

Arbeitspaket 9 **Validierung von Plattform / Diensten**

Entscheidungshilfe für Routenwahl durch Einsatzkräfte
im Starkregenfall per Echtzeit Verkehrsdaten

Dr.-Ing. Stefan Radomski

Verbundpartner



Gefördert durch:



Bundesministerium
für Verkehr und
digitale Infrastruktur

Agenda

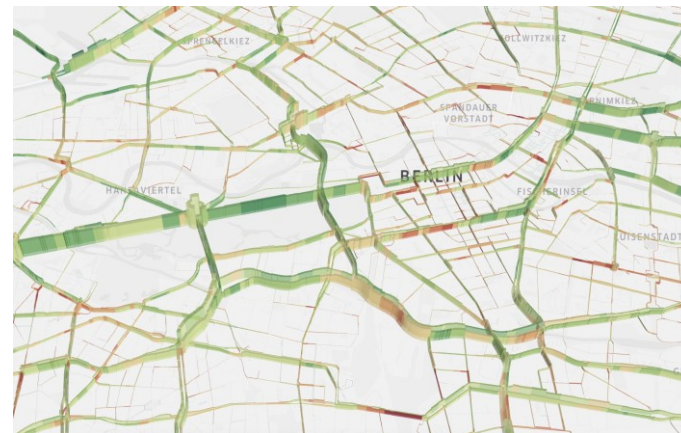
1. Zielsetzung
2. Starkregen in Echtzeit-Verkehrsdaten
 - a) Flutkatastrophe in Mitteleuropa 12-19.07.2021
 - b) Starkregen in Berlin am 02.08.2019
 - c) Starkregen in Berlin am 25.07.2021
3. Umsetzung in SENSARE

Zielsetzung

Validierung von Plattform / Dienste

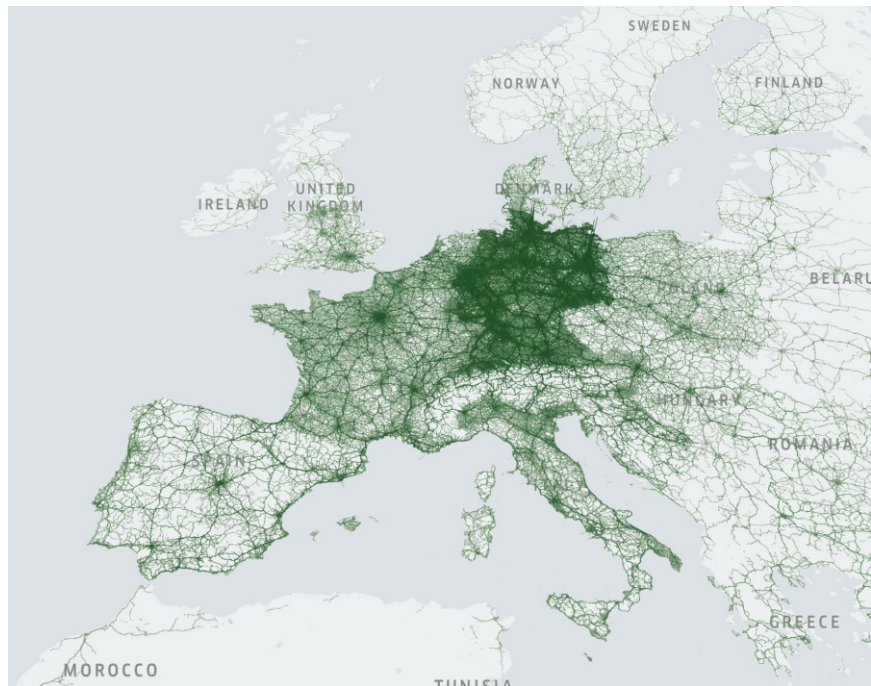
- Aufgaben per Teilvorhabensbeschreibung
 - Routing für Einsatzfahrzeuge
 - Dynamische Verkehrs- & Routenmodelle

- Realisiert im Rahmen des Projektes
 - Verkehrsaufkommens und Geschwindigkeiten in Berlin
 - Erhebung und Visualisierung
 - Mit minuten-aktuellen Floating-Car Daten
 - Einbettung per OGC WMS Protokoll in die SENSARE Plattform



Echtzeit-Verkehrsdaten @ [ui!]

Floating-Car Data (FCD)

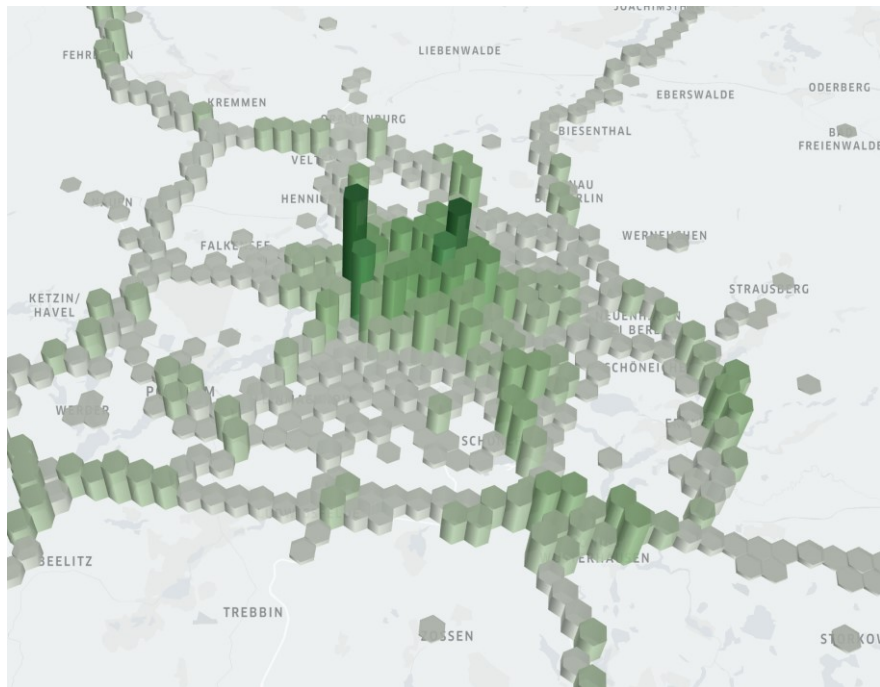


Kennzahlen Gesamt

- 10 Mrd. Datenpunkte / Monat
 - 3-15s Frequenz / 60-90s Latenz
- GPS Daten in hoher Genauigkeit
 - Keine triangulierten Mobilfunkdaten
 - Flotten OBUs / Personal Navigation Devices
- 2-3% Abdeckung in Deutschland
 - Evaluiert mit BAST Daten aus NRW
- > 20TB Historische Daten seit 2018

Echtzeit-Verkehrsdaten @ [ui!]

Floating-Car Data (FCD)



Kennzahlen Berlin

- > 2 Millionen Datenpunkte / Stunde
 - Im Feierabendverkehr
 - Höchste Datendichte im Korpus
- ~ 15.000 Verkehrsteilnehmer / Stunde

A decorative graphic consisting of three concentric, curved lines on the left side of the slide.

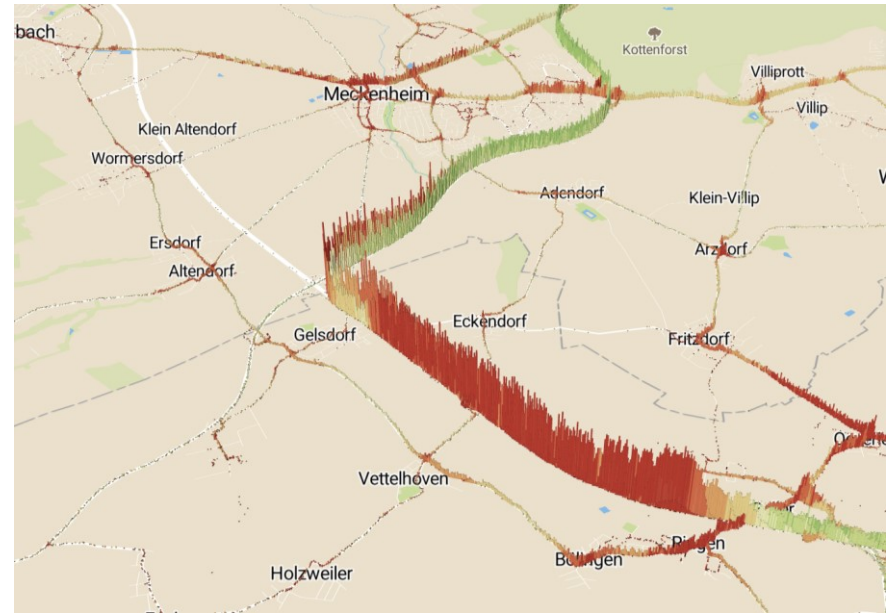
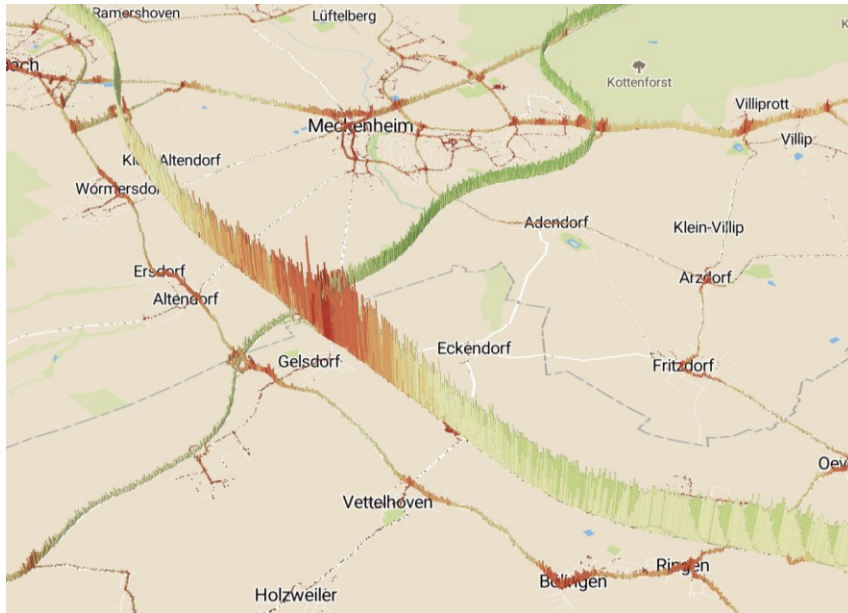
Entscheidungshilfe für Routenwahl für Einsatzkräfte im Starkregenfall

Wie stellen sich Starkregenereignisse in den Floating Car Daten dar?

- Flutkatastrophe in Mitteleuropa 12-19.07.2021
- Starkregen in Berlin am 02.08.2019
- Starkregen in Berlin am 25.07.2021

Echtzeit-Verkehrsdaten @ [ui!]

Flutkatastrophe in Mitteleuropa 12-19.07.2021

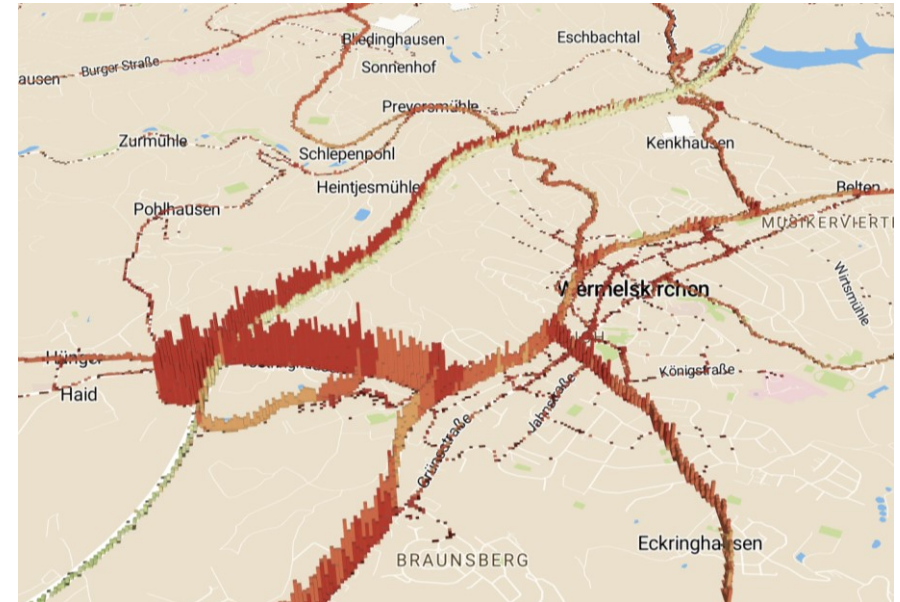
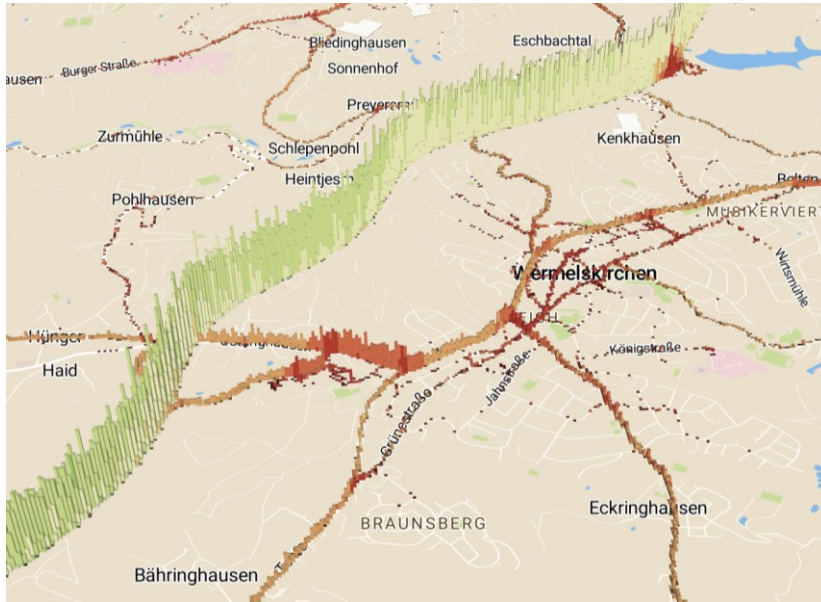


A61 bei Meckenheim (09.07.2021 / 16.07.2021)

<https://analytics.urbanpulse.de/kepler/map?mapUrl=https://dl.dropboxusercontent.com/s/e0ex5ttf00l4q6/20210719-Flooding%20A61%20Kreuz%20Meckenheim%20und%20Rheinbach.json>

Echtzeit-Verkehrsdaten @ [ui!]

Flutkatastrophe in Mitteleuropa 12-19.07.2021

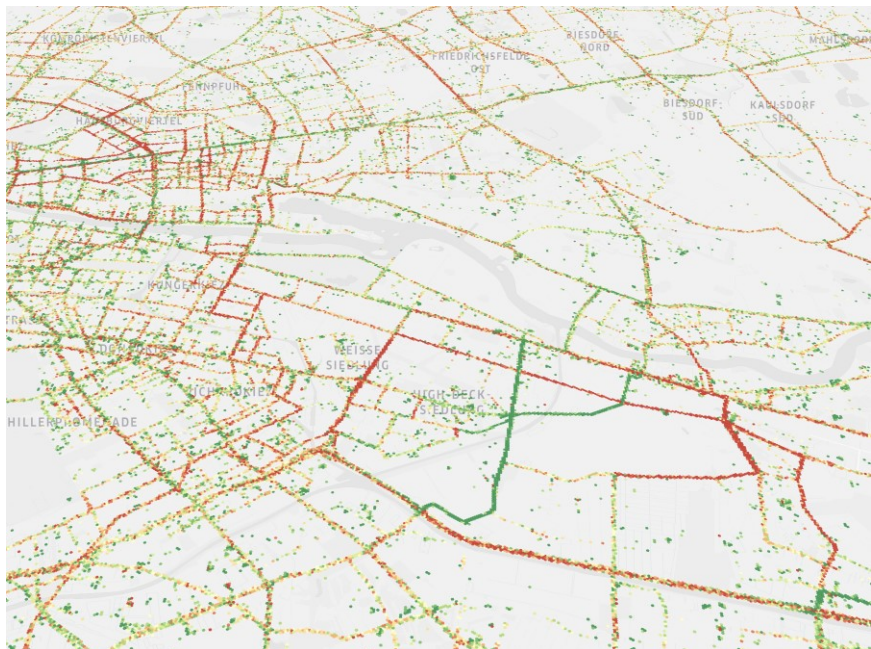


A1 zwischen Wuppertal und Leverkusen (09.07.2021 / 16.07.2021)

<https://analytics.urbanpulse.de/kepler/map?mapUrl=https://dl.dropboxusercontent.com/s/p59j2mum1rbheuu/20210721-Flooding%20A1%20Leverkusen%20und%20Burscheid.json>

Echtzeit-Verkehrsdaten @ [ui!]

Starkregen in Berlin am 02.08.2019



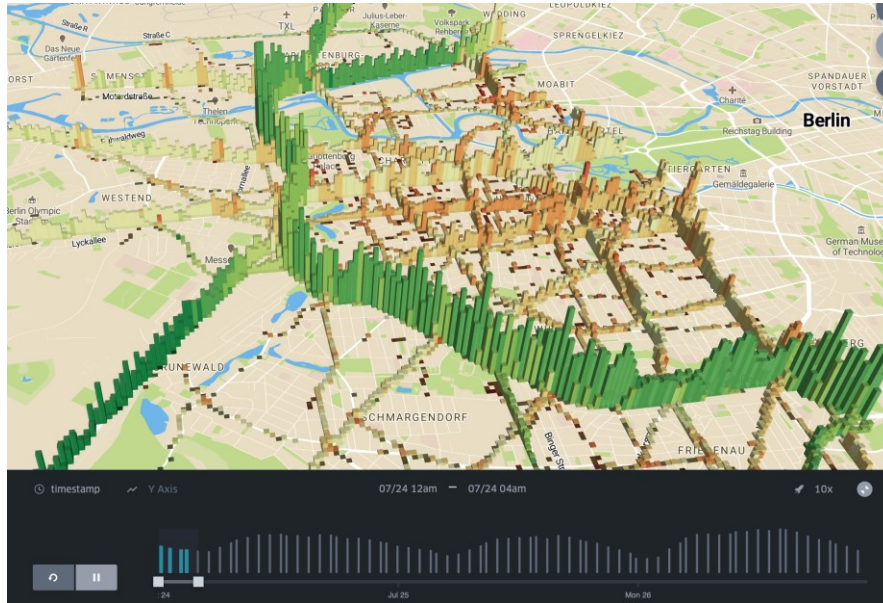
Differenzen Verkehrsaufkommen

- Daten vom Fr. 02.08.2019 und vom 09.08.2019
 - Tageswerte in ca. 10x10m Boxen aggregiert
 - Differenz der beiden Tage farbkodiert
- Grüne Bereiche:
 - Verstärkter Verkehr an normalen Tagen
- Rote Bereiche:
 - Verdrängungsverkehre bei Hochwasser

<https://analytics.urbanpulse.de/kepler/map?mapUrl=https://dl.dropboxusercontent.com/s/glf9a5z32d8foz1/Berlin%20Hochwasser%20Differenz.json>

Echtzeit-Verkehrsdaten @ [ui!]

Starkregen in Berlin am 25.07.2021



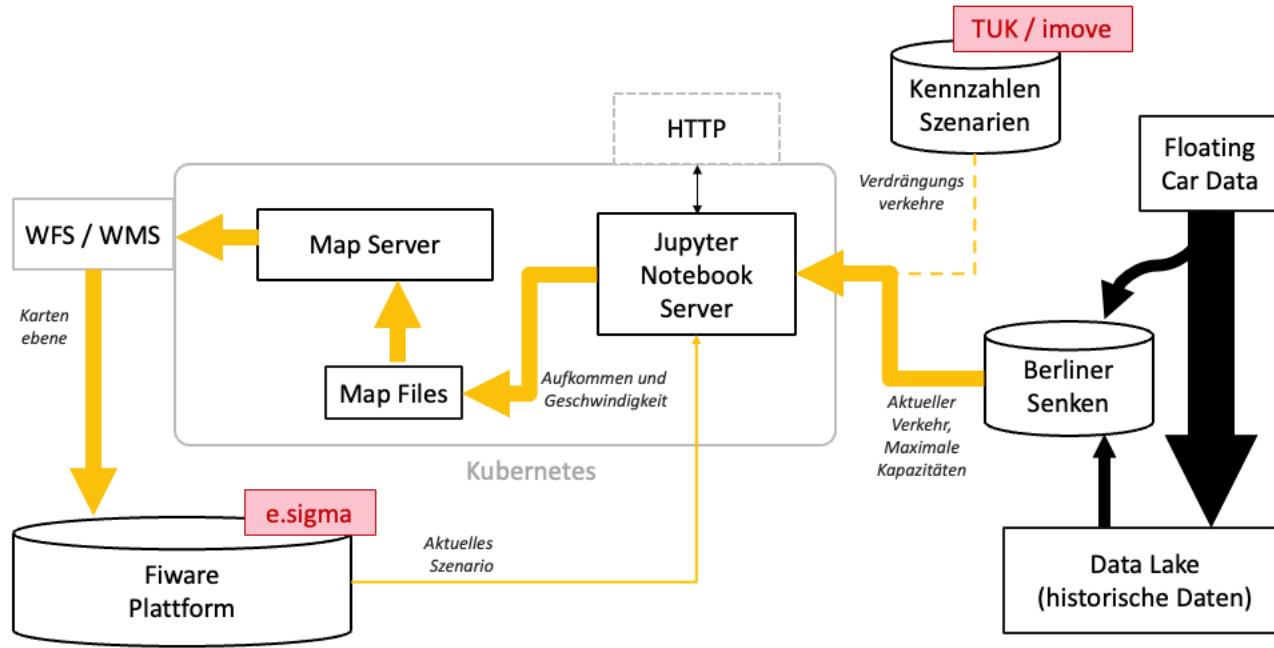
Verkehrsaufkommen und gefahrene Geschwindigkeiten

- Zeitserienanimation
 - Verkehrsaufkommen als Höhe
 - P85 der Geschwindigkeiten als Farbe
- Hohe rote Bereiche:
 - Viele Verkehrsteilnehmer fahren langsam
- Verkehrsanomalie 10-14Uhr (3.11s):
 - A100 zwischen Kaier- und Spandauerdamm

<https://analytics.urbanpulse.de/kepler/map?mapUrl=https://dl.dropboxusercontent.com/s/swu3t6t4fmbtav4/20210803-Berlin-Starkregen.json>

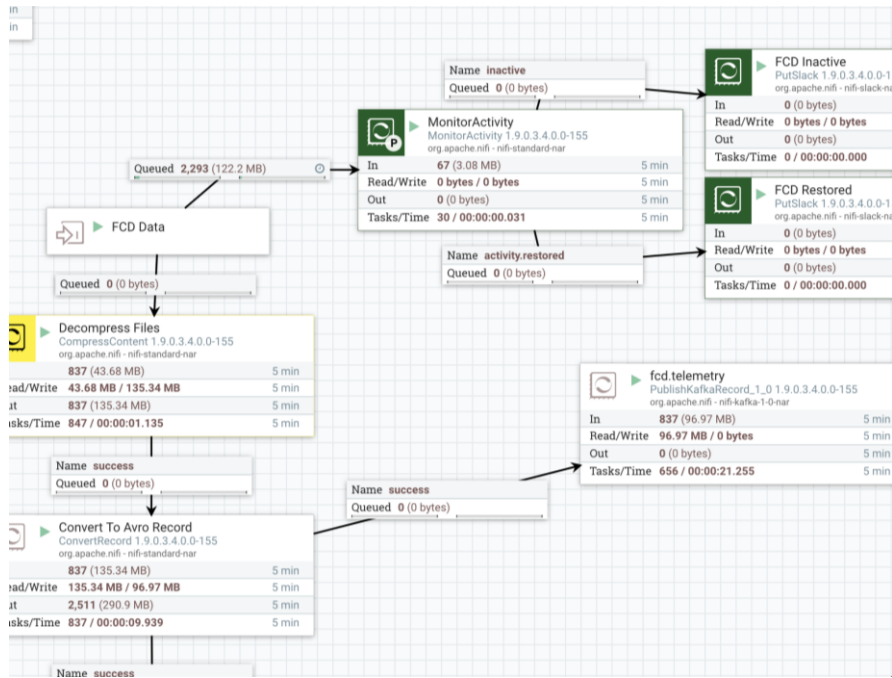
Umsetzung in SENSARE

Technische Plattform und Integration



Umsetzung in SENSARE

Vorverarbeitung / Auffächerung mit Apache Nifi



- Empfang der Floating Car Daten
 - ~10000 Datensätze / Sekunde in der Spitze
- Vorverarbeitung
 - Entpacken der CSV Batches
 - Transformation in typisierte Binärdaten
- Belieferung angrenzender Systeme
 - Apache Kafka Echtzeitbus
 - Gespeichert als spaltenbasierte Dateien im verteilten Dateisystem für historische Analysen

Umsetzung in SENSARE

Verarbeitung in Jupyter Notebooks

568934	534468ad2a962c1a@@@24abc06c-25bf-4fb6-8c30-70e...	2021-11-25T14:26:06+0000	52.480480	13.444864	7.0	125.0	1.637850e+09	5
225872	c00ae46471360d96@@@70b42a95-9793-491b-8e74-017...	2021-11-25T14:26:05+0000	52.533627	13.484335	49.0	264.0	1.637850e+09	5
144050	1218b4477430ba1d@@@f98ce7f0-dce3-46b2-953c-a0e...	2021-11-25T14:26:01+0000	52.445953	13.207435	86.0	38.0	1.637850e+09	5
225871	c00ae46471360d96@@@70b42a95-9793-491b-8e74-017...	2021-11-25T14:26:00+0000	52.533680	13.485344	48.0	264.0	1.637850e+09	5

1125074 rows x 8 columns

Run through Map-Matcher in Batch Mode

```
import socket
import json

def input_for_map_matcher(df):
    mm_input = [dict(id=row.id,
                    point=f"POINT({row.lon} {row.lat})",
                    azimuth=row.heading,
                    speed=row.speed, time=1000 * int(pd.Timestamp(row.timestamp).timestamp())
                    for index, row in df.iterrows())
    ]
    return mm_input

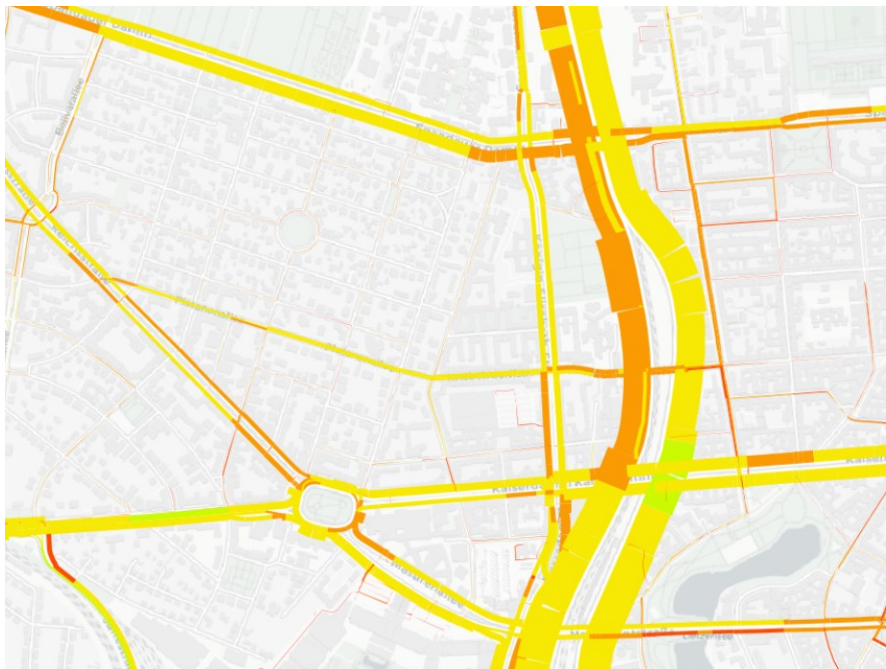
def run_through_map_matcher(input, tracking=False):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    if tracking:
        s.connect(("17a68b5.online-server.cloud", 31072))
        s.sendall(bytes(json.dumps(input) + "\n", encoding="utf8"))
    else:
        config_dict = ""
        s.connect(("17a68b5.online-server.cloud", 31070))
        s.sendall(bytes("[ " + config_dict + json.dumps(input)[1:] + "\n", encoding="utf8"))

    mm_reply = ""
    while True:
```

- Empfang der Daten vom Echtzeit-Bus
 - Gleitendes 30min Fenster von Berlin
- Abbildung auf Strassennetzwerk
 - Strassenmodell aus OpenStreetMap
- Aggregation auf Strassensegmente
 - 85er Perzentil der Geschwindigkeit
 - Anzahl Verkehrsteilnehmer
- Erstellen des GIS Layers
 - Geopackage im OGC GeoServer

Umsetzung in SENSARE

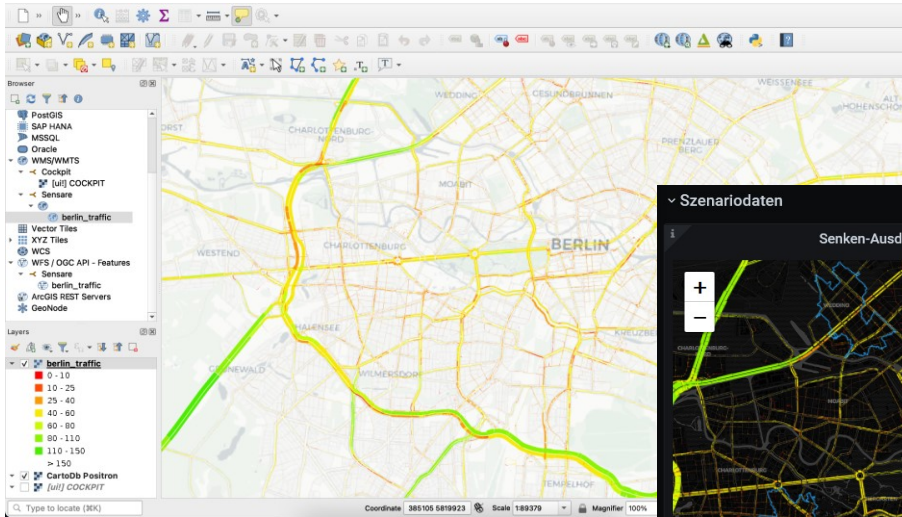
Abbildung auf das Strassennetzwerk



- BMW CarIT Barefoot Map-Matcher
 - "Bester Erklärungswert" der Telemetrie als Trajektorie im OSM Strassenmodell (HMMs)
 - Erlaubt Fahrtrichtungsdifferenzierung
- **Ändert die Interpretation!**
 - Zuvor: Wo bewegen sich wieviele Verkehrsteilnehmer mit welcher Geschwindigkeit?
 - Danach: Wie schnell haben wieviele Verkehrsteilnehmer ein Strassensegment durchfahren?

Umsetzung in SENSARE

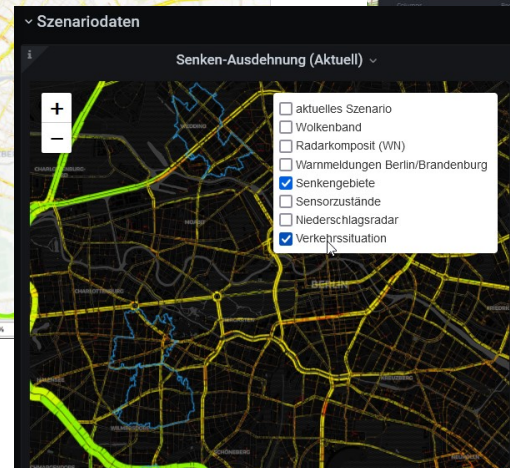
Bereitstellung als Kartenlayer



Oben: Für Kartenebene im QGIS GIS Client



Oben: Für GeoJSON in Kepler.gl



Links: Für Kartenebene im SENSARE Dashboard von e.sigma

Verwertung

Geschäftsmodelle

- B2G:
 - Darstellung der dynamischen Verkehrssituation für Bürgerportale
 - Überwachung der Verkehrssituation durch die Mobilitätsämter
 - Zur Planung / Überwachung von grösseren Baumassnahmen
 - Zur Identifikation von Verkehrsanomalien
- B2B:
 - Lieferverkehre vermeiden verstopfte Strassen
 - ...



[ui!] Urban Institute
Dr.-Ing. Stefan Radomski
Tel.: +49 (0) 6151 49320 65
stefan.radomski@the-urban-institute

