

**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

25. 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 . . . . .                | 6         |
| <b>2</b> | <b>Produkteinsatz</b>                         | <b>7</b>  |
| 2.1      | Anwendungsbereiche . . . . .                  | 7         |
| 2.2      | Zielgruppe . . . . .                          | 7         |
| <b>3</b> | <b>Produktumgebung</b>                        | <b>8</b>  |
| 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         |
| <b>4</b> | <b>Funktionale Anforderungen</b>              | <b>9</b>  |
| 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        |
| <b>5</b> | <b>Produktdaten</b>                           | <b>15</b> |
| 5.1      | Server . . . . .                              | 15        |
| 5.2      | Webinterface . . . . .                        | 15        |
| <b>6</b> | <b>Nichtfunktionale Anforderungen</b>         | <b>16</b> |
| <b>7</b> | <b>Qualitätssicherung</b>                     | <b>18</b> |
| 7.1      | Vorgehen bei der Qualitätssicherung . . . . . | 18        |
| 7.1.1    | Komponententests . . . . .                    | 18        |

|           |  |           |
|-----------|--|-----------|
| 7.1.2     | Integrationstests . . . . .                | 18        |
| 7.1.3     | Systemtests . . . . .                      | 18        |
| 7.2       | Testszzenarien . . . . .                   | 18        |
| 7.2.1     | Komponententests . . . . .                 | 18        |
|           | Server . . . . .                           | 18        |
|           | Webinterface . . . . .                     | 19        |
| 7.2.2     | Integrationstests . . . . .                | 20        |
|           | Server . . . . .                           | 20        |
|           | Webinterface . . . . .                     | 21        |
| 7.2.3     | Systemtests . . . . .                      | 21        |
| <b>8</b>  | <b>Systemmodelle</b>                       | <b>22</b> |
| 8.1       | Anwendungsfälle . . . . .                  | 22        |
| 8.1.1     | Bedienung des Webinterface . . . . .       | 22        |
| 8.1.2     | Bedienung der Konfigurations-GUI . . . . . | 23        |
| 8.2       | Aktivitätsdiagramm . . . . .               | 24        |
| <b>9</b>  | <b>Entwicklungsumgebung</b>                | <b>26</b> |
| 9.1       | Entwicklungstools . . . . .                | 26        |
| 9.2       | Verwendete Technologien . . . . .          | 26        |
| <b>10</b> | <b>Benutzeroberfläche</b>                  | <b>27</b> |
| 10.1      | Admin-GUI . . . . .                        | 27        |
| 10.2      | Webinterface . . . . .                     | 28        |
| <b>11</b> | <b>Anhang</b>                              | <b>29</b> |
| 11.1      | Glossar . . . . .                          | 29        |

# 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 Verzögerung von neuen Daten**

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

**NF1040 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.

**NF1050 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.

**NF1060 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.

**NF1070 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.



**NF1080 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.

**NF1090 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.

**NF1100 Ladezeiten der Darstellungselemente**

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

**NF1110 Genauigkeit der Clusterberechnung**

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

**NF1120 Geschwindigkeit der Clusterberechnung**

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

**NF1130 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]

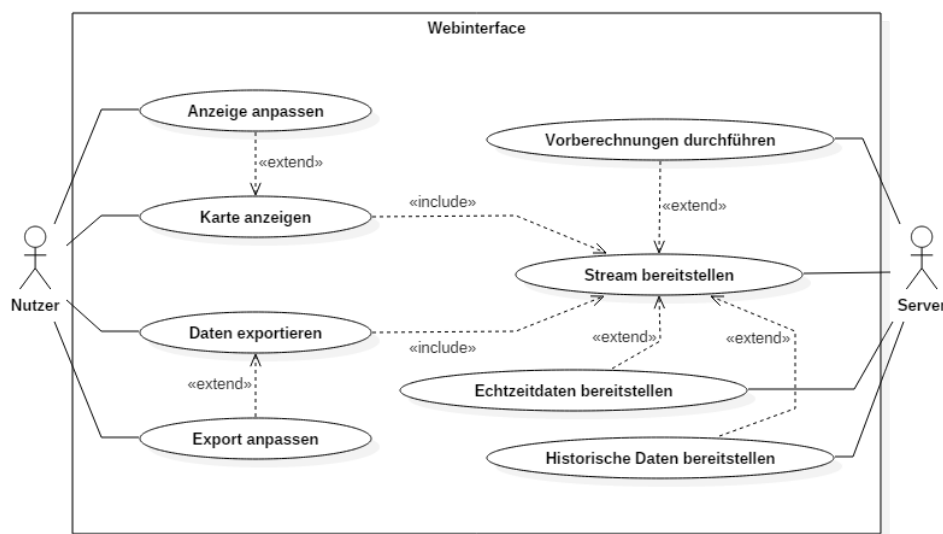
**7.2.3 Systemtests**

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



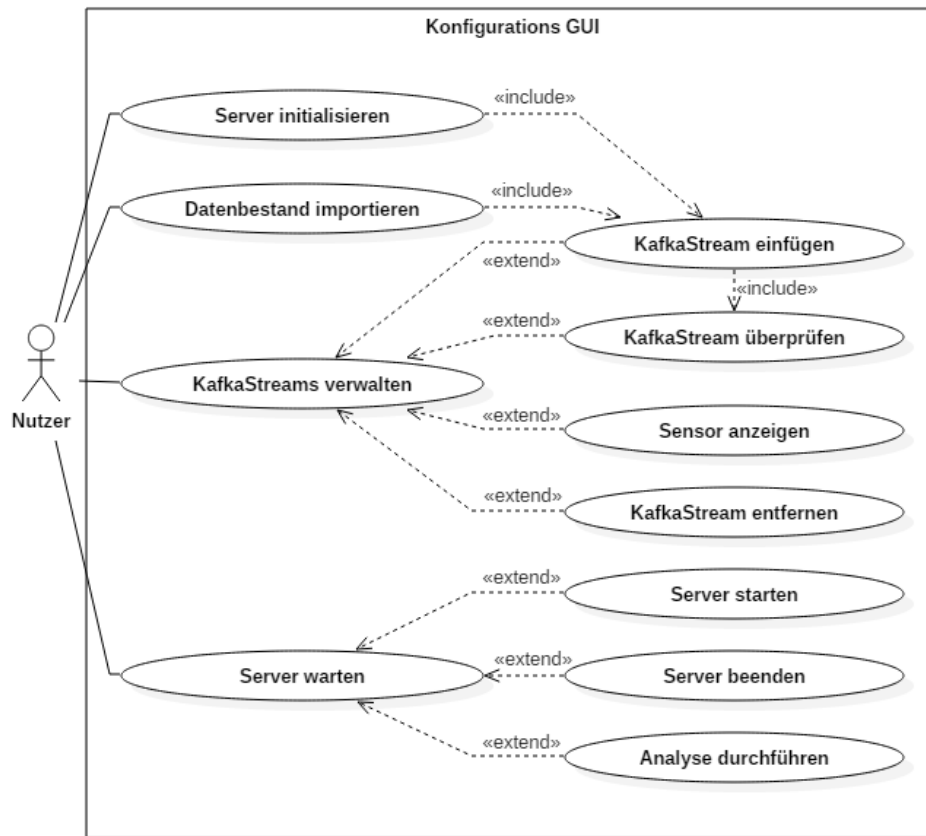
Dieser Anwendungsfall beschreibt die Bedienung des Webinterface. Dem Nutzer stehen folgende Aktionen zur Verfügung:

- 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 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:

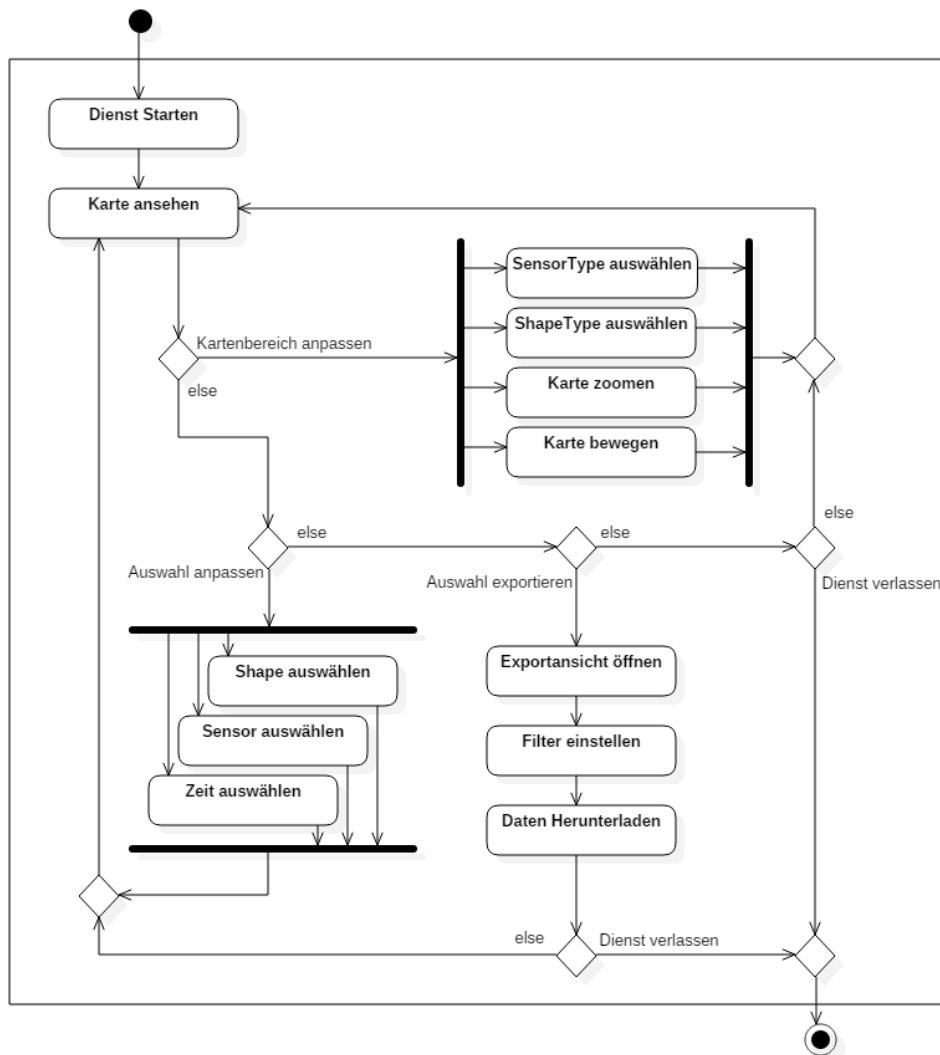
- Server initialisieren
- KafkaStreams verwalten
- Datenbestand importieren
- Server warten

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
- ShapeType/ClusterType auswählen
- Kartenbereich zoomen
- Kartenbereich verschieben

Die Selektion kann angepasst werden:

- Selektion eines Shapes/Clusters
- Selektion eines Sensors
- Selektion eines Datums

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

|   |                                 |
|---|---------------------------------|
| <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

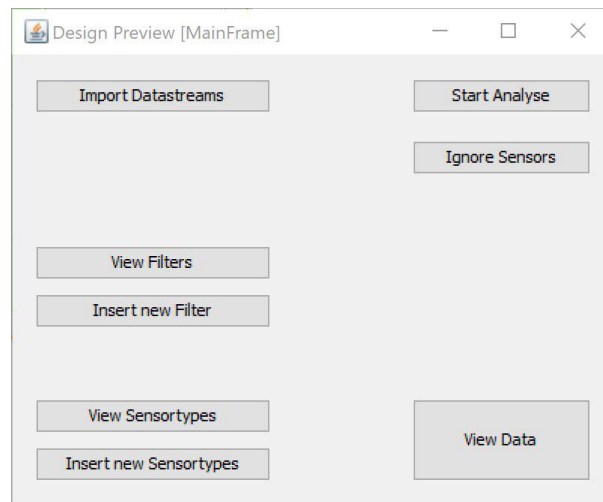


Abbildung 10.1: Administrator-GUI

## 10.2 Webinterface

Abbildung 10.2: Webinterface mit Weltansicht

Abbildung 10.3: Webinterface mit Deutschlandansicht

# 11 Anhang

## 11.1 Glossar

**Apache Kafka** Ein Open-Source-Project der Apache Software Foundation. Dient der Speicherung und Verarbeitung von großen Datenströmen.

**Client** Nimmt angebotene Dienste des Servers wahr.

**Cluster** Eine beliebig große räumliche Ballung und Zusammenfassung von Sensoren und den dazugehörigen Messwerten.

**Clustershape** Eine geometrische Form, die einen bestimmten Raum und darin enthaltene Sensoren überdeckt und durch seine Attribute, beispielsweise Farbe, Sensordaten darstellt.

**Cookie** Eine Textinformation, die durch eine Webseite beim Besucher abgelegt wird. Diese kann im weiteren Verlauf von der Seite verarbeitet werden, beispielsweise zur Nutzeridentifikation oder für die Warenkorbfunktion.

**CSV** Comma-separated values (.csv) ist ein Dateiformat und dient der Speicherung und dem Austausch einfach strukturierter Daten.

**Datenstrom** Ein kontinuierlicher Fluss von Datensätzen.

**Filter** Eine Beschränkung des Datenbestandes nach bestimmten Kriterien.

**FROST Server** Der Fraunhofer Open Source SensorThings API Server ist eine Implementierung des OGC Standards und ermöglicht die effiziente Verarbeitung von Sensordaten im "Internet of Things".

**Internet of Things** Sammelbegriff für die Vernetzung von Gegenständen mit dem und durch das Internet.

**MQTT** Message Queue Telemetry Transport ist ein offenes Nachrichtenprotokoll für Maschine-zu-Maschine Kommunikation.

**NetCDF** Das Network Common Data Format (.nc, .cdf) ist ein standardisiertes Dateiformat für den Austausch wissenschaftlicher Daten.

**OGC** Das Open Geospatial Consortium, eine gemeinnützige Organisation die sich auf Standardisierungen bezüglich der computergestützten Verarbeitung insbesondere von Geodaten spezialisiert.

**Outlier** Ein Messwert oder ein Befund, der nicht in eine erwartete Messreihe passt.

**Raster** Eine Aufteilung der Karte in Shapes zur besseren Visualisierung.

**Sensor** Ein technisches Bauteil, dass physikalische oder chemische Eigenschaften seiner Umgebung qualitativ oder quantitativ erfassen kann.

**Sensordatentyp** Die Art des Messwerts, der durch einen Sensor erfasst wird, z.B Feinstaub, Temperatur, Luftfeuchtigkeit oder Windstärke.

**Server** Ein Computer/Computerprogramm, dass Dienstprogramme, Daten oder andere Ressourcen für Clients bereitstellt.

**Webinterface** Eine Programmschnittstelle, die über einen Internetbrowser angesprochen werden kann.