



**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

16. Mai 2018

# Inhaltsverzeichnis

|           |                                       |           |
|-----------|---------------------------------------|-----------|
| <b>1</b>  | <b>Zielbestimmung</b>                 | <b>3</b>  |
| 1.1       | Musskriterien . . . . .               | 3         |
| 1.1.1     | Backend (Server) . . . . .            | 3         |
| 1.1.2     | Frontend (Webinterface) . . . . .     | 3         |
| 1.2       | Wunschkriterien . . . . .             | 4         |
| 1.2.1     | Backend (Server) . . . . .            | 4         |
| 1.2.2     | Frontend (Webinterface) . . . . .     | 4         |
| 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         |
| 3.2       | Hardware . . . . .                    | 8         |
| <b>4</b>  | <b>Funktionale Anforderungen</b>      | <b>9</b>  |
| <b>5</b>  | <b>Nichtfunktionale Anforderungen</b> | <b>11</b> |
| 5.1       | Test . . . . .                        | 11        |
| <b>6</b>  | <b>Qualitätssicherung</b>             | <b>12</b> |
| 6.1       | Test . . . . .                        | 12        |
| <b>7</b>  | <b>Systemmodelle</b>                  | <b>13</b> |
| 7.1       | Test . . . . .                        | 13        |
| <b>8</b>  | <b>Entwicklungsumgebung</b>           | <b>14</b> |
| 8.1       | Entwicklungstools . . . . .           | 14        |
| 8.2       | Verwendete Technologien . . . . .     | 14        |
| <b>9</b>  | <b>Benutzeroberfläche</b>             | <b>15</b> |
| 9.1       | Webinterface . . . . .                | 15        |
| 9.2       | Admin-GUI . . . . .                   | 15        |
| <b>10</b> | <b>Anhang</b>                         | <b>16</b> |
| 10.1      | Glossar . . . . .                     | 16        |

# 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

### 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 kann durch eine Admin-GUI gesteuert werden
- WK1070** 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 Browseressions 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 **WK1070** 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

### 3.1 Software

### 3.2 Hardware



## 4 Funktionale Anforderungen

**PF1000 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, wenn nicht gibt das Programm eine Fehlermeldung aus und schreibt den Grund in eine Logdatei.

**PF1010 Sensordaten werden gesichert**

Die Sensordaten werden extrahiert und in mehrere Streams aufgeteilt und gruppiert und in einem Datenerhaltungssystem gespeichert. Falls es bei diesem Prozess zu Fehlern kommt, wie dass die Gruppierung fehlschlägt oder kein gültiges Datenerhaltungssystem verfügbar ist, wird auch eine Fehlermeldung ausgegeben und in eine Logdatei geschrieben.

**PF1020 Sensordaten werden verarbeitet**

Die Sensordaten werden, wenn sie verarbeitet sind (Link) werden an das Webinterfaces gesendet.

**PF1030 Sensordaten werden dargestellt**

(Reihenfolge ändern!) Nach dem die Sensordaten empfangen wurden, werden sie auf dem Webinterfaces dargestellt.

**PF1040 Mehrere Instanzen des Webinterface**

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

**PF1050 Hinzufügen weiter 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.

**PF1060 Verarbeitung der Daten für Kartenansicht**

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

**PF1070 Verarbeitung der Daten für Mittelwertberechnung**

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

**PF1080 Verarbeitung der Daten für Export**

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

## 5 Nichtfunktionale Anforderungen

### 5.1 Test

## 6 Qualitätssicherung

### 6.1 Test

## 7 Systemmodelle

### 7.1 Test

## 8 Entwicklungsumgebung

### 8.1 Entwicklungstools

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

### 8.2 Verwendete Technologien

|                           |        |
|---------------------------|--------|
| <b>Programmiersprache</b> | Java 8 |
|---------------------------|--------|

## 9 Benutzeroberfläche

### 9.1 Webinterface

### 9.2 Admin-GUI

## 10 Anhang

### 10.1 Glossar