

**TECO Research Group**

Marcel Köpke  
Matthias Budde  
Till Riedel

**PFLICHTENHEFT**

Version 0.1

---

# Visualizing & Mining of Geospatial Sensorstreams with Apache Kafka

---

Jean Baumgarten  
Oliver Liu  
Patrick Ries  
Erik Wessel  
Thomas Frank

21. Mai 2018

# Inhaltsverzeichnis

<b>1</b>	<b>Zielbestimmung</b>	<b>4</b>
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 . . . . .	7
<b>2</b>	<b>Produkteinsatz</b>	<b>8</b>
2.1	Anwendungsbereiche . . . . .	8
2.2	Zielgruppe . . . . .	8
<b>3</b>	<b>Produktumgebung</b>	<b>9</b>
3.1	Software . . . . .	9
3.1.1	Server . . . . .	9
3.1.2	Client . . . . .	9
3.2	Hardware . . . . .	9
3.2.1	Server . . . . .	9
3.2.2	Client . . . . .	9
<b>4</b>	<b>Funktionale Anforderungen</b>	<b>10</b>
4.1	Pflicht Funktionale Anforderungen . . . . .	10
4.1.1	Server . . . . .	10
4.1.2	Webinterface . . . . .	11
4.2	Wunsch Funktionale Anforderungen . . . . .	12
4.2.1	Server . . . . .	12
4.2.2	Webinterface . . . . .	12
<b>5</b>	<b>Produktdaten</b>	<b>14</b>
5.1	Server . . . . .	14
5.2	Webinterface . . . . .	14
<b>6</b>	<b>Nichtfunktionale Anforderungen</b>	<b>15</b>
<b>7</b>	<b>Qualitätssicherung</b>	<b>17</b>
7.1	Vorgehen bei der Qualitätssicherung . . . . .	17
7.1.1	Komponententests . . . . .	17

7.1.2	Integrationstests . . . . .	17
7.1.3	Systemtests . . . . .	17
7.2	Testszzenarien . . . . .	17
7.2.1	Komponententests . . . . .	17
	Server . . . . .	17
	Webinterface . . . . .	18
7.2.2	Integrationstests . . . . .	19
	Server . . . . .	19
	Webinterface . . . . .	20
7.2.3	Systemtests . . . . .	20
<b>8</b>	<b>Systemmodelle</b>	<b>21</b>
8.1	Anwendungsfälle . . . . .	21
8.1.1	Bedienung des Webinterface . . . . .	21
8.1.2	Bedienung der Admin-GUI . . . . .	22
8.2	Aktivitätsdiagramm . . . . .	23
<b>9</b>	<b>Entwicklungsumgebung</b>	<b>24</b>
9.1	Entwicklungstools . . . . .	24
9.2	Verwendete Technologien . . . . .	24
<b>10</b>	<b>Benutzeroberfläche</b>	<b>25</b>
10.1	Admin-GUI . . . . .	25
10.2	Webinterface . . . . .	26
<b>11</b>	<b>Anhang</b>	<b>28</b>
11.1	Glossar . . . . .	28

# 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

### 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 aktuelle und historische Sensordaten über das Webinterface darstellen lassen

- MK2030** Der Nutzer kann die Sensordaten über das Webinterface herunterladen
- MK2040** Der Nutzer kann kürzlich beobachtete Daten als Wiederholung anzeigen lassen
- MK2050** Das Webinterface unterstützt die Darstellung von erweiterten Informationen bzgl. der Sensordaten in Form von Graphen
- MK2060** Die Standardsprache des Webinterfaces ist Englisch
- MK2070** 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
- WK1050** Der Server unterstützt das Filtern von ausgegebenen und angezeigten Daten
- WK1060** Der Server gibt aussagekräftige Fehlermeldungen aus

### 1.2.2 Frontend (Webinterface)

- WK2000** Der Nutzer kann die Karte in vordefinierten Detaillierungsgraden darstellen lassen
- WK2010** Die Genauigkeit der Darstellung von Clustern wird entsprechend der Zoom-Stufe angepasst
- WK2020** Die Approximation von Clustern führt nicht zu größeren Diskrepanzen oder Wartezeiten
- WK2030** Die Erzeugung der grafischen Komponenten erfolgt zeitnah und parallel zur Darstellung der Benutzeroberfläche selbst
- WK2040** Der Nutzer kann verschiedene Sensordatentypen (Feinstaub, Wind, Temperatur, etc.) an- bzw. ausschalten

- WK2050** Der Nutzer kann Standorte als Favoriten abspeichern
- WK2060** Der Nutzer kann Gebiete als Kombination von Clustern auswählen und diese als Favoriten abspeichern
- WK2070** Der Nutzer kann favorisierte Standorte/Gebiete auswählen um schnell und einfach die optimale Ansicht des gewählten Standortes/Gebiets dargestellt zu bekommen
- WK2080** Der Nutzer kann favorisierte Standorte/Gebiete in einer grafischen Darstellung vergleichen, z.B. nebeneinander als Split-Panel
- WK2090** Der Nutzer kann zwischen mehreren Ansichten wechseln, um die Standardansicht oder geladene Szenarien darzustellen
- WK2100** Der Nutzer kann einzelne Sensoren/Cluster aus der Darstellung ausschließen
- WK2110** Der Nutzer kann Anzeigefilter einstellen
- WK2120** Der Nutzer kann Anzeigefilter als Favoriten speichern
- WK2130** Der Nutzer kann fehleranfällige Sensoren melden
- WK2140** Dem Nutzer wird eine Warnung angezeigt, wenn das Abrufen von Sensordaten nicht möglich ist
- WK2150** Bei fehlenden Sensordaten kann ein Standardwert oder ein approximierter Wert anhand von Umgebungsinformationen ermittelt und angezeigt werden
- WK2160** Der Nutzer kann zwischen einer automatischen und einer manuellen Aktualisierung von Echtzeitsensordaten wechseln
- WK2170** Der Nutzer kann die Aktualisierungsrate von Echtzeitdaten einstellen
- WK2180** Der Nutzer kann Sensordaten in vielen gebräuchlichen Formaten herunterladen
- WK2190** Der Nutzer kann Benachrichtigungen mit Bedingungen einstellen, die dem Nutzer die aktuellen Daten melden, falls die Bedingungen erfüllt sind
- WK2200** Auf Graphen werden Bedingungen für Benachrichtigungen als Grenzwerte angezeigt
- WK2210** Der Nutzer kann Töne und Farben für Benachrichtigungen festlegen
- WK2220** Für standardisierte Displayauflösungen wird immer eine benutzerfreundliche Darstellung angeboten

**WK2230** Der Nutzer kann eine Anzeigesprache auswählen

**WK2240** Die Anwendung speichert automatisch die Einstellungen und Favoriten des Nutzers über Browsersessions hinweg (Cookies)

### 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

### 2.2 Zielgruppe

- jegliche Personen, die Sensordaten auswerten und visuell darstellen wollen, wie z.B. Wissenschaftler, Sensorenenthusiasten und Datenfanatiker
- generelle Bevölkerung, um Informationen über aktuelle Werte von Sensoren zu erhalten, z.B. über Luftverschmutzung, Temperatur usw.



## 3 Produktumgebung

Das Produkt läuft mit Client-Server Architektur im "Thin Client"Konzept.

### 3.1 Software

#### 3.1.1 Server

- Linux-Betriebssystem (min. Debian 8)
- Java-Anwendung mit Java 8
- Java Web-Dienst mit Eclipse Jetty
- Datenbank mit PostgreSQL

#### 3.1.2 Client

- Webbrowser, Referenzstandard Mozilla Firefox 57

### 3.2 Hardware

#### 3.2.1 Server

- Aus dem Internet erreichbarer, leistungsstarker Rechner

#### 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 Englisch als Systemsprache**

Das System wird nur die Englische Sprache unterstützen und alles in Englisch anzeigen.

**PF1010 Sensordaten empfangen**

Beim ersten öffnen der Server Admin GUI soll sich der Nutzer aufgefordert werden einen Kafka Daten stream mit Sensordaten einzufügen oder komprimierte Sensordaten zu importieren. Beim Einfügen oder importieren wird von dem Programm überprüft ob es sich für das Programm lesbare Formate handelt,

**PF1020 Sensordaten werden gesichert**

Die Sensordaten werden extrahiert und in mehr Streams aufgeteilt und gruppiert und in einem Datenerhaltungssystem gespeichert.

**PF1030 Verarbeitung der Daten für Kartenansicht**

Der Dienst verarbeitet so vor, dass sie für spätere Verwendungen genutzt werden wie die Weiterleitung an den Webinterface und auf einer gerasterten Karte angezeigt werden können.

**PF1040 Verarbeitung der Daten für Mittelwertberechnung**

Der Dienst verarbeitet die Daten so vor, dass Mittelwerte berechnet werden von den Sensordaten um diese zur Analyse weiter an das Web-Interface zu geben.

**PF1050 Verarbeitung der Daten für Export**

Der Dienst verarbeitet die Daten so vor, dass sie komprimiert exportiert werden können aus dem dem Datenerhaltungssystem damit diese im Web-Interface downloadet werden können.

**PF1060 Kann gespeicherte Daten wieder abfragen**

Der Dienst kann die vorbereiteten Daten **PF1030,PF1040** jeder Zeit wieder aufrufen und weiterleiten an das Webinterface.

**PF1070 Daten werden an Webinterface gesendet**

Die Daten aus dem Datenerhaltungssystem und vom Kafka Stream können an das Webinterface gesendet werden.

**PF1080 Hinzufügen weiterer Module**

Der Dienst ist so aufgebraucht, dass es dem Nutzer und Programmierer möglich sein wird, auf dem einzelnen Interface zuzugreifen und so neue Module für den Dienst zu entwickeln um das System zu erweitern. Ein Beispiel wäre neue Exportformate hinzuzufügen oder ein Modul zwischen Server-Dienst und Webinterfaces zu setzen.

**PF1090 Der Dienst unterstützt skalar- und vektorwertige Sensortypen**

Der Dienst unterstützt beim Kafka Stream und beim Import von Sensor Daten nur Skalar Sensortypen und vektorwertige Sensortypen. Falls andere Sensortypen im Stream entdeckt werden, dann wird dem Nutzer eine Anleitung angezeigt wie er selbst einen neuen Sensortyp hinzufügen kann.

### 4.1.2 Webinterface

**PF1100 Sensordaten werden dargestellt**

Nach dem die Sensordaten **PF1070** empfangen wurden auf dem Webinterface können sie auf dem Webinterface dargestellt werden

**PF1110 Darstellung der Daten**

Auf dem Webinterface kann man die Sensordaten in verschiedenen Rastern (Beispiel: Quadratenraster) auf einer Weltkarte darstellen mit vordefinierten Formen wie zum Beispiel: Quadrate oder Hexagone.

**PF1120 Darstellung nur auf Deutschland**

Auf dem Webinterface kann man das Land Deutschland anzeigen lassen und sonst keine anderen Länder. So ist die Ansicht nur auf Deutschland beschränkt.

**PF1130 Sensordaten anzeigen**

Auf dem Webinterface kann man aktuellsten Daten von Sensoren anzeigen lassen auf der Karte und auch ältere Daten von Sensoren anzeigen lassen auf der Karte.

**PF1140 Export von Daten**

Auf dem Webinterface kann man die aktuellen Daten und die historischen Daten exportieren in einem vordefinierten Format in Form eines Download.

**PF1150 Wiederholung anzeigen**

Auf dem Webinterface kann man aktuelle und historische Daten mit Hilfe von einem Slider als Wiederholung anzeigen lassen.

**PF1160 Mehrere Instanzen des Webinterface**

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

**PF1170 Detail Ansicht von Sensoren**

Auf dem Webinterface kann man von einzelnen Sensoren eine Detailansicht darstellen um so nähere Informationen über den Sensor zu erhalten. Es wird dann auch eine Möglichkeit geben genauere Statistiken von dem Sensor in Form von Graphen und Text anzuzeigen.

## 4.2 Wunsch Funktionale Anforderungen

### 4.2.1 Server

**WF1000 Der Dienst gibt Fehlermeldungen aus**

Der Dienst bei Fehlschlägen von Prozessen Fehlermeldungen aus und schreibt es in eine Log Datei. Beispiele dazu wären, dass beim Laden von Fehlerhaften Stream das Programm eine Fehlermeldung aus und schreibt den Grund in eine Logdatei oder falls beim sichern der Daten im Datenerhaltungssystem zu Fehlern kommt, wie dass die Gruppierung fehlschlägt oder kein gültiges Datenerhaltungssystem verfügbar ist, wird auch eine Fehlermeldung ausgegeben und in eine Logdatei geschrieben.

**WF1010 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 und ihm die Möglichkeit diese zu löschen.

**WF1020 Mehrere Sprachen**

Der Dienst kann mit mehreren Sprachen außer Englisch umgehen und die Anzeigesprache kann auch entsprechend geändert werden.

### 4.2.2 Webinterface

**WF1030 Filtern von Daten**

Man kann auf der Admin GUI die Daten oder Streams filtern, damit nur bestimmte Daten weiter verarbeitet werden.

**WF1040 Vordefinierten Detaillierungsgraden**

Der Nutzer kann im Webinterface auf der Karte Zoomen in verschiedenen Stufen um so die Sensordaten genauer in bestimmten Regionen zusehen.

**WF1050 Anpassung der Daten**

Bei gezoomter Ansicht werden die Daten automatisch angepasst und den

Durchschnitt genommen um eine angenehme und richtige Darstellungen zu garantieren.

## 5 Produktdaten

### 5.1 Server

#### **PD1000 Sensordaten**

Sowohl vorverarbeitete Sensordaten als auch Echtzeitsensordaten werden vom Server bis zu einer gewissen Menge gespeichert.

### 5.2 Webinterface

#### **PD2000 Cookies**

Die vom Nutzer gesetzten Einstellungen (z.B. Zoom-Stufe, an- bzw. ausgeschaltete Sensordatentypen), Favoriten (z.B. Gebiete, Anzeigefilter) und Benachrichtigungen werden in Form von Cookies im Browser des Nutzers gespeichert.

## 6 Nichtfunktionale Anforderungen

**NF1000 Verzögerung von neuen Daten**

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

**NF1010 Parallele Zugriffe**

Das Webinterface muss von bis zu 100 Nutzern gleichzeitig bedient werden können.

**NF1020 Speicherung von Echtzeitsensordaten**

Echtzeitsensordaten werden bis zu einer Gesamtgröße von 10 GB oder maximal 24 Stunden in einem Datenerhaltungssystem gespeichert, je nachdem welcher Fall zuerst eintritt.

**NF1030 Skalierbarkeit**

Der Server startet automatisch neue Threads um Sensordaten und Anfragen des Webservers zu verarbeiten. Die maximale Anzahl an Threads beträgt  $4 \cdot \text{Anzahl CPU Kerne}$ .

**NF1040 Stabilität**

Der Server muss im normalen Betrieb mindestens eine Woche ohne Absturz laufen. Fehler in den Eingabedaten, wie ungültige Werte oder Syntaxfehler führen nicht zu einem Absturz des Servers.

**NF1050 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. (→ Kriterium?)

**NF1060 Ladezeiten der Darstellungselemente**

Ein ungecacher Zugriff auf das Webinterface lädt Darstellungselemente innerhalb von 8 Sekunden. Ein Zugriff auf das Webinterface mit gecachten Darstellungselementen darf nicht zu Ladezeiten über 4 Sekunden führen.

**NF1070 Clusterberechnung**

Die Approximation der Cluster darf nicht zu einer Abweichung von mehr als 5% von Ergebnissen einer akkuraten Berechnung führen und darf insgesamt für bis zu 1000 sichtbare Cluster nicht mehr als 2 Sekunden dauern. Für mehr sichtbare Cluster steigt die Berechnungszeit proportional.

**NF1080 Größenanpassung**

Das Webinterface soll Responsive Webdesign umsetzen.

**NF1090 Aktualisierungsgeschwindigkeit**

Die Karte aktualisiert Echtzeitsensordaten alle 10 Sekunden.



## 7 Qualitätssicherung

### 7.1 Vorgehen bei der Qualitätssicherung

Wir streben an, bereits während der Implementierung die ersten Testfallszenarien zu schreiben, um Probleme später zu vermeiden. Die Testphase gliedert sich in mehrere Phasen, damit sowohl einzelne Komponenten als auch ihre Zusammenarbeit getrennt getestet werden können. Schlussendlich wird das Produkt als Ganzes im dafür vorgesehenen Einsatzbereich getestet.

#### 7.1.1 Komponententests

In den Komponententests werden der Server und das Webinterface unabhängig voneinander getestet.

#### 7.1.2 Integrationstests

In den Integrationstests wird die Kommunikation zwischen den Komponenten geprüft. Es wird darauf geachtet, dass die Funktion und Implementierung der Schnittstellen gesichert ist.

#### 7.1.3 Systemtests

Die Software wird unter realen Bedingungen und in einer geeigneten Umgebung getestet.

### 7.2 Testszenarien

#### 7.2.1 Komponententests

##### Server

##### **TK1000 Englisch als Systemsprache**

Der Administrator versucht eine andere Sprache zu verwenden. Der Server akzeptiert nur vordefinierte Sprachen und verweigert somit die Aktion.  
[PF1000]

##### **TK1010 Sensordaten empfangen**

Der Administrator öffnet zum ersten Mal die Admin-GUI. Daraufhin wählt er aus:

- Server mit einem Kafka Data-Stream verbinden.
- Server aus einer Datei gesicherte Sensordaten lesen lassen.

Abschließend überprüft das Programm, ob die eingehenden Daten korrekt formatiert sind. [PF1010]

**TK1020 Sensordaten werden gesichert**

Der Administrator sichert Sensordaten in der Admin-GUI. Anschließend ruft der Administrator die Daten aus dem Datenerhaltungssystem wieder ab. [PF1020]

**TK1030 Hinzufügen weiterer Module**

Der Administrator hat ein Modul entwickelt, das

- ein neues Exportformat hinzufügt und dessen Import ermöglicht.
- einen neuen Kommunikationsdienst zwischen Server und Webinterface bereitstellt.

Dieses Modul wird in den Server integriert. Abschließend ruft der Administrator

- exportierte Daten ab und importiert sie wieder über die Admin-GUI.
- das Resultat des Kommunikationsdienstes ab und prüft dessen Korrektheit.

[PF1080]

**TK1040 Unterstützte Sensordatentypen**

Der Administrator legt Daten an das System an, die nicht skalar- bzw. vektorwertig sind. Anschließend erhält er eine Benachrichtigung, dass die Sensordatentypen nicht akzeptiert werden und er wird aufgefordert, neue Daten anzulegen. Hierzu wird ihm eine Anleitung dargestellt, die den Prozess anschaulich visualisiert. [PF1090]

**Webinterface****TK2000 Eingrenzung des Darstellungsbereiches der Sensordaten**

Der Administrator legt korrekte Daten an das Webinterface an und ruft dieses in einem kompatiblen Browser auf. Beim ersten Besuch der Seite werden die Daten im Webinterface in einem Raster angezeigt, das sich aus vordefinierten Formen zusammensetzt. Mehrere vorausgewählte Sensordatentypen, die in einem Filter festgelegt sind, werden visualisiert. Weiterhin sind beim ersten Besuch der Seite der anzuzeigende Detaillierungsgrad, sowie das zu fokussierende Gebiet vordefiniert. Der Administrator

- wählt andere Formen aus
- wählt einen anderen Filter für Sensordatentypen aus
- wählt einen anderen Detaillierungsgrad aus

- wählt ein anderes Gebiet aus (z.B. Deutschland)  
, wobei alle Darstellungen korrekt die Daten widerspiegeln. [PF1110 & PF1120]

**TK2010 Mehrfache Instanzen**

Der Administrator legt korrekte Daten an das Webinterface an und ruft dieses in einem kompatiblen Browser mehrfach auf. Die Anzahl der geöffneten Instanzen führt nicht zu fehlerhaften Darstellungen der Sensordaten und ist gut skalierbar. Der Administrator sieht unter gleichen Einstellungen der Instanzen keine unterschiedlichen Ergebnisse. [PF1160]

**TK2020 Änderung der Ansicht der Sensordatendarstellung**

Der Administrator legt korrekte Daten an das Webinterface an und ruft dieses in einem kompatiblen Browser auf. Anschließend wählt er eine der bereitgestellten Ansichten aus. Die zur Verfügung stehenden Ansichten enthalten Unteransichten wie:

- Kartenansicht
- Graphenansicht
- Textansicht

Eine Ansicht setzt sich immer aus mehreren Unteransichten zusammen und wird so angewandt, dass auch die Benutzeroberfläche weiterhin vollständig dargestellt ist. Beim ersten Besuch der Seite wird eine vordefinierte Ansicht gewählt. Die Ansicht wird dann in die ausgewählte Variante geändert, ohne die Darstellung einzelner Elemente zu verändern. [PF1170]

**7.2.2 Integrationstests****Server****TI1000 Verarbeitung der Daten**

Der Administrator legt korrekte Daten an das System an und ruft das Webinterface in einem kompatiblen Browser auf. Er lässt den Dienst auf diesen Daten arbeiten, sodass sie vor der Nutzung vorverarbeitet werden und

- sieht sich die Kartenansicht des Webinterfaces an, die die Daten im gewählten Raster korrekt anzeigt.
- sieht sich die Darstellung der Bereiche ohne Messwerte an. Diese zeigen den korrekt errechneten Mittelwert an.
- lädt sich die Daten über den korrespondierenden Menüpunkt des Webinterfaces herunter und überprüft dessen Korrektheit.

[PF1030, PF1040, PF1050 & PF1060]

**Webinterface****TI2000 Sensordatendarstellung**

Der Benutzer öffnet das Webinterface in einem kompatiblen Browser. Empfängt das Webinterface Daten vom Server, werden diese korrekt dargestellt. Der Benutzer kann zwischen

- aktuellen Sensordaten
- historischen Sensordaten
- zeitlichen Wiederholungen von Sensordaten mit Zeit-Kontrollregler

wählen. Abschließend lädt der Benutzer die Daten aus dem Webinterface in einem der gegebenen Formate herunter um sie dann als Sensordaten-Datei wieder zu importieren und darstellen zu lassen. [PF1100, PF1130, PF1140 & PF1150]

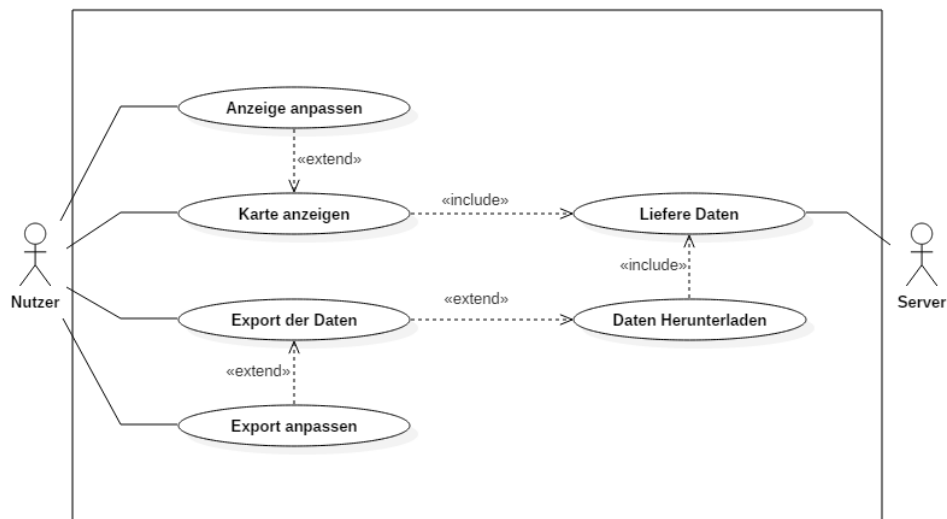
**7.2.3 Systemtests**

Das System wird von uns unter realen Bedingungen mit weiterem Testen auf Vollständigkeit und Korrektheit überprüft. Es erfolgt ebenfalls eine Überprüfung der intuitiven Bedienbarkeit.

## 8 Systemmodelle

### 8.1 Anwendungsfälle

#### 8.1.1 Bedienung des Webinterface

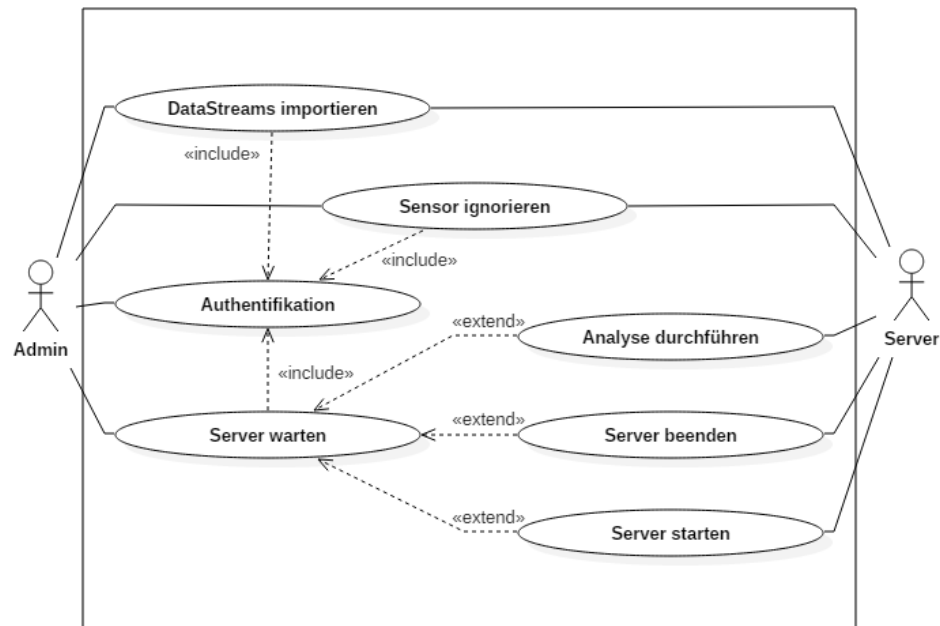


Dieser Anwendungsfall beschreibt die Bedienung des Webinterface. Dem Nutzer stehen folgende Aktionen zur Verfügung:

- 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 bereitgestellt werden. 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 Admin-GUI



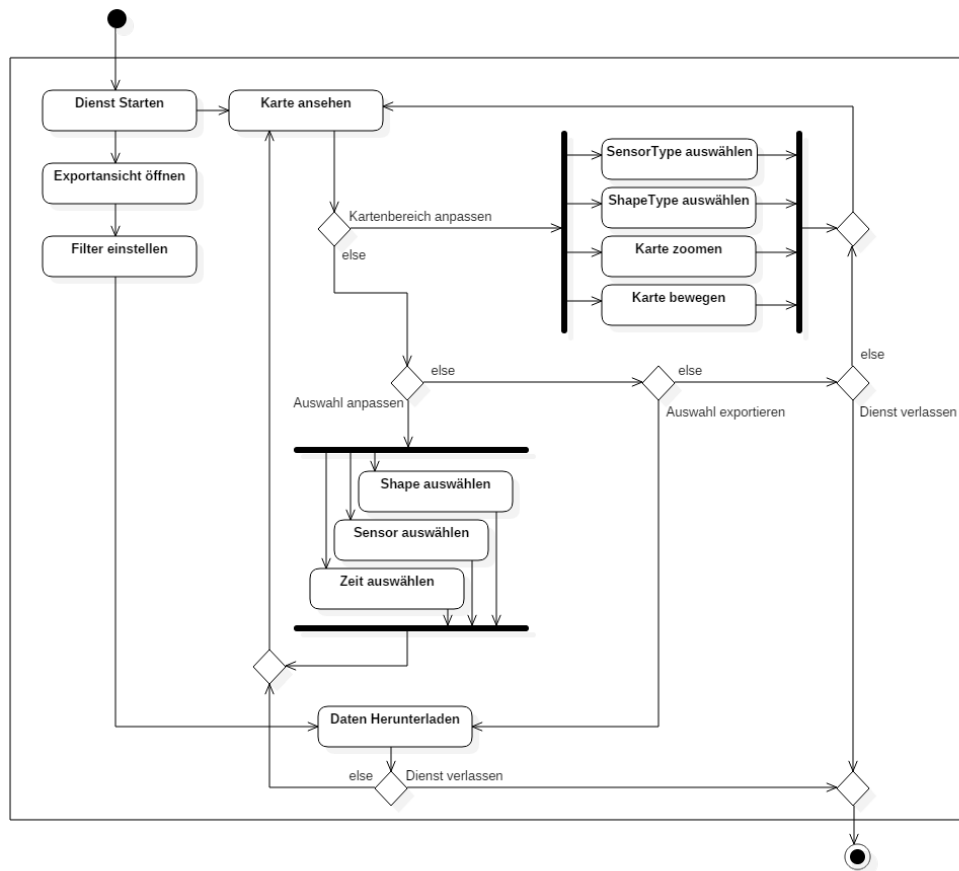
Dieser Anwendungsfall beschreibt die Bedienung der Admin-GUI. Dem Admin stehen folgende Aktionen zur Verfügung:

- Authentifikation
- Serverwartung
- DataStreams importieren
- Sensoren ignorieren

Um Zugriff auf die Funktionen des Admin-GUIs zu erhalten, muss sich der Admin zunächst authentifizieren. War dies erfolgreich, so kann der Server gewartet werden. Der Admin kann ihn starten sowie beenden um Analysen durchzuführen und eventuelle Fehler zu beheben. Weiterhin lassen sich DataStreams importieren, wodurch der Server diese verarbeiten und weiterverbreiten kann. Dadurch können auch historische Datenbestände in den Datensatz aufgenommen werden. Auch hat der Admin die Möglichkeit auszuwählen, welche Sensoren ignoriert werden, also wessen Daten nicht verbreitet und in den Datenbestand aufgenommen werden sollen.

## 8.2 Aktivitätsdiagramm

### 8.2.1



## 9 Entwicklungsumgebung

### 9.1 Entwicklungstools

<b>Java IDE</b>	Eclipse
<b>Projektmanagement</b>	Trello
<b>Textverarbeitung</b>	L <sup>A</sup> T <sub>E</sub> X
<b>TeX-Distribution</b>	MiKTeX
<b>L<sup>A</sup>T<sub>E</sub>X-Editor</b>	TeXstudio
<b>Versionskontrolle</b>	Git
<b>UML Tool</b>	StarUML

### 9.2 Verwendete Technologien

<b>Programmiersprache</b>	Java 8
<b>Web-Framework</b>	?
<b>Webserver</b>	?
<b>Datenbank</b>	?



# 10 Benutzeroberfläche

## 10.1 Admin-GUI

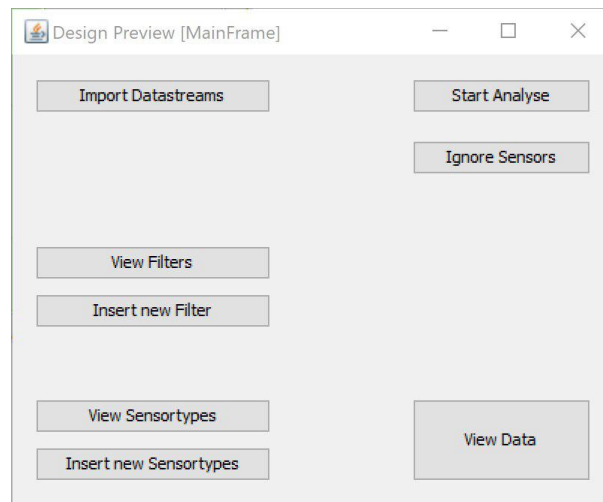


Abbildung 10.1: Administrator-GUI

## 10.2 Webinterface



Abbildung 10.2: Webinterface mit Weltansicht

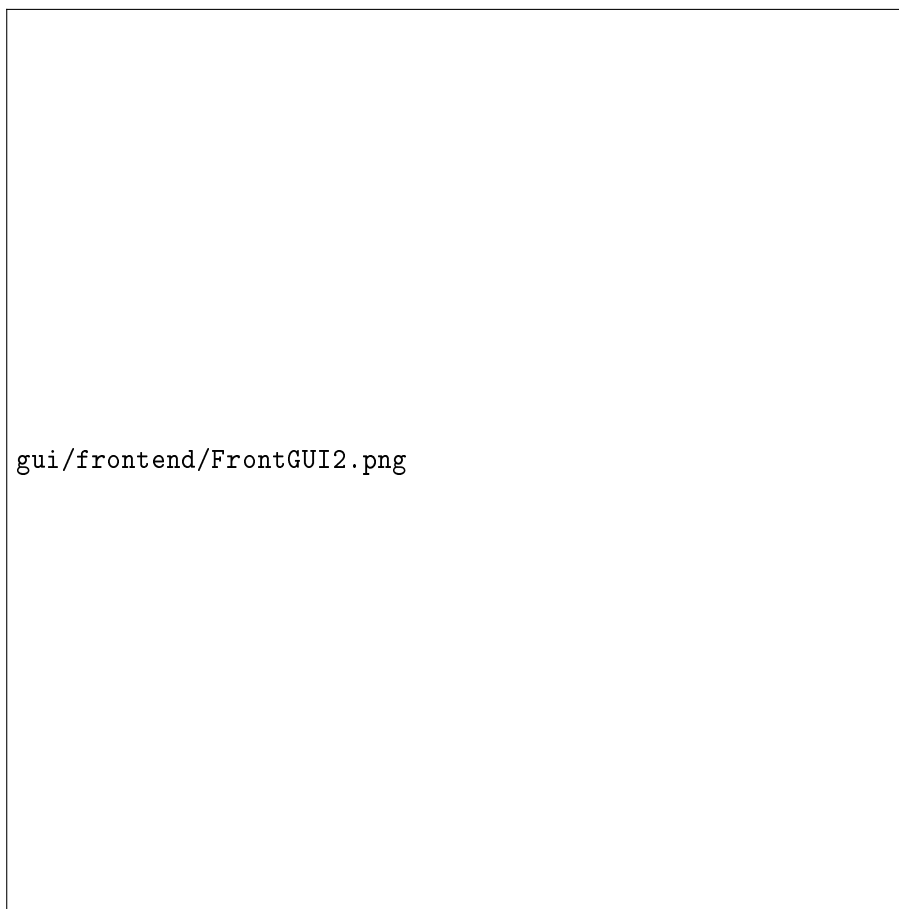


Abbildung 10.3: Webinterface mit Deutschlandansicht

# 11 Anhang

## 11.1 Glossar