



Intelligent Pendeln

# Intelligent Pendeln - IIP

## Zwischenbericht AG CW, JS

Autor:innen:

Tim Seidel	t.seidel@hs-osnabrueck.de
Marco Schaarschmidt	m.schaarschmidt@hs-osnabrueck.de
Axel Schaffland	a.schaffland@hs-osnabrueck.de
Max Bröker	max.broeker@hs-osnabrueck.de
Clemens Westerkamp	c.westerkamp@hs-osnabrueck.de



# I Inhaltsverzeichnis

<b>I</b>	<b>INHALTSVERZEICHNIS</b> .....	<b>II</b>
<b>II</b>	<b>ABBILDUNGSVERZEICHNIS</b> .....	<b>III</b>
<b>III</b>	<b>ABKÜRZUNGSVERZEICHNIS</b> .....	<b>IV</b>
<b>1</b>	<b>EINLEITUNG</b> .....	<b>1</b>
1.1	ÜBERBLICK.....	1
1.2	ANWENDUNGSFALL 1 – AUTO/P+R/BUS .....	3
1.3	WEITERE ANWENDUNGSFÄLLE.....	4
<b>2</b>	<b>IIP-SYSTEMLANDSCHAFT</b> .....	<b>5</b>
<b>3</b>	<b>PLATTFORMUNTERSUCHUNGEN</b> .....	<b>10</b>
3.1	NORMEN.....	10
3.1.1	<i>DIN SPEC 91357</i> .....	10
3.1.2	<i>DIN SPEC 91367</i> .....	15
3.2	PLATTFORMEN ANDERER STÄDTE .....	17
3.2.1	<i>Hamburg</i> .....	18
3.2.2	<i>Darmstadt</i> .....	20
3.2.3	<i>Paderborn</i> .....	23
3.3	VERGLEICH.....	24
3.4	ANDERE PLATTFORMANBIETER.....	25
<b>4</b>	<b>SCHNITTSTELLEN</b> .....	<b>26</b>
4.1	FROST-SERVER.....	26
4.2	VERKEHRSRECHNER .....	28
<b>5</b>	<b>VERKEHRSPROGNOSE</b> .....	<b>32</b>
5.1	MIKROSKOPISCH .....	32
5.2	MAKROSKOPISCH .....	32
<b>6</b>	<b>WETTERDATENQUELLEN</b> .....	<b>34</b>
6.1	ANFORDERUNGEN AN WETTERBEZOGENE QUELLEN IN IIP .....	34
6.2	DEUTSCHER WETTERDIENST .....	34
6.3	BRIGHT SKY.....	35
6.4	AWS WEATHER FORECAST .....	35
6.5	METEOMATICS .....	36
6.6	WETTER.COM.....	37
6.7	OPEN WEATHER MAP.....	38
<b>7</b>	<b>ARCHITEKTURENTWURF</b> .....	<b>41</b>
7.1	SYSTEMLANDSCHAFT.....	41
7.2	ANWENDUNGSFALLORIENTIERTE KONZEPTION .....	44
7.2.1	<i>Qualitätsziele</i> .....	45
7.2.2	<i>Randbedingungen</i> .....	46
7.2.3	<i>Stakeholder und Kontexte – Horizontale Capability Layer</i> .....	47
7.2.4	<i>Lösungsstrategien</i> .....	49
7.2.5	<i>Komponentensicht</i> .....	51
7.3	ALLGEMEINE ARCHITEKTURÜBERSICHT.....	57



## II Abbildungsverzeichnis

ABBILDUNG 1 - ANWENDUNGSFALLDIAGRAMM PENDLER & IIP-SERVER .....	4
ABBILDUNG 2 - AKTIVITÄTSDIAGRAMM PENDLER & IIP-SERVER .....	4
ABBILDUNG 3 - DIE SYSTEMLANDSCHAFT IM IIP-PROJEKT .....	5
ABBILDUNG 4 - DIE SYSTEMLANDSCHAFT IM IIP-PROJEKT - SCHWERPUNKT VERKEHRS- UND UMWELTSENSORIK.....	6
ABBILDUNG 5 - DIE SYSTEMLANDSCHAFT IM IIP-PROJEKT - SCHWERPUNKT PENDEL-APP .....	7
ABBILDUNG 6 - DIE SYSTEMLANDSCHAFT IM IIP-PROJEKT - SCHWERPUNKT (MOBILE) UMWELTSENSORIK.....	8
ABBILDUNG 7 - DIE SYSTEMLANDSCHAFT IM IIP-PROJEKT - SCHWERPUNKT METADATENMANAGEMENT .....	9
ABBILDUNG 8 - DARSTELLUNG EINER EINZELNEN OPEN URBAN PLATFORM (AUS [DI17]) .....	11
ABBILDUNG 9 - SCHICHTEN EINER OPEN URBAN PLATFORM (AUS [DI17]) .....	12
ABBILDUNG 10 - ZUSAMMENARBEIT MEHRERER UOP (AUS [DI17]) .....	15
ABBILDUNG 11 - DARSTELLUNG EINER OPEN URBAN PLATFORM FÜR MOBILITÄTSDATENERFASSUNG (AUS [DI19]) .....	15
ABBILDUNG 12 - URBAN DATA PLATFORM DER STADT HAMBURG (AUS [MRS22]) .....	18
ABBILDUNG 13 - DATENPLATTFORM DER STADT DARMSTADT (AUS [HE21]) .....	20
ABBILDUNG 14 - SCREENSHOTS AUS DER APP "DARMSTADT IM HERZEN".....	21
ABBILDUNG 15 - UDP DARMSTADT ALS SYSTEM OF SYSTEMS (AUS [HE21]) .....	23
ABBILDUNG 16 - ZENTRALE OPEN DATA PLATTFORM DER STADT PADERBORN (AUS [PA19B]) .....	23
ABBILDUNG 17 - SENSORTHINGS API DATENSTRUKTUR ([OP21]).....	26
ABBILDUNG 18 - OCIT STRUKTUR UND KOMPONENTEN (AUS [LI23], [YU23]).....	29
ABBILDUNG 19 - OCIT-C DOKUMENTATION DES TRAFFICDATA_DETECTOR_CURRENTVALUE-OBJEKTS (AUS [OC18]).....	30
ABBILDUNG 20 - DIAGRAMM AMAZON FORECAST (AUS [AM23A]) .....	36
ABBILDUNG 21 - INNOVATIVE MEASUREMENT TECHNOLOGY WITH WEATHER DRONES (AUS [ME23B]) .....	37
ABBILDUNG 22 - SYSTEMLANDSCHAFT DES IIP-PROJEKTS.....	41
ABBILDUNG 23 - ABLAUF DES UMWELTSENSITIVEN VERKEHRSMANAGEMENTS (NACH [OS23]) .....	42
ABBILDUNG 24 - ABLAUF DES ANWENDUNGSFALLS BIS ZUR PENDLERBENACHRICHTIGUNG.....	44
ABBILDUNG 25 - ABLAUF DES ANWENDUNGSFALLS WÄHREND DES PENDELNS.....	44
ABBILDUNG 26 - DATENQUELLEN IMPLEMENTIERUNGSPHASEN (AUS [RE23B]) .....	45
ABBILDUNG 27 - KOMPONENTENANSICHT DER ARCHITEKTUR FÜR DEN BEISPIELANWENDUNGSFALL .....	51
ABBILDUNG 28 - GESAMTARCHITEKTURÜBERSICHT .....	58



### III Abkürzungsverzeichnis

API	Application Programming Interface
AWS	Amazon Web Services
DWD	Deutsche Wetterdienst
FROST	Fraunhofer Open Source SensorThings API Server
GNN	Graph Neural Network
GRU	Gated Recurrent Units
GTFS	General Transit Feed Specification
HTTP(S)	Hypertext Transport Protocol (Secure)
IIP	Intelligenter Intermodaler Pendlerverkehr
IoT	Internet of Things
LSA	Lichtsignalanlage
LSTM	Long Short Term Memory
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
MWE	Minimal Working Example
NN	Neural Network
OCIT	Open Communication Interface for Traffic
ÖPNV	Öffentlicher Personennahverkehr
OUP	Offene Urbane Plattform
REST	Representational State Transfer
StOS	Stadtwerke Osnabrück
UDP	Urbane Datenplattform
XML	Extensible Markup Language
WFS	Web Feature Service
WMS	Web Map Service

# 1 Einleitung

## 1.1 Überblick

Dieser Bericht fasst den Arbeitsstand der AGs Westerkamp und Schöning der Hochschule Osnabrück im IIP-Projekt zusammen. Er gibt einen Überblick über Anwendungsfälle, die durchgeführte Plattformuntersuchungen, vorhandene Schnittstellen im IIP sowie Wetterdatenquellen. Zudem wird ein erster Architekturentwurf für den IIP-Server vorgestellt, dessen Teilkomponenten ausführlich beschrieben werden. Die Inhalte basieren insbesondere auf den Berichten [Sc23], [Br23], [Se23], deren (Teil-)ergebnisse hier zusammengefasst werden. Es ist geplant und erwünscht, den Bericht durch Inputs anderer Projektpartner zu ergänzen

Als Ansatz für den Architekturentwurf wurden zunächst einige existierende Plattformen und Normen betrachtet, um bereits eingesetzte Komponenten, Schnittstellen und Technologien in diesem Bereich zu evaluieren. Die Recherche umfasste dabei die DIN SPECS 91357 und 91367, sowie die Städte Hamburg, Darmstadt und Paderborn. Eine Auflistung weiterer Plattformen und Anbieter ist in Kapitel 3.2 und 3.4 zu finden.

Wichtige Technologien zum Zugriff und der Arbeit mit den vorliegenden Daten und Datenquellen sind der FROST-Server, als Kern der UDP, sowie der Verkehrsserver der Stadt Osnabrück, der über OCIT-C kommuniziert. Hierüber wird in Kapitel 4 ein Überblick gegeben.

Die Ansätze für die Verkehrsprognose auf makroskopischer und mikroskopischer Ebene sind in Kapitel 5 beschrieben.

In Kapitel 6 wird ein Überblick über mögliche Wetterdatenquellen von Wetterservices gegeben.

Kapitel 7 beschäftigt sich mit dem ersten Architekturentwurf. Zunächst werden einige Ziele, Randbedingungen und technologische Lösungsansätze erarbeitet. Anhand des vorgeschlagenen Vorgehens aus den DIN SPECS werden die einzelnen Stakeholder, Anforderungen und Technologien untersucht. Daraus werden einzelne Komponenten identifiziert und beschrieben, deren Interaktion im Rahmen des Anwendungsfalls dargestellt wird. Schließlich werden diese Komponenten in einem Gesamtarchitekturdiagramm, der sich an der Struktur der Normen und Architekturdiagrammen der vorgestellten Städte orientiert, visualisiert. Dies soll einen ersten Überblick über die Struktur und Komponenten des IIP-Servers bieten. In den nächsten Schritten kann diese Architektur erweitert und genauer spezifiziert werden, sodass erste Komponenten konkret definiert und in einem Prototyp umgesetzt werden können. Dies kann sowohl auf Basis der Anwendungsfälle als auch im Rahmen des MWEs erfolgen.

Ansprechpartner für die einzelnen Kapitel sind:



<b>Nr.</b>	<b>Kapitelname</b>	<b>Ansprechpartner</b>
1	Einleitung	alle
2	Systemlandschaft	Die jeweiligen an Komponenten und Systemen beteiligten Projektpartner
3	Plattformuntersuchung	Tim Seidel Marco Schaarschmidt
4	Schnittstellen	Tim Seidel Marco Schaarschmidt
5	Verkehrsprognose	Axel Schaffland
6	Wetterdienste	Clemens Westerkamp Max Bröker
7	Architekturentwurf	Tim Seidel Marco Schaarschmidt

## 1.2 Anwendungsfall 1 – Auto/P+R/Bus

Der erste Anwendungsfall aus der GVB besitzt folgenden Ablauf:

*„Herr Keller ist Mitarbeiter der Sparkasse Osnabrück und pendelt aus einer Vorortgemeinde in die Osnabrücker Innenstadt. Wichtig ist ihm pünktlich um 8 Uhr an seinem Arbeitsplatz zu sitzen.“*

**Situation heute:** *Er setzt sich in sein Fahrzeug und fährt um 7:30 Uhr los, um gegen 8:00 Uhr seinen Arbeitsplatz in Osnabrück zu erreichen. Seine Abfahrtszeit richtet sich dabei nach seiner Erfahrung.*

**Situation zukünftig:** *Am Morgen meldet sich automatisch die Pendler-App. Die App schlägt ihm vor um 7:20 das Haus zu verlassen, damit er garantiert zu 8:00 Uhr seinen Arbeitsplatz erreicht. Da es eine neue Baustelle auf seinem Pendlerweg gibt, schlägt ihm das System einen multimodalen P+R-Parkplatz am Stadtrand vor. Die prognostizierte Verkehrssituation wird bei der Routenführung berücksichtigt, so dass ein situationsabhängiger P+R genutzt wird. Das System bewertet die prognostizierte Auslastung des ÖPNVs und optimiert dafür die Abfahrtszeit von Herrn Keller. So kann dieser sein Auto am P+R Parkplatz abstellen und mit Sicherheit in den nächsten, nicht überfüllten, Bus in Richtung Innenstadt einsteigen“ [IIP22].*

Der Fokus liegt explizit auf Pendlerstrecken und nicht auf einmaligem Routing. Die Interaktion mit der pendelnden Person wird dabei durch eine Pendler-App umgesetzt. In diesem Anwendungsfall besteht die Intermodalität darin, von einem Auto auf ein öffentliches Verkehrsmittel (Bus) umzusteigen. Die Sicherstellung der Pünktlichkeit der Pendelnden ist nicht das einzige Ziel des Projektes, bietet aber einen guten Startpunkt.

Abbildung 1 und Abbildung 2 visualisieren den Beispielanwendungsfall durch ein Anwendungsfall- und Aktivitätsdiagramm. Darin sind Akteure, Funktionen und Aktionen definiert.

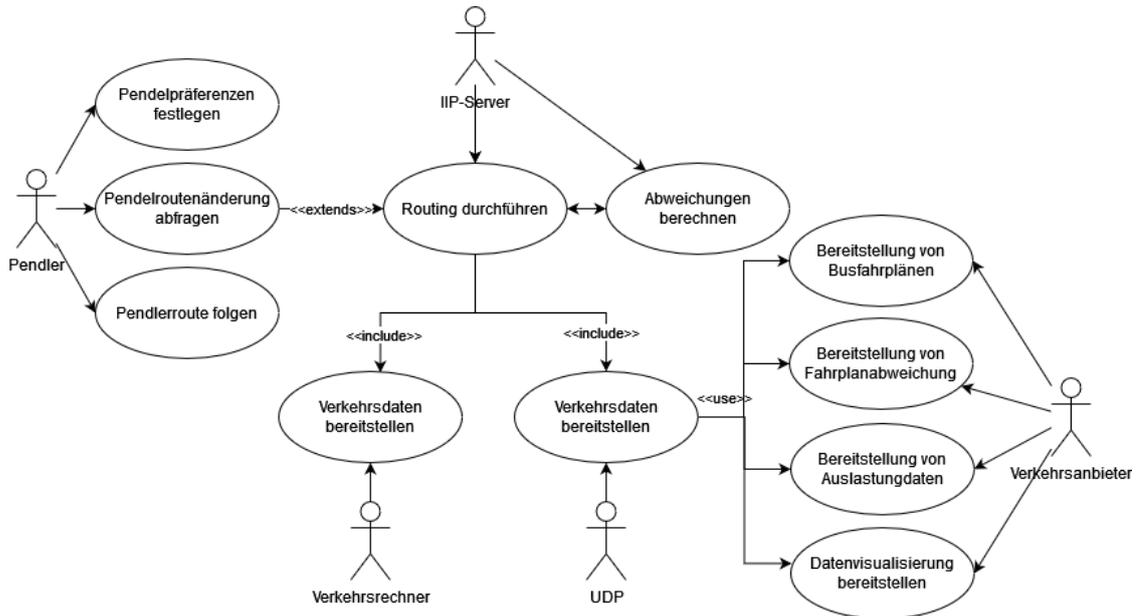


Abbildung 1 - Anwendungsfalldiagramm Pendler & IIP-Server

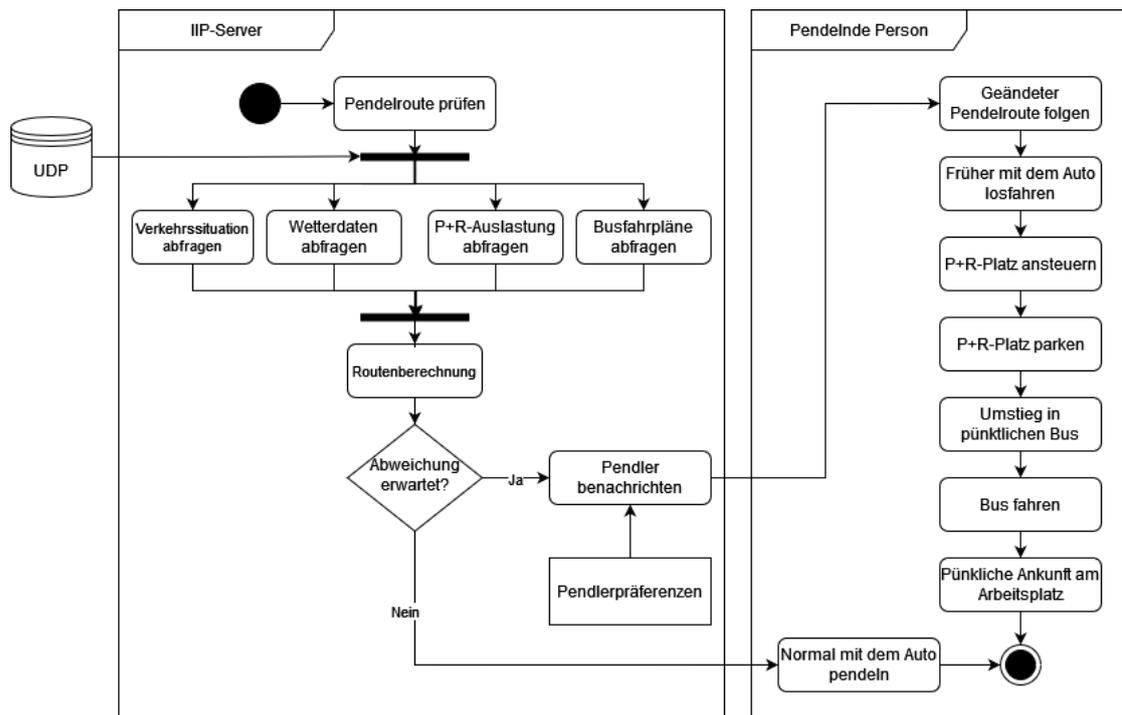


Abbildung 2 - Aktivitätsdiagramm Pendler & IIP-Server

### 1.3 Weitere Anwendungsfälle

Weitere Anwendungsfälle sollen sich im Dialog mit den Projektpartnern ergeben und hier beschrieben werden.



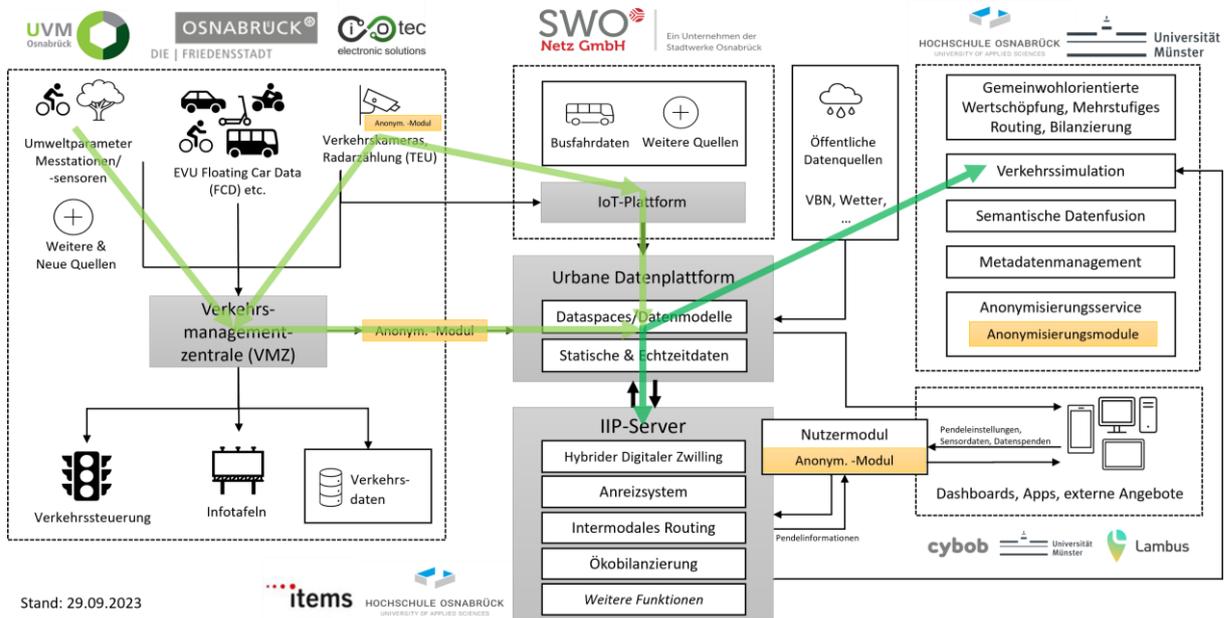


Abbildung 4 - Die Systemlandschaft im IIP-Projekt - Schwerpunkt Verkehrs- und Umweltsensorik

Die in Abbildung 4 eingezeichneten Pfeile beschreiben den Workflow der Verkehrs- und Umweltsensorik. Die Daten der Umweltparametermesstationen und Verkehrssensorik (beispielsweise durch Traffic Eye Universals (TEUs), Kameras, Leiterschleifen) werden in die Verkehrsmanagementzentrale (VMZ) der Stadt Osnabrück eingespeist. Von dort aus können sie in die zu entwickelnde Urbane Datenplattform, gegebenenfalls datenschutzkonform anonymisiert, überführt und im Rahmen des Projekts verwendet und ausgewertet werden. Weitere Verkehrskameras des Projektpartners iotec sind oder werden im Rahmen des Projektes im Stadtgebiet installiert. Neben einer Überführung in die VMZ werden diese zusätzlich über die IoT-Plattform der SWON in die Urbane Datenplattform übertragen. Im Fall der iotec-Kameras erfolgt eine Anonymisierung der erfassten Daten direkt auf den Kameras. Eine Weiterverarbeitung der Daten erfolgt beispielweise in den Modulen Verkehrssimulation oder der Auswahl geeigneter verkehrssituationsabhängiger Routen im Rahmen des intermodalen Routings.

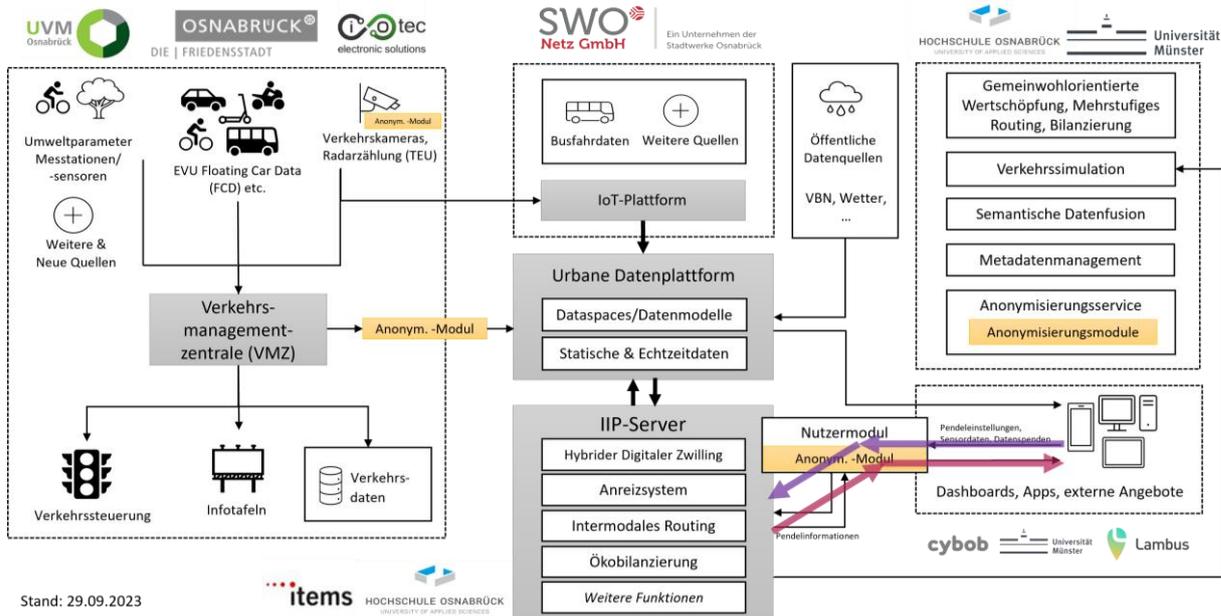


Abbildung 5 - Die Systemlandschaft im IIP-Projekt - Schwerpunkt Pendel-App

Die in Abbildung 5 eingezeichneten Pfeile beschreiben den Workflow der Pendel-App. Zum einen visualisiert der obere Pfeil den Informationsfluss der App zu den Komponenten des IIP-Servers. Darin werden unter anderem Pendelrouteneinstellungen, wie bevorzugte Verkehrsmittel oder präferierte Routen übertragen, um daraus geeignete Pendelrouten berechnen zu können. Weiterhin können anonymisierte Sensordaten oder Datenspenden aufgenommen werden, um den aktuellen Status der Pendelroute oder Verkehrsmittel zu erfassen und ggf. dynamische Verbesserung vornehmen zu können.

Der untere Pfeil beschreibt den Informationsfluss vom IIP-Server zur App. Hierbei werden Informationen zu den Pendelrouten an den Appnutzenden übertragen. Dabei müssen die eingestellten Präferenzen berücksichtigt werden. Anhand der gesamten Datenlage der UDP und des IIP-Servers können beispielsweise Wetterinformationen, erwartete Verkehrsspitzen oder Verkehrsereignisse eingebunden werden, um den Pendelnden über Aktualisierung der Pendelroute zu benachrichtigen und verschiedene Anreize zu schaffen. Die Kommunikation zwischen den beiden System soll über eine geeignete, getrennte Nutzerverwaltung umgesetzt werden, sodass die Anonymität der Nutzerdaten zu jedem Zeitpunkt gewährleistet ist.

Eine genauere Beschreibung der Pendel-App und den zu entwickelnden Funktionen können den Dokumenten [IIP22], [Sc23], [Se23] und eines aktuell von cybob in Bearbeitung befindlichen Entwurfs entnommen werden.

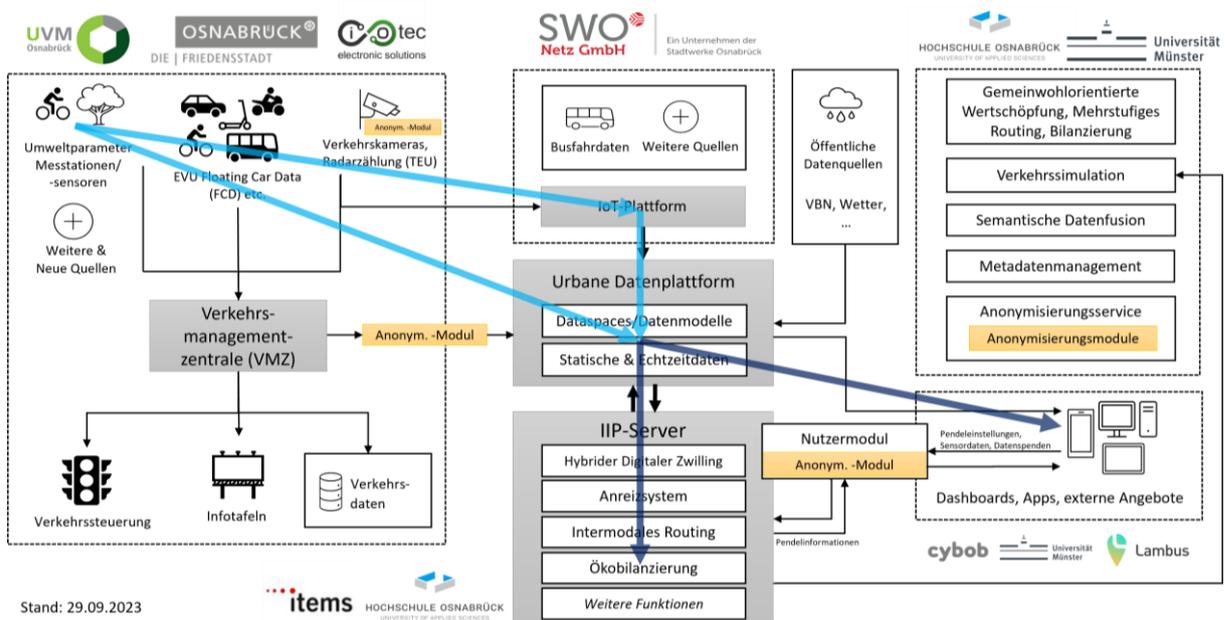


Abbildung 6 - Die Systemlandschaft im IIP-Projekt - Schwerpunkt (mobile) Umweltsensorik

Die in Abbildung 6 eingezeichneten Pfeile beschreiben den Workflow der mobilen Umweltsensorik. Neben der Umweltsensorik der Stadt Osnabrück, deren Daten in der VMZ abrufbar sind, liefern weitere Datenquellen im IIP-Projekt Umweltmesswerte, die ebenfalls betrachtet und erfasst werden sollen. Dabei handelt es sich insbesondere um die SenseBoxen der Universität Münster, die an Fahrrädern angebracht werden. Diese Messdaten werden allerdings nicht über die IoT-Plattform der Stadt in die UDP übertragen, sondern über OpenSenseMap verarbeitet.

Weiterhin werden im Rahmen des Projektes weitere stationäre Sensoren des Projektpartners iotec im Stadtbereich installiert. Diese werden über die IoT-Plattform der SWON in die UDP überführt. Von dort aus können sie beispielweise in Applikationen wie Dashboards sowie im zu entwickelnden Umweltbilanzierungsmodul im IIP-Server weiterverarbeitet und visualisiert werden.

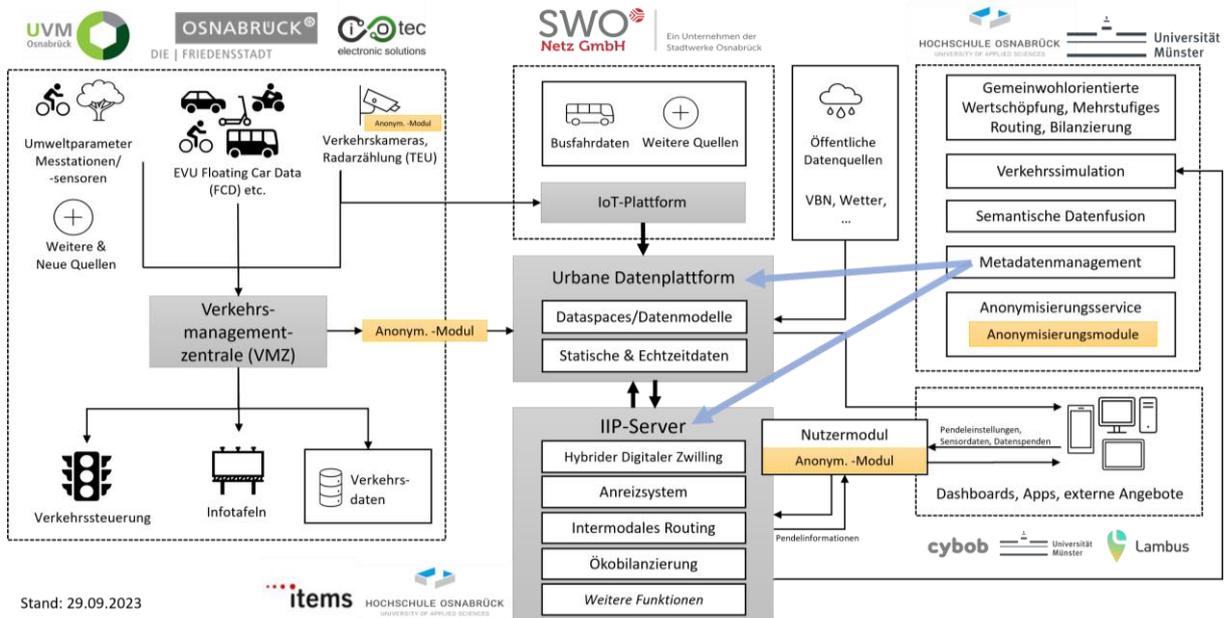


Abbildung 7 - Die Systemlandschaft im IIP-Projekt - Schwerpunkt Metadatenmanagement

Die in Abbildung 7 eingezeichneten Pfeile beschreiben den Einsatz eines Metadatenmanagements (siehe [Se23]). Dazu werden zunächst Datenmodelle festgelegt und anhand von Metadaten beschrieben. Dazu gehören beispielsweise die in der Urbanen Datenplattform abgelegten Daten, die im Rahmen der OGC SensorThings API als Schnittstelle bereitgestellt werden. Weiterhin werden die Daten der VMZ über OCIT-C abgerufen. Die IoT-Plattform mit der Visualisierung über das Masterportal arbeitet zudem mit WMS und WFS. Im Rahmen der Simulation kommen DATEX II und SUMO-Models zum Einsatz.

All diese Schnittstellen und Datenmodelle gilt es in einem einheitlichen System zu beschreiben und zu dokumentieren, wie beispielsweise REST-APIs, um eine Wiederverwendbarkeit sowie eine Möglichkeit zur einfachen Integration in das Gesamtsystem zu erreichen. Dabei gilt es nicht nur die Schnittstellen der einzelnen Komponenten gut und einheitlich zu dokumentieren. Gleichzeitig muss der Workflow und die Zusammenarbeit der einzelnen Komponenten des IIP-Servers sowie des Gesamtsystem des IIP-Projektes erfolgen.

## 3 Plattformuntersuchungen

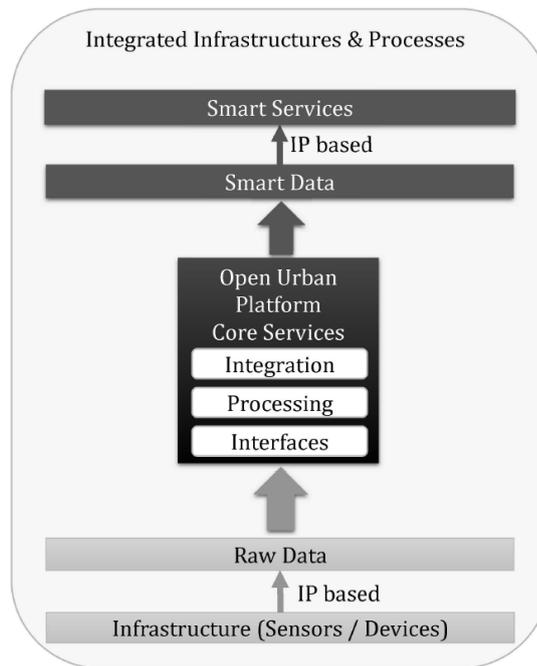
### 3.1 Normen

Im Umfeld der Urbanen Datenplattformen im Bereich der Mobilitätsuntersuchung existieren zwei besonders relevante DIN-Spezifikationen, auf die auch andere Projekte in diesem Bereich zurückgreifen. Dies sind DIN SPEC 91357:2017-12 - Referenzarchitekturmodell Offene Urbane Plattform (OUP) [Di17] sowie DIN SPEC 91367:2019-04 - Urbane Mobilitätsdatensammlung für Echtzeitapplikationen [Di19]. Nachfolgend werden diese deshalb vorgestellt. Hinzuweisen ist an dieser Stelle auch auf die DIN SPEC 91340:2016-10 – Terminologie der intelligenten individuellen urbanen Mobilität [Di16], die einige Begriffe im Rahmen dieses Verkehrsbereich definiert. Wichtige Definitionen sind ein Pendler als „Person, die für den über den Lokalbereich hinausgehenden Weg zwischen Wohn- und Arbeitsort ein Fahrzeug verwendet“ [Di16]. Pendelt diese intermodal ist die Definition der Intermodalität eine „Kombination verschiedener Verkehrsträger bei der Durchführung einer Verkehrsleistung“ [Di16], wobei der intermodale Verkehr ein Teil des multimodalen Verkehrs ist, der die „Nutzung verschiedener Verkehrsmittel für die Zurücklegung eines Weges“ [Di16] beschreibt.

#### 3.1.1 DIN SPEC 91357

Die DIN SPEC 91357:2017-12 - Referenzarchitekturmodell Offene Urbane Plattform (OUP) befasst sich mit Referenzarchitekturen für Datenplattformen. Diese Spezifikation zeigt einige Konzepte und Ansätze auf, die auch im IIP-Projekt wiederzufinden sind. Abbildung 8 zeigt die schematische Übersicht und den Datenfluss einer vorgeschlagenen alleinstehenden OUP-Architektur.

Die Basis bildet eine IoT-Infrastruktur, aus Sensorik und anderen Geräten (*Infrastructure*), die Messdaten liefern können. Diese werden über ein Netzwerk als Rohdaten weitergegeben (dies kann beispielsweise über (W)LAN, LoRaWAN oder Mobilfunknetze geschehen). Diese Rohdaten (*Raw Data*) werden den verschiedenen *Core Services* zugeführt, die als einzelne Untersysteme zu interpretieren sind. Diese bieten Schnittstellen zur Datenaufnahme an (*Interfaces*), können diese weiterverarbeiten (*Processing*) und schließlich zusammenführen und verschneiden (*Integration*), um daraus neue Informationen zu gewinnen (*Smart Data*). Diese Daten können wiederrum bereitgestellt und in weiterführenden Applikationen (*Smart Services*) weiterverwendet, aufbereitet und visualisiert werden, um daraus einen Mehrwert für Städte und Bürger zu gewinnen. Diese Komponenten lassen sich grob in drei Schichten einteilen.



**Abbildung 8 - Darstellung einer einzelnen Open Urban Platform (aus [Di17])**

Eine OUP ist dabei nicht als ein von Grund auf neu zu entwickelndes System zu verstehen. In den meisten Städten (wie auch in Osnabrück) sind bereits einige Systeme im Einsatz, die offene Daten sammeln oder verwalten. Beispielsweise die IoT-Plattform der Stadtwerke Osnabrück Netz GmbH [St23] oder das Geoinformationsportal der Stadt Osnabrück [Os23] vielmehr ist der Ansatz dieses Architekturmodells die bestehenden Systeme aufzugreifen, mit Schnittstellen in die UOP zu integrieren und die Daten wiederum mit offenen und standardisierten Schnittstellen aus der UOP heraus einheitlich bereitzustellen. Dies wird in der DIN SPEC als „System of Systems“ bezeichnet. So kann ein intelligentes Gesamtsystem entstehen, auf dessen Basis Anwendungen entwickelt und bereitgestellt werden. Ein wichtiger Aspekt ist dabei, zu jeder Zeit Datenschutzaspekte zu berücksichtigen. Auch wenn diese Plattform „open“, also offen zugänglich ist und offene Daten bereitstellt, ist dabei explizit nicht ausgeschlossen, dass alle darauf basierenden und arbeitenden Applikationen öffentlich verfügbar und kostenlos sein müssen. Das heißt, dass darauf auch kommerzielle Anwendungen von Unternehmen entwickelt und veröffentlicht werden können.

Die in Abbildung 8 gezeigte Grundstruktur kann durch weitere logische Layer repräsentiert werden, was in Abbildung 9 gezeigt wird. Diese betrachtet neben den Technologien und Funktionalitäten der Layer zusätzlich auch die Stakeholder. Diese Strukturierung eignet sich zur Unterstützung bei der Konzeption Urbaner Plattformen, aber auch bei der Erstellung von Use-

Case-Komponenten und wird daher nachfolgend beschrieben. Die Schichten besitzen Nummerierungen, die hier eingesetzt werden (nach [Di17]).

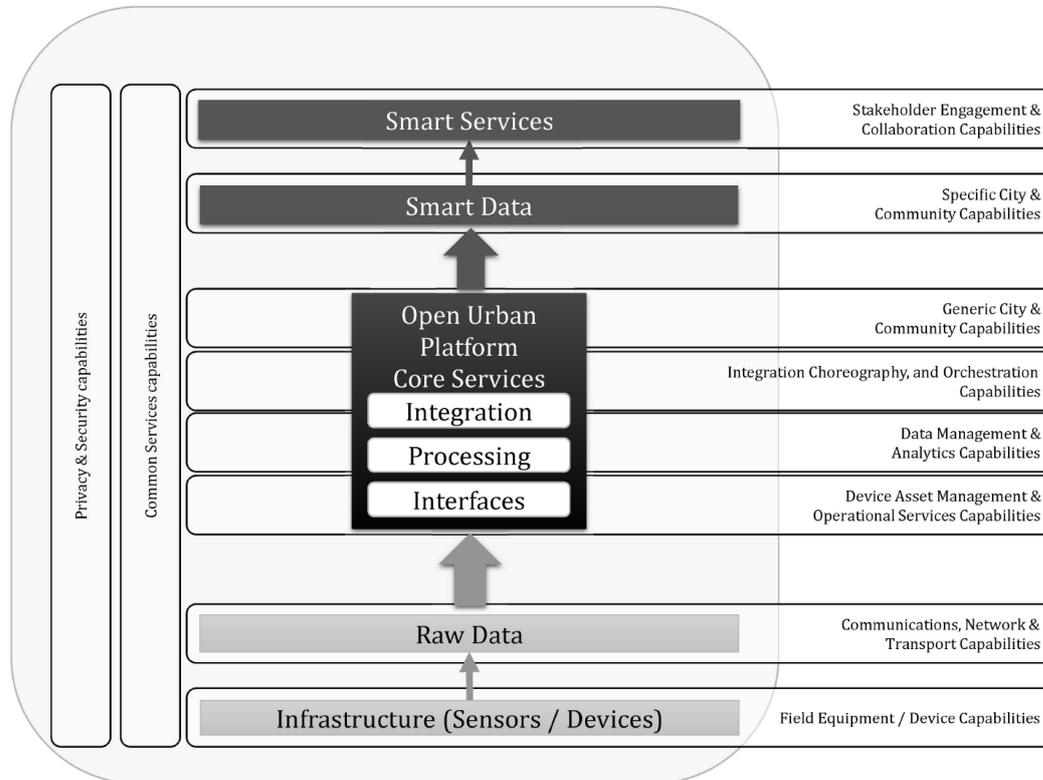


Abbildung 9 - Schichten einer Open Urban Platform (aus [Di17])

0. **Field Equipment / Device Capabilities.** Hierzu gehört die Bereitstellung der Sensorik und Geräte zum Erfassen der Messwerte und Steuern von Systemen. Sensoriken liegen je nach Anwendungsfall bereits vor oder müssen im Rahmen der Anforderungen neu entwickelt oder beschafft werden, um damit die benötigten Daten und geforderten Funktionalitäten abdecken zu können.
1. **Communications, Network & Transport Capabilities.** Zur Übertragung und zum Datenaustausch werden verschiedene Netzwerkinfrastrukturen benötigt. Hier gilt zu beachten, dass das Netzwerk die entsprechende Leistungsfähigkeit und Abdeckung besitzt, um die Sensorik ausreichend zu unterstützen. Beispielsweise kann die Datenmenge durch die Übertragung über LoRaWAN deutlich eingeschränkt sein, was für den jeweiligen Anwendungsfall berücksichtigt werden muss.
2. **Device Asset Management & Operational Service Capabilities.** Zur Verwaltung der Geräte und Netze ist eine Verwaltungsinstanz nötig, um weitere Sensorik in die Plattform einbinden und Netze skalieren zu können.

- 3. Data Management & Analytics Capabilities.** Die gemessenen und versendeten Daten müssen aufgenommen, gespeichert und verwaltet werden. Dazu gehört die Vorverarbeitung, Datenfusion und Aggregation für verschiedene Anwendungsfälle und Endanwender. Hierzu gehört auch das Anpassen der Daten an die erforderlichen Datenmodelle und die gemeinsamen Schnittstellen. Dies kann besonders durch ein Metadatenmanagement unterstützt werden. Durch verschiedene Arten von Daten (beispielsweise zeitbasierte, ortsbasierte oder allgemeinere Daten), können verschiedene Schnittstellen, Datenbanktypen oder Anpassungen erforderlich sein. Die Analyse kann durch die Bereitstellung von Dashboards und eine einfache Visualisierung unterstützt werden, unabhängig von den anhand der Daten entstehenden smarten Applikationen.
- 4. Integration Choreography and Orchestration Capabilities.** Hierbei handelt es sich die Funktionalitäten der Verwaltung der einzelnen Komponenten zur Kommunikation untereinander. Das beinhaltet die Integration der Komponenten durch Festlegung und Bereitstellung von einheitlichen Schnittstellen. Dazu gehören neben Datenschnittstellen inklusive deren öffentliche Beschreibung auch Messagingsysteme über Broker und andere Notification-Systeme. Durch die Anforderung der Skalierbarkeit, gehört auch das Load-Balancing dazu, sodass die Daten jederzeit für die Abnehmer und ihre Applikation verfügbar sind und eine Kommunikation zwischen den Subsystemen jederzeit möglich ist. Dieser Layer stellt den zentralen Layer des Gesamtsystems dar, da er die Grundlage für den Datenaustausch und somit die Bereitstellung der Datengrundlage für alle smarten Applikationen und Use-Cases dient.
- 5. Generic City & Community Capabilities.** Hierzu gehören die städtischen Verwaltungsstrukturen. Zur Entwicklung neuer Businessmodelle und Anwendungsfälle insbesondere zur Optimierung der Umwelt- und Verkehrssituation sind geeignete Prozesse zu definieren. Dazu gehört der Einsatz offener (auch nicht-technischer) Standards und gemeinsame Offenlegung der Daten, aber auch die interdisziplinäre Zusammenarbeit der unterschiedlichen Stakeholder auf Basis vorher festgelegter Ziele und Regularien, die zu erreichen und einzuhalten sind.
- 6. Specific City & Community Capabilities.** Auf Basis der Offenen Urbanen Plattformen können nun spezifische Anwendungen entwickelt werden, die städtische Infrastruktur, Verkehr und Umwelt verbessern und nachhaltiger gestalten können. Die geschieht auf der bisher aufgebauten Datenbasis durch Integrationsprozesse. Dazu gehört ebenfalls die Einhaltung der vorher definierten Standards, sodass die neuen Applikationen in die bestehende Plattform integriert werden kann und eine Einheitlichkeit besteht.
- 7. Stakeholder Engagement & Collaboration Capabilities.** Um einen tatsächlichen Mehrwert aus den entwickelten Prozessen und Anwendungen erzielen zu können ist eine Zusammenarbeit von Stakeholdern untereinander und ein Austausch mit Bürgern nötig. Nur so können Anwendungen erfolgreich umgesetzt und etabliert werden. Dies

geht einher mit einer bürgerzentrierten und zielorientierten Entwicklung, um letztendlich die gesetzten oder erforderlichen Umwelt- oder Verkehrsziele zu erreichen. Die relevanten Stakeholder sollten daher im Rahmen jeder Applikation klar definiert sein und die Zecke der Anwendungsfälle klar kommuniziert werden. Ansätze und Anreize für externe Dienstleistern müssen gegeben werden, damit eine Zusammenarbeit ermöglicht und die Entwicklung neuer Applikationen und Ideen gefördert wird.

8. **Privacy & Security Capabilities.** Über alle Layer hinweg erstrecken sich die *Privacy & Security Capabilities*, also die Sicherstellung der Anonymität der Daten, sowie ein entsprechender Schutz für die einzelnen Komponenten vor unerwünschtem Zugriff und Kompromittierung. Dies sind die Eigenschaften, die auch als wichtiges Projektziel und im Rahmen der Modulanforderungen wiederzufinden sind.
9. **Common Service Capabilities.** Schließlich runden die allgemeinen Services die Gesamtstruktur ab. Dazu gehört die Bereitstellung der (technischen) Infrastruktur und Services für die einzelnen Dienste. Außerdem beinhaltet dies die Kommunikation der Stakeholder mit den Bürgern und Kunden, um einzelne Applikationen bekannt zu machen und diese zu sensibilisieren.

Der Fokus offene und standardisierte Schnittstellen einzusetzen und Daten durch ein Metadatenmanagement zu beschreiben, ermöglicht das Konzept der Interaktion mehrerer verschiedener Offener Urbanen Datenplattform aus unterschiedlichen Städten und Regionen, wie in Abbildung 10 dargestellt ist. Durch den Austausch von Daten und Kommunikation der *Core Services* der jeweiligen Plattformen über die gemeinsamen Schnittstellen können Mehrwerte, wie beispielsweise ein übergreifendes Routing, entstehen. Verbreitete standardisierte Schnittstellen, sollten daher bei der Konzeption der Architekturen besonders berücksichtigt werden. Hierfür eignen sich HTTPS mit REST und MQTT als Protokolle, sowie (Geo-)JSON, XML oder CSV als Datenformate.

Neben der standardisierten und mehrwertorientierten Kommunikation großer vertikalen Komponenten in Form von unterschiedlichen Offenen Urbanen Plattformen wie in Abbildung 10 gezeigt ist, trifft dieses Konzept auch auf einem niedrigeren Level zwischen Subsystemen einer OUP zu. Dies ist auch Teil der DIN SPEC, wird aber noch genauer in der DIN SPEC 91367 mit dem Hintergrund der Urbanen Mobilität beschrieben, die im folgenden Kapitel vorgestellt wird.

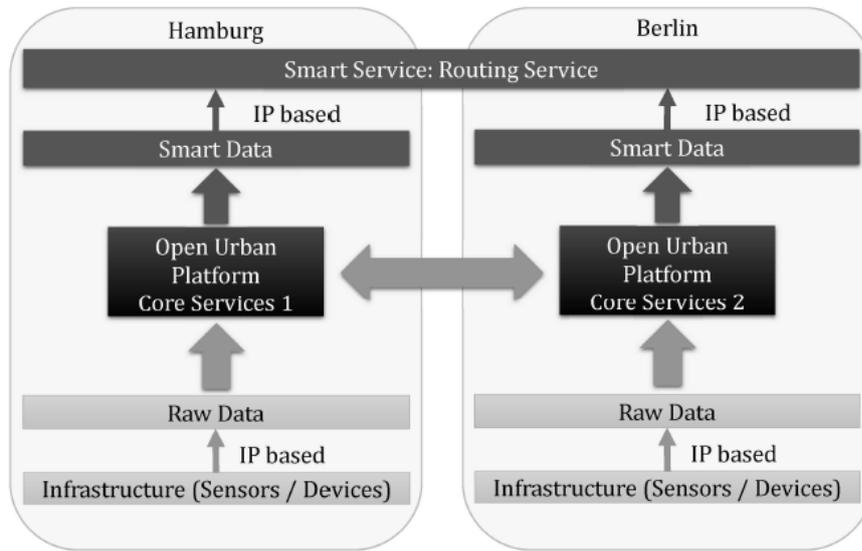


Abbildung 10 - Zusammenarbeit mehrerer UOP (aus [Di17])

### 3.1.2 DIN SPEC 91367

Die DIN SPEC 91367:2019-04 - Urbane Mobilitätsdatensammlung für Echtzeitapplikationen ist eine neuere Spezifikation und greift die Architektur und Ansätze der vorherigen DIN SPEC auf, um diese für den Anwendungsfall der echtzeitnahen Mobilitätsdatensammlung und den daraus abgeleiteten Applikationen, anzupassen und aufzubereiten.

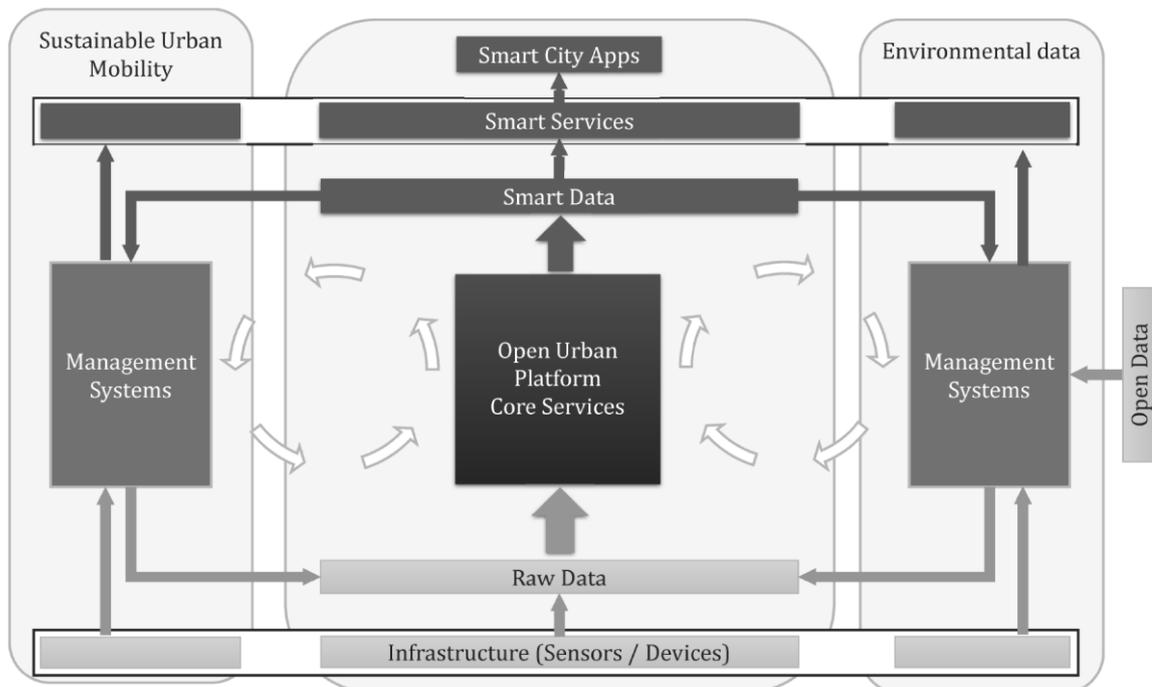


Abbildung 11 - Darstellung einer Open Urban Platform für Mobilitätsdatenerfassung (aus [Di19])

Die angepasste Architektur ist in Abbildung 11 gezeigt. Hier ist besonders die vertikale Trennung der drei herausgestellten Komponenten zu betrachten [Di19].

**Environmental data.** Dieses Teilsystem umfasst die Umwelt- und Geodaten, die in den einzelnen Städten in vielen Fällen bereits als offene Daten aufgenommen werden oder verfügbar sind und in dieser Form bereitgestellt werden. Sie dienen als wichtige Datengrundlage für Verkehrsplanung und -entscheidungen und liegen teilweise bereits in Geo-Portalen vor.

**Sustainable Urban Mobility (SUM).** Diese Komponente der in dieser DIN SPEC vorgeschlagenen Architektur, verfolgt explizit das Ziel, Verkehr effizienter zu gestalten und durch Multimodalität, zusammen mit neuen Businessmodellen, günstigeres oder umweltschonenderes Reisen und Pendeln zu ermöglichen [Di19]. In der Architektur vorgesehene Anwendungsfälle sind die Etablierung beziehungsweise Optimierung des öffentlichen Transports (beispielsweise on-demand Systeme), Laufen und Radfahren (Bike Sharing, Fahrradrouting), Intermodaler Transport, Verkehrssicherheit, Optimierung des Straßenverkehrs (Real-time traffic control), Smart Parking, Optimierung der Logistik, Mobilitätsmanagement in Bezug auf Arbeitgeberanreize und überregionale Mobilität.

**Kernplattform.** Im Kern dieser erweiterten Architektur befindet sich die aus der DIN SPEC 91357 vorgestellte Offene Urbane Plattform. An diese Plattform können die neuen Teilsysteme angebracht werden, indem sie durch bestehende Schnittstellen angebunden und integriert werden, sodass die neuen *Smart City Apps* entwickelt werden können. Neu ist hier insbesondere die Echtzeitanforderung für die OUP. Die vorgestellten Anwendungsfälle erfordern eine große Menge an Daten, die schnelle Änderungen und Aktualisierung erfahren. Dies sind beispielsweise der Status der Ampelschaltung, die Auslastung einer Kreuzung oder die Position eines Verkehrsmittels. Nur wenn die Auflösung dieser Daten fein genug ist, können die Anwendungsfälle umgesetzt werden. Dazu wurden einige Kernanforderungen definiert (nach [Di19]).

- Die Schnittstellen der Managementsysteme müssen Daten in Echtzeit aufnehmen können.
- Die Systeme müssen mit einer sehr geringen Ausfallrate, wie etwa einer Uptime von 99.9% laufen, da ein längerer Ausfall des Systems, die Steuerung des Verkehrs oder des Routings mit veralteten Daten nicht ausreichend möglich ist
- Die Anwendung müssen die Daten in Echtzeit konsumieren können und die benötigten Informationen rechtzeitig bereitstellen können
- Nicht nur die Schnittstellen, sondern auch die Infrastruktur muss die geforderten Echtzeitanforderungen unterstützen, das heißt die Netzwerkkommunikation muss ähnlich schnell ablaufen
- Die OUP muss auch mit anderen Datenplattformen kommunizieren können, um beispielsweise das Routing ausweiten zu können.

Zweck der Gesamtplattform ist insbesondere die Interoperabilität der verschiedenen Komponenten. So können die in der zentralen Plattform verarbeiteten Smarten Daten in allen Komponenten verwendet und bereitgestellt werden. Gewonnene Ergebnisse und weiteren Daten, die im Rahmen der Anwendungsfälle der Komponenten entstehen, können als neue Rohdaten zurückgeführt werden, um so das Gesamtsystem zu bereichern und neue Anwendungsfälle zu generieren und bestehende zu verbessern.

### 3.2 Plattformen anderer Städte

In vielen anderen Städten und Regionen befinden sich aktuell Plattformen im Aufbau. Nachfolgend sind einige Städte aufgeführt.

- Hamburg
- Darmstadt
- Paderborn
- Münster
- Osnabrück
- Leipzig
- Berlin
- Frankfurt
- Bad Hersfeld
- Ludwigshafen
- München
- Lübeck

Die Plattformen unterscheiden sich im Fortschrittsgrad, Tiefe, Unternehmenseinbindungen und Dokumentation. Während einige zunächst nur aus Geoportalen oder einfachen IoT-Dashboards bestehen, setzen einige andere Plattformen bereits auf Smarte Applikationen. Nachfolgend werden daher Überblicke über die Plattformen Hamburg, Darmstadt und Paderborn gegeben, die bereits verschiedene Module und Architekturen dokumentiert haben. Diese können für einen eigenen Architekturentwurf herangezogen werden.

### 3.2.1 Hamburg

Die Urban Data Platform der Stadt Hamburg (kurz UDP\_HH) befindet sich aktuell bereits im Einsatz und wird zudem weiterentwickelt. Kern des Open Data-Ansatzes ist ein Frost-Server, der die gemessenen und gesammelten IoT-Daten der Öffentlichkeit zur Verfügung stellt. Die gesamte Systemlandschaft kann der Abbildung 12 entnommen werden [MRS22].

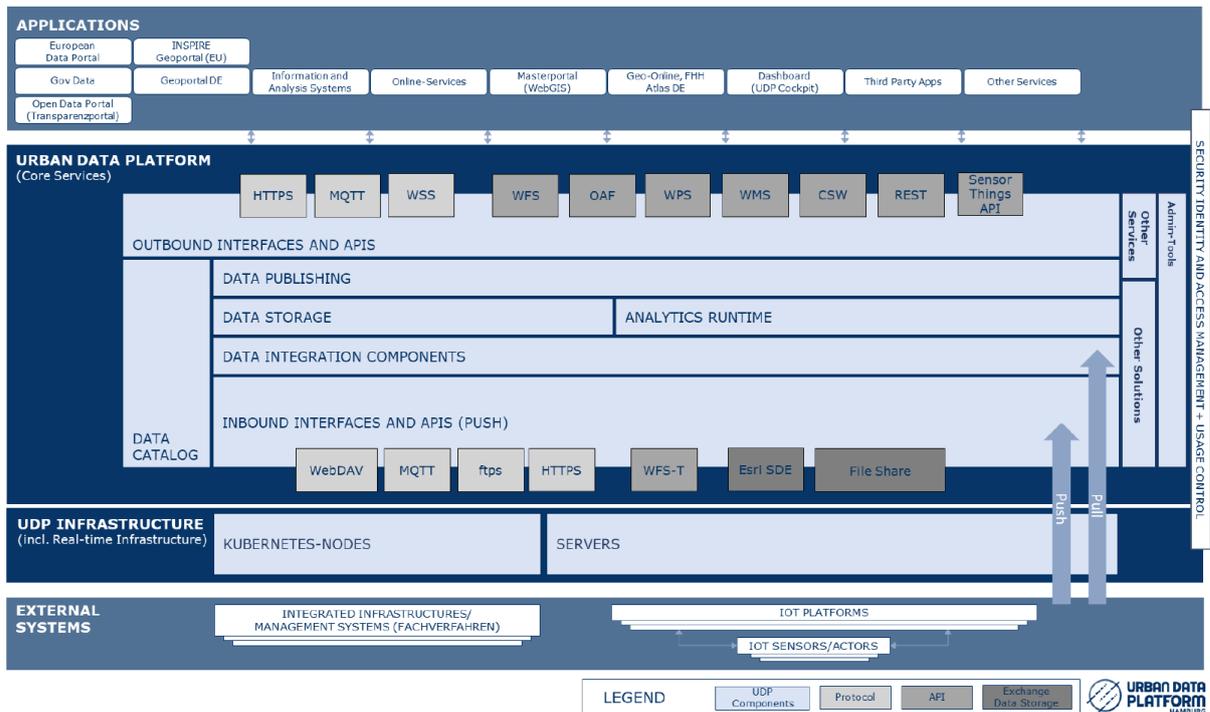


Abbildung 12 - Urban Data Platform der Stadt Hamburg (aus [MRS22])

Die grobe Struktur und Einteilung in die drei groben logischen Applikationsschicht (*Applications*), Urbane Datenplattform (*Urban Data Platform*), sowie die externen Systeme (*External Systems*), die die IOT-Plattformen einbezieht, kann aus den vorgestellten DIN SPECS wiedererkannt werden. Eine zusätzlich eingesetzte Schicht ist die „UDP Infrastructure“, welche die Serverinfrastruktur und die Bereitstellung der Module durch Kubernetes umfasst.

Bei den im Diagramm konkret genannten **Applikationen** handelt es sich insbesondere um Visualisierungstools, wie Metadatenkataloge, Geoinformationssysteme und Darstellung der Sensormesswerte. Applikationen, die andere, komplexere Anwendungsfälle abdecken, sind hier nicht konkret angegeben, bzw. durch die Punkte *Third Party Apps* und *Other Services* zusammengefasst.

In Bereich der **Urban Data Platform** werden alle Daten gespeichert (*Data Storage*), integriert (*Data Integration Components*) und analysiert (*Analytics Runtime*). Zu den Daten gehört ein *Data Catalog*, der über verschiedene Datenkataloge wie MetaVer verfügbar ist.

Über Inbound Interfaces werden die Daten eingespeist und über Outbound Interfaces verfügbar gemacht. Als wichtigste Protokolle zur Datenübertragung werden HTTPS und MQTT, die sowohl für eingehende und ausgehende Daten zum Einsatz kommen, verwendet. Zusätzlich können über WebDAV und ftps Dateien an die UDP übertragen werden. Um Daten aus der IoT-Plattform in die Urbane Datenplattform zu übertragen, wird zwischen Push- und Pull-Mechanismen unterschieden. Bei der Push-Methode werden die Daten ausgehend von der Sensorik, die IoT-Plattform oder andere Dienste an die Plattform geschickt und über Schnittstellen eingespeist. Bei der Pull-Methode werden die Datenquellen dagegen in Integrationskomponenten definiert und hinterlegt, sodass sie automatisiert angefragt und eingepflegt werden. Die Art der Übertragung ist je nach Datenquelle und Bereitstellungskomplexität unterschiedlich gewählt.

Auf der Ausgangsseite sind Websockets ein weiteres Protokoll, falls eine dauerhafte, effiziente Verbindung aufrechterhalten werden soll. Hervorzuhebende ausgehende Schnittstellen sind WFS, die zur Bereitstellung der Geodaten, beispielsweise in einem Kartendienst zur Visualisierung eingesetzt werden. Weiterhin ist hier auch die SensorThings API wiederzufinden, die zum Zugriff auf den FROST-Server bereitgestellt wird.

Insgesamt beruht die UDP\_HH neben statischen Geodaten im Kern auf den über **die IoT-Plattform** erhobenen Messwerten der verschiedenen **Sensoren** (*IoT Sensors*). Darunter befinden sich auch einige Kategorien, die für den Anwendungsfall der Pendlerverkehrsuntersuchung und des Verkehrsroutings relevant sind. Diese befinden sich alle auf dem FROST-Server der UDP\_HH und können entweder über den FROST-Server oder das Masterportal (Geo oder Verkehr) abgefragt werden.

Zum einen werden einige Daten über den Radverkehr erhoben. Dazu gehören die Daten einer permanenten Dauerzählstelle, die den Radverkehr in beide Richtungen erfasst. Diese werden täglich aktualisiert und beruhen auf stündlichen Intervallen, die abgefragt werden können.

Weiterhin werden Daten zu Parkmöglichkeiten erhoben. Abrufbare Daten sind die kontinuierlich erhobenen Auslastungswerte der Park-and-Ride-Parkplätze sowie die Auslastungswerte der Parkhäuser, die alle 10 Minuten aktualisiert werden. Ebenso können die Standorte der Elektroladestation für E-Autos und deren zugehörigen Belegungsstatus abgefragt werden.

Eine der wichtigsten und größten Datenquellen ist die „Traffic Lights Data“-Datensammlung, die sich aktuell in der Betaphase befindet. Diese erfasst Signalanforderungen von Akustik, Fußgänger, Radfahrer und KFZ; die ÖPNV-Voranmeldung, -Anmeldung und, -Abmeldung; den Status der Primär- und Sekundärsignale, sowie die Dauer der aktuellen Wellendauer und dem Status des Signalprogramms. Eine detaillierte Beschreibung dieser Daten ist im Usage Guide [He22] definiert.

Im Rahmen des Smart City Modellprojekts „Connected Urban Twins (CUT)“ arbeitet Hamburg mit den Städten Leipzig und München daran, die Urbanen Datenplattform und Urbanen Digitalen Zwillinge miteinander zu verbinden.

### 3.2.2 Darmstadt

In der Stadt Darmstadt befindet sich eine weitere Urbane Datenplattform in Betrieb, die zugleich weiterentwickelt wird. Diese wird durch folgende „Vision“ definiert: „Die Digitalstadt Darmstadt verfügt über eine zentrale Datenplattform, die relevante öffentliche und private Informationen übergreifend bündelt. Daten, Informationen und Prozesse werden zentral zur Verfügung gestellt, um den Nutzern neue Anwendungen anbieten zu können“ [Fr21]. Abbildung 13 zeigt die Struktur und High-Level-Architekturübersicht.

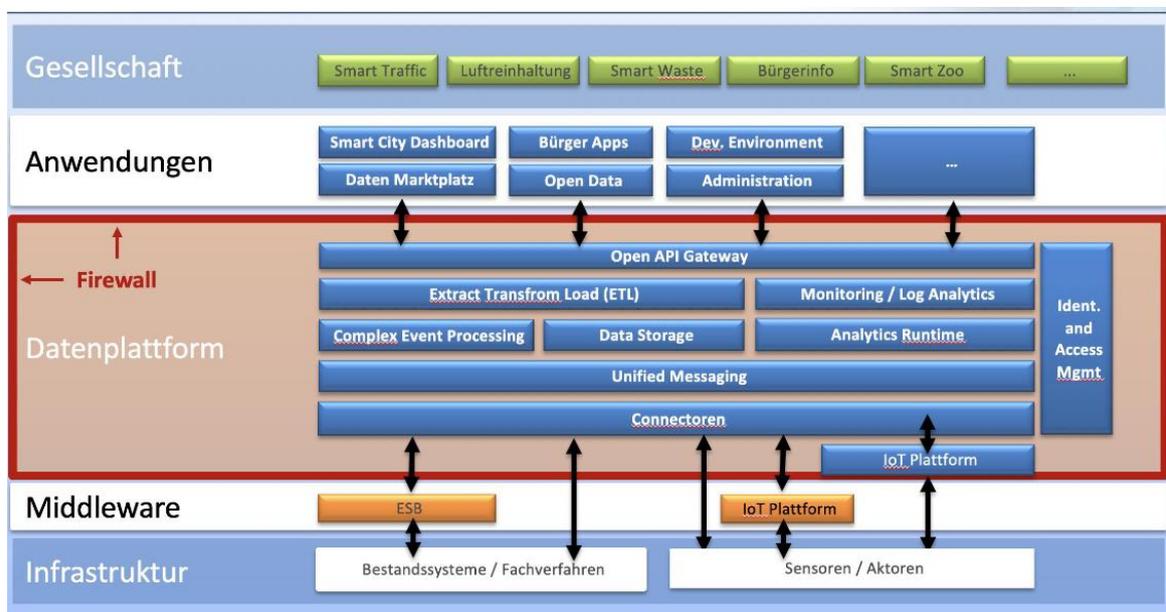


Abbildung 13 - Datenplattform der Stadt Darmstadt (aus [He21])

Auch in dieser UDP-Architektur findet eine grobe Einteilung in logische Schichten. Die unterste Ebene stellt auch in dieser UDP die Basisdateninfrastruktur aus IoT-Sensorik da. In diese Ebene werden zusätzlich bestehende Fachdaten, wie Geodaten hinzugezählt. Die IoT-Plattform, zur Verwaltung und Aufnahme der Messwerte, wird hier allerdings als Teil einer zusätzlichen *Middleware*-Schicht aufgefasst oder bei existierenden Systemen, als Teilsystem der Datenplattform.

Den Kern der Zentralen Datenplattform Darmstadt bildet die *Datenplattform*-Schicht. Über die Komponenten der Konnektoren, die die Schnittstelle zu den Eingangsdaten bildet, können viele Datenquellen unterschiedlicher Formate einheitlich in die Plattform eingespeist werden.

In der Schicht finden ebenfalls die Datenverarbeitung, Speicherung und Analyse in verschiedenen speziellen Komponenten statt. Die Daten werden wiederum über die über die *Open API Gateway* Komponente für Anwendungen und Nutzer verfügbar gemacht.

Die aus der Datenplattformschicht entstehenden Anwendungsfälle und Applikationspotenziale werden noch einmal in zwei Kategorien eingeteilt. In der Schicht *Anwendungen* befinden sich insbesondere visualisierende und datenbereitstellende Anwendungen, wie das *Smart City Dashboard* oder der *Daten Marktplatz*. In der obersten Schicht *Gesellschaft* werden die Anwendungen zusammengefasst, die der Idee der Smart Services der DIN SPECs entsprechen. Repräsentiert durch die Komponenten wie *Smart Traffic* oder *Luftreinhaltung* werden Applikationen entwickelt und erprobt, die ebenfalls Verkehrsflussoptimierungen und Verkehrssteuerungsansätze unter Beachtung von Umweltaspekten umsetzen wollen.

Ein hervorzuhebender Anwendungsfall ist die Bürger-App. Darin ist bereits ein Anreizsystem umgesetzt. Aus Kapitel 3.1.2 des Berichts [Se23] betrifft dies den Bereich des langfristigen Anreizsystems.

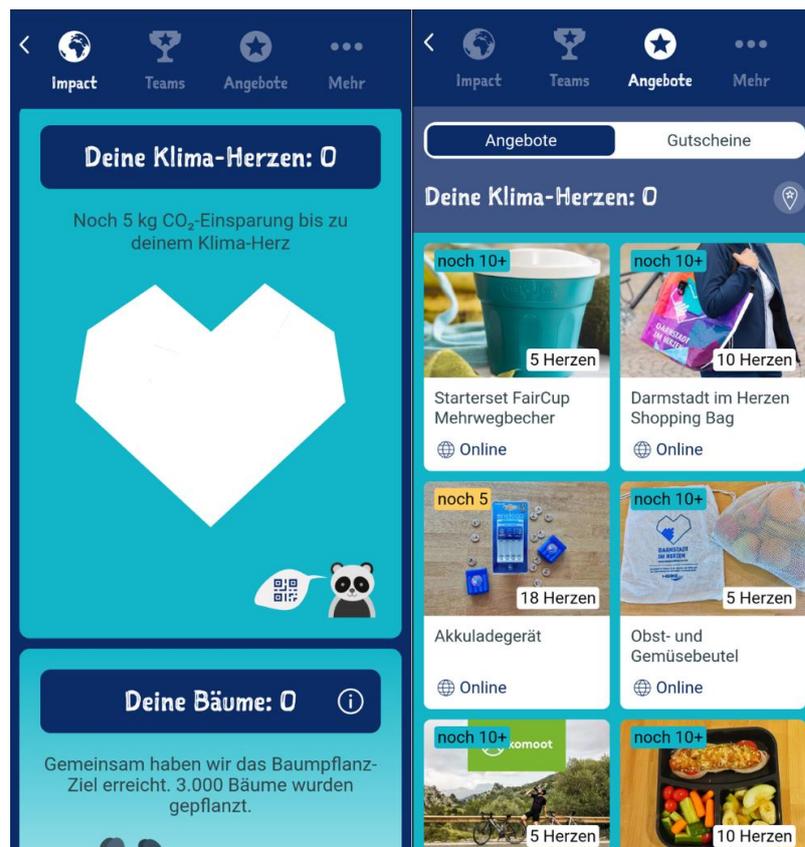


Abbildung 14 - Screenshots aus der App "Darmstadt im Herzen"

Durch das Tracking der zurückgelegten Strecken werden CO<sub>2</sub>-Einsparungen berechnet, beziehungsweise angenähert. Für jeweils 5 Kilogramm eingespartem CO<sub>2</sub> bekommt ein Nutzer eine

Einheit der Währung „Klima-Herzen“. Diese Klima-Herzen können dann gegen Angebote oder Gutscheine eingetauscht werden (siehe Abbildung 14). In einer von den Klima-Herzen unabhängigen Aktion wird zudem je 100 Kilometer, die nutzerübergreifend als Alternative zum Auto zurückgelegt worden sind, ein Baum gepflanzt.

Verkehrsmittel	CO2 Einsparung (g/km) gegenüber Auto	Strecke für 1 Klima-herz (km)	Herzen pro Woche für einen 10km-Pendler
Laufen & Fahrrad fahren	176	28	3,57
ÖPNV	76	66	1,51

Die Umrechnung zwischen zurückgelegter Strecke und CO2 Einsparung wird pauschal angenommen und kann der Tabelle entnommen werden. Ein Pendler mit einem Arbeitsweg von 10 Kilometern kann bei einem Wechsel vom Auto auf das Fahrrad etwa 3,5 Klimaherzen pro Woche erreichen. Bei einem Wechsel auf öffentliche Verkehrsmittel sind es noch 1,5 Klimaherzen pro Woche. Bei der Berechnung der Einsparungen werden noch weitere Faktoren berücksichtigt, wie die Tatsache, dass nicht alle von der App aufgezeichneten Bewegung prinzipiell als komplette Ersetzung des Autoweges betrachtet und daher nicht vollständig als Einsparung interpretiert werden. Diese Parameter können allerdings angepasst werden. Sollen die Klima-Herzen beispielweise nur auf die regelmäßige Pendlerstrecke angerechnet werden, kann ein geographisches Filtern erfolgen oder die Einsparungsabschätzung geändert werden. Die automatische Erkennung des Fortbewegungsmittels erfolgt in Zusammenarbeit mit dem Unternehmen MOTIONTAG, die Applikationen für Android und iOS unterstützen [Mo23].

In der Architektur ist ebenfalls die Zusammenschaltung verschiedener Plattformen berücksichtigt, wie in Abbildung 15 zu sehen ist. Dies verfolgt den „System of Systems“-Ansatz, der ebenfalls in der DIN SPEC der Offenen Urbanen Plattform definiert ist. In diesem Fall beziehen sich die Subsysteme auf die Plattformen der einzelnen Teilbereiche der Stadt Darmstadt.

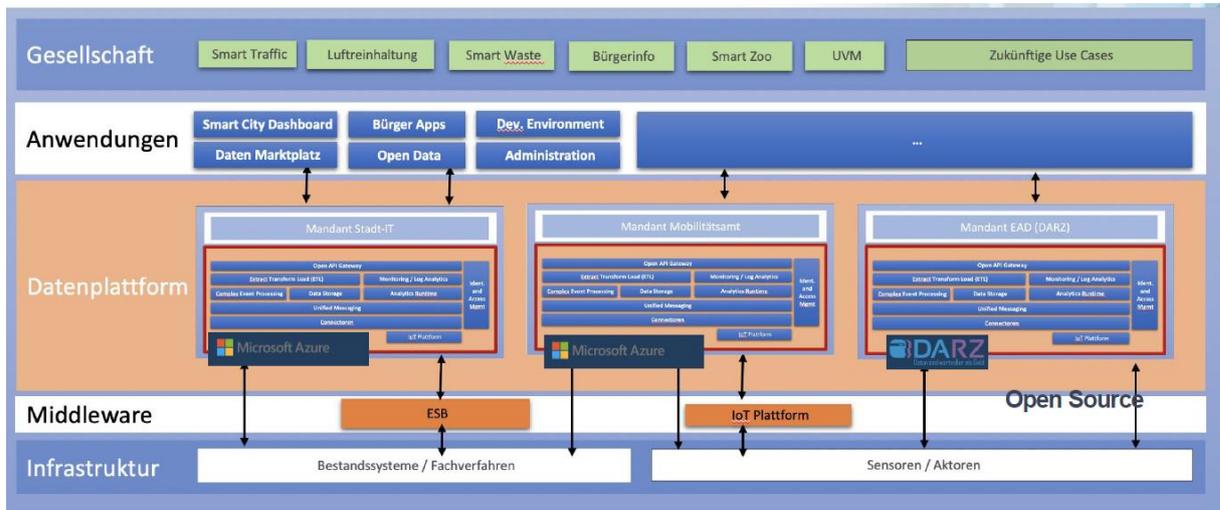


Abbildung 15 - UDP Darmstadt als System of Systems (aus [He21])

### 3.2.3 Paderborn

Auch die Stadt Paderborn forschte im Rahmen eines Projektes mit etwa 3 Mio. € Volumen an einer Urbanen Datenplattform (Zentrale Open Data Plattform) [Pa19a]. Dieses Projekt lief von 2019 bis 2022, soll aber weiterentwickelt werden. Der Stand zum Projektende kann der Abbildung 16 entnommen werden. Auch hier können die groben Schichten der allgemeinen Plattformarchitektur aus den DIN SPECS wiedererkannt werden.

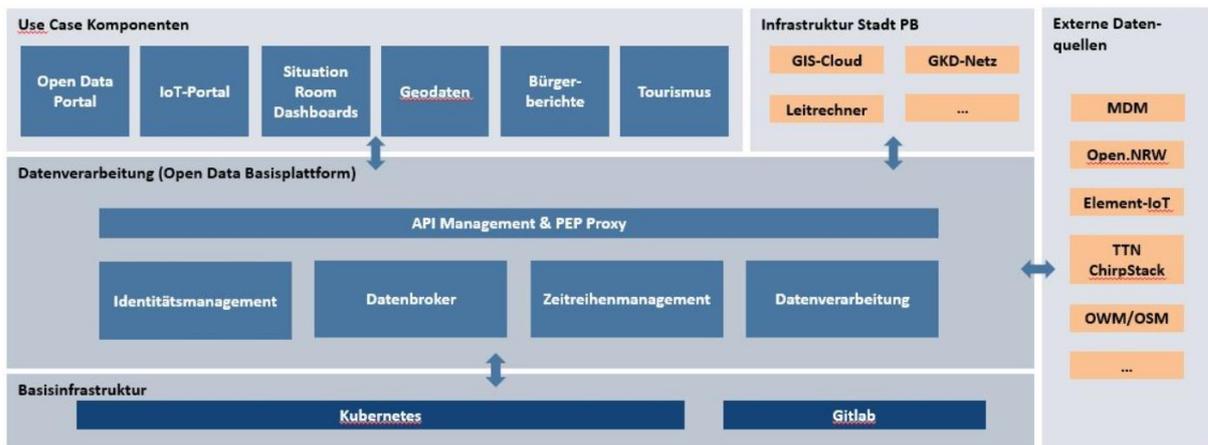


Abbildung 16 - Zentrale Open Data Plattform der Stadt Paderborn (aus [Pa19b])

Die Schicht der *externen Datenquellen* ist an der rechten Seite dargestellt und beinhaltet neben der Sensorik auch die Übertragungsnetze wie TTN. Als weitere Datenquellen ist die bestehende *Dateninfrastruktur der Stadt* angegeben, die insbesondere wie auch in Osnabrück aus Geodaten und Verkehrsdaten besteht.

Eine *Basisinfrastruktur* ist ebenfalls mit Kubernetes geplant und in einer eigenen Schicht gekapselt. Die Kernschicht der Datenlandschaft ist die *Open Data Basisplattform*, die zentral platziert ist. Hier sind ebenfalls die Kernelemente *Interfaces*, *Processing* und *Integration* wiederzufinden. Hervorgehoben ist hier explizit noch einmal das Zeitreihenmanagement mit Zeitreihendatenbanken als eigene Komponente. Der FROST-Server ist in diesem Architekturdiagramm nicht explizit dargestellt, wird aber ebenfalls ein Teil der Basisplattform sein. Durch das Identitätsmanagement ist ebenfalls eine Sicherheitskomponente definiert.

Die Applikationsschicht ist in Form der *Use Case Komponenten* definiert. Hierbei handelt es sich aktuell insbesondere um Visualisierungskomponenten, smarte Applikationen sind hier noch nicht konkret definiert.

Auch wenn die Entwicklung dieser Plattform nicht abgeschlossen ist, lassen sich die Grundgedanken der Plattform an anderen Stellen wiedererkennen. Mit der Architektur, orientiert an der DIN SPEC 91357, ist eine Erweiterbarkeit gegeben.

### 3.3 Vergleich

Auch wenn die vorgestellten Plattformen der Städte Hamburg, Darmstadt und Paderborn unterschiedlich entwickelt wurden und werden, lassen sich übergreifende Übereinstimmung in den Architekturen und der groben Struktur der Urbanen Datenplattformen wiedererkennen. Alle dieser Plattformen sind zudem unter Berücksichtigung der in 3.1 vorgestellten DIN-Spezifikationen definiert und erstellt worden. Dabei hat sich bei diesen OUPs die folgende Struktur bewährt. Die Gemeinsamkeiten beziehen sich insbesondere auf die Städte Hamburg und Darmstadt in Verbindung mit den DIN SPECS, da die Architektur der Stadt Paderborn in dem aktuellen Projekt noch nicht fertiggestellt ist.

Die Basis der Plattform stellt die Intrastruktur dar. Kern ist die IoT-Sensorik, die in den Städten verteilt Messwerte aufnimmt. Den andern Teil der Datenbasis stellen die (meist bereits vorliegenden) statische Fachdaten der Städte, insbesondere der Geoinformationen dar. Die IoT-Sensorik wird über eine IoT-Plattform in die Stadt integriert. Hier gibt es leichte Abweichung in der Architekturdefinition, inwiefern die IoT-Plattform zur Infrastruktur bzw. den externen Systemen gehört oder ein logischer Teil der Urbanen Datenplattform ist. Technisch hat dies jedoch geringe bis keine Folgen.

Die nächste Schicht ist die eigentliche Datenplattform. Hier befinden sich die vorgestellten Konnektoren mit den Schnittstellen zur Datenerfassung und Weitergabe. Im Kern basiert die Urbane Datenplattform auf dem FROST-Server, kann aber durch weitere Datenhaltungssysteme, wie Zeitreihendatenbanken oder Datenbanken für andere Typen von Dateien erweitert werden. Dies hängt jeweils von den Anwendungsfällen ab. Auch die Sicherheitsaspekte, wie

Datenzugriffs- und Verwaltungsmanagement ist hier verortet. Ebenso wie Datenverteilungs-, verarbeitungs- und -integrationskomponenten, wie es auch in den DIN SPECS vorgesehen ist.

Auf dieser Datenplattform setzt die Applikationsebene auf. Hier liegt der Fokus insbesondere auf der Visualisierung der Daten. Dazu gehören die statischen Geodaten über Kartendienste und die Visualisierung und Aufbereitung der gemessenen IoT-Daten durch Dashboards.

Die komplexen Anwendungsfälle sind nicht direkt in der Applikationsebene dargestellt. In Hamburg sind diese in „Other Services“ und „Third Party Apps“ zusammengefasst. In der Paderborner Architektur fehlen sie gänzlich. In Darmstadt gibt es dazu eine eigene, weitere Schicht „Gesellschaft“, die eine Abweichung zur Plattform aus Hamburg darstellt und stärker an den DIN SPECS orientiert ist und sich an der Schicht der Smart Services orientiert. Dort sind die Anwendungskomponenten verortet, die einen (gesellschaftlichen) Mehrwert aus den Daten generieren sollen, wie etwa die Verkehrsoptimierung (Smart Traffic) und der Luftreinhaltung, die auch Anwendungsfälle der Stadt Osnabrück und des IIP-Projektes (mit dem Fokus auf den Pendlerverkehr) sind.

Ein weiterer Unterschied zwischen den Plattformen stellt die Darstellung der Infrastruktur dar. Diese ist in den Plattformen der Stadt Hamburg und Paderborn explizit als Infrastruktur der Server mit Kubernetes als Technologie herausgestellt. Dies führt zu einer technischeren Darstellung.

In den Architekturdarstellung der jeweiligen Plattformen ist eine vertikale Trennung der Subsysteme, wie in der DIN SPEC 91367 nicht dargestellt. In den Architekturen liegt der Fokus auf einem groben Plattformüberblick. Die Architekturen der einzelnen Smart City Applikationen sind hingegen nicht dokumentiert. Nichtsdestotrotz können die vorgestellten Architekturen als Ausgangssituation für den Osnabrücker Ansatz berücksichtigt werden.

### 3.4 Andere Plattformanbieter

- Regio.it (Aachen)
- Travekom (Stadtwerke Lübeck)
- Civitas Connect e.V.
- Urban Software Institute [ui!]
- Fiware
- DKSR
- Hypertegrity
- Fraunhofer IESE
- re:edu (Münster)
- Masterportal (Hamburg, Osnabrück)

## 4 Schnittstellen

### 4.1 FROST-Server

Kern der Urbanen Datenplattform ist ein FROST-Server, der vom Fraunhofer IOSB entwickelt wird und für IoT-Anwendungen ausgelegt ist. Dieser setzt den OGC SensorThings API Standard um. „Die SensorThings API wird von der Europäischen Kommission mittlerweile als »Good Practice« für die Bereitstellung von Messdaten nach der INSPIRE-Richtlinie empfohlen“ [Fr23b]. Die Anforderungen einer hohen Leistungsfähigkeit, eines geringen Ressourcenverbrauchs und eines Open Source Ansatz stehen bei der Entwicklung des FROST-Servers im Vordergrund. Abfragen sind typischerweise ortsbasiert (Sensoren an oder in einem bestimmten Bereich) oder zeitbasiert (Messwerte über einen bestimmten Zeitraum).

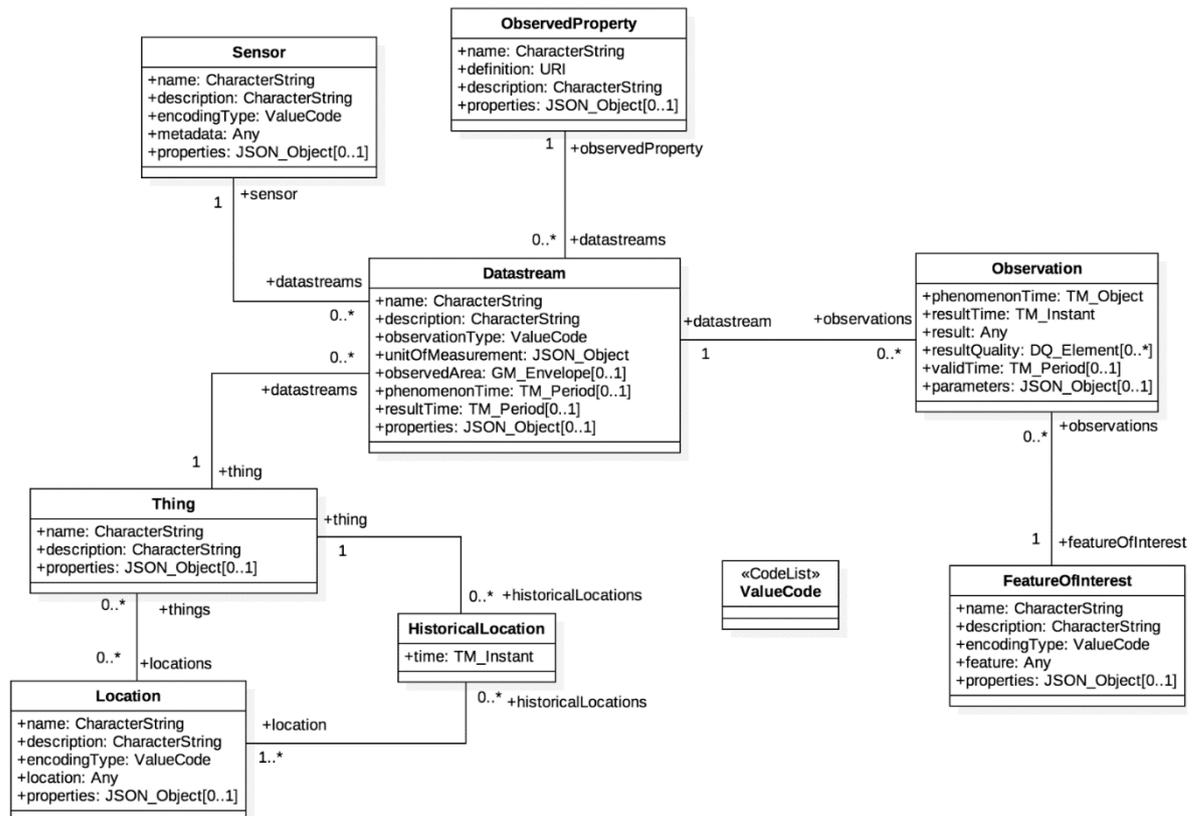


Abbildung 17 - SensorThings API Datenstruktur ([Op21])

Abbildung 17 zeigt die Datenstruktur der SensorThings API [Op21]. Da diese den Kern der Abbildung von Sensoren und Messwerten, sowie der Abfrage der Daten der UDP darstellt, werden die wichtigsten Komponenten nachfolgend erklärt. Abschließend werden diese an einem Beispiel der Verkehrszählung durch eine an einer Kreuzung angebrachten Verkehrskamera in der Stadt Osnabrück veranschaulicht.

**Thing.** Die Things sind die physischen Objekte, die mit dem Internet of Things verbunden sind und Messwerte liefern. Diesen ist jeweils eine **Location** zugeordnet, an dem sich das jeweilige Thing befindet. Da die Objekte teilweise auch an mobilen Geräten (zum Beispiel an Bussen) angebracht sind, kann sich die aktuelle Location ändern, sodass durch die **Historical Locations** alte Messwerte der jeweiligen Positionen des Things zugeordnet werden können. Jedes Thing kann aus mehreren **Sensoren** bestehen (zum Beispiel Temperatursensor und Niederschlags-sensor einer Wetterstation). Der Messwerttyp und die zugehörige Beschreibung wird in der **Observed Property** gespeichert. Die Messwerte zu einem konkreten Zeitpunkt sind jeweils eine **Observation**. Diese können über einen **Datastream** abgefragt werden. Durch die verschiedenen Parameter können detaillierte Informationen über die einzelnen Objekte hinterlegt werden, sodass beispielsweise Beschreibungen oder standardisierte Datentypen und weitere Metadaten in den Datenstrukturen selbst hinterlegt werden können. Auch die Relationen werden bei Anfragen als Verlinkungen zwischen den Objektstrukturen zurückgegeben, sodass durch die SensorThings API navigiert werden kann. Während der Kern zum Auslesen der Messwerte die Datastreams sind, sind jederzeit auch Anfragen zu allen anderen Entitäten möglich.

Nachfolgend wird dies in einem Beispiel konkret dargestellt. Am August-Bebel-Platz in Osnabrück sind Verkehrskameras installiert. Diese erfassen durch Bildverfahren die Anzahl und Typen der an definierten Spuren passierten Verkehrsteilnehmenden (Autos, LKW, Fahrräder, ...), jeweils in 5-Minuten-Intervallen. In Tabelle 1 werden die einzelnen Entitäten jeweils anhand dieses Anwendungsfalls veranschaulicht.

Entität	Funktion im Anwendungsbeispiel Verkehrszählung
Thing	Das Thing repräsentiert die Verkehrszählung durch eine explizite Verkehrskamera.
Location	Die Position der installierten Verkehrskamera.
Historical Location	Da es sich um eine fest installierte Kamera handelt, wird die Position in der Regel nicht verändert, sodass die Historical Locations hier keine Relevanz haben.
Sensor	Die Verkehrskamera hat in diesem Fall nur einen abzubildenden Sensor. Dabei handelt es sich um die Kamera selbst, die die Zählung durch Objekterkennung vornimmt.
Observed Property	Die Anzahl der Verkehrsteilnehmenden jeweils für einen Verkehrsmitteltyp (beispielweise Auto). Das bedeutet, dass der definierte Sensor mehrere Observed Properties besitzt.

Observation	Ein konkreter Zählwert mit Zeitstempel.
Datastream	Eine Menge von Observations zur Abfrage mit detaillierter Information über die Messung.

**Tabelle 1 - Anwendungsbeispiel zur SensorThings API – Verkehrszählung (nach [St22])**

Der FROST-Server ist in Java entwickelt und auf GitHub verfügbar [Fr23a]. Die Referenzimplementierung des FROST-Clients ist ebenfalls in Java geschrieben. Zusätzlich wird ein Python-Client bereitgestellt. Ein Codebeispiel für eine Abfrage der Zählwerte der Verkehrskamera wird nachfolgend dargestellt.

```
SensorThingsService service = new SensorThingsService(new URL("https://daten-api.osnabrueck.de/v1.1/"));  
EntityList<Thing> things = service.things().query().filter("startsWith(name, 'Verkehrszählung')").list();
```

Der Frost-Client wird zunächst mit einer URL initialisiert, hinter der die FROST-Instanz bereitgestellt wird. Durch dieses Objekt können nun Anfragen zu allen Entitäten der SensorThings API gestellt werden. Konkret werden hier alle Things ausgelesen, die „Verkehrszählung“ im Namen enthalten. Da aktuell nur eine Kreuzung eingepflegt ist, sind dies alle Kameras, die am August-Bebel-Platz installiert sind.

Für ein bestimmtes Thing kann nun ein Datastream über die vom FROST-Server definierte ID abgerufen werden:

```
Datastream stream = thing.datastreams().find(1075);
```

Zu diesem Datastream können nun Anfragen zu Observations, also den konkreten Messwerten gestellt werden. Beispielsweise ließen sich die neuesten Zählwerte auslesen, mit denen sich nun Auswertungen durchführen lassen:

```
EntityList<Observation> observations = stream.observations().query().orderBy("phenomenonTime desc").list();  
observations.forEach(observation -> {  
    System.out.println("Observation: " + observation.toString());  
});
```

Aus diesem Datastream kann auch wieder auf das dazugehörige Thing zugegriffen werden, womit die beschriebene Navigation durch die Datenstruktur auch im Code möglich ist. Hier kann über den Rückweg den Streams auch direkt auf die aktuelle Location des Things geschlossen werden:

```
stream.getThing().getLocations();
```

## 4.2 Verkehrsrechner

Die Schnittstelle zum Verkehrsrechner ist OCIT-C über SOAP. „OCIT-C steht für Open Communication Interface for Road Traffic Control Systems – Center to Center. Mit OCIT-C werden die

Funktionen zur Kommunikation zwischen zentralen Verkehrssteuerungs- und Verkehrslenkungssystemen abgedeckt“ [Oc18]. Auch das Auslesen und Einspeisen von Verkehrsdaten aus oder in den Verkehrsrechner, erfolgt über diese Schnittstelle.

Der Aufbau, die Struktur und die einzelnen Komponenten des Verkehrsmanagements mit OCIT kann Abbildung 18 entnommen werden.

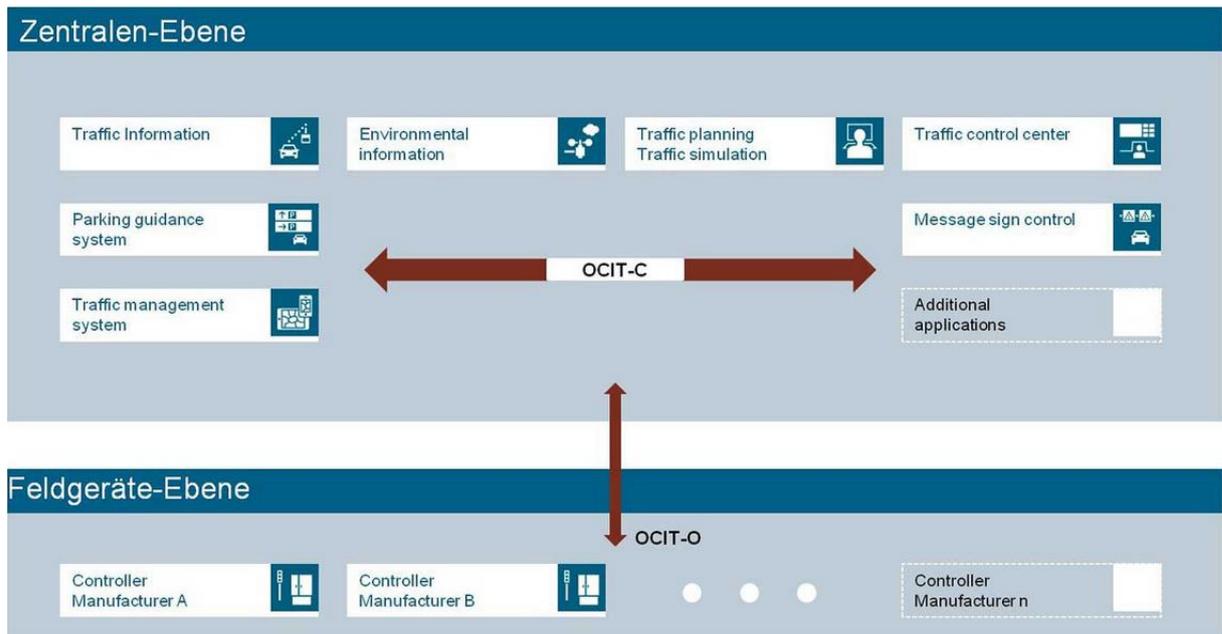


Abbildung 18 - OCIT Struktur und Komponenten (aus [Li23], [Yu23])

Daten, die besondere Relevanz für die Anwendungsfälle des IIP-Projektes haben, sind die aktuellen Messwerte der Detektoren, also der einzelnen Fahrzeugzähl- und Messstationen. In Abbildung 19 ist die dazugehörige Datenstruktur aus der OCIT-C Dokumentation beispielhaft dargestellt. Sie trägt die Bezeichnung *TrafficData\_detector\_currentValue*.

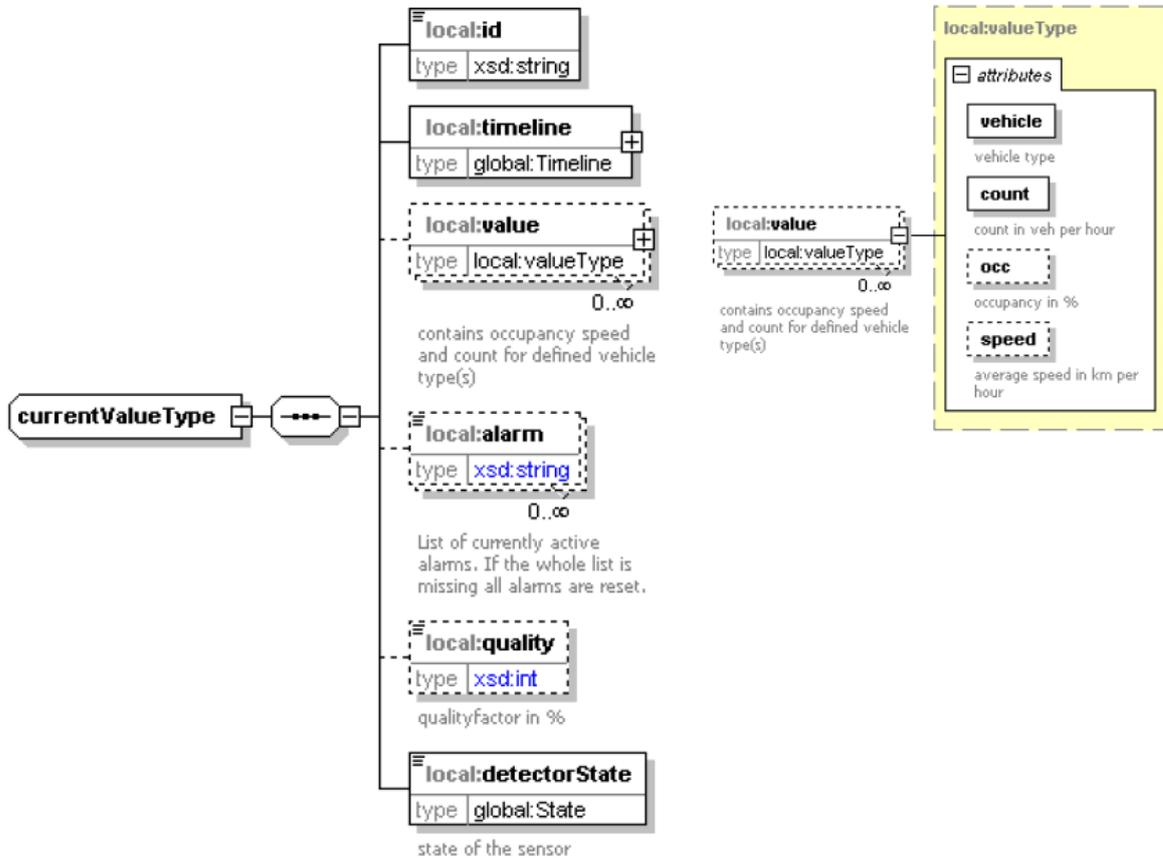


Abbildung 19 - OCIT-C Dokumentation des TrafficData\_detector\_currentValue-Objekts (aus [Oc18])

Herauszustellen ist hierbei, dass die zurückgegebenen Werte keine Rohwerte sind, sondern bereits vorberechnet wurde. Beispielsweise werden die Fahrzeugzählwerte immer auf eine volle Stunde hochgerechnet. Außerdem wird die berechnete prozentuale Belegung der Fahrspur und die durchschnittliche Fließgeschwindigkeit zurückgegeben, was weitere wichtige Daten zur Abschätzung und Ermittlung des Verkehrsflusses sind, die bei der Optimierung des Pendlerverkehrs benötigt werden.

Der API-Zugriff auf die OCIT-C Schnittstelle erfolgt nicht wie beim FROST-Server über REST, sondern über SOAP (Simple Object Access Protocol). SOAP-Anfragen sind XML-basiert senden und liefern eine komplexere XML-Struktur zurück.

```
client = Client('http://172.25.0.1:86/OCIT_CS0AP?wsdl')
return client.service.inquireAll(
    userName=user,
    password=password,
    objectType=TrafficData_detector_currentValue)
```

Mit einem Python-Skript kann auf diesen Objekttyp zugegriffen werden. In Tabelle 2 sind einige gefilterte Messwerte zu sehen. Darin zu erkennen sind sowohl gültige als auch ungültige Messwerte, sowie Messungen für verschiedene Fahrzeugtypen oder alle zusammen. Die Verarbeitung der Daten muss dementsprechend angepasst werden.

data.id	data.timeline.Timestamp	data.timeline.Cycle	data.quality	data.detectorState	vehicle	count	occ	speed
ig12FD264_D4_OP1.4	2023-04-19 08:45:01+00:00	60		substitute_value	all	120.0	10.0	8.0
ig12FD140_D15_15	2023-04-19 08:45:01+00:00	60		substitute_value	all	220.0	32.0	5.0
ig12FD227_D8_OP3.2	2023-04-19 08:45:01+00:00	60		substitute_value	all	0.0	5.0	11.0
es11FD298_D3_OP1.3	2023-04-19 08:45:00+00:00	60		substitute_value	all	180.0	21.0	5.0
es11FD139_D19_T5.6	2023-03-07 17:04:59.399000+0		0.0	unknown				
OSN061S1W	2023-04-19 08:45:00+00:00	300		substitute_value	car	312.0	5.0	32.0
OSN061S1W	2023-04-19 08:45:00+00:00	300		substitute_value	truck	0.0	0.0	39.0
OSN061S1W	2023-04-19 08:45:00+00:00	300		substitute_value	truck_with_trailer	0.0	0.0	49.0
OSN061S1W	2023-04-19 08:45:00+00:00	300		substitute_value	unclassified	0.0	0.0	2.0

Tabelle 2 - Beispieldatensatz einer OCIT-C Abfrage

Die im Hintergrund erstellte und versendete SOAP-Struktur (SOAP Envelope) ist folgendermaßen aufgebaut.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
3  |   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4  |   <SOAP:Body>
5  |   |   <inquireAll>
6  |   |   |   <userName>userName</userName>
7  |   |   |   <password>password</password>
8  |   |   |   <watchdog>
9  |   |   |   |   <clientId>localhost</clientId>
10 |   |   |   |   <timeOut>1440</timeOut>
11 |   |   |   </watchdog>
12 |   |   |   <objectType>TrafficData_detector_currentValue</objectType>
13 |   |   </inquireAll>
14 |   </SOAP:Body>
15 </SOAP:Envelope>

```

In Zeile 2-4 wird die allgemeine Struktur aufgebaut. Die relevanten Daten werden in einem *Envelope* und *Body* gewrappt. Darin wird schließlich der Funktionsaufruf definiert. In diesem Fall *inquireAll* (Zeile 5). Dieser benötigt einige Parameter. Zunächst sind dies die Zugangsdaten für die OCIT-C Schnittstelle. Außerdem wird ein Watchdog für den Timeout der Funktion erstellt. Schließlich wird der gewünschte Objekttyp angegeben, wie in diesem Fall die aktuellen Messwerte der Detektoren. Diese werden als SOAP-Response ebenfalls als XML zurückgegeben und liefern die beschriebenen Daten.

An der Schnittstelle hervorzuheben ist, dass nicht für jeden Datensatz oder Endpunkt eigene Funktionen definiert sind. Die Abfrage von aktuellen Werten oder allgemeinen Datensätzen, wie die Liste aller Detektoren oder anderer Messwerte, sind alle über *inquireAll* mit entsprechenden anzupassenden Parametern umgesetzt. Insgesamt sind für den Lesezugriff drei Funktionen relevant.

**getContentInfo()** liefert eine Liste über alle verfügbaren auslesbaren Objekttypen und deren Lese- und Schreibrechteinformationen.

**inquireAll()** liefert alle vorhandenen Objekte oder die aktuellen Messwerte eines Objekttyps (je nach übergebenen Parameter), wie in den Codebeispielen gezeigt wurde.

**get()** wird eingesetzt, um neben aktuellen Messwerten auch historische Daten auslesen zu können. Hier kann mit Zeiträumen oder Offsets gearbeitet werden.

Weitere Funktionen sind für den IIP aktuell nicht relevant, können aber der OCIT-C Dokumentation entnommen werden [Oc18]. Beispielsweise kann im weiteren Verlauf *put* eingesetzt werden, um Daten auf den Verkehrsrechner einzuspeisen.

## 5 Verkehrsprognose

### 5.1 Mikroskopisch

Neben einer explorativen Datenanalyse der bestehenden Daten wird der Beginn der zwei täglichen Peaks in der Verkehrsmengenmessung, der durch ein- und ausfahrende Pendelnde verursacht wird, sowie die Verkehrsmenge am folgenden Tag untersucht und möglicherweise prognostiziert werden. Einzelne Messpunkte werden zunächst separat betrachtet, später werden sie in einem gemeinsamen Modell fusioniert. Dazu werden verschiedene Deep Learning Methoden verglichen und auf ihre Eignung für diesen Anwendungsfall untersucht. Deep Learning wurde in Verkehrsprognoseproblemen umfassend eingesetzt, um räumliche und zeitliche Abhängigkeiten zu modellieren [YWW23], [NKW18], [KDJ18]. Für einzelne Messpunkte können Long Short Term Memory (LSTM) und Gated Recurrent Units (GRU) Neural Networks (NNs) verwendet werden [FZL16]. Für mehrere Messpunkte eignen sich Graph Neural Networks (GNNs) wie beispielsweise Spatial-Temporal Fusion GNNs [LMZ21]. GNNs sind in den letzten Jahren zur Spitze der Forschung im Bereich Deep Learning geworden und zeigen hervorragende Ergebnisse in verschiedenen Anwendungen. GNNs eignen sich ideal für Verkehrsprognoseprobleme aufgrund ihrer Fähigkeit, räumliche Abhängigkeiten zu erfassen, die durch nicht-euklidische Graphenstrukturen dargestellt werden [JL22] [BCY22].

### 5.2 Makroskopisch

Bis Daten aus der App und der mikroskopischen Sensorik vorliegen, werden synthetische Daten genutzt, um die verschiedenen Funktionalitäten zu testen. Es werden mikroskopische Pendlerdaten, genauer Bewegungsprofile erzeugt. Diese beinhalten Pendelnde die sich mit unterschiedlichen Verkehrsmitteln von Quellen, beispielsweise dem eigenen Zuhause, zu verschiedenen Zielen, beispielsweise der Arbeitsstätte, hin und zurück bewegen. Dabei werden abhängig vom Verkehrsmittel Routen erzeugt, die von den Pendelnden zurückgelegt werden.

Als eine Hauptdatenquelle dient im weiteren Verlauf die App, die verschiedene Daten über die Pendelnden erfasst. Dazu zählen neben der Position und Geschwindigkeit auch Gesundheitsdaten, wie Schrittdaten oder Herzfrequenz, die bei der Erkennung des verwendeten Verkehrsmittels helfen. Ziel ist es Bewegungsprofile zu erstellen, die Zeiträume, Positionen, Geschwindigkeiten und Verkehrsmittel beinhalten. Neben dem Verkehrsmittel wird auch der Zweck der Fahrt, beispielsweise Arbeitsweg, Einkauf, etc., ermittelt. Für die Ökobilanzierung werden die technische Fahrzeugdaten und der Besetzungsgrad benötigt. Die Datenquelle App wird durch die zweite Hauptdatenquelle, die noch zu installierende mikroskopische Verkehrserfassung der Stadtwerke Osnabrück (StOS), mit der Kennzeichen erkannt werden, ergänzt. Information über bestehende und benötigte Datenquellen sind in den Dokumenten Datenquellenbestand IIP – Gesamtübersicht [Re23] und Erschließung neuer Datenquellen [Re23b] sowie in der Zusammenfassung Datenquellenworkshop vom 06.06.23 [Re23c] enthalten.

Eine Simulation umfasst, wie vorher beschrieben, Bewegungsprofile der Pendelnden, die zur Evaluation des Anonymisierungsbaukasten genutzt werden. Die Prognosen aus dem makroskopischen Teil des Minimal Working Examples (MWEs) werden auch für das Routing und die Ankunftszeitvorhersage im mikroskopischen Teil genutzt.

## 6 Wetterdatenquellen

Unter dem Begriff Wetterquellen sind in diesem Bericht Datenplattformen, Clouddienste und APIs gefasst, die meteorologische Daten zur Nutzung anbieten. Im Folgenden werden verschiedene Wetterquellen vorgestellt und beschrieben welche Art von Informationen sie liefern. Zudem werden bei den einzelnen Quellen Aspekte vorgestellt, die förderlich für die Wahl zur Eignung für den IIP-Server sein könnten.

### 6.1 Anforderungen an wetterbezogene Quellen in IIP

- Es werden bevorzugt Dienste betrachtet, die (direkt oder in direkt) in Deutschland eigene Wetterstationen betreiben.
- Angesichts der Bedeutung der intelligenten Verkehrsbeeinflussung für das Gemeinwohl ist es in IIP vorgesehen und abbildbar, kommerzielle Wetterdienste einzusetzen. Auch die späteren Betreiber haben schon jetzt Zugang zu kommerziellen Wetterdaten.
- Um einen Vergleich mit der Mikroklimaerfassung durch Umweltsensoren zu ermöglichen, werden Wetterdaten mit möglichst hoher Auflösung benötigt. Aus Gesprächen mit Projektpartnern hat sich ergeben, dass viele verkehrsbezogene Betrachtungen und Entscheidungen auf Stadtteilebene und feingranularer erfolgen. Im Agrarbereich werden Auflösungen von 2 x 2 km angestrebt und realisiert. Wenn weiterhin die Konzentration der Betrachtungen auf dem Wall liegt, könnte ein Raster von 1x1 km wünschenswert sein.
- Der Wetterdienst soll gut integrierbar sein. Lambus berichtet von guten Erfahrungen mit [www.wetter.com](http://www.wetter.com) bzw. deren kommerzieller API <https://www.meteonormiqs.com/de/wetter-api/>.

### 6.2 Deutscher Wetterdienst

“Der Deutsche Wetterdienst (DWD) ist für die Erfüllung der meteorologischen Erfordernisse aller Wirtschafts- und Gesellschaftsbereiche in Deutschland zuständig.”

Dabei verfolgt der DWD eine OpenData-Strategie. Im Rahmen dieser Strategie wurde der Deutsche Wetterdienst 2017 damit beauftragt, seine Wetter- und Klimainformationen weitestgehend entgeltfrei zur Verfügung zu stellen. Diese Informationen können von einem Datenserver, der unter „[opendata.dwd.de](http://opendata.dwd.de)“ erreichbar ist, abgerufen werden. Bei regulärer Nutzung kann für diesen Dienst kein kontinuierliches Service und Verfügbarkeitslevel zugesichert werden, dies kann bei Geschäftsprozessen allerdings beauftragt werden [De23]. Bei regulärer Nutzung werden die Informationen von dem Datenserver per FTP abgerufen, während bei kommerzieller Nutzung auch SFTP möglich ist [De23b]. Die Daten der Klimastationen in 10-minütigen, stündlichen, täglichen, monatlichen und vieljährigen Mittelwerten abrufbar, die

aktuell von etwa 400 aktiven Klimastationen gemessen werden. Des Weiteren sind die Daten einer Station jeweils mit den Stationsmetadaten als Archiv abgelegt. Zudem gibt es sogenannte Terminwerte, die meteorologische Daten zu vereinbarten Uhrzeiten zur Verfügung stellen. Bei den Niederschlagsstationen liegt zudem eine 1-minütige Auflösung vor, was relevant für kurzfristige Vorhersagen sein kann. Der DWD verfügt über ein spezielles Niederschlagsmessnetz, welches zusammen mit gleichgestellten Partnermessnetzen auf ca. 2000 Niederschlagsstationen kommt. Weitere abrufbare Messungen und weitere Datensammlungen können beim DWD eingesehen werden.

Zusammengefasst bietet der DWD eine große Menge an Daten, die besonders im deutschen Raum ausgeprägt sind. Des Weiteren kann auf die meisten Daten direkt zugegriffen werden, was hilfreich für das Folgeprojekt ist. Bei kommerzieller Nutzung könnten auch vertraglich Sicherheiten bezogen auf Verfügbarkeit usw. festgelegt werden. Ein Nachteil ist die verschachtelte Struktur des FTP-Servers und das Nutzen vieler Abkürzungen. Zudem müssen die Rohdaten des Servers für weitere Nutzung formatiert werden.

### 6.3 Bright Sky

Bright Sky ist ein Open-Source Projekt, welches die Zielsetzung hat Daten des DWD und Vorhersagen des MOSMIX Modells in Form von JSON-Dateien abrufbar und leicht zugänglich für die Öffentlichkeit zu machen. Das Projekt wurde dabei von der Open Knowledge Foundation Deutschland, dem Prototype Fund Programm und dem deutschen Bundesministerium für Bildung und Forschung unterstützt. Es kann bei API-Abfragen zwischen drei Arten von Informationen unterschieden werden. Historical-Data und Current-Data wird direkt von dem DWD bezogen, während Forecast-Data auf das MOSMIX Modell zurückgreift [Ma23]. Es können beliebig viele Anfragen an die API gestellt werden, da sie vollständig kostenlos ist. Neueste Features die Brightsky auf experimenteller Ebene anbietet sind APIs zu Niederschlagswahrscheinlichkeiten, Sonnenstrahlung und einem Wetterradar.

Bright Sky würde sich für eine Implementation eignen, da die API vollständig kostenfrei verfügbar ist und als Quelle den deutschen Wetterdienst nutzt. Die Implementation wäre einfacher als bei dem DWD selbst, es sind jedoch insgesamt weniger Daten abrufbar. Ob die Informationen von Bright Sky ausreichend sind, muss bei einer Anforderungsanalyse an den IIP-Server geklärt werden.

### 6.4 AWS Weather Forecast

Amazon Web Services (AWS) ist generell eine Cloud-Plattform, die verschiedenste Services und Developer-Tools für Softwareprojekte anbietet. So ist AWS Forecast ein Deep Learning Service, der anhand von historischen und aktuellen Daten Prognosen für die Zukunft treffen kann

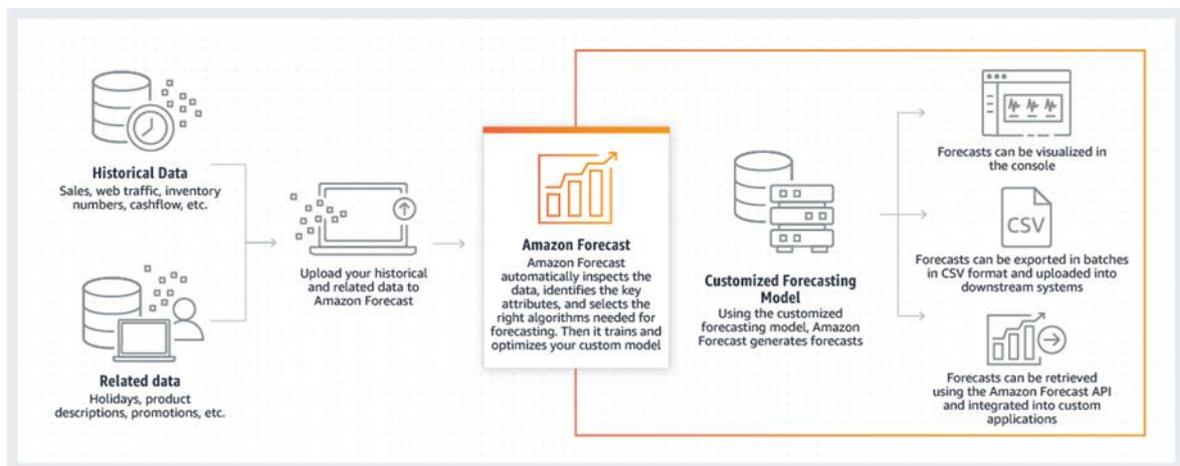


Abbildung 20 - Diagramm Amazon Forecast (aus [Am23a])

In Abbildung 20 ist die generelle Funktionsweise von Amazon Forecast dargestellt. Die oben erwähnten Daten werden in Amazon Forecast eingepflegt, welches diese inspiziert, anhand von Algorithmen interpretiert und anschließend das generierte Ergebnis in verschiedenen Darstellungsvarianten ausgibt [Am23a].

Der Amazon Forecast Weather Index ist ein Built-in-Feature von Amazon Forecast, welches die eingepflegten Wetterdaten nutzt, um Vorhersagen über Temperatur und Nachfrage bestimmter Ressourcen in Abhängigkeit vom Wetter geben zu können [Am23b]. Dabei analysiert Forecast gegebene Wetterinformationen und inkludiert sie in der Vorhersage, wenn sich durch die Informationen eine Verbesserung der Vorhersage ergibt. Trägt die eingegebene Zeitreihe mit Wetterinformationen nicht zu einem, laut Amazon Forecast, besseren Ergebnis bei, wird die Zeitreihe nicht angewendet [Am23b]. Generell enthält das Weather Index Feature Zeitreihen bestimmter Regionen die als Grundlage für Vorhersagen genommen werden.

Die Wettervorhersagen können zudem in verschiedenen Taktungen abgerufen werden. Die abrufbaren Vorhersagezeiten unterteilen sich in minütlich, stündlich und täglich. Maximal kann dabei eine Vorhersage bis 14 Tage in die Zukunft getroffen werden.

Da Amazon Forecast anhand von Datensätzen Empfehlungen ausspricht, würde sich der Dienst primär bei der Vorhersage von Auslastungen anbieten. Somit liefert er keine Wetterinformationen, sondern nutzt sie, um sie in Entscheidungsprozesse mit einfließen zu lassen.

## 6.5 Meteomatics

Meteomatics ist ein Schweizer Unternehmen, welches 2012 mit der Zielsetzung gegründet wurde, Wettervorhersagen und Wetteranalysen durch neue Technologien deutlich zu verbessern. Der Ansatz liegt dabei darin, dass verschiedene Schichten der Atmosphäre unterschiedlich gut erschlossen sind.

## 6.6 Wetter.com

wird nachgeliefert

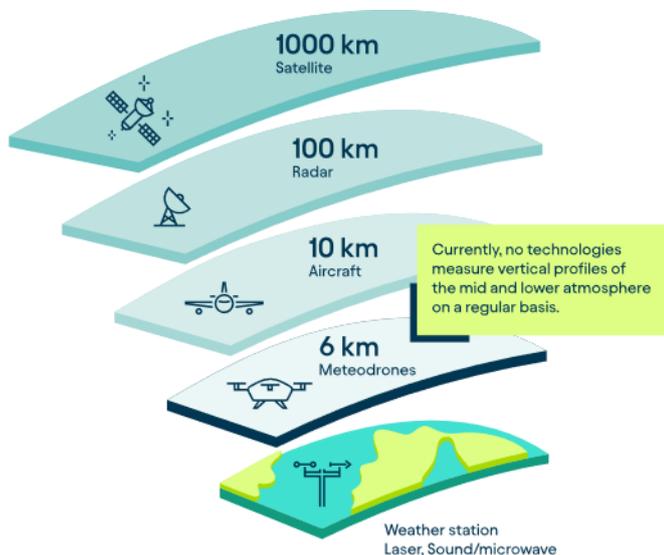


Abbildung 21 - Innovative Measurement Technology with Weather Drones (aus [Me23b])

Des Weiteren verfolgt Meteomatic den Ansatz des „90 Meter Downscalings“. Generell gibt es bei Wettermodellen unterschiedliche Detaillierungsgrade. Von einer „hohen Auflösung“ spricht man, wenn mindestens einmal pro Kilometer ein Messwert vorliegt. Diese Auflösung verwendet Meteomatics zusammen mit lokalen Geländemodellen um mithilfe einer proprietären Software die Datenpunkte auf 90 Meter Distanz herunterzubrechen [Me23].

Neben den Wetterinformationen durch die Drohnen greift Meteomatics auf einige weitere Datenquellen zurück, um den vorher erläuterten Detaillierungsgrad zu erreichen. In der nachfolgenden Tabelle sind die genutzten Datenquellen aufgelistet.

Wetter Datentyp	Herkunft Wetterdaten
Weather model data	ECMWF, GFS, DWD-ICON, MF-Arome, UKMO, HRRR, Ensembles, and many more
Radar data	Precipitation data from national radar networks from Germany, Switzerland, Austria, France, UK, USA, etc
Satellite data	Meteosat, Himawari-8, GOES-16 and 17

Weather station data	Station networks of national weather services: e.g. German and American weather services and some private providers
Atmospheric data	Global weather model data, radiosondes, drones, models for air pollutants/particles/fine dust (ECMWF CAMS, FMI SILAM)
Ocean data	Global ocean models such as ECMWF-WAM, ECMWF-CMEMS, NOAA-HYCOM
River and watercourse data	Individual measuring stations (e.g. Federal Office for the Environment in Switzerland)
Climate data and scenarios until 2100	Global climate model from the Meteorological Research Institute in Japan, climate scenarios based on IPCC's Shared Socioeconomic Pathways
Historical data since 1979	Global Reanalysis Model (ECMWF-ERA5)

**Tabelle 3 - Verfügbare Wetterdaten Meteomatics (aus [Me23c])**

Meteomatics bietet zudem einen 14-Tage API Test Account an, mit dem bis zu 1000 Abfragen ausgeführt werden können und eine Teilmenge an historischen Wetterdaten mitgeliefert wird.

Eine Implementierung für den IIP-Server wäre möglich, nachdem über eine Anforderungsanalyse an den Server geklärt wurde, dass Meteomatics Datenangebot einen Mehrwert gegenüber anderen Anbietern in Bezug auf den Server hat.

## 6.7 Open Weather Map

Open Weather Map ist ein 2012 gegründeter Online-Service, der verschiedene APIs zur Thematik Wetterinformationen anbietet [Op23]. Open Weather Map verwendet neue Verfahren

wie NN und Machine Learning (ML), sowie Open Data und Open Technologies um große Datenmengen und Wettermodelle evaluieren zu können [Op23b]. In der nachfolgenden Tabelle sind die Datenquellen angegeben, die Open Weather Map für ihr Numerical Weather Prediction Model nutzt [Op23c].

Ressourcenart	Datentypen
Global Numerical Weather Prediction Models	NOAA GFS 0.25 and 0.5 grid sizes NOAA CFS ECMWF ERA
Weather stations	METAR stations Users' stations Companies' stations
Weather radar data	
Satellite data	

**Tabelle 4 - Open Weater Map Datenquellen (aus [Op23c])**

Die verarbeiteten Datenmengen können über verschiedene APIs abgerufen werden. Open Weather Map bündelt dafür einige Daten zusammen und bietet sie als Datensammlung an. So kann je nach Unternehmen/Intention eine spezifische Sammlung abonniert, oder für einfache Tests die One Call API 3.0 mit bis zu 1000 kostenlosen API-Calls pro Tag genutzt werden. Weitere Anfragen über dem täglichen Limit würden dann 0,0014 EUR pro Call kosten [Op23]. Weitere APIs und Sammlungen, die angeboten werden, werden in nachfolgender Tabelle dargestellt.

Current & Forecast weather data collection					
Current Weather Data	Hourly Forecast 4 days	Daily Forecast 16 days	Climatic Forecast 30 days	Bulk Download	Global Weather Alerts

					Push notifications
5 Day / 3 Hour Forecast	Road Risk API	Solar Irradiance API			
Historical weather data collection					
History API	History API for Timestamp	History API Full archive	History Bulk	History Forecast Bulk	Statistical Weather Data API
Accumulated Parameters	Historical Weather Data by State for all ZIP codes, USA				
Maps collection					
Weather Maps 2.0 3-hour step	Weather Maps 2.0 1-hour step	Weather Maps 1.0	Global Precipitation Maps Historical data	Global Precipitation Maps Historical & Forecast data	Relief Maps
Other weather API's collection					
Air Pollution	Geocoding	Weather Stations	UV Index	Weather Triggers	

**Tabelle 5 - OpenWeather Products (aus [Op23])**

Insgesamt eignet sich die Open Weather Map für eine modulare Implementation im IIP-Server, da kostenlos auf Grundfunktionen zugegriffen werden darf und so einfach eine prototypische Implementation möglich wäre. Bei einer vollständigen Implementation muss geklärt werden, welche Angebote an Datensätze und Services für den IIP-Server relevant sind und ob diese Informationen bezogen auf den deutschen Raum und insbesondere Osnabrück einen Mehrwert gegenüber lokalen Anbietern haben.

## 7 Architekturf Entwurf

### 7.1 Systemlandschaft

Die Systemlandschaft in Abbildung 22 bietet einen Überblick über die bereits vor Projektstart von den einzelnen Projektpartnern eingebrachten Systeme und Komponenten und solche, die im Rahmen des Projekts entwickelt werden sollen. Es handelt sich dabei sowohl um Softwarekomponenten als auch um technische Komponenten und Datenquellen. Diese bilden die Basis und Rahmenbedingungen für die Architektur und werden daher nachfolgend kurz vorgestellt.

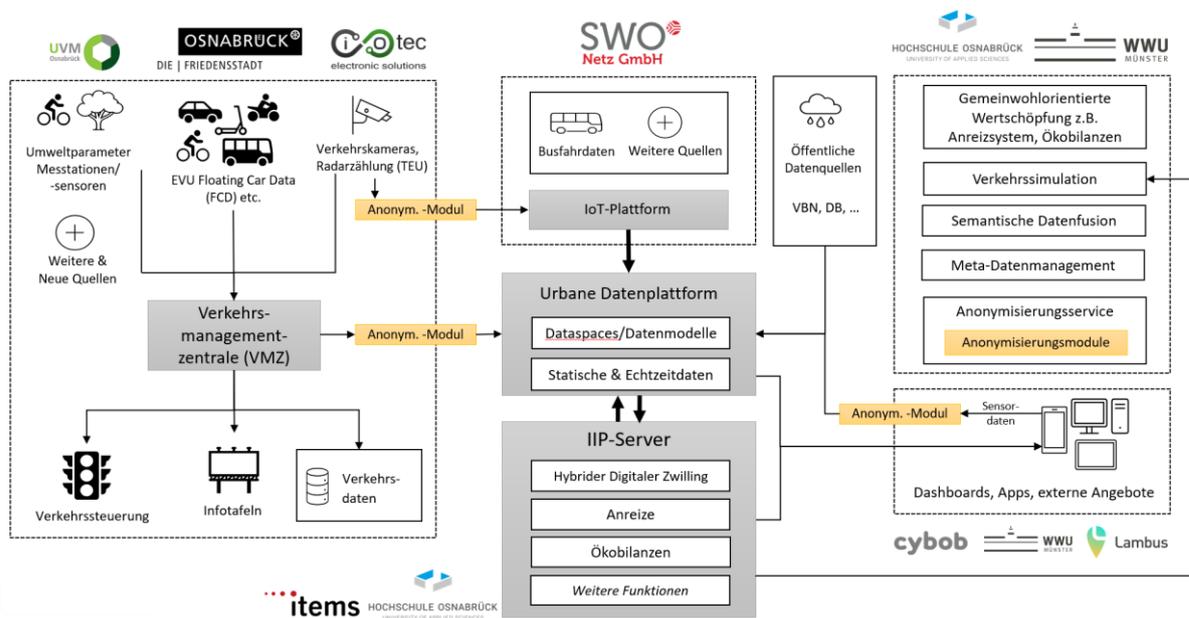


Abbildung 22 - Systemlandschaft des IIP-Projekts

**Umweltparameter-Messstationen.** Dabei handelt es um verschiedene Messstationen, die im Raum Osnabrück installiert sind. Insbesondere um die großen Messstationen am Schlosswall in Osnabrück. Diese spielen eine entscheidende Rolle bei der Messung zur Einhaltung der Feinstaubbelastungsgrenzen, insbesondere der NO<sub>x</sub>-Belastung [Hi20]. Diese Messungen haben auch eine Relevanz für den Verkehr: Im Rahmen des Umweltsensitives Verkehrsmanagements (UVM) der Stadt Osnabrück (siehe Abbildung 23), das im Jahr 2023 erstmals eingesetzt wird, werden je nach Situation verschiedene Verkehrsszenarien geschaltet, um die Feinstaubrichtwerte einhalten zu können [Os23]. Bei zu hoher Belastung wird der Verkehr eingeschränkt, um das Verkehrsaufkommen in der Innenstadt zu begrenzen und besser steuern zu können, was unter anderem auch über die **Infotafeln** kommuniziert wird.

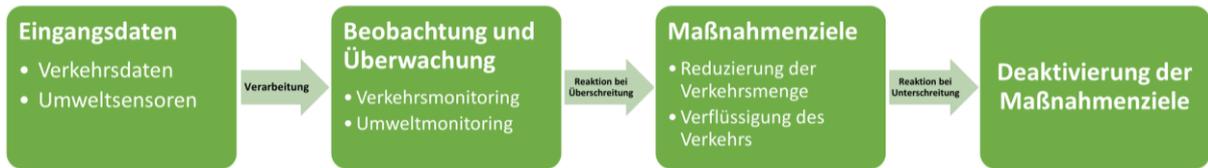


Abbildung 23 - Ablauf des Umweltsensitiven Verkehrsmanagements (nach [Os23])

Dies muss in den Szenarien des Verkehrsroutings und der Verkehrsoptimierung entsprechend berücksichtigt werden.

**Verkehrszählung** durch Verkehrskameras und Radarzählung (TEU). Die Verkehrszählung hat aktuell den größten Einfluss auf die Verkehrsplanung in Osnabrück. Über Verkehrskameras, Leiterschleifen in den Fahrspuren an Kreuzungen, und Radarzählung mit TEUs an den Osnabrücker Zufahrtstraßen wird unter anderem das Verkehrsaufkommen, die Durchschnittsgeschwindigkeit und die Verkehrsauslastung einzelner Spuren gemessen. Diese Daten werden in die Verkehrsmanagementzentrale eingespeist.

Die **Verkehrsmanagementzentrale** (VMZ, Verkehrsrechner). Anhand der beschriebenen Informationen werden verschiedene Verkehrsszenarien zur **Verkehrssteuerung** geschaltet, die je nach Situation beispielsweise morgens den eingehenden Pendlerverkehr oder nachmittags den ausgehenden Pendlerverkehr priorisieren und den Verkehrsfluss dahingehend optimieren. Dies geschieht durch verschiedene vordefinierte Szenarien, die jeweils eingesetzt werden und die Schaltung der Lichtsignalanlagen (LSA, „Ampeln“) ändern. Die Verkehrszählung ist nicht die einzige Quelle; hinzu kommen beispielsweise Baustelleninformationen und Busstandortinformationen. Für den ersten Schritt sollen aber insbesondere die Verkehrszähl-daten des Verkehrsrechners in den Vordergrund gestellt und genutzt werden

**Verkehrsdaten(-bank)**. Über den Verkehrsrechner können viele dieser Daten über Schnittstellen ausgelesen werden. Beispielsweise die aktuellen Verkehrsauslastungen, die im 5-Minuten-Takt aktualisiert werden. Ebenfalls lassen sich die historischen Daten hierzu auslesen. Des Weiteren können die Zustände der Lichtsignale aller Kreuzungen und der einzelnen LSA in Osnabrück ausgelesen werden. Auch hiervon sind historische Daten abfragbar. Grundsätzlich ist auch der umgekehrte Weg – eine Beschreibung des Verkehrsservers – möglich. Dadurch kann auf Verkehrsszenarien eingegriffen werden, was in der Endstufe des Projektes realisierbar wäre, indem der Verkehr aktiv auf Basis von Prognosen, anstatt wie bisher reaktiv, optimiert werden könnte. Dies reicht von der makroskopischen Ebene durch die Steuerung ganzer Pendlerströme, bis zur mikroskopischen Ebene wie der gezielten Durchleitung einzelner Busse. Auf die Schnittstelle zu dieser Datenbank wird im Kapitel der Schnittstellenbeschreibung eingegangen.

Die **IoT-Plattform** gehört zur Stadtwerke Osnabrück Netz GmbH. Sie basiert vor allem auf IoT-Sensorik in der Stadt Osnabrück, die über das speziell für IoT-Geräte entwickelte Kommunikationsprotokoll LoRaWAN betrieben wird [St23c]. Die gemessenen Daten werden in einem FROST-Server gespeichert und teilweise öffentlich verfügbar gemacht. Über die Schnittstelle des FROST-Servers [St23] oder visualisiert durch das Masterportal [St23d], können diese eingesehen werden. Ein Beispiel hierzu ist der Belegungszustand von Parkplätzen für Menschen mit Behinderung.

Die **Urbane Datenplattform (UDP)** ist eine für das Projekt neu zu entwickelnde Plattform, die für die (Roh-)datenhaltung zuständig ist. Hierin sollen alle relevanten in Osnabrück aufgenommene und verfügbare Daten zentralisiert und für alle Bürger frei zugänglich, bereitgestellt werden. Kern der UDP wird ein FROST-Server sein, der besonders für die Aufnahme und Verwaltung von IoT-Daten optimiert ist, die zeitbasiert und georeferenziert abfragt werden können. Der FROST-Server und der Zugriff darauf werden im Verlaufe dieser Arbeit, insbesondere in Kapitel 4.1 noch detaillierter behandelt.

Die Komponente **IIP-Server** fasst die Funktionalitäten zusammen, die über die Verwaltung der Daten in der UDP hinausgehen. Hierbei handelt es sich um eine Begrifflichkeit und nicht um einen einzelnen physischen Server. Dazu gehört die Aufbereitung der Daten für die einzelnen Module, wie etwa die Simulation oder das Anreizsystem, zum Beispiel durch semantische Datenfusion. Im Laufe der nächsten Kapitel wird auf die Funktionen und die Umsetzung weiter eingegangen.

**Dashboards und Apps.** Dabei handelt es sich um die Anwendungen für Bürger und andere Endnutzer. Kern bildet die „Pendler-App“ über die der Service des intermodalen Pendlerverkehrs aufbereitet und bereitgestellt wird. Ergänzen können dies weitere Dashboards, die Informationen aufbereiten und darstellen.

**Anonymisierungsmodule.** Anonymisierung und Datensicherheit sind einer der Kernpunkte des IIP-Projektes. Besonders relevant ist dies, da mit personenbeziehbaren Pendlerdaten, wie beispielsweise individuelle Routen, gearbeitet wird. Deshalb muss an allen Stellen, an denen Nutzerdaten entstehen (etwa die Datenerfassung und Weitergabe in der Pendler-App), garantiert sein, dass diese ausreichend anonymisiert sind. Dies gilt ebenso bei der Bereitstellung aller Arten von Daten, wenn diese, nachdem sie in die Urbane Datenplattform gespeist sind, öffentlich ausgelesen werden können.

(Neue) **Öffentliche Datenquellen.** Die Umsetzung der Anwendungsfälle und die Bereitstellung der Systeme erfordert diverse Datenquellen als Arbeitsgrundlage. Eine wichtige Aufgabe ist es daher, neue relevante Datenquellen zu erschließen und diese über einheitliche Schnittstellen verfügbar zu machen. Eine genauere Auflistung der Datenquellen kann den Dokumenten [Re23] und [Re23b] zum Datenquellenbestand entnommen werden.

Die vorgestellten Systeme sind die „High-Level“-Komponenten der Systemlandschaft, die unterschiedlich miteinander interagieren und in Anwendungsfällen eingesetzt werden. Auf die Interaktion mit dem IIP-Server und auf die einzelnen Subsysteme und Module des IIP-Servers wird im weiteren Verlauf eingegangen.

## 7.2 Anwendungsfallorientierte Konzeption

Zur Entwicklung einer ersten Architekturversion soll der Anwendungsfall des Eingangsbeispiel aus Kapitel 1.2 herangezogen werden. Daran wird die Herangehensweise für eine Architekturentwicklung entwickelt.

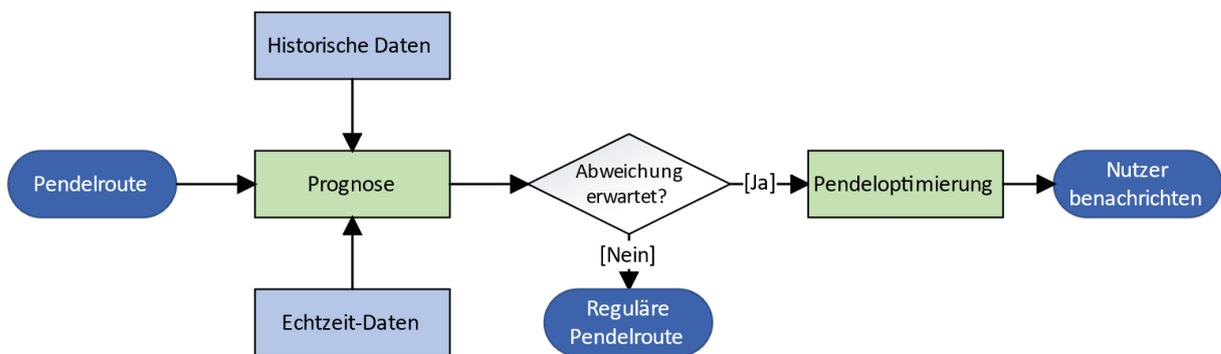


Abbildung 24 - Ablauf des Anwendungsfalls bis zur Pendlerbenachrichtigung

Abbildung 24 und Abbildung 25 zeigen den groben Ablauf des Anwendungsfalls von der Prognose bis zur Ankunft. Dies ist zunächst der Ablauf von der Simulation und Prognose bis zur Benachrichtigung des Pendlers für den nächsten Pendelvorgang, falls eine erwartende Abweichung wahrscheinlich ist. Des Weiteren gibt es den zweiten Teil, beim dem sich der Pendler, aufgrund der Handlungsempfehlung, von der Pendler-App begleitet, auf den Weg von oder zur Arbeit begibt. Die einzelnen Schritte und Aspekte werden nachfolgend als Anwendungsbeispiel herangezogen.

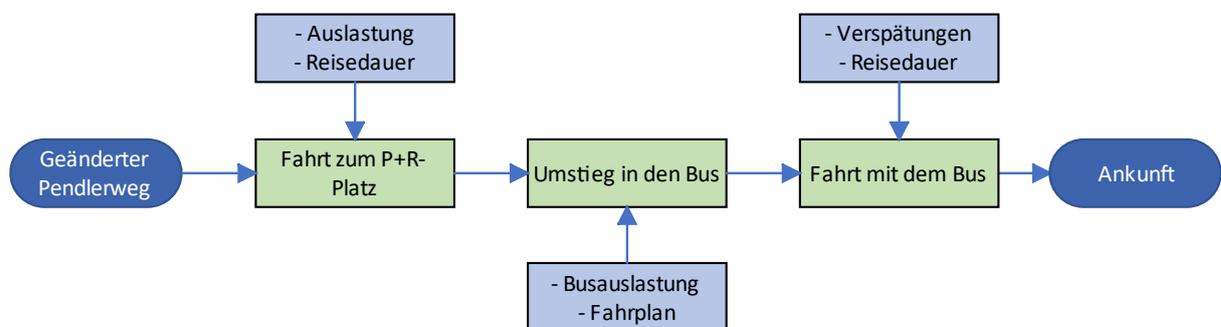


Abbildung 25 - Ablauf des Anwendungsfalls während des Pendelns

### 7.2.1 Qualitätsziele

Für die Architektur im Rahmen dieses Anwendungsfall und den darauf basierenden Erweiterungen werden nachfolgend wichtige Qualitätsziele definiert.

**Sicherheit.** Das IIP-Projekt arbeitet mit den Bewegungsdaten der Verkehrsteilnehmenden. Zusätzlich werden in der Pendler-App Bewegungsprofile und Empfehlungen auf Basis von individuellen Einstellungen und Präferenzen erstellt. Die Sicherheit dieser Daten muss daher jederzeit während der technischen Umsetzung eingehalten werden. Genauer wurde darauf bereits bei der Modulvorstellung in den Kapiteln 2.3 und 3.1.4 des Berichts [Se23] eingegangen.

**Erweiterbarkeit.** Dieser Architekturentwurf bezieht sich auf einen der Kernanwendungsfälle des IIP-Projektes und muss daher auch für weitere Anwendungsfälle kompatibel sein, um damit iterativ verschiedene weitere Szenarien abdecken und integrieren zu können. Dies beinhaltet weiterhin die Integration neuer Datenquellen und Verkehrsmittel in die Datenplattform und Applikationen, die in die Architektur integrierbar sein müssen. Des Weiteren besteht die Vision, die UDP der Stadt Osnabrück mit weiteren bereits bestehenden Applikationen, wie des UVMs oder anderen Datenplattformen zu verbinden. Diese Erweiterbarkeit für externe Systeme muss daher ebenfalls gegeben sein. Die phasenorientierte Erweiterung um verschiedene Verkehrsmittel ist in Abbildung 26 definiert.

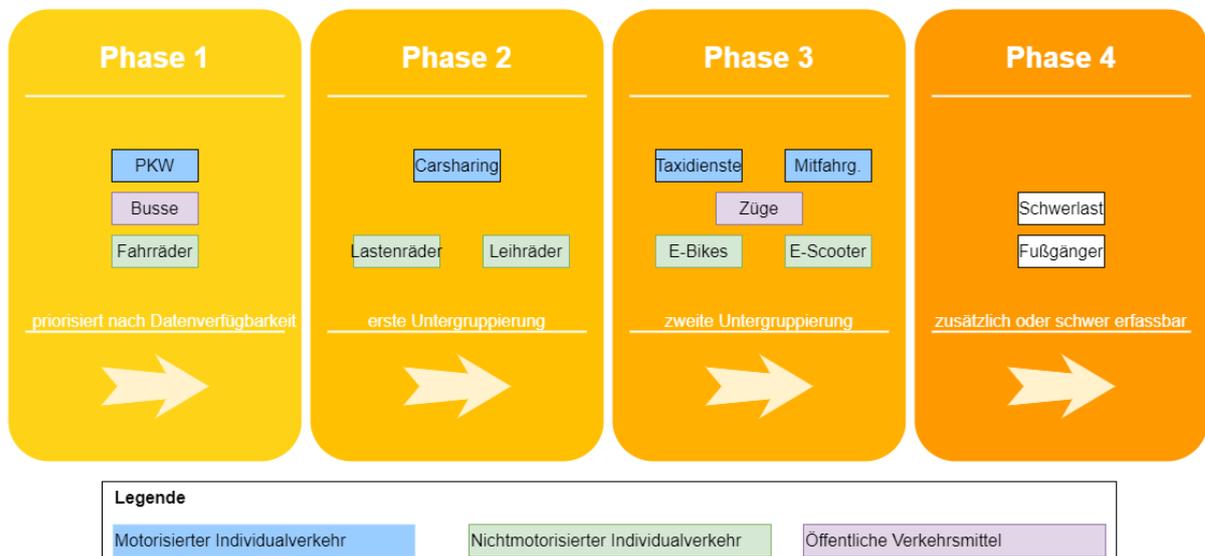


Abbildung 26 - Datenquellen Implementierungsphasen (aus [Re23b])

**Interoperabilität.** Die Interoperabilität setzt eng an der Erweiterbarkeit an. Zum einen sollen mit geringem Aufwand weitere Frontends, wie Dashboard und Geoportale angeschlossen werden können. Zum anderen sollen Kommunikation zwischen verschiedenen Plattformen in der Architektur vorbehalten können. Zudem müssen verschiedene Datenquellen einfach integrierbar sein. Ebenso Datensenden, die Smarte Daten und Applikationen bereitstellen.

**Benutzbarkeit.** Die Anwendungen, insbesondere die Pendler-App, sollen von möglichst vielen Verkehrsteilnehmenden, schwerpunktmäßig den Pendelnden, benutzt werden. Auch, wenn es sich im Hintergrund um komplexe interagierende Systeme handelt, müssen dem Pendelnden Handlungsempfehlungen, mit tatsächlichen Mehrwerten auf einfache und zuverlässige Weise präsentiert werden, damit eine breite Nutzerbasis eine Verkehrsoptimierung ermöglicht. Diese Handlungsempfehlungen müssen jederzeit aktuell und verlässlich sein. Außerdem müssen einfache und verfügbare Schnittstellen angeboten werden, um Anbieter und Entwickler Smarter Applikationen zu gewinnen, die weitere Mehrwerte schaffen oder zusätzliche Datenquellen für die Plattform bereitstellen.

**Skalierbarkeit.** Insbesondere bei steigenden Nutzerzahlen, aber auch bei Integration neuer Applikationen und Stakeholdern, muss das System jederzeit verfügbar und mit einem hohen Zugriffsaufkommen klarkommen können. Diese Punkt ist ein weiterer Aspekt, der die Benutzbarkeit verbessert.

## 7.2.2 Randbedingungen

Weiterhin werden einige allgemeine Randbedingungen definiert.

Die Applikationen müssen auf handelsüblicher Hardware laufen, damit die sie von allen Pendelnden eingesetzt werden kann.

Die Softwarearchitektur soll nicht auf Betriebssysteme und Programmiersprachen beschränkt sein, da diese festgelegten Rahmenbedingungen für verschiedene Applikationen und externe Anbieter nicht gewährleistet werden könnten.

Herangezogene Softwarekomponenten und Datenquellen sollten in der Regel frei verfügbar sein, um eine einfache Wiederverwendbarkeit, Wartbarkeit und Einsatz in externen Systemen gewährleisten zu können.

Die Softwarekomponenten sollen als Open Source zur Verfügung gestellt werden. Zum einen um die Entwicklungen im Rahmen des IIP-Projektes zu dokumentieren und zum anderen, um eine einfachere Erweiterbarkeit und Wiederverwendbarkeit der einzelnen Komponenten für weitere Applikationen zu gewährleisten. Dies soll über das Versionsverwaltungssystem git und beispielweise in einer GitLab-Instanz umgesetzt werden.

Die Dokumentation soll im Rahmen des geförderten Projektes entsprechend in deutscher Sprache erfolgen. Die Entwicklungen des Source Codes gemäß der Standardkonvention in englischer Sprache.

### 7.2.3 Stakeholder und Kontexte – Horizontale Capability Layer

Die Untersuchung und Festlegung der Stakeholder, der benötigten Systeme und des Kontexts wird anhand der in der DIN SPEC 91357 vorgeschlagenen und in Kapitel 3.1.2 präsentierten 10 Layer vorgenommen.

#### 0. Field Equipment / Device Capabilities

Für die Pendler-App wird eine große Menge an Daten benötigt, zunächst werden diese aber auf die im Beispiel vorkommenden Verkehrsmittel beschränkt. Zur Bereitstellung der Prognose müssen für die langfristige Vorhersage zunächst historische Verkehrsdaten vorliegen. Diese müssen mit der aktuellen Verkehrssituation (Zählung von Fahrzeugen, Straßenauslastung und Auslastung der P+R-Plätze) und der Wetterlage anhand von Sensorik kombiniert werden. Insbesondere für die Verkehrsdaten können die vorliegenden Sensoren der Stadt und der Stadtwerke Osnabrück eingesetzt werden.

Auf der anderen Seite muss die App die regulären Pendelinformationen, wie Start- und Zielpunkte und Startzeiten bereitstellen, damit darauf basierend Handlungsempfehlungen und Routing generiert werden können. Weiterhin müssen auf der aktuellen Pendlerroute der Status und die Position des Pendelnden über die Sensoren des Smartphones (z.B. GPS) ausgelesen und übertragen werden.

#### 1. Communications, Network & Transport Capabilities.

Zur Datenübertragung der Messwerte der IoT-Devices kann die IoT-Plattform der SWON über MQTT eingesetzt werden. Die Verkehrssensoren und Detektoren der Stadt Osnabrück werden ebenfalls über ein eigenes Netzwerk in den Verkehrsrechner eingespeist. Die Kommunikation mit den Pendelnden sowie der Datenaustausch mit der UDP erfolgt dagegen über normale Netzwerke wie Internet und Mobilfunk, sodass dafür kein weiteres internes Netzwerk bereitgestellt werden muss.

#### 2. Device Asset Management & Operational Service Capabilities.

Auch für diese Schicht wird die Geräteinfrastruktur und Verwaltung bereits über die IoT-Plattform und den Verkehrsrechner umgesetzt. Neue IoT-Geräte (beispielsweise Sensoren zur Erfassung der Auslastung von P+R-Plätzen) können im Rahmen dessen über die IoT-Plattform der SWON integriert werden. Eine Erweiterung der Infrastruktur der Stadt Osnabrück ist im Rahmen dieses Projektes nicht vorgesehen.

#### 3. Data Management & Analytics Capabilities.

Für das Datenmanagement und die Datenanalyse werden die Urbane Datenplattform und der IIP-Server aufgebaut. Auf dem FROST- und Verkehrsserver werden Rohdaten bereitgestellt. Weiterhin können andere Daten in den verschiedenen Modulen des IIP-Servers entstehen. Anhand dieser Daten können Optimierungen vorgenommen werden, indem diese Daten in ein

gemeinsames Modell überführt und bereitgestellt werden. Diese Daten können einerseits historisch analysiert und dargestellt und andererseits für das Training diverser Verkehrsmodelle zur Prognose und Simulation sowie der Routenberechnungen für Handlungsempfehlungen eingesetzt werden.

#### **4. Integration Choreography and Orchestration Capabilities.**

Insbesondere der IIP-Server dient dazu, die Rohdaten und die Erkenntnisse aus den einzelnen Modulen zu integrieren und die Submodule zusammenzuführen, um daraus smarte Daten und smarte Applikationen generieren zu können. Dazu gehört insbesondere die Erstellung der Handlungsempfehlung und Bereitstellung für die Pendler-App, auf Basis der Berechnungen der übrigen Module.

#### **5. Generic City & Community Capabilities.**

Hiebei handelt es sich um die Schicht zur Einhaltung von Standards, um intermodales Pendeln zu ermöglichen und die Möglichkeit offen zu halten anhand dieser Standards auch überregionales Routing und Pendeln beispielsweise auch in Verbindung mit anderen Urbanen Datenplattformen unterstützen. Dazu sind einheitliche Datenmodelle und Schnittstellen zu definieren. Beispielsweise kann ein Austausch der Busfahrpläne und -verspätungen über das standardisierte Austauschformat GTFS (General Transit Feed Specification) erfolgen [Go23]. Die Informationen über P+R-Plätze können wiederum als definiertes Datenmodell über die SensorThings API eines Frost-Servers gespeichert und abgefragt werden.

#### **6. Specific City & Community Capabilities.**

Diese Schicht umfasst die Anbindung und Bereitstellung städtespezifischer Daten und Dienstleister, um die Anwendungsfälle auch speziell für die vorliegende Infrastruktur und Interessen der Stadt Osnabrück zu verbessern. Diese Daten und Dienste sind ebenfalls an die Datenmodelle und Schnittstellen anzupassen und anhand dieser zusammen zu entwickeln. Für den vorgestellten Beispielanwendungsfall umfasst dies, neben den in der Schicht 0 durch Sensorik erfassten Daten, insbesondere regionale Busfahrpläne, Busauslastung, Termine (beispielsweise Großveranstaltungen oder Ferien) und Wetterdaten (Mikroklima und Wetterservices), die in die Urbane Plattform integriert werden müssen.

#### **7. Stakeholder Engagement & Collaboration Capabilities.**

Die Integration und Zusammenarbeit von und mit neuen Dienstleistern und Businessmodellen ist Teil dieser Schicht. Konkret für diesen Anwendungsfall könnte dies die Anbindung neuer Dienstleister alternativer Verkehrsmittel, wie Anbieter von E-Scootern, Parkhausbetreiber als weitere Parkmöglichkeiten für den Umstieg auf andere Verkehrsmittel oder Fahrradverleihe sein. So lassen sich dem Pendler verschiedene neue Alternativen anbieten, die je nach Präferenzen oder Wetterlage vielseitigere Möglichkeiten bieten können.

## 8. Privacy & Security Capabilities.

Hier sind insbesondere die Anonymisierungsmodule, die im Kapitel 3.1.4 des Berichts [Se23] vorgestellt wurden, herauszustellen. Diese werden bereitgestellt und sind in den einzelnen Subsystemen zu integrieren. Insbesondere sind diese direkt auf den Geräten unterzubringen, wie auf den Verkehrszählkameras oder in der Pendler-App. Weitere Methoden, wie die Rechteverwaltung und der verschlüsselte Datenaustausch sind in den einzelnen Modulen wie Datenbanken und Schnittstellen selbst, anhand der definierten Standards, umzusetzen und eine Grundvoraussetzung in der Entwicklung. Wie in den Architekturvorlagen der DIN SPECS betreffen sie deshalb auch hier alle Komponenten und Schichten.

## 9. Common Service Capabilities.

Hierbei geht es um die finale Bereitstellung der Dienste rund um die Optimierung des Pendlerverkehrs. Einerseits werden die beschriebenen Infrastrukturen der SWON als auch die aktuelle Verkehrssteuerung beibehalten. Neu bereitgestellt werden müssen die zu entwickelnden Komponenten des IIP-Projektes. Dazu gehört einerseits die Serverinfrastruktur zum Hosting der neuen Services, als auch die mobile Pendler-App, die das Frontend und die Schnittstelle zum Pendler bietet. Beide müssen eine ausreichende Echtzeitfähigkeit unterstützen, damit dem Pendler rechtzeitig Handlungsempfehlungen und daraus resultierend eine zuverlässige und aktuelle (intermodale) Pendlerroute präsentiert wird. Die tatsächliche Bereitstellung der Serverinfrastruktur und das Hosting der App-Services ist zu diesem Zeitpunkt noch nicht vollständig auf die verschiedenen Projektpartner zugeordnet.

Anhand dieser 10 Schichten konnten die erforderlichen Daten, Aufgaben der Stakeholder und nötigen Teilsysteme aufgezeigt werden. Konkretere Lösungsansätze und ein erster Architekturentwurf mit genauer definierten Teilsystemen und Modulen wird in den nachfolgenden Abschnitten vorgenommen.

### 7.2.4 Lösungsstrategien

Nachfolgend werden erste Lösungsstrategien für die in Kapitel 7.2.1 vorgestellten Qualitätsziele aufgezeigt.

**Sicherheit.** Zur Einhaltung der Sicherheitsvoraussetzung sollen zum einen nur gesicherte Kommunikationskanäle wie HTTPS eingesetzt werden. Durch die Bereitstellung von Anonymisierungsmodulen wird sichergestellt, dass personenbezogene Daten nie das Aufnahmemodul dieser Daten verlässt. Alle Datenbankmanagementsysteme, die Nutzerdaten verwalten, müssen durch ein Berechtigungssystem geschützt werden. Passwörter dürfen nie im Klartext, sondern nur gehashed gespeichert werden. Weitere Sicherheitsansätze, wie sie in der Modulbeschreibung in Kapitel 3.1.4 des Berichts [Se23] angedeutet wurden, können dem Bericht [Pa23] detaillierter entnommen werden.

**Erweiterbarkeit.** Für die Erweiterbarkeit wird das Prinzip der Modularität vorgesehen, sowie die Verwendung einheitlicher Schnittstellen. Für die Arbeit mit dem Verkehrsserver kommt der Standard OCIT-C zum Einsatz, der in Kapitel 4.2 beispielhaft eingesetzt wurde. Der Datenaustausch mit dem FROST-Server, dem Kern der Urbanen Datenplattform, erfolgt über die SensorThings API, die ebenfalls öffentlich und gut dokumentiert ist, wie in Kapitel 4.1 vorgestellt wurde. Weitere allgemeine Datenanfragen sollen im Allgemeinen über eine REST-Schnittstelle umgesetzt werden. Diese zeichnet sich über eine einfache Struktur aus und kann über Tools wie Open API einfach dokumentiert und getestet werden [Li23]. Dadurch dass eine REST-Schnittstelle über einzelne Abfragen umgesetzt ist, lässt sich eine lose Kopplung einfacher umsetzen, da keine komplexe Verbindungsaufbauten und Protokolle implementiert werden müssen. Komponenten und Funktionalitäten können dadurch einfacher ausgetauscht werden. Weiterhin kann eine REST-API versioniert werden, sodass neue Features bereitgestellt, aber eine Kompatibilität zu älteren Versionen der Softwarekomponenten bestehen bleiben kann, bis diese aktualisiert und angepasst sind. Durch eine Modularisierung der Softwarekomponenten wird ebenfalls eine einfachere Wartbarkeit erreicht, da kleinere Komponenten einfacher getestet werden können.

**Interoperabilität.** Die Interoperabilität wird insbesondere durch die Orientierung der Architektur an den DIN-SPECs gefördert, auf denen auch die Architektur anderer Urbanen Plattformen orientiert sind. Dies wird weiterhin durch den Einsatz der im vorherigen Punkt eingesetzten Schnittstellen unterstützt, sodass auch fremde Komponenten einfach mit dieser Plattform interagieren können. Die Festlegung und Dokumentation einheitlicher Datenmodelle zum Austausch der Daten verschiedener Plattformen und Module, wird ebenfalls dadurch ein Metadatenmanagement wie InGrid unterstützt.

**Skalierbarkeit.** Zum Erreichen der Skalierbarkeit soll Docker eingesetzt werden. Die entwickelten Container können mit Kubernetes verwaltet und orchestriert werden. Beide Technologien wurden in Kapitel 3.5 des Berichts [Se23] vorgestellt. Mit Kubernetes kann insbesondere eine horizontale Skalierung leichter umgesetzt werden, da neue Geräte mit zusätzlichen Containern hinzugeschaltet und verteilt werden können, falls Lastspitzen oder eine dauerhaft höhere Nutzung verschiedener Systeme auftreten. Eine vertikale Skalierung ist allerdings ebenfalls möglich. Sollten einzelne Kubernetes Nodes aufgerüstet werden, können auf diesen zusätzliche Container laufen, da mehr Ressourcen bereitgestellt sind. Kubernetes wird ebenfalls in anderen Urbanen Datenplattformen eingesetzt und hat sich daher bewährt.

**Benutzbarkeit.** Die vorherigen Qualitätsziele spielen auch bei der Benutzbarkeit eine Rolle. Zunächst kann durch die Skalierbarkeit ein zuverlässiger Ablauf garantiert werden. Weiterhin lassen sich durch die Erweiterbarkeit im Zusammenhang mit der Modularität einfacher neue Funktionalitäten hinzufügen, die den Nutzungsumfang der Applikation für Pendler verbessern. Durch die Interoperabilität mit anderen Systemen können diese ebenfalls leichter integrierten

werden, sodass Nutzer auf bekannte Systeme und Funktionalitäten, aufgrund eingehaltener Standards, zugreifen können.

Trotz dieser Möglichkeiten sollte der Nutzer nicht mit Funktionalitäten und Entscheidungsmöglichkeiten (wie zu viele Auswahlmöglichkeiten für Pendlerwege oder Anreize) überfordert werden. Da eine möglichst große Anzahl und Breite von Pendelnden angesprochen werden soll, muss die Pendler-App und andere Tools möglichst unkompliziert funktionieren. Dafür sind auch Usability-Standards heranzuziehen und zu definieren, die dann dokumentiert und einheitlich umgesetzt werden. Diese sollten die Bedürfnisse aller Persona berücksichtigen. Neben den wichtigen Softwaretests der einzelnen Komponenten, die einen robusten Ablauf garantieren sind daher auch Usability-Tests durchzuführen.

### 7.2.5 Komponentensicht

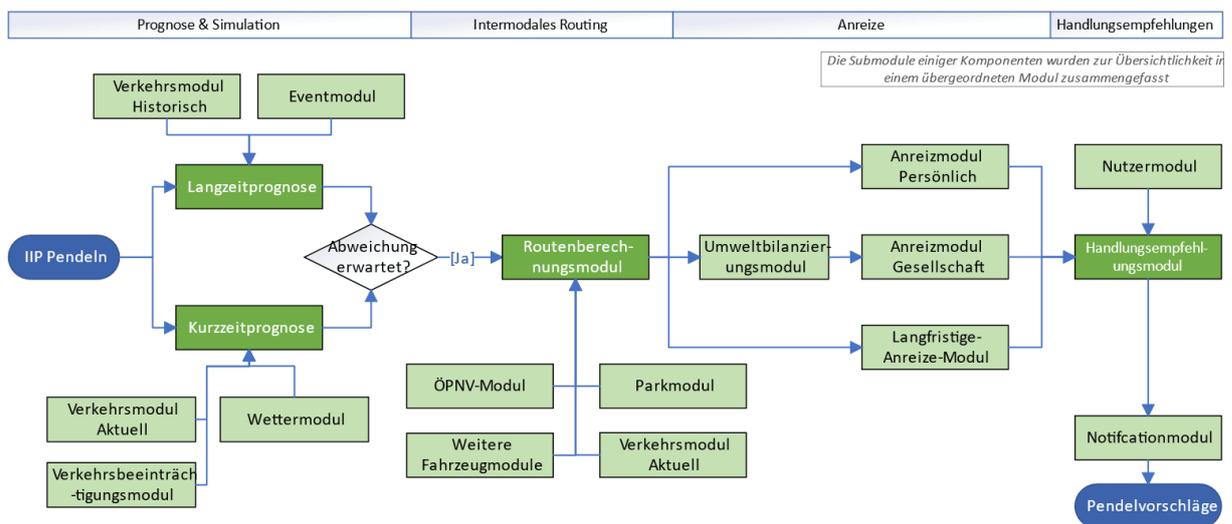


Abbildung 27 - Komponentensicht der Architektur für den Beispielanwendungsfall

Abbildung 27 zeigt den Ablauf in einer Komponentensicht der Architektur anhand der involvierten Softwarekomponenten, die im IIP-Server verortet sind und entwickelt werden (eine größere Version der Abbildung befindet sich im Anhang). Nachfolgend werden diese, zusammen mit den technischen Rahmenbedingungen, vorgestellt.

**Eventmodul.** Großveranstaltungen oder langfristige Events wie die Maiwoche können den Verkehr beeinflussen. Dies kann in Form von Straßensperren, gesperrten oder volleren Parkplätzen oder einem generell erwartbaren höheren Verkehrsaufkommen in einem Stadtbereich auftreten. In einigen Fällen können zusätzliche Sonderbusse Auswirkungen auf den Pendelwege nach sich ziehen. Für die Simulation und Prognose ist eine Berücksichtigung dieser

Termine erforderlich. Das Eventmodul muss daher in der Lage sein, Veranstaltungen und Termine in einer Datenbank zu speichern, verwalten und in einer Schnittstelle bereitzustellen. Dafür muss ein geeignetes Datenmodell erstellt werden, das die Informationen wie Orte der Veranstaltung, erwartete Besucherzahlen innerhalb der Stadt und der Auswirkungsbereich, bereitstellt. Das Datenmodell ist zusätzlich im Metadatenmanagement zu hinterlegen. Für die Speicherung dieses Modells ist eine eigene Datenbank nötig, da sich die Datenstruktur nicht für die SensorThings API eignet. Als Schnittstelle zur Abfrage der Veranstaltung kann REST über HTTPS eingesetzt werden. So kann für die Simulation und Prognose ein spezieller Use-Case allein für die Auswirkungen von Veranstaltung umgesetzt werden.

**Verkehrsbeeinträchtigungsmodul.** Die Verkehrsbeeinträchtigungen sind mit den Einschränkungen durch Veranstaltungen verwandt, haben allerdings andere Schwerpunkte und Eigenschaften. So werden teilweise einzelne Spuren gesperrt, sodass einige Verkehrsdetektoren keine Werte und andere wiederum etwa doppelt so viele Fahrzeuge zählen. Dies sollte daher nicht als Ausschlag oder Sonderfall eines kurzfristig Auftreten eines Verkehrssonderfalls interpretiert werden. Außerdem gehören hierzu auch aktuelle Verkehrsbeeinträchtigungen von Unfällen, die ein charakteristisches Umfahrvverhalten des Pendlerverkehrs nach sich ziehen. So kann als ein Anwendungsfall zur Optimierung im Bereich des Verkehrsrechners bereits präventiv auf andere Signalprogramme auf den Ausweichstraßen mit erwartbarer Mehrbelastung geschaltet werden, bis die Unfallstelle wieder freigegeben ist. In diesem Fall kann sowohl beim Auslesen der Daten als auch beim Einspeisen der Vorschläge in den Verkehrsserver, OCIT-C eingesetzt werden. Dieses Modul muss eine Echtzeitanforderung im Rahmen der Aktualisierungsintervalle der Verkehrsdetektoren umsetzen. Bei den Aktualisierungsintervallen der Detektorwerte handelt es sich aktuell um fünf Minuten.

**Wettermodul.** Es existieren zwei Hauptkategorien von Wetterdatenquellen, die unterschiedliche Informationen bereitstellen. Zum einen Messungen von lokalen Wetterstationen, die einzelne Sensorwerte liefern. Zum anderen Wetterinformationen und -prognosen von Wetterdiensten, für regionale Gebiete mit größerer Auflösung. Beide müssen in ein gemeinsames Datenmodell überführt werden, um damit beim intermodalen Routing Pendlern nur für die Wettersituation geeignete Verkehrsmittel vorzuschlagen. Das umfasst mindestens eine Vorhersage für den nächsten Arbeitstag. Weiterhin muss auch für eine spezifische Pendlerroute evaluiert werden, ob der nächste Verkehrsmittelwechsel tatsächlich durchgeführt werden kann oder ob sich die Wettersituation geändert hat, sodass beispielsweise doch weiter im Bus gefahren werden sollte, obwohl der Umstieg auf ein Fahrrad der schnellere Weg wäre, um den Arbeitsplatz pünktlich zu erreichen. In einem weiteren Schritt sind neben den reinen Wetterinformationen wie Regen und Temperatur auch weitere Phänomene wie Glätte zu analysieren, die schwerer messbar, aber ebenfalls relevant sind.

Für dieses Modul kann eine REST-Schnittstelle eingesetzt werden. Die Schnittstelle muss zudem so flexibel sein, dass verschiedene Wetterdienste austauschbar und einfach integrierbar sind. Abschließend sind im Regelfall keine großen Rechenoperationen für die einzelnen Anfragen zu erwarten. Allerdings muss ein regelmäßiges Parsing der externen Wetterquellen vorgenommen werden. Welche Datenmodelle, Datenbanken, Schnittstellen und Wetterdienste im Projekt eingesetzt werden, ist Teil einer anderen Arbeit, die im Rahmen des IIP-Projektes entsteht [Br23].

**Verkehrsmodul.** Das Verkehrsmodul arbeitet eng mit dem Verkehrsrechner, der Simulation und Prognose zusammen. Es Modul kann in Submodule eingeteilt werden. Diese betreffen nicht alle die Berechnung einer individuellen Pendlerstrecke, sind allerdings die Basis dafür und werden deshalb kurz vorgestellt.

**Verkehrsteilmodul 1 - Verkehrsrechnerkommunikation.** Teilmodul 1 ist zuständig für die Überführung von Detektormesswerten des Verkehrsservers und Übertragung der Datenmodelle für die Urbane Datenplattform. Der Zugriff erfolgt, wie in Kapitel 4.2 vorgestellt, mit OCIT-C über SOAP. Die Aktualisierung erfolgt fortlaufend in 5-Minuten-Intervallen. Falls für einen Anwendungsfall historische Daten angefordert werden, die noch nicht in der UDP vorliegen, kann ebenfalls gecheckt werden, ob diese im Verkehrsrechner vorliegen, sodass sie nachträglich in die UDP integriert werden können. Dies kann im Hintergrund geschehen.

**Verkehrsteilmodul 2 - Verkehrsdatenverarbeitung.** In diesem Modul ist die Funktionalität zur Aufbereitung der Verkehrsdaten- und Messwerte für die Simulations- und Prognosemodule gekapselt. Dazu gehört die Datenfusion und -aggregation. Hierzu werden vom Modul die Schnittstellen zu der UDP mit der SensorThings API und zum Verkehrsrechner mit OCIT-C eingesetzt. Wie in Kapitel 3.4.1 des Berichts [Se23] erläutert, ist die Funktionalität für Operationen und Vorverarbeitung der Zeitreihendaten des FROST-Servers begrenzt. Ebenfalls sollte der Verkehrsrechner der Stadt Osnabrück nicht zu stark mit Anfragen belastet werden, da die Hauptressourcen für die Verkehrssteuerung vorgesehen werden. Daher kann in diesem Modul eine eigene Datenhaltung als eine Art Caching erforderlich sein. Hierzu können Zeitreihendatenbanken, zum Beispiel InfluxDB, in einem eigenen Container eingesetzt werden. Die im ersten Teilmodul erfassten Daten können zusätzlich in die InfluxDB überführt werden, sodass sie für diese Anwendungsfälle zur Verfügung stehen. Dadurch wird das Netzwerk und die anderen Datenbanken entlastet. Wenn diese Datenbank nah an den angeschlossenen Modulen angesiedelt ist, ist eine Kommunikation über Internet mit den anderen Datenbanken nicht mehr nötig, sodass Prognose, Simulation und Training effizienter auf die Daten zugreifen können. Die Schnittstellen dieses Teilmoduls müssen so gestaltet werden, dass die InfluxDB entsprechend gekapselt wird, damit die übrigen Module nicht explizit mit der InfluxDB-Syntax arbeiten müssen. InfluxDB und andere Zeitreihendatenbanken bietet oft Möglichkeiten zur Skalierung. Hier ist vor allem eine vertikale Skalierbarkeit möglich, damit das Modul den Anforderungen der Simulation entsprechen kann.

**Fahrzeugmodule (z.B. ÖPNV-Modul, Parkmodul).** Neben den allgemeinen Verkehrsdaten müssen auch die übrigen Daten der verschiedenen Fortbewegungsmittel öffentlicher und privater Anbieter integriert werden, um ein vielseitiges intermodales Pendeln bieten zu können. Dazu müssen die einzelnen Schnittstellen der Anbieter angesprochen und in kompatible Datenformate für die UDP übertragen werden. Abfragen zu aktuellen Zuständen bei den jeweiligen Diensten sind ebenfalls hier gekapselt. Die einzelnen Module können als eigene Container deployed werden und sind deshalb einfach in die Plattform integrierbar, da sie gegen bekannte Schnittstellen, wie etwa die SensorThings API entwickelt werden. Die verschiedenen Module haben jeweils verschiedene Aktualisierungsintervalle, die von den Datenquellen abhängen. Zu beachten ist, dass diese genau genug sind, um daraus ein verlässliches Routing zu erstellen.

**Kurzzeitprognose und Langzeitprognose.** Die beiden Prognose- und Simulationsmodule greifen insbesondere auf die Schnittstellen des Verkehrsmoduls zu. Hierbei handelt es sich um die Module mit dem höchsten Rechenaufwand und Rohdatenbedarf. Dementsprechend müssen diesen Modulen ausreichende Ressourcen bereitgestellt werden. Die Zusammenarbeit mit dem Verkehrsmodul sollte eng sein, um unnötige Overheads und Lasten auf das Gesamtsystem zu vermeiden. Auf die eingesetzten Techniken und Technologien wird in einer anderen Arbeit eingegangen und eine Prognose entwickelt. Die aktuellen Ansätze werden im Minimal Working Example dokumentiert [Sc23].

Wenn in diesem Bereich erwartbare Abweichungen festgestellt werden, beginnt der Programmteil der Handlungsempfehlungsberechnung mit den dazugehörigen Teilsystemen.

**Routenberechnungsmodul.** Das Routenberechnungsmodul beinhaltet die Funktionalitäten zur Berechnung möglicher Pendelrouten zwischen Startpunkt und Arbeitsplatz. Dies ist nötig, um darauf basierend Vorschläge zu geben, die dann für Handlungsempfehlungen ausgewählt werden können. In diesem Modul werden die Verkehrsstörungen (Baustellen, Unfälle und Veranstaltungen) aus den bisher vorgestellten Modulen berücksichtigt. Es arbeitet daher mit deren Schnittstellen zusammen. Die Routenberechnung erfordert eine hohe Verfügbarkeit und geringe Antwortzeit, da nur darauf basierend Handlungsempfehlungen und Anreize berechnet werden können. Da die Routen grundsätzlich unabhängig voneinander berechnet werden können, eignet sich dieses Modul gut für eine horizontale Skalierung. Ein detailliertes Routing von Kreuzung zu Kreuzung während des Pendelweges wie etwa bei Navigationssystem, ist nicht Teil dieses Moduls. Da Pendelrouten intermodal vorgeschlagen werden, müssen ebenfalls Schnittstellen zu den Informationen der anderen Verkehrsmittel angesprochen werden.

**Umweltbilanzierungsmodul.** Das Umweltbilanzierungsmodul untergliedert sich in vier Subsysteme, die wiederum als einzelne Module gekapselt werden können. Für dieses Modul ist besonders die Intermodalität und Erweiterbarkeit relevant, sodass neue Verkehrsmittel in das

System integriert und andere Regionen und OUP angeschlossen werden können. Diese Anforderung überträgt sich auch auf alle Teilmodule. Die Teilmodule 2-4 betreffen nicht direkt die Berechnung auf eine Pendlerroute bezogen, werden aber durch diese beeinflusst und daher ebenfalls definiert.

**Umweltbilanzierungsteilmodul 1 - Routenbilanzierung.** Für eine gegebene Route werden die erwartbaren Emissionen und daraus resultierende Einsparungen berechnet. Hier werden die Routen- und Straßeninformationen mit den Verkehrsmitteln und den Eigenschaften, die beispielsweise in der HBEFA hinterlegt, fusioniert. Die Schnittstelle ist dementsprechend zu definieren. Für Teilstrecken oder feste Routings kann gegebenenfalls ein eigene Datenhaltung durch ein Caching implementiert werden, sodass keine wiederholten aufwändigeren Berechnungen, beispielsweise für Teilstücke der Hauptverkehrsstraßen, durchgeführt werden müssen. Der Vorteil dieses Modul ist die einfache Möglichkeit es zu skalieren, da Routen unabhängig voneinander berechnet werden können.

**Umweltbilanzierungsteilmodul 2 - Flottenbilanzierung.** Dies Modul umfasst die Bilanzierung auf Flottenebene. Zusätzlich zu den Aspekten des ersten Teilmoduls muss dieses Modul auf bereits absolvierte Routen aufrufen können. Weiterhin wird dieses Modul auf das Verkehrsmodul zugreifen, um die Zählinformationen der einzelnen Fahrzeugtypen für verschiedene Zeiträume einzubeziehen. Hier kommen OCIT-C und die SensorThings API zum Einsatz. Über OCIT-C können zudem die Schadstoffwerte einzelner Messstationen ausgewertet werden. Ergebnisse und Historiendaten zu diesen Bilanzierungsberechnungen können in einer eigenen Schnittstelle des Moduls, beispielweise über REST bereitgestellt werden. Neue Anfragen können unmittelbar berechnet werden. Im Hintergrund können allerdings automatische Berechnungen für bestimmte Zeiträume oder relevante Streckenabschnitte oder automatisch erstellte Reporte generiert werden.

**Umweltbilanzierungsteilmodul 3 - Prognose.** Auf Basis der aktuellen Informationen und Historiendaten können in einem späteren Schritt ebenfalls Prognosen für zu erwartende, die Schadstoffgrenzwerte überschreitende, Messperioden berechnet werden. Diese Informationen können dann über Push-Service einzelnen andern Modulen zugestellt werden.

**Umweltbilanzierungsteilmodul 4 - Gesamtbilanzierung.** Abschließend muss ein Modul zur geforderten Gesamtbilanzierung des IIP-Projektes implementiert werden. Dies fusioniert die Daten der anderen Teilmodule, um neben den Emissionen und Einsparungen der intermodalen Pendler Routen auch den Energieverbrauch der Infrastrukturen berechnen und einzubeziehen, und daraus Gesamtbilanzen generiert. Dieses Teilsystem interagiert weniger datengebend mit anderen Modulen und ist daher in diesem Bereich in sich geschlossen und wird eher für die Generierung von Reports oder Anzeige von Informationen eingesetzt.

**Langfristige-Anreize-Verwaltungsmodul.** Das Modul kapselt die Funktionalitäten, falls ein langfristiges Anreizsystem mit „Belohnungen“ wie in Kapitel 3.1.2 des Berichts [Se23] erläutert, etwa ähnlich den Klima-Herzen aus Kapitel 3.2.2, umgesetzt wird. Hierfür ist wiederrum

eine eigene Datenhaltung nötig, die die Nutzerdaten mit den zurückgelegten Strecken kombiniert. Dies stellt auch die Anonymisierung wieder in den Fokus. In diesem Modul steht die Konsistenz stärker als die Echtzeitanforderung im Vordergrund, damit die Ergebnisse für den Pendelnden zuverlässig berechnet werden und keine Fortschritte verloren gehen. Für die Interoperabilität und Erweiterbarkeit ist ein einheitliches Datenmodell zu definieren, um anderen Anbietern das Hinterlegen von Angeboten zu erleichtern und zu fördern. Ob ein solches langfristiges Anreizsystem eingeführt wird, ist zum aktuellen Zeitpunkt noch offen. Das Auto an einem P+R-Platz abzustellen und auf den ÖPNV zu wechseln, ist ein Fall der „belohnt“ werden kann.

Das **Anreizmodul** untersucht anhand der möglichen alternativen Routen verschiedene Anreiztypen. Für die persönlichen Vorteile wie in dem Beispielanwendungsfall, können die erwarteten Zeitunterschiede bei Nutzung verschiedener intermodaler Routen und die schnellste Ankunft untersucht werden (**Anreizteilmodul 1 – Persönliche Anreize**). Dabei erfolgt eine enge Zusammenarbeit mit dem Routingmodul. Für die gesellschaftlichen Anreize können in Arbeit mit den Umweltbilanzierungsmodulen verschiedene Optimierungen dahingehend berechnet und verglichen werden (**Anreizteilmodul 2- Gesellschaftliche Anreize**). Schließlich kann für ein eventuelles langfristiges Anreizsystem in Zusammenspiel mit dem Langfristige-Anreize-Verwaltungsmodul (**Anreizmodul 3 – Langfristige Anreize**) eine den persönlichen Präferenzen orientierten Route vorgeschlagen werden. Dieses Modul erfordert ebenfalls eine hohe Verfügbarkeit, die die Anreize eine Grundlage des Gesamtsystems sind. Eine Skalierbarkeit ist auch hier wieder horizontal realisierbar, da die Berechnungen insbesondere für verschiedene Nutzer unabhängig voneinander durchführbar sind. Dieses Modul arbeitet eng mit den beschriebenen anderen Systemen zusammen, in diesem Fall kann eine engere Kopplung nicht gänzlich vermieden werden.

**Handlungsempfehlungsmodul.** Die errechneten möglichen Anreize können nun zusammen mit den Routen in das Handlungsempfehlungsmodul überführt werden. Dies kapselt die Funktionalitäten, um die berechneten Ergebnisse in passende Vorschläge für die Nutzer aufzubereiten und an das Notificationmodul zur letztendlichen Weitergabe und Benachrichtigung an die Pendler weiterzugeben.

**Nutzermodul.** Das Nutzermodul verwaltet die Nutzerdaten entsprechend den Datenschutzanforderungen. Für die Nutzerdaten ist eine eigene Datenbank einzurichten. Bei der Speicherung dieser ist zwischen Funktionalität und Komfort abzuwägen, beispielsweise ob Start- und Zielpunkt der Pendelroute und die Verkehrsmittelpräferenzen dauerhaft gespeichert werden, oder nur lokal auf den Endgeräten. Daher muss das Datenmodell fest definiert werden, sodass es auch auf andere Umgebungen übertragbar ist. Für den Anwendungsfall ist dies beispielsweise die Information, ob der Nutzer nur den ÖPNV als Alternative nehmen möchte oder auch

bereit ist, vom P+R-Platz mit einem Leihrad weiterzufahren. Falls nicht, muss die Handlungsempfehlungen dahingehend vorgefiltert werden.

**Notificationmodul.** Das Notificationmodul kapselt die Funktionalitäten des Versendens der Handlungsempfehlungsnachrichten über Push-Dienste, falls Änderungen an der regulären Pendlerroute festgestellt und neue Handlungsempfehlungen berechnet wurden. Dieses Modul erfordert eine sehr hohe Verfügbarkeit und Zuverlässigkeit, da die Pendler-App auf diese Updates angewiesen ist, damit rechtzeitig auf Änderungen hingewiesen werden kann. Die DIN SPEC 91367 empfiehlt beispielsweise eine Uptime von mindestens 99.9%. Ein dauerhaftes manuelles Checken durch den Pendelnden ist nicht zielführend.

**Anonymisierungsmodule.** Die Anonymisierungsmodule werden mit einem in Kapitel 3.1.4 des Berichts [Se23] vorgestellten Baukastensystem zur Verfügung gestellt. Dort wurden bereits einige Funktionalitäten und in Kapitel 7.2.4 Lösungsstrategien vorgestellt. Das Baukastensystem erfordert eine gute Dokumentation und den Einsatz offener Schnittstellen und Funktionen, um eine einfache Wiederverwendbarkeit und Flexibilität zu gewährleisten.

Durch das Zusammenspiel aller vorgestellten Komponenten kann der vorgestellte Beispielanwendungsfall umgesetzt werden und darüber hinaus auf die weiteren Anforderungen und Anwendungen des IIP-Projektes erweitert werden. Durch die Verwendung der vorgestellten Schnittstellen und die Aufteilung in kleinere Komponenten (die auf Softwareebene in weitere Subkomponenten unterteilt werden), kann die erforderliche lose Kopplung und eine einfachere Erweiterung und die Interoperabilität sichergestellt werden. So können einfach neue Komponenten, wie neue Oberflächen, neue Verkehrsmittel, weitere Datenbanken, die ein bestimmtes Subsystem benötigt, oder ganze neue Anwendungsfälle hinzugefügt werden. Das Gesamtsystem wird dann über eine Kubernetes-Serverinfrastruktur mit containerisierten Softwarekomponenten und verschiedenen Applikationen und Frontends bereitgestellt.

### 7.3 Allgemeine Architekturübersicht

Abbildung 28 zeigt den Entwurf einer Gesamtarchitekturübersicht mit den geplanten Komponenten zur Arbeit mit der Urbanen Datenplattform und daraus resultierenden smarten Applikationen wie der Pendler-App. Dabei folgt die allgemeine Struktur die der anderen vorgestellten Offenen Urbanen Plattformen aus Kapitel 3.2, die sich wiederum auf die Struktur der vorgestellten DIN SPECS aus Kapitel 3.1 orientieren. Die einzelnen Komponenten werden nachfolgend erläutert.

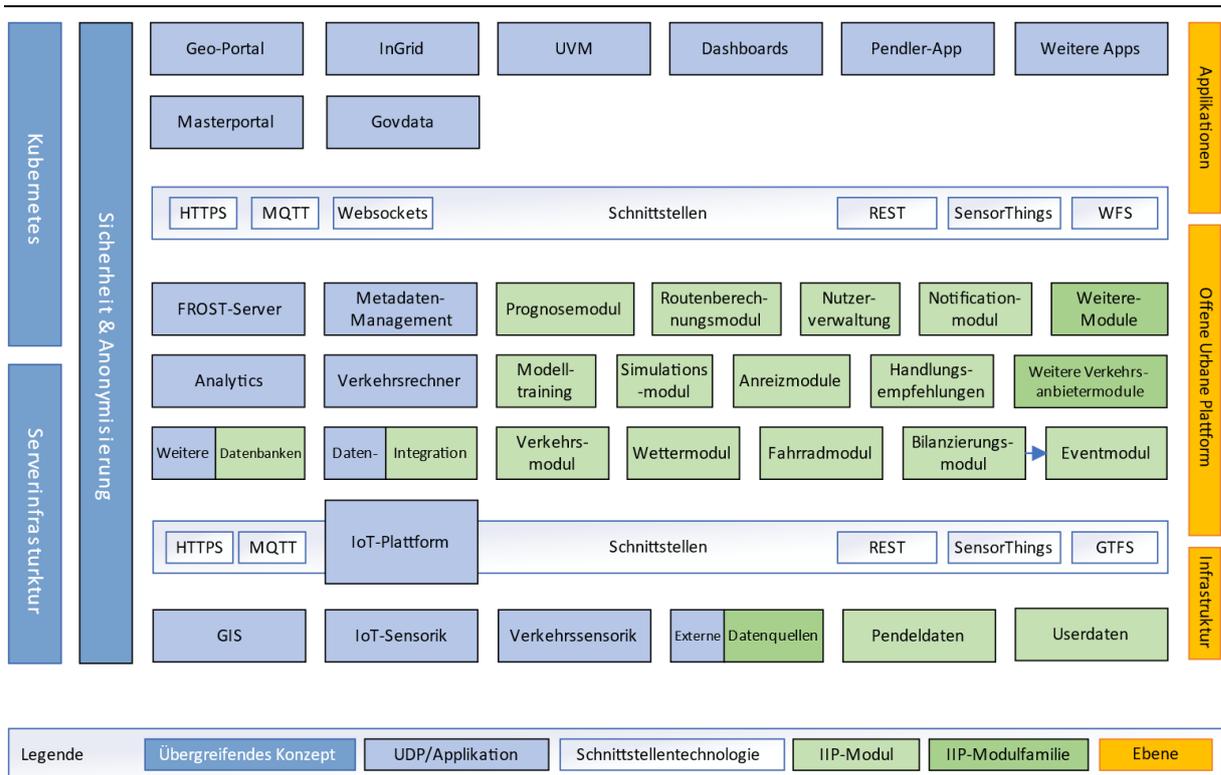


Abbildung 28 - Gesamtarchitekturübersicht

Zunächst wurden die einzelnen *groben* Schichten aus DIN SPEC 91357 umgesetzt. Im untersten Bereich befindet sich die **Infrastrukturschicht**, sowohl in Form der (IoT-) Sensorik, als auch den Geodaten. Neben den bereits existierenden Datensätzen, umfasst dies auch neu zu erschließende Datenquellen, sowie die anfallenden und erhobenen Daten aus der Pendler-App, die in das Gesamtsystem zurückgeführt werden können, wie es auch im Kreislauf aus Abbildung 11 dargestellt ist, der hier aus Übersichtlichkeit und durch den Fokus auf die Komponentenstruktur weggelassen wurde.

Die Infrastrukturschicht kommuniziert über verschiedene Schnittstellen und Protokolle mit der Offenen Urbanen Plattform, die aus der UDP und dem IIP-Server besteht. Die Kommunikation erfolgt über die in den Lösungsansätze vorgestellten Standards, die möglichst für alle Datenquellen und Schnittstellen der Komponenten eingehalten werden sollen, sofern es die Datenmodelle zulassen. Besonders hervorzuheben sind hier MQTT und REST über HTTPS für die meisten Datenquellen.

Die Kernschicht stellt die **Offene Urbane Plattform** dar, welche sowohl für die Datenhaltung, hauptsächlich in der UDP, als auch für die Datenverarbeitung mittels des IIP-Servers, zuständig ist. Wichtige Datenhaltungskomponenten sind insbesondere der FROST-Server zur Verwaltung der IoT-Daten. Weiterhin ist der Verkehrsrechner ein wichtiger Verkehrsdatenlieferant. Über ein Metadatenmanagement können die verwalteten Daten beschrieben und damit auch

dokumentiert, auffindbar und durchsuchbar gemacht werden. In den *Platzhalter*komponenten *Weitere Datenbanken* sind mögliche Zeitreihendatenbanken oder objektorientierte Datenbankmanagementsysteme verortet, falls sie für neue Anwendungsfälle und Datenquellen benötigt werden, die nicht in ein SensorThings-Format überführbar sind. *Datenintegrations*module können direkt in der UDP implementiert werden. Auf diese Weise können Daten aus der IoT-Plattform oder dem Verkehrsrechner bei der Übertragung in den FROST-Server direkt in die korrekten Formate konvertiert werden. Beide Modultypen können auch im Bereich des IIP-Servers eingesetzt werden, falls sie speziell für eines der Funktionsmodule benötigt werden, wie die hybride Darstellung symbolisiert.

Die in grün dargestellten Module bilden die Funktionalitäten des **IIP-Servers** ab, die den zweiten Teil der Offenen Urbanen Plattform bilden. Dazu gehören zum einen die in Kapitel 3.1 des Berichts [Se23] vorgestellten Kernmodule Simulation und Prognose, das Anreizsystem und die Ökobilanzierung. Diese werden in einige weitere Module unterteilt und durch weitere Module mit zusätzlichen Funktionalitäten, die für den Beispielanwendungsfall und die daraus erweiterte Pendler-App erforderlich sind, ergänzt. Dazu gehört unter anderem ein Wettermodul zur Erfassung und Berücksichtigung der Wetterdaten für die Wahl des geeigneten Verkehrsmittels. Beispielweise ist für die Arbeit mit diversen Fahrraddaten ebenfalls ein Modul vorgesehen. Zur Arbeit mit Pendlern ist ein Nutzerverwaltungsmodul und ein Notificationmodul nötig, das jene bei nötigen Routenupdates benachrichtigt. Im Zusammenhang mit dem Anreizmodul werden Handlungsempfehlungen in einem weiteren Modul berechnet. Dies integriert und fusioniert die Erkenntnisse aus dem Routing, der Ökobilanzierung und den möglichen Anreizen für die Nutzer. Außerdem sind auch hier weitere Platzhalter installiert, die beispielweise die Anbindung und Untersuchung weiterer Verkehrsmittel für ein intermodales Routing umsetzen. Darüber hinaus können auch allgemeine weitere Module in den IIP-Server integrieren, die die Funktionalitäten und intelligente Datenfusion für weitere smarte Applikationen von externen Anbietern kapseln. Ein weiteres Modul kann Analytics implementieren. Darin kann untersucht und getrackt werden, welche Funktionen und Daten in welchen Frequenzen abgefragt werden, an welchen Strecken häufig mit der App gependelt wird und welche Verkehrsmittel häufig für Handlungsempfehlungen herangezogen werden, was ebenfalls bei der Evaluation des Gesamtprojekts hilft.

Auf dieser Plattform setzt nun eine weitere Schnittschicht auf, die die Daten für die Applikationen bereitstellt. Der zentrale FROST-Server kann über HTTPS mit der SensorThings API oder MQTT angesprochen werden. Die einzelnen Schnittstellen der Module des IIP-Servers sollten möglichst über eine REST-API kommunizieren, um eine lose Kopplung und eine flexible Schnittstelle zu erhalten. Mit Geodaten kann entsprechend über den WFS-Standard gearbeitet

tet werden. In besonders intensiven Operationen und Funktionalitäten müssen gegebenenfalls auch andere Schnittstellentypen und Protokolle eingesetzt werden. Dies erfordert eine gute Dokumentation und vorher festzulegende Rahmenbedingungen.

Die mit den Schnittstellen arbeitenden Applikationen, zusammengefasst in der obersten **Applikationsschicht**, sind zum einen einfache Visualisierungskomponenten wie Geoportale oder das Masterportal, die verschiedene Datenquellen durch eigene Module anschließen können. Weiterhin gehören dazu die Tools für das Metadatenmanagement, die Kataloge und Suchfunktionalitäten für die vorliegenden Metadatenbeschreibungen bieten.

Daneben liegen die smarten Applikationen, die aus dem gesamten Datenbestand einen Mehrwert für die Pendler und Bürger bieten. Zum einen sind dies Pendlerdashboards. Zum anderen ist insbesondere die Pendler-App, die die vielen Module der Datenverarbeitung und Funktionalitäten des IIP-Servers nutzt, um die Pendler Routen intermodal und dadurch besser zu gestalten und auch einen Mehrwert für den gesamten Stadtverkehr generieren. Auch diese Schicht lässt sich durch neue Applikationen erweitern, was ebenfalls durch die gespeicherten und öffentlichen Daten der UDP, den wiederverwendbaren und öffentlichen Modulen des IIP-Servers und den festgelegten öffentlich standardisierten Schnittstellen unterstützt wird.

Zu diesen Schichten kommen die insbesondere in DIN SPEC 91367 definierten Schichten der Privacy und Sicherheit (8), sowie der Common Services (9). Privacy durch Anonymisierung und funktionale Sicherheit wurden bereits als querschnittliches Konzept und Grundvoraussetzung beschrieben. Weiterhin müssen alle Komponenten des IIP-Servers verlässlich bereitgestellt werden. Dies ist die Komponente der Kubernetes dargestellt, wie sich auch zusammen mit der Serverinfrastruktur als eigene Schicht in anderen OUP verschiedener Städte dargestellt ist. Der Kubernetes-Ansatz wurde ebenfalls in den Lösungsansätzen zur Erweiterbarkeit und Skalierbarkeit vorgestellt.

## 8 Literaturverzeichnis

- [Am23a] Amazon Web Services, I.: Time Series Forecasting Service – Amazon Forecast – Amazon Web Services. <https://aws.amazon.com/forecast/>, Stand: 29.05.2023.
- [Am23b] Amazon Web Services, I.: Weather Index - Amazon Forecast. <https://docs.aws.amazon.com/forecast/latest/dg/weather.html>, Stand: 29.05.2023.
- [BCY22] Bui, K.-H. N.; Cho, J.; Yi, H.: Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence* 3/52, S. 2763–2774, 2022.
- [Br23] Bröker, M.: Untersuchung von geeigneten Datenquellen für den IIP-Server der Hochschule Osnabrück, Osnabrück, 2023.
- [De23] Deutscher Wetterdienst: Wetter und Klima - Deutscher Wetterdienst - Leistungen - Open Data. <https://www.dwd.de/DE/leistungen/opendata/opendata.html>, Stand: 29.05.2023.
- [De23b] Deutscher Wetterdienst: Wetter und Klima - Deutscher Wetterdienst - Leistungen - Geodaten-Serverdienst. [https://www.dwd.de/DE/leistungen/geodaten\\_serverdienst/geodaten\\_serverdienst.html?nn=16102](https://www.dwd.de/DE/leistungen/geodaten_serverdienst/geodaten_serverdienst.html?nn=16102), Stand: 29.05.2023.
- [Di16]:2016, DIN SPEC 91340:2016-10 - Terminologie der intelligenten individuellen urbanen Mobilität.
- [Di17]:2017, DIN SPEC 91357:2017-12 - Referenzarchitekturmodell Offene Urbane Plattform (OUP).
- [Di19]:2019, DIN SPEC 91367:2019-04 - Urbane Mobilitätsdatensammlung für Echtzeitanwendungen.
- [Fr21] Fröhlich, J.: Die Digitale Datenplattform der Digitalstadt Darmstadt, Darmstadt, 2021.
- [Fr23a] FraunhoferIOSB: FROST-Server GitHub Repository. <https://github.com/FraunhoferIOSB/FROST-Server>, Stand: 10.05.2023.
- [Fr23b] FraunhoferIOSB: FROST-Server - Open-Source-Implementierung der OGC SensorThings API. <https://www.iosb.fraunhofer.de/de/projekte-produkte/frost-server.html>, Stand: 17.05.2023.
- [FZL16] Fu, R.; Zhang, Z.; Li, L.: Using LSTM and GRU neural network methods for traffic flow prediction: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). *IEEE*, S. 324–328, 2016.

- [Go23] Google: GTFS: Öffentliche Verkehrsdaten universell zugänglich machen. <https://gtfs.org/de/>, Stand: 10.07.2023.
- [He21] Heuser, L.: Datenplattform Darmstadt aus Sicht des Plattformanbieters, Darmstadt, 2021.
- [He22] Heller: Traffic Lights Data Hamburg - Usage Guide. [https://daten-hamburg.de/tlf\\_public/TLD\\_UsageGuide\\_V1.1.pdf](https://daten-hamburg.de/tlf_public/TLD_UsageGuide_V1.1.pdf).
- [Hi20] Staatliches Gewerbeaufsichtsamt Hildesheim: Untersuchung der Stickstoffdioxidkonzentration der verkehrsnahen Probeentnahmestellen des LÜN Osnabrück, Hildesheim, 2020.
- [IIP22] Projektpartner des IIP-Projekts: GVB IIP - Intelligenter Intermodaler Pendlerverkehr, Osnabrück, 2022.
- [JL22] Jiang, W.; Luo, J.: Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* 207, S. 117921, 2022.
- [KDJ18] Lana, I. et al.: Road Traffic Forecasting: Recent Advances and New Challenges. *IEEE Intelligent Transportation Systems Magazine* 2/10, S. 93–109, 2018.
- [Li23] The Linux Foundations: What is OpenAPI? <https://www.openapis.org/what-is-openapi>, Stand: 10.06.2023.
- [LMZ21] Li, M.; Zhu, Z.: Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 5/35, S. 4189–4196, 2021.
- [Ma23] de Maeyer, J.: Bright Sky API Documentation. <https://brightsky.dev/docs/#overview--data-origin>, Stand: 29.05.2023.
- [Me23] Meteomatics AG: Meteomatics' Approach Offers Unprecedented Accuracy of Weather Forecasts | Meteomatics. <https://www.meteomatics.com/en/news/meteomatics-approach-offers-unprecedented-accuracy-of-weather-forecasts/>, Stand: 29.05.2023.
- [Me23b] Meteomatics AG: Weather Forecast With Meteodrones Weather Drones | Meteomatics. <https://www.meteomatics.com/en/meteodrones-weather-drones/>, Stand: 29.05.2023.
- [Me23c] Meteomatics AG: Where Does Meteomatics Get Its Data From? <https://www.meteomatics.com/en/weather-data/>, Stand: 13.07.2023.
- [Mo23] MOTIONTAG: Das Analytics SDK, Stand: 10.07.2023.

- [MRS22] Meier, A. et al.: Baseline Assessment Urban Data Platform Freie und Hansestadt Hamburg (FHH), Hamburg, 2022.
- [NKW18] Nguyen, H. et al.: Deep learning methods in transportation domain: a review. IET Intelligent Transport Systems 9/12, S. 998–1004, 2018.
- [Oc18] OCIT Developer Group: OCIT-C\_Daten\_V2.0\_A01, Braunschweig, 2018.
- [Op21] Open Geospatial Consortium: OGC SensorThings API Part 1: Sensing Version 1.1. <https://docs.ogc.org/is/18-088/18-088.pdf>.
- [Op23] OpenWeather: Weather API - OpenWeatherMap. <https://openweathermap.org/api>, Stand: 29.05.2023.
- [Op23b] OpenWeather: Weather model - OpenWeatherMap. <https://openweathermap.org/technology>, Stand: 29.05.2023.
- [Op23c] OpenWeather: Accuracy and quality of weather data - OpenWeatherMap. <https://openweathermap.org/accuracy-and-quality>, Stand: 29.05.2023.
- [Os23] Stadt Osnabrück: Umweltsensitives Verkehrsmanagement. <https://geo.osnabrueck.de/uvm/?i=start>, Stand: 17.05.2023.
- [Pa19a] Stadt Paderborn: Zentrale Open Data Plattform. <https://digitale-heimat-pb.de/projekte/zentrale-open-data-plattform/>, Stand: 15.06.2023.
- [Pa19b] Stadt Paderborn: Projektdokumentation: Zentrale Open Data Plattform der Stadt Paderborn, Stand: 15.06.2023.
- [Pa23] Paßfeld, T.: Privacy durch Anonymisierung.
- [Re23] Reuwer, A.-K.: Datenquellenbestand IIP – Gesamtübersicht, 2023.
- [Re23b] Reuwer, A.-K.: Erschließung neuer Datenquellen, 2023.
- [Re23c] Reuwer, A.-K.: Zusammenfassung Datenquellenworkshop vom 06.06.23, 06.06.23.
- [Sc23] Schaffland, A.: Minimal Working Example IIP, Osnabrück, 2023.
- [Se23] Seidel, T.: Konzeption einer modularen Serverumgebung für die Datenerfassung und -verarbeitung der Nutzung verschiedener Verkehrsmittel und deren Kontexte. Wissenschaftlicher Projektbericht, Osnabrück, 2023.
- [St22] Stadtwerke Osnabrück Netz GmbH: Verkehrszählung Fahrspur Thing. [https://datenplattform.osnabrueck.de/wpcontent/uploads/2022/12/verkehrszaehlung\\_fahrspur\\_thing.pdf](https://datenplattform.osnabrueck.de/wpcontent/uploads/2022/12/verkehrszaehlung_fahrspur_thing.pdf), Stand: 17.05.2023.

- [St23] Stadtwerke Osnabrück Netz GmbH: Daten-API Osnabrück. <https://daten-api.osnabrueck.de/>, Stand: 17.05.2023.
- [St23c] Stadtwerke Osnabrück Netz GmbH: LoRaWAN als Basis für das Internet der Dinge. <https://www.stadtwerke-osnabrueck.de/geschaeftskunden/lorawan-internet-der-dinge>, Stand: 17.05.2023.
- [St23d] Stadtwerke Osnabrück Netz GmbH: Datenfenster Osnabrück. <https://daten-fenster.osnabrueck.de>, Stand: 17.05.2023.
- [Yu23] Yunex Traffic: Schnittstellen und Standards für Ihr Verkehrsmanagement. <https://www.yunextraffic.com/de/portfolio/intelligentes-verkehrsmanagement/yunex-traffic-verkehrszentralen/schnittstellen-und-standards/>, Stand: 12.07.2023.
- [YWW23] Yin, X. et al.: Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions. IEEE Transactions on Intelligent Transportation Systems 6/23, S. 4927–4943, 2022.