CONNECTED URBAN TWINS

City Science Lab
A Cooperation with the MIT Media Lab

Rico Herzog
M.Sc. Engineering and Policy Analysis

Die Urban Model Platform

Silos aufbrechen für Algorithmen

30.11.23

Partnerstädte:

Hamburg

Stadt Leipzig

Landeshauptstadt München

Gefördert durch:

Bundesministerium für Wohnen, Stadtentwicklung und Bauwesen

KFW

Übersicht

I. Idee & Konzept

II. Technische Umsetzung

Übersicht

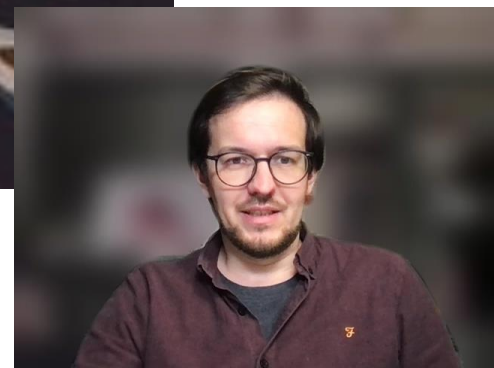
I. Idee & Konzept

1) Einführung & Hintergrund
2) Zielsetzung
3) Prototyp
4) Offene Fragen & nächste Schritte
5) Zusammenfassung

Einführung & Hintergrund

Was wäre, wenn?

Aktuelle Situation



GIF: giphy.com/sammhenshaw

Open Forecast
Für Luftqualität

PALM
+ andere
Klimamodelle

MATSim
+ andere
Mobilitätsmodelle

CUT

[1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11]

Agentenbasierte Modelle
(MESA, GAMA, NetLogo,
uvm. …)

HERoS
+ andere epidemio-
logische Modelle

GPT4
+ andere
LLMs

System Dynamic Modelle
(Vensim, PySD, Insight
Maker, …)

UMEP
+ andere Umwelt-
modelle

SUMO
+ andere Ver
modelle

gtfs2emis
+ andere
Emissionsmodelle

CFD Modelle
für Strömungsc
Simulationen (L

30.11.23

CUT

Public Sector

OpenFOAM®

palm

UMEP

Open Forecast

Academia

HEROS

INSIGHT MAKER

Private Sector

SUMO

Civil Society

gtfs2emis

30.11.23

Bildnachweise [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

HEROS

INSIGHT MAKER

OpenFOAM®

palm

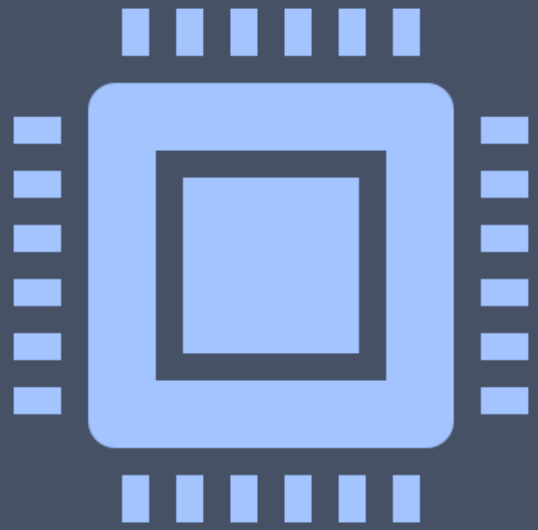
UMEP

SUMO

Open Forecast

MATSim
Multi-Agent Transport Simulation

gtfs2emis



Hohe Rechenleistung erforderlich

Bildnachweise [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

Zielsetzung

Entwicklung eines Prototypen, der…

1. Mehrere Modelle und Algorithmen über offene Schnittstellen auf verschiedenen Servern bereitstellt
2. Diese Modelle und Algorithmen auf einer offenen Plattform integriert
3. Die Modelle über das Masterportal abruft, ausführt und die Ergebnisse anzeigt

Prototyp – Grundlagen

Konzeptionelle Zusammenarbeit:

Geoinformation Vermessung

x

City Science Lab
A Cooperation with the MIT Media Lab

DIN SPEC 91357

[12]

TECHNISCHE REGEL [AKTUELL]

DIN SPEC 91357:2017-12

Referenzarchitekturmodell Offene Urbane Plattform (OUP); Text Englisch

Englischer Titel:
Reference Architecture Model Open Urban Platform (OUP); Text in English

Ausgabedatum:
2017-12

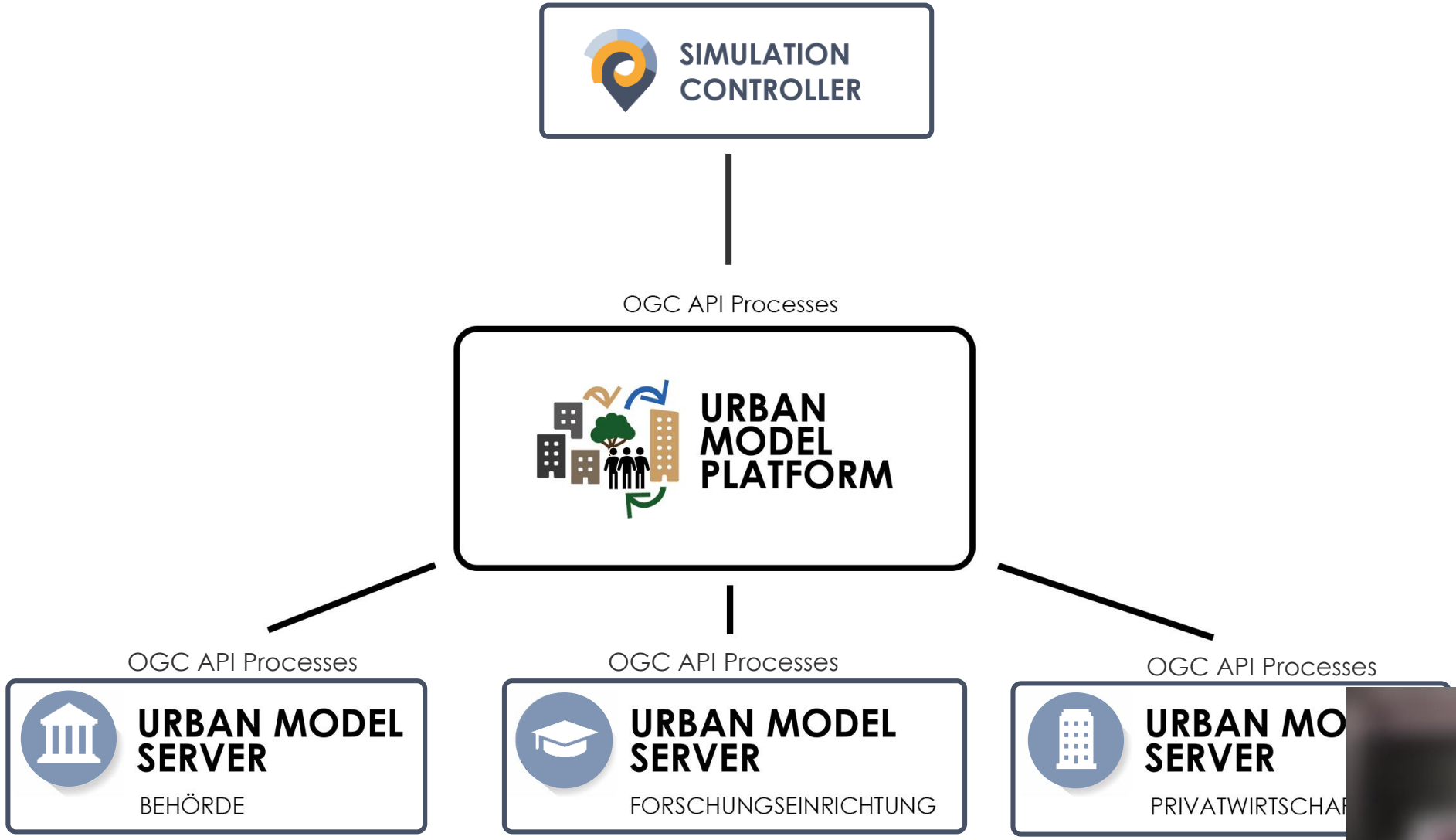
Originalsprachen:
Englisch

Seiten:
56

Verfahren:
PAS

OGC API Processes

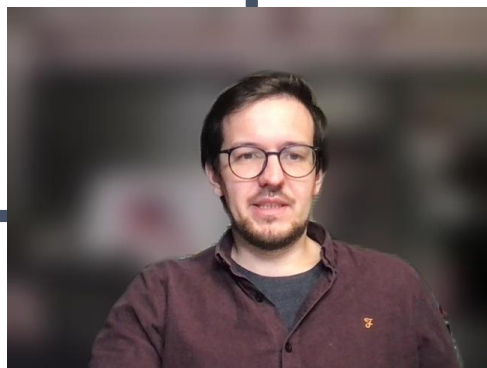
Prototyp – Architektur



Prototyp – Urban Model Platform

URBAN MODEL PLATFORM

- Bereitstellung beliebig vieler Modelle über die Verknüpfung zu Modellserver per OGC API Processes Standard
- Berechnet selbst nichts: Leitet die Inputdaten an die Modellserver weiter und speichert die Ergebnisse
- Abrufen und Filtern der Ergebnisse serverseitig möglich



Prototyp – Urban Model Server

URBAN MODEL SERVER

- Implementierung der OGC API Standards (insb. API Processes)
- Basis: Open-Source Software pygeoapi
- Erweiterung, um beliebig viele Modelle in beliebig vielen Programmiersprachen auf einem Server ausführen zu können

Academia

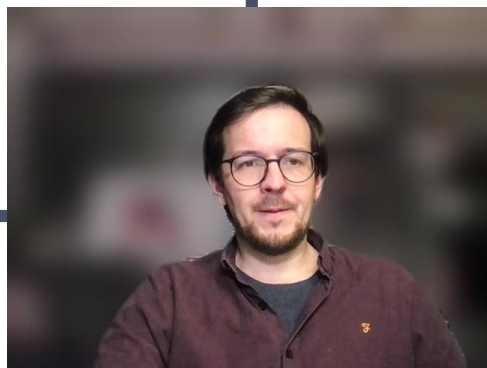
Civil Society

Private Se

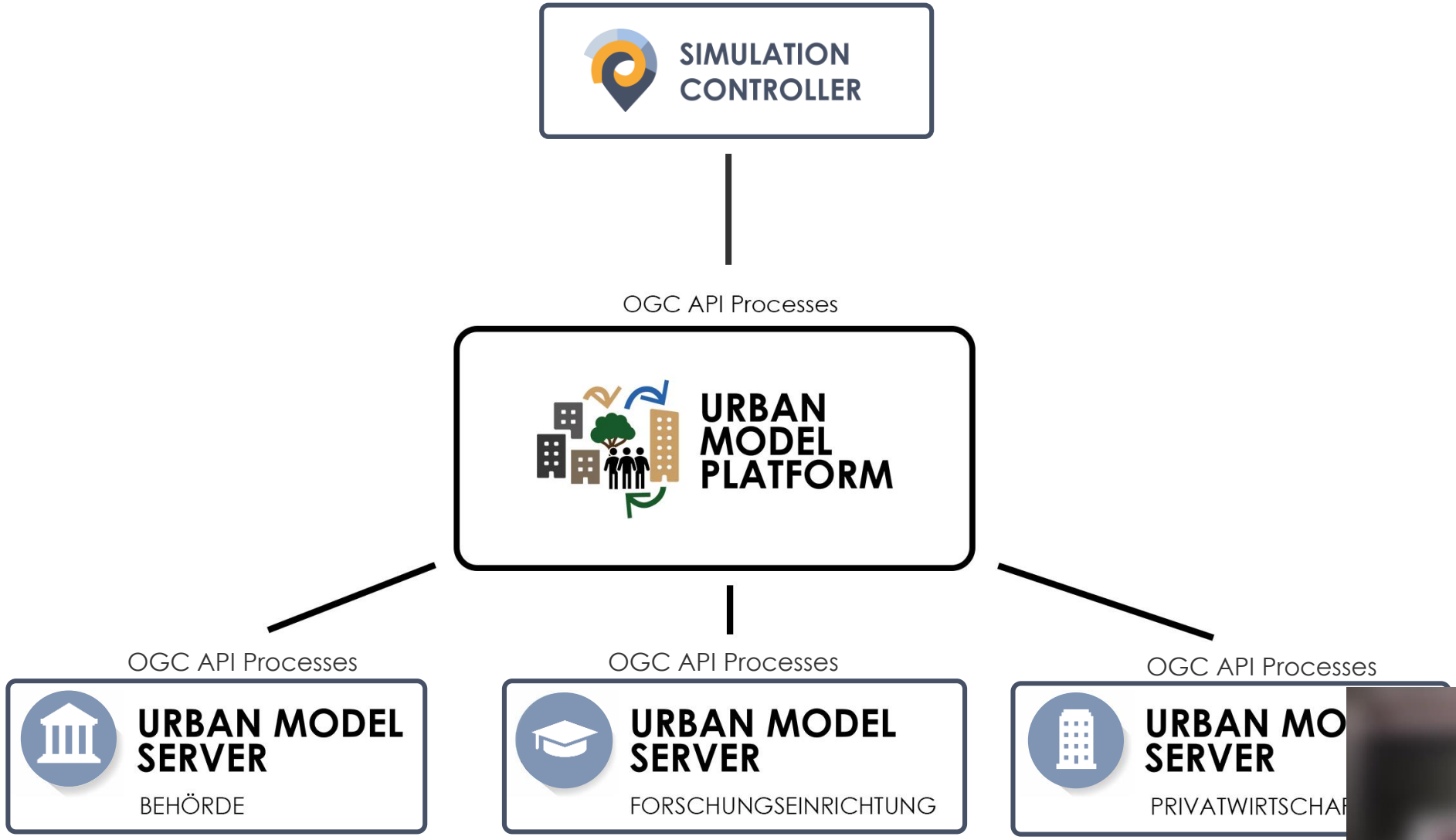
Prototyp – Simulation Controller

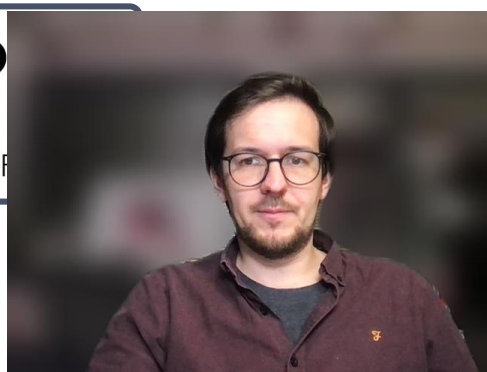
SIMULATION CONTROLLER

- Add-On zum Masterportal
- Konfiguration von „Simulationslayern“ im Masterportal
- Dynamisches Abrufen der Modellinputs
- Ausführen der Modelle und Ansicht der Jobs
- Dynamisches Filtern der Ergebnisse



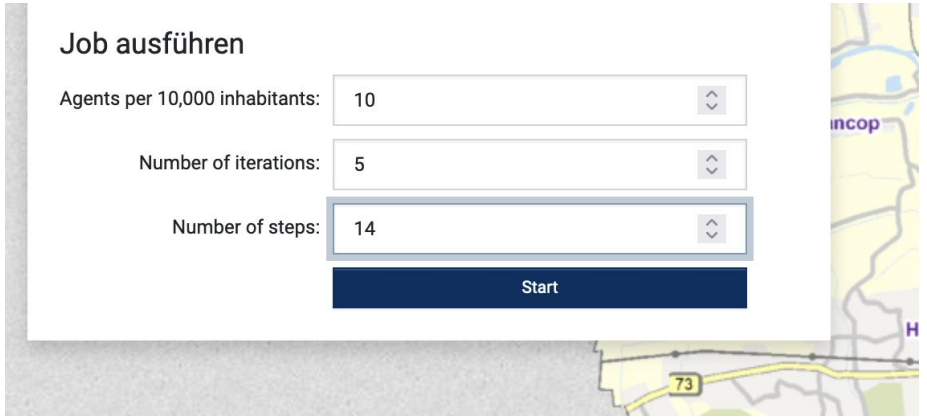
Prototyp – Architektur





Prototyp – Architektur

Nutzer:in



SIMULATION CONTROLLER

OGC API Processes

URBAN MODEL PLATFORM

OGC API Processes

URBAN MODEL SERVER
BEHÖRDE

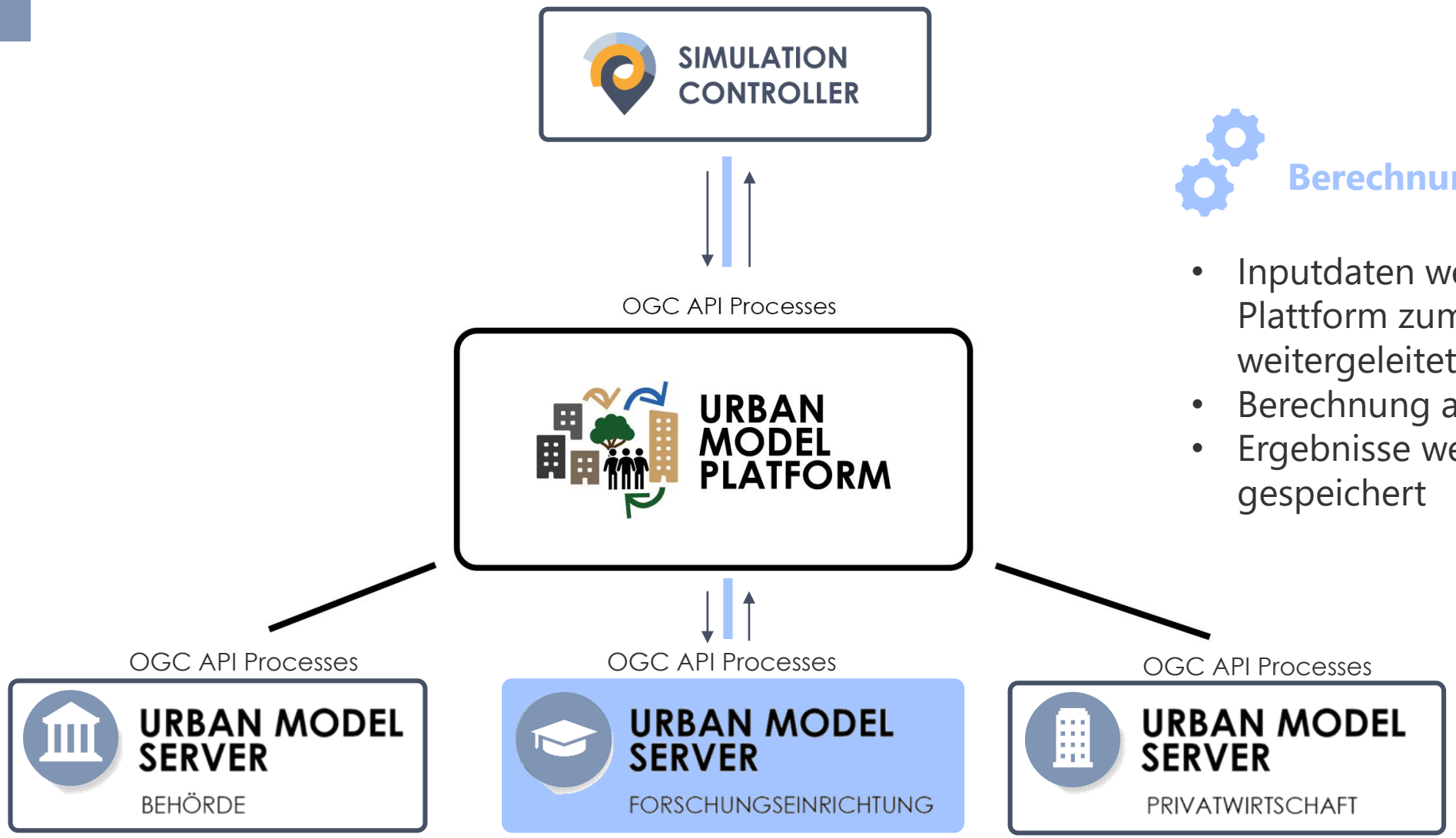
OGC API Processes

URBAN MODEL SERVER
FORSCHUNGSEINRICHTUNG

OGC API Processes

URBAN MODEL SERVER
PRIVATWIRTSCHAFT

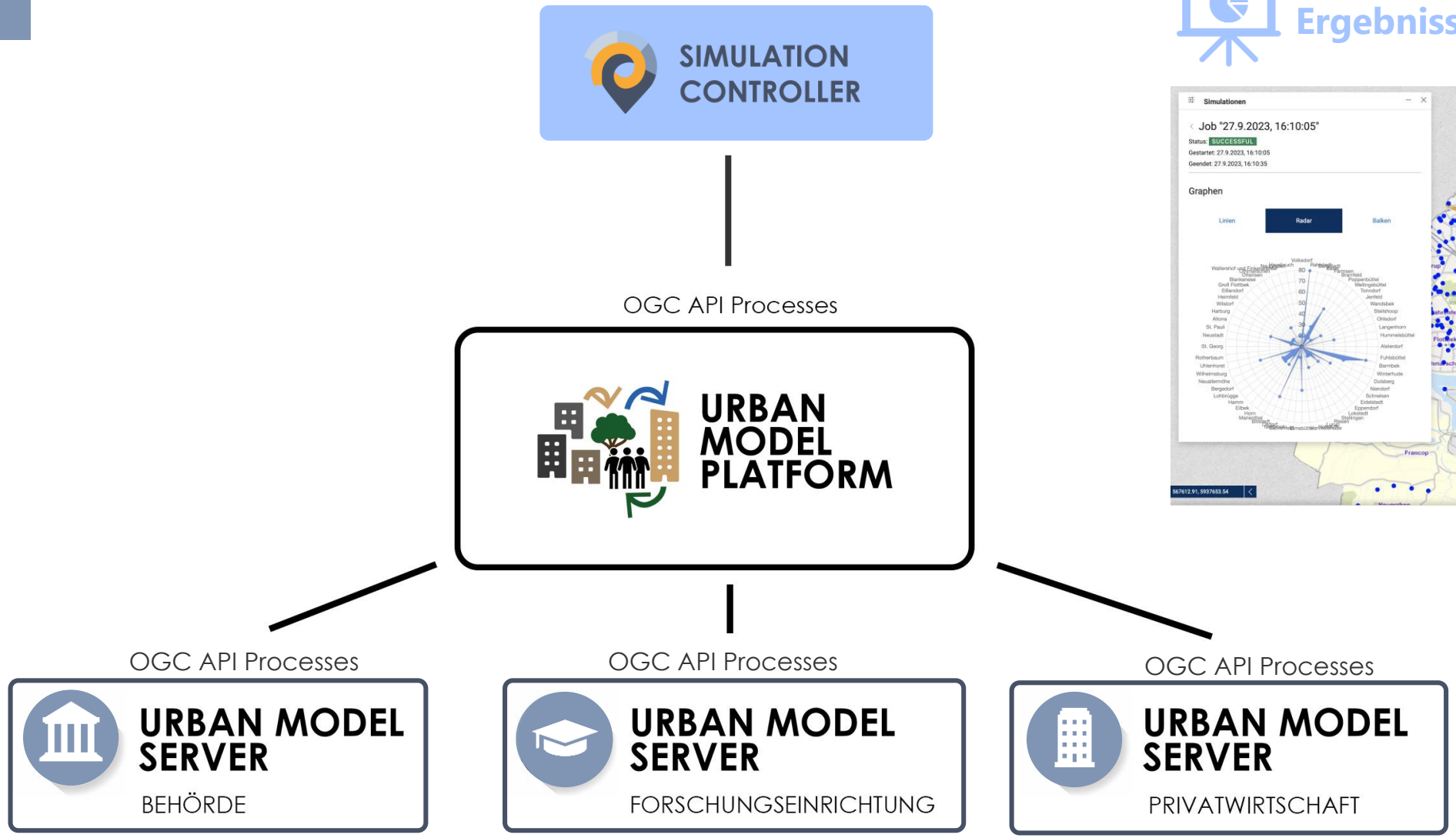
Prototyp – Architektur



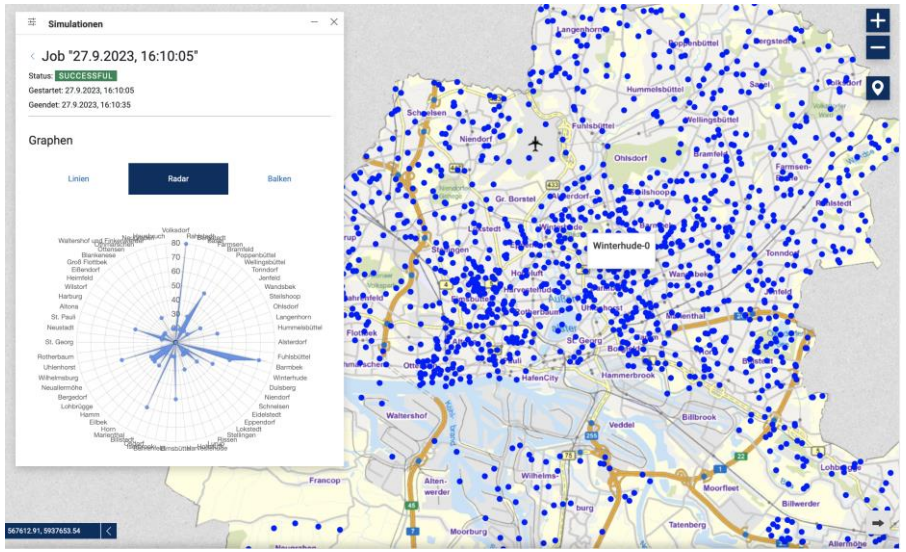
Berechnung

- Inputdaten werden über die Plattform zum richtigen Server weitergeleitet
- Berechnung auf dem Server
- Ergebnisse werden auf der Plattform gespeichert

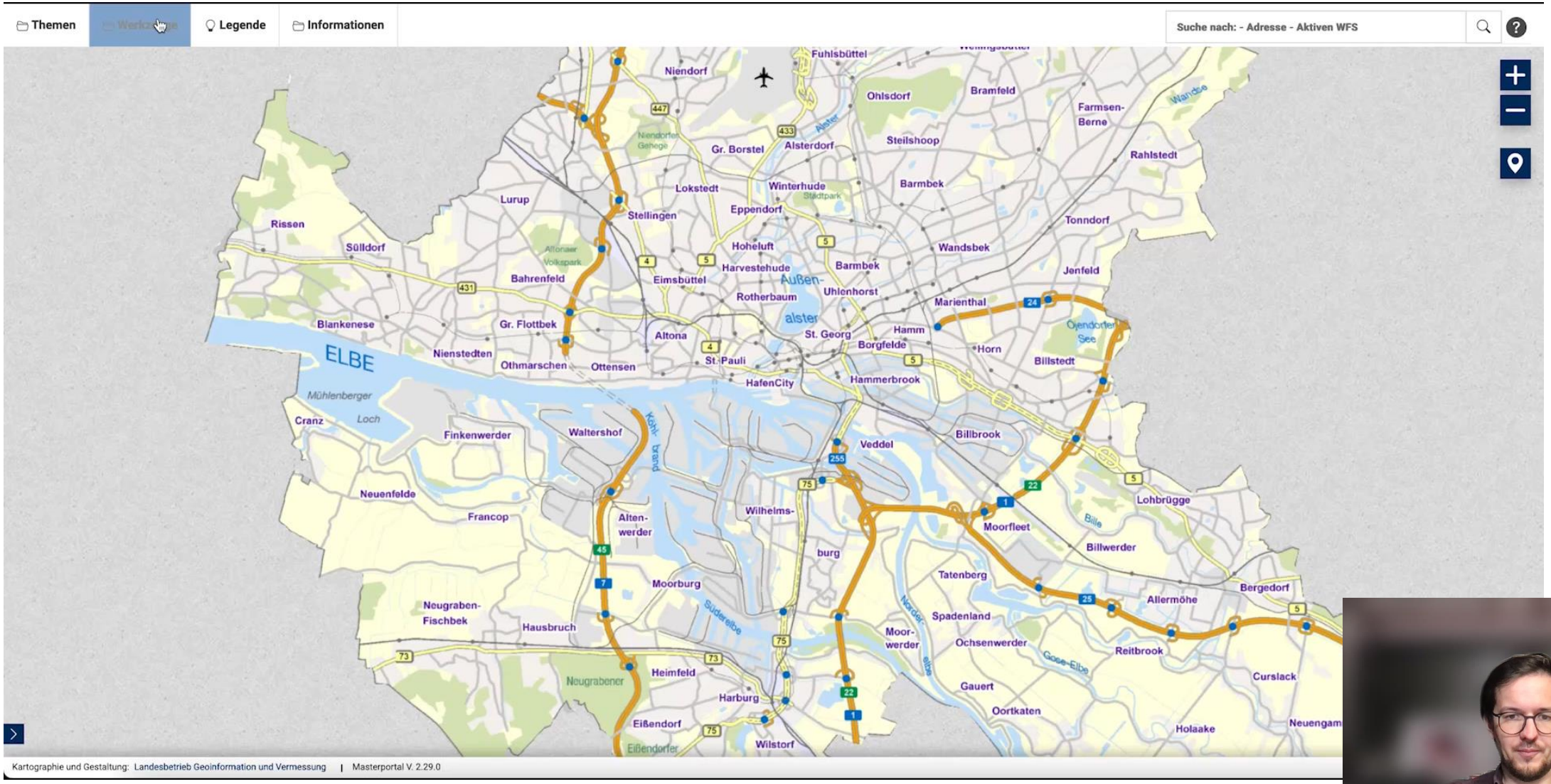
Prototyp – Architektur



Ergebnisse



Prototyp – Demo



Prototyp - Architektur

Vorteile

Robustheit und Erweiterbarkeit
durch Dezentralität und offene Standards

Multi-dimensionale Szenarien
durch Kommunikation der Modelle untereinander

Wiederverwendbarkeit
durch Entkopplung Modelle & Anwendungen

Offene Fragen & nächste Schritte

- **Governance**: Wie kann in einem dezentralen System aus Modellservern transparent mit den Rechenkosten und Zugriffen umgegangen werden?
- **Modellbereitstellung und Versionierung**: Wie können standardisiert & versioniert Algorithmen auf Modellservern und der UMP bereitgestellt werden?
- **Weitere Input- und Outputformate**
- **Personalisierung:** Zugriff auf Modellergebnisse, Teilen etc. → Backend benötigt

Zusammenfassung

- Es gibt bereits eine **Vielzahl an Modellen und Algorithmen**, die mehr oder weniger unzugänglich in **Silos** liegen
- Bislang **fehlt eine offene urbane Plattform** zum Bereitstellen und Verknüpfen der Modelle
- Der Prototyp basierend auf der OGC API Processes zeigt, dass eine Urban Model Platform **technisch möglich** und **vielversprechend** für multidimensionale „Was wäre wenn?“ Szenarien ist

Übersicht

II. Technische Umsetzung

1) OGC API Processes
2) Übersicht Backend-Architektur
3) Urban Model Server
    i. PyGeoAPI
    ii. Processing Microservices
4) Urban Model Platform
5) Simulation Controlle

OGC API Processes

Überblick

- Standard für die Beschreibung und Bereitstellung verschiedener Prozesse in einem Webserver
- Prozesse können synchron (die Ergebnisse werden direkt an einen Client zurückgeschickt) oder asynchron ausgeführt werden
- Im letzteren Fall gibt es ein Job-Management
  - Client erhält eine Job-ID -> kann jederzeit Status-Updates zu diesem Job erhalten
- Maschinenlesbare Liste von Prozessen, erwarteten Eingabe- und Ausgabeparametern, etc.

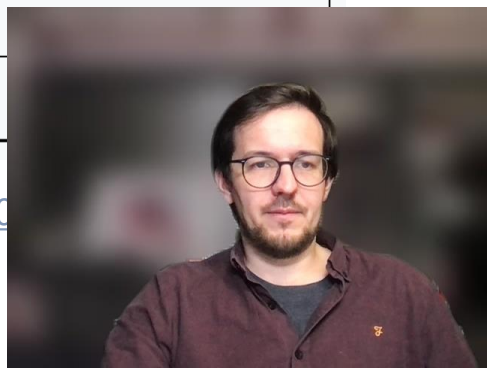
→ Grundsätzlich können alle Berechnungen, die zu rechenintensiv für Clients sind, von Servern mit der API Processes übernommen werden

OGC API Processes

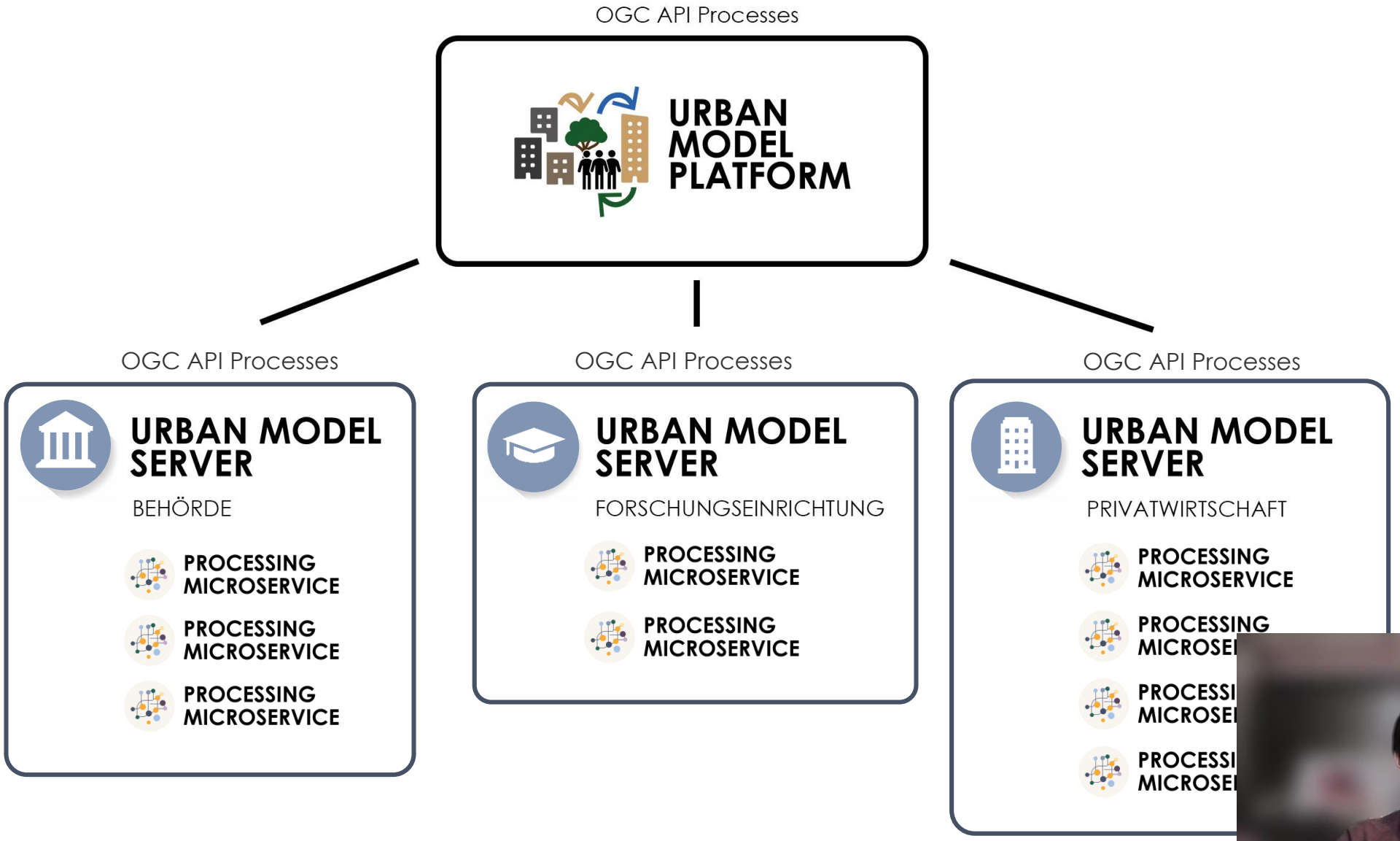
Core

Resource	Path	HTTP method	Parameter	Document reference
Landing page	/	GET	N/A	Clause 7.2
Conformance classes	/conformance	GET	N/A	Clause 7.4
Process list	/processes	GET	N/A	Clause 7.9
Process description	/processes/{processID}	GET	processID (in path)	Clause 7.10
Process execution	/processes/{processID}/execution	POST	processID (in path), Execute request (contained in body)	Clause 7.11
Job status info	/jobs/{jobID}	GET	jobID (in path)	Clause 7.12
Job results	/jobs/{jobID}/results	GET	jobID (in path)	Clause 7.13

Abgerufen von: https://docs.ogc.org/is/18-0



Übersicht Backend-Architektur



Urban Model Server

Pygeoapi

pygeoapi

URBAN MODEL SERVER

GitHub Repository

- Implementierung der OGC API Standards in einem Python-basierten Server
- Open-Source (MIT Lizenz)
- Job-Management mit integrierter TinyDB
- Erweiterung im Rahmen CUT:
  - Websocket-Endpunkt
  - Registrieren & Ausführen von Processing Microservices per WebSockets

Urban Model Server

Processing Microservices

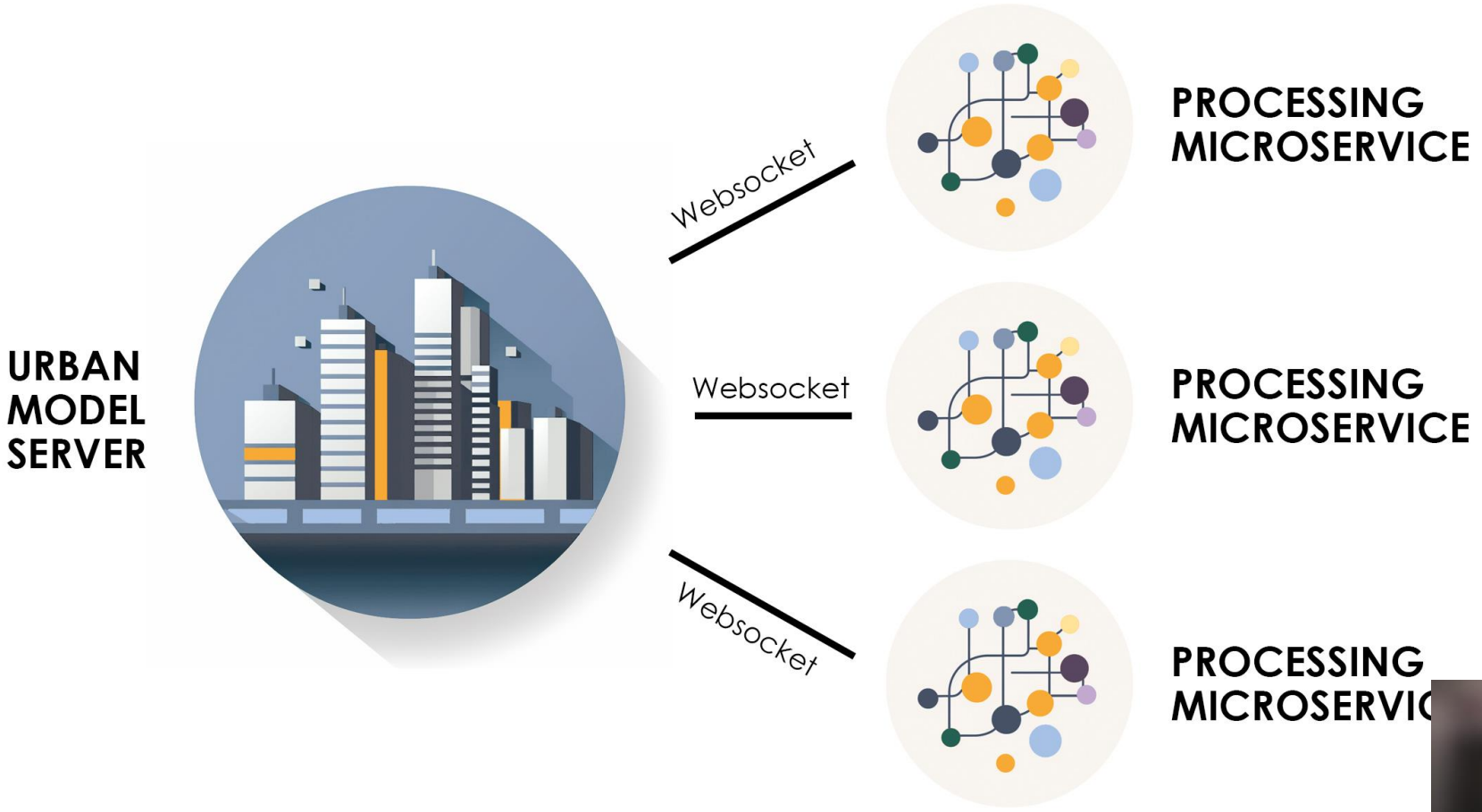
PROCESSING MICROSERVICE
node.js

PROCESSING MICROSERVICE
Python

- Microservices, die in einer isolierten, dockerisierten Umgebung ausgeführt werden
- Aktuell
  - Node.js Microservice
  - Python Microservice
- In Planung
  - R Microservice
  - Java Microservice

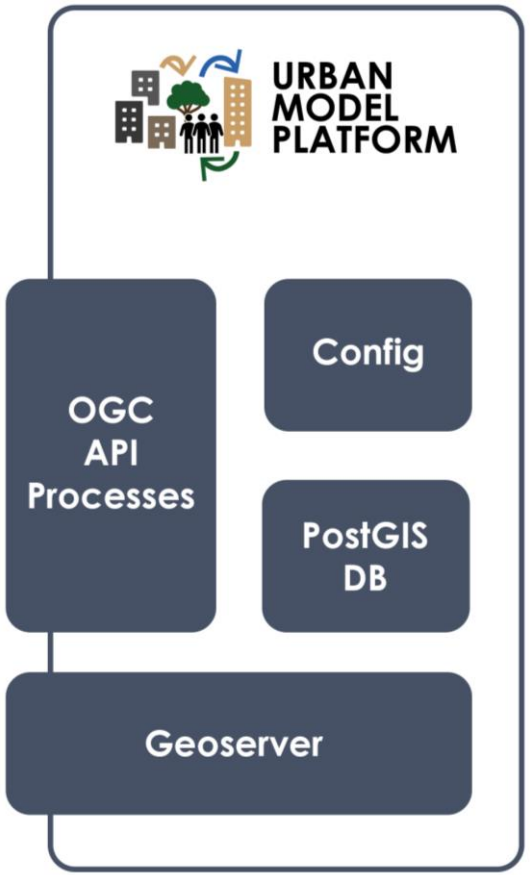
GitHub Repositories

Urban Model Server



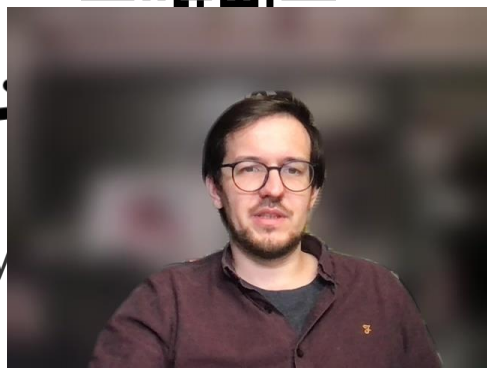
Urban Model Platform

Aufbau



- OGC API Processes
  - RESTful API nach OGC Standard
  - Modellserver werden in der config deklariert
- PostGIS DB
  - Speichern von Input-Parametern der einzelnen Jobs
  - Speichern von Ergebnisse und Metadaten
- Geoserver
  - Abfrage der Ergebnisse
  - Filtern

GitHub Repository



Urban Model Platform

Config

```yaml
# This is the configuration file for setting up simulation servers.
# The servers should provide an OGC processes api which will be retrieved
# on the fly to provide all existing processes via this api. Please provide the base url
# of the api, e.g. if there is an endpoint https://example.org/rest/api/processes
# then provide https://example.org/rest/api as url.
# This file should be considered as a secret.

modelserver-1:
  url: "https://modelserver1.cut.hcu-hamburg.de"
  user: "cut"
  password: "password"
  timeout:  1800
  exclude:
    - "abm-test-model"

modelserver-2:
  url: "https://modelserver2.cut.hcu-hamburg.de"
  user: "cut"
  password: "password"
  timeout:  1800
```

Simulation Controller

Konfiguration von Simulationslayern in Portalinstanzen

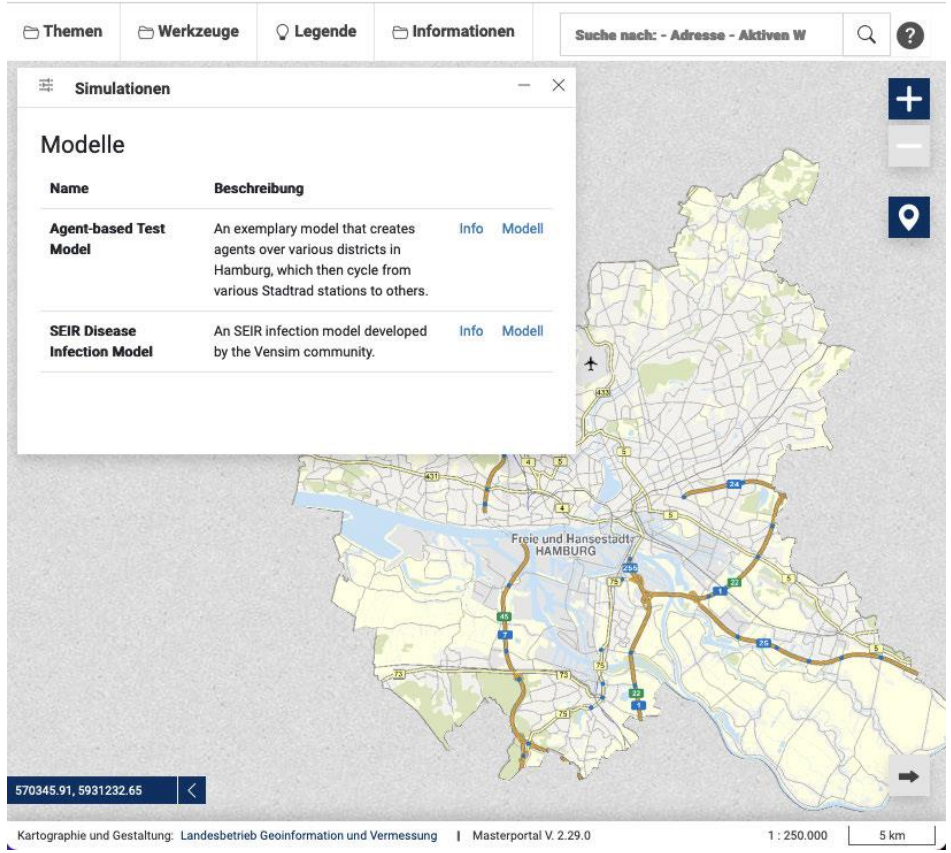
SIMULATION CONTROLLER

MASTER PORTAL

```
[
  {
    "id": "123456",
    "isSimulationLayer": true,
    "filterOnClient": false,
    "simModelId": "cut:abm-test-model",
    "styleId": "654321",
    "name": "Simulation Layer",
    "url": "http://localhost:3000/geoserver/CUT/ows",
    "featureType": "",
    "version": "1.0.0",
    "typ": "WebGL",
    "isVisibleInMap": true,
    "legend": false,
    "style": {
      "symbol": {
        "symbolType": "circle",
        "size": 10,
        "color": [0, 0, 255],
        "opacity": 1,
        "rotateWithView": true
      }
    }
  }
]
```

Simulation Controller

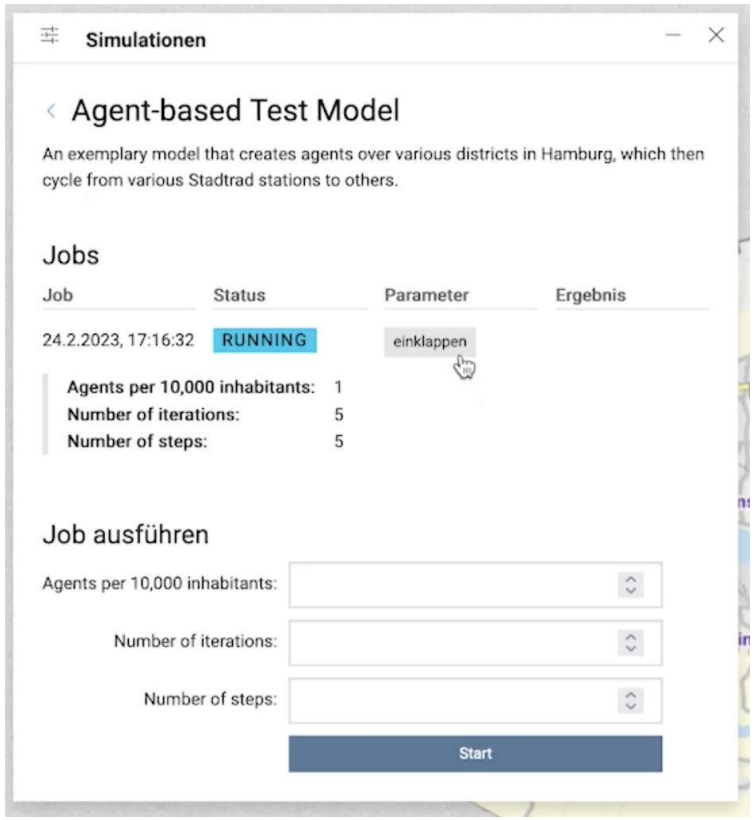
Übersichtsseite



- Anzeige aller Modelle/Algorithmen, die in die Portalinstanz konfiguriert wurden
- Beschreibung
- Links zu weiterführenden Informationen

Simulation Controller

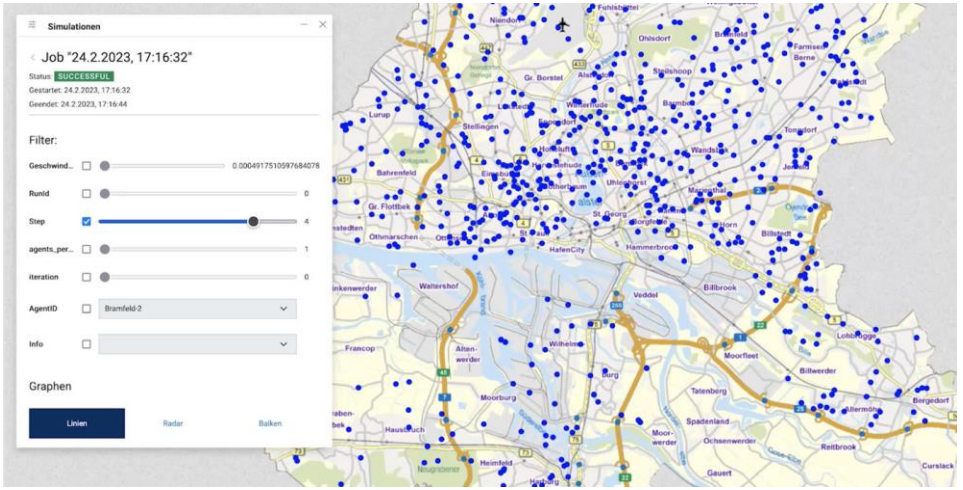
Detailseite Modell



- **Jobliste**: Anzeige aller Jobs, die bereits ausgeführt wurden und werden (inkl. Parametern)
- **Job ausführen**:
  - Parameter werden dynamisch von der Urban Model Platform abgerufen und im Masterportal gerendert
  - Automatische Prüfung verschiedener Inputs (String, Number, ...)
  - Starten des Jobs

Simulation Controller

Detailseite Ergebnis



- Anzeige der georeferenzierten Daten auf der Karte
- Dynamisches Filtern der Attribute durch den User
- Prototypisch: Anzeige von Graphen und Diagrammen

CONNECTED URBAN TWINS

Partnerstädte:

Hamburg

Stadt Leipzig

Landeshauptstadt München

Gefördert durch:

Bundesministerium für Wohnen, Stadtentwicklung und Bauwesen

KFW

CUT

Vielen Dank für Ihre Aufmerksamkeit!

Rico Herzog
M.Sc. Engineering and Policy Analysis

rico.herzog@hcu-hamburg.de

www.cityscie

www.connect

Referenzen

[1] Open Forecast. https://open-forecast.eu/ (Letzter Abruf: 24.11.23)

[2] PALM. https://palm.muk.uni-hannover.de/trac (Letzter Abruf: 24.11.23)

[3] MATSim. https://www.matsim.org/ (Letzter Abruf: 24.11.23)

[4] GAMA Platform. https://gama-platform.org/ (Letzter Abruf: 24.11.23)

[5] HERoS. https://www.heros-project.eu (Letzter Abruf: 24.11.23)

[6] ChatGPT. https://chat.openai.com/ (Letzter Abruf: 24.11.23)

[7] Insight Maker. https://insightmaker.com/ (Letzter Abruf: 24.11.23)

[8] UMEP. https://umep-docs.readthedocs.io/en/latest/Introduction.html (Letzter Abruf: 24.11.23)

[9] SUMO. https://sumo.dlr.de (Letzter Abruf: 24.11.23)

[10] gtfs2emis. https://ipeagit.github.io/gtfs2emis/ (Letzter Abruf: 24.11.23)

[11] Open Foam. https://www.openfoam.com/ (Letzter Abruf: 24.11.23)

[12] DIN SPEC 91357. https://www.beuth.de/de/technische-regel/din-spec-91357/281077528 (Letzter Abruf: 24.11.23)

Logos der Urban Model Platform, des Urban Model Servers und der Processing Microservices