbereich verschieben
- Liveansicht anzeigen
- Loopansicht anzeigen

Hierbei handelt es sich bei der Liveansicht, um die aktuellen Daten, während der Loop eine Schleife über die letzten 24 Stunden anzeigt.

Die Selektion kann angepasst werden:

- Selektion eines Clusters
- Selektion eines Sensors
- Selektion eines Datums
- Favoriten als Selektion übernehmen

Wobei ein Favorit eine zuvor definierte Selektion aus Clustern auswählt.

Daten können exportiert werden. Dazu kann entweder die letzte Selektion verwendet werden. Alternativ wird über Filter der gewünschte Datensatz eingegrenzt. Darauf folgt dann der Download der Daten.

9 Entwicklungsumgebung

9.1 Entwicklungstools

Java IDE	Eclipse
Projektmanagement	Trello
Textverarbeitung	L ^A T _E X
TeX-Distribution	MiKTeX
L^AT_EX-Editor	TeXstudio
Versionskontrolle	Git
UML Tool	StarUML

9.2 Verwendete Technologien

Programmiersprache	Java 8
Web-Framework	?
Webserver	?
Datenbank	?

10 Benutzeroberfläche

Die folgenden Darstellungen sollen eine grobe Orientierung für die Benutzeroberflächen des Endprodukts geben. Diese können sich im Verlauf des Projektes noch ändern.

10.1 Konfigurations-GUI

Die Konfigurations-GUI soll einige Optionen bieten, die für reguläre Nutzer des Webinterface nicht verfügbar sein sollte.

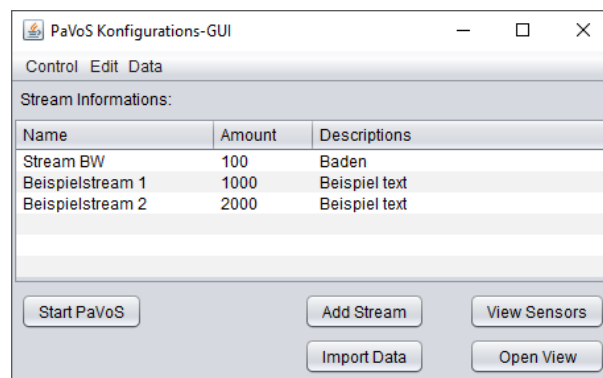


Abb. 10.1: Die Konfigurations-GUI zeigt die Streams des Dienstes an. Alle Funktionen sind in den Reitern enthalten. Wichtige Funktionen haben zusätzlich einen Button im unteren Bereich der GUI.

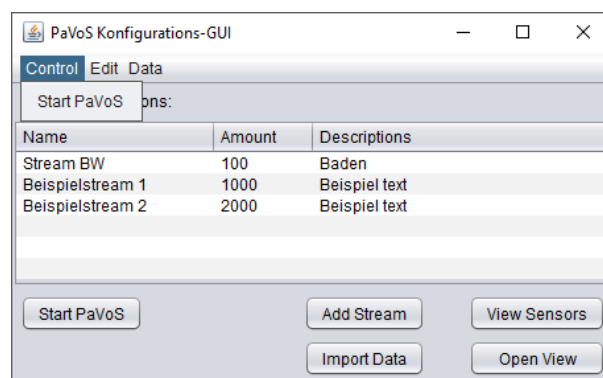


Abb. 10.2: Control-Reiter: Hier kann der Dienst gestartet und beendet werden.

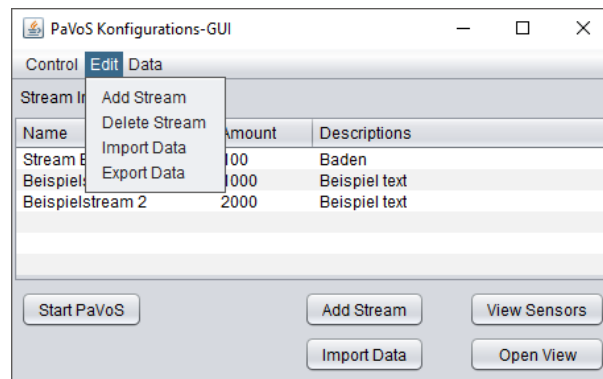


Abb. 10.3: Edit-Reiter: Dieser Reiter stellt die Funktionen zum hinzufügen oder entfernen der Streams bereit. Außerdem ist es hier möglich Daten zu importieren oder zu exportieren.

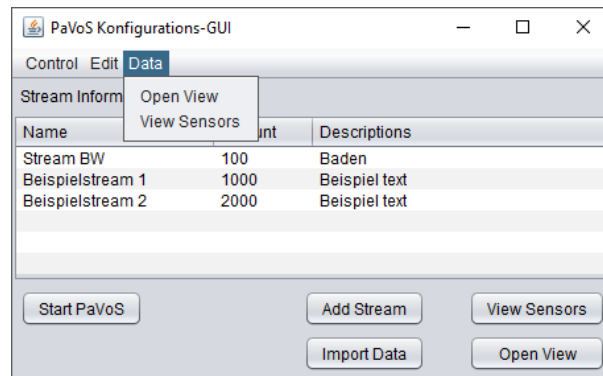


Abb. 10.4: Data-Reiter: Über den Data-Reiter kann eine Auflistung aller im System enthaltenen Sensoren angezeigt oder das Webinterface geöffnet werden.

10.2 Webinterface

Im Folgenden wird das Webinterface des Dienstes vorgestellt. Diese ist in den Grundzügen vergleichbar mit der von `maps.luftdaten.info`, bietet jedoch viele zusätzliche Funktionen an.

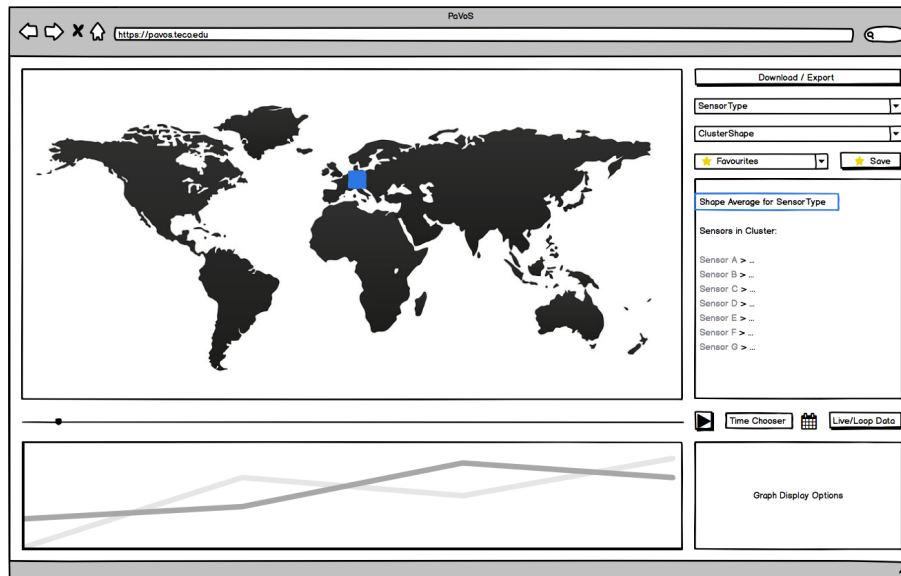


Abb. 10.5: Webinterface mit Weltansicht und ausgewähltem Cluster.

Dies ist der Startpunkt des Dienstes. Von hier aus können alle angebotenen Features bedient werden. Die Karte besteht aus einer Sammlung von Clustern, die - sofern Sensordaten darin enthalten sind - selektiert werden können, um detaillierte Daten einzusehen.

Hier ist auf der Karte ein Cluster selektiert. In der erweiterten Ansicht wird "Shape Average for SensorType" ausgewählt, so werden berechnete Durchschnittsdaten in der Detailansicht angezeigt.

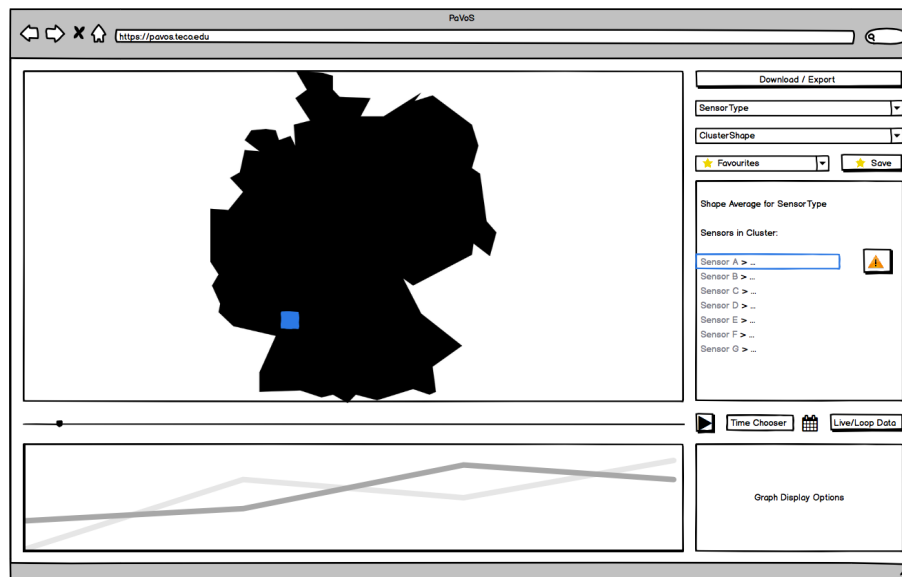


Abb. 10.6: Webinterface mit Deutschlandansicht und ausgewähltem Cluster.
Hier ist in der erweiterten Ansicht ein bestimmter Sensor selektiert,
so werden seine historischen Daten in der Detailansicht angezeigt.

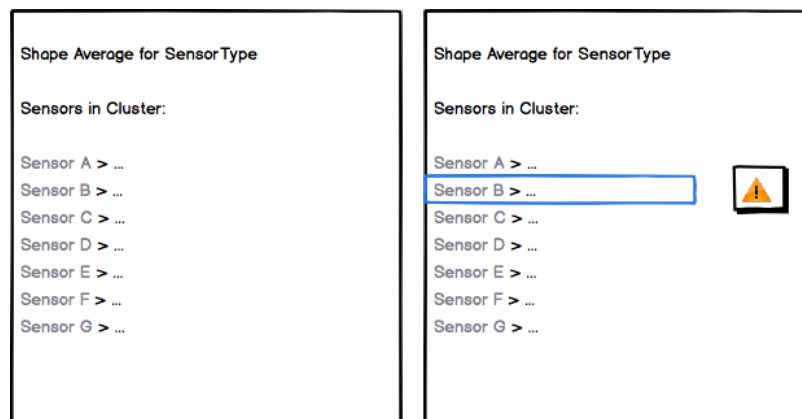


Abb. 10.7: Erweiterte Ansicht: Anzeige aller Sensoren innerhalb des ausgewählten Clusters (links). Jeder einzelne Sensor kann ausgewählt werden (rechts), um detaillierte historische Daten anzuzeigen (siehe Detailansicht). Ist ein Sensor ausgewählt, kann er von den Nutzer, unter Angabe eines Grundes, gemeldet werden, indem der Meldebutton genutzt wird (rechts von der Selektion).



Abb. 10.8: Dropdown-Listen für den Sensortyp (SensorType) und die Form des Clusters (ClusterShape)

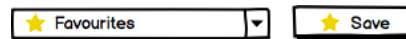


Abb. 10.9: Dropdown-Liste für die Favoriten und Button zur Speicherung der derzeitigen Selektion als neuer Favorit.

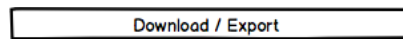


Abb. 10.10: Downloadbutton zum Export von Daten. Wird der Download angefordert, öffnet sich ein Formular, das die genaue Auswahl der gewünschten Daten ermöglicht.

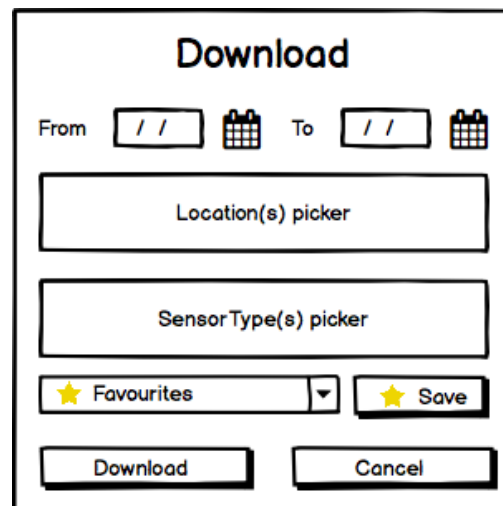


Abb. 10.11: Downloadformular zum Export von Daten. Wird der Download angefordert, wird immer zuerst der Download der derzeit ausgewählten Daten angeboten. Das Formular ermöglicht es dem Nutzer den Download nach Zeitintervallen, Lokalitäten und Sensortypen auf seine Bedürfnisse anzupassen. Alternativ können auch gespeicherte Favoriten ausgewählt, oder neue gespeichert werden.

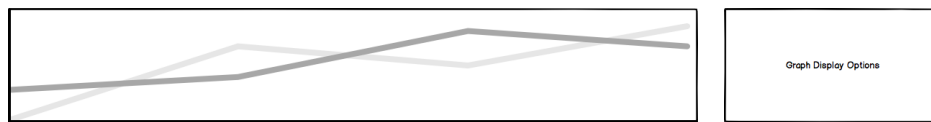


Abb. 10.12: Detailansicht: zeitlicher Verlauf spezifischer Sensordaten (Kann ein einzelner Sensor sein, oder der Schnitt eines Clusters). Zusätzlich können rechts Anzeigeoptionen für den Graphen eingestellt werden.

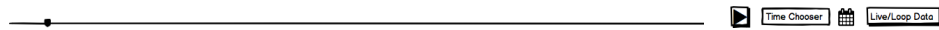


Abb. 10.13: Der Zeitregler dient dazu, historische Daten auf der Karte und der erweiterten Ansicht anzuzeigen. Befindet man sich in der Liveansicht, kann man in einen Loop wechseln, der historische Daten in einer Schleife anzeigt. Dieser kann mit dem Play/Pause Button jederzeit unterbrochen oder fortgesetzt werden. Es ist ebenfalls möglich einen gewünschten Zeitpunkt direkt einzugeben. Mit dem Liveansicht-Button kann jederzeit in wieder auf Livedaten gewechselt werden.

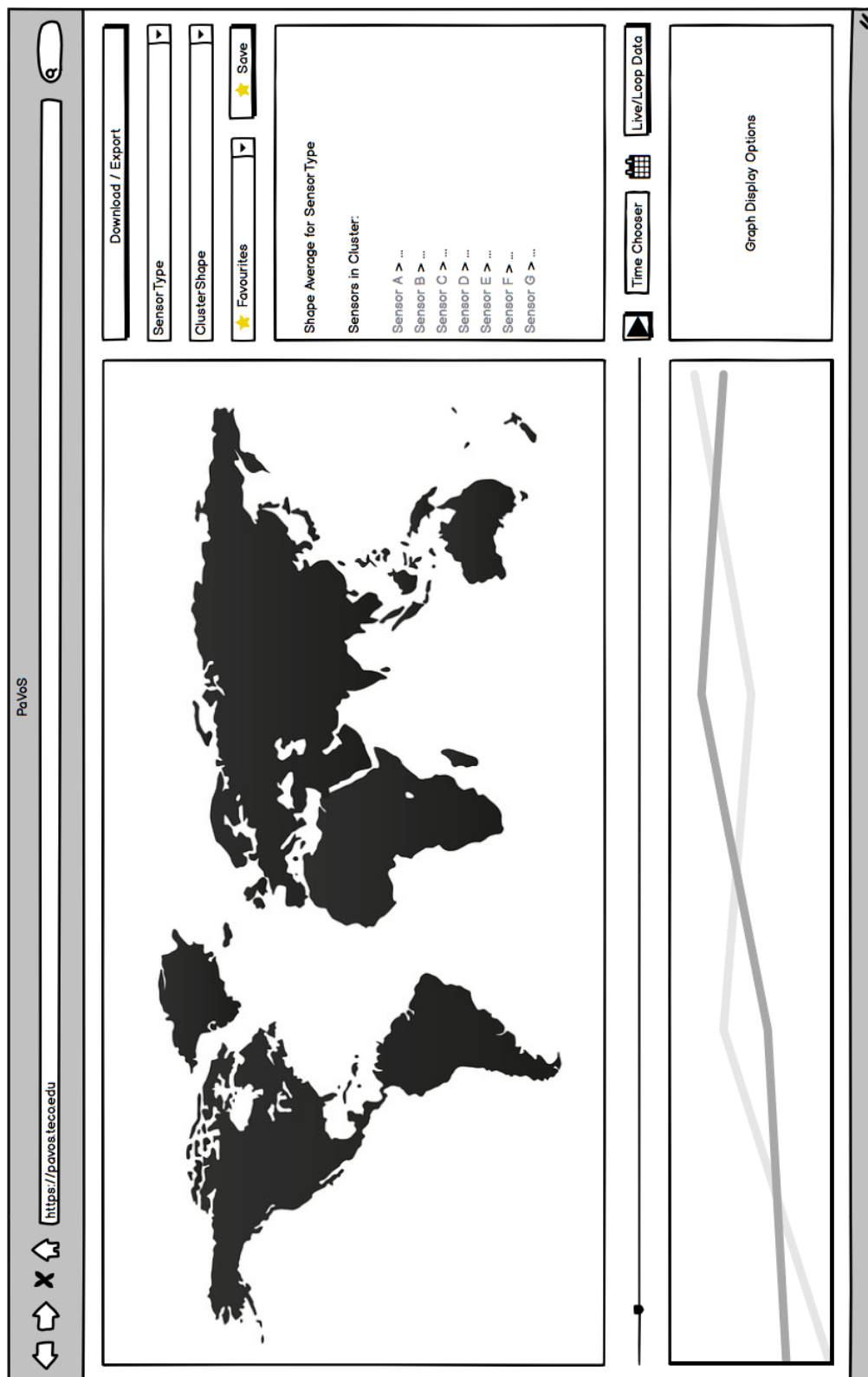


Abb. 10.14: Großansicht Webinterface mit Weltkarte

11 Anhang