



RADWEG RADAR

Abschlusspräsentation
Projektmanagement

PG INFRASeNSE

Michael Birke, Florian Dyck, Mukhran Kamashidze, Malte Kuhlmann, Malte Schott, Richard Schulte, Alexander Tesch

25.09.2023

INFRASense

Projektziel



- Unterstützung der Radverkehrsplanung durch Crowdsourcing und automatisierte Sensordatenauswertung



INFRASense

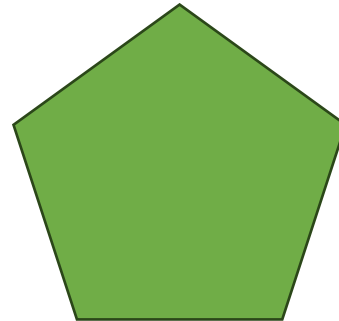
Projekt-Konsortium



[1]



[1]



[1]



[1]



[1]

PG INFRASeNSE

Projektziele

- Entwicklung eines interaktiven Dashboards und Analysemethoden zur Unterstützung der Radverkehrsplanung
- Recherche und Implementierung geeigneter KI-Methoden zur Auswertung von Bilddaten zur Fahrradinfrastruktur



PG INFRA Sense

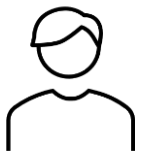
Arbeitsaufteilung



Dashboard



Birke, Michael



Kamashidze, Mukhran



Tesch, Alexander



Data



Dyck, Florian



Schulte, Richard



AI



Kuhlmann, Malte



Schott, Malte



PG INFRASeNSE

Projektmanagement

- 2x wöchentliche fachübergreifende Meetings (intern/extern)
- Regelmäßiger Projektaustausch mit Worldiety & VIA
- Technologien: Jira / Confluence / Discord / MS Teams
- Tickets nach Anforderungen definiert und abgearbeitet
- Agile Entwicklung
- Publikationen, Bürgerveranstaltung und Vorträge
 - EnviroInfo
 - CRBAM

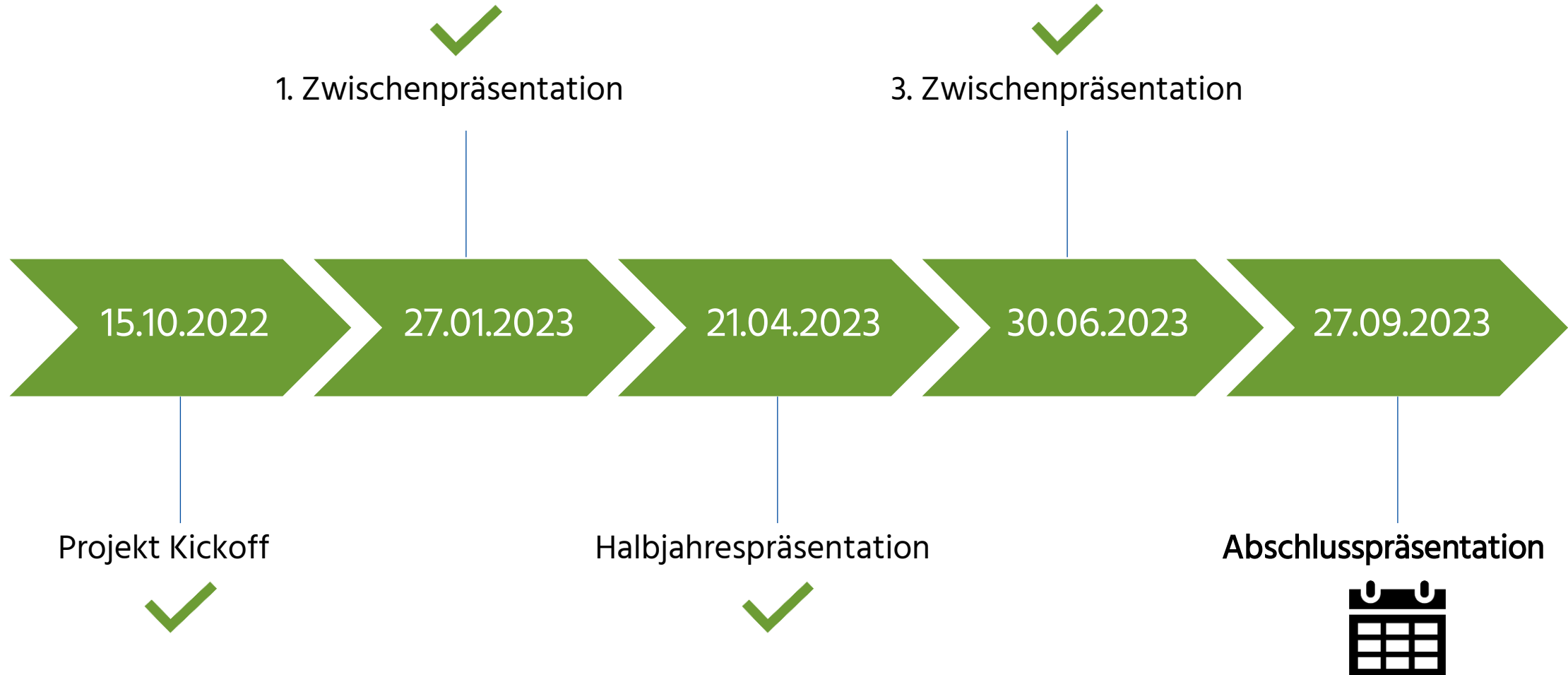


The Bike Path Radar: A Dashboard to provide new Information about Bicycle Infrastructure Quality

Michael Birke¹, Florian Dyck², Mukhran Kamashidze³, Malte, Kuhlmann⁴, Malte Schott⁵, Richard Schulte⁶, Alexander Tesch⁷, Johannes Schering⁸, Pascal Säfken⁹, Jorge Marx Gómez¹⁰, Kathrin Krienke¹¹, Peter Gwiasda¹²

Abstract: A growing interest in cycling can be perceived. The existing bike infrastructure is not sufficient anymore. Improvements are urgently needed. Data can support the decision making process in bicycle infrastructure planning. Especially dashboards can make a positive contribution to learn more about infrastructure shortcomings if these provide relevant Key Performance Indicators (KPIs) and visualizations. The Bike Path Radar (Radweg Radar) should fill this research gap by providing relevant information about cycling infrastructure. The frontend enables the end user to create different KPIs regarding cycling accidents, citizen reportings, traffic volume etc. of highest interest. A role concept enables the provision of a suitable degree of information traffic planning experts and citizens. The most important KPIs were identified based on expert interviews. The dashboard is connected to a database in the background that includes heterogeneous cycling and bicycle infrastructure data by an API. In addition to that, the dashboard gives new opportunities for citizen engagement. Users can upload images of bike path damages. The images will be processed by an object detection algorithm to find the damage. The damages will be displayed on a map by a marker to find locations with surface shortcomings. This contribution will give a short overview about the current state of development of the Bike Path Radar. The outlook provides some additional information about the forthcoming working steps.

Projektverlauf





RADWEG RADAR

Meldeportal für Radwegschäden

PG INFRASense

Malte Kuhlmann, Malte Schott

25.09.2023

Agenda

1

Ziele / Usecase

2

Anforderungen

3

Realisierung - Backend

4

Evaluation

5

Live Demo

6

Limitationen

7

Ausblick / Lessons Learned

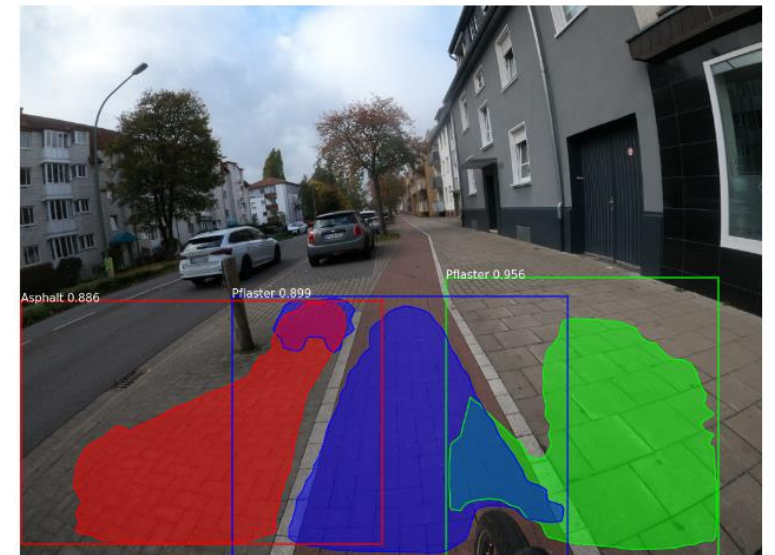
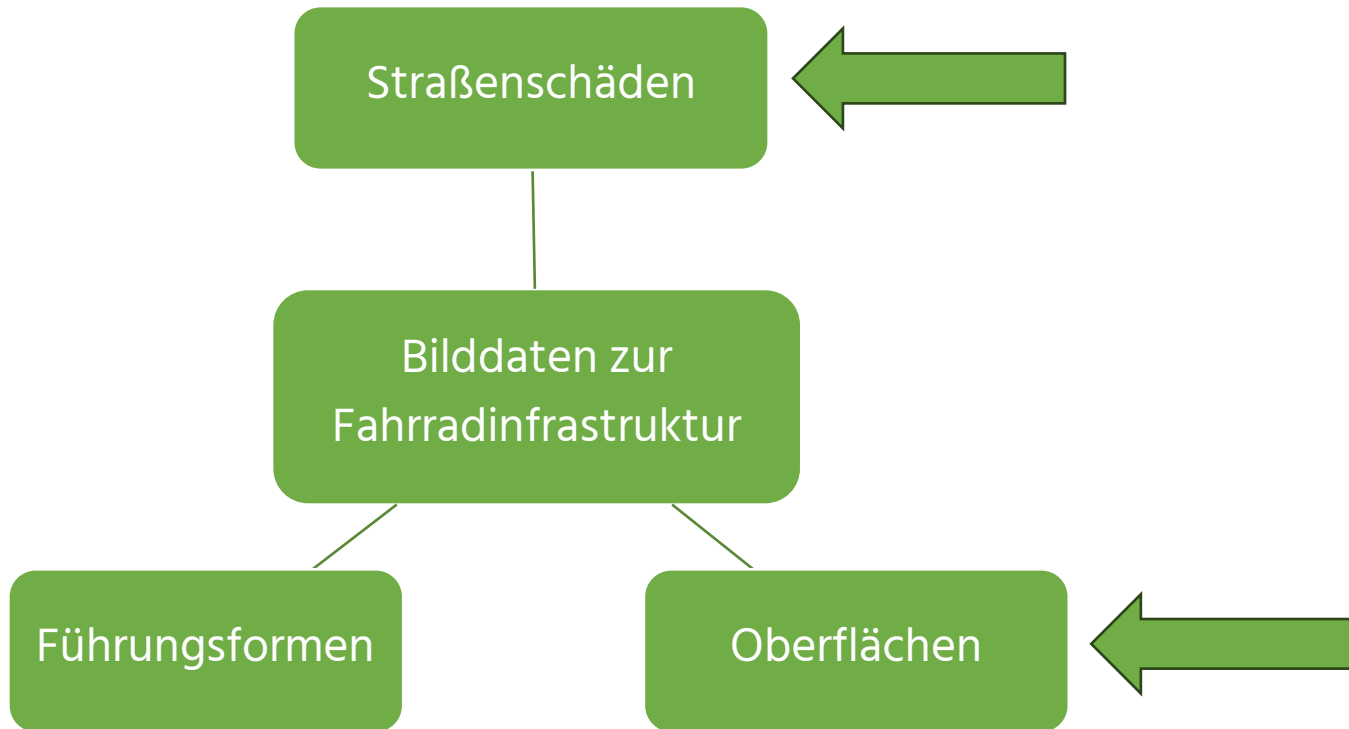
8

Diskussion / Fragerunde



Ziele / Usecase

Recherche und Implementierung geeigneter KI-Methoden zur Auswertung von Bilddaten zur Fahrradinfrastruktur



Usecase / Ziel KI Gruppe

- Fragestellung: Wie kann das entwickelte Modell am Ende eingesetzt werden?
- Brainstorming von diversen Ideen
- Idee: Meldeportal für Radwegschäden, die mit einem KI-Modell klassifiziert werden, für Bürgerinnen und Bürger
 - Crowdsourcing liefert Daten / Bilder
 - Klares Ziel bzw. Ergebnis



Referenzprojekte

Oldenburg Stadtverbesserer



[2]

EMSOS - EreignisMeldeSystemOSnabrück



[3]



[4]

ABPA



[5]



automatisierte Zustandserfassung
der Radwegeinfrastruktur

[6]

Funktionale Anforderungen

Nr.	Usecase	Art	Priorität	Beschreibung
FA1	Schaden melden	Frontend	Must	Ein User soll einen Schaden per Bild melden können. Das Bild soll abgespeichert werden.
FA2	Schaden anzeigen	Frontend	Must	Ein User soll sich die bisher gemeldeten Schäden anzeigen lassen können.
FA3	Schaden klassifizieren	Backend	Must	Das KI-Modell soll einen Schaden erkennen und klassifizieren können.
FA4	Standort bestimmen	Frontend	Must	Ein User soll je nach Bild den Standort manuell auf der Karte setzen können oder automatisch ausgelesen werden.



Nicht-funktionale Anforderungen

ISO 9126

Nr.	Usecase	Beschreibung
NFA1	Wartbarkeit	Die Software sollte einfach aktualisiert und gewartet werden können, um langfristige Änderungen und Fehlerbehebungen zu ermöglichen.
NFA2	Benutzbarkeit	Die Software sollte benutzerfreundlich sein und eine leicht verständliche Benutzeroberfläche bieten, um die Nutzerzufriedenheit zu gewährleisten.
NFA3	Effizienz	Die Software sollte Ressourcen wie CPU und Speicher effizient nutzen und schnell auf Benutzeranfragen reagieren, um die Leistung zu optimieren.
NFA4	Funktionalität	Die Software sollte alle erforderlichen Funktionen und Features bieten, um die beabsichtigten Aufgaben zu erfüllen.
NFA5	Übertragbarkeit	Die Software sollte auf verschiedenen Plattformen und Systemen leicht implementiert und ausgeführt werden können.
NFA6	Zuverlässigkeit	Die Software sollte stabil und fehlerfrei funktionieren, um unerwartete Ausfälle oder Datenverluste zu vermeiden.



Realisierung

Architektur

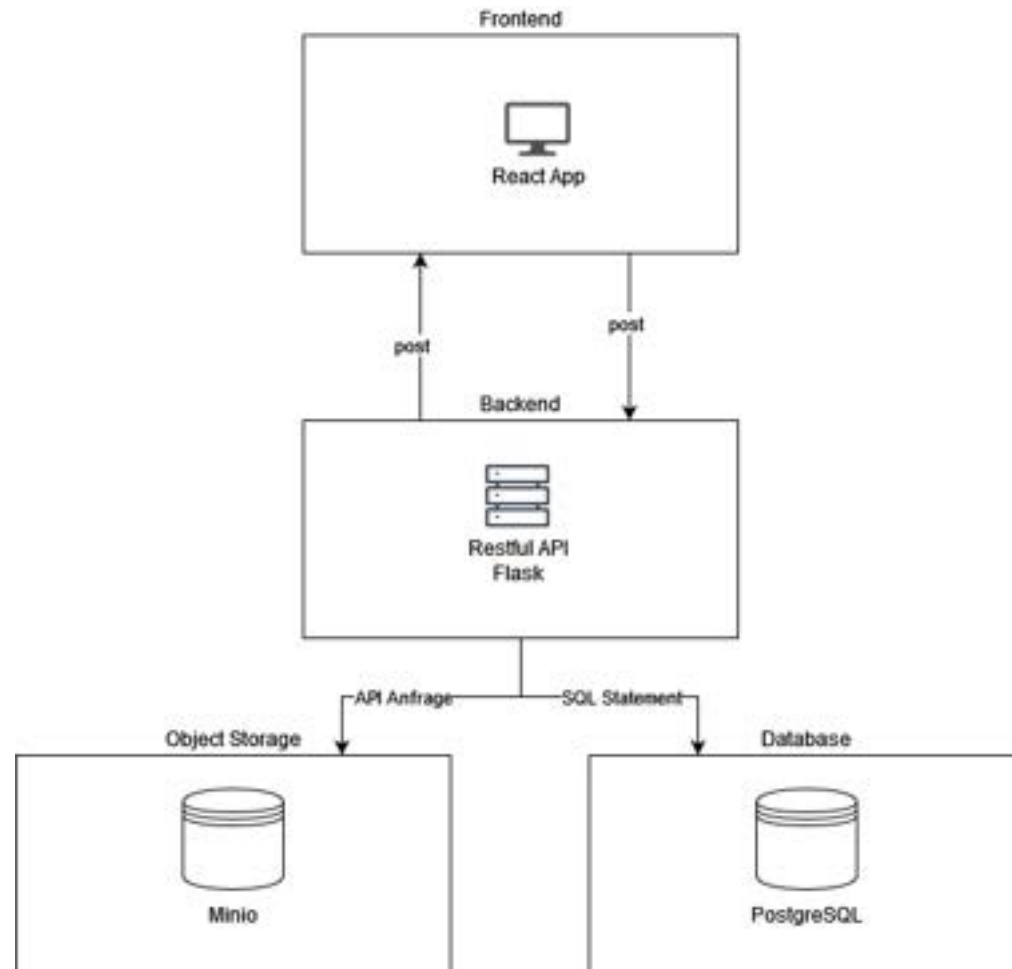
Architektur

Client-Server-Architektur

3-Schichten-Architektur

Microservice-Architektur

Event-basierte-Architektur

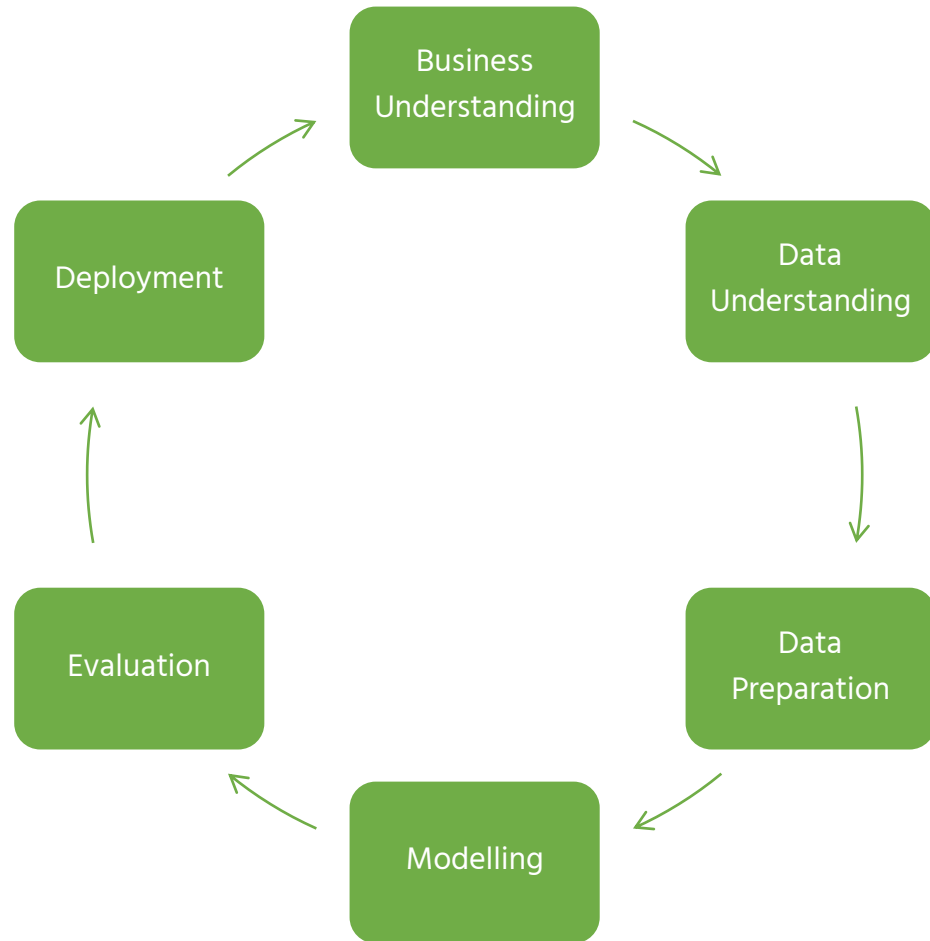


Grundlagen

- Recherche und Implementierung geeigneter KI-Methoden zur Auswertung von Bilddaten zur Fahrradinfrastruktur



Vorgehensmodell



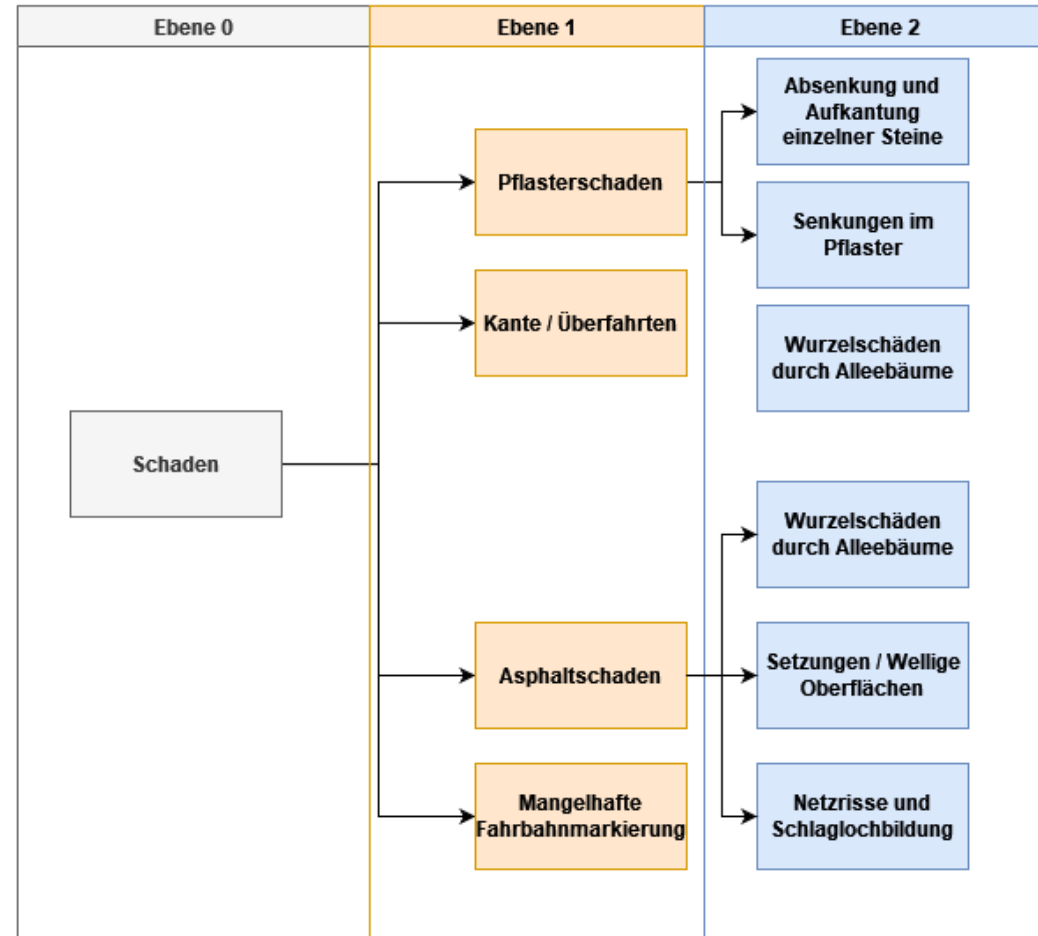
CRISP-DM



Business Understanding

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

- Ebenenkonzept
- Absprache u. A. mit Planungsbüro VIA
- Regelwerk H EBRA
- Fokus auf Asphalt- und Pflasterschäden



Data Understanding

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment



Schlagloch



Riss



Erhebung



Absenkung

- Bilder für Datensatz selbst aufgenommen
- Bilder aus verschiedenen Winkeln, bei verschiedenem Wetter
- Vorhandene Bilder aus Data Lake genutzt

Data Preparation

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

Iteration	Anzahl Bilder	Instanzen	Klassen	Beschreibung
1	500	650	Schlagloch, Riss, Absenkung, Erhebung	
2	1100	1500	Schlagloch, Riss, Absenkung, Erhebung	
3*	1500	2100	Schlagloch, Riss	
4	2000	2800	Schlagloch, Riss, Flicker	
5	6000	15200	Schlagloch, Riss	Einbindung Google Data Challenge
6	7500	15200	Schlagloch, Riss, Background	Hinzufügen von Noise

*ab Iteration 3 zusätzlich mit Klassifikation auf "damage" gearbeitet

Realisierung

Business Understanding – Data Understanding – Data Preparation – **Modelling** – Evaluation – Deployment



Objekterkennung
mit YoloV8
als „damage“



Erkannter „damage“ wird
ausgeschnitten



Klassifikation mit
efficientnet

- Unterteilung in
„Riss“ und „Schlagloch“



Realisierung

Business Understanding – Data Understanding – Data Preparation – **Modelling** – Evaluation – Deployment



Realisierung

Business Understanding – Data Understanding – Data Preparation – **Modelling** – Evaluation – Deployment



Realisierung

Business Understanding – Data Understanding – Data Preparation – **Modelling** – Evaluation – Deployment



Realisierung

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

- Manuelle Test mit neu hochgeladenen Bilder
- Metriken:
 - Precision
 - Recall
 - mAP50
 - Konfusionsmatrix



Realisierung

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

Konfusionsmatrix		Wahr		
		Riss	Schlagloch	Background
Vorhersage	Riss	0.83		0.45
	Schlagloch		0.90	0.55
	Background	0.17	0.10	



Realisierung

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

YOLO-Modelle:

Model	Precision	Recall	mAP50
Iteration 4D	0.75357	0.59375	0.65509
Iteration 5	0.96457	0.86105	0.91608
Iteration 5D	0.96128	0.84733	0.91604
Iteration 6	0.67665	0.42463	0.49554
Iteration 6D	0.60201	0.49097	0.51116

efficientNet-Modelle:

Model	Precision	Recall
Iteration 4	0.96938776	0.87962963
Iteration 5	0.950735321	0.99126467



Realisierung

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation – Deployment

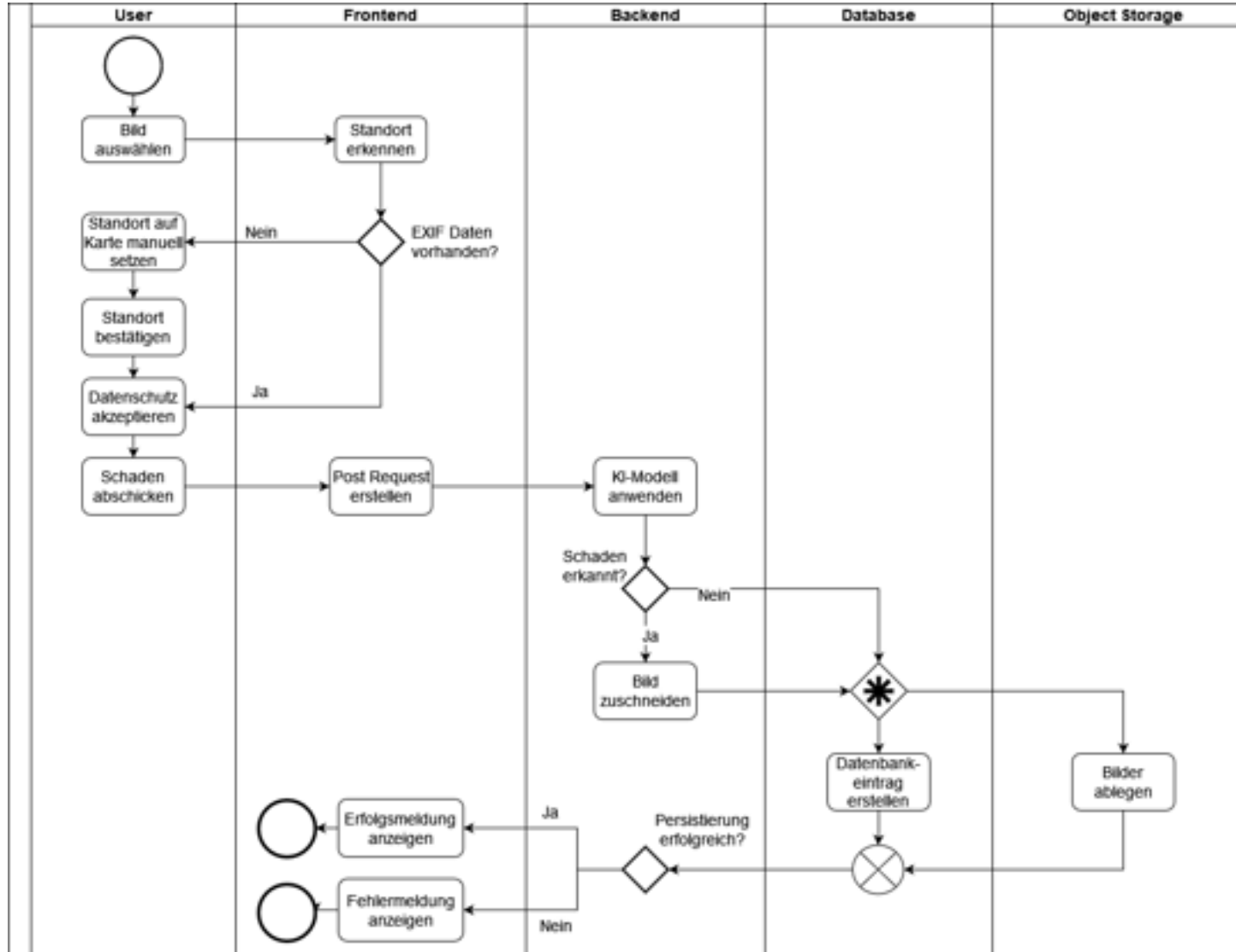
- Herausforderungen:
 - Klassen sehr komplex
 - Background wird zu oft erkannt (zu viel False Positiv)

- Lösungsansätze:
 - Datensatz erhöhen und weiter aufbauen
 - Data Augmentation



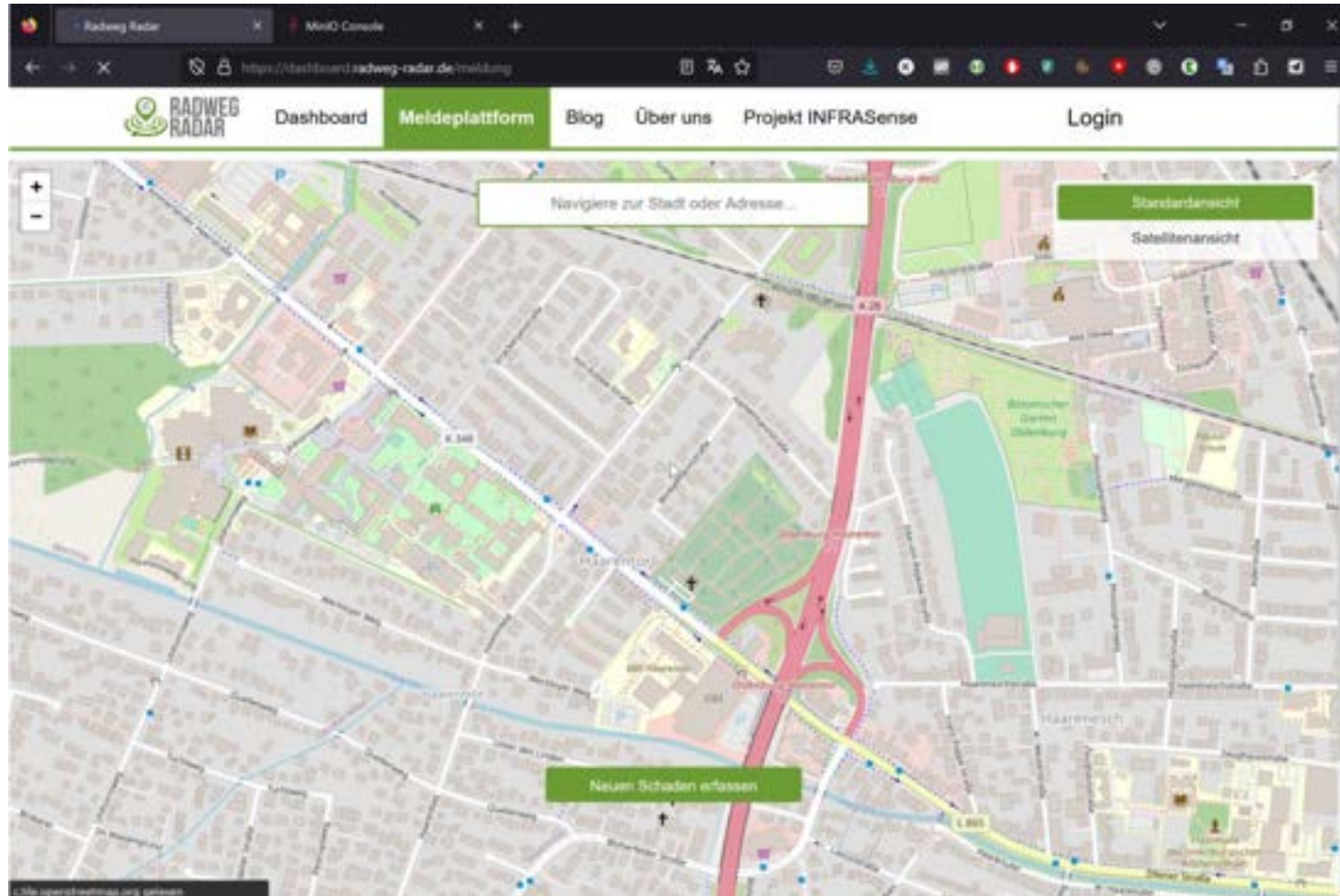
Realisierung

Business Understanding – Data Understanding – Data Preparation – Modelling – Evaluation - **Deployment**



Realisierung

www.radweg-radar.de/meldung



Funktionale Anforderungen

Nr.	Usecase	Art	Priorität
U1	Schaden melden	Frontend	Must
U2	Schaden anzeigen	Frontend	Must
U3	Schaden klassifizieren	Backend	Must
U4	Standort bestimmen	Frontend	Must



Nicht-funktionale Anforderungen

Nr.	Usecase	Beispiel
NFA1	Wartbarkeit	Software ist dokumentiert
NFA2	Benutzbarkeit	Einfache Gestaltung, mobile
NFA3	Effizienz	Wahl der Architektur
NFA4	Funktionalität	Funktionale Anforderungen weitestgehend umgesetzt
NFA5	Übertragbarkeit	Endpunkte im Backend lassen sich implementieren, auch ohne KI-Modell mit ein paar Änderungen nutzbar
NFA6	Zuverlässigkeit	Fehlertoleranz ausbaufähig im Frontend, da Prototyp



Limitationen / Ausblick

Frontend	Backend
Ansatz, wie mit reparierten Schäden umgegangen werden kann	Qualität des KI-Modells verbessern
Prüfung, ob Schaden bereits gemeldet ist	Aufnahme weiterer Schäden (z.B. Pflasterschäden)
Information an Stadt/Straßenamt über Schadensmeldung	Detaillierteres Labelling
Freitext	Übergang Schlagloch / Riss
Benutzerverwaltung	Bewertung der Schwere eines Schadens
Mehrere Schäden gleichzeitig verarbeiten	Mehrere Schäden gleichzeitig verarbeiten



Fazit & Lessons Learned

- Projekt nutzbar und weiterentwickelbar
 - Ergebnisse wurden kontinuierlich verbessert
- Neue Iteration: Vorgehensweise bei weiteren Schäden / Bilder
 - Plattform zum Sammeln der Daten geschaffen
- Evaluation eher fertigstellen, um fundiertere Entscheidungen treffen zu können

- Hohe Lernkurve
- Zielsetzung/Aufgabenstellung nach Projektbeginn selbst ausgearbeitet
 - eher schärfen und kritischer hinterfragen



Vielen Dank für Ihre Aufmerksamkeit!

Gibt es Fragen?

Kontakt

Malte Kuhlmann, Malte Schott
malte.kuhlmann@uni-oldenburg.de
malte.schott@uni-oldenburg.de

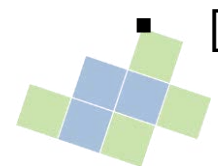
..oder bevorzugt:

pg@radweg-radar.de



Literaturverzeichnis

- [Ary21] Arya, Deeksha; Maeda, Hiroya; Ghosh, Sanjay Kumar; Toshniwal, Durga ; Omata, Hiroshi ; Kashiya, Takehiro ; Seto, Toshikazu; Mraz, Alexander ; Sekimoto, Yoshihide (2021), "RDD2020: An Image Dataset for Smartphone-based Road Damage Detection and Classification", Mendeley Data, V1, doi: 10.17632/5ty2wb6gvg.1
- [Bal11] Balzert, Helmut, and Helmut Balzert. "Nichtfunktionale Anforderungen." Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb (2011): 109-133.
- [DFOO11] Jurgen Dunkel, Alberto Fernandez, Ruben Ortiz, and Sascha Ossowski. Event-driven architecture for decision support in traffic management systems. Expert Systems with Applications, 38(6):6530–6539, 2011.
- [Gri18] Grinberg, M. (2018). Flask web development: developing web applications with python. " O'Reilly Media, Inc."
- [Haa21] Haase, M.: Hinweise zur einheitlichen Bewertung von Radverkehrsanlagen (H EBRA). In: Straßenverkehrstechnik 2021, H. 10, S. 765-767
- [JCQ23] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>
- [Kam10] Channu Kambalyal. 3-tier architecture. Retrieved On, 2(34): 2010, 2010.



Literaturverzeichnis

- [Koo21] Koonce, Brett. "EfficientNet." Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization (2021): 109-123.
- [Mar03] Mariano, Vladimir Y., et al. "Performance evaluation of object detection algorithms." 2002 International Conference on Pattern Recognition. Vol. 3. IEEE, 2002.
- [Olu14] Haroon Shakirat Oluwatosin. Client-server model. IOSR Journal of Computer Engineering, 16(1):67–71, 2014.
- [SZE22] Szeliski, Richard. Computer vision: algorithms and applications. Springer Nature, 2022.
- [UCA19] Uzayr, S., Cloud, N., Ambler, T. (2019). React. In: JavaScript Frameworks for Modern Web Development. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4995-6_13
- [Ull88] Jeffrey D. Ullman. Principles of Database and Knowledge-Base Systems. Classical Database Systems. Volume 1. Computer Science Press. 1988.
- [Wir00] Wirth, Rüdiger, and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining." Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. Vol. 1. 2000.
- [WLS20] Waseem, Muhammad, Peng Liang, and Mojtaba Shahin. A systematic mapping study on microservices architecture in devops. Journal of Systems and Software, 170:110798, 2



Bildquellen

- [1] <https://www.infrasense.de>
- [2] <https://gemeinsam.oldenburg.de/oldenburg/de/flawRep/54305>
- [3] <https://www.emsos.osnabrueck.de/discover>
- [4] <https://www.hs-furtwagen.de>
- [5] https://www.radar-online.net/fileadmin/radar_logo.png
- [6] <https://www.bmdv.bund.de/SharedDocs/DE/Artikel/DG/mfund-projekte/azur.html>
- [7] eigene Darstellung, in Anlehnung an http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf





RADWEG RADAR

Data

PG INFRASense

Florian Dyck, Richard Schulte

27.09.2023

Agenda

1

Zielsetzung

2

Technologieauswahl

3

Daten

4

Data Warehouse

5

Score-Berechnung

6

Live-Demo

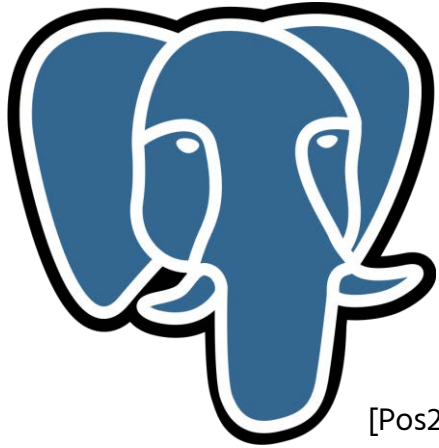


Zielsetzung

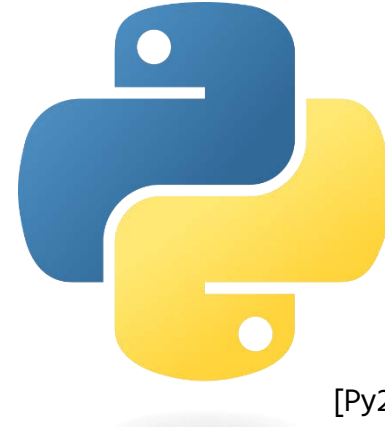
- Anbindung unterschiedlicher radverkehrsbezogener Datenquellen
- Datenaufbereitung für effiziente Analysen
- Erstellung einer API zur Datenabfrage im Dashboard
- Berechnung eines Straßen-Scores zur Infrastruktur-Bewertung



Technologieauswahl



[Pos23, 1]



[Py23, 3]



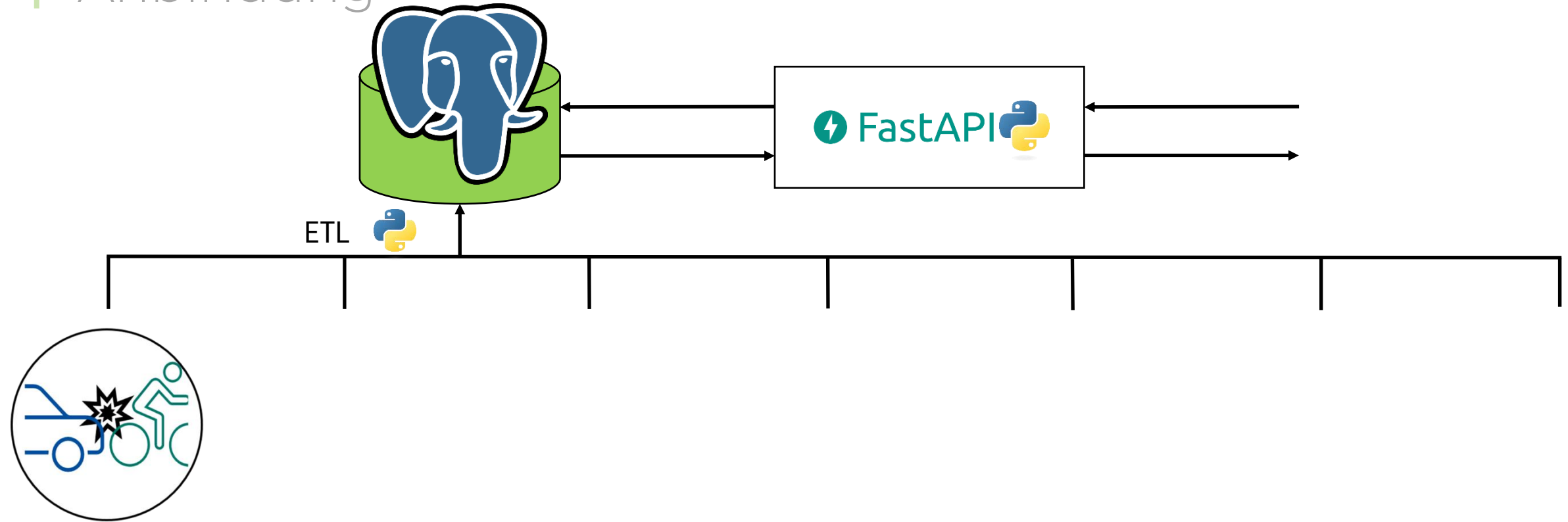
[Ppg23, 2]



[Fas23, 4]

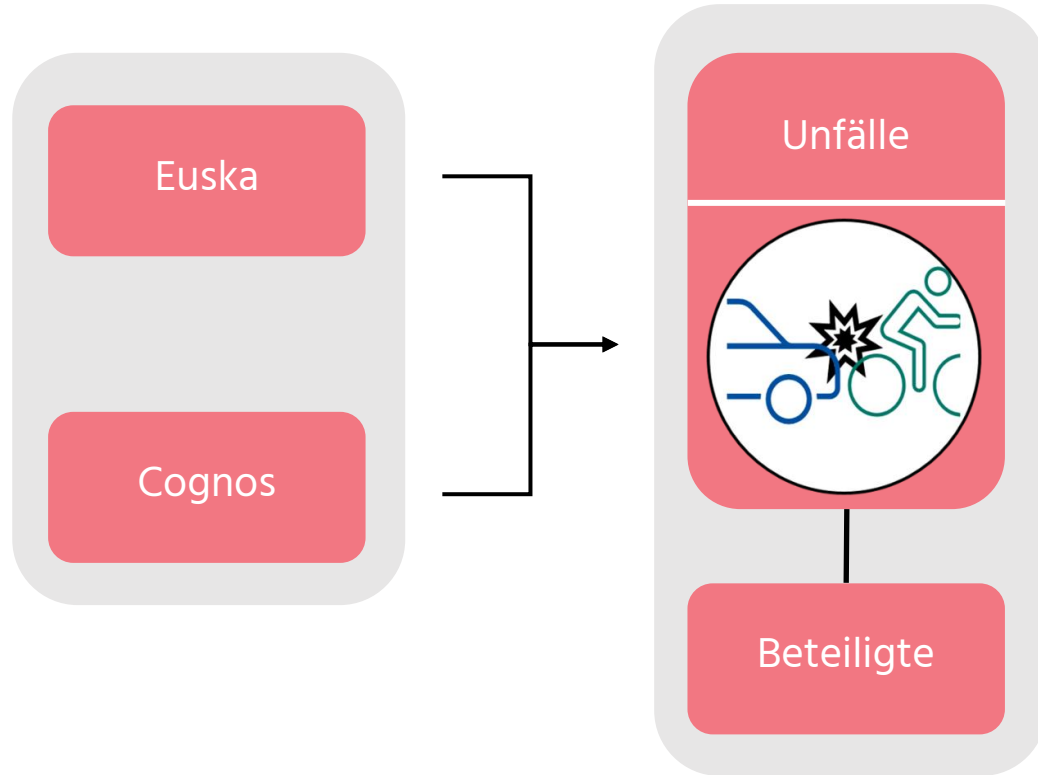
Daten

Anbindung



Unfälle

Daten Unfälle



- Beteiligte und Verursacher
- Unfallschwere
- Alkohol- und Drogeneinfluss
- Schadenshöhe
- Standort
- Persönliche Informationen
- Unfallbeschreibung



Daten

1/m:n-Beziehungen

Unfall #1	Fahrrad
	Auto



Daten

1/m:n-Beziehungen

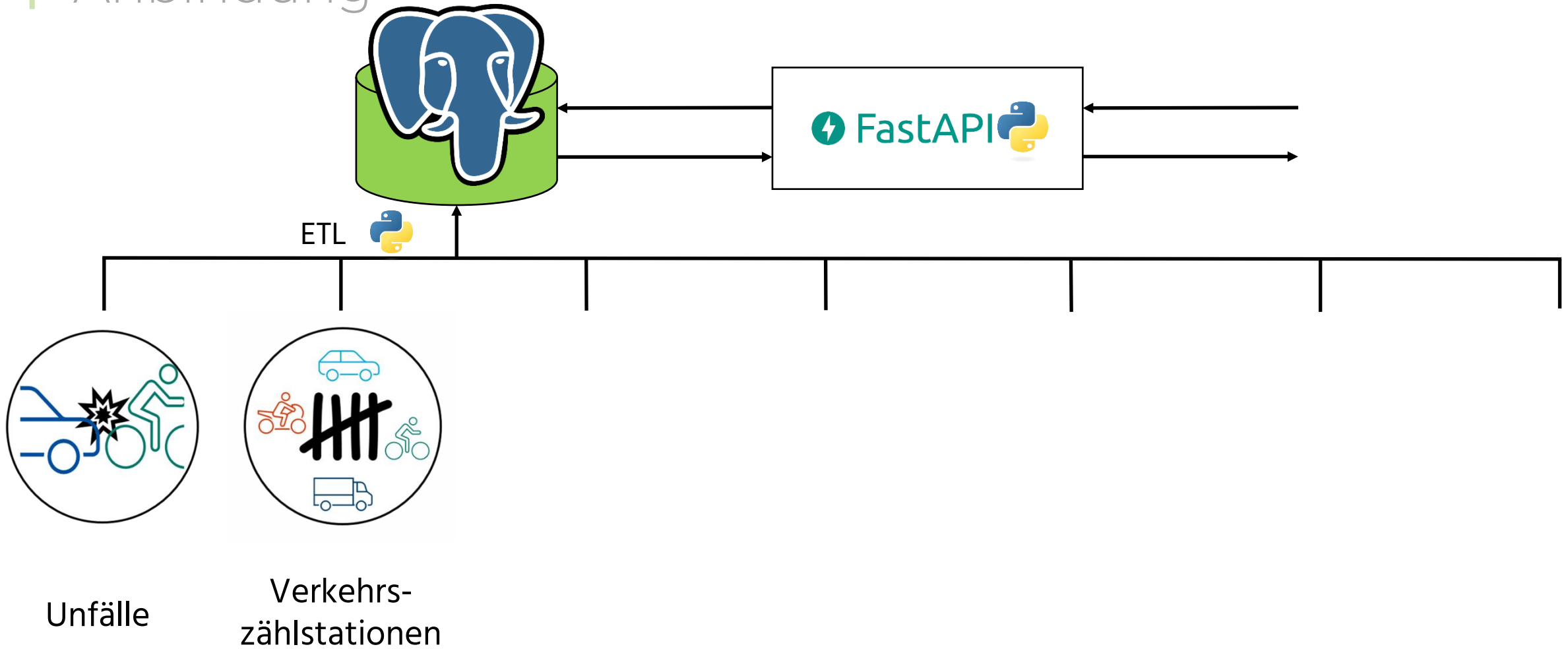
Unfall #1

Unfall #1	Fahrrad
Unfall #1	Auto



Daten

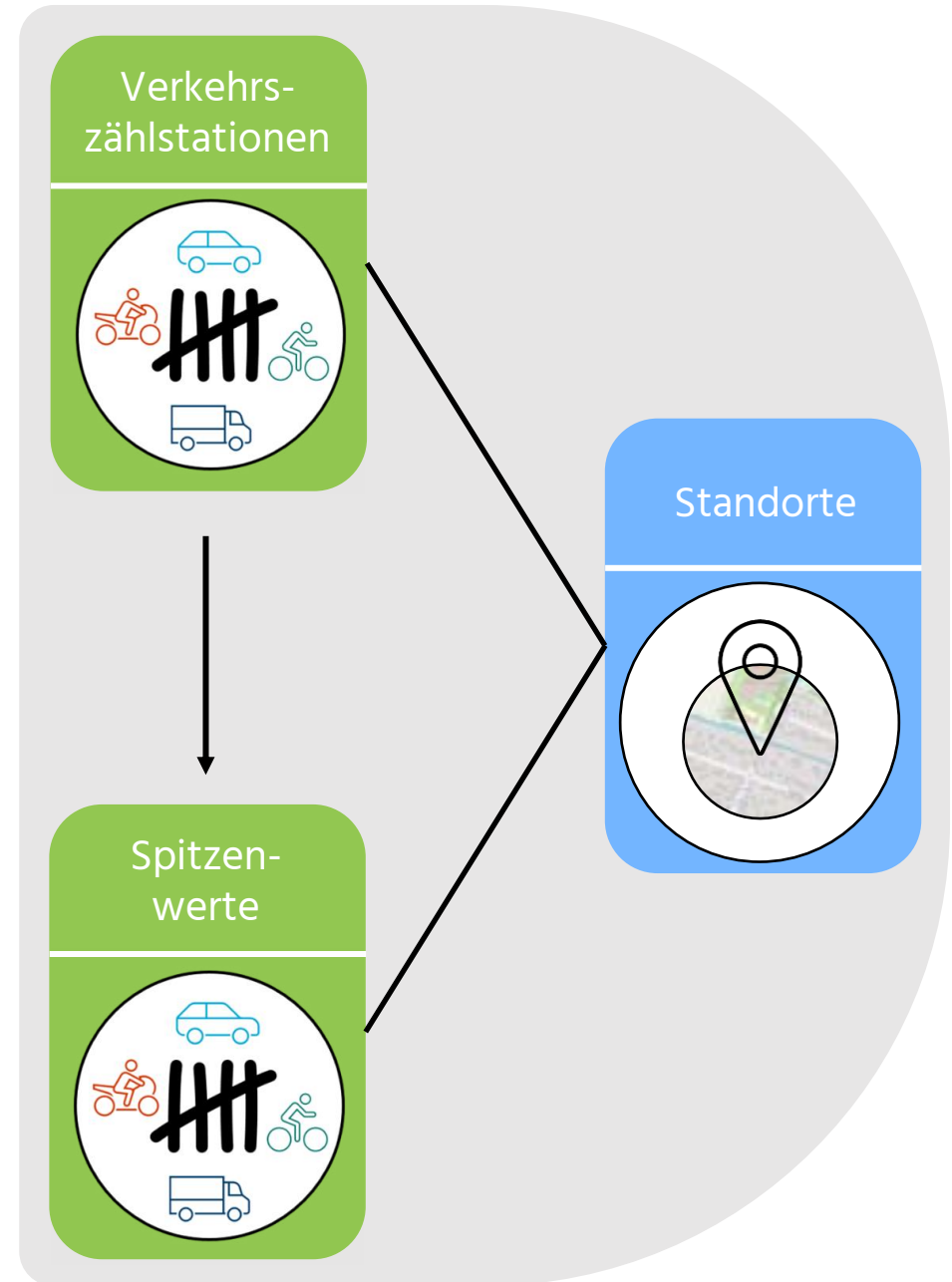
Anbindung



Daten

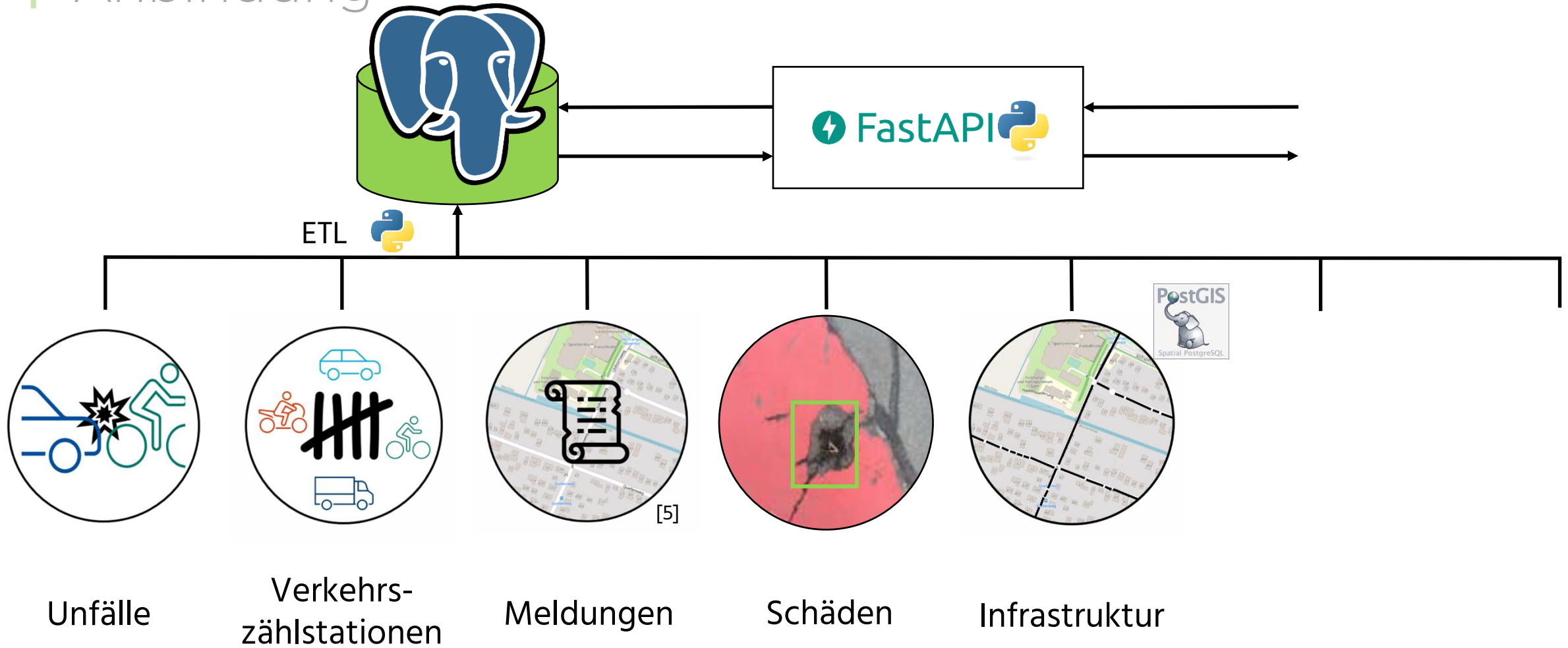
Verkehrszählstationen

- Temporär und permanent
- PKW, LKW, Motorrad, Fahrrad, Fußgänger
- Verkehrsvolumen, Geschwindigkeit, Verkehrsauslastung



Daten

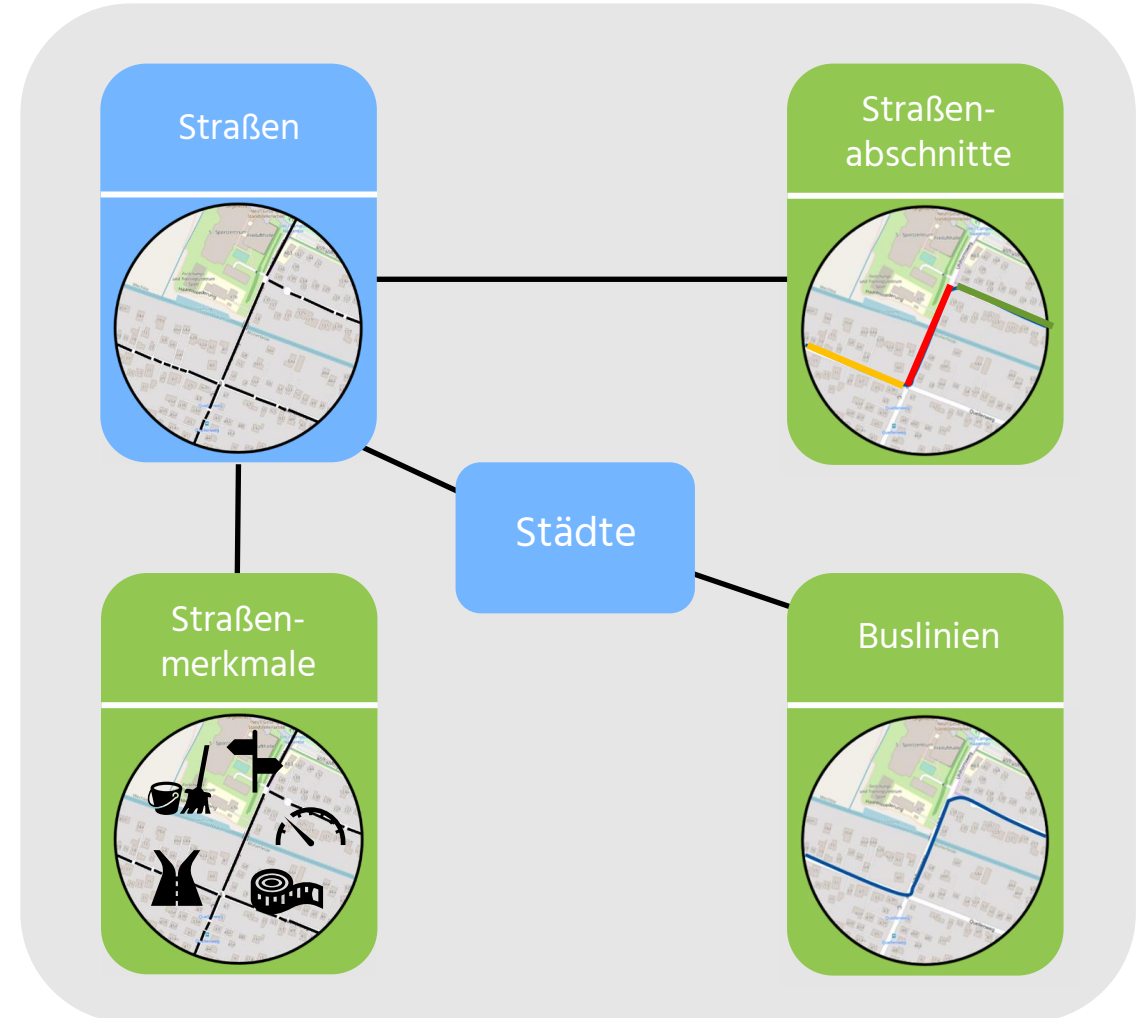
Anbindung



Daten

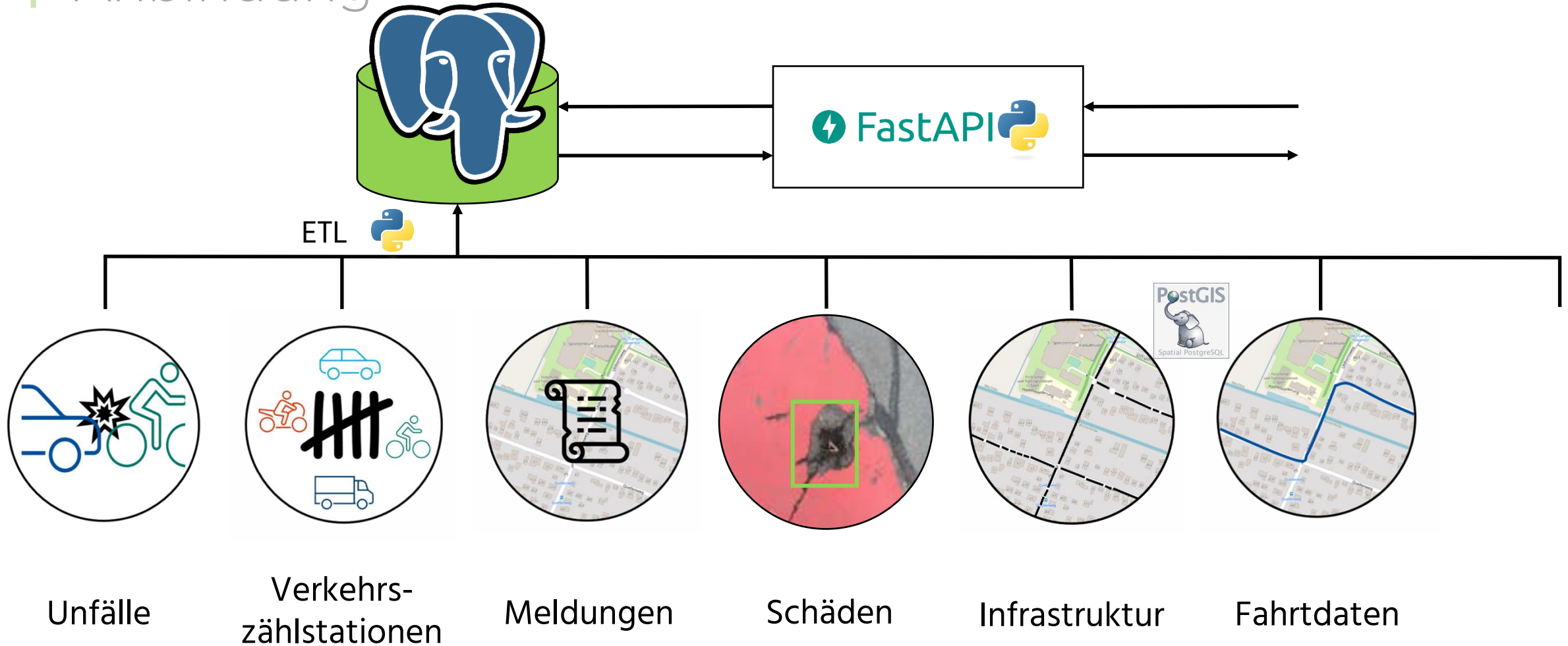
Infrastruktur

- Geodaten/Geometrien
 - Straßen
 - Straßenabschnitte
 - Buslinien und Haltestellen
- Beschreibende Daten
 - Längen, Steigungen, Breiten
 - Straßenreinigung
 - Oberflächen, Führungsformen
 - Wegetypen



Daten

Anbindung



Daten

Fahrtdaten

- Rohdaten

- Sensordaten
- Metadaten



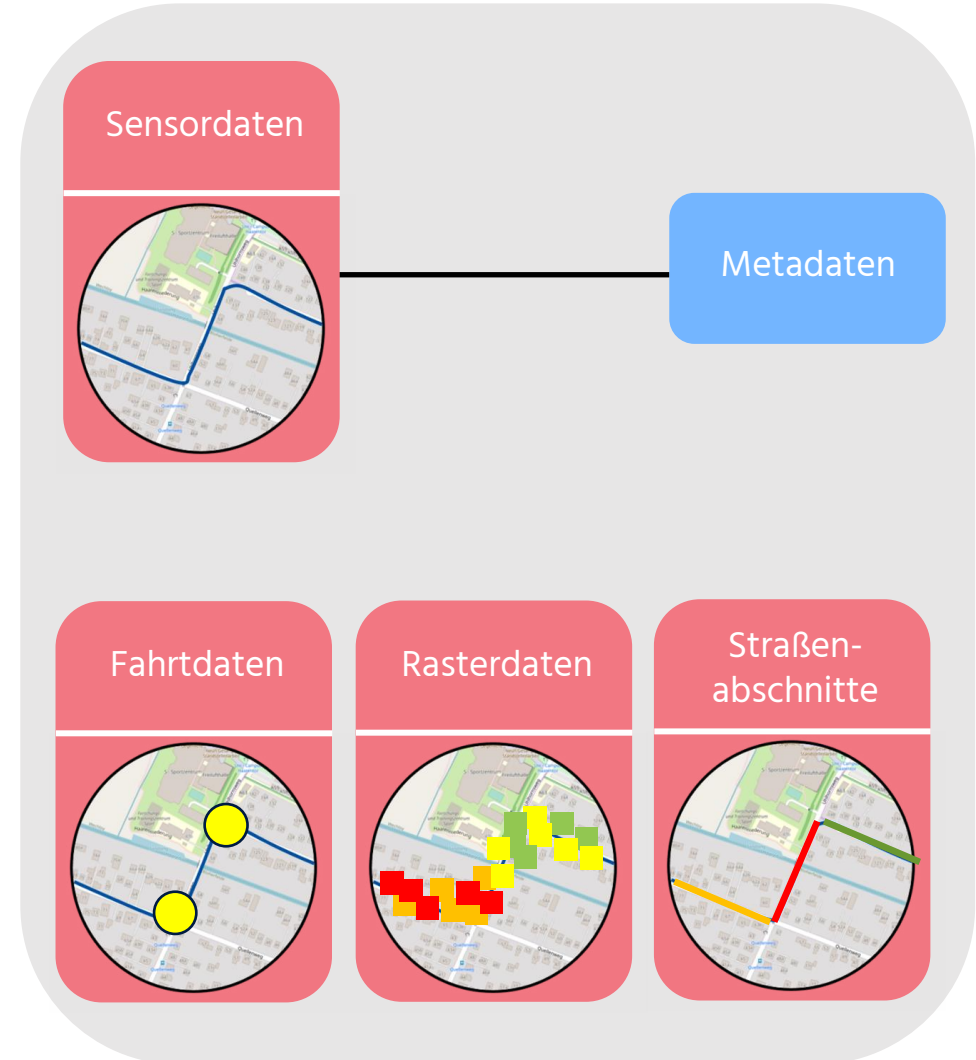
[7]

- Aggregierte Daten

- Straßenabschnitte, Kreuzungen, Raster
- Qualitätsdaten
- Nutzerbewertungen

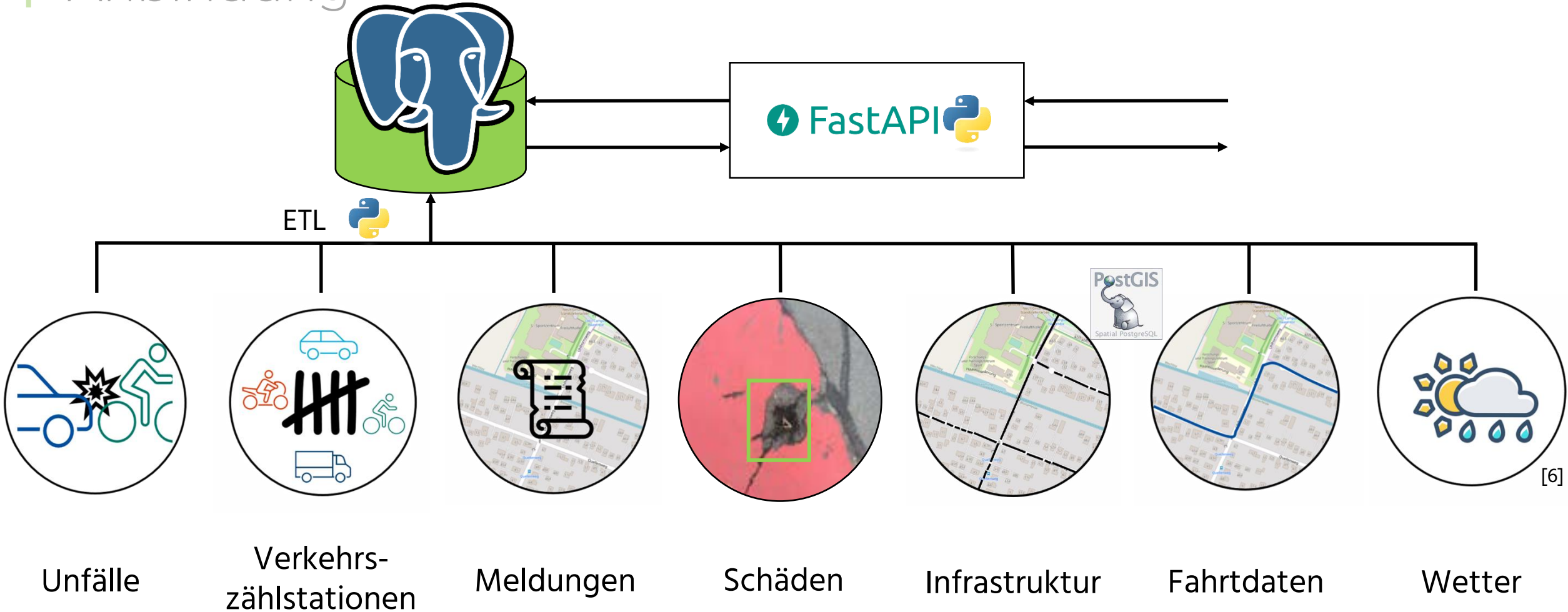


[7]



Daten

Anbindung

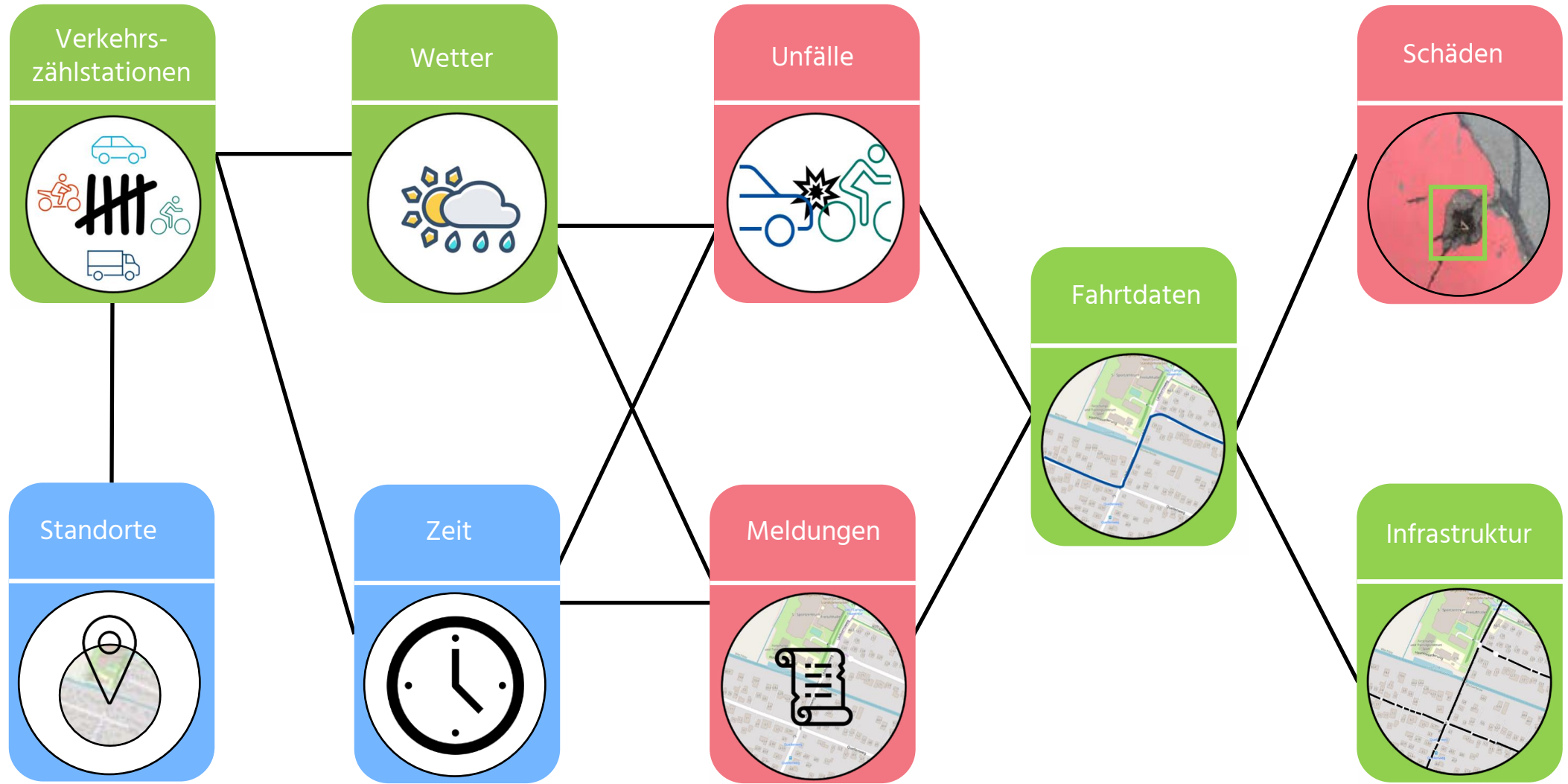


[6]



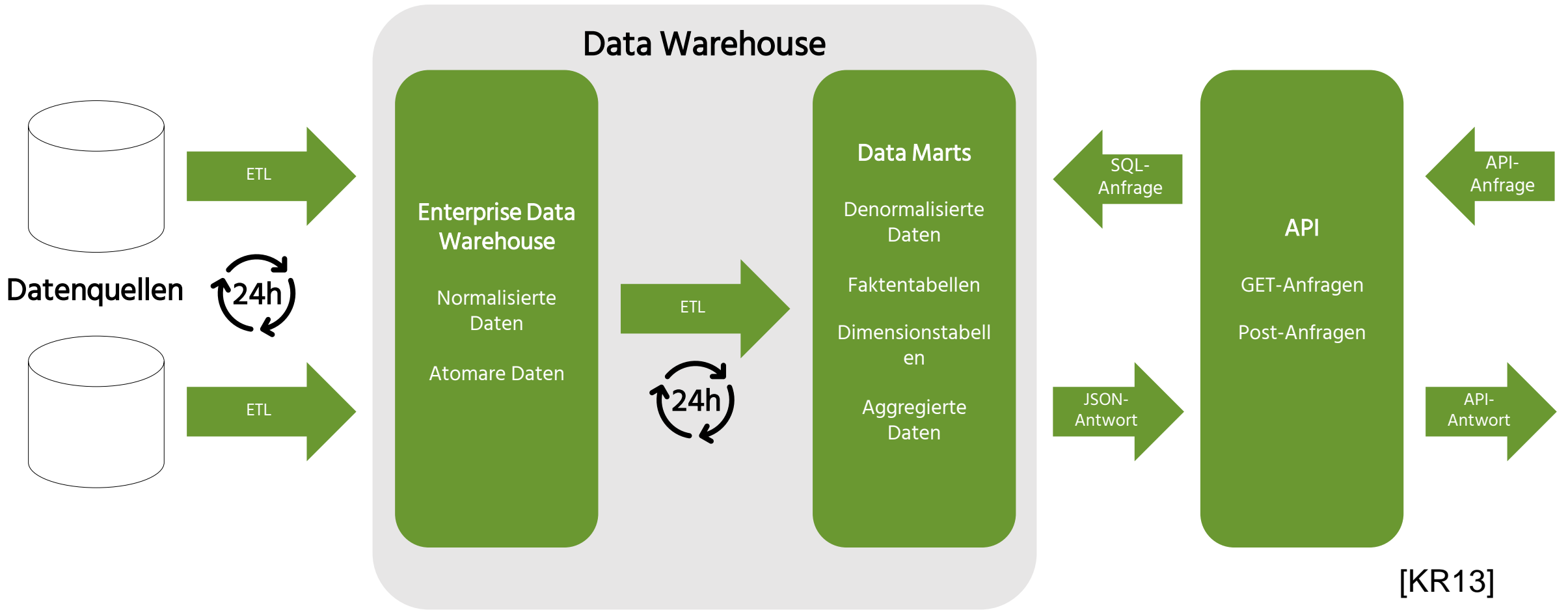
Data Warehouse

Schema



Data Warehouse

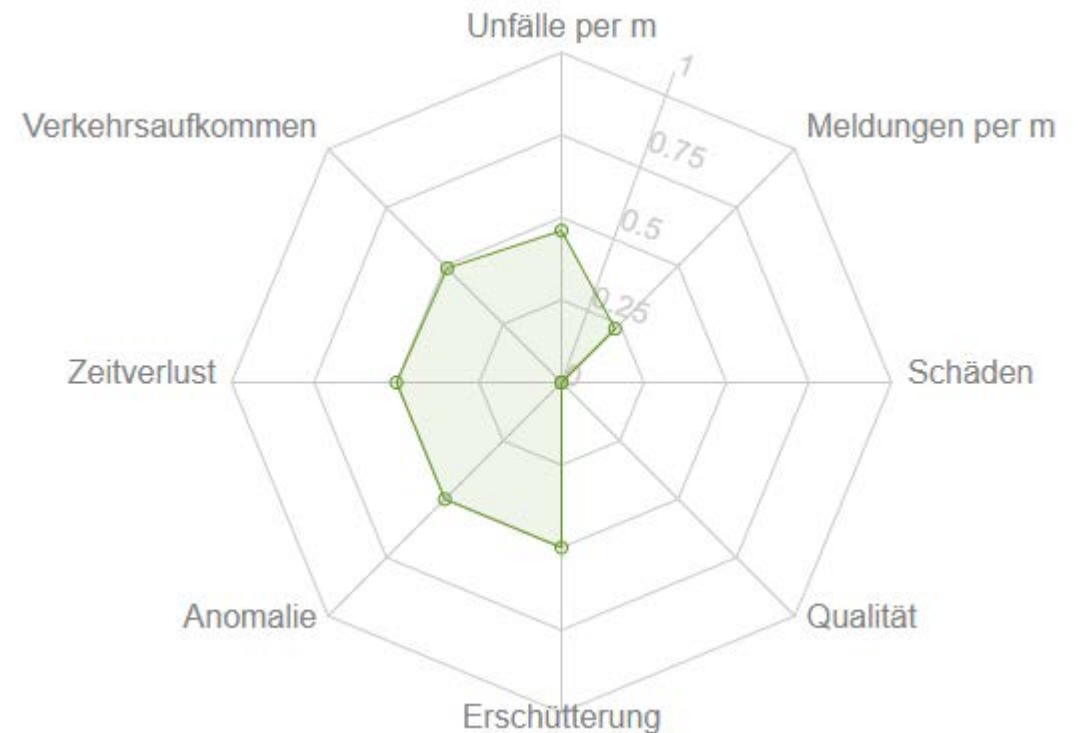
Design



Score-Berechnung

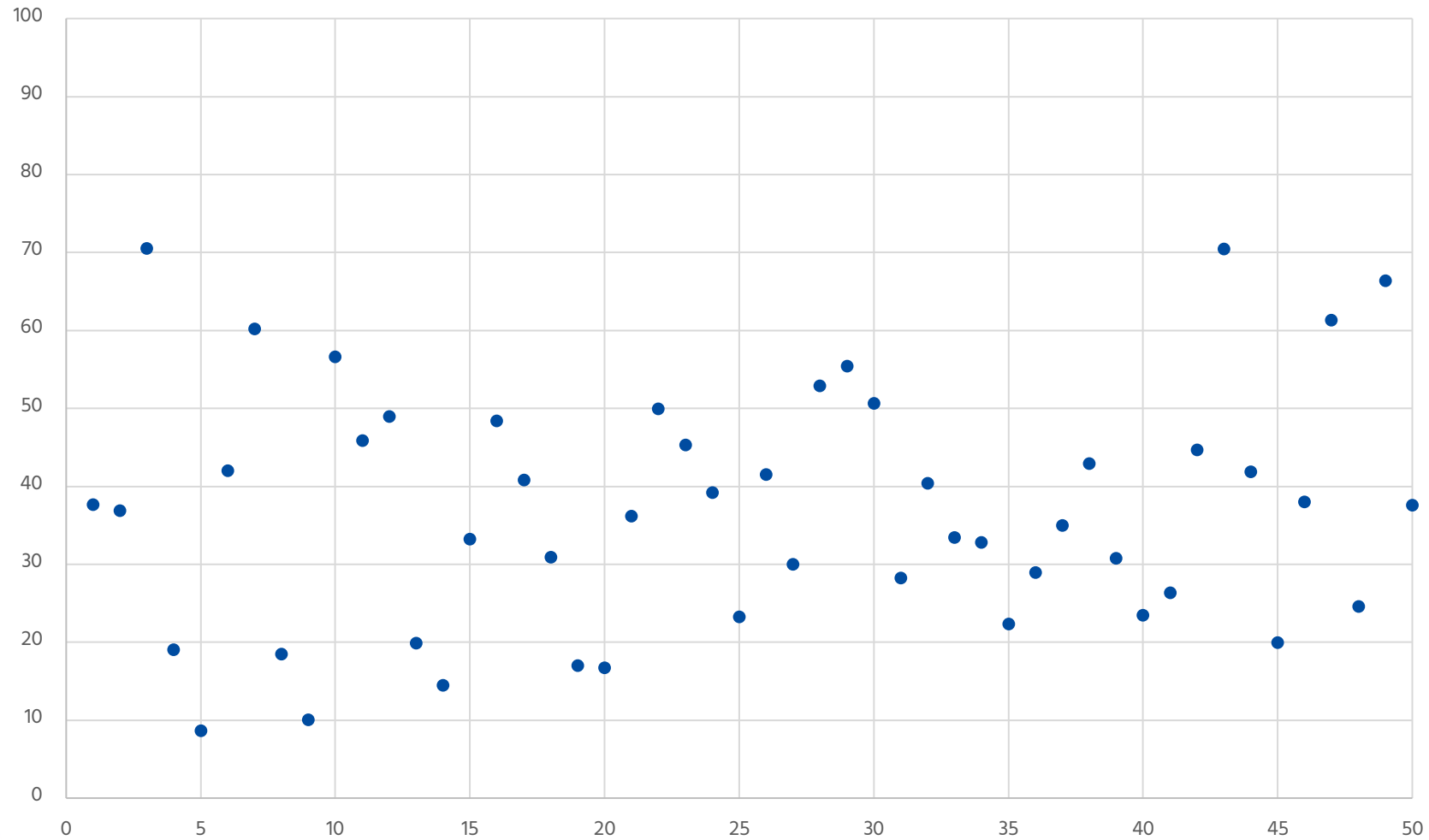
Verwendete Metriken

- Mengen pro Streckenlänge
 - Straßenschäden
 - Unfälle
 - Meldungen
- Fahrtdaten
 - Erschütterungsgrad unter Berücksichtigung der Geschwindigkeit
 - Anomaliegrad
 - Zeitverlust
- Nutzerbewertungen



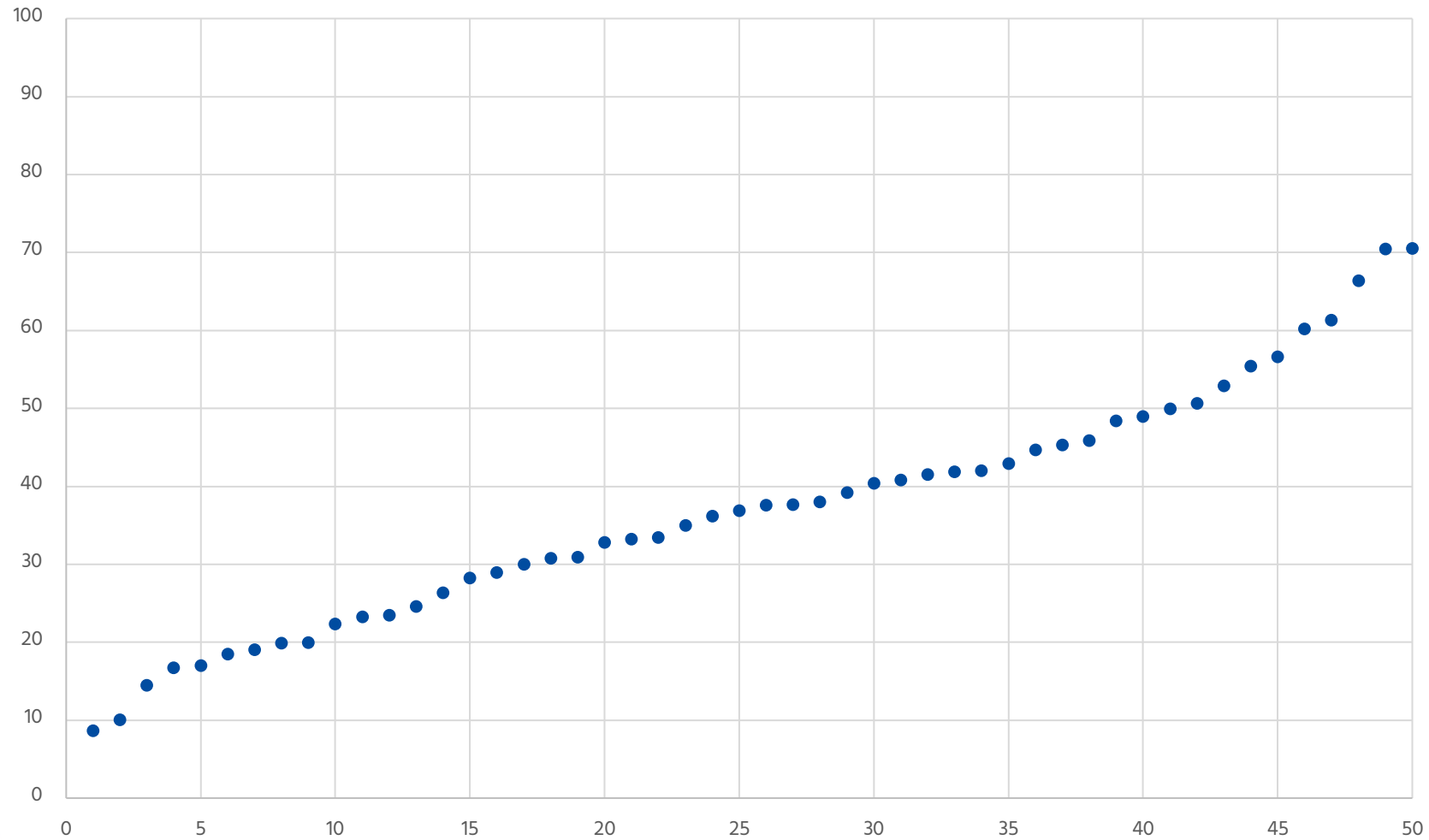
Score-Berechnung

Messdaten



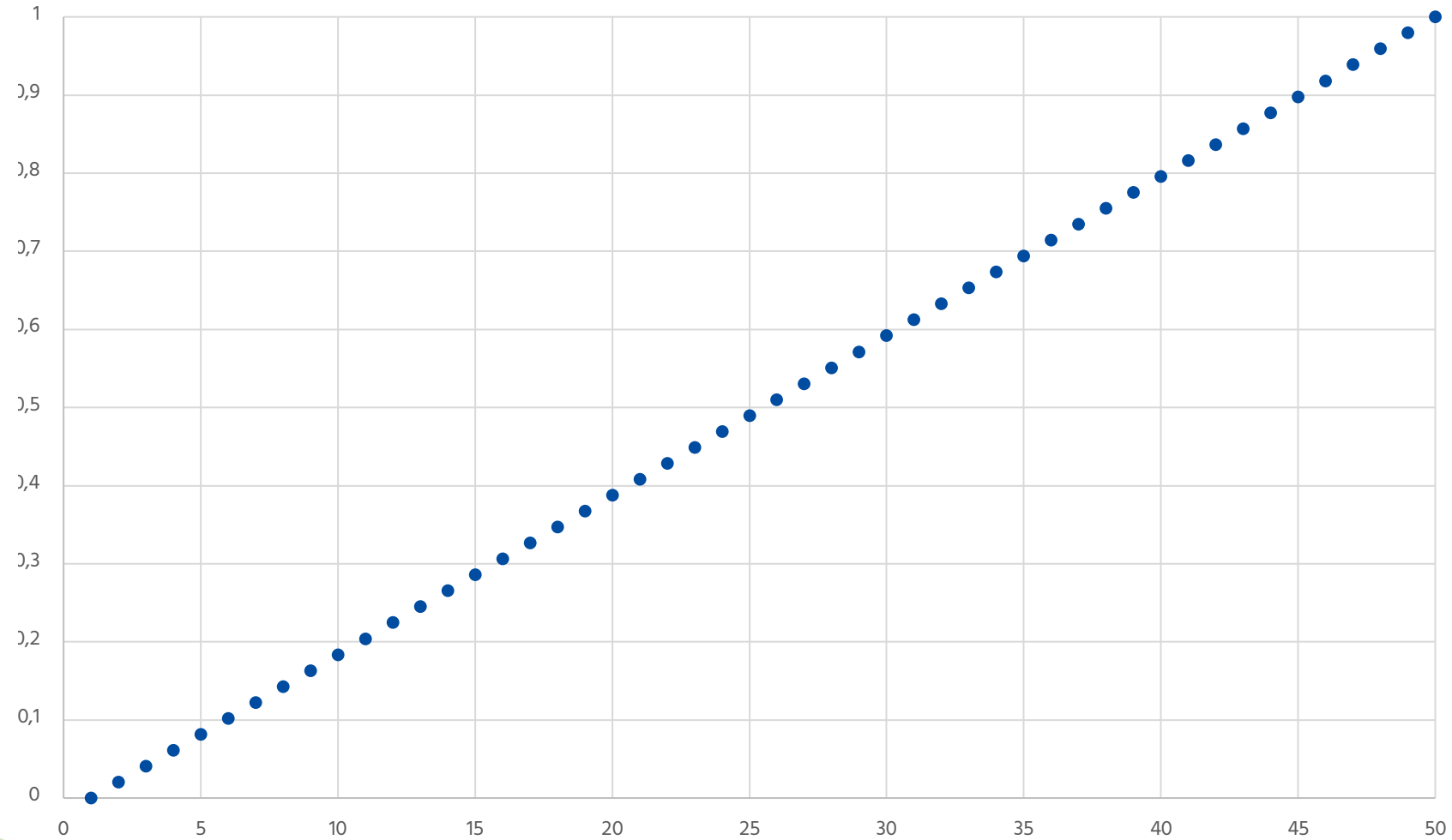
Score-Berechnung

Sortiert



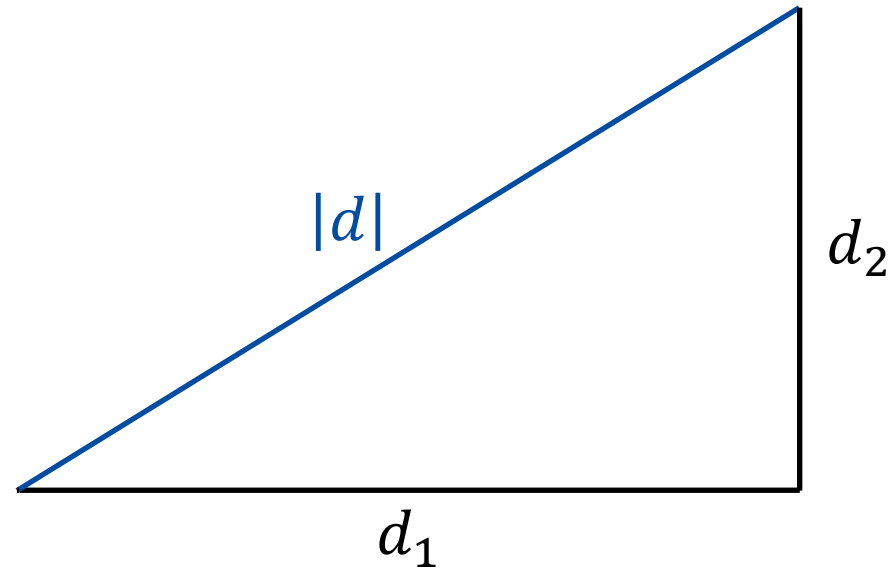
Score-Berechnung

Perzentil



Score-Berechnung

Euklidischer Abstand

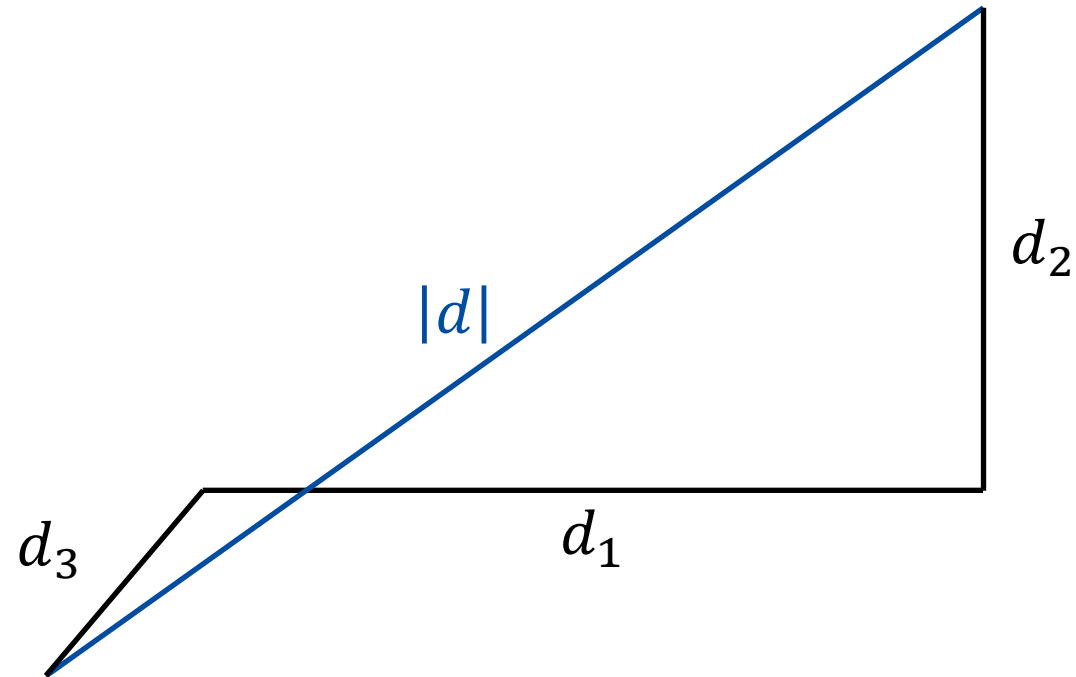


$$|d| = \sqrt{d_1^2 + d_2^2}$$



Score-Berechnung

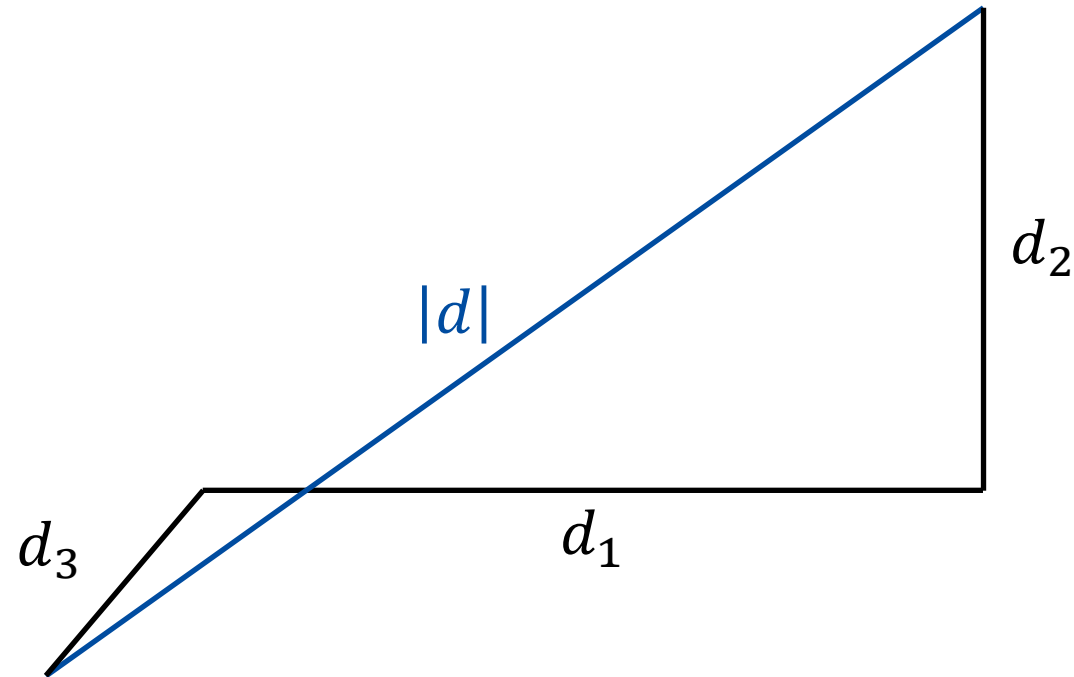
Euklidischer Abstand



$$|d| = \sqrt{d_1^2 + d_2^2 + d_3^2}$$

Score-Berechnung

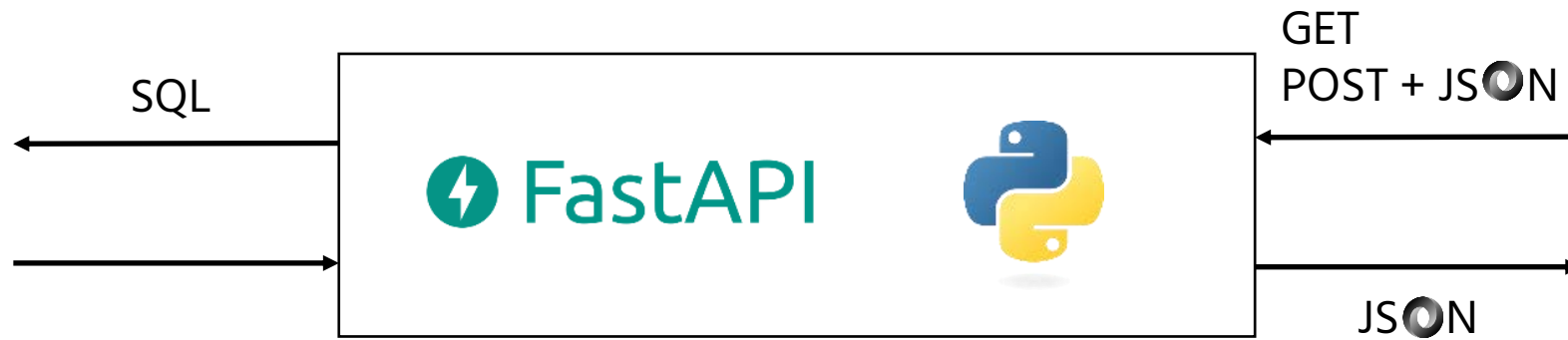
Euklidischer Abstand



$$|d| = \sqrt{d_1^2 + d_2^2 + d_3^2 + \dots}$$



API



API

Live Demo



RadwegRadar FastAPI 0.1.0 QA33

[openapi.json](#)

How to build a query

When not using `group_by` all columns can be used as columns like this:
`{column_name}`

Number columns can be filtered like this:
`{column}: {column_name}, {operator}: {ConditionOperator}, {value}: {number}`

Dates can be filtered like numbers, with earlier dates being smaller:
`{column}: {column_name}, {operator}: {ConditionOperator}, {value}: {date}`

String columns can be filtered for specific values:
`{column}: {column_name}, {value}: {string}`

When using `group_by` only aggregated number columns (and "count") can be used like this:
`{column}: {column_name}, {aggregation}: {Aggregate}`

To use `group_by` simply group by column names:
`{column_name}`

The allowed values for "sort" are "ASC", "DESC" and "DEFAULT".
The result will be ordered ascending or descending by the first selected column respectively for the first two
"DEFAULT" will leave the result in its default order.

data

POST `/api/v1/bicycle_volume2` Bicycle Volume2

POST `/api/v1/bicycle_volume` bicycle_volume



RADWEG RADAR

Dashboard

PG INFRASense

Michael Birke, Mukhran Kamashidze, Alexander Tesch

27.09.2023

Agenda

1

Ziele

2

Technologien und Entwicklung

3

KPI

4

Visualisierung

5

Rollenkonzept

6

Live-Demo

7

Erweiterungsmöglichkeiten

8

Diskussion / Fragerunde



Ziele

- Individuelles, selbst konfigurierbares Dashboard
- Interaktives Design
- Filtern der Daten über Zeit, Ort, Eigenschaften
- Vergleich von Städten
- Straßenscore zum Vergleich verschiedener Straßen bzw. Straßenabschnitte
- Einbindung der BIQEmonitor-Metriken





Technologien

Technologien

radweg-radar.de



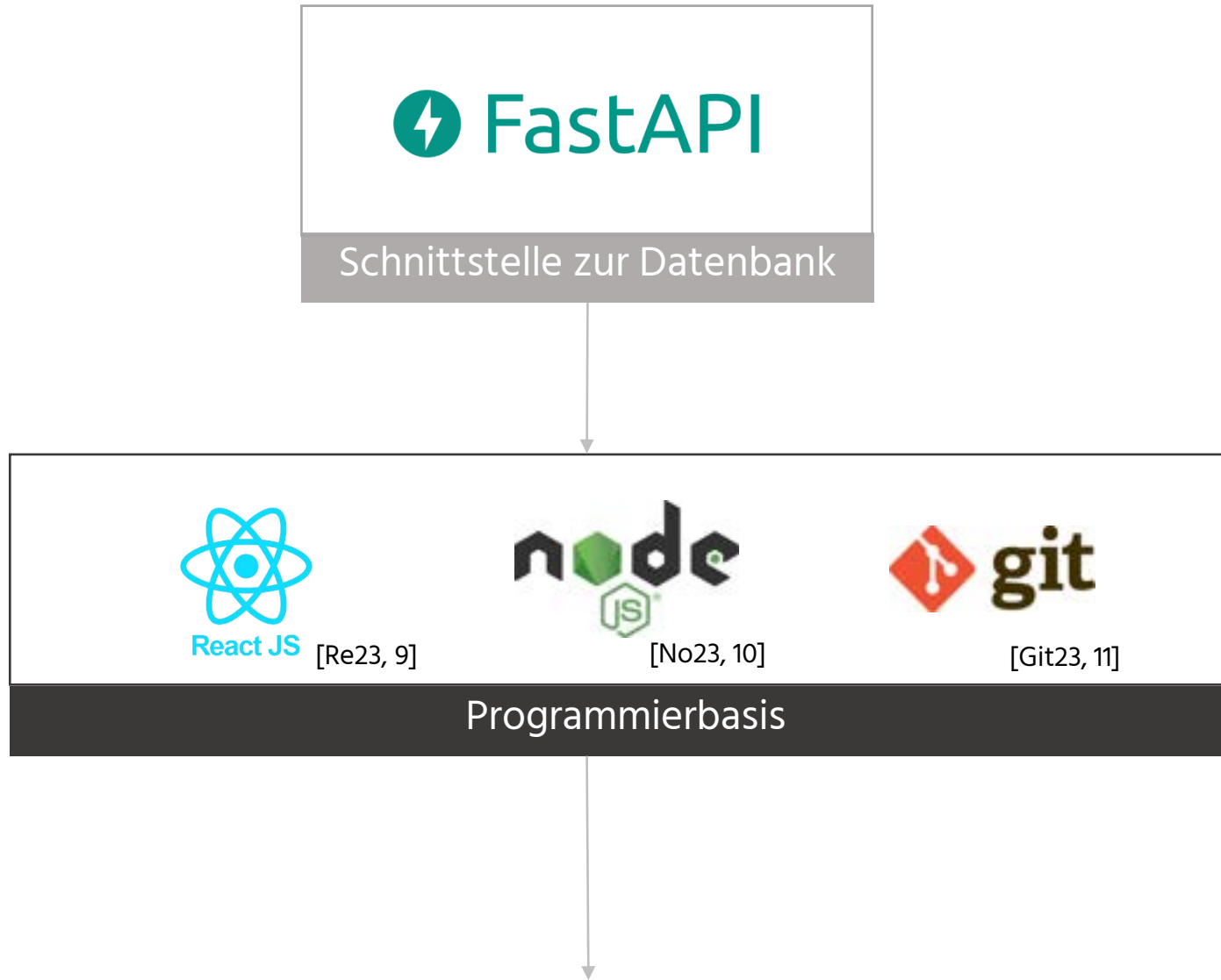
radweg-radar.de

Onlinepräsenz (Melo Free Template)

[Wp23, 8]

Technologien

Technischer Unterbau Dashboard



Technologien

Visualisierungstools



[Rl23, 12]



[Le23, 13]

Kartenvisualisierung

Recharts

[Rch23]

Graphenvisualisierung



Diagramm-Framework

Recharts

- Open Source
- State-of-the-Art Optik
- Benutzerfreundliche Animationen
- Ergänzung durch eigenen Code □ Extreme Flexibilität bei Anpassungen
- Umfangreiche Online-Dokumentation



Entwicklung

Virtuelle Maschinen



TEST VM

- Nur intern sichtbar
- Für die Entwicklung
- Testen von Anpassungen



PROD VM

- Frei im Web sichtbar
- Spiegelt fertige Änderungen wider





KPI-Realisierung

KPI-Realisierung

Oberkategorien

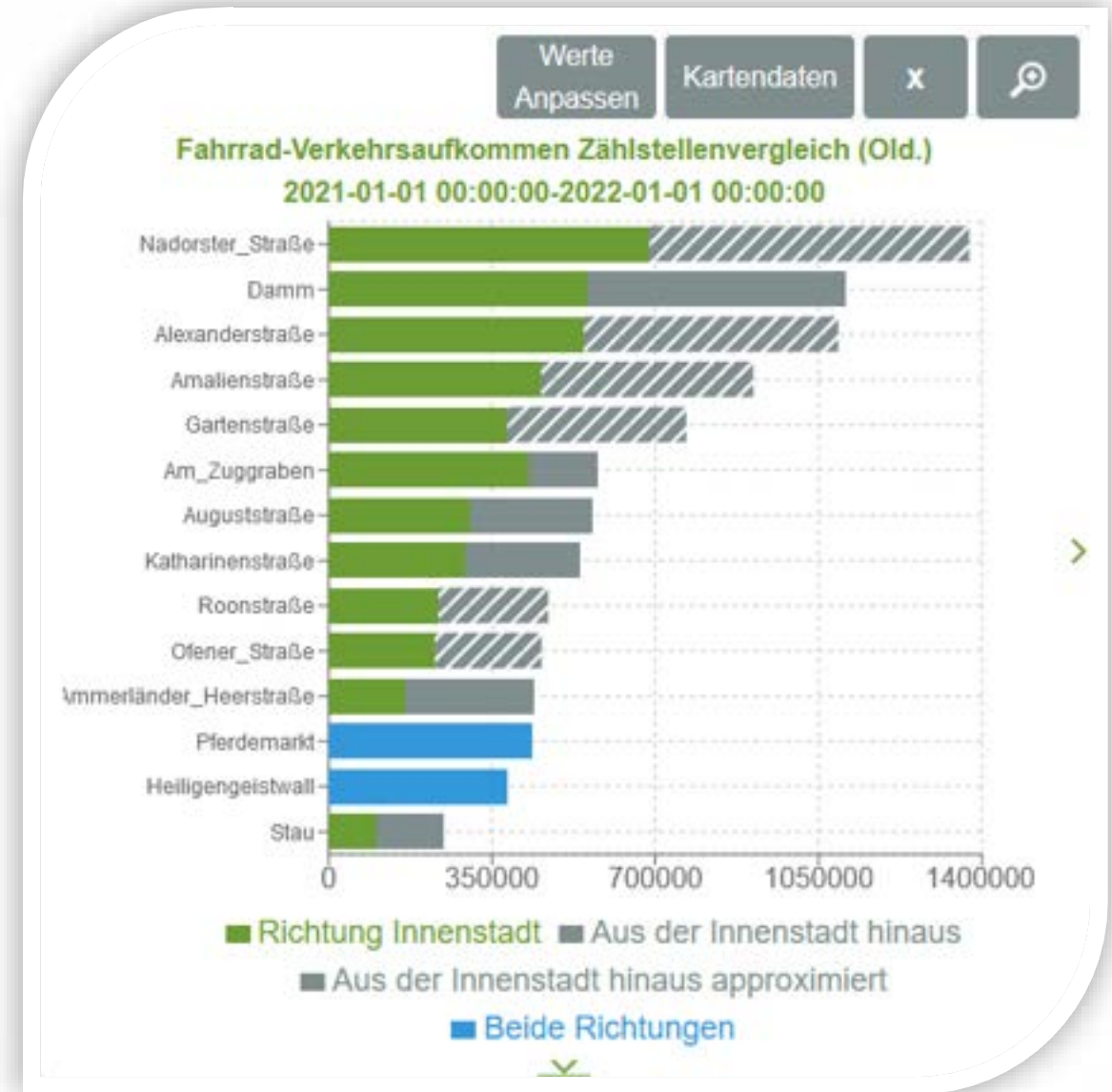
- Fahrrad-Verkehrsaufkommen
- Bürgermeldungen
- Unfälle
- Kartendarstellungen



KPI-Realisierung

Fahrrad-Verkehrsaufkommen

- Auswahl einzelner Zählstellen
- Zählstellenvergleich



KPI-Realisierung

Bürgermeldungen

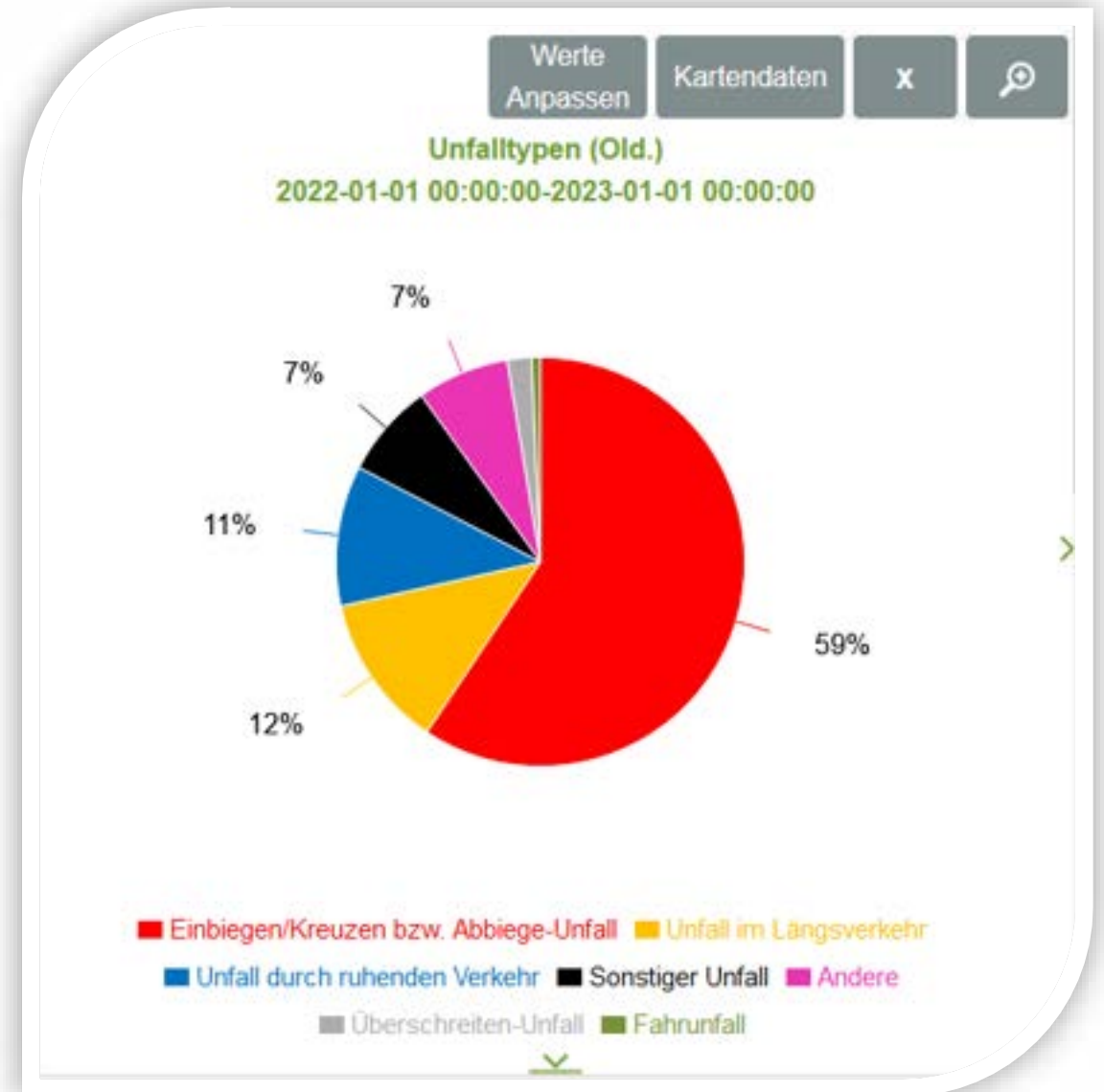
- Zeitverteilung der Meldungen
- Verteilung Meldungstypen
- Erkannte Probleme
- Dauer, bis Meldung behoben ist



KPI-Realisierung

Unfälle

- Zeitverteilung der Unfälle
- Verteilung Unfalltypen
- Unfälle nach Verursacher
- Beteiligte nach Zeit
- Unfallbeteiligte nach Unfalltyp
- Unfälle nach Oberfläche
- Unfälle nach Witterung



KPI-Realisierung

Kartendarstellungen: Kreuzungswerte und Straßeneinfärbungen



- Erschütterungen
- Erschütterungen unter Berücksichtigung der Geschwindigkeit
- Zeitverlust
- Anomalien



KPI-Realisierung

Kartendarstellungen: Heatmaps



- Anomalien
- Anomalien unter Berücksichtigung der Geschwindigkeit
- Vollbremsungen
- Absolute Geschwindigkeit
- Normalisierte Geschwindigkeit



KPI-Realisierung

Kartendarstellungen: Heatmaps



- Erschütterungen
- Erschütterungen unter Berücksichtigung der Geschwindigkeit
- Verkehrsstärke
- Wartezeiten

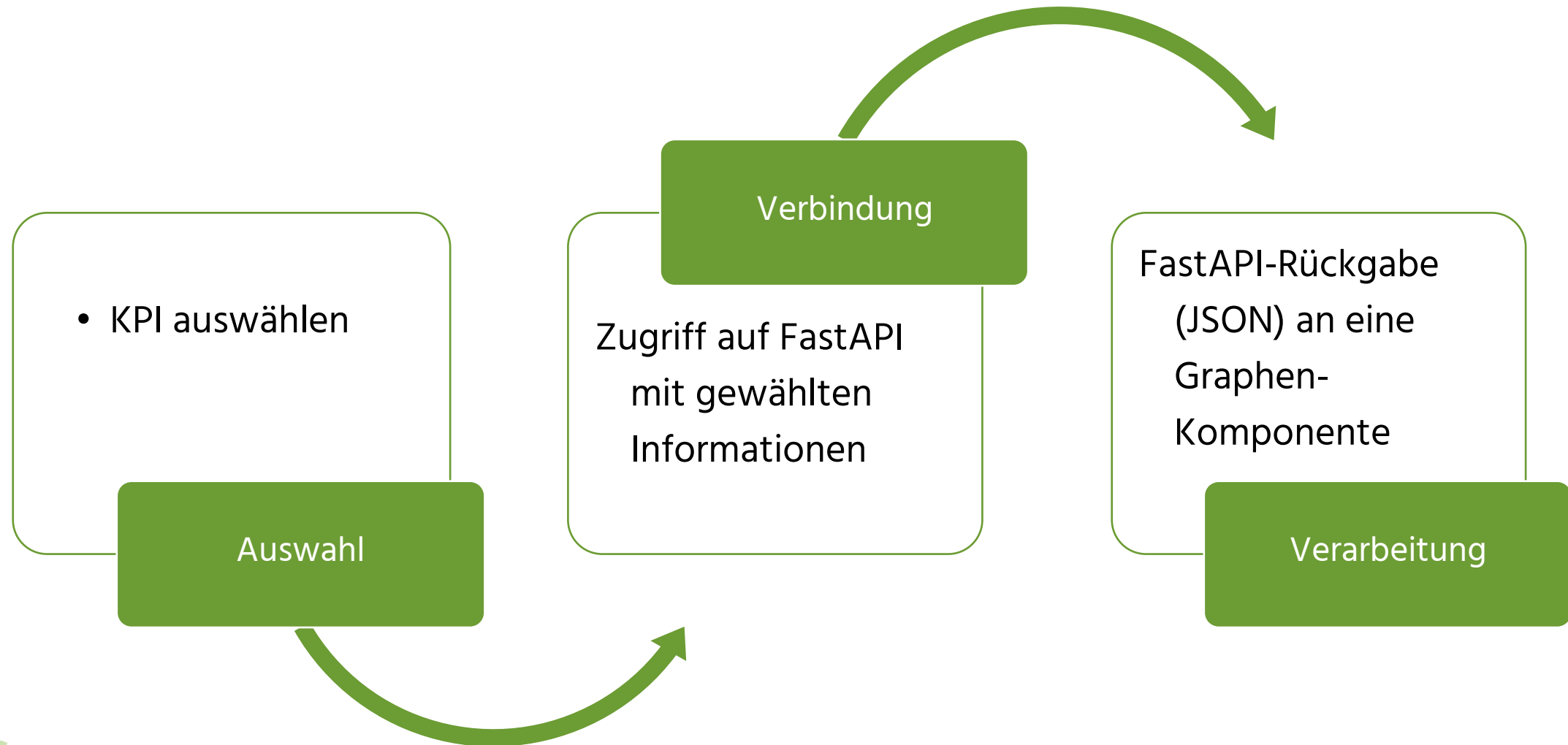
30 KPI

KPI-Realisierung

Kartendarstellungen: Zusätzliches

- Anzeigen der Unfalldaten und Meldungsdaten
- Anzeigen der Fahrrad- und Verkehrszählstellen inklusive der Daten
- Buslinien, Haltestellen
- Filterung für alle Daten durch den Nutzer möglich

Datenanbindung: FastAPI





Visualisierung

Visualisierung

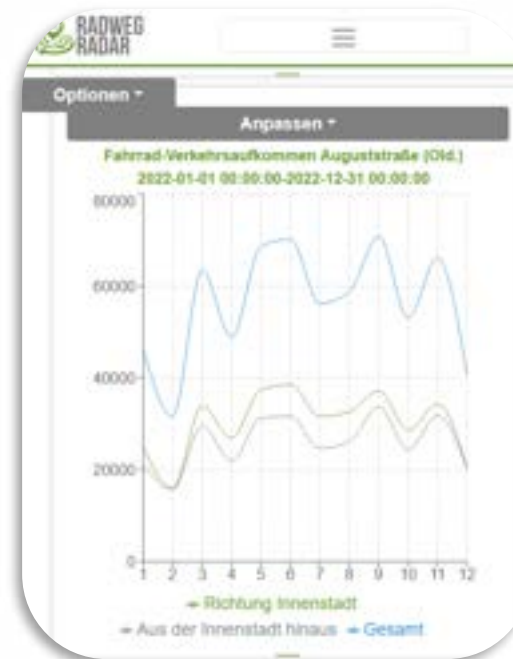
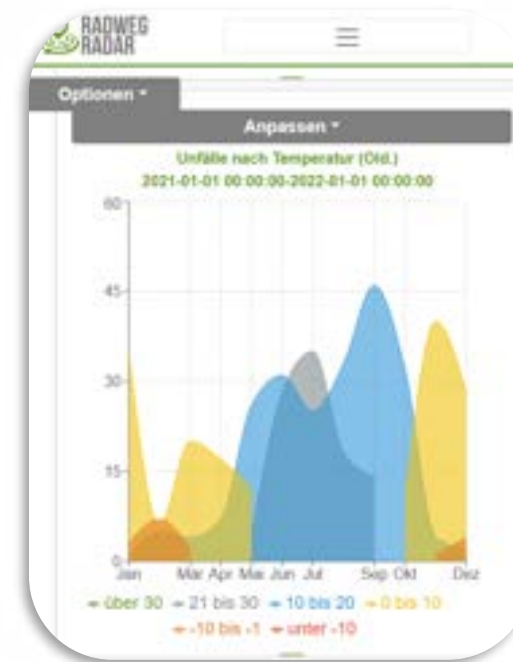
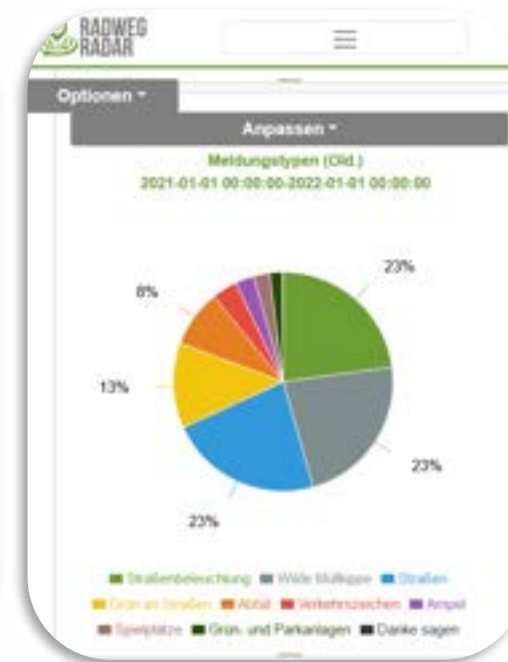
Box Layout

- Dynamisch anpassbare Boxen
- Responsives Boxlayout
- Graphen je nach Bildschirmgröße
- Box hinzufügen/entfernen
- Fullscreen
- Größe individuell
- Sicht wechseln zwischen
 - Karte
 - Kennzahlen
 - Score / Ranking



Visualisierung Kennzahlen

- Verschiedene Darstellungsarten
 - Tortendiagramm
 - Liniendiagramm
 - Flächendiagramm / Gestapelt
 - Balkendiagramm / Gestapelt
- Anpassbar in eigenem Modal
 - Stadt
 - Kennzahlauswahl
 - Zeitraum
- Titel zur Orientierung
 - Kennzahl
 - Zeitraum
 - Stadt



Visualisierung

Karte

- Verschiedene Darstellungsarten
 - Polylines
 - Marker(icons)
 - Heatmaps
 - Kreuzungen
- Layer (übereinanderlegbar)
 - Straßenscore
 - Kreuzungswerte (BIQEmonitor)
 - Straßeneinfärbungen (BIQEmonitor)
 - Heatmaps (BIQEmonitor)
 - Buslinien mit Haltestellen
 - Rad- Zählstellen
 - Auto-Zählstellen
 - Unfalldaten
 - Meldungen
 - KI-Modell Daten



Visualisierung

Karte

- Verschiedene Darstellungsarten
 - Polylines
 - Marker(icons)
 - Heatmaps
 - Kreuzungen
- Layer (übereinanderlegbar)
 - Straßenscore
 - Kreuzungswerte (BIQEmonitor)
 - Straßeneinfärbungen (BIQEmonitor)
 - Heatmaps (BIQEmonitor)
 - Buslinien mit Haltestellen
 - Rad- Zählstellen
 - Auto-Zählstellen
 - Unfalldaten
 - Meldungen
 - KI-Modell Daten



Visualisierung

Kartenfilter

- 3 Kategorien
 - Zeit
 - Filter nach Kennzahlkategorie
 - Bus
- Möglichkeiten
 - Kontexte bestimmter Zusammenhänge erschließen
 - Kontexte auf Zeiträume und spezielle Events zurückführen
 - Kontext mit ÖPNV herstellen
 - Freitextfiltern bei z.B.
 - Schadenshöhe und
 - Anz. Verletzter

Ansicht filtern

Zeitfilter

Filter

Allgemein

Schadenshöhe in €

Verursacher

Oberfläche

Unfalltyp

Straßenscore

Knotenwerte

Straßeneinfärbungen

Heatmaps

Wetter

Meldungen

Buslinien und Haltestellen

Ansicht filtern

Zeitfilter

Von: 03.11.2022 00:00

Bis: 01.02.2023 23:59

Tagesfilter:

Letzte 7 Tage Letzte 14 Tage Letzte 30 Tage

Monats-/Jahresfilter:

Letzter voller Monat Letztes volles Jahr

Filter

Buslinien und Haltestellen

Buslinien und Haltestellen

Haltestellen (ein) Alle aus/an

✓ 10 ✓ 12 ✓ 13

✓ 14 ✓ 15 ✓ 151

✓ 152 ✓ 16 ✓ 160

Visualisierung

Score/Ranking

- Verschiedene Darstellungsarten
 - Straßenscore als Radar Chart mit 8 gewählten Kategorien
 - Listendarstellung der Top/Flop 5 der jeweiligen Kategorie
- Merkmale
 - Freie Straßenauswahl aus den erfassten Straßen
 - Vergleich möglich
 - Intelligente Auswahl macht Vorschläge
 - Farbige Icons geben Aufschluss über Einordnung der Kennzahl



Wähle Deine Straße

Pferdemarkt X

- Pferdemarkt

nad X

- Grenadierweg
- Hafenpromenade
- Nadorster Straße

Übernehmen

radweg-radar.de

Live Demo

<https://radweg-radar.de/>





Rollenkonzept

Potentielle Nutzer und Rollen

- Stadtverwaltung, Verkehrsbehörde
- Fachexperten
- Fahrradfahrer

2 Rollen:

- Normaler Nutzer
- Experte





Erweiterungsmöglichkeiten

Login-Konzept

Keycloak

- Flexibilität: Integration mit verschiedenen Benutzerdatenbanken und -diensten
- Skalierbarkeit: Geeignet für kleine bis große Anwendungen und Dienste
- Sicherheit: Ständige Updates und Patches, um die Sicherheit zu gewährleisten



[Kc23, 14]

Login-Konzept

Keycloak

- Anpassbarkeit
- Multifaktor-Authentifizierung
- Open-Source-Tool
- Keine Lizenzkosten



Login-Konzept

- Experten können sich einloggen
- Gewinnen somit tieferen Einblick, z.B. bezüglich detaillierten Unfalldaten



The screenshot shows the registration page of the RADWEG-RADAR system. The page title is "RADWEG-RADAR" and the sub-header is "Registrierung". There is a language selector "Deutsch v" in the top right corner. The form contains the following fields: "Vorname", "Nachname", "E-Mail", "Passwort", and "Passwort bestätigen". A blue button labeled "Registrieren" is at the bottom right. A link "zur Anmeldung" is visible at the bottom left.



The screenshot shows the login page of the RADWEG-RADAR system. The page title is "RADWEG-RADAR" and the sub-header is "Bei Ihrem Konto anmelden". There is a language selector "Deutsch v" in the top right corner. The form contains the following fields: "E-Mail" and "Passwort". There is a checkbox labeled "Angemeldet bleiben" and a link "Passwort vergessen?". A blue button labeled "Anmelden" is at the bottom. A link "Neuer Benutzer? Registrieren" is at the bottom right.



Erweiterungsmöglichkeiten

- Nutzerprofile erweitern
 - Weitere Städte
 - Weitere Kennzahlkombinationen
 - Weitere Vertiefung von Infrastrukturdaten
 - Export von Daten und Diagrammen
 - Umsetzung weiterer Diagrammtypen
 - Klare Trennung von Website
- Komponenten unabhängig voneinander entwickelbar und wartbar
- nachhaltiger und flexibler Entwicklungsprozess in der Zukunft





Vielen Dank für Ihre Aufmerksamkeit!

Wir freuen uns auf Ihre Fragen.

pg@radweg-radar.de

Quellenverzeichnis

Bilder

1. https://de.wikipedia.org/wiki/PostgreSQL#/media/Datei:Postgresql_elephant.svg
2. https://en.wikipedia.org/wiki/Python_%28programming_language%29#/media/File:Python-logo-notext.svg
3. <https://en.wikipedia.org/wiki/PostGIS>
4. <https://fastapi.tiangolo.com/>
5. <https://www.openstreetmap.org/#map=17/53.14593/8.18101>
6. <https://pixabay.com/vectors/sun-cloud-rain-icon-summer-sky-3000986/>
7. <https://infrasense.de/projektpartner>
8. <https://seeklogo.com/images/W/wordpress-logo-FC322694E8-seeklogo.com.png>
9. <https://bilqinc.com/editorFiles/3147aa77.png>
10. https://de.wikipedia.org/wiki/Node.js#/media/Datei:Node.js_logo.svg
11. <https://de.m.wikipedia.org/wiki/Datei:Git-logo.svg>
12. <https://react-leaflet.js.org/img/logo-title.svg>
13. https://de.m.wikipedia.org/wiki/Datei:Leaflet_logo.svg
14. <https://pretius.com/wp-content/uploads/2023/02/Keycloak.png>



Quellenverzeichnis

Literatur

- [Pos23] <https://www.postgresql.org/>
- [Py23] <https://www.python.org/>
- [Fas23] <https://fastapi.tiangolo.com/>
- [Ppg23] <https://postgis.net/>
- [KR13] Ralph Kimball and Margy Ross. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. Wiley, Indianapolis, IN, 3 edition, 2013.
- [Wik23] https://de.wikipedia.org/wiki/Euklidischer_Abstand
- [Wp23] <https://wordpress.com/de/>
- [Re23] <https://react.dev/>
- [No23] <https://nodejs.org/de>
- [Git23] <https://git-scm.com/>
- [RI23] <https://react-leaflet.js.org/>
- [Le23] <https://leafletjs.com/>
- [Rch23] <https://recharts.org/en-US/>
- [Kc23] <https://www.keycloak.org/>

