



Master – Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Interfakultären Fachbereich für GeoInformatik (Z_GIS)
der Paris Lodron – Universität Salzburg

zum Thema

„Real Time Data Streaming im GIS“

Ansätze für den Aufbau
einer Infrastruktur

vorgelegt von

Kevin Schmidt

U106765, UNIGIS MSc Jahrgang 2021

Betreuer/in: Prof. Dr. Josef Strobl

Zur Erlangung des Grades

„Master of Science – MSc“

Gladbeck, 31.07.2023

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die Masterarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solches gekennzeichnet habe.

Gladbeck, 31.07.2023

Ort, Datum

A handwritten signature in blue ink that reads "Kevin Schür". The signature is written in a cursive style with a long horizontal stroke at the end.

Unterschrift

Danksagung

An dieser Stelle bedanke ich mich bei dem Team von Reedu GmbH, insbesondere bei Thomas Bartoscheck, Jan Wirwahn sowie Eric Thieme – Garmann für die Unterstützung beim Aufbau der konzipierten Messstation mit senseBox – Komponenten.

Des Weiteren bei meinen Arbeitskollegen bei der PLEdoc GmbH, die mir auf verschiedenen Wegen weitergeholfen und in Diskussionen neuen Input geliefert haben.

Auf diesem Wege bedanke ich mich auch bei Michell Spennhoff für die Unterstützung im praktischen Teil beim Scripting.

Mein Dank gilt ebenfalls meinem Betreuer Prof. Dr. Josef Strobl und dass er sich meiner für die Bewältigung der Master – Thesis angenommen hat.

Für den zahlreichen Input und die Motivationsbekundungen von meinen Freunden und meiner Familie, sodass ich immer am Ball blieb.

Zuletzt bedanke ich mich herzlichst bei meiner Frau, die mich auf allen erdenklichen Wegen unterstützt hat. Ohne sie wäre vieles schwerer gefallen.

Zusammenfassung

Das Erfassen von Daten verschiedener Umweltphänomene ist von großem Interesse. Sensordaten – Infrastrukturen spielen beim Monitoring von Temperaturen und anderen Umweltphänomenen im Zuge von Stadtplanungen (Smart Cities) eine bedeutende Rolle. Mit stationären Messstationen können beispielsweise die Feinstaubbelastung in Gebieten überwacht werden, in denen festgelegte Werte nicht überschritten werden dürfen.

Das Monitoring mit der Erfassung von Messwerten in kleinmaßstäblichen Projektgebieten mit einer hohen Dichte von Daten wird seltener betrachtet. Dadurch entstehen räumliche Lücken, die gerade bei der Betrachtung von lokalen und kleinräumlichen Gegebenheiten mit berücksichtigt werden müssen.

Dafür sind mobile Messstationen geeignet, die mit aktueller Technik ausgestattet sind. Solche Messstationen, wie die von senseBox, bieten nach dem Baukasten – Prinzip die Möglichkeit, nach eigenen Anforderungen die erforderlichen Komponenten zusammenzustellen und zu programmieren. Durch die Programmierung kann die Station auch mobil verwendet werden.

Um ein effektives Erfassen von Daten durchführen zu können, wird in dieser Master – Thesis eine mögliche Real Time Data Streaming – Infrastruktur konzeptioniert. Mithilfe einer smarten Messstation und einer Prozesskette, nachgelagert über diverse Software – Komponenten, soll ein Echtzeitdatenstreaming erfolgen.

Dazu wird im ersten praktischen Teil ein vergleichender Aspekt von mehreren Anbietern für Umweltmessdaten durchgeführt, um festzustellen, welche Bausteine in einer Datenbereitstellung in Echtzeit betrachtet werden müssen. Dafür werden im Vorfeld Kriterien festgelegt, die im Anschluss bewertet werden. Unter anderem wird auf die aktuellen Standards von Open Geospatial Consortium eingegangen.

Anhand der Ergebnisse aus dem ersten praktischen Teil wird eine mögliche Architektur für ein Echtzeitdatenstreaming für mobile Messstationen skizziert.

Im praktischen Teil dieser Thesis wird mithilfe des senseBox – und openSenseMap – System eine praktische Durchführung geplant und umgesetzt. Dazu wurde als Untersuchungsgebiet eine Bergbauhalde vor Ort ausgewählt.

Das Erfassen von Daten sowie die Bereitstellung und Anbindung in einem Desktop – GIS konnten nach den allgemeingültigen Aspekten von Echtzeit, dessen Definition in der Thesis diskutiert wird, erfolgreich durchgeführt werden.

Somit besteht die Möglichkeit mit einer smarten und kostenarmen Konstruktion kleinräumliche Phänomene untersuchen zu können. Entgegen dem Trend, urbane digitale Zwillinge oder ein umfangreiches Sensor – Überwachungsnetzwerk aufzubauen müssen, ist diese Variante gerade für kleinere und mittelgroße temporäre Projekte gut geeignet.

Stichwörter:

Citizen Science, senseBox, Real Time Data Streaming, Sensoren, Umweltphänomene

Inhaltsverzeichnis

Eidesstattliche Erklärung	2
Danksagung	3
Zusammenfassung	4
Inhaltsverzeichnis	6
1) Einleitung	8
1.1) Problemfeld	8
1.2) Ziele	9
1.2.1) Mögliche Anwendungsfelder	10
1.3) Ansätze	11
2) Strukturierung dieser Arbeit	13
3) Vergleich verschiedener Anbieter für Umweltmessungen mit Real Time Data Zugriff und Diskussion derer Architekturen	14
3.1) Kriterien für den Vergleich	14
3.2) Erstellen einer Vergleichstabelle	19
3.3) Durchführung Vergleich	20
3.3.1) OpenGeodata.NRW	21
3.3.2) Deutscher Wetterdienst	23
3.3.3) Bundesamt für Strahlung ODL	25
3.6) Schlussfolgerungen	27
4) Konzeptionierung einer möglichen Architektur	29
4.1) Anforderungen	29
4.2) Grafische Strukturierung	35
5) Durchführung einer Anbindung von IoT – Daten in einem Desktop – GIS an Beispiel von senseBox	36
5.1) Definition der Struktur für die Anbindung	36
5.2) Messstation vorbereiten	41
5.3) Aufbau Desktop – GIS – Struktur	43
5.4) Durchführen von Messungen	44
6) Ergebnisse	50
7) Diskussion	58
7.1) Definition Echtzeit	58
7.2) Erfassung von Messwerten	59
7.3) Software	61

7.4) Bereitstellung im Server	61
7.5) Vergleich Anforderungen	62
8) Fazit.....	64
8.1) Echtzeitdatenerfassung und Geostreaming	64
8.2) Aufwand Aufbau	64
8.3) Probleme	65
8.4) Abgleich konzeptionierte Architektur	66
8.5) Anwendungsgebiete	66
8.6) Ausschluss	67
9) Ausblick.....	69
Literaturverzeichnis	71
Anhang	76

1) Einleitung

1.1) Problemfeld

In nahezu jeder Domäne werden Sensoren eingesetzt und damit Daten erfasst. Solche Sensordaten werden in vielfältiger Weise weiterverwendet, sei es für die Überwachung oder die Auswertung von Umweltphänomenen.

Sensordaten aus Messungen für verschiedenste Umweltphänomene werden oftmals für die Weiterverarbeitung zur Verfügung gestellt. Die Art der Bereitstellung variiert von Schnittstellen bis zu Tabellen – Formaten, die zum Abrufen bereitstehen. Die Städte und Kommunen haben in Zeiten der Digitalisierung großes Interesse, eine möglichst schlanke Prozesskette von der Erfassung bis zur möglichen Weiterverarbeitung in einem GIS umzusetzen.

Klassisches Umweltmonitoring findet oftmals auf stationärer Ebene statt. Die statischen Messstationen sind mit hochpräziser, kostenintensiver Technik ausgestattet. Durch die Verteilung weniger Stationen auf einer weiträumigen Untersuchungsfläche entstehen räumliche Lücken. In Bezug auf ein Monitoring für lokale, kleinräumliche Phänomene sind diese Systeme nicht geeignet (*Borges, Pallas et al. 2018*). Beispielsweise konstruiert die Freie und Hansestadt Hamburg derzeit einen digitalen urbanen Zwilling (*Fischer, Gras et al. 2021*), der aufgrund seiner Komplexität und Größe nicht für kleinere Projekte geeignet ist.

Diverse Anbieter und Unternehmen bieten Lösungen über API – oder MQTT – Schnittstellen an, deren Schwerpunkt allerdings im Monitoring oder Übermitteln von Sensorwerten über Endgeräte liegt. Der Fokus hierbei liegt auf den Möglichkeiten einer Anbindung von Real Time Data für Umweltphänomene in einem Desktop – GIS oder anderen weiterverarbeitenden GI – Werkzeugen.

Der Fokus auf die Anbindung in einem Desktop – GIS in dieser Master – Thesis liegt darin begründet, dass insbesondere bei der Erstellung von Karten beispielsweise Web – GIS – Applikationen keinen großen Umfang bieten (*Fast and Hossain 2020*), gerade im Hinblick auf die Präsentation einer durchgeführten Erfassung und Analyse eines Projektgebietes kleineren Maßstabes. Web – GIS – Applikationen bieten eine grundlegende Ausgestaltungsmöglichkeit, reichen nach wie vor aber vom Umfang

bezüglich der Anbindungsmöglichkeiten und der Visualisierung nicht an ein vollwertiges GIS heran (*Fast and Hossain 2020*).

Es wurden bereits verschiedene Lösungen und Ansätze betrachtet, beispielsweise die Visualisierung der Echtzeitdaten in einem Dashboard (*Li, Batty et al. 2020*) oder die Erfassung von Verkehrsdaten mit Android Apps, die über einen Server bereitgestellt werden (*Fernández-Rodríguez, Álvarez-García et al. 2017*). Seltener liegt der Fokus solcher Lösungen auf die Anbindung der Daten für eine Weiterverarbeitung durch das Hauptwerkzeug eines Geoinformatikers:

Dem Desktop – Geoinformationssystem.

Die Erfassung von Umweltmessdaten hat in den letzten Jahren stark zugenommen, der Aufbau von Sensordaten – Infrastrukturen spielt eine aktuelle Rolle in der digitalen Vernetzung von Lebensräumen (*Fischer, Gras et al. 2021*). Kontrollen von Feinstaubbelastungen oder Temperaturen in urbanen Umgebungen sorgen für die Einhaltung der Lebensqualität oder der Identifikation von sogenannten urbanen Hitzeinseln (*Fekih, Bechkit et al. 2021*).

1.2) Ziele

Es wird bei den Messstationen unterschieden zwischen Bewegungsdaten (trajectory data) und Zeitdaten (temporal data) (*Li, Batty et al. 2020*):

- Bei Bewegungsdaten besitzen die Daten verschiedene Koordinaten
- Bei Zeitdaten wird in den Daten immer der gleiche Standort verwendet mit verschiedenen Zeitangaben

Um ein solches Micro – Sensoring durchführen zu können, werden Bewegungsdaten mit einer mobilen Einheit erfasst. Hierfür sind gerade kleinere Städte und Kommunen auf eine smarte, aber nicht zu komplexe Lösung angewiesen, deren Kosten sich in einem kontrollierbaren Rahmen bewegen.

Es wurden bereits verschiedene Einsatzgebiete bewertet (*Verma, Kawamoto et al. 2017*), in denen Echtzeitdatenstreaming im Internet of Things – Netzwerk zutragen kommen kann. Unter anderen auch im Transportwesen, wo große Mengen zeit – und eventabhängige Geodaten erhoben werden.

Zielgruppen, beispielsweise städtische Behörden oder Kommunen, sollen auf Live – Daten zugreifen und weiterverarbeiten können.

1.2.1) Mögliche Anwendungsfelder

Als mögliche Use – Cases für eine solche mobile Messstation mit Echtzeitdaten – Übertragung wurden folgende mögliche Szenarien zusammengetragen:

- Das Monitoring von Umweltphänomenen in definierten, kurzen Zeiträumen mit der Prämisse, an verschiedenen Stellen Werte zu erfassen und zeitreal in einem Desktop – GIS abzubilden
- Monitoring von kleinräumlichen Projektgebieten in einen definierten Zeitraum, zum Beispiel bei Bergbauhalden:
 - o Monitoring von Radonaustritt (*Al-Akel, Kunze et al. 2022*)
 - o UV – Strahlung und Temperaturen bei unterschiedlicher Besonnung einer Halde durch die Hanglagen (*Geiger 1942*)
 - o Messen von Bodentemperaturen bei sogenannten brennenden Halden (*Schüppel and Wrede 2022*)
- Monitoring von Lautstärken bei Ereignissen z.B. Konzerte oder Baustellen (*Maisonneuve, Stevens et al. 2009*), durch die Mobilität an verschiedenen Orten am zu untersuchenden Gelände
- In einen Workshop des Futuriums Berlin wurden an Fahrrädern Umweltmesser von senseBox installiert, die unter anderen Feinstaub in urbanen Gebieten messen sollen. Bisher wurden die Daten auf einer SD – Karte gespeichert und benötigen eine händische Nachbereitung für die Weiterverarbeitung (*Futurium 2021*). Hierfür wäre beispielsweise die mobile Messstation mit Echtzeitübertragung ein Fortschritt. Dafür wird ein Smartphone für die aktive Internetübertragung und Stromversorgung gebraucht. Nahezu jeder Bürger besitzt heutzutage ein Smartphone und gerade in urbanen Gebieten ist die Netzabdeckung ebenfalls gegeben
- Monitoring von Wasserqualitäten (*Usali and Ismail 2010*)
- Monitoring von Luftqualität (*Akbari, Zahmatkesh et al. 2021*) oder short – lived climate pollutants (SLCP) (*Tönisson, Voigtländer et al. 2021*)
- Brandschutzmonitoring von Wäldern (*Novkovic, Markovic et al. 2021*)

- In dieser Master – Thesis wird insbesondere durch die gegebene Lokalität (Ruhrgebiet) der Fokus im praktischen Teil auf Bergbauhalden gelegt

Daher wird in dieser Master – Thesis ein methodischer Ansatz verfolgt, der ermitteln soll, ob es mit einer möglichst straight – forward – Architektur machbar ist, in Echtzeit Umweltmessdaten in einer kleinräumlichen Projektumgebung zu streamen.

1.3) Ansätze

In dieser Master – Thesis werden verschiedene Anbieter betrachtet, um ein Verständnis für die Art der Bereitstellung und der Aktualität der Daten zu gewinnen. Der Erkenntnisgewinn fließt in die Konzeptionierung und der anschließenden Durchführung im praktischen Teil dieser Master – Thesis ein.

Der Erkenntnisgewinn aus dem vergleichenden Abschnitt soll in die Konzeptionierung einer Prozesskette fließen, die auch im praktischen Teil Anwendung finden soll. Die Darstellung des Ergebnisses erfolgt in einem Desktop – GIS.

Als mögliches kleinräumliches Untersuchungsgebiet sind Bergbauhalden zu nennen. Diese haben nachweislich mikroklimatische Phänomene, die lokal auftreten (*Al-Akel, Kunze et al. 2022*). Hierfür bieten sich Systeme wie die senseBox (*senseBox 2023*) an, Umweltdaten auf einem kleinen räumlichen Maßstab zu erfassen, die fest installierte Messstationen nicht erreichen könnten (*Borges, Pallas et al. 2018*).

Nachgewiesen wurde, dass mit den aktuell im freien Handel erhältlichen Modulen eine hohe Genauigkeit erreicht werden kann, die durchaus für solche Einsatzgebiete ausreichend sind (*Abbas, Habelalmateen et al. 2019*). Die Module können für verschiedenste Projekte genutzt werden und sind simpel in der Installation. Die mannigfaltigen Anbieter von IoT – Systemen (Internet of Things), die im Citizen Science – Umfeld zum Einsatz kommen, wie zum Beispiel senseBox, hackAir oder auch das Airrohr – Konstrukt (*Sensor.Community 2020*), nutzen die gleichen technischen Komponenten.

Solche Systeme sind nachweislich in der Lage qualitative Daten zu erheben, die weiterverarbeitet werden können und für die Bevölkerung einen Mehrwert bieten (*Banse, Schmalriede et al. 2021*).

Das openSenseMap – Portal bietet mithilfe der Low – Cost – Messstationen von senseBox einen einfachen Zugang, Daten selbst zu erheben und anzeigen zu lassen.

Anfänglich für Education – Zwecke entwickelt, kann man mit Sensoren aus dem senseBox – Baukasten verschiedene Wetter – Phänomene erfassen, wie Temperatur, Luftdruck, Luftfeuchtigkeit, Lärmpegel und andere. Über das Portal können die erfassten Daten betrachtet und heruntergeladen werden. Somit können interessierte Bürger mit wenig Mitteln und geringen Aufwand in ihrer Umgebung Phänomene messen und ein transparenteres Bild von der Lebensqualität ihrer Heimat abbilden (*Bartoschek, Wirwahn et al. 2018*).

senseBox plant mit dem Forschungsprojekt „senseBox:pro“ die Technik auch für Unternehmen und Behörden interessanter zu gestalten (*senseBox 2023*). Diese Technik bietet ein einfach zu handhabendes Werkzeug, welches durch die geringe Stromversorgung auch mobil zu betreiben ist.

Aktuell ist es mit der openSenseMap nur möglich, ausgewählte Daten als .csv – Format herunterzuladen und in einem Desktop – GIS weiterzuverarbeiten. openSenseMap bietet des Weiteren eine Rest – API – Schnittstelle an, über die der Client Informationen abfragen und modifizieren kann (*Bartoschek, Wirwahn et al. 2018*). Die alten Sensor Web Enablement – Vorgaben der OGC (zum Beispiel der SOS – Web Service) finden hier bereits keine Anwendung mehr, da diese Standards nicht mehr auf Höhe der Zeit sind (*Marsh-Hunn, Trilles et al. 2021*).

Dazu wird im praktischen Beispiel das senseBox – System als mobile Messstation konstruiert, die Speicherung der Daten auf dem Server von openSenseMap und dessen Weiterverarbeitung über FME sowie die Anbindung in QGIS durchgeführt. Die Programme werden durch den Windows Aufgabenplaner als Batch in bestimmten Zeitabständen ausgeführt.

2) Strukturierung dieser Arbeit

Die Master – Thesis gliedert sich in vier Abschnitte, die nachfolgend kurz erklärt werden:

Im ersten Abschnitt erfolgt ein Vergleich zwischen verschiedenen Anbietern für Umweltmessdaten und ob diese den festgelegten Anforderungen entsprechen. In diesem Vergleich soll festgestellt werden, ob es Anbieter gibt, die Daten zur Verfügung stellen, die möglichst in Echtzeit in einem Desktop – GIS angebunden werden können. Hieraus soll abgeleitet werden, welche Parameter für eine mögliche Architektur für Echtzeitdaten – Streaming übernommen werden können.

Im darauffolgenden Kapitel 4 wird eine Konzeptionierung einer möglichen Architektur für das Real Time Data Streaming erstellt. In diesem Abschnitt soll eine Vorgehensweise für Echtzeitdatenstreaming konstruiert werden. Hierbei gliedert sich die Struktur in vier Blöcke auf:

- Messstation
- Datenerfassung,
- Datenbereitstellung
- Bereitstellung Schnittstelle

In Kapitel 5 behandelt die Durchführung im praktischen Beispiel anhand der senseBox, openSenseMap sowie verschiedener weiterer Komponenten. Mithilfe der senseBox wird eine mobile Messstation konstruiert, welche Daten in bestimmten Zeitabständen und Koordinaten erfasst. Diese werden an den openSenseMap – Server gesendet, welche dann in weiteren Bearbeitungsschritten am PC automatisiert herangezogen werden. Als Software kommt der Windows Aufgabenplaner, Safe FME sowie QGIS zum Einsatz. Hierfür wird ein potenzielles Projektgebiet festgelegt, das für eine mobile Messstation mit Echtzeitdatenstreaming geeignet ist. Aufgrund der komplexen Gebilde und der schnellen Erreichbarkeit wird eine Halde im Ruhrgebiet ausgewählt.

Zuletzt werden die Ergebnisse aus dem praktischen Abschnitt bewertet und diskutiert. Daraus resultiert ein Fazit, aus welchem Erkenntnisse für weitere potenzielle wissenschaftliche Untersuchungen gezogen werden könnten (Ausblick).

3) Vergleich verschiedener Anbieter für Umweltmessungen mit Real Time Data Zugriff und Diskussion derer Architekturen

3.1) Kriterien für den Vergleich

Um eine Grundlage für einen Vergleich von verschiedenen Anbietern von Umweltdaten schaffen zu können, müssen gewisse Kriterien im Hinblick auf die Datenstruktur sowie Architektur der Bereitstellung festgelegt werden. Folgende Kriterien wurden festgelegt:

- a) Geografische Verortung der Daten
 - a. Nach welchem System wurden die Daten verortet
 - b. Welche Koordinatensysteme werden dafür angeboten
 - c. Was für eine Qualität besitzen die Verortungen
- b) Die Art der Datenbereitstellung
 - a. In welchen Formaten werden die Daten bereitgestellt
 - b. Was für ein Kostenfaktor gilt beim Anbieter
 - c. Erfolgt eine dynamische Bereitstellung der Daten zum Beispiel über Schnittstellen wie API (Application Programming Interface)
- c) Umweltphänomene
 - a. Welche gemessenen Phänomene werden angeboten
 - b. In welcher Genauigkeit werden diese bereitgestellt
 - c. Welche Technik wird genutzt (Sensoren)
- d) Sicherheit / Kontinuität
 - a. Wie zuverlässig kann die Datenbereitstellung erfolgen
 - b. Für welche Szenarien können diese Anwendung finden
 - c. Sind diese für kritische Infrastrukturen brauchbar
- e) Datenerhebung
 - a. Wie erfolgt die Erhebung dieser Daten
- f) Zeitliche Häufigkeit
 - a. Wie oft werden die Phänomene gemessen
 - b. In welchen Abständen werden die gemessenen Werte zur Verfügung gestellt
- g) Qualitätsstandards
 - a. Nach welchen Standards arbeitet der Anbieter
- h) Schwerpunkt des Anbieters

i) Flexibilität

- a. Können die Daten flexibel genutzt werden (zum Beispiel auch für mobile Anwendungen)

Da hier bereits sehr umfangreiche Kriterien aufgezählt werden, wird im Zuge dieser Master – Thesis der Fokus auf die Punkte gelegt, mit welchen insbesondere die Bereitstellung der Daten für weiterverarbeitende Programme, zum Beispiel in einem Desktop – GIS, bewertet werden können.

(Fischer, Gras et al. 2021) definieren für eine echtzeitfähige *Urban Data Platform* vier Anforderungen:

- 1) Eventbasierte Datenbereitstellung in Echtzeit
- 2) Kostengünstige und zuverlässige Datenbereitstellung
- 3) Gesetzeskonforme Datenbereitstellung
- 4) Sichere Datenbereitstellung

In dieser Master – Thesis wird insbesondere die erste Anforderung betrachtet:

Kriterium	Aspekte
Geografische Verortung	<ul style="list-style-type: none"> a. In welchem System wurden die Daten verortet b. Welche Koordinatensysteme bzw. welches Koordinatensystem werden dafür angeboten c. Was für eine Qualität besitzen die Verortungen
Art der Datenbereitstellung	<ul style="list-style-type: none"> a. In welchen Formaten werden die Daten bereitgestellt b. Was für ein Kostenfaktor gilt beim Anbieter c. Erfolgt eine dynamische Bereitstellung der Daten zum Beispiel über Schnittstellen (API?)
Zeitliche Häufigkeit	<ul style="list-style-type: none"> a. Wie oft werden die Phänomene gemessen b. In welchen Abständen werden die gemessenen Werte zur Verfügung gestellt

Tabelle 01: Anforderungen eventbasierte Datenbereitstellung

Es muss klar sein, wie die Daten eines Anbieters herangezogen werden können, um damit eine Weiterverarbeitung in anderen Instanzen, insbesondere in einem Desktop – GIS, zu ermöglichen. Daher sollte bei den Überlegungen, welche Kriterien erfüllt sein

müssen, ein besonderer Fokus auf die möglichen Datenformate und deren dynamischer Bereitstellung gelegt werden.

Besonders im Hinblick auf das Ziel, solche Daten möglichst zeitreal in einen Desktop – GIS anzeigen und weiterverarbeiten zu können, müssen die bereitgestellten Daten auch eine gewisse Qualität bei der koordinativen Speicherung einhalten. Ein Datenstream, in dieser Domäne ein Geostream, wird durch drei Aspekte definiert (*Liu, Li et al. 2022*):

- Es handelt sich um einen Datenstream.
- Es hat räumliche Informationen.
- Es hat zeitliche Informationen.

In dieser Arbeit wird ein Echtzeit – Geostream angestrebt. Da hier von Real Time Data Streaming gesprochen wird, spielt der zeitliche Aspekt zum einen bei der Datenerfassung, aber auch zum anderen bei der Datenbereitstellung eine besondere Rolle. Im Prinzip kann die Frage gestellt werden, können die erfassten Daten zeitnah, im Idealfall zeitgleich, im einen Desktop – GIS dargestellt werden?

Wenn in dieser Master – Thesis von Echtzeit gesprochen wird, bezieht es sich auf einen Echtzeitzugriff auf erhobene Daten. Echtzeit aus informatischer Sicht definiert sich durch die Bereitstellung von Ergebnissen in einer bestimmten Zeitspanne (*Wikipedia 2023*). Es wird ausschließlich ein Echtzeitzugriff auf einen Datenstream im Zuge dieser Master – Thesis betrachtet und nicht eine mögliche technische Latenz beim Geostream.

Die nachfolgenden Diagramme wurden mit der Android – App *miMind* erstellt:

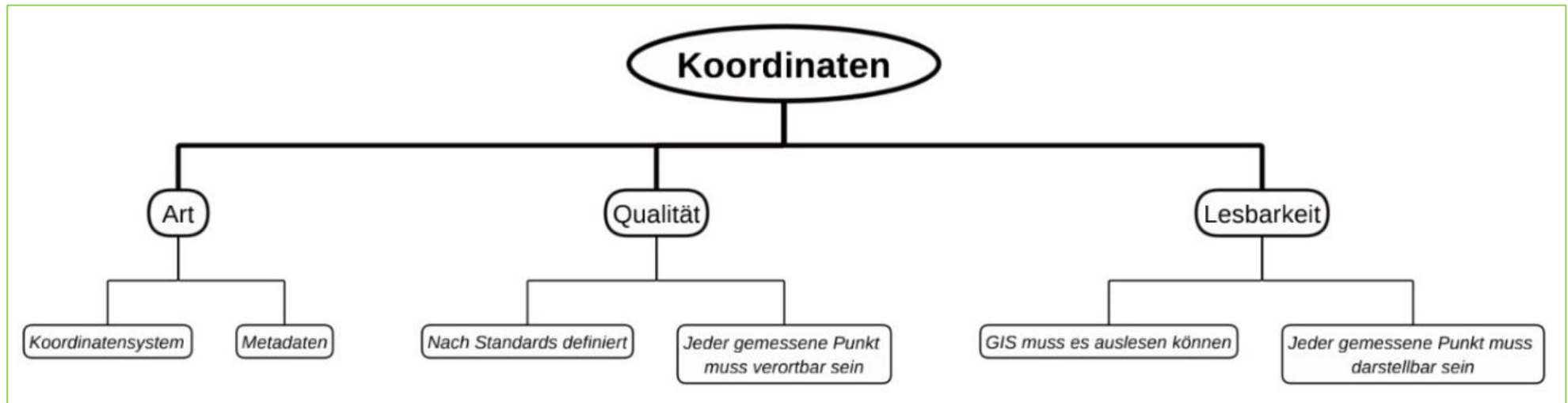


Abb. 01: Anforderungen Koordinaten

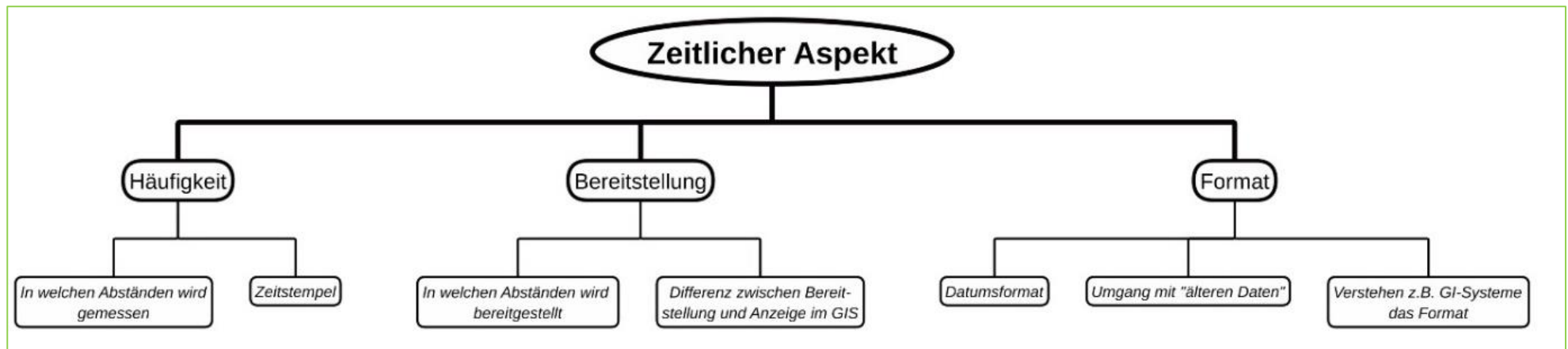


Abb. 02: Anforderungen zeitlicher Aspekt

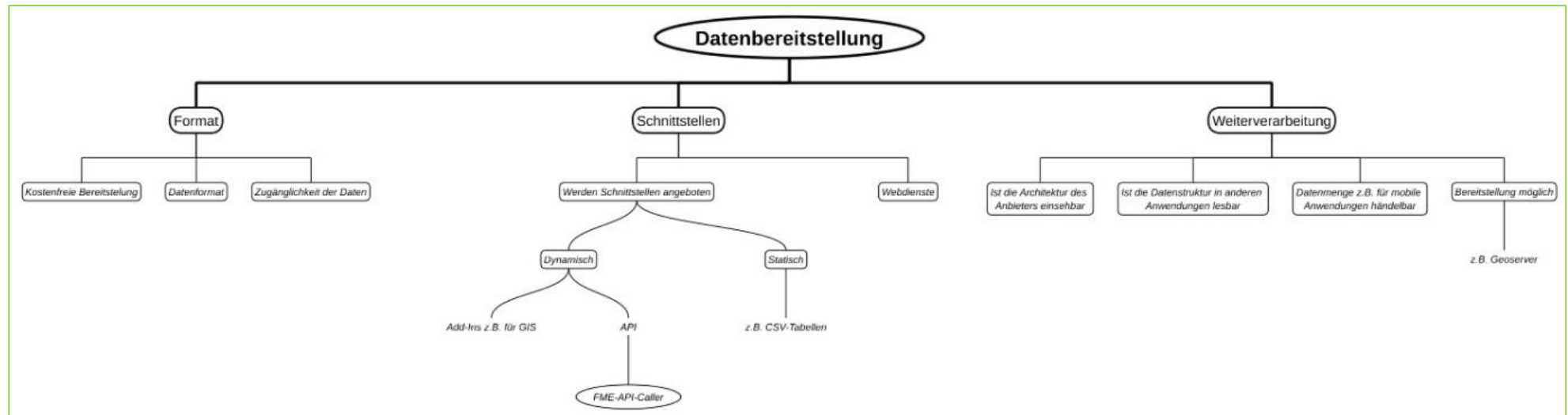


Abb. 03: Anforderungen an Datenbereitstellung

3.2) Erstellen einer Vergleichstabelle

Um die verschiedenen Anbieter vergleichen zu können, wird mit den ausgewählten Schwerpunkten eine Vorlage erstellt. Mit dieser Vorlage können die verschiedenen Punkte gefüllt und bewertet werden.

Ziel dieses Vergleiches soll es sein, einen Überblick über gängige Methoden der Anbieter zu gewinnen. Im nachfolgenden Abschnitt werden drei Anbieter genauer untersucht. Hierbei gilt zu berücksichtigen, dass es auch weitere Anbieter solcher Formate gibt. Für die Beantwortung der Frage dieser Thesis reicht es aus, eine Schnittmenge zu betrachten, um im Rahmen dieser Master – Thesis zu bleiben. Des Weiteren bieten die hier zu vergleichenden Anbieter in der Regel nur Zeitdaten (temporal data) an, das heißt die zu erwartenden Werte wurden von stationären, fest installierten Messstationen erfasst.

Unabhängig davon kann hiermit eine Grundlage geschaffen werden für die weitere Betrachtung.

Anbieter		
Datenbereitstellung	Format	Kosten
		Datenformate
		Zugänglichkeit
	Schnittstellen	Schnittstellen
		Webdienste
	Weiterverarbeitung	Architektur
		Datenstruktur maschinell lesbar
		Datenmengen Größe
Bereitstellung möglich		
Zeitlicher Aspekt	Häufigkeit	Häufigkeit Messungen
		Zeitstempel
	Bereitstellung	Häufigkeit Bereitstellung
		Zeitdifferenz
		Bereitstellung/Weiterverarbeitendes Tool
	Format	Datumsformat
		Umgang mit in der Vergangenheit liegenden Daten
GI – Systeme können das Format auslesen		
Koordinaten	Art	Koordinatensystem
		Metadaten
	Qualität	Koordinatenstandard
		Punkte verortbar
	Lesbarkeit	Desktop – GIS – tauglich
		Darstellung möglich

Tabelle 02: Vorlage Vergleich Datenanbieter

3.3) Durchführung Vergleich

In den nachfolgenden Unterkapiteln werden drei verschiedene Anbieter mit den festgelegten Kriterien abgeglichen und bewertet. Dafür wird die in Kapitel 3.2) erstellte Tabelle verwendet.

3.3.1) OpenGeodata.NRW

(<https://www.opengeodata.nrw.de/produkte/>)

Bezogen auf den Aspekt von aktuellen Umweltmessdaten wird hier das Produkt „Kontinuierliches Luftqualitätsüberwachungsmessnetz LUQS, Aktuelle Luftqualität) betrachtet.

Datenbereitstellung	Format	Kosten	Kostenfrei
		Datenformate	.csv
		Zugänglichkeit	Frei verfügbar als Open Data
	Schnittstellen	Schnittstellen	Keine
		Webdienste	Keine
	Weiterverarbeitung	Architektur	.csv – Tabelle zum Download
		Datenstruktur maschinell lesbar	Nein. Daten sind nur mit einem Kürzel des Messortes verortet. Die Zuordnung dieser Kürzel muss über eine weitere Tabelle vorgenommen werden. Das bedarf einer Aufbereitung der Daten für die Weiterverarbeitung.
		Datenmengen Größe	Weniger als 1 MB
		Bereitstellung möglich	Nur manueller Download vom Anbieter
	Zeitlicher Aspekt	Häufigkeit	Häufigkeit Messungen
Zeitstempel			Ja

	Bereitstellung	Häufigkeit Bereitstellung	Einmal täglich
		Zeitdifferenz Bereitstellung/Weiter- verarbeitendes Tool	Daten sind ab den Moment der Bereitstellung bereits „veraltet“
	Format	Datumsformat	Attributfeld für Datum (tt.mm.jjjj) und ein weiteres für Zeit (hh:mm)
		Umgang mit in der Vergangenheit liegenden Daten	Durch Bereitstellung als .csv kein Heranziehen von älteren Daten möglich. Das Portal stellt Tages – /Monatsberichte zur Verfügung.
		GI – Systeme können das Format auslesen	Ja
Koordinaten	Art	Koordinatensystem	Nur Kürzel der Messstationen enthalten
		Metadaten	Im Geoportal.Nrw enthalten
	Qualität	Koordinatenstandard	Nicht vorhanden
		Punkte verortbar	Über die Tabelle der Messstationen mit den Koordinaten – Angaben möglich
	Lesbarkeit	Desktop – GIS – tauglich	Nur über die oben genannten Umwege
		Darstellung möglich	Ja

Tabelle 03: OpenGeodata.NRW

3.3.2) Deutscher Wetterdienst

(https://www.dwd.de/DE/leistungen/opendata/faqs_opendata.html)

Datenbereitstellung	Format	Kosten	Open Data (kostenfrei)
		Datenformate	WMS, WFS, WCS, WPS, TMS, WMS – C, WMTS
		Zugänglichkeit	Bereitstellung der Services über Geoserver
	Schnittstellen	Schnittstellen	Aktuell keine API – Schnittstelle vorhanden
		Webdienste	Auf dem Geoserver können die GetCapabilities abgerufen werden
	Weiterverarbeitung	Architektur	Geoserver (Die zugrundeliegende Datenbank wird nicht offengelegt)
		Datenstruktur maschinell lesbar	Ja
		Datenmengen Größe	Nicht bekannt. Es wird für jede erfasste Temperatur ein neues Objekt erzeugt
		Bereitstellung möglich	Ja
	Zeitlicher Aspekt	Häufigkeit	Häufigkeit Messungen
Zeitstempel			Ja

	Bereitstellung	Häufigkeit Bereitstellung	Laufend
		Zeitdifferenz Bereitstellung/Weiter- verarbeitendes Tool	Daten werden zur vollen Stunde erfasst und sofort zur Verfügung gestellt
	Format	Datumsformat	xsd:dateTime (xml – Schema)
		Umgang mit in der Vergangenheit liegenden Daten	Es werden Daten aus aktuellen Zeiträumen zur Verfügung gestellt
		GI – Systeme können das Format auslesen	Ja, in QGIS getestet
	Koordinaten	Art	Koordinatensystem
Metadaten			Über getCapabilities abrufbar
Qualität		Koordinatenstandard	EPSG:4258
		Punkte verortbar	Ja
Lesbarkeit		Desktop – GIS – tauglich	Ja
		Darstellung möglich	Ja

Tabelle 04: Deutscher Wetterdienst

3.3.3) Bundesamt für Strahlung ODL

(https://odlinfo.bfs.de/ODL/DE/service/datenschnittstelle/datenschnittstelle_node.html)

Datenbereitstellung	Format	Kosten	Open Data (kostenfrei)
		Datenformate	WFS
		Zugänglichkeit	Bereitstellung der URL (Die zugrundeliegende Datenbank wird nicht offengelegt)
	Schnittstellen	Schnittstellen	Nein
		Webdienste	WFS
	Weiterverarbeitung	Architektur	Unbekannt
		Datenstruktur maschinell lesbar	Ja
		Datenmengen Größe	Nicht bekannt
		Bereitstellung möglich	Ja
	Zeitlicher Aspekt	Häufigkeit	Häufigkeit Messungen
Zeitstempel			Ja
Bereitstellung		Häufigkeit Bereitstellung	Je nach beschriebenen Aktualisierungsintervall
		Zeitdifferenz Bereitstellung/Weiterverarbeitendes Tool	Es wird eine Anfangsmesszeit und eine Endmesszeit angegeben. Der gemessene Wert (Strahlung) in dieser Zeit

			wird bereitgestellt. D.h. der Wert gilt für den gesamten Zeitraum (zum Beispiel 18 – 19 Uhr)
	Format	Datumsformat	xsd:dateTime
		Umgang mit in der Vergangenheit liegenden Daten	Es werden Daten aus aktuellen Zeiträumen zur Verfügung gestellt, z.B. bei den stündlichen Intervallen für den aktuellen Monat, für die 24 – Stunden – Messdaten eine Zeitspanne von 365 Tagen
		GI – Systeme können das Format auslesen	Ja
Koordinaten	Art	Koordinatensystem	WGS 84
		Metadaten	Über getCapabilities abrufbar
	Qualität	Koordinatenstandard	EPSG:4326
		Punkte verortbar	Ja
	Lesbarkeit	Desktop – GIS – tauglich	Ja
		Darstellung möglich	Ja

Tabelle 05: Bundesamt für Strahlung

3.6) Schlussfolgerungen

Anhand dieser Vergleiche wird ersichtlich, dass es bereits Möglichkeiten für die Anbindung von umweltbezogenen Geodaten in einem GIS gibt. Ein WFS – Dienst ermöglicht es beispielsweise aktuelle Messwerte von Messstationen darzustellen, ohne dass eine weitere Aufbereitung der Daten im Vorfeld nötig ist. Zulasten der Bedienung gehen Bereitstellungen solcher Daten als Listen (hier .csv). Durch den manuellen Download und Einladen in das Desktop – GIS kann keine Automatisierung erfolgen.

Alle drei Anbieter bieten Daten bis auf die Stunde genau an. In Hinblick auf die Ablösung der Webdienste hin zu API – Schnittstellen hat noch keiner der drei Anbieter eine vollwertige Lösung.

Bei den Daten vom Deutschen Wetterdienst wird für jede Messung ein neues Objekt erzeugt. Solches bietet sich für mobile Messstationen an, die in einem großmaßstäbigen Gebiet eingesetzt werden (zum Beispiel einem Park oder Straßenzug). Bei den Daten vom Deutschen Wetterdienst handelt es sich um temporale Daten, die von fest installierten Messstationen erfasst werden. Das heißt, für jeden Messwert wird mit der immer gleichen Koordinate ein neues Objekt erzeugt. Beim Recherchieren und Abgleichen ist aufgefallen, dass es keine implementierte Funktion in QGIS gibt, die automatisch in bestimmten Intervallen die Layerverbindungen aktualisiert. Abhilfe schaffen hier Automatisierungen über Python – Skripte.

Es gibt durchaus Möglichkeiten eines Datastreaming in einem Desktop – GIS, allerdings ist die Handhabe eingeschränkt. Zum einen muss der WFS – Anbieter in der Lage sein, die Daten zeitnah bereitzustellen, sodass gewährleistet ist, dass die Abbildung im Desktop – GIS auch möglichst aktuell ist. Zum anderen muss auf der technischen Seite das Desktop – GIS auch selbstständig in kurzen Intervallen die Verbindung beziehungsweise den dargestellten Layer aktualisieren können.

Für die Konzeptionierung einer Architektur für ein Geodatenstream kristallisieren sich also anhand dieser Recherche folgende Bausteine heraus:

- Zeitnahe Aktualisierung
- Bereitstellung Daten als OGC – konformer Dienst (WFS oder API – Schnittstelle)

- Automatische Aktualisierung der Verbindung sowie Darstellung im Desktop – GIS
- Jede Messung ergibt ein Punkt – Objekt (Bewegungsdaten)

Diese Bausteine finden bei der Konzeptionierung einer möglichen Architektur Anwendung, worauf im nächsten Kapitel eingegangen wird.

4) Konzeptionierung einer möglichen Architektur

4.1) Anforderungen

Bei der Datenerfassung wird eine Messstation benötigt. Diese Messstation muss in der Lage sein, Umweltphänomene zu erfassen und mit einer Standortangabe zu speichern.

In diesen Fall soll die Messstation mobil sein, es wird also eine tragbare Lösung gebraucht:

- 1) Autarke Energieversorgung mittels Smartphone
- 2) Das Smartphone selbst dient als Hotspot für die Datenübertragung
- 3) Der Sensor selbst erfasst die Daten und überträgt diese über ein WLAN – Modul an das Smartphone
- 4) Das Smartphone überträgt diese Daten an den Server, wo es weiterverarbeitet wird. Dieses kabellose Vorgehen entspricht den aktuellen Standards (*Sejdiu, Ismaili et al. 2021*).

Die Standortangabe kann zum Beispiel über Vorgabe – Werte bei der Erzeugung von neuen Objekten attributiv erfasst werden. Dieses Vorgehen ist aber dann nur sinnig, wenn eine manuelle Nachbereitung der erfassten Werte vorgesehen ist oder die Messstation stationär ist.

Ist eine mobile Messstation vorgesehen, sollten die Koordinaten der Standorte automatisch über ein GPS – Modul erfasst werden. Bei der Weiterverarbeitung der erfassten Daten sollte die Qualität der Koordinaten, sprich die Genauigkeit, bekannt sein. Der Anwender (Privatperson, Bürger) der mobilen Messstation kann bei entsprechender Schulung hochqualitative Daten erzeugen, (*Kosmala, Wiggins et al. 2016*) die in die Metadaten der Objekte geschrieben werden. Moderne GPS – Module, die mit wenig Energie betrieben werden können (zum Beispiel mit einem üblichen 5 Volt USB – Anschluss) verfügen über eine ausreichende Genauigkeit, wie zum Beispiel das NEO – 7M GPS Modul (*u-blox 2023*) (*Ramesh, Sivaraman et al. 2021*). Bei der Erfassung von Messwerten muss ein Zeitstempel erzeugt werden. Dieser Timestamp wird als Attribut am Objekt festgehalten. Der Timestamp sollte einer Norm entsprechen, üblicherweise der ISO 8601 (YYYY.MM.DD hh:mm:ss + Offset). Ebenso wird der erfasste Messwert vom verbauten Sensor am entsprechenden Attribut eingetragen.

Bei der Programmierung der Messstation muss festgelegt werden, in welchen zeitlichen Abständen neue Werte erfasst werden sollen. Dieser zeitliche Abstand hängt vom Szenario ab. Bei einer mobilen Station mit koordinativer Erfassung über ein GPS sollten diese Abstände nicht zu groß sein, damit innerhalb eines Untersuchungsgebietes ausreichend Daten erfasst werden können. Bei der Datenbereitstellung dieser erfassten Werte müssen diese aus einer Tabelle in ein geometrisches Objekt (Punkt – Objekt) verarbeitet werden. Idealerweise werden diese in einer Geodatenbank gespeichert. In den Metadaten sollte ein Koordinatenbezugssystem angegeben sein (zum Beispiel über einen EPSG – Code). In der OGC SensorThings – API werden geographische Koordinaten in WGS84 erwartet (*Kotsev, Schleidt et al. 2018*).

Die GO FAIR Initiative beschreibt die FAIR Prinzipien, die die Auffindbarkeit (findable), Zugänglichkeit (accessible), Interoperabilität (interoperable) sowie die Wiederverwendbarkeit (reusable) für digitale Assets festlegen. Damit die Daten auffindbar sind, ist es wichtig, dass die Daten und Metadaten maschinenlesbar sind. Das gelingt nur, wenn die (Meta –) Daten eine global eindeutige und dauerhafte Kennung erhalten (F1), die Daten mit Metadaten beschrieben werden (F2), die Metadaten eindeutig und explizit die Kennungen der zugehörigen Daten erhalten (F3) und die (Meta –) Daten in einer durchsuchbaren Ressource nachgehalten werden (F4) (*Wilkinson, Dumontier et al. 2016*).

Damit der Geodatenstream gefunden werden kann, wird ein Metadaten – Katalog benötigt. Beispielsweise kann in Geoserver über ein Add – On der bereitgestellte Layer umfangreich mit Metadaten beschrieben werden. Diese Metadaten können über einen CSW – Dienst (Catalog Services for the Web) nach INSPIRE – Standards bereitgestellt werden. Somit ist gewährleistet, dass andere User über die Metadaten alle benötigten Informationen bekommen und diese auch finden können.

Sind die Daten in einer Geodatenbank gespeichert, kann der eingerichtete Server (beispielsweise Geoserver) darauf zugreifen und somit die Daten im Internet oder Netzwerk bereitstellen. Hierbei muss bei der Konfigurierung ein automatisiertes Bereitstellungsintervall festgelegt werden, damit die Daten möglichst aktuell beim Endanwender angezeigt werden.

Die Datenformate der Attribute werden nach dem OGC – XSD – Schema definiert, um einen gewissen Standard für die Bereitstellung einzuhalten. Mindestens benötigt werden:

- ID
- Name
- zu messendes Umweltphänomen
- Datum

Der eingerichtete Server stellt eine Schnittstelle bereit. Das OGC stellt einen neuen Standard für Schnittstellen mit IoT – Netzwerken bereit (*Consortium 2023*). Die SensorThings API ist der aktuellste und vollständigste Standard für Daten aus IoT – Netzwerken (*Huang and Chen 2019*). Der OGC Sensor Observation Service, kurz SOS, gilt als veraltet und wird bereits vielerorts nicht mehr angewendet (*Marsh-Hunn, Trilles et al. 2021*).

Ab 2015 wurde vom Open Geospatial Consortium ein neuer Standard für die Bereitstellung in Form einer Schnittstelle für Sensordaten vorgestellt, das SensorThings – API. Bisher erfolgen Bereitstellungen von Sensordaten wie beispielsweise Temperaturen oder Strömungsgeschwindigkeiten im Meer (Aeonometer) als SOS – Dienst. Beim Aeonometer handelt es sich im konkreten Beispiel um eine Boje in der Verantwortlichkeit des NDBC (National Data Buoy Center) USA im Meer, die über NOAA (National Oceanic And Atmospheric Administration) als SOS – Dienst bereitgestellt wurde. Siehe auch:

<https://sdf.ndbc.noaa.gov/sos/server.php>

Stationäre Messstationen haben somit die Möglichkeit, Messdaten für verschiedene Umweltphänomene über einen Web – Service bereitzustellen und End – Anwender können beispielsweise über einen GET – Aufruf erfasste Daten aus gewünschten Zeiträumen aufrufen oder in einen GIS anbinden. Das Sensor Observation Service gehört zur SWE – Initiative von OGC. SWE steht für *Sensor Web Enablement* und dient zur Standardisierung von Sensoren sowie Sensornetzwerken. Diese SOS – Service werden heute kaum noch genutzt oder die Dienste werden nicht mehr aktualisiert. In Zuge von wissenschaftlichen Arbeiten wurde bereits nachgewiesen, dass die Nutzung von SOS – Services nicht mehr zeitgemäß ist und

das SensorThings – API eine bessere Interoperabilität bietet (*Marsh-Hunn, Trilles et al. 2021*). Das 2015 eingeführte SensorThings – API löst somit den SOS – Service ab mit dem Fokus auf Interoperabilität für das IoT – Netzwerk. Beim SensorThings – API handelt es sich um eine REST – basierte Schnittstelle, in dem die Daten in JSON codiert werden und die URLs auf den OASIS – OData – URL – Konventionen basieren (*Consortium 2023*).

Dem Web Feature Service (WFS) wird von der OGC das neue API – Features zur Seite gestellt. Das API – Features wurde 2019 eingeführt und wird das WFS ablösen (*Consortium 2023*).

OGC API – Features werden in HTML und GeoJSON enkodiert, was den aktuellen Ansprüchen an Schnittstellen entspricht. Der OGC Web Feature Service wird in GML enkodiert. Über eine API – Features – Schnittstelle können vektorielle Geodaten bereitgestellt und weiterverarbeitet werden (*Consortium 2023*). Grundsätzlich sind API – Schnittstellen der aktuelle Standard für Datenbereitstellungen und konnten sich bereits in verschiedenen Fachbereichen etablieren. Dazu kommt eine breitere Akzeptanz im Mainstream, da API – Schnittstellen bekannter und weiter verbreitet sind als die speziellen Web – Services (*Jirka, Autermann et al. 2021*). Die Betrachtung der beiden Schnittstellen SensorThings – API sowie API – Features in dieser Master – Thesis liefern folgende Grundlage:

Welche Schnittstelle ist besser für eine mobile Datenerfassung mit Bewegungsdaten geeignet? Während das SensorThings – API grundsätzlich für Messstationen für die Erfassung von Umweltmessdaten konzipiert ist, wird es durch die Voraussetzung eines fixen Standorts (temporal data) definiert. Das lässt sich zum Teil am Entity – Relationship – Diagramm erkennen, dass für die Location ein Standort angegeben werden muss (*Consortium 2023*).

Insofern bietet das API – Features den entscheidenden Vorteil mit Daten, die an verschiedenen Standorten erfasst werden (trajectory data), umgehen zu können. Die Daten werden als Geometrien mit Koordinaten gespeichert (*Consortium 2023*). Sollte also im praktischen Teil dieser Master – Thesis eine erfolgreiche Erfassung von Daten mit unterschiedlichen Koordinaten und ein Echtzeitdatenstreaming erfolgen, böte sich eine genauere Betrachtung dieser Prozesskette mit Einbindung einer von OGC definierten Schnittstelle an.

Ziel dieser ganzen Betrachtungen und Bereitstellungen von Standards vom Open Geospatial Consortium ist eine Möglichkeit, weltweit einheitliche Datenaustausch – Formate nutzen zu können, beispielsweise auch für Marine – Sensoren – Architekturen (*Partescano, Brosich et al. 2017*).

Für die Durchführung des praktischen Teils wird die Bereitstellung der Daten im ersten Aufbau als OGC KML – Format erfolgen. Das Open Geospatial Consortium KML ist eine XML – Kodierung für Geodaten. KML – Dateien haben sich als beliebtes Austauschformat bewährt und können in einer Vielzahl von Geodaten – Programmen geöffnet werden, beispielsweise Google Earth oder auch GI – Systemen. Eine KML bietet nicht denselben Umfang wie eine Speicherung von Daten in eine Datenbank, ist also eher weniger für sehr umfangreiche Datensammlungen geeignet (*Burggraf 2015*). Daher bietet sich dieses Dateiformat für den praktischen Teil aufgrund ihrer einfachen Handhabung an und mit dem Fokus auf mittelgroße temporäre Projektgebiete wird in der Regel keine Datenbank benötigt.

Aktuell kann in einem Desktop – GIS diese Schnittstelle nicht ohne weiteres angebunden werden. Im Server muss die Häufigkeit der Bereitstellung konfiguriert werden. Das muss im Zusammenspiel mit der Bereitstellung aus der Datenbank bzw. Zeitintervalle der Messungen/Datenerfassung funktionieren. Sind die Zeitabstände zu groß, findet kein Real Time Data Streaming mehr statt (Auszug *techopedia: The basic definition of real – time data is that it is data that is not kept or stored, but is passed along to the end user as quickly as it is gathered [...] (techopedia 2023)*). Bei zu kurzen Zeitabständen besteht die Gefahr, dass häufige Aktualisierungsprozesse angestoßen werden müssen, das führt zu Performance – Einbußen.

Eine mögliche Archivierung der Altdaten erfolgt über die Datenbank. Bedingt zum Einsatzfeld kann eine Archivierung der Daten zum Beispiel nach einem Tag vorgenommen werden. Damit wird verhindert, dass eine zu große Datenmenge im Server bereitgestellt und im Desktop – GIS herangezogen wird. In den möglichen Einsatzfeldern werden lokale Gegebenheiten in einem kleinen Zeitraum untersucht, daher wird hier eine langfristige Archivierung oder der Aufbau einer solchen Architektur nicht betrachtet.

Im letzten Teil der Prozesskette, der Anbindung im Desktop – GIS muss die definierte OGC – konforme Schnittstelle angebunden werden. Im Desktop – GIS (zum Beispiel QGIS) können die Daten weiterverarbeitet werden für tieferegehende Analysen,

Verschneidungen und Auswertungen. Hierbei müssen zwei Bausteine eine automatische Aktualisierung durchführen können:

- Die Verbindung mit der API – Schnittstelle, damit die neuesten Daten herangezogen werden.
- Der dargestellte Layer muss aktualisiert werden.

Optional können über ein Datenmanagement – Werkzeug die Daten über die API – Schnittstelle herangezogen werden und für weitere Zwecke entsprechend transformiert werden (zum Beispiel mit Safe FME oder HALE Studio).

4.2) Grafische Strukturierung

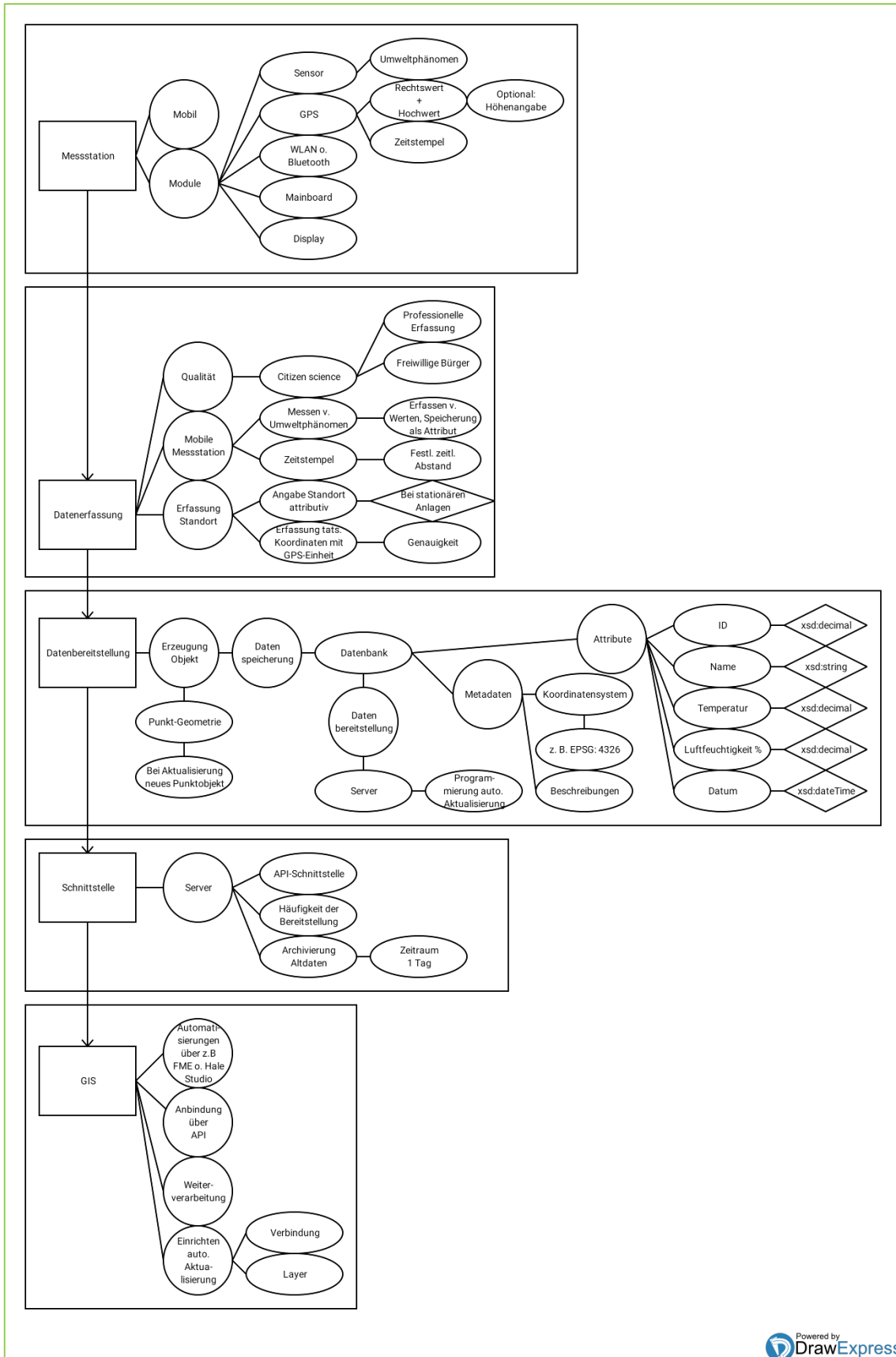


Abb. 04: Strukturelle Abfolge einer mobilen Messstation und Anbindung in Desktop – GIS

5) Durchführung einer Anbindung von IoT – Daten in einem Desktop – GIS an Beispiel von senseBox

5.1) Definition der Struktur für die Anbindung

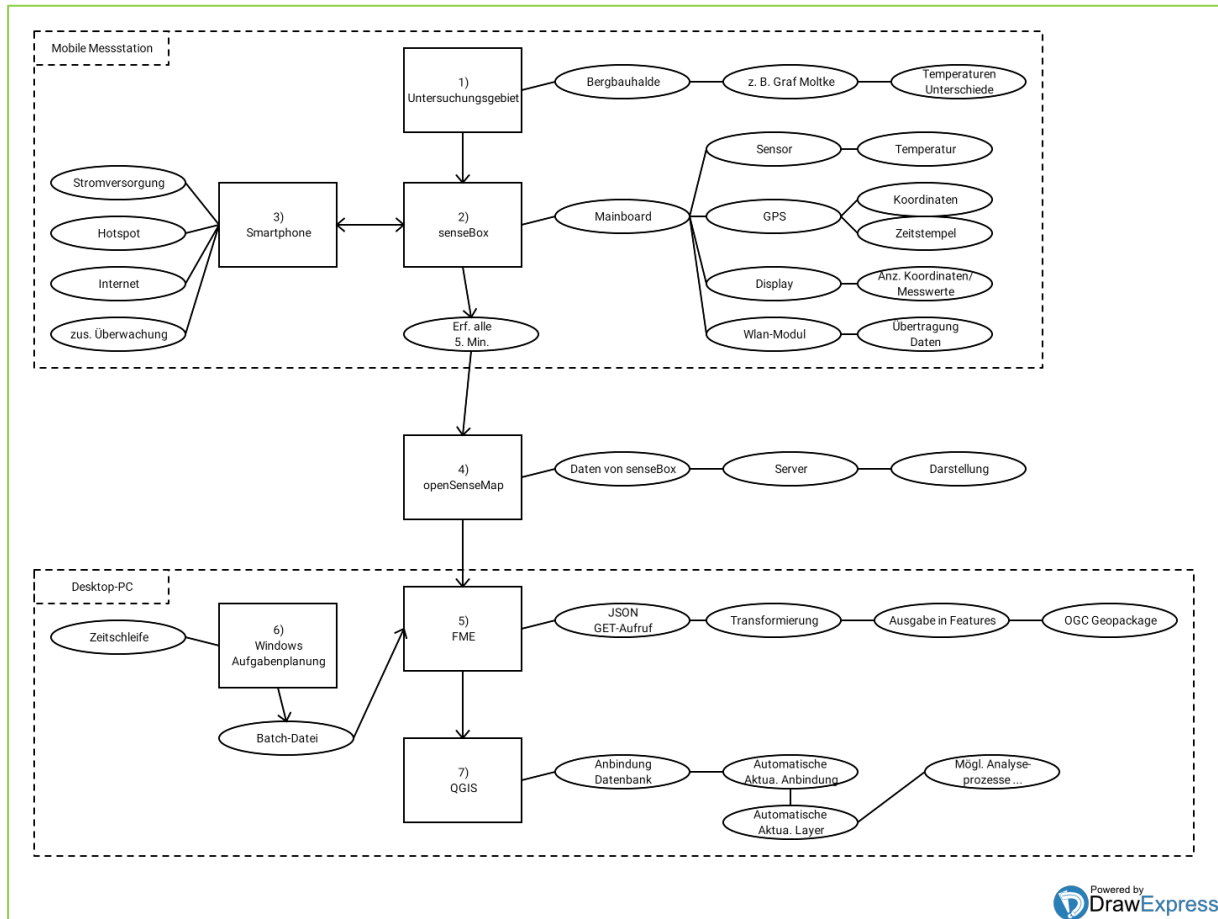


Abb. 05: Abfolge der mobilen Messstation mit dem senseBox – System

In dieser Beispielkonfiguration soll mit dem modularen System von senseBox eine mobile Messstation konfiguriert werden. Im Diagramm (Abb. 05, Seite 36) werden die einzelnen benötigten Bausteine aufgezeigt, die nachfolgend im Detail erörtert werden:

Die senseBox (Baustein Nr. 2) selbst besteht aus einem Mainboard, an das folgende Hardware angeschlossen wird: der Sensor für das Umweltphänomen, in diesem Falle für Temperatur und Luftfeuchtigkeit. Es handelt sich hierbei um den HDC1080 von Reedu. Für die Verortung der gemessenen Punkte wird ein GPS – Modul verwendet (CAM – M8Q), welches Längengrad, Breitengrad und Höhe erfasst. Dieses Modul kann keine Korrekturdaten empfangen, daher sind die Koordinaten mit der Genauigkeit von handelsüblichen Navigationsgeräten (oder zum Beispiel auch von Smartphones) zu bewerten. Die Genauigkeit liegt über dem optimalen Wert von kleiner fünf Metern.

Um im Einsatz einen Überblick über die Funktionsfähigkeit oder auch die aktuellen Messwerte zu bekommen, wird ein Display benötigt. Die Darstellungen auf dem Display (GUI) können über die Programmierung in „Blockly“ frei gestaltet werden.

Damit die Messwerte auch in den openSenseMap – Server geladen werden können, wird ein Konnektivitäts – Modul benötigt. senseBox bietet hier verschiedene Möglichkeiten, zum Beispiel mit Bluetooth oder einen SD – Kartenschreiber. Für in dieser Master – Thesis angestrebten Konfiguration wird das WLAN – Modul verwendet, das in der Programmierung so definiert wird, dass das Smartphone als Hotspot verwendet werden kann. Das Smartphone (Baustein Nr. 3) dient über den USB – Anschluss als Energie – Quelle für die senseBox (5 Volt). Es wird ein Hotspot eingerichtet, mit dem die Box die aufgezeichneten Messwerte per WLAN überträgt. Dafür wird eine aktive und stabile mobile Internetverbindung benötigt. Voraussetzung ist hierbei also ein Projektgebiet, in dem sich eine flächendeckende Netz – Abdeckung befindet.

Die erfassten Daten werden an die openSenseMap (Baustein Nr. 4) bzw. deren Server übertragen. Gleichzeitig erfolgt eine Darstellung auf der openSenseMap der Daten mit Standort, erfassten Messwerten vom konfigurierten Umweltphänomen (hier Temperatur und Luftfeuchtigkeit) und als Diagramm.

Am Desktop – PC selbst wird über FME von Safe Software eine Workbench (Baustein Nr. 5) verwendet, die über diverse Transformer die Daten aus der OSeM heranzieht, umwandelt und in einer Datenbank oder Datei bereitstellt. In der Workbench werden über einen Feature Reader mit einem REST – GET – Aufruf die letzten 10.000 Messungen im JSON – Format abgerufen. Wird über den Aufruf kein Zeitraum definiert, werden die letzten 48 Stunden herangezogen. Der Aufruf gestaltet sich wie folgt:

<https://api.opensensemap.org/boxes/632378f5d0e7a8001b19f943/data/632378f5d0e7a8001b19f945>
[Zugriff auf die API-Schnittstelle] [ID der senseBox] [ID vom (Temperatur-)Sensor]

Der Link setzt sich aus mehreren Parametern zusammen:

- <https://api.opensensemap.org/> ist der Aufruf an die API – Schnittstelle von OSeM. Dieser Parameter bleibt immer gleich

- `boxes/632378f5d0e7a8001b19f943/` adressiert die Messstationen auf dem Server („boxes“). Die nachfolgende ID spricht die gewünschte Messstation selbst an, in diesen Fall die in dieser Thesis verwendete, namentlich „Kiwisstation_22“.
- `data/632378f5d0e7a8001b19f945` ist die ID des anzusprechenden Sensors („data“). Eine Messstation kann auch mehrere Sensoren enthalten, die jeweils eine eindeutige ID haben, mit der die angesprochen werden können.

Sämtliche IDs von der Box als auch die von den enthaltenden Sensoren können auf der Seite von openSenseMap bei den eigenen Boxen nachgeschlagen werden. Die IDs stehen allen Nutzern öffentlich zur Verfügung, so dass andere Anwender ebenfalls die installierten Boxen ansprechen können.

Aus den Attributen werden die Koordinaten herangezogen im Transformer AttributeManager (Abb. 07, Seite 39) und mit dem VertexCreator – Transformer Punktgeometrien erzeugt. Im VertexCreator werden die vorher attributiv erfassten Werte als Koordinatenwerte für die Punktgeometrien verwendet.

Damit im GIS ein Koordinatensystem erkannt wird, muss ein solches definiert werden („CoordinatesystemSetter“), hier im EPSG – Code 4326. Dieses Koordinatensystem besitzt eine Genauigkeit bis maximal zwei Meter. Zusätzlich wird das definierte Koordinatensystem zur Info als weiteres Attribut an die Objekte vererbt („CoordinateSystemExtractor“). Im nächsten Bookmark („Datetime formatieren“) wird das Attribut für die Zeit definiert.

Im OSeM wird die Zeit wie im folgenden Beispiel abgespeichert:

2023 – 02 – 16T16:00:55.000Z (ISO 8601)

Im Skript wird dieses Datumsformat so umgewandelt, dass das geläufige datetime – Format ohne das Time – offset gespeichert wird:

2023:02:16T17:00:55

Dazu wird in den beiden Transformern „SubstringExtractor“ jeweils einmal aus dem OSeM – Datetime das Datum sowie die Zeit in eigene Attribute zwischengespeichert. Im anschließenden „AttributeCreator“ werden die beiden Attribute so zusammengesetzt, dass im anschließenden Transformer „DateTimeCalculator“ das benötigte Offset dazugerechnet werden kann. Im letzten Schritt, im Transformer „DateTimeConverter“, wird das nun lesbare Attribut in das geläufige Datetime – Format

(siehe oben) konvertiert. In weiteren Transformern werden die Attribute angepasst und unbenannt. Zuletzt werden die erzeugten Features in ein OGC KML lokal gespeichert.

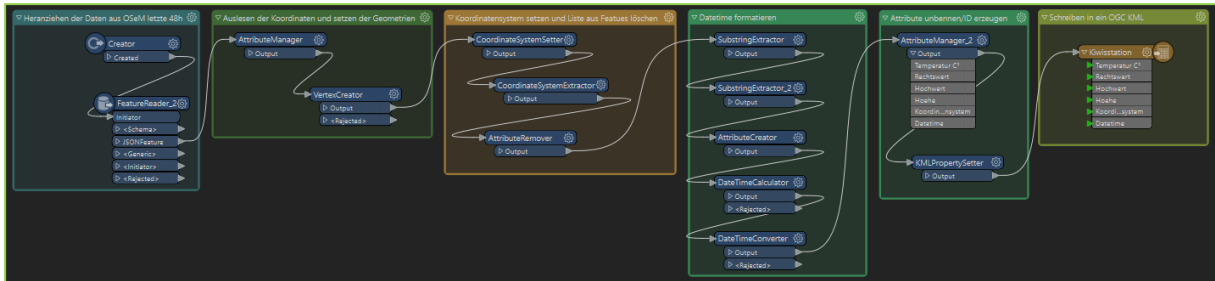


Abb. 06: FME Workbench für die Umwandlung der Daten in ein QGIS – taugliches Format (Hinweis: Eine größere Abbildung befindet sich im Anhang 03)

Input Attribute	Output Attribute	Attribute Value	Action
value	Temperatur C°	<Enter new value (optional)>	Rename
location{}	location{}	<Enter value (optional)>	Do Nothing
createdAt	ISO_Time	<Enter new value (optional)>	Rename
location(0)	Rechtswert	<Enter new value (optional)>	Rename
location(1)	Hochwert	<Enter new value (optional)>	Rename
location(2)	Hoehe	<Enter new value (optional)>	Rename

Abb. 07: Heranziehen der erforderlichen Attribute im Transformer

Dieser Schritt der Umwandlung in einer FME – Workbench wird benötigt, da zum Zeitpunkt dieser Master – Arbeit QGIS keine Möglichkeit einer Anbindung von Daten aus dem OSeM – Server zuließ. Stand Frühjahr 2023 ist es in QGIS möglich, nur OGC API – Features anzubinden. Es ist möglich, in QGIS über ein https – Protokoll eine URL mit GeoJSON – Inhalten anzubinden. Allerdings nur, wenn diese direkt bereitgestellt wird und nicht über eine API – Schnittstelle.

Damit diese Workbench automatisch angestoßen wird, wird über die Windows Aufgabenplanung eine Aufgabe (6) definiert, die in bestimmten Abständen einen Batch startet. Die Batch – Datei besteht aus einer .cmd – Datei, in der mit dem installierten Programm eine bestimmte Workbench, in diesem Falle für die Bereitstellung der Daten aus OSeM als OGC KML – File, ausgeführt wird (Abb. 08, Seite 39).

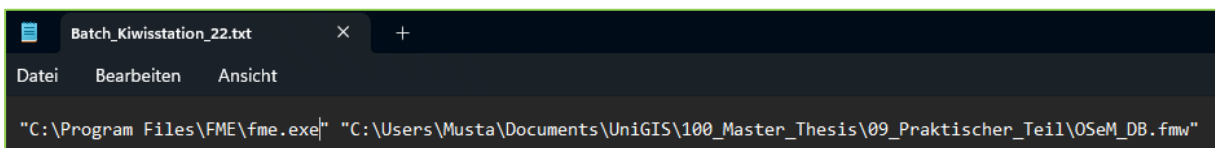


Abb. 08: Erstellung Batch – .cmd in einer .txt – Datei

In dem bei Windows standardmäßig installierten Programm „Aufgabenplanung“ können Aufgaben automatisiert werden. Typischerweise gehören hierzu wiederholende Aufgaben, die über bestimmte Trigger angestoßen und in festgelegten Abständen wiederholt werden.

In der Aufgabenplanung wird eine Aufgabe festgelegt, dass alle fünf Minuten über die Batch – Datei die Workbench angestoßen wird (Abb. 09, Seite 40). Der Trigger wird auf den Projekttag festgelegt, so dass die Aufgabe zu einer bestimmten Zeit automatisch startet.

Name	Status	Trigger	Nächste Laufzeit	Letzte Laufzeit	Ergebnis der letzten Ausführung	Autor	Erstellt
Create_GPK...	Deaktiviert	Um 12:40 Uhr am 05.03.2023 - Nach Auslösung alle 5 Minuten für die Dauer von 1 Tag wiederholen.		05.03.2023 17:55:01	(0x1)	SCHMIDDENS\Musta	28.02.2023 20:23:53

Allgemein	Trigger	Aktionen	Bedingungen	Einstellungen	Verlauf
Name:	Create_GPKG_Kiwistation				
Speicherort:	\\Kivis Aufgaben				
Autor:	SCHMIDDENS\Musta				
Beschreibung:	Aufgabe für die Erzeugung der Geopackage aus der Workbench heraus alle 5 Minuten				

Abb. 09: Aufgabe für die Batch – Datei

Sobald neue Daten erfasst werden, erzeugt die Workbench alle fünf Minuten automatisch ein neues OGC KML – File. Dazu muss der Computer, auf dem das Skript sowie das Desktop – GIS ausgeführt werden, in der gesamten Zeit eingeschaltet sein.

Im letzten Abschnitt erfolgt die Anbindung der transformierten Daten in einem Desktop – GIS, hier QGIS (7). Der Layer muss automatisch aktualisiert werden ohne dass ein manuelles Eingreifen nötig wäre.

Ist beispielsweise in einem Projekt die Messungen und Datenerfassung abgeschlossen, kann die erzeugte Datei für die Bereitstellung im GIS archiviert werden. In der Workbench kann der Writer so definiert werden, dass bestehende Dateien überschrieben werden. Somit bietet sich eine Archivierung vergangener Bereitstellungen an. Dieser Schritt kann auch automatisiert werden. Die Anbindung im QGIS und dessen Verhalten wird in Kapitel 5.3 genauer erörtert. Das KML – Format ist ein weit verbreitetes (Austausch –) Format, welches mit nahezu jedem GI-System geöffnet werden kann. Mit beispielsweise Google Earth bietet es sich auch für Casual User an, die nicht häufig mit GI – Systemen arbeiten.

Die Dateistruktur lässt es zudem zu, dass die Datei im einen GIS geöffnet sein kann und zeitgleich die Datei von einem anderen Programm überschrieben oder gelöscht/neu erzeugt wird.

5.2) Messstation vorbereiten

Über die grafische Programmieroberfläche „Blockly“ von senseBox können ohne Programmierkenntnisse die erforderlichen Befehle für die eigene senseBox geschrieben werden (Abb. 10, Seite 41). In dem für diese Beispielanwendung geschriebenen Code (Arduino) wird der Hotspot vom Smartphone notiert sowie die Anbindung an den openSenseMap – Server konfiguriert. Im Code werden das GPS – Modul, das Display sowie der Sensor angesteuert.

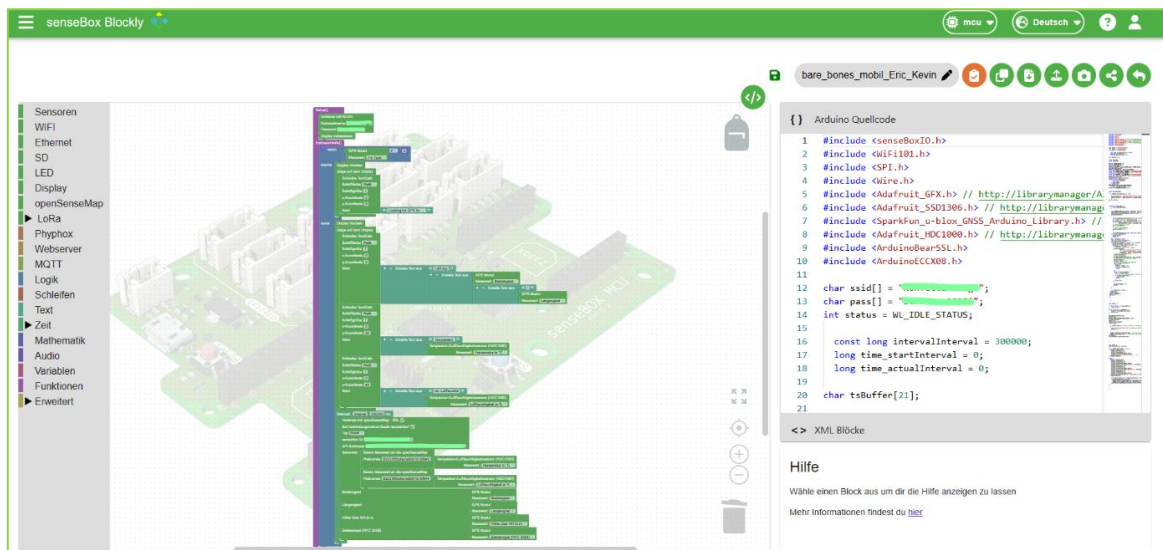


Abb. 10: Grafische Programmieroberfläche „Blockly“

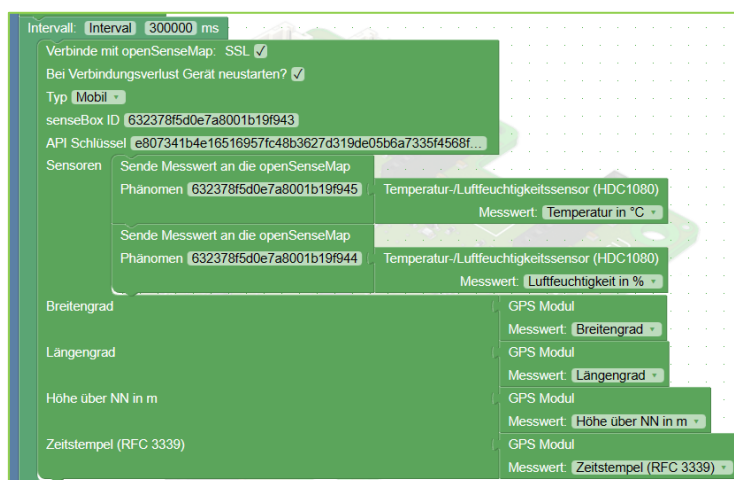


Abb. 11: Vergrößerter Ausschnitt aus Blockly

Die mobile senseBox – Messstation (Abb. 12, Seite 42) besteht aus folgenden Komponenten:

Dem eigenen Smartphone für die Datenübertragung und Stromübertragung. Das Smartphone ist über ein USB – Kabel mit der senseBox verbunden und ist aktiv mit dem mobilen Internet gekoppelt.

Den Sensor für die Temperatur und Luftfeuchtigkeit. Um vor den äußeren Witterungen geschützt zu werden, ist der Sensor in einen Strahlenschutz verbaut, der aufgeschraubt wird.

Im Outdoor – Gehäuse verbergen sich die restlichen Komponenten: Das Mainboard, an dem der Sensor, das GPS – Modul, das Display sowie das WLAN – Modul verbunden sind. Über Abstandshalter sind alle Komponenten einzeln zugänglich und durch die durchsichtige Abdeckung ist das Display ablesbar.

Der Strahlenschutz und das Outdoor – Gehäuse werden auf einer Sperrholzplatte mit Gurt befestigt, welches dann bequem getragen werden kann (Abb. 13, Seite 43).



Abb. 12: Die senseBox mit angeschlossenem Smartphone

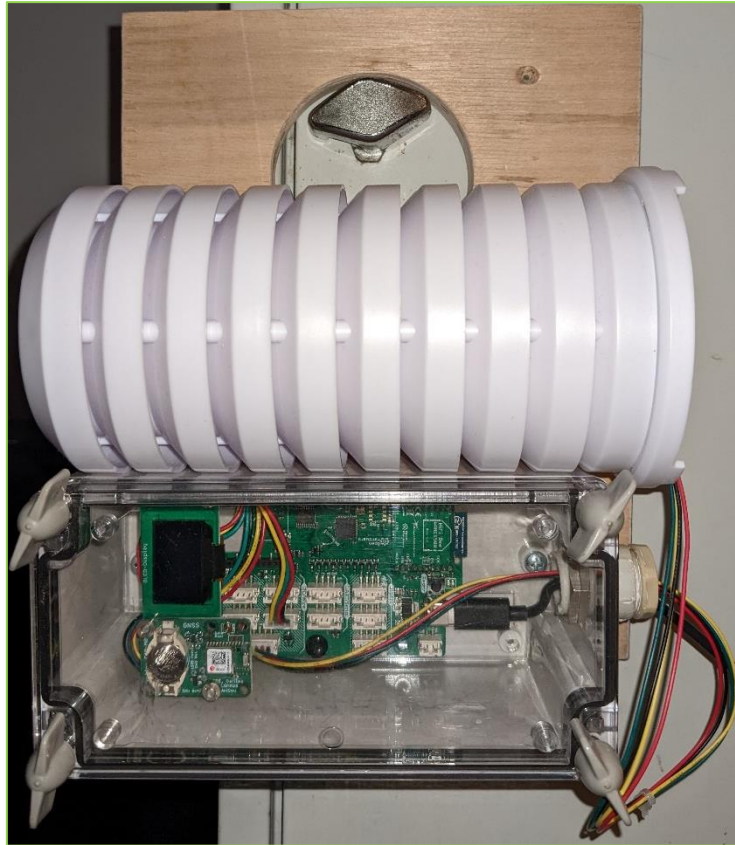


Abb. 13: Einsatzbereite mobile Messstation

5.3) Aufbau Desktop – GIS – Struktur

In QGIS sollen die gemessenen Standorte direkt angezeigt werden. Es wird ein QGIS – Workspace angelegt. Im diesen Workspace wird über die Datenquellenverwaltung das OGC KML – File herangezogen und der Layer mit den Messpunkten wird als Eintrag in der Legende dargestellt. Hierbei muss der Layer die gleiche Benennung haben wie im benötigten Python – Skript.

Dieses Python – Skript wird für die regelmäßige Aktualisierung des Layers in festgelegten Zeitabständen ausgeführt. Um einen Timestamp für die letzte Aktualisierung zu bekommen, wird die Zeit im Layernamen ergänzt. Über die Python – Konsole in QGIS wird das Skript (Abb. 14, Seite 44) eingeladen und gestartet.

```

Auto_Update_Kwi_Skript.py
1 import threading
2 import datetime
3 import re
4
5 def autoUpdateLayers():
6     threading.Timer(300.0, autoUpdateLayers).start()
7     for layer in QgsProject.instance().mapLayers().values():
8         if 'Kiwisstation' in layer.name():
9             print('autoUpdating layer: '+layer.name())
10            layer.dataProvider().forceReload()
11            layer.setName(
12                re.sub(
13                    'Kiwisstation.*',
14                    'Kiwisstation_akt.'+datetime.datetime.now().strftime('%c'),
15                    layer.name()
16                )
17            )
18
19 autoUpdateLayers()

```

Abb. 14: Python – Skript für automatisches Intervall – Aktualisieren von Layer in QGIS

```

7:59:2023
22 autoUpdating layer: -Kiwisstation_akt.Sun Mar 19 13:5
2:59:2023
23 autoUpdating layer: -Kiwisstation_akt.Sun Mar 19 13:5
7:59:2023
24 autoUpdating layer: -Kiwisstation_akt.Sun Mar 19 14:0

```

Abb. 15: Protokoll vom Python – Skript

Das Skript updatet den angesprochenen Layer (hier, wenn der Begriff „Kiwisstation“ im Layernamen auftaucht) alle fünf Minuten und ergänzt/aktualisiert einen Timestamp im Layernamen. Somit werden in QGIS die vor Ort erfassten Werte direkt angezeigt und der Timestamp ermöglicht eine sofortige Blick – Kontrolle, ob die Aktualisierung erfolgt ist. Ob das Python – Skript erfolgreich läuft, lässt sich im Protokoll (Abb. 15, Seite 44) nachverfolgen. Das Skript wurde auf *GitHub Gist* freundlicherweise für die freie Verwendung von Keith Jenkins zur Verfügung gestellt (Jenkins 2022).

5.4) Durchführen von Messungen

Als möglicher Use – Case für solche Anwendungen wurden Bergbau – Halden identifiziert. Im Ruhrgebiet gibt es eine große Anzahl an solchen Halden, die durch Aufschüttungen von Nebengestein aus dem Steinkohlebergbau (Schüppel and Wrede 2022) entstanden sind. Durch Restkohle im Abraum kann durch Sauerstoffzufuhr und Druck die Halde anfangen zu brennen. Solche nennt man „brennende Halden“. Davon gibt es nachweislich noch mehrere, die aktiv sind, darunter auch die Halde Graf Moltke in Gladbeck oder die Halde Rheinelbe in Gelsenkirchen.

Die Halde in Gladbeck ist teilweise gesperrt und nicht zugänglich, auf Nachfrage wurde auch keine Genehmigung zum Betreten gegeben. Diese Halde wird aktuell saniert und es wird versucht, den Brand weiter einzudämmen. Der Brand auf der Halde Rheinelbe wurde in den Neunzigern bereits eingedämmt.

Durch diese Halden sind in einer eher flachen Landschaft nun plötzlich Erhebungen entstanden, die Hänge in alle Himmelsrichtungen aufzeigen. Das sorgt für Unterschiede im Klima auf kleinsten Raum, zum Beispiel bei der Besonnung (Geiger 1942).

Aufgrund der Nähe und dass diese zu den höchsten Halden (knapp 200 Meter NHN) im Ruhrgebiet zählt, wird für dieses Beispiel die Halde Haniel in Bottrop ausgewählt (Abb.16).



Abb. 16: Untersuchte Halden in nördlichen Ruhrgebiet

Im Vorfeld wird eine Route festgelegt, um möglichst verschiedene Standorte auf und an der Halde zu begehen und Messwerte zu erfassen (Abb. 17, Seite 46):

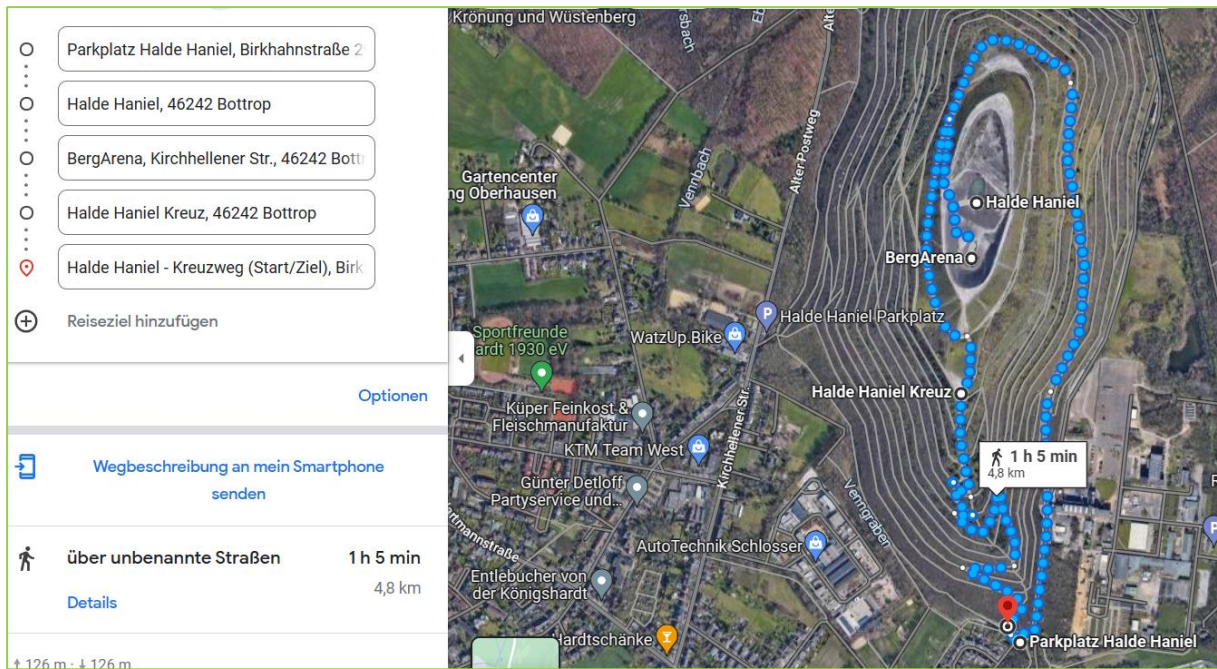


Abb. 17: Routenplanung in Google Maps

Die geplante Route wird auf das eigene Smartphone geladen und kann beim Durchführen der Messungen direkt abgelaufen werden.

Am gewählten Tag der Durchführung wird mit der programmierten senseBox sowie dem eigenen Smartphone als Stromquelle und Hotspot die Halde Haniel mit der geplanten Route abgelaufen. Parallel dazu wird der PC vorbereitet: Die Aufgabe im Aufgabenplaner bekommt eine bestimmte Uhrzeit, an dem es die Aufgabe starten soll als Trigger (Abb. 18, Seite 47). Die gestartete Aufgabe spricht die .cmd – Datei an.

Die Batch – Datei öffnet die Workbench (Abb. 06) und führt das Skript aus, welches die KML – Datei mit den Daten vom OSeM – Server erzeugt. Die Erläuterungen für die Workbench im Detail sind im Kapitel 5.1) zu finden. Diese Aufgabe wird alle fünf Minuten automatisch durchgeführt.

Parallel nimmt die aktivierte senseBox auf der Halde ebenfalls alle fünf Minuten einen Standort auf und erfasst die Temperatur dazu. Über den Hotspot werden die erfassten Daten an den Server gesendet. Auf dem Smartphone kann eine Sichtprüfung für die bisher übermittelten Daten auf der mobilen openSenseMap – Seite durchgeführt werden (Abb. 19, Seite 47).

Am PC wird zeitgleich eine Instanz von QGIS geöffnet und die KML einmalig als Layer eingeladen. Über die Erweiterungen wird die Python – Konsole geöffnet sowie das in Kapitel 5.3 vorgestellte Skript gestartet. Nun wird alle fünf Minuten der angesprochene

Layer aktualisiert, also die KML neu eingeladen. Dabei wird im Layernamen ein Timestamp geschrieben (Abb. 20, Seite 48). Die QGIS – Instanz bleibt dabei stabil geöffnet und kann weiterhin genutzt werden. Das wurde im Rahmen dieser Master – Thesis nicht objektiv untersucht. Hierfür müssten technische Parameter untersucht werden und eine laufende Kontrolle am Arbeitsplatz erfolgen, was unter diesen Projektbedingungen nicht möglich ist.

Ist das Ablaufen der Halde beendet, kann die senseBox ausgeschaltet werden. Am Arbeitsplatz wird die Aufgabe im Aufgabenplaner deaktiviert und die aufgenommenen Ergebnisse können betrachtet werden. In den Protokollen sowie der Verlaufsbeschreibung wird kontrolliert, ob alle Instanzen erfolgreich gelaufen sind.

Name	Status	Trigger	Nächste Laufzeit	Letzte Laufzeit	Ergebnis der letzten Ausführung	Autor	Erstellt
Create_GPK...	Bereit	Um 12:30 Uhr am 19.03.2023 - Nach Auslösung alle 5 Minuten für die Dauer von 1 Tag wiederholen.		19.03.2023 14:40:01	Der Vorgang wurde erfolgreich beendet. (0x0)	SCHMIDDENS/Musta	28.02.2023 20

Abb. 18: Definieren von Trigger und Laufzeit im Aufgabenplaner

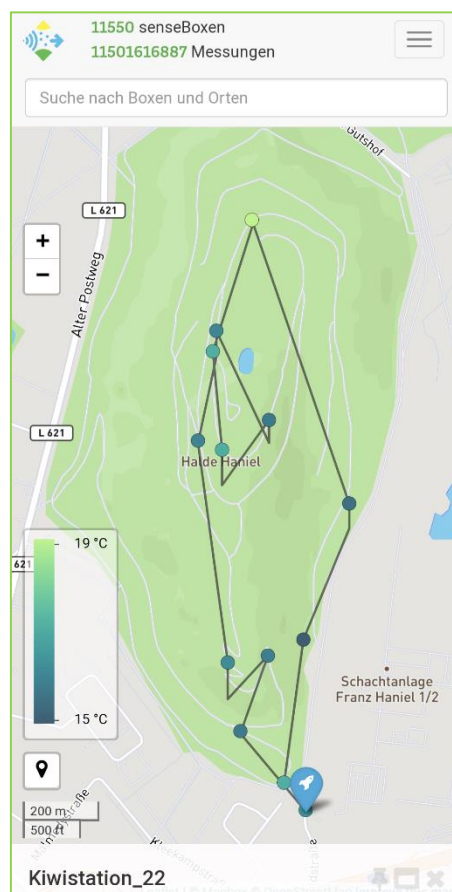


Abb. 19: openSenseMap auf dem Smartphone

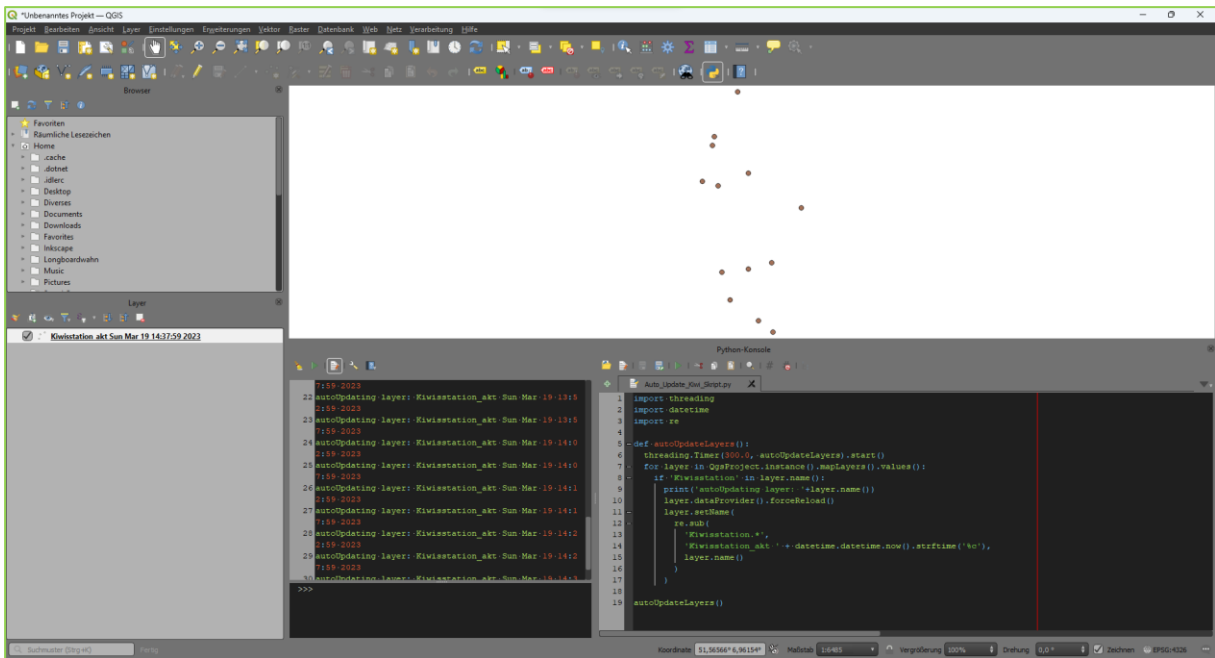


Abb. 20: QGIS Instanz mit den Messpunkten, Python – Skript, den zugehörigen Verlauf und Timestamp im Layernamen



Abb. 21: Das Bergtheater auf der Spitze der Halde Haniel

In der Beispielkarte (Anhang 02) werden die Temperaturunterschiede auf der Halde ersichtlich. Die Durchschnittstemperatur an diesen Tag betrug etwa 13 Grad. Innerhalb der Erfassung von etwa einer Stunde ist eine Differenz von etwa 4 Grad festgestellt worden. Das ist darauf zurückzuführen, dass in einer sonst flachen Landschaft die Halde herausragt und verschiedenen Wetterphänomenen stärker

ausgesetzt ist. Der nördliche Punkt mit etwa 19 Grad war zum Zeitpunkt der Messung der am stärksten besonnte Hang, während in den mittleren Höhenlagen die Temperaturen niedriger waren. Für eine objektivere Auswertung braucht man weitere Messwerte sowie die Angaben von Zeit sowie die Höhenwerte der Punkte. Die Auswertung solcher Daten ist aber kein Bestandteil dieser Master – Thesis.

6) Ergebnisse

Während der gesamten Projektlaufzeit von etwa einer Stunde wurden dreizehn Messpunkte aufgenommen. Das entspricht gemäß der Intervalle von fünf Minuten auch der zu erwartenden Anzahl an Werten.

Die Werte wurden alle vollständig und korrekt an den OSeM – Server übergeben (Abb. 22, Seite 56). In einen Trockentest vor der Durchführung auf der Halde Haniel wurde eine Datenerfassung mit weniger als fünf Minuten zeitlichen Abstand, hier mit drei Minuten, getestet. Es wurde nicht kontinuierlich in gleichen Zeitabständen die Daten erfasst und an den Server gesendet. Die senseBox versuchte eine Verbindung aufzubauen, konnte diese aufgrund der hohen Datenlast nicht aufrechterhalten. Dadurch wurde der Upload abgebrochen und es wurden in verschiedenen Zeitabständen Daten hochgeladen.

Im Aufgabenplaner ist an der definierten Aufgabe der Verlauf, der belegt, dass alle fünf Minuten die Batch – Datei geöffnet wurde (Abb. 23, Seite 57). Die Aufgabe wird durch den Anwender dann manuell beendet. Die Workbench erzeugt alle fünf Minuten eine neue KML – Datei (Abb. 24, Seite 57), wenn auf den Server neue Daten vorhanden sind. Diese werden dann in der neuen KML – Datei mit abgespeichert.

In der Workbench wird per JSON – Aufruf die senseBox mit den Werten auf dem Server abgefragt. Wenn kein Zeitraum genannt wird, werden standardmäßig die letzten 48 Stunden herangezogen (siehe Kapitel 5.1).

In der letzten Etappe, in einer QGIS – Instanz, wird protokolliert, dass alle fünf Minuten der Layer für die Punkte aus der KML – Datei neu geladen und mit einem Timestamp versehen wurde (Abb. 20, Seite 48). Es ist also an allen notwendigen Abschnitten protokolliert und nachweisbar, dass die vor Ort erfassten Daten im definierten Zeitraum bis zu einem Desktop – GIS eingeladen werden konnten.

Die eingeladenen Daten können beispielsweise mit den Bordmitteln von QGIS weiterverarbeitet werden. In diesem Beispiel wird eine statische Karte erzeugt (Anhang 02).

Die erzeugte KML – Datei kann ebenfalls weiterverarbeitet werden, zum Beispiel zur Visualisierung und Veranschaulichung.

In beiden Fällen handelt es sich um statische Ergebnisse, d.h. sollten in ein laufendes Projekt weitere Messpunkte erfasst werden, werden diese nicht mit aktualisiert. Es ist also der Fokus auf mögliche dynamische Weiterverarbeitungen zu legen, bei denen die Werte der Messungen eine Auswirkung haben können. Diese Weiterverarbeitung kann in QGIS erfolgen mithilfe von Python – Skripten oder auch in FME selbst, beispielsweise mit einem automatischen E-Mail – Versand, wenn bestimmte Ereignisse wie Temperaturüberschreitungen (Events) eintreten.

Die erzeugte KML kann in einen Geoserver geladen werden, wo diese für gebräuchliche OGC – Dienste (WMS, WFS, ...) zur Verfügung gestellt werden kann.

Nachfolgend wird mit der aus Kapitel 3 bekannten Tabelle ein Vergleich angestrebt. Damit wird die angewendete Architektur mit den verglichenen Anbietern aus dem Kapitel examiniert.

Datenbereitstellung	Format	Kosten	Wird hier nicht betrachtet
		Datenformate	Kann über die Workbench definiert werden, hier KML
		Zugänglichkeit	Kann frei definiert werden: Zugriff auf die Datei oder Bereitstellung in Server
	Schnittstellen	Schnittstellen	Hier nicht konfiguriert
		Webdienste	Hier nicht konfiguriert
	Weiterverarbeitung	Architektur	In Kapitel 5 erläutert
		Datenstruktur maschinell lesbar	Ja
		Datenmengen Größe	Je nach Ausprägung gering
		Bereitstellung möglich	Ja
	Zeitlicher Aspekt	Häufigkeit	Häufigkeit Messungen
Zeitstempel			Ja, an Datei und Objekten
Bereitstellung		Häufigkeit Bereitstellung	Alle fünf Minuten
		Zeitdifferenz Bereitstellung/Weiter- verarbeitendes Tool	Maximal fünf Minuten
Format		Datumsformat	yyyy:mm:ddThh:mm:ss
		Umgang mit in der Vergangenheit liegenden Daten	Wenn Messungen älter als 48 Stunden sind, wird die Datei nicht mehr aktualisiert -> manuelle Archivierung möglich

		GI – Systeme können das Format auslesen	Ja
Koordinaten	Art	Koordinatensystem	EPSG:4326
		Metadaten	Keine vorhanden
	Qualität	Koordinatenstandard	EPSG:4326
		Punkte verortbar	Ja

Tabelle 06: Vergleich eigene Datenbereitstellung in Echtzeit

In dieser praktischen Ausführung wurde die Bereitstellung der Daten als OGC – konformes Format auf einen Server nicht angegangen. Es fehlt hier also die serverseitige Bereitstellung über zum Beispiel Geoserver.

Betrachtet man den Vergleich, ist diese Architektur flexibel skalierbar. Eine Vielzahl von Writern steht in FME zur Verfügung. Es ist also möglich, auf die jeweiligen Ansprüche für den zu nutzenden Server das Format zuzuschneiden. Somit ist eine Schnittstelle gegeben, die für die individuelle Datenbereitstellung genutzt werden kann.

Des Weiteren ermöglicht die Anbindung in einem Desktop – GIS die umfangreiche Weiterverarbeitung und Visualisierung der Daten mit Werkzeugen, die beispielsweise in einer Cloud – GIS – Lösung nicht gegeben sind. Wie in Kapitel 1.1) beschrieben, bieten Cloud – basierte GIS – Lösungen keinen großen Umfang für die Aufbereitung der Daten an (*Fast and Hossain 2020*). Eine nachgelagerte Bereitstellung der aufbereiteten Daten über das Desktop – GIS in eine cloud – basierte Lösung oder als OGC – Dienst (z.B. WFS) sind optional umsetzbar.

Weswegen der Weg über diese Architektur überhaupt gegangen werden muss, ist dem Umstand geschuldet, dass es keine Möglichkeit gibt, die Daten aus der openSenseMap direkt in einem Desktop – GIS anzubinden. Es wird zwar eine API – Schnittstelle angeboten (*openSenseMap 2023*), diese wird aber nicht im Desktop – GIS ohne weitere Bearbeitung der Daten herangezogen werden können. Grundsätzlich werden häufig RESTful API – Schnittstellen von Datenanbietern angeboten, diese sind oftmals aber nicht nach bestimmten Standards strukturiert (*Bandara, Yashiro et al. 2016*).

Der komplette Prozess kann automatisiert und maschinell durchlaufen werden. Es ist zu keinem Zeitpunkt während des laufenden Projektes notwendig, manuell einzugreifen.

In Hinblick auf die Konstruktion dieser Architektur kann nicht im vollen Umfang von einer offenen Architektur gesprochen werden: Der Baustein FME, welcher für die Datentransformation benötigt wird, ist eine kostenpflichtige geschlossene Software. Die meisten anderen Komponenten erfüllen den Anspruch von Open Source (*Initiative 2022*):

- senseBox

- openSenseMap
- OGC Datenformate
- QGIS

Die zeitlich kleinst – mögliche Einheit für die Erfassung mit der senseBox sind aktuell fünf Minuten (*Network 2022*). Dementsprechend sind alle anderen zeitlichen Parameter der Komponenten (Aufgabe im Aufgabenplaner und das Python – Skript in QGIS) ebenfalls auf fünf Minuten definiert. Wie kritisch dieser Zeitraum zu bewerten ist, hängt stark vom Einsatzgebiet und den gewünschten Ergebnissen ab. In diesem praktischen Beispiel ist die Erfassung von Temperaturen auf einem Gebiet von ca. 110 ha mit einem Höhenunterschied von fast 160 Metern in einen Zeitraum von fünf Minuten ausreichend. Die Dichte der Messpunkte variiert auch stark mit der Geschwindigkeit der Durchführung der Messungen. Würde man im Schnitt halb so schnell laufen, hätte man doppelt so viele Werte.

Das Datumsformat kann in der FME – Workbench nahezu frei konfiguriert werden. Hierbei muss beachtet werden, was die jeweilige nachgelagerte Instanz auslesen kann. In diesen Fall werden in QGIS die Werte im Datumsformat als Datentyp string eingeladen, können dann über den Feldrechner in den Datentyp datetime umgewandelt werden (Abb. 25, Seite 57). Damit können in QGIS die Daten mit der zeitlichen Komponente weiterverarbeitet werden.

Durch die Erzeugung einer Datei aus der Workbench kann diese kopiert, verschoben oder gelöscht werden. Sobald in der Abfrage im GET – Aufruf keine neuen Werte gefunden werden (also älter als 48 Stunden sind), wird keine neue Datei mehr erzeugt. Die Archivierung der Dateien kann automatisiert werden.

Als Koordinatensystem ist das geläufige EPSG:4326 (WGS 84 – Geographic) definiert, welches von den meisten GI – Anwendungen gelesen werden kann. Auch dieses kann in der FME Workbench konfiguriert werden.

In dieser praktischen Ausführung wurden keine Metadaten definiert. Es ist möglich, in der gleichen Workbench die Inhalte für Metadaten zu definieren und als ISO 19115 XML – File auszugeben. Je nach definierten Ausgabeformat werden die Metadaten direkt in die Datei geschrieben (*Knowledge 2022*).

Die in openSenseMap – Server hochgeladenen und darüber bereitgestellten Daten werden nur mit den notwendigsten Metadaten über die Weboberfläche beschrieben (Abb. 26, Seite 57). Diese Angaben sind nicht nach einer gültigen Norm definiert und werden nicht in einen Metadaten – Katalog bereitgestellt. Für eine Erreichbarkeit und Auffindbarkeit sind also weitere Schritte erforderlich.

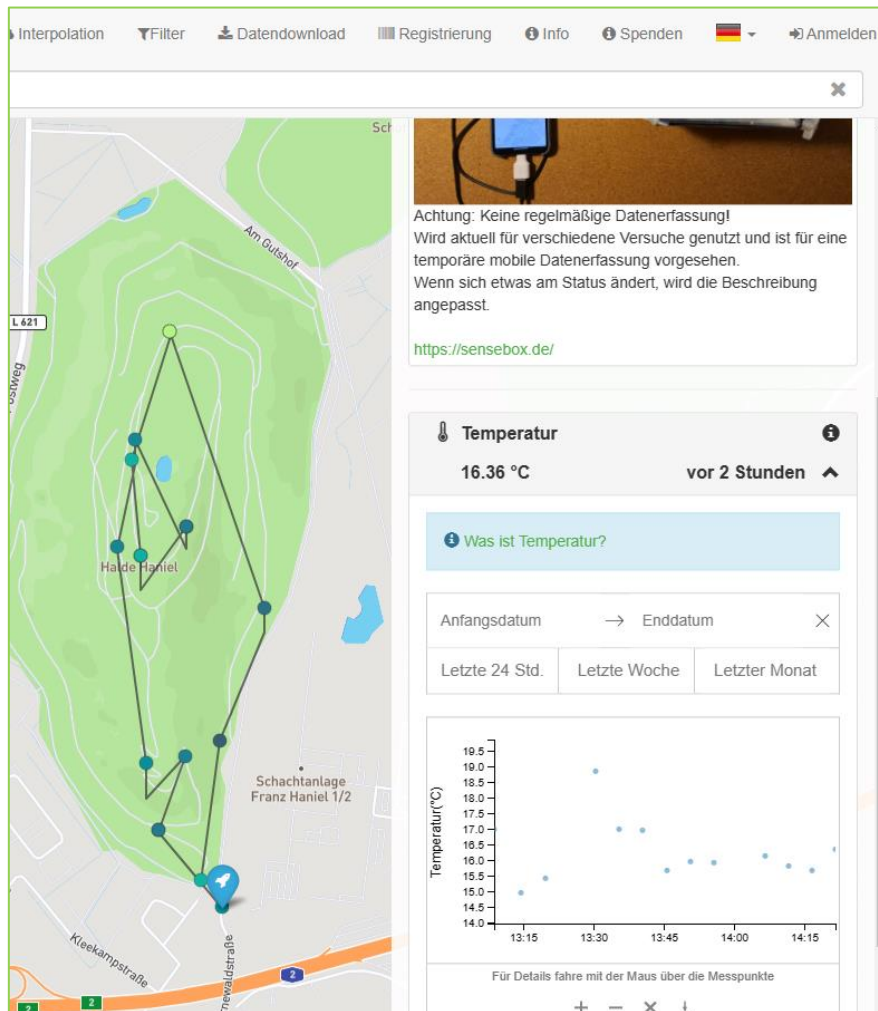


Abb. 22: Ausschnitt openSenseMap

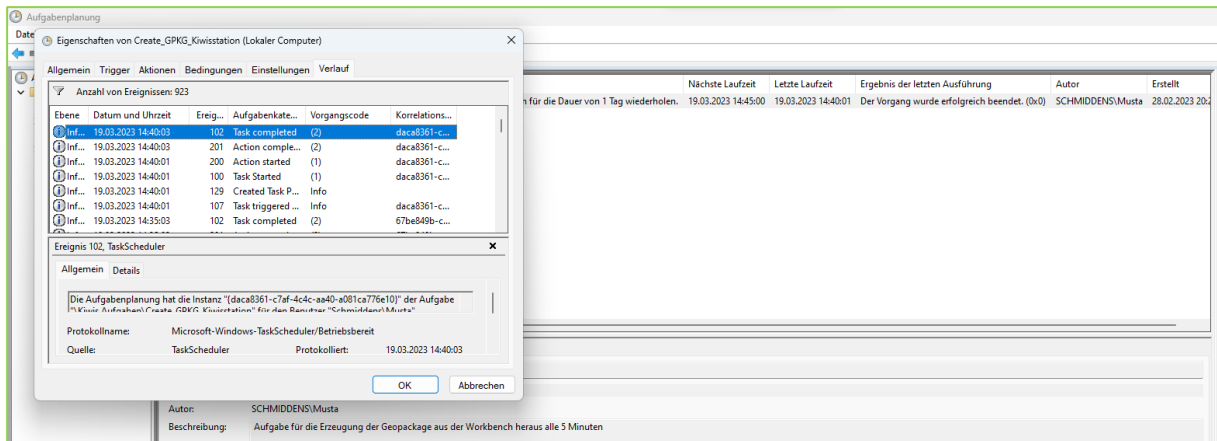


Abb. 23: Verlauf der Aufgabe im Aufgabenplaner

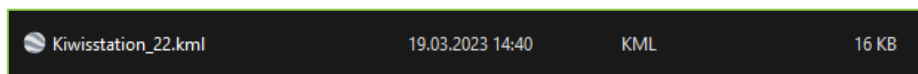


Abb. 24: Die erzeugte Datei mit dem „zuletzt geändert“ – Datum

Name	description	resta	begir	end	jdelv	sstella	struid	sibill	swOn	icon	Temperatur C°	Rechtswert	Hochwert	Hoehe	koordinatensystem	Datetime	hippe	np_D	date_date
2023-03-19T14:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	16.36	6.878906	51.54142	632.87	EPSPG:4326	2023-03-19T14:21:34	16.36	19.03.2023	14:21:34 (Mitteleuropäische Zeit)
2023-03-19T14:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.67	6.876457	51.543266	949.54	EPSPG:4326	2023-03-19T14:16:33	15.67	19.03.2023	14:16:33 (Mitteleuropäische Zeit)
2023-03-19T14:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.81	6.877494	51.545029	1231.5	EPSPG:4326	2023-03-19T14:11:32	15.81	19.03.2023	14:11:32 (Mitteleuropäische Zeit)
2023-03-19T14:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	16.13	6.875989	51.54488	1269.78	EPSPG:4326	2023-03-19T14:06:31	16.13	19.03.2023	14:06:31 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.92	6.874878	51.550053	1651.66	EPSPG:4326	2023-03-19T13:55:40	15.92	19.03.2023	13:55:40 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.96	6.875558	51.552616	1637.23	EPSPG:4326	2023-03-19T13:50:39	15.96	19.03.2023	13:50:39 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.68	6.877513	51.550529	1722.54	EPSPG:4326	2023-03-19T13:45:38	15.68	19.03.2023	13:45:38 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	16.95	6.875773	51.54982	1750.1	EPSPG:4326	2023-03-19T13:40:36	16.95	19.03.2023	13:40:36 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	16.99	6.875444	51.55212	1616.29	EPSPG:4326	2023-03-19T13:35:35	16.99	19.03.2023	13:35:35 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	18.86	6.876896	51.555206	1581.33	EPSPG:4326	2023-03-19T13:30:33	18.86	19.03.2023	13:30:33 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	15.41	6.880514	51.54856499999...	1437.49	EPSPG:4326	2023-03-19T13:19:44	15.41	19.03.2023	13:19:44 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	14.95	6.878829	51.545414	837.85	EPSPG:4326	2023-03-19T13:14:43	14.95	19.03.2023	13:14:43 (Mitteleuropäische Zeit)
2023-03-19T13:...	<tr><tr bgcol...	NUL	NUL	NUL	NUL	-1	0	1	NUL	NUL	16.99	6.878009	51.542076	776.73	EPSPG:4326	2023-03-19T13:08:55	16.99	19.03.2023	13:08:55 (Mitteleuropäische Zeit)

Abb. 25: Attributtabelle in QGIS

Allgemein

Name*

Gruppenkennzeichnung

Standort*

Beschreibung

Achtung: Keine regelmäßige D...
Wird aktuell für verschiedene V...
Wenn sich etwas am Status än...

Webseite

Bild hochladen

Abb. 26: Metadaten in openSenseMap an der Box

7) Diskussion

7.1) Definition Echtzeit

Im Vergleich von verschiedenen Anbietern (Kapitel 3) fällt auf, dass aktuell das Bereitstellen von Real Time Data im Zusammenhang mit der Anbindung über eine Schnittstelle in ein weiterverarbeitendes Programm keine weite Verbreitung hat. Die im Vergleich ausgewählten Anbieter bieten Daten zwar an, diese müssen aber entweder heruntergeladen werden oder sind nicht aktuell. Wann kann man von Echtzeitdaten sprechen:

Ab wann etwas als Real Time angesehen oder von Real Time Data Streaming gesprochen werden kann, hängt von den festgelegten Rahmenbedingungen des Projektes ab. Beispielsweise bietet der Anbieter Bundesamt für Strahlung ODL einen WFS – Dienst an, in den auf die vergangene Stunde aktuell Daten bereitgestellt werden. Hier müsste in einem Desktop – GIS eine Funktion für das regelmäßige Aktualisieren der Layer und Darstellung erfolgen. Dann kann nach vorher festgelegten Kriterien im Projekt von einem Echtzeitdatenstream gesprochen werden.

Laut Definition im Duden (*Duden o.J.*) bedeutet Echtzeit, dass bestimmte Ergebnisse zuverlässig in einer definierten Zeitspanne geliefert werden können. Dabei kann es sich also je nach Einsatzgebiet um Sekunden bis zu Tagen handeln. Die Schnittstellen bei den untersuchten Anbietern stellen die Daten auf die Stunde genau bereit. Gerade für Sensordaten außerhalb von Gebäuden und kritischen, explizit zu untersuchenden Gebieten ist die Zeitspanne ausreichend (Temperaturen, Strahlungswerte, Luftqualitäten, etc.). Bei den untersuchten Anbietern bietet keiner eine Schnittstelle mit den neuen OGC – Standards als API – Schnittstellen für Sensor – Daten an. Entweder sind es Datensätze zum Download in verschiedenen Formaten, hier zum Beispiel als comma – separated values (csv) – Datei, oder wenn die Daten als Schnittstelle angeboten werden, oftmals in veralteten Standards, zum Beispiel als Web Feature Service (WFS) – Dienst. Grundsätzlich ist somit die Struktur für ein Real Time Data Streaming also bei nahezu allen Anbietern vorhanden. Folgende Anpassungen wären nötig, damit ein nach aktuellen Standards geltender Echtzeitstream in einem Desktop – GIS möglich wäre:

- Bereitstellung der Daten in einer Schnittstelle nach aktuellem Standard des Open Geospatial Consortiums (beispielsweise als API – Schnittstelle für Sensordaten).

- Festlegen der Aktualisierungsintervalle: Der Anbieter muss klar kommunizieren, in welchen Zeitabständen die Daten erfasst sowie in welchen Zeitabständen diese erfassten Daten bereitgestellt werden.
- Die eingesetzte Software beim Enduser muss in der Lage sein können, die angebundene Schnittstelle automatisch zu aktualisieren, sprich dem vom Anbieter aktualisierten Datenbestand auch in der eingesetzten Software darzustellen.

Real Time Data Streaming kann also auch in größeren Zeitabständen stattfinden, wie im vergleichenden Teil dieser Master – Thesis nachgewiesen wurde. Im praktischen Beispiel wurde nachgewiesen, dass ein Desktop – GIS – Anwender mit der erprobten Architektur Echtzeitdaten auf fünf Minuten genau empfangen und weiterverarbeiten kann. Eine häufigere Erfassung von Daten im gleichen Zeitraum kann durch Festlegen von geringeren Zeitabständen bewerkstelligt werden.

Aktuell wird von The Things Network ein Zeitabstand von fünf Minuten empfohlen (*Network 2022*). Im selben Artikel wird ebenfalls beschrieben, dass das LoRaWAN (Long Range Wide Area Network) nicht geeignet für Realtime Data ist. Wie bereits in einen vorherigen Abschnitt in diesem Kapitel erwähnt wird, ist es abhängig vom Einsatzgebiet der Datenerfassung und Auswertung.

Die Systeme können ebenfalls so programmiert werden, dass nur Daten erfasst werden, wenn beispielsweise ein bestimmter Wert überschritten wird oder der Sensor eine Änderung registriert. Dann handelt es sich um eine eventbasierte Datenerfassung und nicht um ein reines Datenstreaming.

7.2) Erfassung von Messwerten

Bei der Messstation ist zu unterscheiden zwischen einer stationär aufgebauten und einer mobilen Einheit. Im praktischen Teil dieser Master – Thesis wurde aus dem Grunde eine mobile Messstation konzipiert.

Wie in der Einleitung dieser Master – Thesis erwähnt wird, findet das Erfassen von Sensordaten immer mehr Verbreitung. Allgemein sind technische Gerätschaften, die mobil handzuhaben sind, immer flexibler als solche stationär festverbaute. Daher bieten sich solche tragbaren Messstationen an, die flexibel für verschiedene Einsatzszenarien angewendet werden können. Wesentliche Unterschiede in der

Erfassung von Daten zwischen einer stationär verorteten sowie einer mobilen Messeinrichtung sind:

- Mobil findet eine Zuordnung des gemessenen Sensor – Wertes zu dem aktuellen Standort statt. Es kann also eine Verschneidung zwischen den Messwerten mit den zugehörigen Standorten innerhalb eines Datensatzes durchgeführt werden. Hier gilt, pro gemessenem Punkt gibt es eine individuelle Koordinate. Jedes Objekt setzt sich also einzigartig aus seiner Koordinate, dem Timestamp sowie des Messwertes zusammen.

- Stationär findet bei den Daten die Unterscheidung nur zwischen den Messwerten und dem Timestamp statt. Es muss keine Koordinate pro Messung erfasst werden, diese kann vorher festgelegt werden. Beispielsweise in der openSenseMap kann über das Festlegen eines Standortes in einer Karte der Standort definiert werden.

Gemein haben beide Varianten die Angabe des Zeitraumes sowie eines Messwertes. Beide Komponenten können umfangreich in einem Desktop – GIS weiterverarbeitet werden. Durch die Erfassung von Messwerten mit verschiedenen Standortangaben bieten sich in einem Desktop – GIS weitere tiefgehende Analysen an, beispielsweise mit einer räumlichen Verschneidung oder der Abbildung von Verläufen.

Zwischen der Erfassung von Daten in einem Real Time Data Streaming – System und einer normalen Erfassung mit Datenspeicherung auf einem lokalen Gerät sind folgende Unterschiede festzustellen:

- Bei einer Echtzeiterfassung entsteht kein Zeitverlust durch die manuelle Nachbearbeitung von Daten, die vom Speichergerät gelesen werden müssen. Der Datenstream läuft vom Moment der Erfassung eines Standortes und seinen Informationen durch bis zum Desktop – GIS des Anwenders. Dies wäre für den genannten Use – Case der Erfassung von Feinstaubwerten mit Fahrrädern in urbanen Gebieten eine sinnvolle Ergänzung (*Futurium 2021*).

- Der Mehraufwand für den Aufbau und die Inbetriebnahme einer solchen mobilen Messstation, welches die Daten in Echtzeit weitersendet, ist im Vergleich zu einer stationären Messeinheit gering. Es muss zwingend eine mobile Internetverbindung hergestellt werden (zum Beispiel über einen Hotspot vom Smartphone).

- Der größte Unterschied besteht im Aufbau einer solchen Echtzeitdateninfrastruktur: Es wird ein Mehr an Software sowie Programmen benötigt, damit der Datenstream erfolgreich bis zum Anwender (Desktop – GIS) durchläuft. Im praktischen Teil der Master – Thesis wurden dafür unter anderem die Software Safe FME und Windows Aufgabenplaner benötigt. Im GIS, hier QGIS, muss ein Python – Skript angewendet werden.

7.3) Software

In dieser Master – Thesis wurde insbesondere im praktischen Teil versucht, möglichst Software im Sinne des Open Source zu verwenden. Das senseBox – System mit der openSenseMap sowie QGIS werden im Sinne des Open Source zur Verfügung gestellt.

Den Schwerpunkt auf Open Source – Software zu legen, kann insbesondere kostentechnisch vom Vorteil sein, wenn die Anwender auf eine möglichst günstige Umsetzung setzen müssen.

7.4) Bereitstellung im Server

In dieser Master – Thesis wurde nicht betrachtet, inwieweit eine Bereitstellung solcher aufbereiteten Daten in einen weiteren Server möglich sind oder eine sinnvolle Ergänzung dieser Prozesskette darstellen. Die im praktischen Beispiel genutzte Architektur von openSenseMap als Server bietet aktuell keine direkte Möglichkeit zur Anbindung der Daten in einem Desktop – GIS an. Würde man die aufbereiteten Daten ebenfalls als Serverlösung bereitstellen wollen, würde die Architektur aufgebläht werden. Um das zu vermeiden, müssten die Daten aus dem openSenseMap – Server in einer vom Desktop – GIS auslesbaren Schnittstelle bereitgestellt werden können. Dies kann über entsprechende Erweiterungen in beispielsweise Geoserver von Open Source Geospatial Foundation erfolgen. Dort können die Daten sowohl als WFS – Dienst oder auch als API – Features – Schnittstelle bereitgestellt werden (*Santhanavanich, Padsala et al. 2022*).

Zusätzlich zur möglichen Bereitstellung der Daten in einen weiteren Server, beispielsweise Geoserver, muss eine automatische Aktualisierung der Daten an die Schnittstelle in bestimmten Zeitabständen ermöglicht werden. In der üblichen Vorgehensweise wird eine Datei in den Geoserver geladen, welche dann als Dienst

bereitgestellt werden kann. Hierzu müsste also im Vorfeld eine Programmierung des Uploads in den Server stattfinden, dass diese Datei in regelmäßigen Abständen aktualisiert wird.

Für die Bereitstellung einer Schnittstelle als SensorThings – API wurde vom Fraunhofer Institut aus München der FROST – Server entwickelt (*Santhanavanich, Padsala et al. 2022*). Dieser Server wäre für eine Einbindung in die konzeptionierte Architektur möglich.

In ArcGIS Online können neue Layer über eine URL hinzugefügt werden. Bei den zugelassenen URLs sind unter anderem OGC – API – Features und andere OGC – Services enthalten. Werden die Daten aus dem praktischen Teil in einen Server geladen, der OGC – API – Features bereitstellen kann, ist es möglich, diese in einer ArcGIS Online Karte darzustellen. Über das Aktualisieren des Browserfensters werden auch die Inhalte aktualisiert dargestellt.

7.5) Vergleich Anforderungen

Zuletzt werden die in Kapitel 3.1) von (*Fischer, Gras et al. 2021*) definierten Anforderungen für eine echtzeitfähige Urban Data Platform aufgegriffen und bewertet:

- 1) Es erfolgt eine eventbasierte Datenbereitstellung in Echtzeit.
- 2) Es ist eine kostengünstige und zuverlässige Datenbereitstellung.
- 3) Weitergehende INSPIRE – Richtlinien wurden in diesen praktischen Teil nicht betrachtet.
- 4) Es ist keine sichere Datenbereitstellung gegeben, da die Bereitstellung der Daten auf openSenseMap erfolgt, welche frei zugänglich ist.

Zusammengefasst ist also festzustellen, dass die Datenbereitstellung in Echtzeit zuverlässig und kostengünstig erfolgen kann. Allerdings werden hier die Aspekte der Einhaltung von Richtlinien und der Sicherheit für vertrauliche Daten nicht beachtet.

Es wurde im Verlauf dieser Master – Thesis mehrfach eine eventbasierte Datenbereitstellung als Anforderung definiert. Im praktischen Teil erfolgte keine Deklaration eines Events, nach dem Daten bereitgestellt werden. Es wurde unabhängig von weiteren Faktoren alle fünf Minuten ein neuer Messwert erfasst, damit ein Echtzeitdatenstream nachgewiesen werden kann. Nachgelagert können aber

eventabhängige Faktoren definiert werden, die eine Aktion auslösen können, beispielsweise eine Push – Messwertbereitstellung bei einer Schwellwertüberschreitung als MQTT – Message oder per E-Mail.

8) Fazit

8.1) Echtzeitdatenerfassung und Geostreaming

In dieser Master – Thesis wurde somit nachgewiesen, dass eine mobile Echtzeitdatenerfassung mit Real Time Data Streaming im kleineren Maßstab möglich ist. Werden die richtigen Werkzeuge genutzt, spielt es keine Rolle, ob die Daten stationär oder in einer mobilen Lösung erfasst werden. Wenn die Daten in einer Schnittstelle bereitgestellt werden, die von einem Desktop – GIS gelesen werden kann, ist eine Anbindung solcher im Echtzeit – Format möglich. Nachweislich wurde hier ein Echtzeit – Geostream (*Liu, Li et al. 2022*) aufgebaut, in dem die drei dazugehörigen Aspekte (Datenstream, räumliche und zeitliche Informationen) aufgegriffen worden sind. Das Desktop – GIS benötigt eine Funktion zur automatischen Aktualisierung der Verbindung bzw. des dargestellten Layers.

Die Erfassung selbst mit dem im praktischen Teil dieser Thesis vorgestellten Konfiguration ist mit sehr geringem Aufwand verbunden. Durch solche Bausätze, wie von senseBox vorgestellt, ist es möglich, ohne besondere Fachkenntnisse eine Datenerhebung nach eigenen Vorgaben durchzuführen.

8.2) Aufwand Aufbau

Der Aufwand vom Aufbau dieser Messstation ist überschaubar, hier werden keine Fachkenntnisse gebraucht, zahlreiche Anleitungen und Tutorials vereinfachen den Aufbau. Durch die graphische Programmieroberfläche „Blockly“ kann ohne Fachwissen der Programmiersprache Arduino das benötigte Programm nach eigenen Vorgaben programmiert werden.

Der Vorteil einer solchen Struktur ist die zeitliche Bereitstellung der Daten in einem Desktop – GIS. Einhergehend mit einem solchen Echtzeitdatenstreaming sind ein Wegfall von manueller Nachbereitung der Daten nach der Erfassung sowie die Anbindung im GIS.

Die konzipierte Architektur im praktischen Teil ist im Sinne des Open Data konfiguriert: Die erhobenen Daten werden kostenfrei und für alle nutzbar im frei zur Verfügung stehenden openSenseMap – Portal gestellt. Ein Teil der verwendeten Programme sind Open Source:

- Blockly von senseBox
- openSenseMap
- QGIS

Folgende Programme sind nicht Open Source:

- Aufgabenplaner (enthalten im Windows OS)
- Safe FME

Das hier gewählte OS Windows wurde aufgrund der Gegebenheiten ausgewählt (installiert auf dem ausführenden Rechner). Es sind bei FME von Safe Software Vorkenntnisse aus dem beruflichen Umfeld vorhanden.

8.3) Probleme

Der Vergleich aus Kapitel 3 verdeutlicht, dass Anbieter von Sensor – Daten noch nicht auf aktuelle OGC – Vorgaben umgestellt haben. Das OGC stellt speziell für Sensor – Daten eine API – Schnittstelle zur Verfügung, die für das Echtzeitdatenstreaming geeignet ist (*Consortium 2023*). Im Umkehrschluss bedeutet es, dass die Software für Desktop – GIS die OGC SensorThings API lesen können muss. Im Falle von QGIS existiert bereits ein Plug – In, welches Zugriff auf Daten von SensorThings – API – Schnittstellen ermöglichen soll.

Weiterhin fehlt die Möglichkeit einer automatisierten Aktualisierung der dargestellten Layer, also sobald neue Objekte auf der Datenbank erzeugt worden sind. Alternativ besteht die Möglichkeit, dass der Layer in bestimmten Abständen aktualisiert wird, so wie es hier im praktischen Teil programmiert wurde.

Der im Vergleich (Kapitel 3.3.2) betrachtete Anbieter Deutscher Wetterdienst stellt die Daten in einem WFS – Dienst bereit. Die Daten dort werden stündlich bereitgestellt und für jeden Messwert wird ein eigenes Objekt erzeugt. Diese Vorgehensweise eignet sich auch für eine mobile Messstation, wie die im praktischen Teil dieser Thesis konstruiert wurde. Der Anbieter kann die Daten über die jetzt bestehende Schnittstelle also auch für eine mobile Variante einer Messstation bereitstellen. Mögliche Einsatzgebiete können die Auswertung von verschiedenen Wetter – Phänomenen in einer mittelgroßen räumlichen Ausprägung sein. Dafür kann die konzeptionierte mobile Messstation im Zusammenhang mit der Schnittstelle des Deutschen Wetterdienstes

verwendet werden. Um diese Architektur auf den aktuellsten Stand der Technik zu heben, kann die Schnittstelle als beispielsweise Open Geospatial Consortium Features – API oder auch SensorThings – API definiert werden.

Wie in Kapitel 6 beschrieben, kann mit der im praktischen Teil erdachten Architektur kein Norm – gerechter Metadaten – Katalog aufgebaut werden, dafür wären weitere Schritte erforderlich.

8.4) Abgleich konzeptionierte Architektur

Im Vergleich zur konzeptionierten Architektur aus Kapitel 4 eines Real Time Data Streaming wurden mit der im praktischen Teil angewendeten Architektur folgende Kriterien angegangen und bewertet:

- Eine manuelle Vorbereitung von erfassten Daten entfällt durch die Transformation in FME komplett
- Die Struktur ist für Projekte bis zu mittlerer Größe gut geeignet aufgrund der vereinfachten Struktur ohne weitere Serveranbindung. Es ist also unter Umständen nicht erforderlich, dass eine dauerhafte Server – Architektur eingerichtet werden muss.
- Ein Desktop – GIS bietet nach wie vor den größten Umfang für nachgelagerte Auswertungen der erfassten Daten.
- Das System der senseBox kann mit einer Vielzahl von verschiedenen Sensoren ausgestattet werden. Je nach Projektziel ist ohne großen Aufwand eine Umprogrammierung auf den gewünschten Sensor möglich. Die Infrastruktur bleibt bestehen.
- Durch die Erfassung mit Zeitstempel und der Anpassung des Datumsformats in FME kann die zeitliche Dimension für die Auswertung oder Weiterverarbeitung im Desktop – GIS genutzt werden. Häufig finden Auswertungen ohne Berücksichtigung der zeitlichen Komponente statt.

8.5) Anwendungsgebiete

Mögliche Anwendungsbereiche eines Real Time Data Streaming sind beispielsweise die Erstellung eines digitalen Zwillings der Stadt oder Kommune. Ein digitaler Zwilling bildet die Stadt digital ab und wird mit Echtzeitdaten aktualisiert. Mit den betrachteten

Möglichkeiten in dieser Master – Thesis ist der Aufbau einer solchen Infrastruktur im mittleren Maßstab möglich. Als mittlerer Maßstab wird hier der Bereich von Kommunalverwaltung, Gemeinden oder Behörden mit dem Fokus auf einen kostengünstigen Aufbau und einer nachvollziehbaren Architektur verstanden. Zusätzlich sollte dabei der Schwerpunkt auf lokale Besonderheiten, wie in Kapitel 1.2.1 beschrieben, gesetzt werden.

Auch kann nach wie vor das Bildungssystem von solchen Ergänzungen für Citizen Science profitieren. Kleinmaßstäbliche Wetterphänomene werden beispielsweise von Schulklassen untersucht (*Kox, Rust et al. 2021*). Hier kann das Echtzeitdatenstreaming – Modell Anwendung finden.

Untersuchungen von Bergbauhalden auf die Auswirkungen ihrer Hanglagen (UV – Strahlung und Temperaturen durch unterschiedliche Besonnung) wie in Kapitel 1.2.1 beschrieben können mit der konzipierten Architektur in Echtzeitverfahren durchgeführt werden. Dazu kann die Messstation um einen UV – Sensor ergänzt werden. Auf dem Mainboard der senseBox ist bereits ein Sensor enthalten, der die Neigung erfassen kann. Somit kann also die Hangneigung ebenfalls ermittelt werden. Die erforderlichen Werte werden im Code mit programmiert und vor Ort im Projektgebiet mit erfasst.

Für ein Gebäudemanagement ist die mobile Struktur nicht geeignet. Da hierbei ein aktiver GPS – Empfang benötigt wird, um die Objekte zu erzeugen, kann davon ausgegangen werden, dass innerhalb von Gebäudekomplexen keine Daten erfasst werden können. Hier ist wiederum die lokal fest installierte Messstation zu bevorzugen, da in den erfassten Daten der Standort gleich bleibt und vom Anwender festgelegt wird. Für die Erfassung und Bereitstellung im Echtzeitdatenstream der Sensor – Daten selbst ist die erstellte mögliche Architektur dennoch anwendbar.

8.6) Ausschluss

Die Betrachtung einer möglichen Architektur mit einer Bereitstellung der Daten über einen Server, beispielsweise Geoserver, wurde in dieser Master – Thesis nicht angegangen. Die im praktischen Teil erfassten Daten wurden bereits in einen Server geladen und über eine API – Schnittstelle bereitgestellt. Es wurde festgestellt, dass keine direkte Anbindung mit Darstellung der Objekte als Echtzeitdatenstream möglich ist. Es hätten also keine weiteren Erkenntnisse über die zusätzliche Verwendung eines

Servers gewonnen werden können und die Architektur würde mit einer Schleife (vom Server in den Server) arbeiten müssen.

In Geoserver ist es Stand 2022 möglich, Daten über eine OGC API – Features bereitzustellen, jedoch nicht als SensorThingsAPI (*Blanc, Cannata et al. 2022*). Somit können die als KMZ bereitgestellten Daten auch direkt als Features in einen Server bereitgestellt werden mit den in Kapitel 7.4 beschriebenen Ergänzungen.

Diametral zum Erfassen von Messwerten und Bereitstellen als Echtzeitdatenstream wurde hier das Abfragen („polling“) von Messwerten in Echtzeit nicht betrachtet. Im praktischen Teil und der Konzeptionierung einer Architektur wurde nur die Bereitstellung der Daten einer mobilen Lösung in einem Desktop – GIS fokussiert. Für eine zyklische Abfrage von Messwerten böte sich eine stationäre Lösung an, an dem ein regelmäßiges polling durchgeführt werden kann.

9) Ausblick

In diesem Kapitel soll ein Ausblick über mögliche weitere wissenschaftliche Arbeiten gegeben werden, die im Zuge dieser Master – Thesis nicht behandelt worden sind.

Es wurde bereits im Fazit die Erkenntnis gezogen, dass die Erhebung von Sensordaten im Echtzeitdatenstreaming auch die zeitliche Dimension einbringt. Es ist herauszufinden, in welchem Umfang ein Desktop – GIS, zum Beispiel QGIS Sensordaten Zeitwerte weiterverwerten und analysieren kann.

Im vergleichenden Teil dieser Master – Thesis wurde am Beispiel des Deutschen Wetterdienstes ein Layer in QGIS eingeladen und festgestellt, dass dort sämtliche Wetterstationen mit allen Messwerten der letzten 24 Stunden dargestellt werden. Gerade im Hinblick auf Controlling – Prozesse, wie die beispielsweise bei der Überwachung vom Katastrophenschutz benötigt werden, sind oftmals bestimmte Stationen im Fokus. Ist ein Real Time Data Streaming in einem Desktop – GIS mit gezielt ausgewählten Messstationen für ein zeitechtes Controlling möglich?

In diesem Zusammenhang sollte auch wissenschaftlich evaluiert werden, für welche Einsatzgebiete oder Projekte eine wie im praktischen Teil dieser Master – Thesis aufgebaute Streaming – Architektur geeignet ist. Welche Szenarien sind dafür geeignet?

In dieser Thesis wurden Safe FME sowie Tools aus dem Windows OS verwendet. Um sämtliche Komponenten dieser Architektur im Sinne des Open Source zu strukturieren, sollte geprüft werden, ob die zu betrachtenden Bausteine durch solche ersetzt werden können. Kann eine Open Source Software wie beispielsweise hale Studio von wetransform den transformierenden Abschnitt in FME ersetzen? Als Betriebssystem wurde hier das Windows 11 als OS eingesetzt, kann die Struktur auf einen Open Source Betriebssystem übertragen werden oder kann es beispielsweise auf einen Linux – OS sogar umfangreicher konfiguriert werden?

Es ist festzustellen was die Vorteile einer direkten Bereitstellung von Sensordaten in einer OGC – konformen Schnittstelle, zum Beispiel als SensorThings – API, in Gegenüberstellung mit sonstigen Datenbereitstellungs – Formaten wäre. Im praktischen Teil dieser Master – Thesis wurden die Daten auf dem Server der openSenseMap gespeichert und in einer eigenen API – Schnittstelle bereitgestellt.

Diese API – Schnittstelle kann nicht unmittelbar von QGIS ausgelesen werden. In diesem Zuge muss recherchiert werden, was gegen eine Anbindung der SensorThings – API von Open Geospatial Consortium in Desktop – Geoinformationssystemen spricht. Aktuell ist es nicht möglich, direkt eine SensorThings – API anzubinden. In QGIS kann bereits die aktuelle API – Features verwendet werden.

Im praktischen Teil dieser Master – Thesis wird ein Python – Skript in QGIS verwendet, um den Layer mit den Objekten aus der laufenden Datenerfassung zu aktualisieren. Durch den Umstand, dass die Erfassung außerhalb der Griffweite vom Workspace stattfindet, kann hier nicht festgestellt werden, wie das Desktop – GIS sich verhält, wenn man parallel zur Datenerfassung die Daten weiterverarbeitet. Sollte es performant sein, welche Vorteile ergeben sich daraus?

Es werden Systeme für Citizen Science bereitgestellt, wie zum Beispiel Spotteron (*SPOTTERON 2023*). Spotteron bietet komplette Pakete für Citizen Science – Projekte oder für Umweltmonitoring an, die als Smartphone – Apps verwendet werden. Hier wäre ein Vergleich angebracht, ob mit der senseBox und der im praktischen Teil erdachten Struktur eine qualitative Verbesserung bei der Datenerhebung und Weiterverarbeitung erbracht werden kann.

In dieser Master – Thesis erfolgte die Erfassung der Temperaturdaten automatisch über die Messstation. Beispielsweise von QGIS gibt es mobile Lösungen (zum Beispiel als App für Android), womit räumliche Daten mobil auf dem Smartphone erfasst werden können. Ein Vergleich dieser Art von Messwerterfassung ist anzustreben.

Literaturverzeichnis

Abbas, A. H., M. I. Habelalmateen, S. Jurdi, L. Audah and N. A. M. Alduais (2019). "GPS based location monitoring system with geo-fencing capabilities." AIP Conference Proceedings 2173(1): 020014.

Akbari, M., H. Zahmatkesh and M. Eftekhari (2021). "A GIS-Based System for Real-Time Air Pollution Monitoring and Alerting Based on OGC Sensors Web Enablement Standards." Pollution 7(1): 25-41.

Al-Akel, I. S., C. Kunze, I. M. Müller and P. Schneider (2022). "Langzeitstabilität sanierter Uranbergbauhalden."

Bandara, S., T. Yashiro, N. Koshizuka and K. Sakamura (2016). Towards a standard API design for open services in smart buildings. 2016 TRON Symposium (TRONSHOW).

Banse, M. B., F. Schmalriede, O. Theel and A. Winter (2021). "Environmental Wellbeing through Guerilla Sensing." INFORMATIK 2021.

Bartoschek, T., J. Wirwahn and M. Pesch (2018). "senseBox und openSenseMap: Umweltmonitoring für Jedermann." Umweltinformationssysteme 2018- Umweltbeobachtung: Nah und Fern.

Blanc, N., M. Cannata, M. Collombin, O. Ertz, G. Giuliani and J. Ingensand (2022). "OGC API STATE OF PLAY – A PRACTICAL TESTBED FOR THE NATIONAL SPATIAL DATA INFRASTRUCTURE IN SWITZERLAND." Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLVIII-4/W1-2022: 59-65.

Borges, M. C., F. Pallas and M. Peise (2018). Providing Open Environmental Data— The Scalable and Web-Friendly Way. Advances and New Trends in Environmental Informatics, Cham, Springer International Publishing.

Burggraf, D. (2015). "OGC KML 2.3, Version 1.0."

Consortium, O. G. (2023). "An Introduction to OGC API - Features." Retrieved 27.05.2023, from <http://opengeospatial.github.io/e-learning/ogcapi-features/text/basic-main.html>.

- Consortium, O. G. (2023). "OGC API-Features." Retrieved 23.05.2023, from <https://ogcapi.ogc.org/features/>.
- Consortium, O. G. (2023). "OGC SensorThings API." Retrieved 07.05.2023, from <https://www.ogc.org/standard/sensorthings/>.
- Duden. (o.J.). "Echtzeit, die." 07.05.2023, from <https://www.duden.de/node/36444/revision/1334833>.
- Fast, V. and F. Hossain (2020). "An Alternative to Desktop GIS? Evaluating the Cartographic and Analytical Capabilities of WebGIS Platforms for Teaching." *The Cartographic Journal* 57(2): 175-186.
- Fekih, M. A., W. Bechkit, H. Rivano, M. Dahan, F. Renard, L. Alonso and F. Pineau (2021). "Participatory Air Quality and Urban Heat Islands Monitoring System." *IEEE Transactions on Instrumentation and Measurement* 70: 1-14.
- Fernández-Rodríguez, J. Y., J. A. Álvarez-García, J. Arias Fisteus, M. R. Luaces and V. Corcoba Magaña (2017). "Benchmarking real-time vehicle data streaming models for a smart city." *Information Systems* 72(**ISSN:** 03064379 **DOI:** 10.1016/j.is.2017.09.002): 77.
- Fischer, M., P. Gras, S. Löwa and S. Schuhart (2021). "Urban Data Platform Hamburg: Integration von Echtzeit IoT-Daten mittels SensorThings API." *ZfV-Zeitschrift für Geodäsie, Geoinformation und Landmanagement (zfv 1/2021)*.
- Futurium. (2021). "UMWELTMESSER AUF DEM FAHRRAD." Retrieved 11.05.2023, from <https://futurium.de/en/blog/umweltmesser-auf-dem-fahrrad>.
- Geiger, R. (1942). *Die Besonnung der verschiedenen Hanglagen. Das Klima der bodennahen Luftschicht: ein Lehrbuch der Mikroklimatologie*. R. Geiger. Wiesbaden, Vieweg+Teubner Verlag: 201-213.
- Huang, C. Y. and H. H. Chen (2019). "An Automatic Embedded Device Registration Procedure Based on the OGC SensorThings API." *Sensors* 19(3): 34.
- Initiative, O. S. (2022). "The Open Source Definition." Retrieved 07.05.2023, from <https://opensource.org/osd/>.

Jenkins, K. (2022). "QGIS Python to automatically update layer from data source." Retrieved 07.05.2023, from <https://gist.github.com/kgjenkins/29f8dd4e7da3cba3e0acbd72844748a>.

Jirka, S., C. Autermann, J. Speckamp and M. Rieke (2021). SensorThings API and the OGC API family of standards: a new generation of interoperability standards for research data infrastructures to further improve the sharing of ocean observation data

SensorThings API et la famille de normes OGC API : une nouvelle génération de normes d'interopérabilité pour les infrastructures de données de recherche afin d'améliorer le partage des données d'observation des océans. 9th EuroGOOS International conference, Brest (virtual), France.

Knowledge, F. (2022). "FME and Metadata." Retrieved 07.05.2023, from <https://community.safe.com/s/article/fme-and-metadata>.

Kosmala, M., A. Wiggins, A. Swanson and B. Simmons (2016). "Assessing data quality in citizen science." *Frontiers in Ecology and the Environment* 14(10): 551-560.

Kotsev, A., K. Schleidt, S. Liang, H. Van der Schaaf, T. Khalafbeigi, S. Grellet, M. Lutz, S. Jirka and M. Beaufils (2018). "Extending INSPIRE to the Internet of Things through SensorThings API." *Geosciences* 8(6): 221.

Kox, T., H. W. Rust, B. Wentzel, M. Göber, C. Böttcher, J. Lehmke, E. Freundl, M. Garschagen and T. Kox (2021). "Build and measure: Students report weather impacts and collect weather data using self-built weather stations." *Australasian Journal of Disaster and Trauma Studies* 25(3): 79-86.

Li, W., M. Batty and M. F. Goodchild (2020). "Real-time GIS for smart cities." *International Journal of Geographical Information Science* 34(2): 311-324.

Liu, Y., W. Li and S. Wang (2022). *Real-time GIS Programming and Geocomputation*, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Maisonneuve, N., M. Stevens, M. E. Niessen, P. Hanappe and L. Steels (2009). "Citizen noise pollution monitoring."

Marsh-Hunn, D., S. Trilles, A. Gonzalez-Perez, J. Torres-Sospedra and F. Ramos (2021). "A Comparative Study in the Standardization of IoT Devices Using Geospatial Web Standards." *Ieee Sensors Journal* 21(4): 5512-5528.

Network, T. T. (2022). "LoRaWan Limitations." 07.05.2023, from <https://www.thethingsnetwork.org/docs/lorawan/limitations/>.

Novkovic, I., G. B. Markovic, D. Lukic, S. Dragicevic, M. Milosevic, S. Djurdjic, I. Samardzic, T. Lezaic and M. Tadic (2021). "Gis-based forest fire susceptibility zonation with iot sensor network support, case study—nature park Golija, Serbia." *Sensors (Basel, Switzerland)* 21(ISSN: 1424-8220).

openSenseMap. (2023). "openSenseMap API." Retrieved 25.05.2023, from <https://docs.opensensemap.org/>.

Partescano, E., A. Brosich, M. Lipizer, V. Cardin and A. Giorgetti (2017). "From heterogeneous marine sensors to sensor web: (near) real-time open data access adopting OGC sensor web enablement standards." *Open geospatial data, software and standards* 2(ISSN: 2363-7501): 9.

Ramesh, G., K. Sivaraman, V. Subramani, P. Y. Vignesh and S. V. V. Bhogachari (2021). Farm Animal Location Tracking System Using Arduino and GPS Module. 2021 International Conference on Computer Communication and Informatics (ICCCI).

Santhanavanich, T., R. Padsala, P. Würstle and V. Coors (2022). THE SPATIAL DATA INFRASTRUCTURE OF AN URBAN DIGITAL TWIN IN THE BUILDING ENERGY DOMAIN USING OGC STANDARDS. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*. 10.

Schüppel, K. and V. Wrede (2022). Von der Metropole ins Münsterland: Industriedenkmäler, Emscherpark und Kreidesedimente. Nationaler GeoPark Ruhrgebiet. K. Schüppel and V. Wrede. Berlin, Heidelberg, Springer Berlin Heidelberg: 141-181.

Sejdiu, B., F. Ismaili and L. Ahmedi (2021). "IoTAS: An Integrated System for Real-Time Semantic Annotation and Interpretation of IoT Sensor Stream Data." *Computers* 10(10): 24.

senseBox. (2023). "senseBox." Retrieved 07.05.2023, from <https://sensebox.de/>.

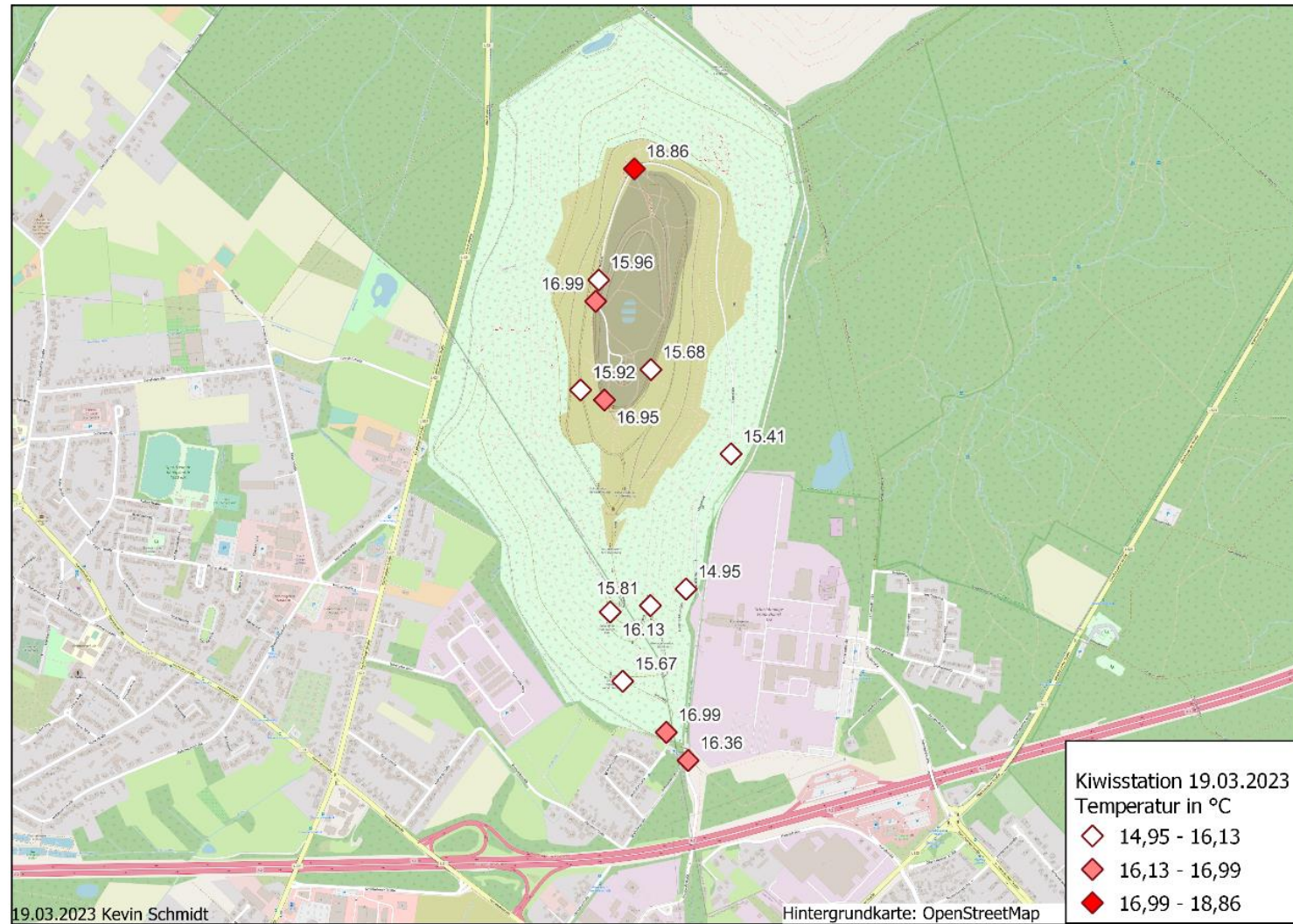
- senseBox. (2023). "senseBox Research." Retrieved 07.05.2023, from <https://sensebox.de/de/research.html>.
- Sensor.Community. (2020). "Airrohr." Retrieved 07.05.2023, from <https://sensor.community/de/sensors/airrohr/>.
- SPOTTERON. (2023). "SPOTTERON Citizen Science." Retrieved 24.06.2023, from <https://www.spotteron.net/de/>.
- techopedia. (2023). "Definition Real Time Data." Retrieved 11.05.2023, from <https://www.techopedia.com/definition/31256/real-time-data>.
- Tönisson, L., J. Voigtländer, M. Weger, D. Assmann, R. Käthner, B. Heinold and A. Macke (2021). "Knowledge Transfer with Citizen Science: Luft-Leipzig Case Study." *Sustainability* 13(14): 7855.
- u-blox. (2023). "NEO-7 series." Retrieved 23.05.2023, from <https://www.u-blox.com/en/product/neo-7-series>.
- Usali, N. and M. H. Ismail (2010). "Use of remote sensing and GIS in monitoring water quality." *Journal of sustainable development* 3(3): 228.
- Verma, S., Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama and N. Kato (2017). "A Survey on Network Methodologies for Real-Time Analytics of Massive IoT Data and Open Research Issues." *IEEE Communications Surveys & Tutorials* 19(3): 1457-1477.
- Wikipedia (2023). Echtzeit.
- Wilkinson, M. D., M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons (2016). "The FAIR Guiding Principles for scientific data management and stewardship." *Scientific Data* 3(1): 160018.

Anhang

<https://opensensemap.org/explore/632378f5d0e7a8001b19f943>

Anhang 01: Link zur Messstation dieser Master – Thesis auf openSenseMap (Stand 2023)

Beispielkarte Weiterverarbeitung Daten aus Echtzeiterfassung



Anhang 02: Beispielkarte für eine Weiterverarbeitung in QGIS mit Daten (statisch)

