



Sieker

Die Regenwasserexperten
The Stormwater Experts



WaterGridSense 4.0 Endmeeting 08.12.2021

Entwicklung der Betriebs- und Wartungsplattform Frauke Jakobs, Dr. Harald Sommer



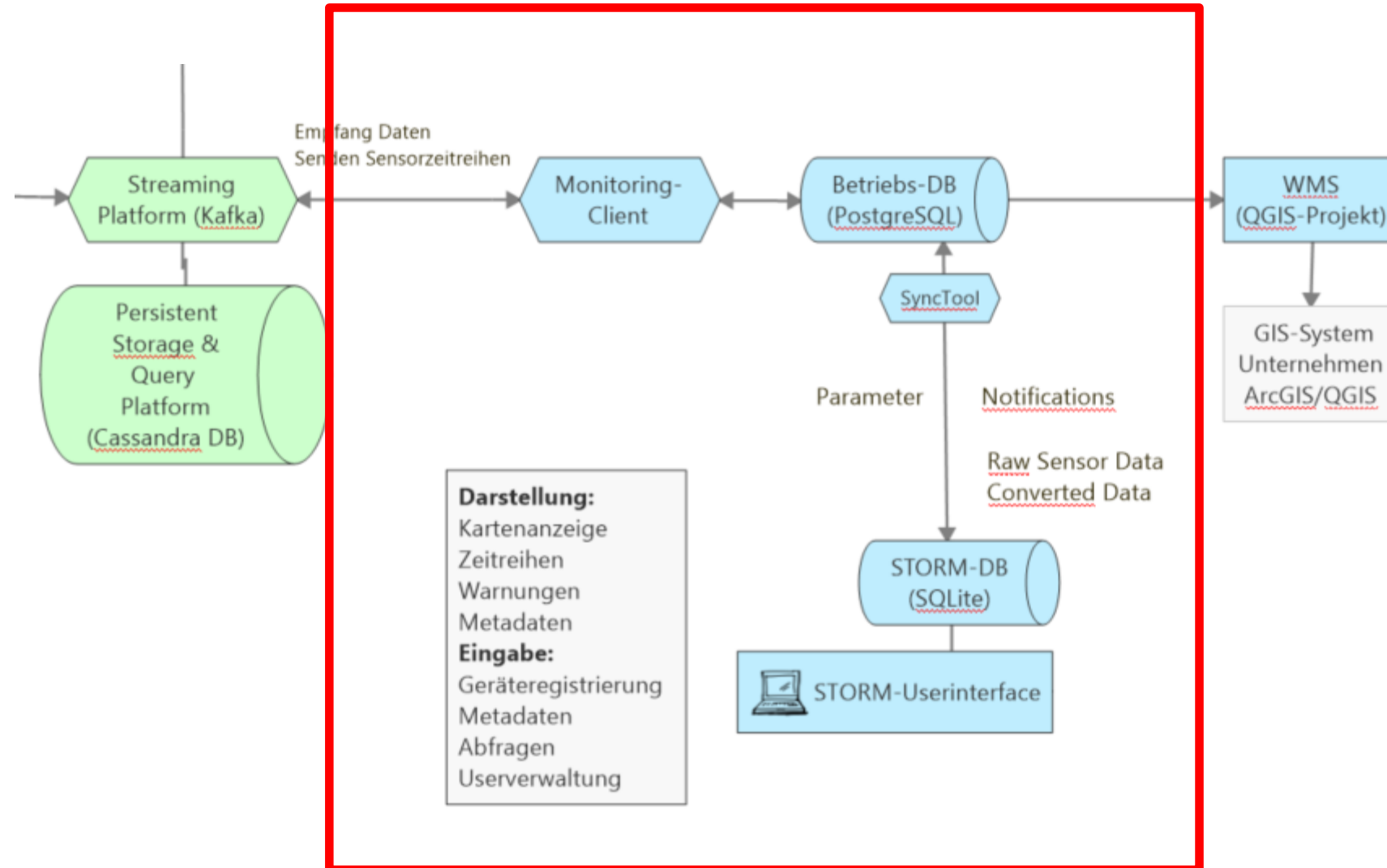
BMBF-Fördermaßnahme
Industrie 4.0 - Kollaborationen in
dynamischen
Wertschöpfungsnetzwerken
(InKoWe)

– Cluster Wasser



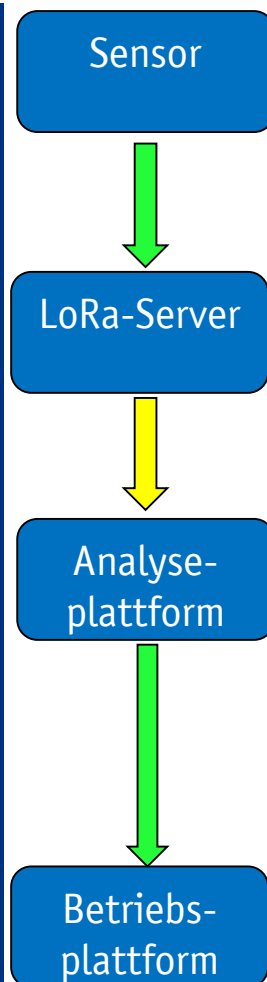


Betriebs- und Wartungsplattform





Schnittstellen-Übersicht



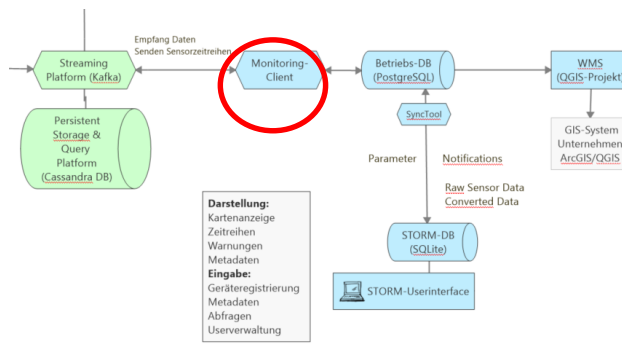
Schnittstelle	Protokoll	Format	Konvertierung
Stationärer Sensor			
Stationärer Sensor → LoRa-TCIP-Gateway	LoRaWAN	CayenneLPP, bzw. hersteller-spezifische Bitstrings	
LoRa-TCIP-Gateway → LoRa-Server	Schnittstelle wird von externem Betreiber bereitgestellt, nur Weiterleitung		
LoRa-Server			Decodieren bitstring → sensor- und netzwerkspezifischer JSON-String
LoRa-Sever → Analyseplattform	MQTT	JSON, Struktur abhängig vom LoRa-Betreiber (TTN, SCS, Zenner IoT)	LoRa-Betreiber TTN, SCS -> Daten werden gepusht Zenner IoT -> Daten werden gepullt
Analyseplattform			Parsen JSON → SenML
Analyseplattform → Monitoring-Client	TCP/IP	SenML	
Analyseplattform ← Monitoring-Client	TCP/IP	SenML	
Monitoring-Client			Komprimierung SenML → Binärobjekt für Jahresabschnitt
PostGres/Sqlite-DB		BLOB	
Userinterface			Dekomprimierung für Ausgabe als Tabelle und Graph



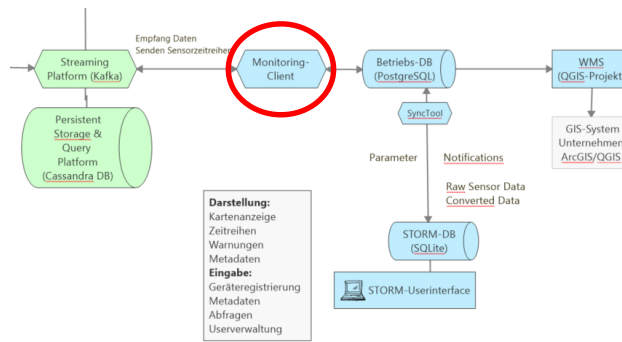
Arbeiten



- Entwicklung eines Tools zur Zeitreihen-Komprimierung „**TS-Compress**“
- Entwicklung des **Monitoring-Client**
 - Anbindung an Kafka
 - Import in die Postgres-Datenbank
 - Synchronisation der Datenbanken
- Entwicklung des **Sync-Tools**
 - Synchronisations lokales Projekt mit Datenbank
- Entwicklung **Benutzeroberflächendarstellung**
 - GIS- und Zeitreihendarstellung
 - Implementierung der Sensorregistrierung
- Durchführung von Sensortests: Übertragungstest, Kanaltest



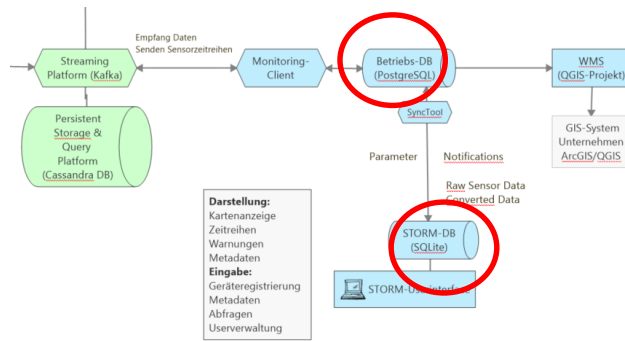
- Kommunikationsschnittstelle zwischen Kafka und Betriebs- und Wartungsplattform
- Vielfältige Aufgaben
 - Abonnieren von Datenströmen aus Kafka
 - Import der Daten über die TS-Compress-Schnittstelle in die Postgres-Datenbank
 - Anhängen von Daten
 - Aktualisierung von Daten
 - Aktivitätsgetriggertes Senden von Daten an Kafka
 - Registrieren von Geräten und Sensoren
 - Senden von Kennlinien der Zeitreihen



Entwicklung Monitoring Client



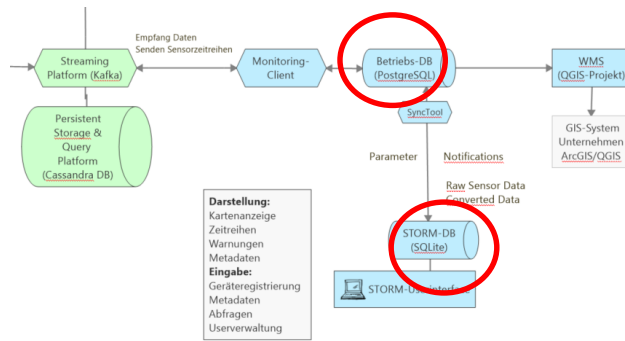
- Registrierung Geräte und Sensoren in Cassandra DB
 - Publizieren von Geräte- und Sensordaten an Kafka/Cassandra
 - inkl. Verortung und Transformationskennlinien
 - Aktionsgesteuerte Eintragung von Werten in die Cassandra über den Kafka-Broker
- Empfang und Verarbeitung Zeitreihen
 - Empfang Zeitreihen als Kafka Abonnement
 - Weiterleitung und Eintragung in die PostGRES DB im BLOB Format
 - Abfrage aus der PostGRES DB im BLOB Format
 - Publizieren von Zeitreihen an Kafka/Cassandra
- Zentrale Instanz für die Kommunikation mit der Middleware
 - Möglichkeit zur Kommunikation mit anderen Datenbanken



Entwicklung eines Tools zur Zeitreihen-Komprimierung



- Ausgangssituation:
 - kein standardisiertes Binärformat für Zeitreihen in relationalen Datenbanken
 - Bisher in STORM genutztes Format nur für äquidistante Zeitreihen geeignet
- Recherche von gebräuchlichen Zeitreihenformaten
 - für unterschiedliche Einsatzgebiete (Wetter-/Klimadaten, Hydrologie, IoT)
 - Anwendungsbereiche (Datenübertragung, -speicherung)
- Ermittlung der Anforderungen des Userinterface/Zeitreihenanzeige
 - Verlustfreie Komprimierung
 - Information von Zeitpunkt und Wert für jeden Messwert
- Recherche und Test geeigneter Verfahren zur Zeitreihenkomprimierung
 - Trennung von Zeitstempel und Values
 - Delta encoding, Run-length encoding



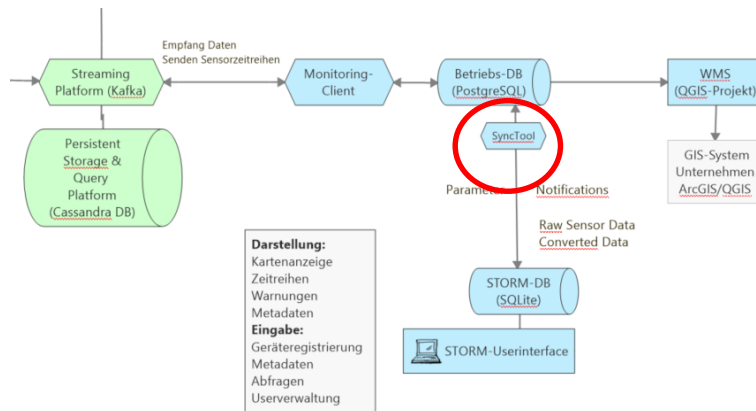
Entwicklung eines Tools zur Zeitreihen-Komprimierung



- Entwicklung von „TSCompress“
 - Codierung der Zeitreihen als Bitblöcke
 - Zusammenfassen zu BLOBS (Binary Large Objects) je Jahr
- Speicherung in Postgres und SQLite
 - Zeitreihen als BLOBS
 - Metadaten als String- und Zahlenformate in der Datenbank
- Routine wird zukünftig auch weiteren Projekten genutzt
 - Automatisierter Import/Export in die Simulationssoftware „InfoWorks ICM“

Bitblock

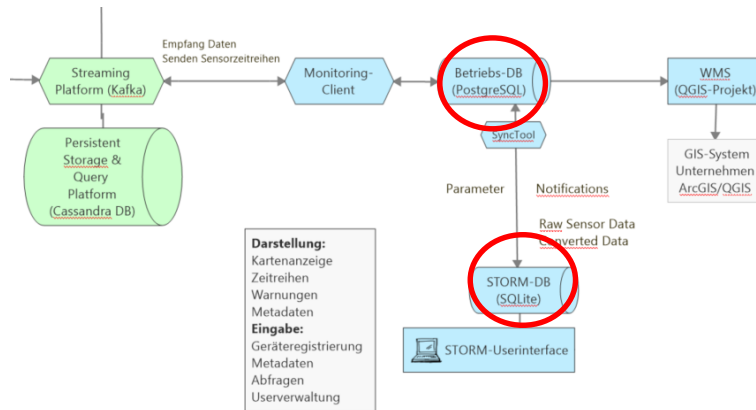
4bits				5bits				1bit	dynamisch (1-31 bits)			
Anzahl von Wiederholung, max. 15				Anzahl der Bits für den zu speichernden Integer-Wert, max. 31				+/-	Wert im Bereich 0 bis 2.147.483.648			



Entwicklung eines Sync-Tools zum Abgleich der Daten PostGres/Sqlite

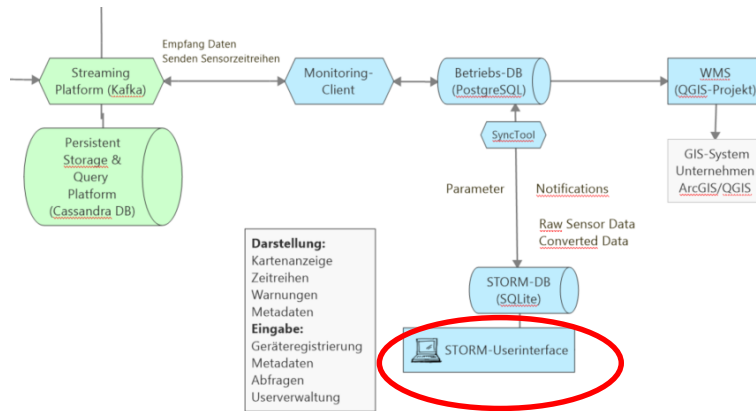


- Aufgabe des Sync-Tools
 - Einloggen auf den Server erforderlich
 - Lokales Erstellen von Projekten in PostGres DB aus einer Vorlagendatei
 - Lokales Registrieren von Geräten und Hochladen auf PostGres DB
 - Für alle User verfügbar
 - Für User können eigene Projekte erstellt werden, die auf die gleichen Daten zugreifen
 - Herunterladen von Zeitreihen von PostGres in Sqlite DB
 - Festlegen von Warnwerten für alle User in einem Projekt
 - Laden der Datei bei Start
 - Abspeichern der Konfiguration beim Abschluss



Userverwaltung PostGres/Sqlite

- Aufgabe des User-Verwaltung
 - Ziel: Multi User – Multi Projekt
 - Lokales Erstellen von Usern, Hochladen auf PostGres DB
 - BeZugriff
 - Herunterladen von Projekten, die dem User zugeordnet sind von PostGres DB in Sqlite DB
 - Erstellen von Projekten auf der PostGres DB, die von mehreren Usern verwendet werden können.



Entwicklung der GIS-UI zur Verwaltung von Geräten und Sensoren



- Aufgabe der GIS-UI
 - Anlegen und Verortung von Geräten
 - Zuordnung von Sensoren zu den Geräten
 - Herunterladung Visualisierung der Zeitreihen
 - Darstellung der Zeitreihen unterschiedlicher Messgrößen
 - Visualisierung der Zustände im GIS-UI



Workflow Arbeit mit Projekten

- Registrierung User
 - **User auf Projekt** angelegt
 - **Mit PostGres DB synchronisiert**
 - **Über Kafka an Cassandra DB**, inkl. Verortung und Transformationskennlinien
- Anlegen Projekt
 - Anlage **Projekt in SQLite (lokal)**, **Übertragen an PostGres DB**
 - **Zuordnung User** zum Projekt
 - **Synchronisieren Zeitreihen als BLOBS** in Projekt sqlite lokal
 - **Metadaten** als String- und Zahlenformate in der Datenbank



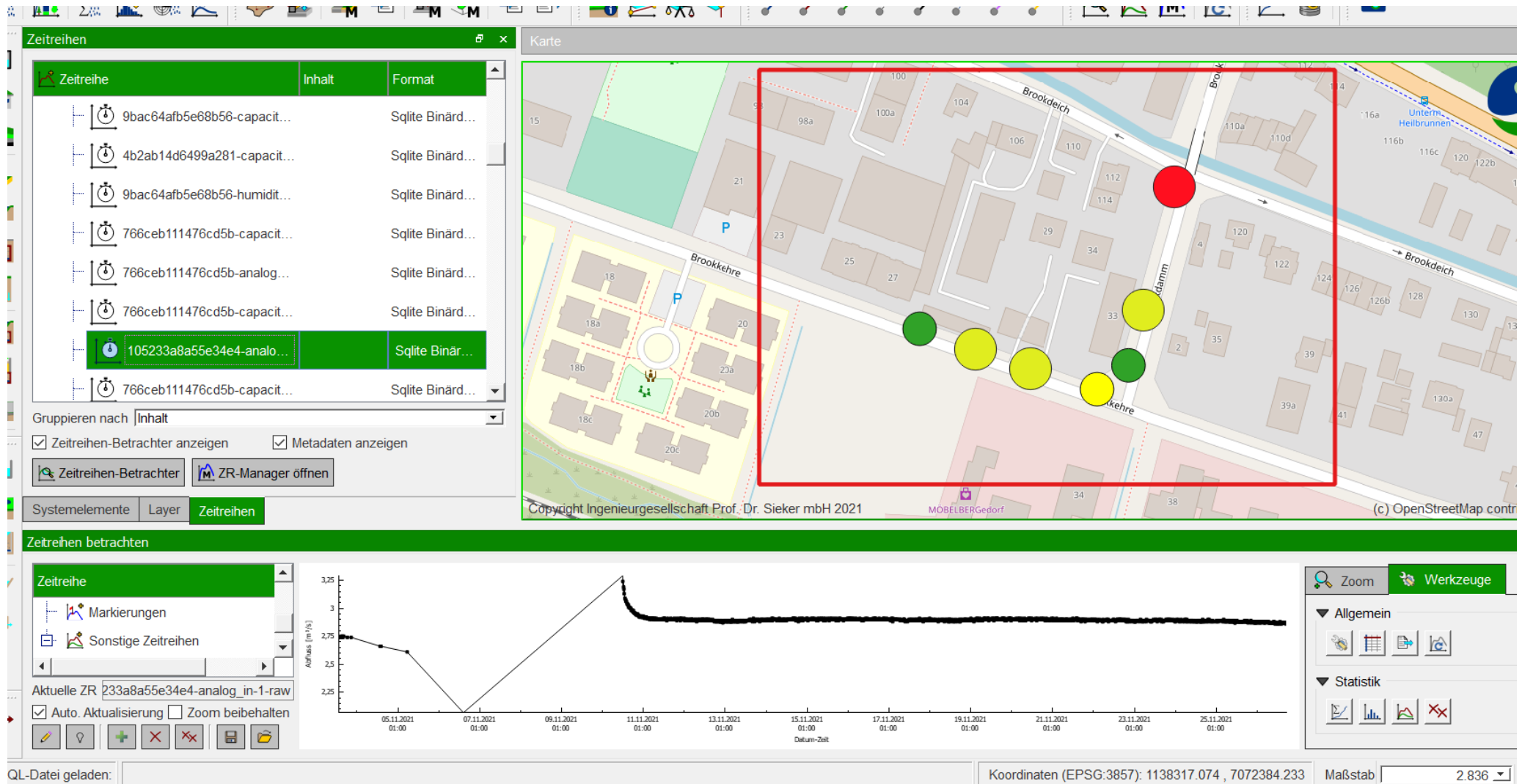
Workflow Registrierung Geräte und Sensoren



- Registrierung Geräte und Sensoren
 - **Lokal in der GIS-GUI** angelegt (sqlite DB)
 - Festlegen von **Gerätegruppen** und **Warnwerten**
 - Möglichkeit der Erstellung von Daten-Transformationskennlinien
 - Mit **PostGRES DB synchronisiert**
 - Über Monitoring-Client Kafka an Cassandra DB, inkl. Verortung und Transformationskennlinien
- Empfang und Verarbeitung Zeitreihen
 - **Empfang Zeitreihen** als Kafka Abonnement
 - **Empfang Zustandsauswertung** auf Analyseplattform
 - **Eintragung in die PostGRES DB im BLOB Format**
 - **Synchronisieren Zeitreihen als BLOBS** in Projekt sqlite lokal
 - Metadaten als String- und Zahlenformate in der Datenbank



Entwicklung Zeitreihenanzeige in der Benutzeroberfläche





Livedemo





Vielen Dank für Ihre Aufmerksamkeit!



Kontakt

- Frauke Jakobs, Dr. Harald Sommer
- f.jakobs@sieker.de, h.sommer@sieker.de

