Data-driven and Dynamic Space and Assets for Physical Internet-led Urban Logistics and Planning

D2.4 - DISCO-X Enabling Tools

CERTH

30/10/2025

Vasileios Giannoudis, Dimos Touloumidis, Dr Georgia Ayfantopoulou

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101103954. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

List of Figures

Executive Summary

This deliverable presents an analysis of the DISCO-X enabling tools, which function as the technical basis for the five innovations of the DISCO project. The purpose of this document is to analyze the suite of software, platforms and analytical models used in the project, moving beyond description to identify specific components that make them suitable for realizing the Physical Internet (PI) concept. Essentially, it provides a blueprint for understanding how the strategic PI vision was translated into operational reality through the adaptation and validation of these tools in the Living Labs of the project.

The analysis followed by the writers employs a five-step retrospective framework. The methodology observes the outcomes from the work of the Living Labs to identify the PI-enabling innovations that emerged. The process included a foundational analysis of PI requirements; the identification of baseline tools and their alignment with the ALICE PI Roadmap; a detailed analysis of enhancements made by the Living Labs; validation of these enhancements using empirical pilot results; and the synthesis and identification of specific PI-enabling components.

The core of the deliverable (Chapters 3-7) applies this framework to each of the five DISCO-X innovations, detailing validated enhancements to key tools. DISCOCURB tools were enhanced to transform the curb into a digitally managed, shared asset. DISCOPROXI tools were enhanced to enable shared last-mile networks, including planning models for shared lockers and operational tools for route orchestration. The WareM&O platform for DISCOESTATE was enhanced with blockchain e-signatures and Building Management System integration, transforming it into a trusted, operational marketplace. DISCOBAY tools were enhanced to support multimodal strategies through simulation and operational suggestions. DISCOLLECTION tools were enhanced to create the foundational "data backbone" for the PI, including the Smart Data Platform (SDP) for data standardization (DATEX II, APDS) and the proof-of-concept urban logistics dataspace for data aggregation.

Furthermore, the deliverable provides a proof-of-concept use case (Chapter 8) that demonstrates the end-to-end integration of the project's core digital infrastructures. This use case illustrates how a user, operating within the Meta-Model Suite, can discover and book shared logistics space from the WareM&O platform (DISCOESTATE), with the data query and transaction being seamlessly federated and harmonised by the DISCO Urban Freight Dataspace.

Finally, the concluding analysis (Chapter 9) maps the validated toolkit against the five building blocks of the ALICE Physical Internet Roadmap. This mapping demonstrates that the DISCO-X tools provide tangible contributions across the PI landscape (Logistics Nodes, Logistics Networks, System of Logistics Networks, Access & Adoption and Governance), offering a practical resource for other cities to advance their own PI transition.

Summary sheet

Deliverable No.	D2.4
Project Acronym	DISCO
Full Title	DATA-DRIVEN, INTEGRATED, SYNCHROMODAL, COLLABORATIVE AND OPTIMISED URBAN FREIGHT META MODEL FOR A NEW GENERATION OF URBAN LOGISTICS AND PLANNING WITH DATA SHARING AT EUROPEAN LIVING LABS
Grant Agreement No.	101103954
Responsible Author(s)	Vasileios Giannoudis, Dimos Touloumidis, Dr Georgia Ayfantopoulou (CERTH)
Peer Review	Paola Astegiano (FIT), Laurence Bannerman (EPA)
Quality Assurance Committee Review	FIT
Date	30.10.2025
Status	Final
Dissemination level	Public
Version	0.7
Work Package No.	2
Work Package Title	The DISCO urban logistics and planning PI-led Digital Transition
Programme	Horizon Europe Innovation Actions
Coordinator	FIT CONSULTING SRL
Website	https://discoprojecteu.com/
Starting date	01.05.2023
Number of months	42
Deliverable No.	D2.4

Project partners

Organisation	Country	Abbreviation
FIT CONSULTING SRL IT Coordinator	IT	FIT
RUPPRECHT CONSULT-FORSCHUNG & BERATUNG GMBH	DE	RC
INLECOM INNOVATION ASTIKI MI KERDOSKOPIKI ETAIREIA	EL	INLE
PNO INNOVATION SL	ES	PNO
INTERNATIONAL DATA SPACES EV	DE	IDSA
FM LOGISTIC IBERICA SL	ES	FM
AKKA INDUSTRY CONSULTING GMBH	DE	AKKA
FONDAZIONE ISTITUTO SUI TRASPORTI E LA LOGISTICA	IT	ITL
JLL	UK	JLL
ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTY XIS	EL	CERTH
LINDHOLMEN SCIENCE PARK AB	SE	LSP
KUHNE LOGISTICS UNIVERSITY GGMBH	DE	KLU
INSTITUT DE RECHERCHE TECHNOLOGIQUE SYSTEM X	FR	IRTX
STICHTING BREDA UNIVERSITY OF APPLIED SCIENCES	NL	BUAS
POLIS - PROMOTION OF OPERATIONAL LINKS WITH INTEGRATED SERVICES, ASSOCIATION INTERNATIONALE	BE	POLIS
EUROPEAN PARKING ASSOCIATION EPA AISBL	DE	EPA
ALLIANCE FOR LOGISTICS INNOVATION THROUGH COLLABORATION IN EUROPE	BE	ALICE
ERASMUS CENTRE FOR URBAN,PORT AND TRANSPORT ECONOMICS BV	NL	ERASMUS
INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM	BE	IMEC

VLAAMS INSTITUUT VOOR DE LOGISTIEK VZW	BE	VIL
FUNDACION ZARAGOZA LOGISTICS CENTER	ES	ZLC
BE-MOBILE	BE	BE-MOBILE
STAD GENT	BE	GENT
OPLEIDINGSCENTRUM VOOR HOUT EN BOUW VZW	BE	OHB
CITYLOGIN IBERICA SL	ES	CITYLOGIN
UNIVERSITAT POLITECNICA DE CATALUNYA	ES	UPC
AJUNTAMENT DE BARCELONA	ES	BCN
VENICE INTERNATIONAL UNIVERSITY	IT	VIU
FUNDACION DE LA COMUNIDAD VALENCIANA PARA LA INVESTIGACION, PROMOCION Y ESTUDIOS COMERCIALES DE VALENCIAPORT	ES	VPF
FUNDACION DE LA COMUNITAT VALENCIANA PARA LA PROMOCION ESTRATEGICA EL DESARROLLO Y LA INNOVACION URBANA	ES	LAS NAVES
T-BOX DELIVERY & SOLUTIONS SL	ES	T-BOX
AYUNTAMIENTO DE ZARAGOZA	ES	ZARAGOZA
FUNDACION ZARAGOZA CIUDAD DE CONOCIMIENTO	ES	FZCC
FORUM VIRIUM HELSINKI OY	FI	FVH
KOBENHAVNS KOMMUNE	DK	COPENHAGEN
REGION HOVEDSTADEN DK Partner	DK	REGIONH
COMUNE DI PIACENZA	IT	PIACENZA
MESTSKA CAST PRAHA 6 / District Prague	CZ	PRAHA
REGIONAL MANAGEMENT NORDHESSEN GMBH	DE	RMNH
AARHUS KOMMUNE	DK	AAKS

DIMOS THESSALONIKIS	EL	THESSALONIKI
DIETHNIS EKTHESI THESSALONIKI AE	EL	TIF HELEXPO
ACS TACHIDROMIKES IPIRESIES MONOPROSOPI ANONYM	EL	ACS
ROLAN OY	FI	ROLAN
ASOCIACIÓN LOGÍSTICA INNOVADORA DE ARAGÓN	ES	ALIA
A to B Finland Oy	FI	A2B
GETPLUS srl IT Partner	IT	NEXT
COMUNE DI PADOVA IT	IT	ComPADUA

Document history

Version	Date	Organisation	Main area of changes	Comments
V0.1	01/08/2025	CERTH	Initial version of the D2.4 Table of Contents	-
V0.2	20/09/2025	CERTH	First draft	-
V0.3	30/09/2025	CERTH	Second draft	-
V0.4	08/10/2025	CERTH	Third and final draft, sent to reviewers	Peer-reviewing partners: 1st: FIT, 2nd: EPA
V0.5	22/10/2025	FIT, EPA	Reviewers feedback	Peer-reviewing partners: 1st: FIT, 2nd: EPA
V0.6	22/10/2025	CERTH	Inclusion of comments received on XXXX	The final version is ready to be submitted
V0.7	30/10/2025	FIT	Deliverable 2.4 submission to the portal	

List of acronyms

ALICE	Alliance for Logistics Innovation through Collaboration in Europe
APDS	Alliance for Parking Data Standards
API	Application Programming Interface
BMS	Building Management System
CMM	Capability Maturity Model
EIF	European Interoperability Framework
ETL	Extract, Transform, Load
FCD	Floating Car Data
GIS	Geographic Information System
GLM	Green Last Mile
KER	Key Exploitable Result
KPI	Key Performance Indicator
LCV	Light Commercial Vehicles
LEZ	Low-Emission Zone
LSP	Logistics Service Provider
MMS	Meta Model Suite
PI	Physical Internet
SDP	Smart Data Platform
SLL	Starring Living Lab
SULP	Sustainable Urban Logistics Plan

SUMP	Sustainable Urban Mobility Plan
UAC	Urban Access Control
UCC	Urban Consolidation Center
UVAR	Urban Vehicle Access Regulation
WaaS	Warehouse-as-a-Service
WMS	Warehouse Management System
WP	Work Package
ZEZ	Zero-Emission Zone

1. Introduction

1.1. Context and Purpose

European cities face escalating challenges in urban logistics, driven by rapid urbanisation, the exponential growth of e-commerce and the urgent need to meet climate neutrality goals. The Physical Internet (PI) paradigm offers a transformative vision to address these issues: **a hyperconnected, shared and interoperable logistics network operating with the efficiency of the digital internet.** However, this vision raises a fundamental question regarding the practical role of digital tools within the Physical Internet concept. They are not merely accessories; they represent the fundamental **operating system required to realize the PI.**

This document aims to deconstruct and analyze the DISCO-X enabling tools to demonstrate how they fulfill the core functions required by any PI ecosystem. Specifically, this deliverable will show:

- *How tools create visibility and transparency across the logistics network.*
- *How they enable the digital management of shared physical assets.*
- *How they are used to optimise and orchestrate the network for system-wide efficiency.*
- *How they facilitate access and drive adoption among a diverse set of urban stakeholders.*

1.2. How this deliverable relates to tasks

As a key output of Work Package 2, this deliverable integrates inputs from previous tasks and establishes a basis for subsequent project phases. Its position within the Agile methodology of the project ensures that tool development aligns with real-world needs and pilot experiences.

Inputs to this deliverable include:

D2.1 - Urban Logistics Transition Requirements: This document established the initial requirements ("demand") by mapping the specific needs, challenges and implementation requirements of the Living Labs. The tools detailed herein represent the corresponding capabilities ("supply") designed to meet those specifications.

D2.3 - DISCO Meta Model Suite architecture: The MMS architecture supplied the conceptual framework for a "DISCO-X Tools Library". This document furnishes that library with specific, tested tools.

D4.2 - Starring LL measures implementation: This constitutes the most significant input, offering the real-world results, lessons learned and validated performance of the tools following a 24-month implementation period in the four Starring Living Labs. The tools described herein are not theoretical; they have undergone refinement through practical application.

Outputs from this deliverable include:

This deliverable yields two primary outputs. First, it generates a confirmed and validated toolkit of PI-enabling solutions, serving as an important component of the Meta Model Suite and offering a practical, evidence-based resource for future implementations. Second, the specific analysis of the components and validated enhancements of each tool establishes the necessary technical basis for the interoperability assessment in **D2.5, "Measures and Meta Model Suite interoperability**".

1.3. Structure of the Document

The structure guides the reader through a systematic, evidence-based analysis, progressing from a base methodology to a strategic synthesis. Following this introduction, Chapter 2 details the analytical framework used throughout the deliverable (a five-step retrospective analysis designed to identify the Physical Internet-enabling components of each tool). Chapters 3 through 7 constitute the primary analytical section of the report. Each chapter focuses on one of the five DISCO-X innovations, applying the framework from Chapter 2 to the relevant enabling tools. This arrangement permits a thorough examination of the architecture, functionality and practical application for the tools associated with DISCOCURB, DISCOPROXI, DISCOESTATE, DISCOBAY and DISCOLLECTION, respectively, utilizing specific examples from the implementations within the Starring Living Lab. Chapter 8 then provides a practical, proof-of-concept use case demonstrating the end-to-end integration of the Meta-Model Suite, the Urban Freight Dataspace and the operational WareM&O enabling tool. Finally, Chapter 9 offers a high-level synthesis of the preceding analysis; it maps the validated DISCO-X toolkit against the strategic ALICE Physical Internet Roadmap (ALICE-ETP, 2020), to show the overall contribution of the project and presents the final conclusions and remarks for the deliverable.

2. Framework definition for Developing Enabling Tools

The primary purpose of this deliverable is to analyze the enabling tools used and enhanced within the DISCO project, identifying and documenting the specific components that make them suitable for realizing the Physical Internet concept. Tool creation followed a bottom-up approach instead of top-down development; each Living Lab independently adapted and improved existing, high-potential solutions to satisfy the distinct operational needs associated with its local measures.

Consequently, the framework detailed in this chapter employs retrospective analysis. This systematic process examines the outcomes stemming from the work of the Living Labs, identifying the principal PI-enabling innovations that arose. The five-step approach integrates continuous alignment with the ALICE PI Roadmap (ALICE-ETP, 2020) and is outlined below.

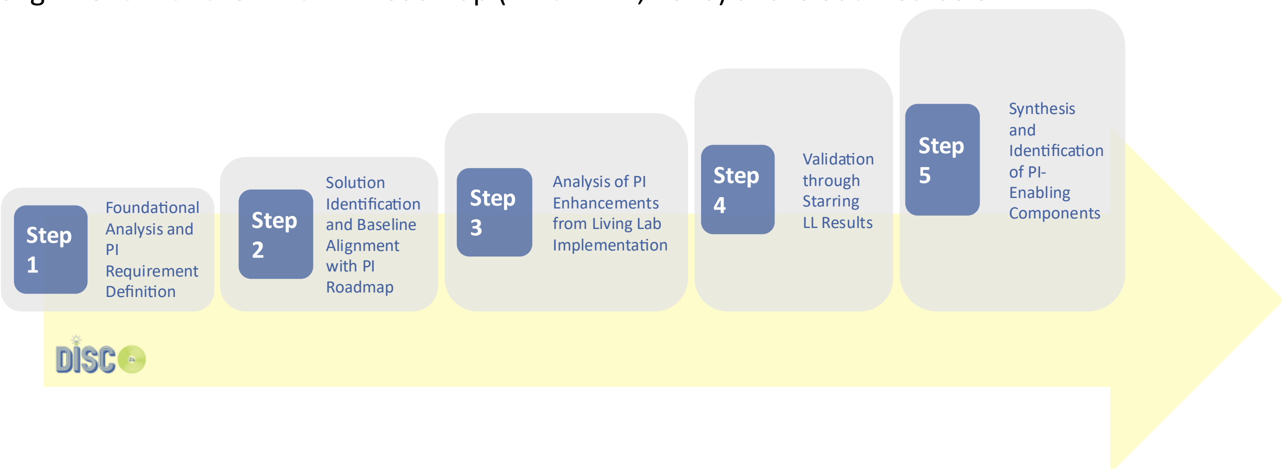


Figure 1: Framework for the Analysis and Identification of PI-led Enabling Tool Components

The initial phase set the analytical baseline, involving a review of the foundational documents of the project to formulate a distinct "demand profile", essentially a set of requirements defining a PI-led tool. This profile encompassed the specific challenges, goals and operational requirements identified by the Starring Living Labs (from Deliverable D2.1) and the architectural principles from the Meta Model Suite (from Deliverable D2.3) defining the goals for a PI-compliant tool architecture.

Step 2 centered on identifying appropriate existing tools within the portfolios of the partners, which acted as the foundation for each DISCO-X innovation (e.g., the WareM&O platform for DISCOESTATE, the UAC system for DISCOPROXI). An "as-is" analysis documented the baseline functionalities of these tools. This baseline underwent conceptual alignment with pertinent building blocks (the five development areas) of the ALICE PI Roadmap (Logistics Nodes, Logistics Networks, System of Logistics Networks, Access & Adoption and Governance) to pinpoint the initial "PI Gap";

this gap represents the conceptual and functional distance separating the existing tool from a fully PI-compliant solution.

Step 3 represents the central analytical phase. Rather than adhering to top-down directives, each Living Lab independently executed necessary enhancements and adaptations to its selected tools, enabling successful measure implementation and closure of the "PI Gap". This phase entails analyzing the work documented in Deliverable 4.2 to identify these bottom-up modifications. Attention centered on identifying adaptations specifically intended to improve alignment between the tool and the principles underpinning the PI Roadmap building blocks, including enabling shared capacity management (Logistics Nodes), integrating standardized data models for interoperability (System of Logistics Networks) and supporting novel, collaborative business models (Access and Adoption).

Empirical evidence gathered during the 24-month implementation period validated the success of the enhancements identified in Step 3. The detailed results and lessons learned documented in Deliverable 4.2 provided the basis for this validation. For instance, the successful brokering of the agreement between TIF-HELEXPO and ACS Courier validated enhancements to the e-signature and multi-tenant features within the WareM&O platform; this confirmed the effectiveness of the platform in enabling a shared PI Logistics Node within an actual operational environment.

This final step, forming the primary content for subsequent chapters, synthesizes findings from the preceding steps. Following analysis of the validated enhancements, this step formally identifies and documents the specific components, features and functionalities deemed ***“proper for realising the PI concept”***. Consequently, this deliverable examines each tool, highlighting these principal **PI-enabling components and explaining how they facilitate the core functions associated with the PI building blocks** (e.g., how a tool enables shared nodes, orchestrates networks, contributes to the system of networks or improves access and adoption).

3. Tools for Dynamic Space Management (DISCOCURB)

3.1. Introduction to the DISCOCURB tools

DISCOCURB refers to the overall approach to be implemented by the cities to address the challenges of curbside management (Neels, 2023). It represents a data-driven, technology-enabled and dynamic process that supports the objectives of enhancing land use allocation and executing dynamic curbside management utilizing advanced data analytics and digital tools. It is not just a single solution or technology but represents an integrated approach that combines various elements to optimize the use of curbside spaces effectively. DISCOCURB relies heavily on data analytics and real-time information to make informed decisions and drive actions.

This data-driven approach ensures that decisions are based on actual usage patterns and helps city planners understand the demand for curbside spaces at different times and locations. The city can then dynamically adjust curbside regulations, such as loading/unloading zones and parking restrictions, based on the current needs and traffic conditions. This dynamic process ensures that curbside spaces are utilized optimally throughout the day, reducing congestion and enhancing overall efficiency.

In its essence, the problem that DISCOCURB comes to solve for cities is the inefficiency and suboptimal use of curbside spaces in urban areas. The current curbside management practices often lack the use of advanced data analytics and digital tools, leading to challenges in optimizing land use allocation and managing loading/unloading areas in urban logistics operations. This inefficiency results in congestion, delays, and reduced accessibility for logistic operators, businesses, residents, and transportation services.

Within DISCO, the DISCOCURB approach was employed by Copenhagen and Helsinki (as Starring Living Labs) and will be utilized by Barcelona and Padua (as Twinning Living Labs). At the time of submitting this deliverable, the most detailed examples pertain to Copenhagen and Helsinki; however, a brief overview for Barcelona and Padua will also be provided.

3.1.1. The Digital Handshake: Dynamic Booking and Management Tools

Dynamic booking and management tools represent an important strategy for optimizing curbside space allocation within congested urban environments. These systems permit users (e.g., delivery operators) to reserve curb space beforehand or in real time, frequently via mobile applications or integrated fleet management platforms. By transitioning from static, first-come-first-served allocation towards a demand-responsive model, cities can decrease double-parking, idling and traffic circulation resulting from drivers seeking available space. Dynamic tools also enable municipalities to apply variable pricing and time restrictions contingent on demand patterns, vehicle type or specific policy objectives, such as prioritizing zero-emission vehicles or freight deliveries during off-peak hours. Integration into broader smart mobility ecosystems improves effectiveness, as real-time occupancy data and predictive analytics assist in adapting allocation rules dynamically. Nonetheless, broad adoption necessitates addressing interoperability, user compliance, coordination and integration with enforcement measures and equitable access, particularly for smaller operators lacking advanced digital capabilities.

Identified gaps in the literature encompass interoperability standards, equitable access for small operators and reliable methods for establishing dynamic prices that balance efficiency with social aims; several recent practice-focused guides and federal reports advocate for integrated pilots combining bookings, pricing and enforcement to assess these trade-offs (Burns et al., 2025; Diehl et al., 2021). Within DISCO, solution types introduced in this paragraph were tested by Copenhagen (as a Starring Living Lab) and will be tested by Padua and Barcelona (as Twinning Living Labs).

In Copenhagen, the solution provided by Coding the Curbs (https://www.codingthecurbs.com/) was implemented. It comprises a user-friendly booking platform, smart signage at the location, sensors and an integrated platform. This technology underwent testing at two delivery sites in Tivoli Gardens. Specifically, two Smart Zones were established, covering a total curbside length of 41.6 meters. Each Smart Zone accommodated either two standard-sized delivery vehicles or one large truck for unloading. Six sensors were installed (three per Smart Zone) and three traffic signs were mounted on existing poles, providing user instructions and a link to the booking application.

Implementation results indicated: (i) 89 bookings occurred during the 71-day pilot, with 60% of users completing deliveries within 15 minutes; (ii) peak booking time was Saturdays around 14:30, showing minimal activity after 17:00; (iii) user feedback collected via Tivoli Management was uniformly positive, validating the usability of the booking system. Further details are available in D4.2 (L'Istituto sui Trasporti e la Logistica, 2024) and D4.3 (LSP, 2025).

In Padua, the DISCOCURB measure involves smart on-demand loading/unloading stalls linked to a digital booking system. These stalls feature programmable message signalling and can only be occupied following advance booking via a web page or mobile application managed by the municipality. Dynamic lockers can utilize these stalls, ensuring space availability for daily deployment while sharing curb space with other traditional freight operators who also have access to the booking platform. The Padua implementation is currently underway, continuing until the end of 2025, with details forthcoming in D5.2 and D5.3.

3.1.2. Smart Routing and Access: Simulating PI-Compliant Access Policies (LEZ/UVAR)

The utilization of cameras and floating car data (FCD) presents considerable capabilities for monitoring, enforcing and optimizing curbside use. Camera-based systems, often integrated with automated license plate recognition, facilitate real-time detection of illegal parking, overstays and unauthorized use of designated zones. This improves enforcement efficiency, freeing human resources for other duties while supplying granular usage data for planning. Floating car data, gathered from connected vehicles, GPS-enabled fleets and smartphone applications, supplements camera surveillance by providing continuous spatio-temporal insights into traffic flow, stopping patterns and curbside dwell times. Combined, these technologies permit cities to generate dynamic heatmaps, identify areas of under- or over-utilization and adjust regulations or pricing schemes accordingly. Privacy, data governance and interoperability continue to present challenges, necessitating careful design for data-sharing agreements and adherence to regulatory frameworks.

Remaining challenges highlighted in the literature include data quality and representativeness (FCD biases towards participating fleets), sensor maintenance, standardized data schemas and strict privacy-preserving data-sharing agreements between cities and private data holders (Ajmar et al., 2019; Comi & Polimeni, 2021; Maxner et al., 2023). Within DISCO, Helsinki utilized FCD in conjunction with cameras, while Barcelona plans to employ a license plate recognition method during autumn 2025.

In Helsinki, the initial concept aimed to develop a real-time monitoring solution for loading zones. Fixed installations were first considered, but limited electricity availability near loading zones prompted the selection of a flexible, battery-based system. This allows the City to collect data from multiple zones over time by relocating equipment. The collected data assists planners in assessing zone usage relative to expectations (e.g., underutilization or misuse like resident parking), thereby supporting data-driven decisions. Real-time loading zone occupancy was subsequently visualized on the Tietorahti application. However, as monitoring existing zones did not address the planning of new ones, Helsinki LL investigated FCD usage to analyze commercial vehicle stopping patterns; this aimed to identify streets with high stopping activity lacking designated loading zones, supporting data-informed site selection.

Despite a political decision halting LEZ planning, Helsinki LL conducted a simulation to evaluate potential CO_2 reductions in the logistics sector. The analysis employed the platform of XYZT and a Bridgestone LCV origin-destination dataset to simulate LEZ impacts on logistics emissions. Additionally, FVH surveyed local LSPs, collecting data from over 1,500 individual vehicles. In total, combining the acquired FCD dataset and survey data, the simulation encompassed approximately 15,000 individual trips. The simulation included estimations for the proportion of EVs versus gas/diesel LCVs affected and calculated potential CO2 reductions assuming LSPs replaced restricted vehicles with EVs.

Collected data comprised vehicle counts, occupancy rates, vehicle types and counts of pedestrians and cyclists near loading zones. A significant finding from this data was the frequent misuse of monitored loading zones for private car parking, impeding unloading operations. This information proved particularly useful for city planners, who are now devising actions to reduce illegal resident parking.

Helsinki LL also surveyed professional LSP drivers via Tietorahti, gathering valuable service feedback; around 60% found the real-time occupancy visualization helpful in daily operations, facilitating smoother deliveries, while 40% considered the service somewhat unnecessary because route planning often occurs days before actual delivery.

In Barcelona, during autumn 2025, a new curb management model supporting the SUMP 2024 and the SuperBlock plan will be implemented. This model will regulate loading and unloading bays, optimizing space utility to improve logistics operations and urban space usage. It seeks to enhance the last-mile transportation system of the city by digitally managing parking slots for logistics. The AI tools utilized are applicable in other urban mobility contexts, enabling parking slot pre-booking, flexible space management and pricing tools for carriers. Similar to Padua, further details will become available in D5.2 and D5.3.

3.2. Analysis of DISCOCURB PI-Enabling Enhancements

This section utilizes the five-step framework defined in Chapter 2 for the DISCOCURB tools previously introduced. It offers a systematic, retrospective analysis detailing how these tools underwent enhancement and validation via the Starring Living Labs to align them with Physical Internet principles.

Step 1: Foundational Analysis and PI Requirement Definition

The foundational requirement for DISCOCURB, established during the initial analysis of the project, aimed to address the "inefficiency and suboptimal use of curbside spaces in urban areas". The objective involved creating a data-driven, dynamic process for managing this shared asset, shifting beyond static regulations toward a system capable of responding to real-time demand and optimizing for overall efficiency.

Step 2: Solution Identification and Baseline Alignment with PI Roadmap

Two primary categories of existing tools were identified to satisfy these requirements. First, Dynamic Booking and Management Tools were considered, with the "Coding the Curbs" platform selected as a representative example; in its "as-is" condition, it functioned as a commercial solution for booking defined spaces, conceptually aligned with the "Access and Adoption" building block within the ALICE PI Roadmap. Second, Smart Monitoring and Simulation Tools were identified, encompassing existing technologies such as camera/sensor monitoring, FCD analysis and the Tietorahti map service; in their baseline state, these tools aligned with the data and network layers pertaining to the PI Roadmap, specifically concerning the **"Logistics Networks"** and **"Access and Adoption"** building blocks.

Step 3: Analysis of PI Enhancements from Living Lab Implementation

The Living Labs independently made crucial enhancements to align these tools with PI principles:

- In Copenhagen, the key enhancement for the **"Coding the Curbs" platform** was its deployment in a complex, multi-stakeholder environment, converting it from a simple reservation system into a tool for managing a shared, PI-compliant resource among multiple, independent logistics operators.

- In Helsinki, the enhancements focused on data integration. The project facilitated the integration of **live occupancy data** from new, battery-powered monitoring systems directly into the **Tietorahti map service**. This transformed Tietorahti from a static navigation tool into a dynamic, PI-compliant decision-support tool. The **FCD analysis** was also enhanced by combining it with other GIS data layers to enable data-driven planning for new shared infrastructure.

Step 4: Validation through Living Lab Results

The empirical evidence from the Living Labs validated the success of these enhancements:

- The "Coding the Curbs" deployment in Copenhagen was validated by its successful 71-day pilot, which handled **89 distinct bookings** with **entirely positive user feedback**, proving its viability for managing a shared logistics space.

- The Tietorahti integration in Helsinki was validated by a user survey, where **60% of professional drivers** found the real-time occupancy visualization useful. The monitoring data also provided the first objective evidence of loading zone misuse, giving city planners the data needed to improve the management of these shared spaces.

Step 5: Synthesis and Identification of PI-Enabling Components

The analysis of these validated enhancements reveals several key PI-enabling components within the DISCOCURB toolkit:

- **Digital Marketplace Functionality:** The ability to create a reservation system for a shared physical asset is a core component, transforming public space into a digitally manageable, bookable resource, which is fundamental to the PI principle of shared assets and aligns with the **"Access and Adoption" building block.**

- **Real-Time Data Integration and Visualization:** The enhancement of the Tietorahti app demonstrated that the integration and visualization of live data on shared infrastructure is a crucial PI-enabling feature. It provides the transparency necessary for stakeholders to operate efficiently within a shared **Logistics Network.**

- **Data-Driven Strategic Planning:** The enhancement of FCD analysis to include other data layers creates a PI-compliant planning tool. It enables the strategic optimization of the physical network of **Logistics Nodes** (the loading zones) based on empirical evidence of demand.

- **Establishment of a Digital Governance Model:** The successful deployment and positive user feedback for the "Coding the Curbs" platform in Copenhagen validated a new, digital-first governance model for managing access to shared public assets like the curb, directly contributing to the **"Governance"** building block by defining the rules and processes for shared use.

4. Tools for Proximity-Based Logistics (DISCOPROXI)

4.1. Introduction to the DISCOPROXI tools

The DISCOPROXI innovation, centered on establishing efficient and sustainable last-mile logistics via shared infrastructure such as micro-hubs and lockers, utilized two main categories of tools i) analytical planning models and ii) operational delivery platforms.

4.1.1. Planning and Optimising Micro-hub & Locker Networks

In the Thessaloniki Living Lab, the DISCOPROXI innovation concentrated on mitigating the inefficiency in last-mile delivery arising from the uncontrolled expansion of private, provider-specific parcel lockers. This fragmented system results in redundant delivery journeys and inequitable service access. The objective was to lay the groundwork for a regulated, shared and more efficient public network of micro-hubs, employing a dual approach that merged data-driven strategic planning with stakeholder collaboration and physical implementation.

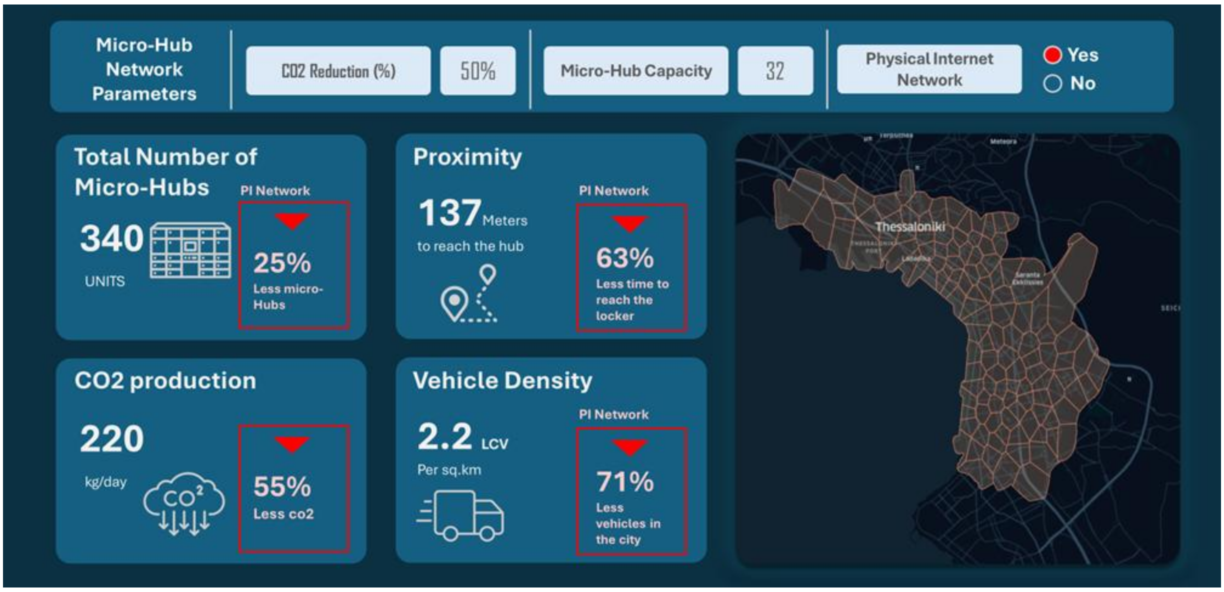


Figure 2: The case of replacing the private parcel locker networks with a shared one and the expected impacts (the area of Thessaloniki)

The initial phase entailed mapping all extant public and private Automated Parcel Machines (APMs) to establish a distinct baseline for the current infrastructure. Subsequently, an optimal facility location tool was developed and utilized to design a novel, shared public network. Effective functioning of this tool necessitated a comprehensive dataset to model the logistics demand of the city. Project partner ACS Courier supplied this essential data, which included the daily quantity of parcels for delivery within the city; the specific geospatial distribution for the final delivery demand points (retrieved from live ACS data); the market share for each courier company operating within the area; and the location of existing depots (dynamically retrieved from the database of the Hellenic Telecommunications and Post Commission (https://www.eett.gr/en/)) along with the average number of parcels delivered per vehicle.

Employing these aforementioned data sources, the tool calculated the optimal number and general locations for new shared lockers required to meet the demand of the city while adhering to specific sustainability objectives. The second phase centered on stakeholder collaboration to translate the output of the model into a practical implementation. Following the identification of optimal locations by the analytical tool, a series of discussions occurred with the Municipality of Thessaloniki. During these meetings, proposed locations underwent review against the strategic criteria of the city, such as avoiding areas with high pedestrian traffic, ensuring accessibility for individuals with disabilities and guaranteeing the safety of installation sites. After this collaborative review, the Municipality granted final approval for two smart lockers in public spaces within the borders of the Municipalities.

4.1.2. Platforms for Sustainable Last-Mile Delivery

Ghent

The overarching goal for this measure involved developing and demonstrating the functionality of an Urban Access Control (UAC) system to improve logistics efficiency, safety and sustainability. This was achieved by integrating digitised Urban Vehicle Access Regulations (UVARs) via a developed API into a logistics planning tool from a provider of Transport Management Systems. This configuration allowed relevant access regulations to be displayed directly within the interface of the Transport Management System for utilization by logistics service providers (LSPs). During the implementation phase, two logistic operators physically tested the solution. Insights from D4.2 indicate that the UAC system positively affects safety and logistics planning concerning stress reduction. Its implementation highlighted numerous sources of risk to road safety and underscored the complexity associated with urban deliveries and the stress experienced by drivers. A high-level architecture is depicted below.

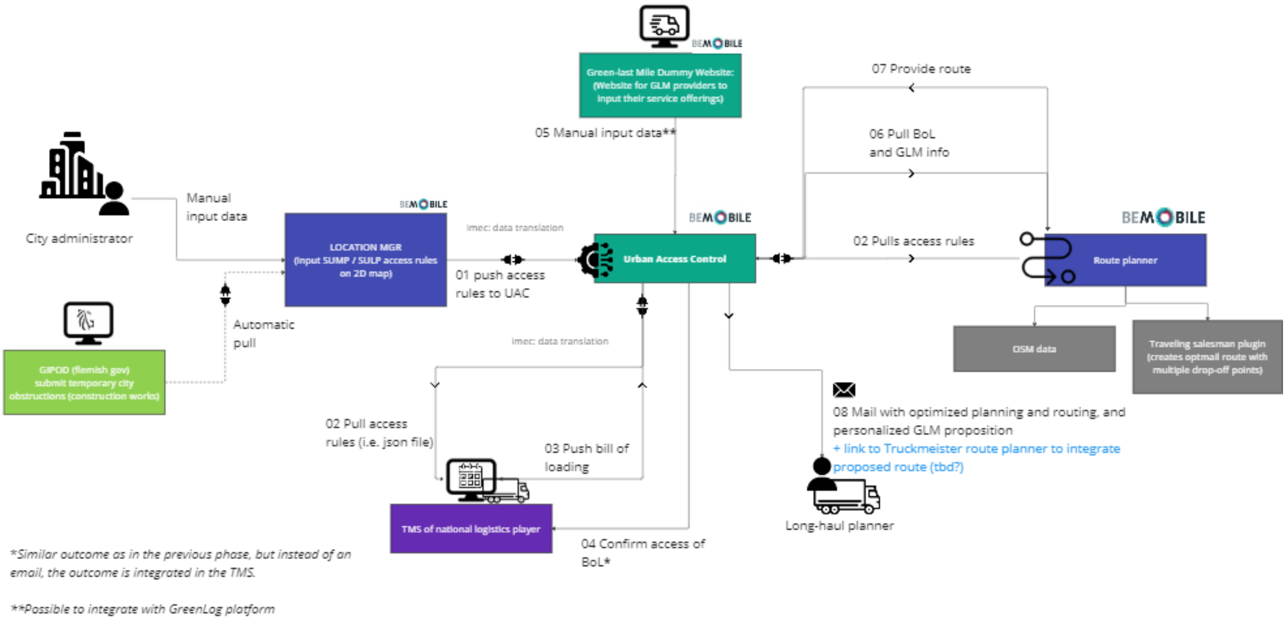


Figure 3: A high-level architecture of Ghent's platform for sustainable last-mile delivery

The subsequent figure illustrates the functional architecture and a user story for the UAC system, offering an overview of the functionalities of the tool, for instance, how the use of Green Last Mile (GLM) providers is suggested to LSPs by providing personalized information pertinent to their delivery.

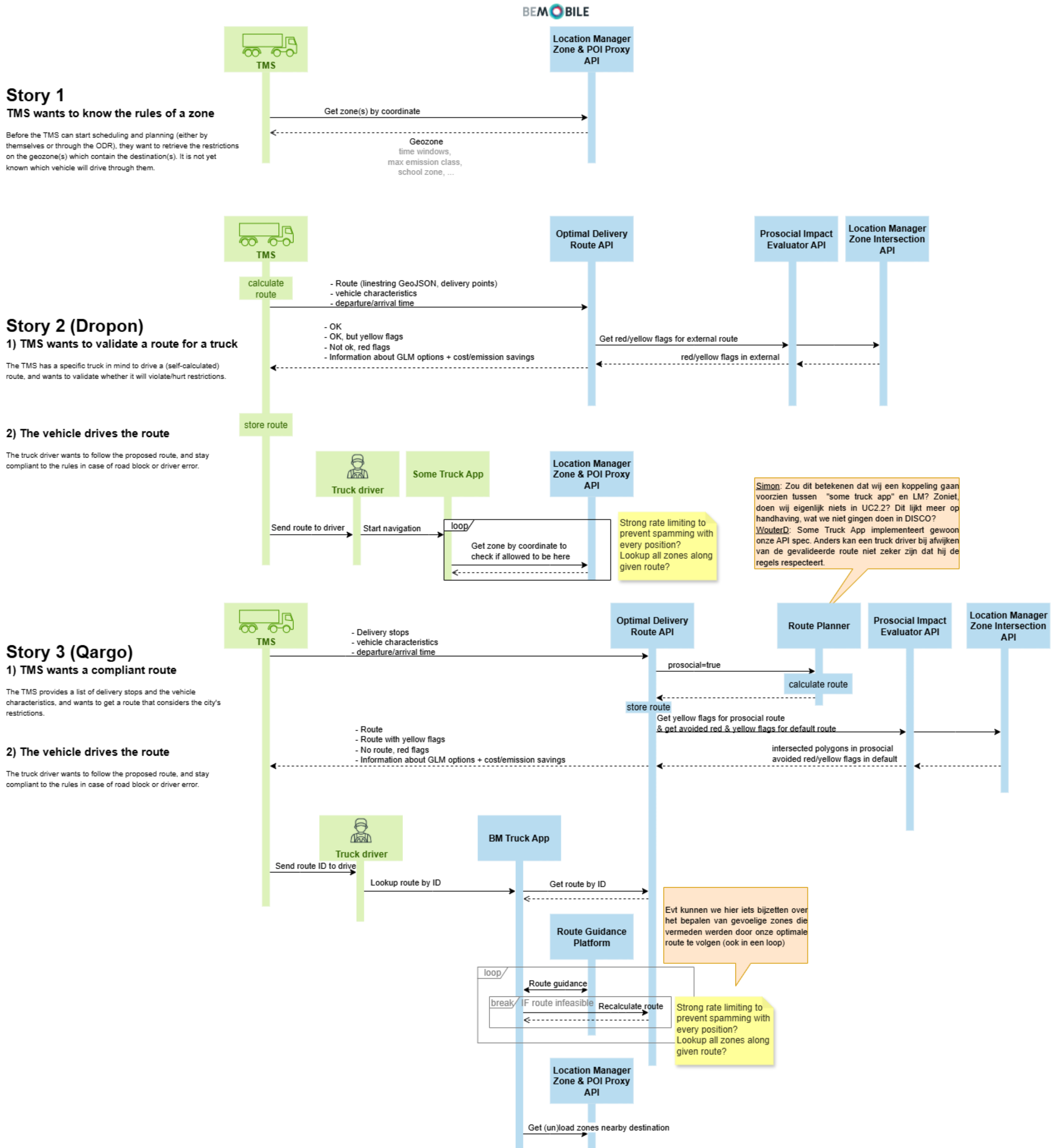


Figure 4: Functional Architecture and User Journey for the Ghent UAC System, Illustrating the Process from Route Planning to Green Last Mile Recommendation

Driver	LSP planner	TMS (Dropon)	UAC
	I enter a number of delivery locations and my available fleet into the TMS		
		I solve the multi-vehicle question and calculate delivery route(s)	
		I wish to validate a delivery route against the access/delivery rules of the city of interest	
		I perform a \validate call to the UAC. As input I provide the route as a linestring, and optionally a departure time for that route. I can also optionally provide GLM options.	
			I use the truck route planner and the Location Manager rules to validate the input route. I can thus find: • Warnings: e.g. “you will pass an LEZ, you need X type of vehicle/fuel” or “your route passes a school street, which is only accesible

Figure 5: User functional path of the Ghent DISCOPROXI Solution (1/2)

			between hours X and Y" • Errors: "Your route passes a car-free zone, which is not accessible at this time, even with a permit" I return these warnings and/or errors to the TMS as machine-readable information
			Additionally, I calculate the GLM options and return them in the same response to the TMS
		I communicate the errors and/or warnings to the planner (e.g. as a visualization)	
		Additionally, I can choose to send the GLM options to the planner	
	In case I receive an error for a certain route, I can choose to resubmit that route with a change (e.g. delayed departure time, removal of a delivery location,…)		
	I have validated delivery route(s) for all my deliveries. I can choose to send one to my driver.		
I use any navigation app to follow my validated deliver route (e.g. Be-Mobile truck navigation app)			
If I use Be-Mobile's navigation app: when approaching a delivery destination, the app will show and direct me to nearby unloading parking spaces			

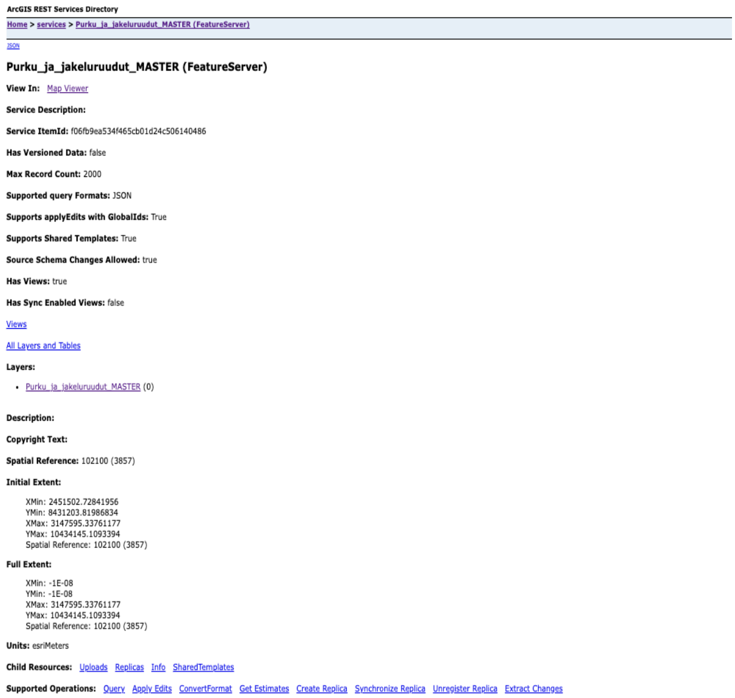
Figure 6: User functional path of the Ghent DISCOPROXI Solution (2/2)

Helsinki

The Helsinki Living Lab planned and implemented a real-time monitoring solution for loading zones within the City of Helsinki. This measure centered on the Tietorahti tool, which was important in visualizing data collected from the loading zones. The tool assisted professional logistics drivers in daily operations by supplying real-time information regarding the occupancy of monitored loading zones.

The requirement for this solution, identified collaboratively with the logistics coordinators of the city, initially contemplated fixed sensor installations. However, stakeholder discussions and a city-led electricity mapping exercise confirmed the necessity for a more practical, battery-powered movable system. This flexible arrangement enabled the city to gather data from various loading zones over time by simply relocating the equipment. The collected data (including vehicle counts, occupancy rates and vehicle types) proved valuable for city planners in assessing whether zones functioned as intended, were underutilized or misused (particularly by private cars).

A significant aspect of the measure involved the Tietorahti tool. Integrating sensor data into the Tietorahti application (a widely used logistics map service accessed by nearly 20,000 drivers daily) permitted real-time visualization of loading zone occupancy. This represented a new feature for the Tietorahti tool, offering an additional service to drivers. Integration between various sensor platforms and the Tietorahti tool occurred through collaboration involving Tietorahti and sensor service providers, coordinated by Forum Virium Helsinki. Tietorahti constructed an ArcGIS REST API for sensor providers, utilized for transmitting real-time occupancy status information from sensor-equipped loading zones, which was then visualized in real-time within the Tietorahti tool. All three sensor providers established Tietorahti integration during the implementation of the measure.



ArcGIS REST Services Directory

Home > services > Purku_ja_jakeluruudut_MASTER (FeatureServer)

JSON

Purku_ja_jakeluruudut_MASTER (FeatureServer)

View In: Map Viewer

Service Description:

Service ItemId: f06fb9ea534f465cb01d24c506140486

Has Versioned Data: false

Max Record Count: 2000

Supported query Formats: JSON

Supports applyEdits with GlobalIds: True

Supports Shared Templates: True

Source Schema Changes Allowed: true

Has Views: true

Has Sync Enabled Views: false

Views

All Layers and Tables

Layers:

- Purku_ja_jakeluruudut_MASTER (0)

Description:

Copyright Text:

Spatial Reference: 102100 (3857)

Initial Extent:

XMin: 2451502.72841956
YMin: 8431203.81986834
XMax: 3147595.33761177
YMax: 10434145.1093394
Spatial Reference: 102100 (3857)

Full Extent:

XMin: -1E-08
YMin: -1E-08
XMax: 3147595.33761177
YMax: 10434145.1093394
Spatial Reference: 102100 (3857)

Units: esriMeters

Child Resources: Uploads Replicas Info SharedTemplates

Supported Operations: Query Apply Edits ConvertFormat Get Estimates Create Replica Synchronize Replica Unregister Replica Extract Changes

Figure 7: the Tietorahti ArcGIS REST Services Directory

Loading zone monitoring sensor providers also shared collected data as a data dump, subsequently added to the Helsinki Mobility Catalogue and the Urban Freight Dataspace as an openly available dataset.

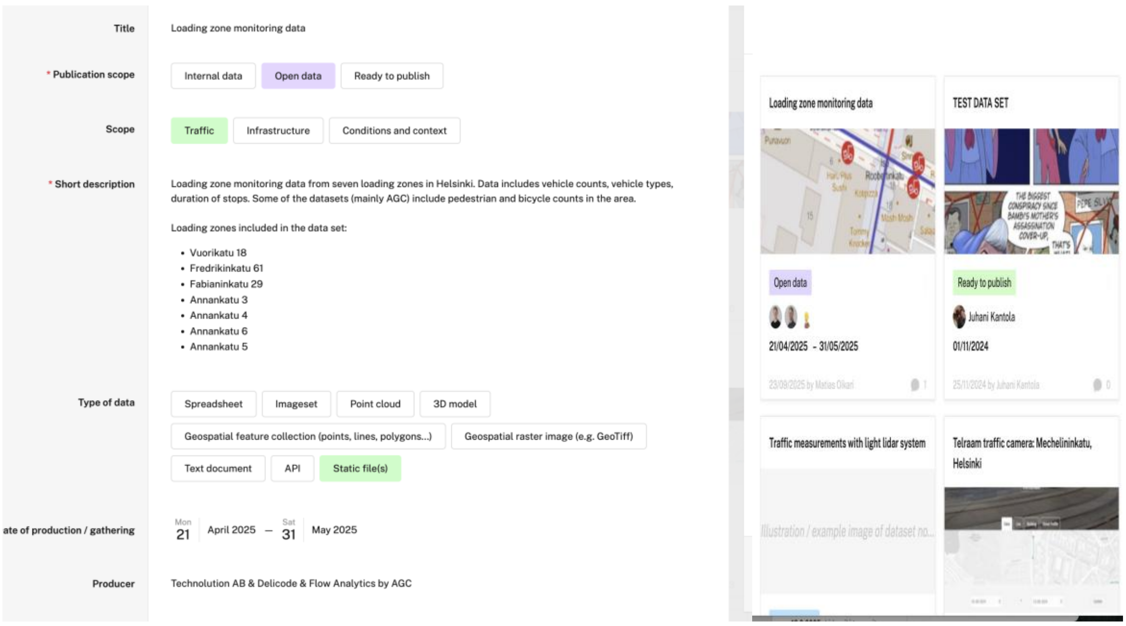


Figure 8: the data set is added to the Helsinki Mobility Data Catalog, which is still under development.

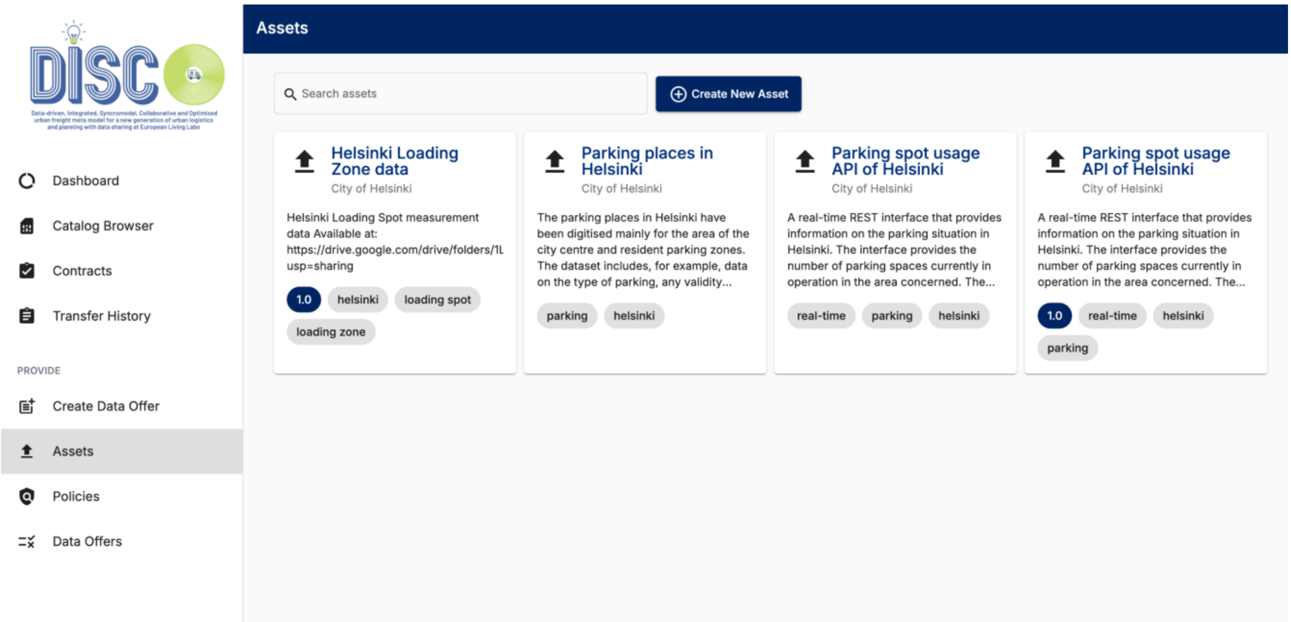


Figure 9: the Loading Zone Data set in the Urban Freight Data Space

This integration demonstrated that real-time information could be shared easily and effectively with drivers, potentially reducing unnecessary driving and congestion, via a tool like the Tietorahti application. A survey of professional drivers confirmed the utility of the Tietorahti application, with approximately 60% of respondents finding the real-time occupancy visualization highly useful for enhancing the smoothness and efficiency of delivery operations. Although 40% considered it somewhat unnecessary due to advance route planning, the overall positive impact of the application was evident.

Beyond its benefits for drivers, the app and the sensor data displayed provided valuable insights for city planners. A major finding from the collected data was that many monitored loading zones were suffering from private car parking, a significant hindrance to effective loading and unloading operations. This information has been handed over to the Urban Environment Division of the City of Helsinki, which is now planning actions to address illegal parking at these zones as part of the upcoming strategic plan for loading zone development.

4.1.3. The Digital Twin as a Strategic Planning Tool for Micro-hubs

In the Copenhagen Living Lab, the DISCOPROXI measure concentrated on developing a sustainable business model for an open, shared consolidation hub, utilizing the existing "Køleskabet" micro-hub as a case study. The principal enabling tool was not an operational platform but the Digital Twin for Urban Logistics, employed as an analytical and simulation engine to evaluate the feasibility and city-wide impact of such a hub.

The core activity involved enhancing the Digital Twin to model a new scenario: consolidating last-mile deliveries through a shared micro-hub combined with a cargo bike fleet for the final delivery segment. This necessitated considerable effort in organizational interoperability, culminating in 13 data-sharing partnership agreements with major logistics companies to supply the model with real-world, anonymized route data. The value of the tool resided in providing data-driven evidence for policymaking; it successfully simulated potential impacts of this PI-led concept on CO_2 emissions and logistics efficiency, directly informing the development of the new Sustainable Urban Logistics Plan of Copenhagen.

4.2. Analysis of DISCOPROXI PI-Enabling Enhancements

Step 1: Foundational Analysis and PI Requirement Definition

The fundamental requirement for DISCOPROXI, established in the initial analysis of the project, involved addressing inefficiencies in last-mile delivery within dense urban areas by creating a system comprising shared, proximity-based logistics infrastructure (micro-hubs and lockers). The objective aimed to transition from fragmented, single-provider networks toward a collaborative model that streamlines operations, reduces vehicle kilometers and encourages the adoption of green transport modes, as detailed in section 4.1.

Step 2: Solution Identification and Baseline Alignment with PI Roadmap

Four primary categories of existing tools were identified to meet these requirements. First, Analytical Planning Models (Thessaloniki) began with an existing continuous approximation model for calculating the optimal number of logistics facilities; this "as-is" tool aligns with the strategic planning aspect related to the "Logistics Nodes" building block from the ALICE PI Roadmap. Second, Operational Route Management Platforms (Ghent) used the Urban Access Control (UAC) system as a baseline; in its "as-is" state, it served as a compliance tool for road transport, aligning with the operational management aspect pertaining to the "Logistics Networks" building block. Third, User-Facing Navigation Applications (Helsinki) identified the Tietorahti map service as a widely adopted commercial navigation application; in its baseline state, it provided static and real-time traffic data, aligning with the "Access and Adoption" building block. Fourth, Strategic Simulation Tools (Copenhagen) recognized the Digital Twin platform as an advanced tool for modeling complex logistics networks, aligning with the strategic planning associated with "Logistics Networks".

Step 3: Analysis of PI Enhancements from Living Lab Implementation

The Living Labs independently implemented important enhancements to align these tools with PI principles. In Thessaloniki, the analytical model underwent significant enhancement to analyze specifically the impact of transitioning from multiple private locker networks to a single, shared public network; this transformed a generic optimization model into a planning tool compliant with PI that provides a distinct business case for shared infrastructure. In Ghent, the UAC system received enhancement via a GLM recommendation engine; this new module proactively suggested more sustainable alternatives compliant with PI (such as a micro-hub and cargo bike combination) instead of traditional truck deliveries, elevating the tool from a reactive compliance checker to a dynamic network orchestrator. In Helsinki, the Tietorahti map service was enhanced through the integration of a live data feed concerning shared infrastructure status (loading zone occupancy) via a newly developed ArcGIS REST API; this converted a general navigation tool into a decision-support tool compliant with PI, granting drivers real-time visibility into the shared logistics network. In Copenhagen, the Digital Twin underwent enhancement by integrating real-world, multi-stakeholder route data from 13 logistics partners and establishing a new scenario to model specifically the city-wide impact associated with a shared consolidation hub; this transformed a generic simulation platform into a strategic planning instrument compliant with PI for collaborative logistics.

Step 4: Validation through Living Lab Results

The empirical evidence from the pilots validated the success of these enhancements:

- The enhanced analytical model was validated in **Thessaloniki**, where its data-driven analysis guided the successful collaboration with the Municipality and the physical deployment of **two new public smart lockers**.

- The UAC's GLM recommendation engine in **Ghent** was validated through physical tests with two logistics operators, who confirmed its positive effect on logistics planning and driver stress reduction.

- The Tietorahti integration in **Helsinki** was validated by a user survey of its professional user base, where **60% of respondents** found the new real-time occupancy feature highly useful for their daily operations.

- The enhanced Digital Twin in **Copenhagen** was validated by its ability to provide powerful, data-driven evidence for the city's new SULP, including a key finding that a shared hub with cargo bikes could reduce CO_2 emissions by 90% for those routes.

Step 5: Synthesis and Identification of PI-Enabling Components

The analysis of these validated enhancements reveals several key PI-enabling components within the DISCOPROXI toolkit:

- **Data-Driven Network Design:** The ability of the analytical model to plan an optimized, *shared* network of physical nodes (lockers and micro-hubs) is a foundational component for building the **"Logistics Nodes"** of the PI.

- **Dynamic Network Orchestration:** The capacity of the UAC system to proactively suggest and facilitate modal shifts based on real-time conditions is a key component for managing PI **"Logistics Networks"**.

- **Real-Time Infrastructure Visibility:** The integration of live data on shared assets into user-facing applications like Tietorahti is a crucial component for driving the **"Access and Adoption"** of the PI ecosystem.

- **Collaborative Scenario Simulation:** The ability of the Digital Twin to integrate data from multiple, independent commercial actors to model the system-wide benefits of collaboration is a key component for making the strategic case for the PI.

- **Facilitation of Municipal Governance Agreements:** The use of the validated optimal location tool in Thessaloniki provided the data-driven evidence necessary to achieve municipal agreement for installing shared logistics infrastructure (lockers) in public spaces. This demonstrates a key aspect of the **"Governance"** building block with

	establishing the necessary stakeholder agreements and permissions for deploying shared PI infrastructure.

5. Tools for Collaborative Logistics Spaces (DISCOESTATE)

The DISCOESTATE innovation addresses a critical challenge in modern urban logistics: the scarcity and high cost of space. As e-commerce drives demand for logistics facilities closer to urban centers, the traditional model of developing new, single-tenant warehouses becomes increasingly unsustainable. DISCOESTATE proposes a Physical Internet-led solution: the dynamic and shared use of existing, underutilized building spaces. This involves retrofitting multi-tenant hubs and enabling the temporary use of spaces like exhibition centers or vacant retail properties for logistics operations. The enabling tools for DISCOESTATE are designed to create a digital marketplace that makes this vision of shared space a practical reality.

5.1. Introduction to the DISCOESTATE tools

5.1.1. The Warehouse-as-a-Service (WaaS) Platform

The primary enabling tool for the DISCOESTATE innovation is the WareM&O platform (accessed through this link: http://waremo.tredit.gr:8084/vfc/home), a virtual freight center designed to create a digital marketplace for underutilized storage spaces, with a particular focus on urban areas. The platform's main goal is to maximize the use of existing but underused storage facilities, enhance the efficiency of storage operations, ensure a fair pricing system, and foster a collaborative environment for logistics stakeholders. It functions as the digital foundation for the DISCOESTATE concept, transforming static buildings into dynamic, sharable assets in the urban logistics network.

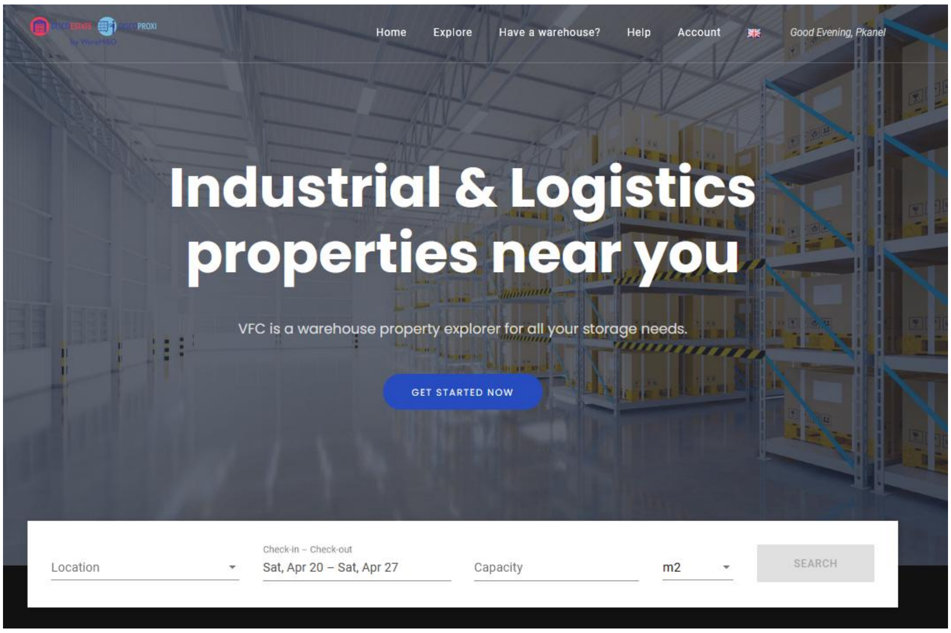


Figure 10: The User Interface of WareM&O by DISCOESTATE & DISCOPROXI

The WareM&O platform is designed as a two-sided marketplace, providing distinct functionalities for both space owners and space users.

- **For space owners,** the platform allows for the effective management and monetization of their assets. Owners can list their available storage spaces, detailing the size, features, pricing, and availability. They can manage these listings, including updating information, communicating with potential renters, and even partitioning larger areas into smaller, bookable compartments. The system streamlines the booking process by allowing owners to handle reservation requests, and it provides access to performance metrics and analytics, including key performance indicators that track storage utilization and booking trends, enabling them to optimize their operations.
- **For space users** (such as logistics service providers), the platform offers a tailored search and booking experience. Users can find storage spaces that meet specific criteria such as size, location, amenities, and rental duration. Once a suitable space is identified, users can directly book it through the platform and confirm their agreement with an e-signature. A reservation management system provides users with the ability to view, amend, or cancel their bookings, offering flexibility in managing their storage requirements. Additionally, the platform supports collaborative storage by facilitating communication among users interested in sharing spaces and includes a feedback and rating system to enhance transparency and trust within the community.

The architecture of the WareM&O platform is designed for real-time management and secure data exchange, as depicted in Figure 11. The WareM&O platform is built on a multi-tiered, service-oriented architecture designed to operate as a high-availability Virtual Freight Centre. The presentation tier consists of a client-facing User Interface that handles all user interactions, session management, and role-based access control. The core business logic resides in the application tier, which is decomposed into three distinct services. The Planning Service executes real-time matching algorithms to couple supply and demand for storage and manages the lifecycle of digital storage certificates. The Booking Service orchestrates the entire transactional process, from reservation to invoking a dynamic pricing module and automating the generation of digital contracts. Complementing these, there is the Visualisation Service, an analytics and reporting component that aggregates operational data to render user-facing dashboards for Key Performance Indicator (KPI) evaluation and personal transaction histories. The application tier interfaces with a dedicated persistence layer, the Database, which stores all transactional data, supply/demand states, legal document templates, and pricing models. A critical architectural feature is the data ingestion mechanism for real-time supply availability, which supports two modes: direct API-driven synchronization with third-party Warehouse Management Systems, Terminal Operating Systems, and ERPs, and manual data entry via a secure portal. The entire platform is managed through the BackOffice, an administrative backend that provides tooling for data management (CRUD operations), systems analysis, and audit logging of user processes.

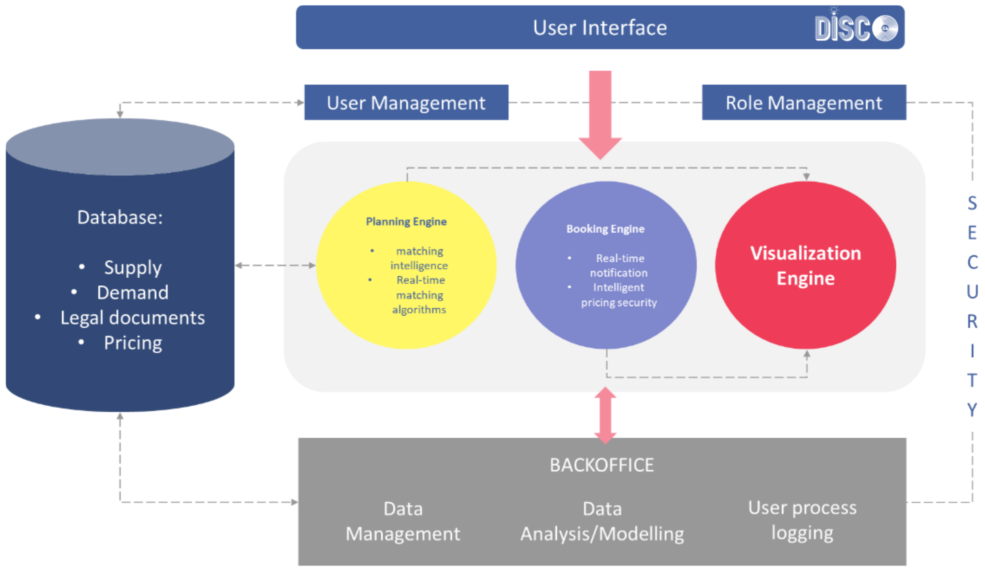


Figure 11: The architecture of the WareM&O platform

At its core is a central database that stores detailed information on the real estate assets, which is accessed through a series of APIs.

The WareM&O Platform itself acts as the central hub, utilising blockchain technology to ensure the security, transparency, and immutability of data transactions, which are crucial for maintaining trust in a multi-stakeholder environment. To be adopted for the DISCOESTATE innovation, the platform was significantly enhanced in several key areas:

- **Standardisation and Enrichment of Data Model:** The platform's data model was standardised and enriched to align with the specific requirements of both the FM Logistic dataset and the TIF-HELEXPO real estate system. The main purpose of this adjustment was to make sure that data related to warehouse capacity and building specifications can be seamlessly integrated and accurately reflected within the platform.
- **Proof of Concept for Electronic Agreements:** A proof of concept was developed for a secure electronic agreement system. This digital solution, enhanced with blockchain for e-signatures, was designed to simplify and streamline the process of finalizing contracts between facility owners and tenants, reducing paperwork and increasing efficiency.

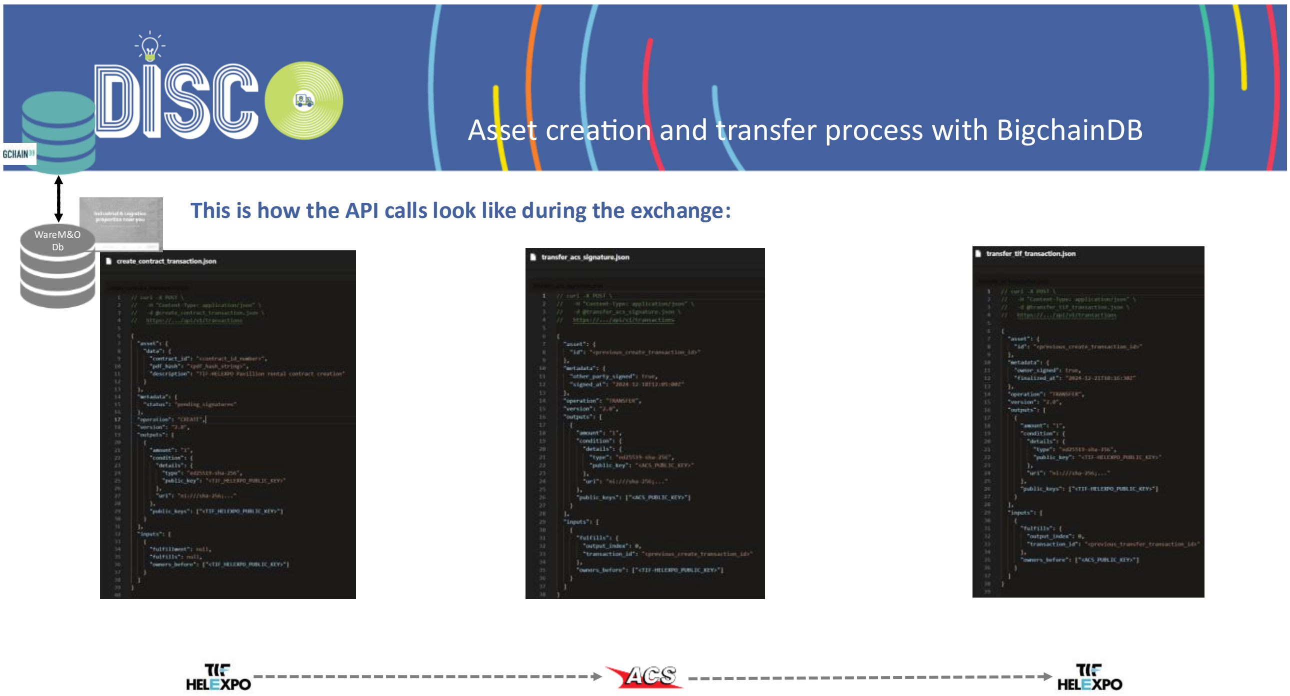


Figure 12: The asset creation and transfer process within WareM&O by DISCOESTATE & DISCOPROXI e-signature functionality

- **Interface with Thessaloniki's UL Dataspace:** A crucial enhancement was the development of an interface to connect the WareM&O platform with Thessaloniki's local urban Logistics dataspace (as shown in Figure 11). This integration facilitates the sharing and utilization of urban logistics data, enhancing the platform's functionality and its ability to support coordinated logistics solutions in the city.

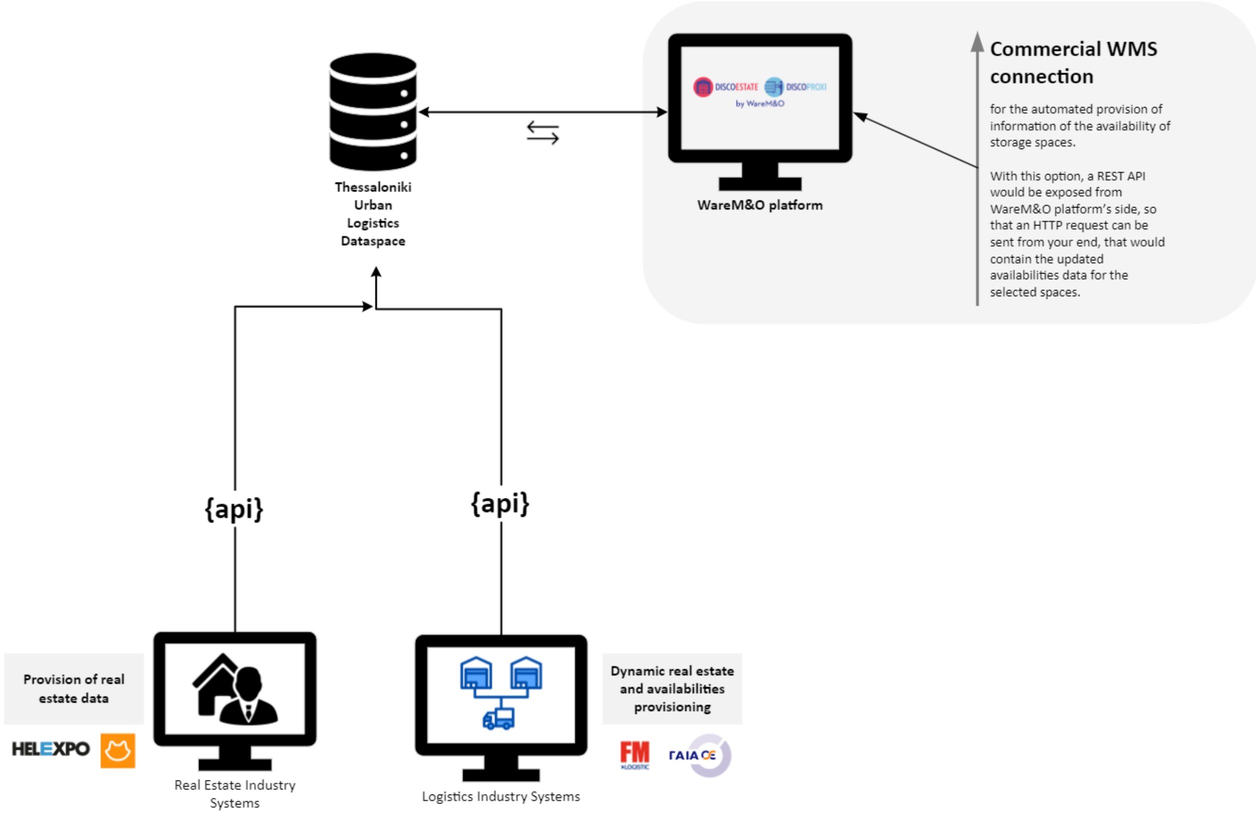


Figure 13: Architectural structure of WareM&O by DISCOESTATE & DISCOPROXI platform

This database is accessed through a series of Application Programming Interfaces (APIs). The WareM&O Platform itself acts as the central hub, utilizing blockchain technology to ensure the security, transparency, and immutability of data transactions, which is crucial for maintaining trust in a multi-stakeholder environment.

A key feature of the architecture is its capability to integrate with external systems. A commercial Warehouse Management System (WMS) connection highlights the system's ability to communicate directly with the operational systems of logistics and storage facilities, providing automated updates on space availability via a REST API. This allows for a dynamic and accurate reflection of the real-world status of the listed spaces, ensuring that space utilization is optimized and that the information provided to users is always up to date.

5.1.2. Integrating with Commercial Warehouse and Building Management Systems

A core principle of the Physical Internet is the seamless, real-time flow of information. For a WaaS platform to be truly dynamic, it cannot rely on manually updated listings. It must integrate directly with the systems that manage the physical spaces. This integration allows for live data on space availability to be automatically pushed to the platform, ensuring the marketplace reflects the true, up-to-the-minute status of the logistics network. This moves the platform from a simple classifieds board to a dynamic, PI-compliant operational tool. The DISCO project explored two key integration points: with commercial Warehouse Management Systems and with the bespoke Building Management System of the TIF-HELEXPO exhibition centre.

Connection to Warehouse Management Systems (tested with FM Logistic)

To enable the dynamic management of shared logistics spaces, a core objective of DISCOESTATE, an automated integration was established between the internal Warehouse Management System of FM Logistic and the WareM&O platform. The goal of the integration was to optimise the management of warehouse capacity and create a seamless, automated process for updating the availability of pallet positions on the shared platform.

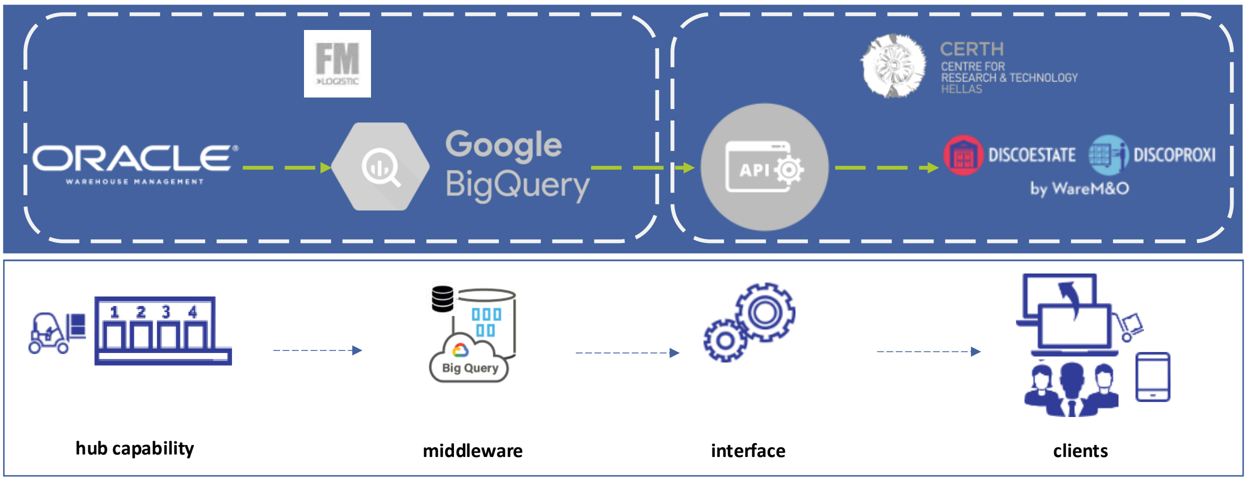


Figure 14: the FM LOGISTIC warehouse availability data flow

The implementation of a middleware application that acts as a bridge between the two systems is responsible for this. Here, operational data from the WMS—such as available pallet positions, capacity by zone and specific product requirements (e.g., dangerous goods, alcoholic beverages and controlled temperature goods)—is first consolidated and stored in Google BigQuery, which serves as the central data hub.

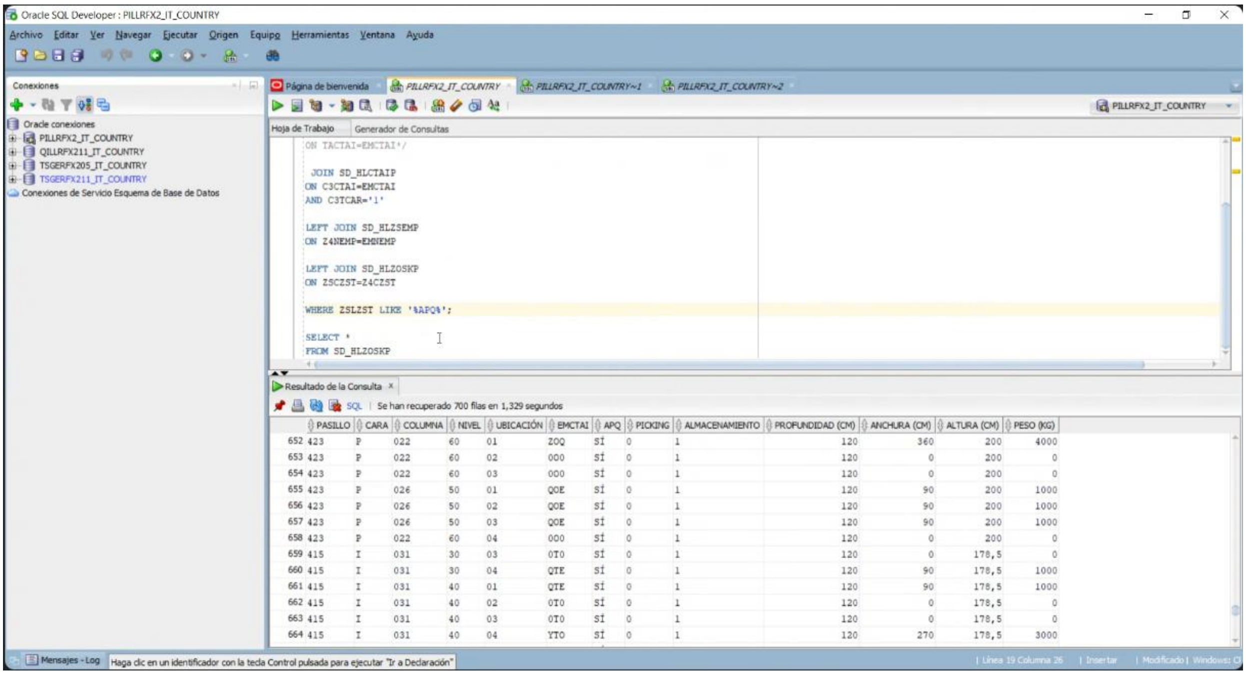


Figure 15: The Oracle SQL developer software used by FM Logistic for DISCOESTATE

On a daily schedule, the middleware executes a set of queries to extract the most recent availability information from BigQuery. This data is then processed and formatted to match the structure required by the application programming interface (API) of the WareM&O platform. The final data payload is sent securely via REST API calls, ensuring accurate and timely updates to the WareM&O interface without the need for manual intervention.

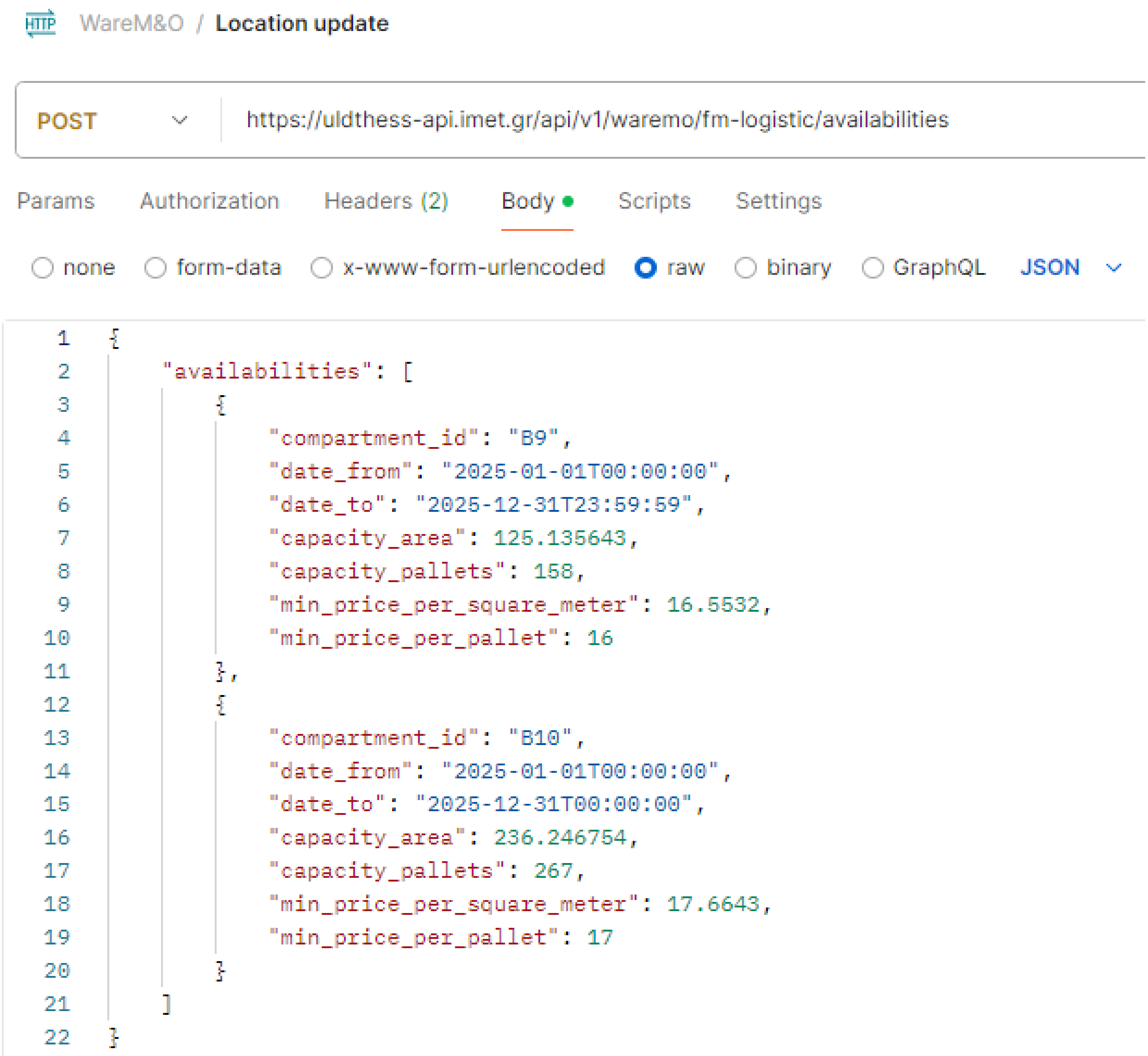


Figure 16: the JSON body that is exchanged between the WareM&O API and FM Logistic.

This automated integration provides several key benefits that align with the principles of the Physical Internet. It ensures real-time accuracy, giving clients up-to-date visibility into warehouse availability and allowing them to reserve space with confidence. It also improves operational efficiency by eliminating manual reporting and reducing the risk of errors. The solution is designed to be scalable across multiple sites and reliable, with automated checks in place to monitor data quality and ensure successful transmissions.

Connection to Pavilion Management Systems (tested with TIF-HELEXPO)

As mentioned during the introduction of DISCOESTATE tools, a key objective for the DISCOESTATE innovation was to enable a dynamic, real-time marketplace for underutilized spaces. To achieve this for the TIF-HELEXPO pavilions, an automated data exchange was required between the internal Pavilion Management System of TIF-HELEXPO and the WareM&O platform. The Pavilion Management System of TIF-HELEXPO contains critical availability data, including the schedule of expos and other events that make the pavilions unavailable, as well as warehousing information such as the total square meters available. A direct, live connection to this internal system was not feasible.

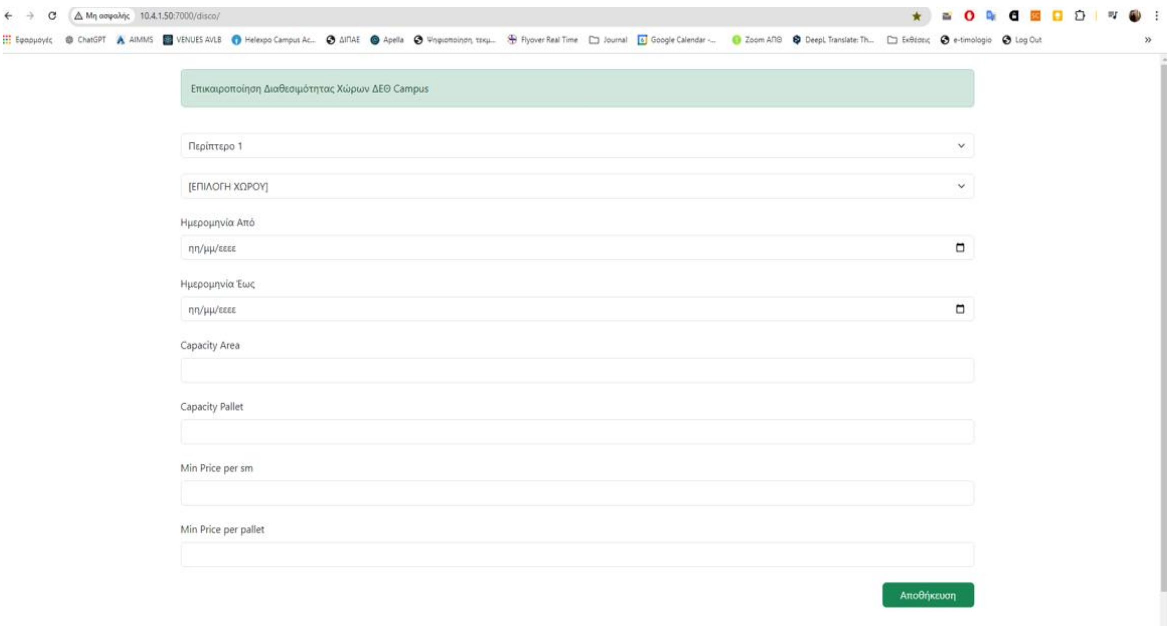


Figure 17: Internal Pavilion (Warehouse) availability management system/web interface

To solve this, TIF-HELEXPO developed a dedicated web service using NodeJS and React to act as an intermediary. This web service securely manages a subset of the information from the Pavilion Management System. The primary function of this web service is to "speak" with the Application Programming Interface of CERTH's WareM&O platform. It is responsible for automatically pushing updates regarding the availability of the pavilions listed on the WareM&O platform. This automated integration ensures that the information on WareM&O accurately reflects the real-world availability of the TIF-HELEXPO spaces, which is a crucial PI-enabling enhancement for creating a reliable and dynamic marketplace for shared logistics assets.

Connecting to DISCO's Data Ecosystem

The WareM&O tool, as a DISCOESTATE tool, is designed as an integral component of Thessaloniki's broader data ecosystem, offering smooth interoperability and real-time information exchange. This is implemented:

- By consuming data from external systems such as Warehouse Management Systems (WMS) and Building Management Systems (BMS) to obtain real-time space availability
- By providing warehouse and space data to the Urban Freight Dataspace (UFDS) for city-wide visibility and strategic planning.

In the Thessaloniki Living Lab, the WareM&O platform was directly connected to the local UFDS instance (https://uldthess.imet.gr) through a standardized REST API. This integration enabled the UFDS to maintain a live, harmonized inventory of all available logistics spaces, including the TIF-HELEXPO pavilions and smart lockers deployed under the DISCOPROXI measure. As a result, the UFDS evolved into a central observatory for urban logistics real estate, supporting data-driven decision-making and policy development.

The technical implementation followed a structured approach. First, Data Space Connectors were deployed on Inlecom's cloud infrastructure for all project Living Labs, including Thessaloniki, to ensure secure and standardized data exchange. Using the Thessaloniki Data Space Connector, a new Data Offer was created for WareM&O space data and published in the UFDS catalog, making it discoverable and reusable by stakeholders. The integration leveraged the open-source Sovity Dataspace Connector (Community Edition EDC), enabling rapid deployment and compliance with interoperability standards. A dedicated REST API was declared by the Thessaloniki Urban Logistics Dataspace, acting as a proxy for the WareM&O platform to fetch real-time warehouse and space data. The endpoint (https://uldthess-api.imet.gr/api/v1/waremo/warehouses) was secured with two authentication HTTP headers to prevent unauthorized access, while metadata and keywords were added to facilitate dataset discovery. Comprehensive documentation was provided to describe the data schema and expected formats. Each API call returns the full, real-time WareM&O dataset and the Data Offer was published with unrestricted access, ensuring that all UFDS users can benefit from this integration.

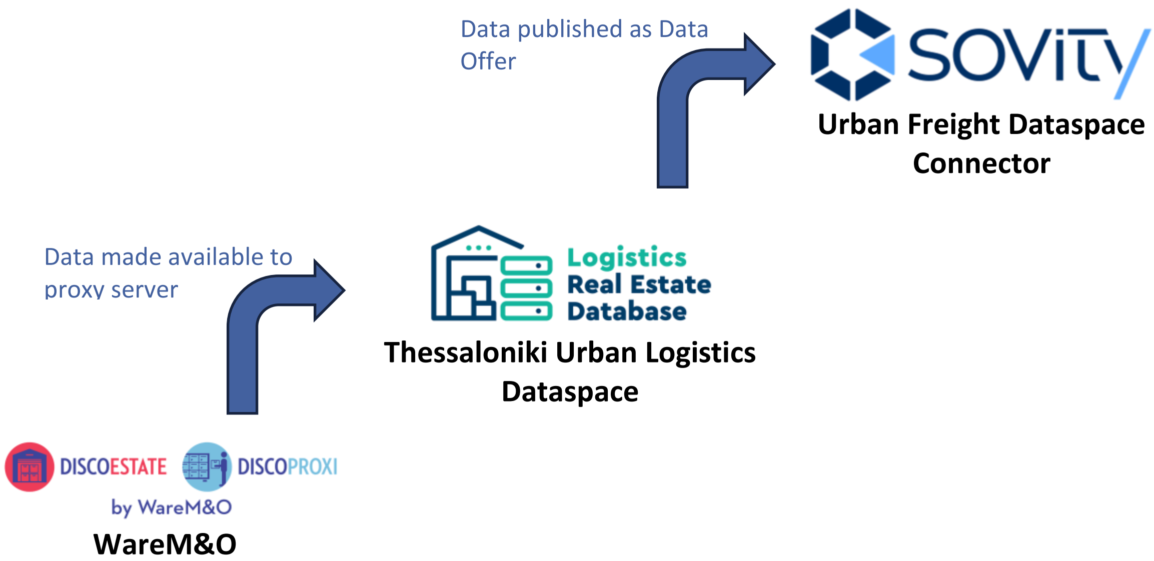


Figure 18: Connecting Thessaloniki LL's DISCOESTATE solution, WareM&O to the Urban Freight Dataspace

5.2. Analysis of DISCOESTATE PI-Enabling Enhancements

Step 1: Foundational Analysis and PI Requirement Definition

The foundational requirement for DISCOESTATE, as established in the project's initial analysis and confirmed by the Thessaloniki Living Lab, was to address the critical challenges of traffic congestion and the lack of available logistics space in the dense urban center. The goal was to create a data-driven process for exploiting underutilized, strategically located spaces to serve as new, shared logistics hubs. This would reduce the number of delivery vehicles entering the city center and improve overall efficiency, in line with the principles of the Meta Model Suite.

Step 2: Solution Identification and Baseline Alignment with PI Roadmap

The primary existing tool identified to meet these requirements was the **WareM&O platform**, a virtual freight centre developed in a previous research project. In its "as-is" state, this platform was a functional prototype for a digital marketplace where owners could list available warehouse space. This baseline functionality aligns conceptually with three core **building blocks of the ALICE PI Roadmap**: **"Logistics Nodes"**, by providing a digital layer for physical infrastructure, and **"Access and Adoption"**, by creating an interface for stakeholders to access these assets and **“Governance”.**

Step 3: Analysis of PI Enhancements from Living Lab Implementation

The Thessaloniki Living Lab made several crucial enhancements to the WareM&O platform to close the "PI Gap" and transform it into a robust, PI-compliant operational tool:

- **Blockchain-Enabled Smart Contracts:** To facilitate a secure and trustworthy agreement between the property owner (TIF-HELEXPO) and the logistics operator (ACS Courier), the WareM&O platform was enhanced with a **blockchain-based e-signature feature**. This moved the platform beyond a simple listing service to a transactional tool capable of managing formal, multi-stakeholder agreements in a PI ecosystem.

- **Integration with Building Management Systems (BMS):** A key operational challenge was that the exhibition centre’s standard operating hours did not meet the 24/7 needs of a logistics depot. To solve this, a direct technical integration was implemented. The TIF-HELEXPO facility was physically upgraded with **RFID readers, cameras and automated barriers**. These systems were connected to the building's BMS, which was programmed to grant automated, secure, round-the-clock access to authorized ACS personnel and vehicles. This enhancement was critical for transforming a static building into a dynamic, shared logistics node.

- **Standardization and Enrichment of Data Model:** To ensure interoperability, the platform's data model was standardized and enriched to align with the specific requirements of both the FM Logistic dataset and the TIF-HELEXPO real estate system.

- **Connection to the Urban Freight Dataspace:** The WareM&O platform was enhanced with an API to connect directly to the local **Urban Freight Dataspace (UFDS)**, enabling it to share real-time data on space availability with the broader city logistics ecosystem.

Step 4: Validation through Living Lab Results

The tangible outcomes of the pilot in Thessaloniki validated the success of these enhancements:

- The most significant validation was the **successful relocation of the ACS Courier central city depot** from the congested historical center to the TIF-HELEXPO pavilion. This real-world outcome, which was a central objective of the pilot, proved the technical and business viability of the DISCOESTATE concept and the effectiveness of the enhanced WareM&O platform.

- The pilot **validated the platform's capability** to not only list but also to successfully broker and manage a complex, real-world agreement between a public-access venue and a major commercial logistics operator, a key finding documented in D4.2.

- The implementation resulted in the creation of a **replicable business model** for the flexible use of underutilized urban spaces, a key exploitable result for other cities looking to implement similar PI-led solutions.

Step 5: Synthesis and Identification of PI-Enabling Components

The analysis of these validated enhancements reveals several key PI-enabling components within the DISCOESTATE toolkit that directly align with the vision of the Physical Internet:

- **Digital Marketplace for Shared Assets:** The core function of the WareM&O platform directly enables the PI principle of **shared assets**, transforming privately or publicly owned buildings into bookable resources on an open network, thus contributing to the **"Logistics Nodes"** building block.

- **Trust and Governance Mechanisms:** The integration of blockchain for secure e-signatures provides a crucial trust and governance mechanism, which is essential for facilitating collaboration between independent actors in a PI ecosystem and aligns with the **"Access and Adoption"** and **“Governance”** building blocks.

- **Physical and Digital Interoperability:** The successful integration of the digital platform (WareM&O) with the physical infrastructure (the BMS and access control systems of the building) is a key component that demonstrates how to create a truly **interconnected PI node**.

- **Data-Driven Decision Making:** The connection of the platform to the UFDS enables **data-driven decisions** by providing a city-wide view of available logistics real estate, which is a foundational element for the **"System of Logistics Networks"** building block.

- **Trust and Governance Mechanisms**: The integration of blockchain for secure e-signatures, validated by the successful TIF-HELEXPO/ACS agreement, provides a crucial trust and governance mechanism essential for facilitating collaboration between independent actors in a PI ecosystem, directly contributing to the **"Governance"** building block by creating a secure and trusted contractual framework.

6. Tools for Multimodal Consolidation (DISCOBAY)

DISCOBAY refers to the part of the DISCO multimodal suite that enables a city to use the latent potential in its multimodal transport system to reduce emissions in city logistics. Whereas other parts of the suite address important issues of public space management, warehousing, microhubs in the city and data sharing, DISCOBAY connects this logistics ecosystem to the networks and suppliers outside of the city. It is a stepwise approach to engage underused transport and warehouse capacity of multimodal nodes near the urban core for green last-mile urban logistics. Examples of multimodal nodes are barge and rail terminals, but also urban quays and suburban cross-docking sites. Stimulating the modal shift away from road transport is an urgent topic, since the external costs of road transport are about three times those in barge and rail - see the Combined Transport Directive, EU, 2025 (92/106/EEC) and the issue of decarbonization of logistics (McKinnon, 2018). About half of those costs concern traffic accidents (29,7%) and congestion (18,8%). A large part of external costs of transport is generated in urban areas (Comi & Savchenko, 2021; F. Russo & Comi, 2024).

The DISCOBAY concept fits the Physical Internet paradigm (Ballot et al., 2014; Cossu et al., 2025; Crainic & Montreuil, 2016; Montreuil, 2011) in the following ways:

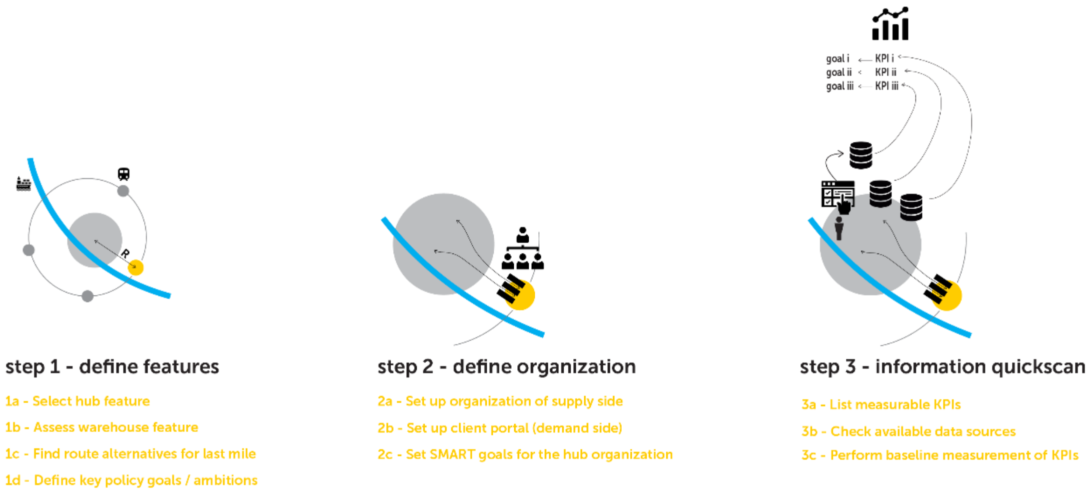
- By consolidating and transhipping urban goods of different supply chains, market segments and companies, across different transport modes in a white label hub or system of hubs;
- By stimulating data-driven operations of both the multimodal hub and the associated warehousing space to optimize the utilization of their capacities to the full.

DISCOBAY has been implemented in three of the DISCO living labs. Copenhagen features a suburban warehousing and fresh food hub where crossdocking is planned between heavy long haul road transport and light electric vehicles for the last mile. Ghent features electric urban waterway barge transportation from suburban hubs, focusing on construction materials as well as parcels in combination with cargo bikes. Thessaloniki features reuse of a freight rail terminal and adjacent warehousing near the centre for urban deliveries. In the following paragraphs the DISCOBAY stepwise approach is explained and positioned in the recent sustainable city logistics literature.

At present, the available transport systems operating in and around cities are not centrally managed and controlled. This is one of the main reasons for suboptimal utilization of infrastructure, means of transport and warehousing (Hesse & Rodrigue, 2004). By pooling resources and orchestrating city-bound flows going through specific hubs, these inefficiencies can be turned into assets to stimulate green last mile transport in the city.

Orchestrating flows of urban freight between multimodal hubs and urban centers has several challenges. In general, the context of each city, transport system and each regional real estate market is quite different. Therefore, DISCOBAY cannot offer a one-size-fits-all approach but rather proposes a modular approach, in which it is possible to include locally available types of solutions, innovations and hubs (Mohammad et al., 2023); and to aim at societal goals that are viable in the local and regional political context, including timelines that fit the actors involved. The approach thus provides flexibility to make custom divisions of tasks concerning initiative, planning and operations; and custom agreements on investments, contracts and procurement regarding logistics service providers, as well as monitoring the impacts of DISCOBAY.

The DISCOBAY stepwise approach, in summary, consists of three main steps and – after these have been implemented – an evaluation and monitoring step (see diagram below).



With regard to the physical infrastructure, cities are quite diverse and often lack specific elements such as cargo bike lanes and microhubs, which need investment (POLIS & Environmental Defense Fund Europe, 2022; Transport Forum International, 2024). Expanding city logistics operations increase congestion and competitive use of routes and public space, which need to be managed (ALICE-ETP, 2020; POLIS & TDA, 2020). Furthermore, policies for noise and emission reduction are increasingly implemented, differently in each city. The urban freight sector consists of many fragmented small players, which makes it challenging to adopt a uniform set of goals and standards in sustainability measures, while it causes delays in policies such as zero-emission zones and urban consolidation hubs (CIVITAS, 2015; TDA, 2019). In each city, use cases for B2B and B2C, environmental problems as well as urban logistics practices need to be assessed to establish adequate societal goals and hub types (Bates et al., 2018; BCI, 2023; Demir et al., 2022; Desrousseaux, n.d.; Sweere, 2023). For these reasons, *Step 1* in the stepwise approach to DISCOBAY includes the definition of the multimodal hub feature, associated warehousing, assessing routes to the urban core, as well as setting key policy goals. These goals can vary from CO2 reduction to road safety in different ranks of priority.

The organizational side of the DISCOBAY solution is challenging as well. The urban logistics sector, as well as the multimodal transportation and warehousing sectors, involve numerous actors with unaligned interests and great operational complexity (ALICE-ETP, 2020; Nefs & Daamen, 2022; TDA, 2019). Public policies on urban logistics have a comparable fragmentation, involving city planners, spatial-economic policies, zero-emission zoning and mobility planners, as well as public real estate departments. This makes it challenging to develop clear incentives, regulations and standards (ALICE-ETP, 2020; POLIS & TDA, 2020). Furthermore, this organizational complexity and variations in markets and governance systems across cities makes it crucial to assess the possibilities to allocate responsibilities and investments among the stakeholders to implement DISCOBAY. Initial investments in this area can be high (ALICE-ETP, 2020; POLIS, 2023). For example, the organization of the cooperative or white-label hub involves assessing the potentials for reaching the desired policy goals and dealing with various dilemmas and design choices (Prandtstetter et al., 2019; Pufahl et al., 2020; S. M. Russo et al., 2021). Therefore, *Step 2* in the approach deals with setting up the organizational side of DISCOBAY, involving the supply side (hub/crossdock function as well as storage), the demand side (getting urban businesses and logistics service providers to use the hub) and determining SMART goals for the operation of the hub that can be evaluated and monitored.

Cities use heterogeneous information systems and data structures for planning, managing, monitoring and regulating urban logistics. At the same time, digital platforms are an increasingly important instrument in the modal shift (Jain et al., 2021; Pourrahmani & Jaller, 2021). Thus, the availability and interoperability of data are a great challenge (see DISCOLLECTION for pertinent data-platform solutions). Specifically, in DISCOBAY, sharing of information regarding the (available) capacity in the multimodal hub and the associated warehouses, as well as the demand for urban deliveries via logistics service providers, is crucial. Using CO2 as an effective KPI for sustainable logistics, for example, also depends on available data (Dobers et al., 2023). Unwillingness to share company data and lack of platforms limit the functionality and real-time responsiveness of the DISCOBAY solution, as well as the analytical capabilities necessary to plan, simulate and evaluate the effects (ALICE-ETP, 2020; de Bok et al., 2022). This is why *Step 3* in the approach aims to select measurable KPIs for the chosen SMART goals of DISCOBAY, defined in Step 2 (which in turn reflect the policy goals of Step 1), checking of available data sources in the city and performing a baseline measurement of the KPIs.

6.1. Introduction to the DISCOBAY tools

The implementation of DISCOBAY in the Starring Living Labs was supported by a set of analytical and operational tools designed to either plan for or directly manage multimodal freight flows, aligning with the three-step DISCOBAY approach. The tools and their application varied depending on the local context and the specific focus of each Living Lab.

In Copenhagen, the DISCOBAY measure adopted a "test before invest" strategic planning approach (Step 1). The primary enabling tool was the Digital Twin for Urban Logistics. The objective was to use this analytical platform to simulate and measure the effects (such as energy demand and charging time) of a potential physical retrofit at a peri-urban hub in the Høje Taustrup Municipality—specifically, the installation of new charging infrastructure for an EV fleet.

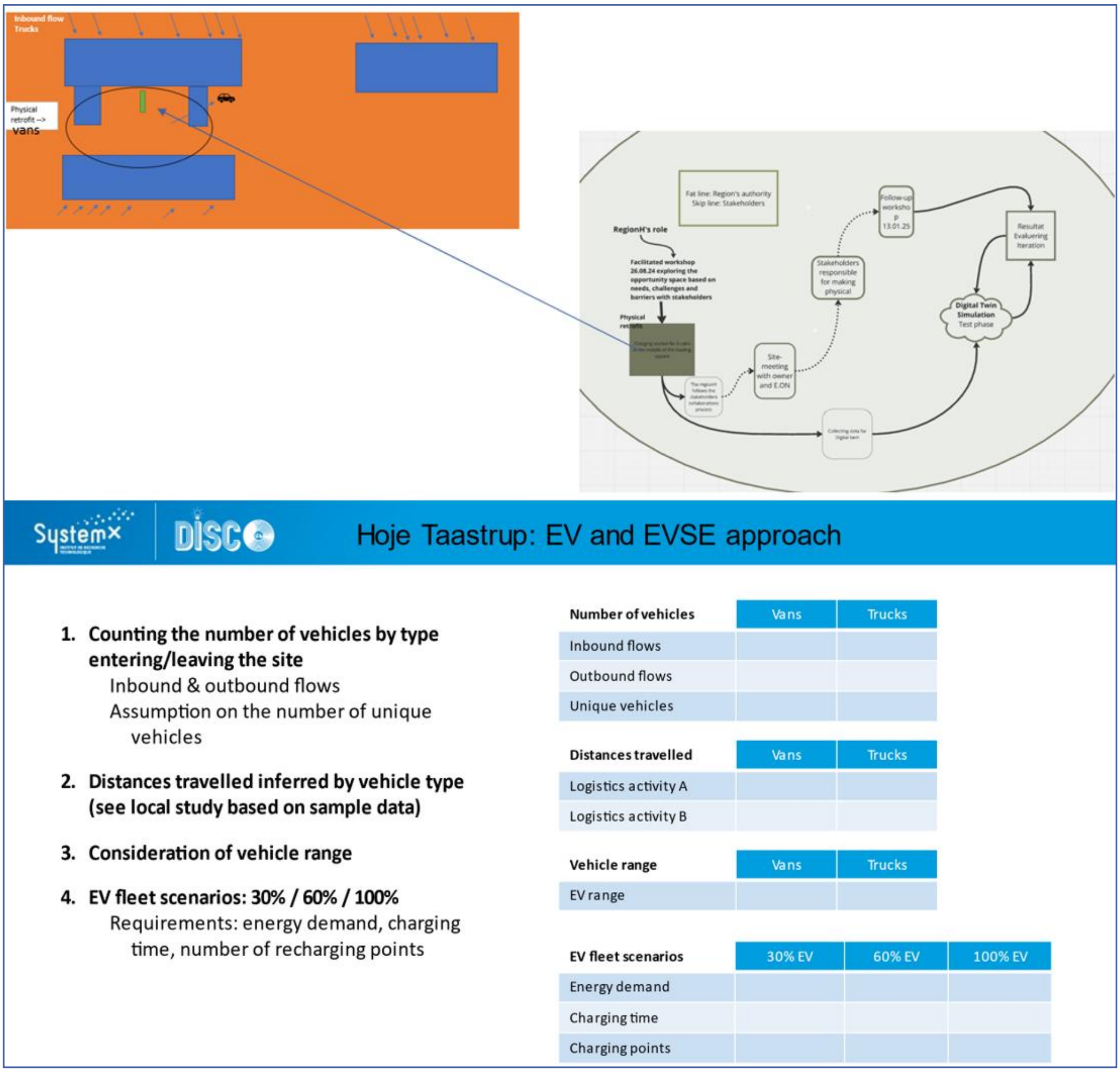


Figure 19: The Høje Taustrup approach of Copenhagen Living Lab as part of the DISCOBAY measure

The simulation required a multi-stakeholder collaboration involving the Høje Taastrup Municipality, a charging infrastructure company, a wholesaler, and the building's owners and operators. The Digital Twin was fed with several key datasets to ensure the accuracy of the model, including traffic counts from private consultancies, qualitative data from stakeholder workshops, and technical data on the electrical grid's capacity from the local utility provider. This data was also cross-referenced with public traffic data from the Danish national mobility data space. The tool's core function was to use these inputs to map and forecast various EV fleet scenarios, providing data-driven evidence on the required energy demand and the potential usage of the new charging infrastructure before any physical investment was made.

In Ghent, the DISCOBAY measure was highly operational, focusing on the implementation of waterway logistics combined with Green Last Mile (GLM) operators in a multimodal delivery network. The ambition was to achieve a modal shift from road to waterway for urban deliveries, with a particular focus on construction materials and parcels. This involved a multi-stakeholder collaboration with decentralized control, which is a key principle of the Physical Internet in city logistics. The primary enabling tool for orchestrating these flows was the Urban Access Control system.

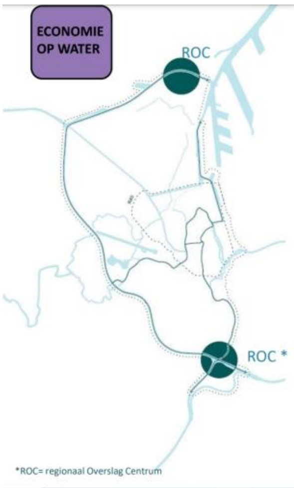


Figure 20: The Ghent DISCOBAY measure concept

It was enhanced to function as a multimodal recommendation engine, capable of suggesting the barge-to-bike service as a viable "Green Last Mile" alternative to traditional truck deliveries. The implementation process involved defining specific loading and unloading locations along the city's waterways and developing a network of stakeholders and collaboration agreements. The concept was validated through a real-life demonstration during the project's second Open Day, where a delivery was successfully executed using an electric barge for the middle-mile and cargo bikes for the final meter, proving the operational feasibility of the multimodal chain.

In Thessaloniki, the approach combined strategic planning with data infrastructure development. The key tools were a Logistics Real Estate Database and a local microsimulation platform. The database was used to map and catalogue underutilized transport infrastructure, specifically identifying assets at the freight rail terminal and the port.

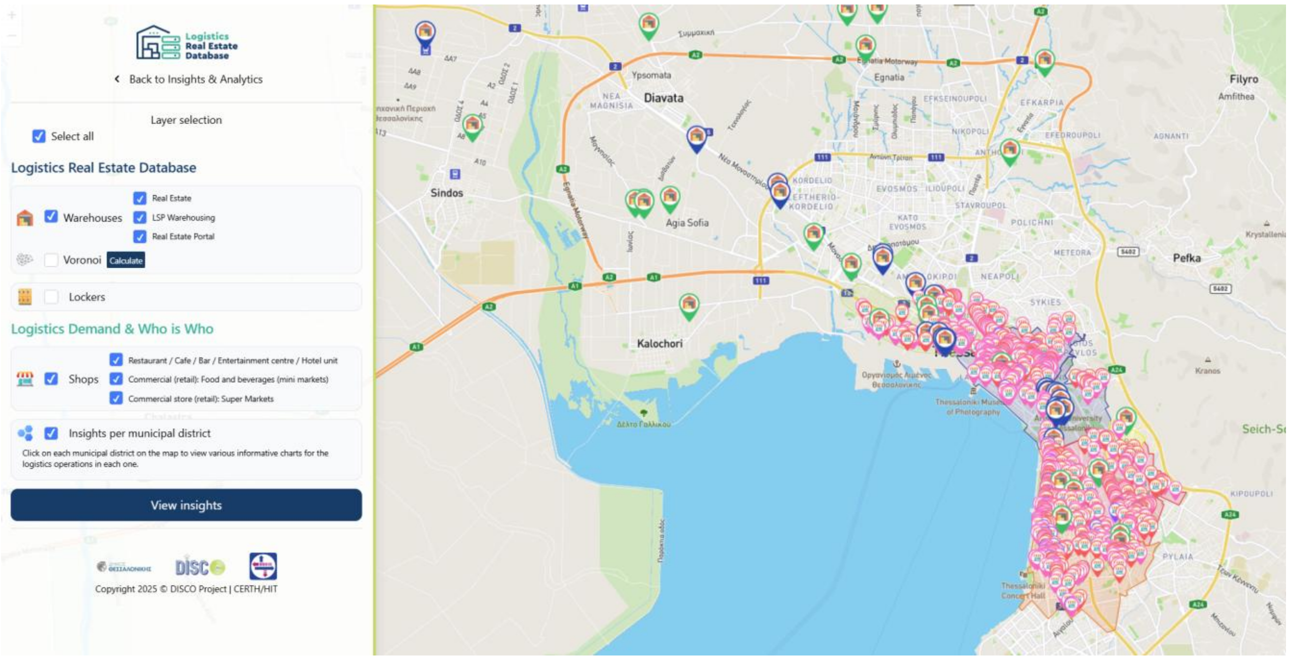


Figure 21: The Logistics Real Estate Database tool of Thessaloniki for DISCOBAY

A locally built microsimulation platform was then used to model the potential impact of activating these sites as urban consolidation centers, providing quantitative evidence on the potential reduction in vehicle kilometers and fleet size. This data served as the baseline measurement and strategic justification for future investment in multimodal hubs.

6.2. Analysis of DISCOBAY PI-Enabling Enhancements

Step 1: Foundational Analysis and PI Requirement Definition

The foundational requirement for DISCOBAY, as established in the project's initial analysis, was to connect the urban logistics ecosystem to the sustainable transport networks outside the city, stimulating a modal shift away from road transport for the "middle mile". The goal was to create a data-driven process for identifying and utilizing underused multimodal nodes (like rail and barge terminals) as consolidation points for green last-mile delivery, a core principle of the Physical Internet.

Step 2: Solution Identification and Baseline Alignment with PI Roadmap

Three main categories of existing tools were identified to meet these requirements:

- **Strategic Simulation Tools (Copenhagen):** The **Digital Twin platform** existed as an advanced tool for modelling complex logistics networks. In its "as-is" state, it conceptually aligned with the strategic planning of **"Logistics Networks"**.

- **Operational Route Management Platforms (Ghent):** The **Urban Access Control (UAC) system** served as the baseline. In its "as-is" state, it was a compliance tool for road-based transport, aligning with the operational management aspect of the **"Logistics Networks"** building block.

- **Analytical Planning Tools (Thessaloniki):** The baseline tools were a standard **GIS database** for mapping urban assets and a **microsimulation platform** for modeling traffic flows. In their "as-is" state, these tools aligned with the strategic planning aspects of the **"Logistics Nodes"** and **"Logistics Networks"** building blocks of the ALICE PI Roadmap.

Step 3: Analysis of PI Enhancements from Living Lab Implementation

The Living Labs independently made crucial enhancements to align these tools with PI principles:

- In Copenhagen, the **Digital Twin** was enhanced to specifically simulate a "test before invest" scenario for a physical retrofit. It was adapted to model the energy demand and charging time effects of installing new EV charging infrastructure at the Høje Taustrup peri-urban hub, requiring the integration of new datasets like electrical grid capacity and detailed traffic counts.

- In Ghent, the **UAC system** was enhanced with a **multimodal recommendation feature**. This involved adapting the system to ingest data on the availability of the waterway barge service and, when an LSP submitted a truck route, to proactively analyze if a barge-to-bike alternative was a viable option and suggest it to the planner.

- In Thessaloniki, the **analytical tools** were enhanced to focus specifically on *multimodal* opportunities. The **Logistics Real Estate Database** was expanded to map underutilized transport infrastructure, including assets from the national rail operator (GaiaOSE) and the Port of Thessaloniki. The **microsimulation platform** was then enhanced with a new scenario to model the impact of consolidating freight at these peri-urban terminals and using a smaller fleet of EVs for last-mile delivery.

Step 4: Validation through Living Lab Results

The empirical evidence from the pilots validated the success of these enhancements:

- The enhanced Digital Twin in **Copenhagen** was validated by its ability to provide data-driven forecasts on energy demand for various EV fleet scenarios. This provided the multi-stakeholder partnership with the evidence needed to make an informed investment decision regarding the charging infrastructure, directly supporting the city's SULP.

- The UAC's multimodal recommendation engine in **Ghent** was validated during the project's second Open Day, where a live barge-to-bike delivery was successfully executed and coordinated using the tool, proving its operational feasibility for facilitating a real-time modal shift.

- The enhanced analytical tools in **Thessaloniki** were validated by the quantitative evidence they produced. The microsimulation demonstrated that activating the identified multimodal sites as urban consolidation centers could lead to a significant reduction in vehicle kilometers and fleet size, providing a strong, data-driven justification for future investment.

Step 5: Synthesis and Identification of PI-Enabling Components

The analysis of these validated enhancements reveals several key PI-enabling components within the DISCOBAY toolkit:

- **"Test Before Invest" Simulation Capability:** The ability of a Digital Twin to model and de-risk major physical and policy changes is a key component for the strategic planning of PI **"Logistics Networks"** and **"Logistics Nodes"**.

- **Dynamic Multimodal Orchestration:** The UAC system's capacity to proactively suggest and facilitate a real-time modal shift based on live conditions is a core component that enables the dynamic orchestration of a PI **"Logistics Network"**.

- **Multimodal Asset Identification:** The capability of the enhanced database to identify and catalogue underutilized, strategically located multimodal assets is a foundational component for expanding the PI's network of **"Logistics Nodes"**.

7. Smart Data Integration Tools (DISCOLLECTION)

The Physical Internet relies on transparency and seamless collaboration between stakeholders in urban logistics networks. This collaboration requires structured and standardised data sharing, which functions as the common language enabling coordination across cities, logistics service providers, retailers, and transport operators. DISCOLLECTION addresses this requirement by providing smart data integration tools that enable cities and other stakeholders to connect their data to the Urban Freight Data Space (UFDS) in structured and standardised formats.

7.1. Introduction to the DISCOLLECTION tools

7.1.1. The DISCOLLECTION Smart Data Platform

The platform eliminates the need for costly data space connectors that create participation barriers by offering a cost-effective Smart Data Platform (SDP). DISCOLLECTION transforms fragmented, unstandardised datasets into internationally recognised formats such as APDS (1SO TS 5206-1) for parking data and DATEX II for urban access regulations. This standardisation creates the data infrastructure necessary for Physical Internet operations by establishing a shared vocabulary that enables coordination, optimisation, and sustainability improvements in urban freight operations, supporting the project's objective of data-driven, collaborative urban logistics planning.

Many cities possess valuable logistics-related data but face significant challenges in making it data space-ready. Critical datasets often remain unpublished in internal databases or archives, while others provide only real-time measurements without the historical context necessary for analytics and planning. Much of this data exists in unstandardised formats that cannot be directly shared with external stakeholders, and valuable information is frequently fragmented across multiple sources and systems, requiring complex integration efforts before it can be effectively utilised.

The Smart Data Platform, which is the main tool of the DISCOLLECTION solution, addresses these challenges by serving as a comprehensive, packaged solution that transforms cities into active data space participants. Rather than requiring expensive custom development or technical expertise, the SDP provides cities with everything needed to prepare, structure, and share their data through a single, easy-to-deploy package. The platform's value proposition lies in dramatically reducing the technical barriers and costs associated with data space participation while ensuring compliance with international standards and data sovereignty requirements.

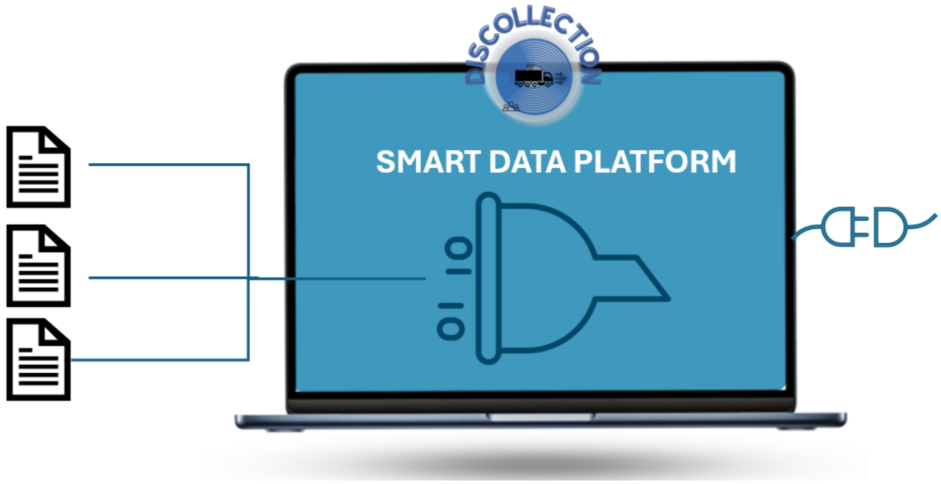


Figure 22: Conceptual image of the Smart Data Platform

The SDP delivers a complete suite of capabilities specifically designed for municipal data challenges. Its ETL pipeline and scheduler automate data extraction, transformation, and loading processes from multiple sources while building historical datasets from real-time feeds. Built-in transformation and standardisation tools convert local datasets into DISCO-compliant formats, including APDS for parking data and DATEX II for urban access regulations. The platform includes TimescaleDB for historical data storage and analytics, a REST API for seamless data publishing, and most critically, an integrated DISCO Urban Freight Data Space Connector that eliminates the need for separate connector deployment while ensuring secure, policy-controlled data sharing with logistics stakeholders.

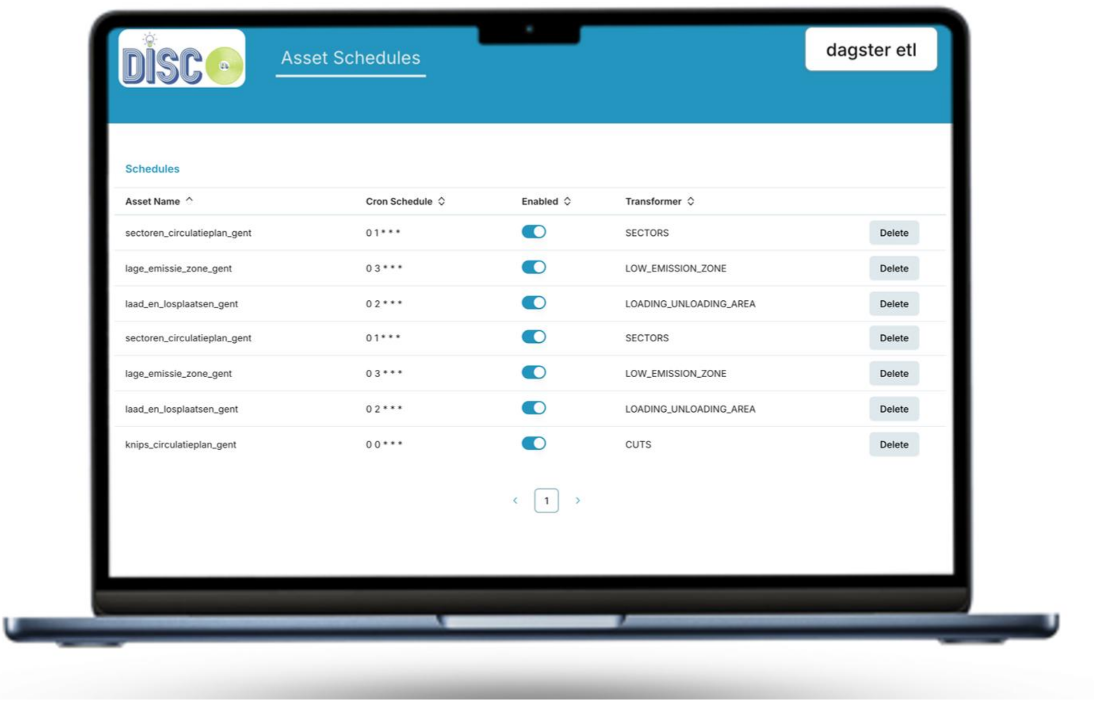


Figure 23: Interface of the Smart Data Platform, displaying the integrated data sets

Automated Data Processing: The ETL Pipeline and Scheduler

The DISCOLLECTION platform employs a Dagster-based ETL (Extract, Transform, Load) pipeline that automates the complex process of extracting data from diverse city sources, including REST APIs, open data portals, and internal databases. The Dagster orchestration engine manages these operations through configurable schedules, such as hourly extraction for dynamic occupancy data and daily updates for static datasets, ensuring that data flows remain current while building comprehensive historical records essential for logistics planning and analysis.

This automated data integration process represents an important step in creating the standardised "digital containers" of information necessary for a fluid, Physical Internet-based logistics system. Just as physical containers enabled the standardisation of global shipping, these digital containers, structured according to international data standards and formats, provide the common format required for seamless data exchange between cities, logistics providers, and other stakeholders. The pipeline's ability to continuously transform raw municipal data into these standardised formats ensures that cities can contribute consistent, interoperable datasets to the UFDS, enabling the real-time coordination and optimisation that defines the PI vision.

Technical implementation details of the Dagster architecture and pipeline configurations are provided in the DISCOLLECTION Design and Architecture Documentation.

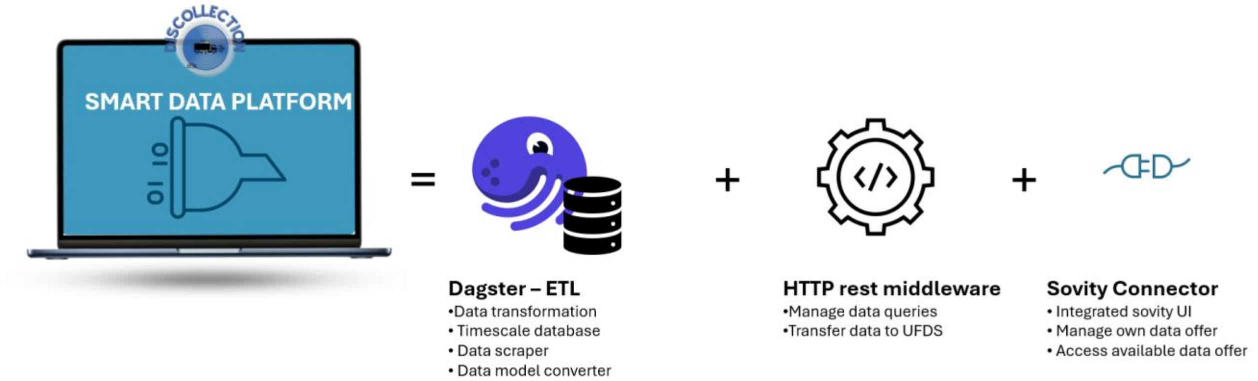


Figure 24: The components of the Smart Data Platform

Data Transformation and Standardization

The transformation of raw city data into standardized formats represents the critical process of establishing the "common language" essential for PI operations. DISCOLLECTION implements data mapping procedures that convert municipal datasets into internationally recognized standards, ensuring interoperability across diverse urban logistics networks. The Ghent Living Lab provides concrete examples of this transformation process: the city's real-time parking occupancy data, originally containing fields such as "`totalcapacity`," "`availablecapacity`," and "`operatorinformation`," is systematically mapped to the APDS (Alliance for Parking Data Standards) format, creating structured Place objects with standardized `hierarchyElementReference`, `supply`, and `demandTable` components that enable seamless integration with parking applications across different cities and platforms.

Similarly, Ghent's urban access regulation datasets undergo transformation into DATEX II 3.6 standards, with Low Emission Zone polygons mapped to `ControlledZoneTablePublication` structures and circulation plan restrictions converted to `TrafficRegulationPublication` formats. For example, the city's Low Emission Zone open data set, originally containing simple geometry coordinates and basic identifiers like "`gentid`" and "`urid`," is transformed into comprehensive DATEX II XML structures that include standardized elements such as `UvarZone` definitions, `controlledZoneType` classifications, and `trafficRegulationOrder` specifications with proper geographic referencing through `GmlMultiPolygon` coordinates.

This standardization process, detailed extensively in the DISCO Data Model Conversion Documentation, ensures that diverse municipal datasets can be universally understood and utilized by any Physical Internet actor, regardless of their local data formats or technical infrastructure, thereby enabling the seamless data exchange and coordination that underpins collaborative urban logistics optimization.

Secure Data Storage and API Access

DISCOLLECTION employs a robust storage architecture designed to handle both structured time-series data and diverse data assets required for urban logistics applications. TimescaleDB serves as the primary database for storing transformed datasets, particularly optimized for time-series data that is crucial for logistics planning optimization. This specialized database efficiently manages historical parking occupancy records, traffic flow patterns, and other temporal datasets while supporting complex queries across time ranges, which are essential capabilities for analytics and predictive modeling in PI applications.

Complementing this structured storage, Minio Object Store provides scalable storage for unstructured and bulk data, including original raw datasets, transformed files, and metadata, ensuring comprehensive data preservation and accessibility.

The platform's REST API, built using the FastAPI framework, serves as the secure gateway for publishing processed data to the Urban Freight Data Space. This API follows OpenAPI standards and provides comprehensive endpoint documentation through an integrated Swagger UI, enabling standardized access to transformed datasets while maintaining strict security controls. The API supports dynamic query parameters that allow consumers to filter data by various criteria, such as location, time ranges, or specific asset characteristics, while ensuring that data access remains governed by the usage policies defined within the data space connector.

As detailed in the DISCOLLECTION Design and Architecture Documentation, the API operates within a private network architecture, accessible only through the integrated Sovity Data Space Connector, ensuring that all data exchanges are authenticated, authorized, and compliant with the data sovereignty principles. This architecture guarantees that while cities can seamlessly publish their standardized data assets, they maintain complete control over access permissions and usage policies.

7.1.2. The Integrated Data Space Connector

The integrated data space connector represents the gateway that enables cities to participate as active nodes in the Physical Internet network, transforming isolated municipal data silos into interconnected components of a collaborative logistics ecosystem. Unlike traditional approaches that require cities to implement or deploy costly and complex standalone connectors, the DISCOLLECTION platform embeds the Sovity-based UFDS connector directly within the Smart Data Platform architecture, eliminating technical barriers while ensuring seamless integration.

This embedded connector’s architecture implements the essential principles of data sovereignty and decentralized data exchange that underpin PI operations. The connector operates through secure OAuth2-based authentication protocols, utilising public-private key pairs and JWT tokens to establish trusted connections between participating nodes while maintaining strict control over data access and usage policies.

Cities thus retain complete ownership and control over their data assets, defining precisely which datasets can be accessed, by whom, and under what conditions, ensuring that participation in the PI network enhances rather than compromises municipal data governance.

From a technical perspective, the connector provides comprehensive functionality through its management API, enabling cities to register data assets, establish contract definitions, manage policy frameworks, and monitor transfer processes. The connectors support both real-time catalog discovery, allowing other Physical Internet participants to identify available city datasets and dynamic data transfer capabilities that facilitate secure, policy–controlled data exchange.

7.1.3. The Smart Routing and Access Control Engine (use case)

The standardized, accessible data generated through DISCOLLECTION's transformation processes serves as the essential foundation for intelligent routing algorithms and dynamic access management platforms that define modern Physical Internet operations. Urban Access rules transformed into DATEX II format, including Low Emission Zone boundaries, circulation plan restrictions, and loading/unloading point regulations, provide routing systems with machine-readable, semantically consistent information necessary for real-time compliance verification and route optimization. Similarly, parking and loading zone data standardized according to APDS specifications enables logistics applications to dynamically assess space availability, duration restrictions and access permissions, creating the data infrastructure required for efficient last-mile delivery coordination.

The standardized data architecture proved particularly valuable in sophisticated urban access control systems such as the Urban Access Control platform deployed in Ghent, as shown in the figure below. The UAC system could leverage DISCOLLECTION's DATEX II-formatted circulation restrictions and APDS-compliant parking data to make real-time planning decisions, automatically validating planning requests against current urban regulations while optimising space allocation for logistics operations. The platform can process complex scenarios where delivery vehicles must navigate multiple regulatory layers, from time-restricted loading zones to emission-based access controls, because all relevant data exists in standardised, interoperable formats that enable seamless integration across different municipal systems and private logistics applications.

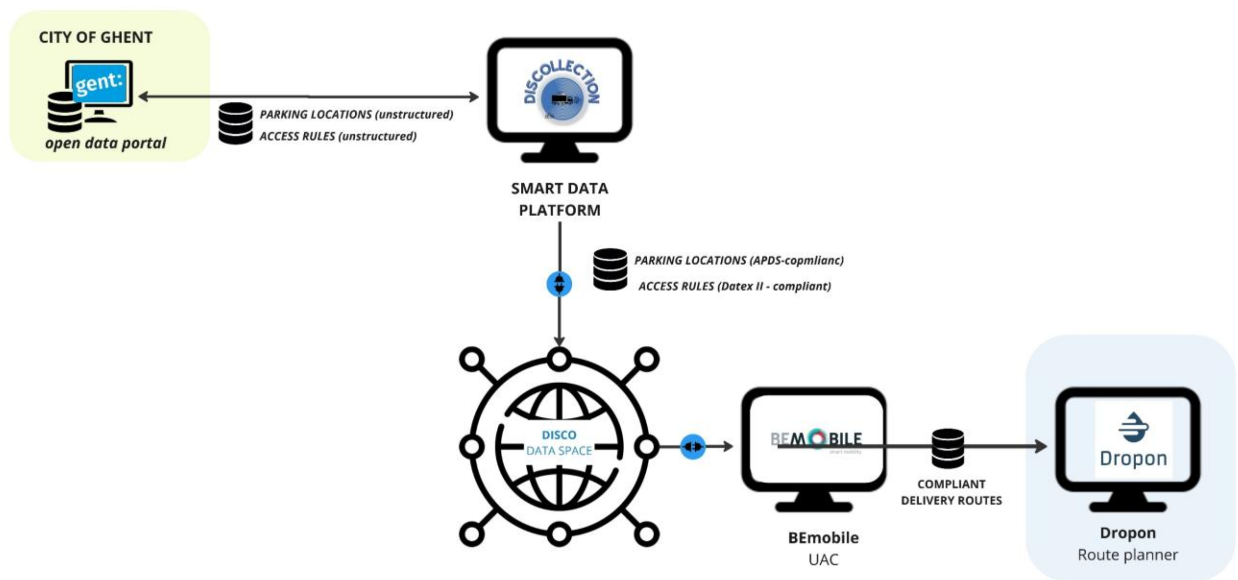


Figure 25: Diagram of the Discollection Ghent Living Lab, with the UAC

The broader implications extend beyond individual city implementations to support the Physical Internet's vision of interconnected urban logistics networks. When cities across different regions publish their access rules and parking data using consistent DATEX II and APDS standards, logistics providers can develop unified routing and planning systems that operate seamlessly across municipal boundaries.

The DISCOLLECTION Ghent Living Lab setup was extended to two additional cities and an additional booking tool, demonstrating the scalability and impact of the DISCOLLECT solution. In this extension, BeMobile took up the role of the intermediary data provider, structuring and standardizing the cities' data sets. More information on the Ghent UAC use case can be found in D4.2 Starring LL measures implementations (L'Istituto sui Trasporti e la Logistica, 2024) and D4.3 Evaluation of impacts at Starring LL's (LSP, 2025).

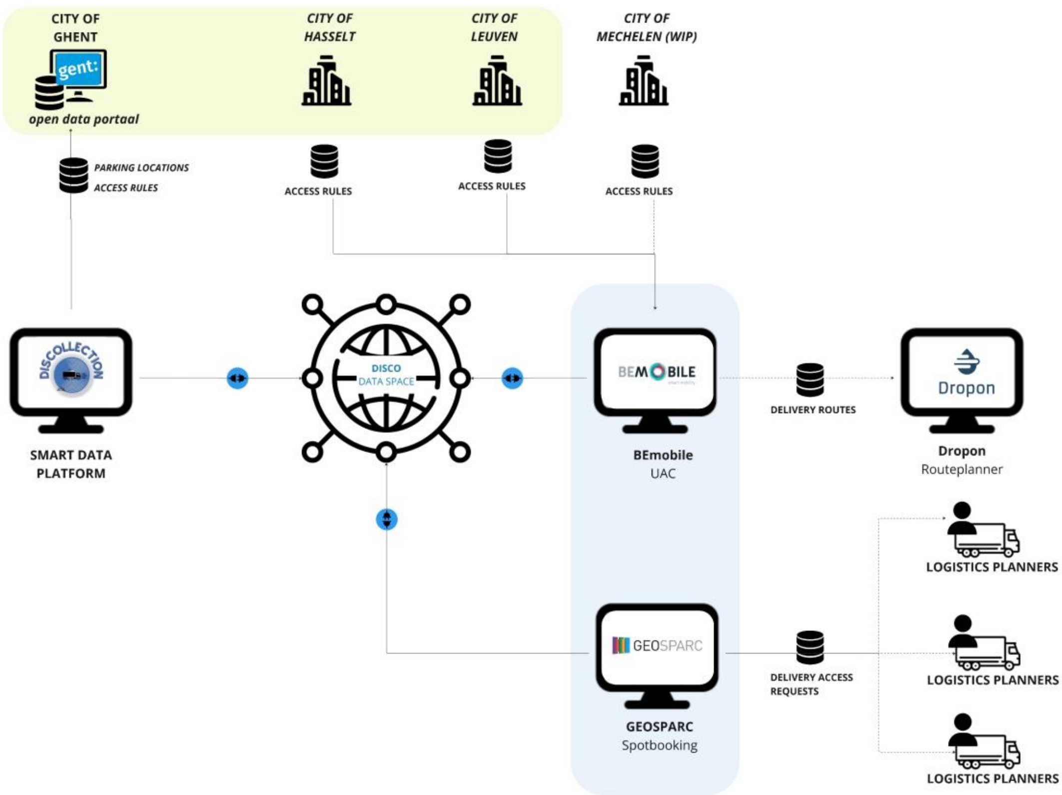


Figure 26: diagram of the extended DISCOLLECTION Ghent Living Lab setup

This standardization enables the development of pan-European logistics applications that can automatically adapt to local regulations, optimize multi-city delivery routes, and coordinate with municipal access management systems, transforming fragmented urban logistics into the integrated, efficient network that defines the Physical Internet paradigm.

7.1.4. The Digital Twin for Urban Logistics in Copenhagen

In the Copenhagen Living Lab, the primary enabling tool for DISCOLLECTION was the Digital Twin for Urban Logistics. The Digital twin that was built on the LIAISON system by the DISCO partner IRTX was designed to measure the city-wide effects of potential policies and infrastructure interventions before they are implemented.

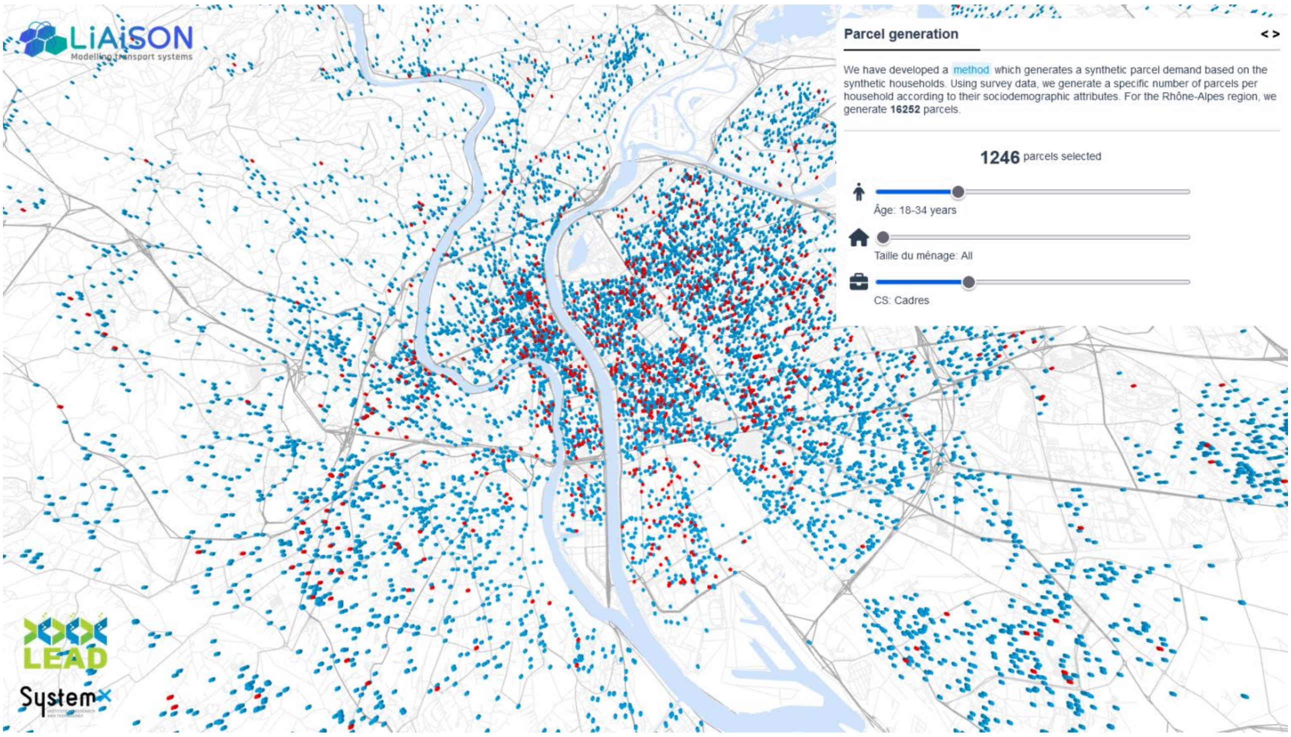


Figure 27: The LiAison modelling platform where the Digital Twin for Copenhagen was developed

The core of the DISCOLLECTION measure was to develop a dynamic, real-time data collection model by establishing data-sharing agreements with private logistics companies.

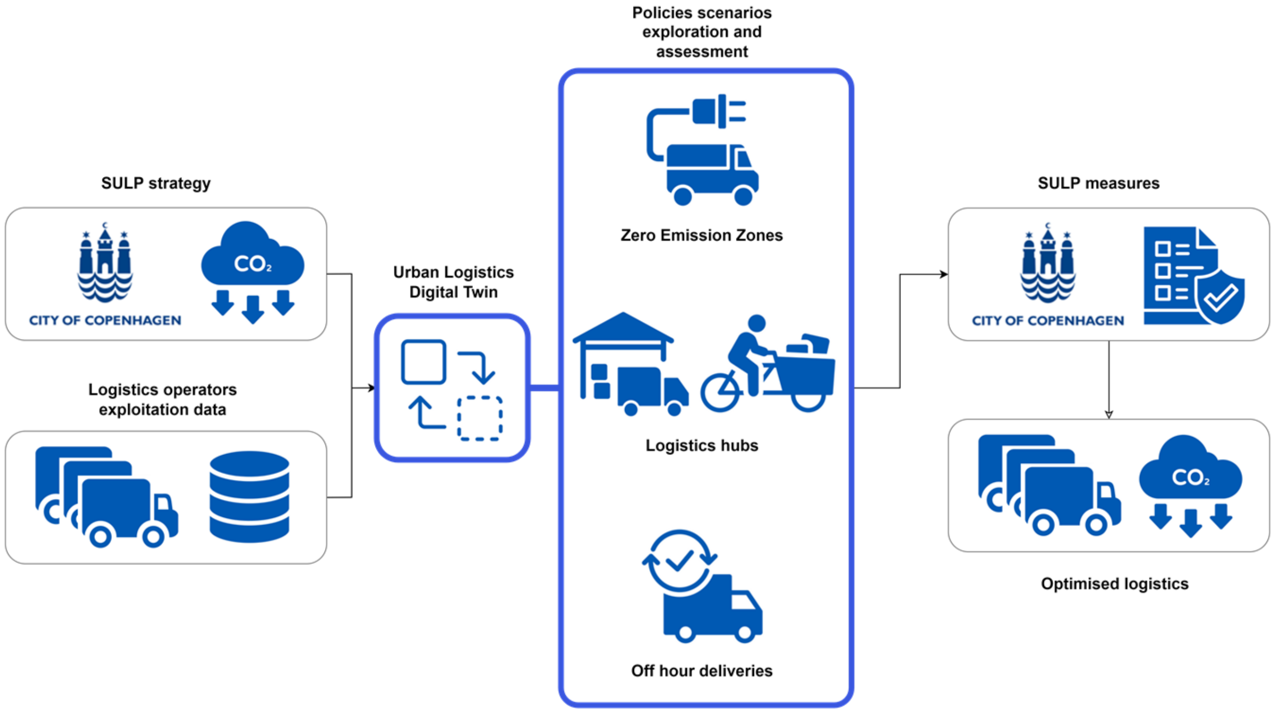
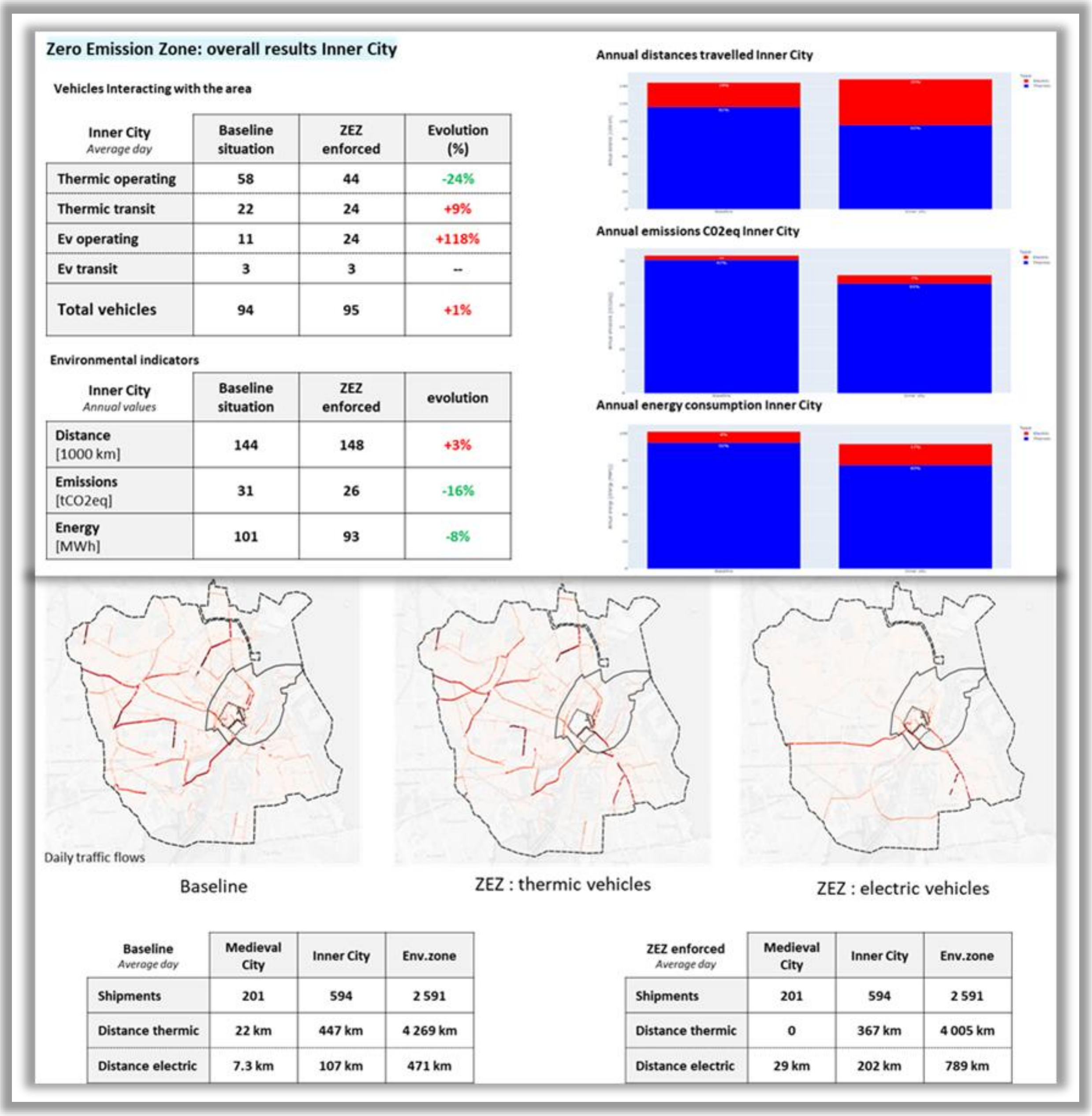


Figure 28: The Copenhagen Data Collection Model, from LSP Partnership Agreements to SULP Scenario Simulation

A key achievement was the successful negotiation of 13 data-sharing partnership agreements, which allowed for the collection of anonymized, real-world route data from 11 different LSPs. This data was then fed into the Digital Twin to run a series of "what-if" simulations. For example, the tool was used to model the impact of a Zero-Emission Zone (ZEZ) in the city center, predicting a potential 83% reduction in CO_2 emissions for the affected routes.



Inner City *Average day*	Baseline situation	ZEZ enforced	Evolution (%)
Thermic operating	58	44	-24%
Thermic transit	22	24	+9%
Ev operating	11	24	+118%
Ev transit	3	3	--
Total vehicles	94	95	+1%

Inner City *Annual values*	Baseline situation	ZEZ enforced	evolution
Distance [1000 km]	144	148	+3%
Emissions [tCO2eq]	31	26	-16%
Energy [MWh]	101	93	-8%

Baseline *Average day*	Medieval City	Inner City	Env.zone
Shipments	201	594	2 591
Distance thermic	22 km	447 km	4 269 km
Distance electric	7.3 km	107 km	471 km

ZEZ enforced *Average day*	Medieval City	Inner City	Env.zone
Shipments	201	594	2 591
Distance thermic	0	367 km	4 005 km
Distance electric	29 km	202 km	789 km

Figure 29: The dashboard of the Copenhagen Digital Twin, showing the simulation of a Zero-Emission Zone (ZEZ) scenario

The Digital Twin's value was in its ability to provide data-driven evidence for policymaking, directly informing the development of Copenhagen's new Sustainable Urban Logistics Plan.

7.1.5. The local Urban Logistics Dataspace as a Local Observatory in Thessaloniki

In the Thessaloniki Living Lab, the DISCOLLECTION measure was focused on creating a proof-of-concept local Logistics Dataspace, which now functions as a public online observatory (uldthess.imet.gr).

Figure 30: The welcome screen of Thessaloniki's local logistics dataspace

The platform was designed to act as a central hub for collecting, integrating, and visualising a wide range of data related to the city's logistics ecosystem, as shown on Figure 31 and Figure 33.

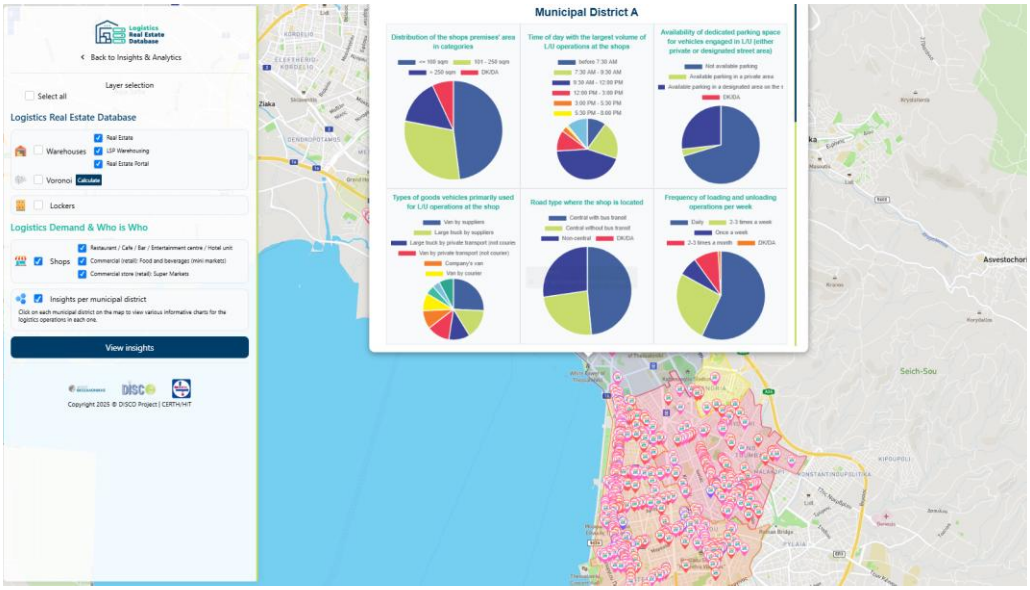


Figure 31: Urban Logistics Activity insights per Municipal District

The local urban logistics dataspace platform successfully integrated data from all the other local DISCO-X measures via APIs, including the WareM&O platform (DISCOESTATE) and the Logistics Real Estate Database (DISCOPROXI and DISCOBAY).

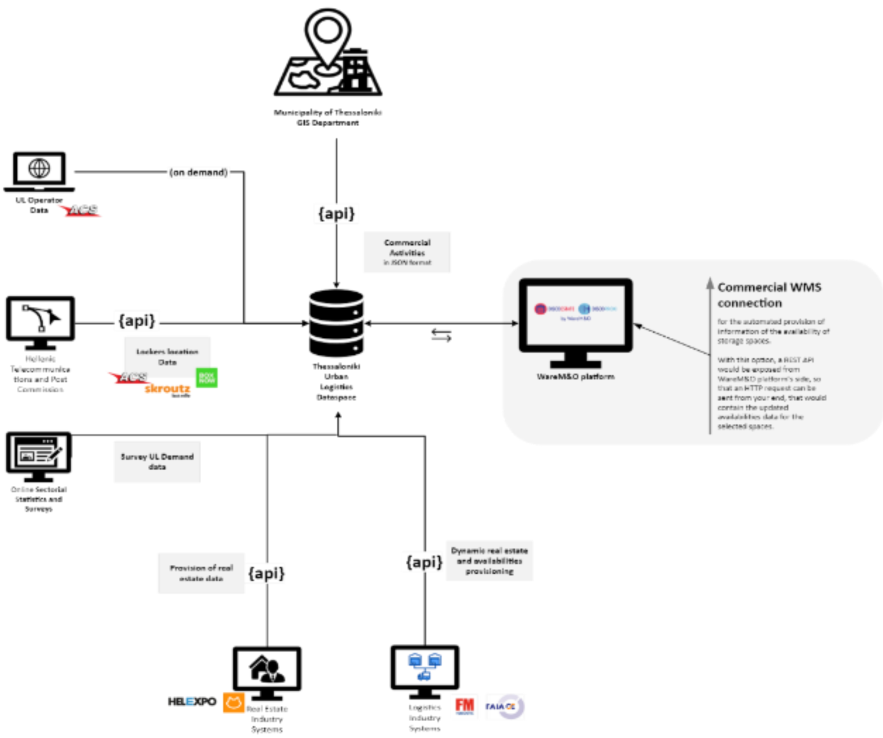


Figure 32: A high-level Architecture of the Urban Logistics dataspace of Thessaloniki

It also incorporated data from extensive stakeholder surveys, including input from over 1,000 local retailers. The platform features interactive dashboards and a "Who is Who" registry of local logistics actors and a “City in Numbers” interface (Figure 33), providing a powerful, evidence-based tool for municipal planners.

Figure 33: The "City in Numbers" section tab of the local urban logistics dataspace of Thessaloniki

It represents a foundational step in building the data-driven governance capacity required to orchestrate a fully interconnected, PI-led urban logistics ecosystem.

7.1.6. The Data Pipeline to the Helsinki Mobility Data Catalog (Helsinki)

In the Helsinki Living Lab, the DISCOLLECTION measure focused on creating a practical data pipeline to make the real-time data from the DISCOCURB pilot openly available. The loading zone monitoring sensors (from Ramboll, Technolution, and Flow Analytics) were required to share the collected data as a data dump. The raw data was then processed and added as a new, openly shareable dataset to the Helsinki Mobility Data Catalog, which is the city's central repository for mobility-related data. A version of this dataset was also added to the DISCO project's Urban Freight Dataspace. This implementation demonstrated a successful end-to-end data flow, from the initial collection by on-site sensors to its final publication in a city-wide, open data catalog, ensuring the data's long-term value for planners and researchers beyond the life of the project.

7.2. Analysis of DISCOLLECTION PI-Enabling Enhancements

Step 1: Foundational Analysis and PI Requirement Definition

The foundational requirement for DISCOLLECTION was to create the data backbone for a PI-led ecosystem. As established in the project's initial analysis, the goal was to provide tools and methods for smart data collection, standardization, and sharing. This was essential to enable transparency, seamless collaboration between stakeholders, and the data-driven optimization of logistics operations, such as real-time routing and dynamic access control.

Step 2: Solution Identification and Baseline Alignment with PI Roadmap

Four main categories of existing tools or concepts were identified to meet these requirements:

- **Data Integration Platforms (Ghent):** The starting point was the concept of an ETL (Extract, Transform, Load) pipeline. This "as-is" concept aligns with the **"System of Logistics Networks"** building block of the ALICE PI Roadmap, as it provides the mechanism for data harmonization.

- **Strategic Simulation Tools (Copenhagen):** The **Digital Twin platform (LIAISON)** existed as an advanced tool for modelling complex logistics networks, aligning with the strategic planning of **"Logistics Networks"**.

- **Data Observatories (Thessaloniki):** The baseline was the concept of a web-based data portal. This aligns with the **"System of Logistics Networks"** building block by providing a centralized point of access for shared data.

- **Data Archiving Processes (Helsinki):** The "as-is" process involved standard data management practices. This aligns with the foundational data management aspect of the **"System of Logistics Networks"**.

Step 3: Analysis of PI Enhancements from Living Lab Implementation

The Living Labs independently made crucial enhancements to align these tools with PI principles:

- In Ghent, the baseline ETL concept was developed into a full **Smart Data Platform (SDP)**. The key enhancement was the integration of transformation modules to convert raw city data into PI-compliant standards (**DATEX II and APDS**) and the embedding of a **UFDS connector**, creating a packaged, easy-to-deploy solution for cities to join a federated data ecosystem.

- In Copenhagen, the **Digital Twin** was enhanced by **integrating real-world, multi-stakeholder route data** from 13 logistics partners. This transformed a generic simulation platform into a PI-compliant strategic planning instrument for collaborative logistics, capable of modeling the system-wide benefits of data sharing.

- In Thessaloniki, the data portal concept was developed into a proof-of-concept local Urban logistics dataspace, having both a static and a dynamic aspect when it comes to data provisioning. The platform was enhanced with APIs and data ingestion pipelines to connect to multiple sources, including: the **Open Data Portal of Greece**, the **Greek Observatory of Transport and Logistics**, the constantly updated **municipal GIS for logistics infrastructure and policies**, **online real estate registries**, the **WareM&O database**, **rail real estate and warehouse registries**, and data from the Hellenic Regulatory Authority of Telecommunications and Post **(EETT)**.

- In Helsinki, the data archiving process was enhanced by creating a formal **data pipeline**. This involved establishing a process to not only collect the data dumps from the sensor providers but also to process and publish this data as a new, openly shareable dataset in the **Helsinki Mobility Data Catalog**.

Step 4: Validation through Living Lab Results

The empirical evidence from the pilots validated the success of these enhancements:

- The **Smart Data Platform** in Ghent was validated by its successful use in the UAC system, where the standardized DATEX II data it produced was essential for the platform's core functionality.

- The enhanced **Digital Twin** in Copenhagen was validated by its ability to provide powerful, data-driven evidence for the city's new SULP, including a key finding that a shared hub with cargo bikes could reduce CO_2 emissions by 90% for those routes.

- The Thessaloniki UFDS was validated by its successful launch as a public online observatory (uldthess.imet.gr). The key validation was its demonstrated ability to successfully ingest, harmonize, and visualize data from the diverse and previously siloed data sources, including municipal GIS, national open data portals, and private real estate registries.

- The **Helsinki data pipeline** was validated by the successful publication of the loading zone dataset in the official Helsinki Mobility Data Catalogue, confirming the end-to-end data flow from collection to public access.

Step 5: Synthesis and Identification of PI-Enabling Components

The analysis of these validated enhancements reveals several key PI-enabling components within the DISCOLLECTION toolkit:

- **Data Standardization and Harmonization:** The ability of the Smart Data Platform to transform fragmented, proprietary data into internationally recognized standards is a foundational component for creating the common language required by the **"System of Logistics Networks"**.

- **Collaborative Scenario Simulation:** The capacity of the Digital Twin to integrate data from multiple, competing actors to model the system-wide benefits of collaboration is a key component for making the strategic case for the PI and for optimizing **"Logistics Networks"**.

- **Federated Data Sharing Infrastructure:** The UFDS concept, and particularly the Thessaloniki pilot's successful integration of numerous, disparate data sources (including the Open Data Portal of Greece, municipal GIS, online real estate registries, the WareM&O database, rail real estate registries and data from the Hellenic Regulatory Authority) is a core PI-enabling component. It proves the concept of creating a "single source of truth" from a fragmented data landscape, which is essential for enabling an interoperable **"System of Logistics Networks"**.

- **Open Data Pipelines:** The process of making privately collected operational data available in public repositories, as demonstrated in Helsinki, is a key component for transparency and for driving the **"Access and Adoption"** of new, data-driven planning approaches.

- **Establishment of Data Governance Frameworks**: The successful implementation of the Smart Data Platform (in Ghent), the negotiation of data-sharing agreements (in Copenhagen), the integration of diverse sources into the UldThess (in Thessaloniki), and the creation of open data pipelines (in Helsinki) collectively validated the establishment of new data governance frameworks. These frameworks define the rules and enable the necessary agreements for sharing and standardizing previously siloed public and private data, a core component of the PI **"Governance"** building block.

8. The Meta-Model Suite and Urban freight dataspace integration

In the preceding chapters, a retrospective analysis identified the PI-enabling components of the DISCO-X tools. The aim of this chapter is to show how these components are composed into a single, functional system that carries a user from data to decision to action through one access point. The Urban Freight Dataspace serves as the federated data foundation; the Meta-Model Suite serves as the neutral planning platform where users work; and the DISCO-X tools provide the operational capabilities that implement Physical Internet-led logistics. Together, these three elements allow a planner or operator to discover harmonised datasets, invoke the right tool, and execute an action without navigating multiple systems.

8.1. Scope and purpose

The scope of the integration is to enable DISCO-X tools with trusted, harmonised, multi-source data while keeping the user workflow coherent and source-agnostic. UFDS, implemented through DISCOLLECTION, operates as a true dataspace: it ingests datasets of the same type from different owners and systems; it aligns them to shared vocabularies and agreed formats; it preserves provenance, versioning and freshness; and it publishes the resulting products through a stable, documented interface. The catalogue that emerges in UFDS is therefore not a loose index of links but a set of interoperable, comparable data products that can be consumed by applications without custom adapters per source.

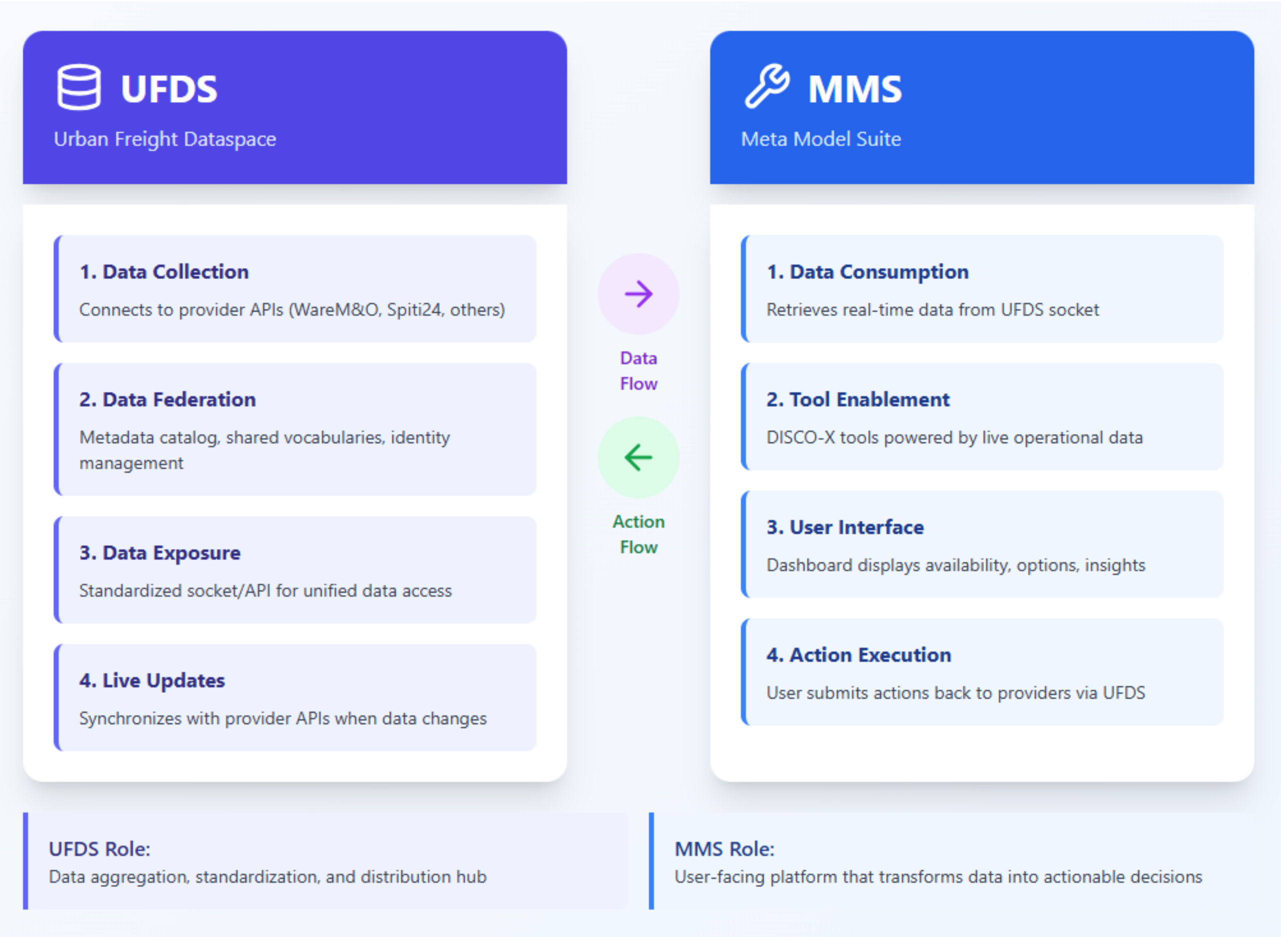


Figure 34: the MMS and UFDS Integration

MMS sits directly above this foundation and acts as the single entry point for users. As a planning platform, MMS discovers the relevant UFDS products, binds them to its internal models and exposes them through a consistent user interface. From the perspective of the user, this means that diverse domains (curb and loading assets, access regulations, micro-hubs and lockers, multimodal options and logistics real estate) appear in a single environment, already reconciled and ready to use. When the user decides to act, MMS activates the appropriate DISCO-X tool. In analytical cases, that may mean running a model or a simulation using UFDS data; in operational cases, it may mean initiating an execution step with an external provider. In both situations, the user remains in MMS while the system retrieves, transforms and routes the necessary information behind the scenes. Figure 34 depicts this relationship: UFDS supplies federated, harmonised data; MMS provides the one-stop planning and orchestration layer; and DISCO-X tools deliver the PI-aligned operational capabilities that turn harmonised data into tangible outcomes.

8.2. Use case

To demonstrate that the integration is not only conceptual but operational, a full proof-of-concept was implemented in Thessaloniki using the Warehouse-as-a-Service tool. The user begins in MMS and opens the “Spaces” module of the DISCO-X library. From this single interface the user specifies an area of interest around the wider city centre, sets dates and capacity needs and, if necessary, adds operational constraints such as 24/7 access, BMS or RFID control and temperature conditions. MMS translates this request into a standard query and submits it to UFDS. The dataspace resolves the query against its harmonised products compiled from contributing providers, including WareM&O, and returns a normalised result set. Because harmonisation has already taken place within UFDS, the results that appear in MMS are directly comparable: units, attributes and terminology are consistent; source and freshness are explicit; provenance is retained.

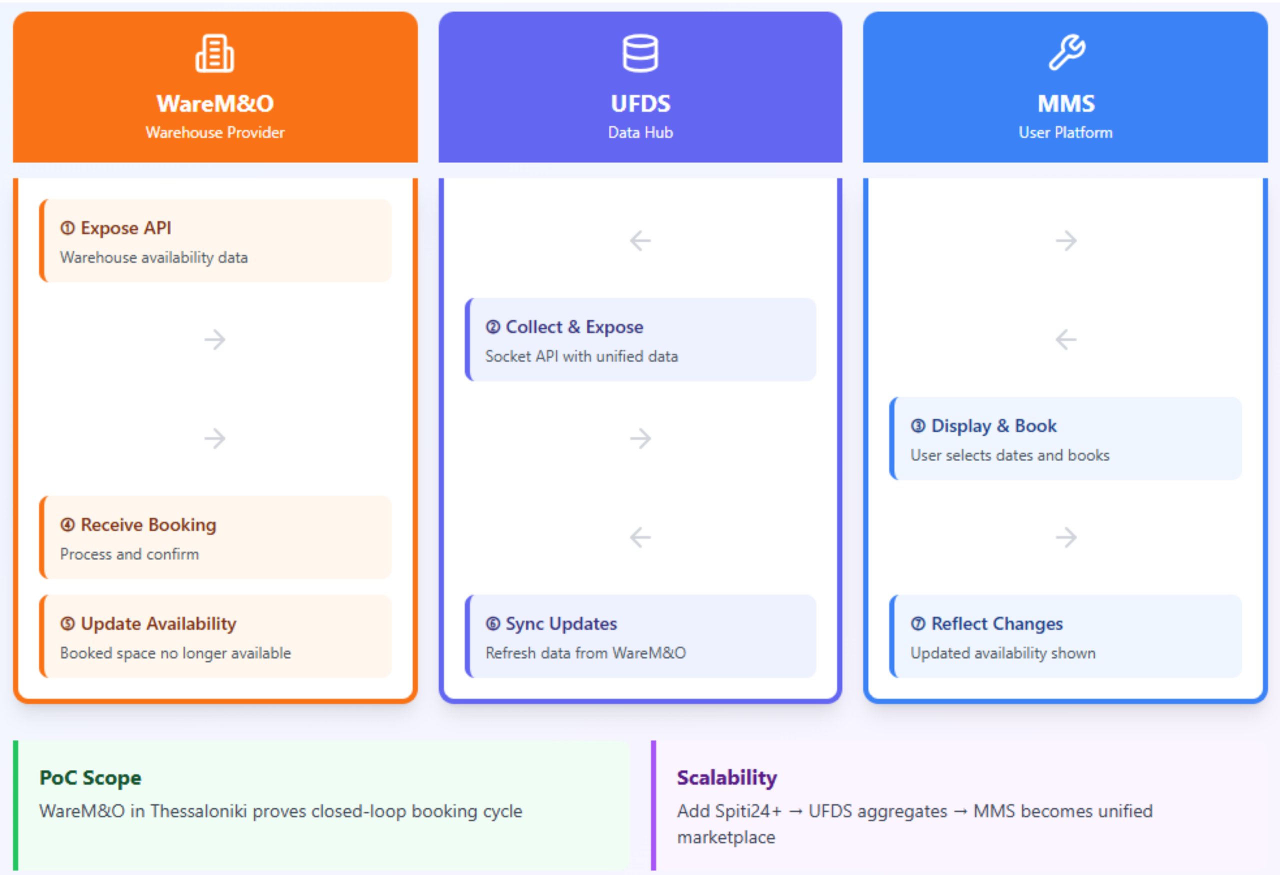


Figure 35: Proof of Concept: Warehouse booking via WareM&O utilizing the DISCO UFDS

The user reviews the catalogue and selects a specific WareM&O listing, for example a pavilion at TIF-HELEXPO. The action is then completed in two tightly coupled steps. First, MMS initiates the transaction using the connection pattern agreed with the provider: where a full API integration is available, MMS posts the booking request programmatically and awaits a provider confirmation; where a redirect remains appropriate, MMS sends the user through a controlled hand-off to the site of the provider using a UFDS deep link that carries the selected parameters and a secure return token. Second, once the provider confirms, an availability update is published back to UFDS so that the catalogue reflects the new state; this update contains no personal or contractual information, only the stock change and the booking reference needed for consistency. MMS, which continuously consumes the UFDS catalogue, refreshes the view and displays the current status without storing sensitive data. Figure 35 illustrates this end-to-end sequence from discovery in MMS, to harmonised data provision via UFDS, to finalisation with the provider and back to state reflection in the unified view. The pattern is general: the same flow can support curbside booking, locker deployment and multimodal hub operations with only domain-specific models changing in the tool invoked by MMS.

8.3. Next Steps

The next phase focuses on deepening interoperability between MMS and UFDS and on broadening the set of DISCO-X tools that can be invoked end-to-end from MMS alone. At the interface level, MMS and UFDS should finalise a stable API contract that covers catalogue discovery, filtered retrieval of harmonised products and execution hooks. This contract should define request and response structures, versioning semantics, authentication and consent, error handling and retry strategies and an event channel for state changes so that MMS can reflect updates without polling. With this API in place, MMS can programmatically initialise tool runs and operational steps using the same data products it used for discovery, and UFDS can broadcast provenance-rich deltas when providers update availability, access rules or other operational characteristics.

In parallel, additional DISCO-X tools should be integrated so that PI-led operations can be enabled from the same one-stop entry point. A priority example is the analytical tool that identifies optimal locations for parcel lockers and micro-hubs based on demand. In this case, MMS would draw on UFDS products combining parcel demand patterns, land-use and accessibility constraints and regulatory layers, run the optimisation within the tool, and then publish the candidate sites back to UFDS for review and collaboration. The same pattern can apply to routing and access-control engines that consume UFDS access rules and curb data and return compliant, lower-emission plans directly into the MMS session of the user.

Finally, connections to enabling tools such as WareM&O should be automated wherever feasible so that the entire flow (discovery, decision and execution) can be driven from MMS without a manual context switch. Where providers support it, MMS should call booking and cancellation endpoints directly and receive confirmations asynchronously through well-defined callbacks; where a redirect remains necessary, the hand-off should be standardised so that it feels like a continuation of the same journey. In both patterns, UFDS remains the authoritative layer for harmonised data and state changes, and MMS remains the single, neutral access point through which users enable DISCO-X tools to deliver PI-aligned outcomes.

9. Conclusions

The DISCO project was initiated to address pressing challenges in urban logistics by developing and implementing innovative solutions (the DISCO-Xs) across several European living labs. This initiative translated the conceptual framework of the Physical Internet into tangible, operational tools tailored to specific urban contexts. The results from the project demonstrate a meaningful advance in making city logistics more efficient, sustainable and interconnected; this provides a pathway for cities to advance their logistics capabilities in alignment with the PI vision. The strategic alignment of the DISCO-X solutions with the five core PI Building Blocks (areas of development) (summarized in Figure 36) offers an overview of the contributions from the project.

A key conclusion from this alignment is that the DISCO project has made tangible contributions across the full spectrum of the Physical Internet building blocks. While foundational efforts concentrated on Logistics Nodes (e.g., the WareM&O Platform, Optimal Facility Location Tool) and Logistics Networks (e.g., UAC System, Digital Twin), demonstrating with Living Lab urban logistics measures the optimization of assets (both physical and digital) and operational flows, the project also produced advancements in the enabling layers.

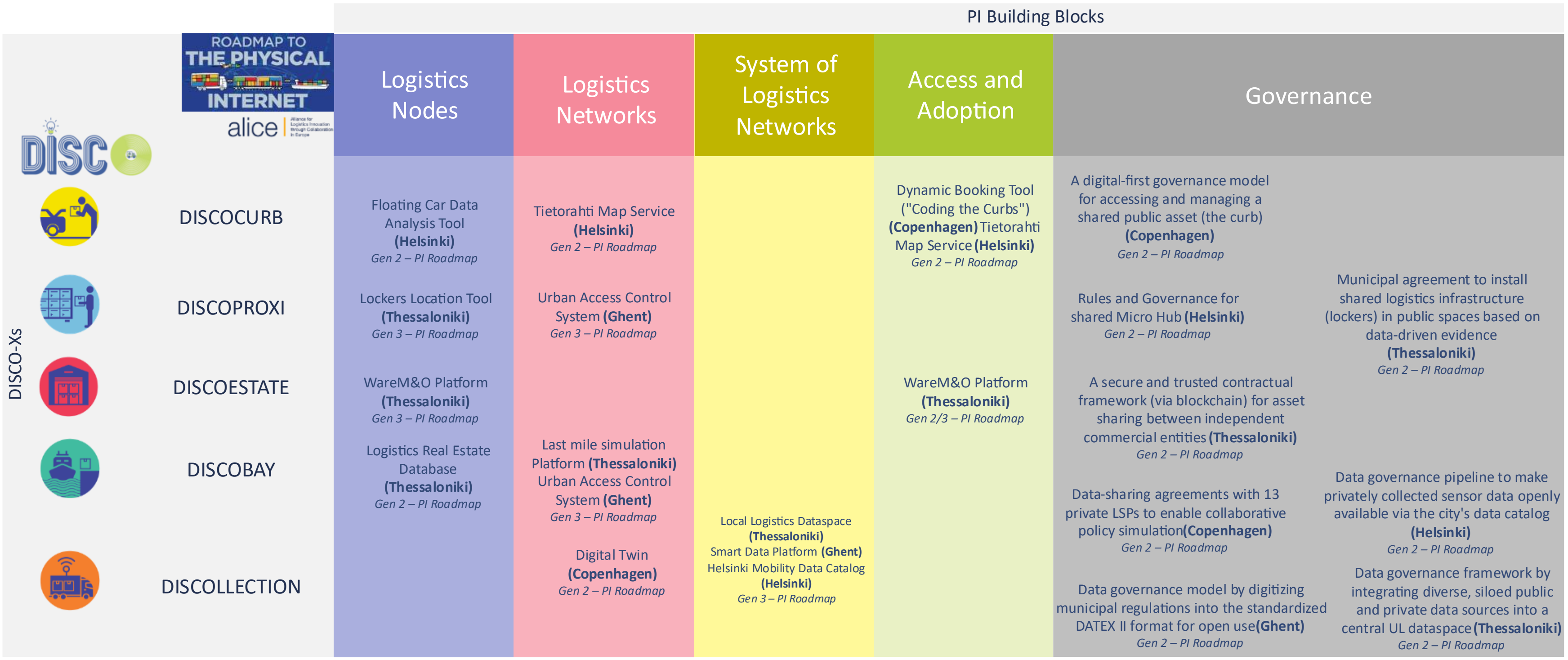


Figure 36: DISCO-Xs x PI Building Blocks Alignment

The project also addresses the highest level of integration (the System of Logistics Networks) through the data-centric solutions within DISCOLLECTION. The development of the Local Logistics Dataspace, the Smart Data Platform and the Helsinki Mobility Data Catalog represents a significant development that creates the foundation for an interconnected "network of networks". The practical validation of this architecture is demonstrated in Chapter 8, where a proof-of-concept use case shows the MMS, UFDS and WareM&O tools functioning as a single, federated system for asset discovery and booking. Furthermore, tools focused on user interfaces and new interaction models (such as the Dynamic Booking Tool and Tietorahti Map Service integration for DISCOCURB) directly enhanced *Access and Adoption*.

The analysis also indicates that the project facilitated progress in the Governance building block. The implementation of these tools also led to new governance models, such as establishing multi-stakeholder data-sharing agreements (Copenhagen), creating secure contractual frameworks for asset sharing (Thessaloniki) and achieving municipal agreement for installing shared infrastructure in public spaces (Thessaloniki). This demonstrates that technological innovation and governance evolution are intrinsically linked in the transition towards the PI.

Placing the DISCO-X tools in the context of the respective generations of these building blocks shows that the project contributes to an evolutionary path. This means that, initially, first-generation capabilities focused on digitizing and understanding core assets (e.g., Logistics Real Estate Database, Helsinki Mobility Data Catalog). Building on this, second-generation tools introduced dynamic management and operational control (e.g., Dynamic Booking Tool, UAC System). Finally, the most advanced tools of the project (e.g., Digital Twin, Smart Data Platform and Local Logistics Dataspace) serve as precursors to a third generation, representing an integrated and intelligent ecosystem.

In summary, the DISCO project demonstrated that a portfolio of targeted, city-specific interventions can build momentum towards a PI-aligned logistics system. The living lab approach across Thessaloniki, Helsinki, Ghent and Copenhagen confirms these solutions are practically applicable and adaptable to diverse urban environments in Europe. The groundwork from DISCO provides a trajectory for future development (fostering integration between the deployed DISCO-X solutions, scaling their implementation and leveraging the data infrastructure established by DISCOLLECTION to achieve greater system-wide interoperability). Continued engagement with stakeholders is necessary to ensure these advanced tools are adopted, which will transform urban logistics into an efficient, collaborative and sustainable ecosystem.

References

Alliance for Logistics Innovation through Collaboration in Europe (ALICE-ETP). (2020). Roadmap to the Physical Internet. https://www.etp-logistics.eu/wp-content/uploads/2022/11/Roadmap-to-Physical-Intenet-Executive-Version_Final-web.pdf

Neels, M. (2023, March). Empowering local authorities towards successful dynamic curbside management. POLIS Network. https://www.polisnetwork.eu/document/empowering-local-authorities-towards-successful-dynamic-curbside-management/

Burns, A., Forsythe, C. R., Michalek, J. J., & Whitefoot, K. (2025). Estimating the potential for dynamic parking reservation systems to increase delivery vehicle accommodation. Transportation Research Part A: Policy and Practice, 193, 104380. https://doi.org/10.1016/j.tra.2025.104380

Diehl, C., Ranjbari, A., & Goodchild, A. (2021). Curbspace management challenges and opportunities from public and private sector perspectives. Transportation Research Record: Journal of the Transportation Research Board, 2675(11). https://doi.org/10.1177/03611981211027156

L'Istituto sui Trasporti e la Logistica (ITL). (2024). Deliverable D4.2 – Starring LL measures implementation. DISCO Project.

Lindholmen Science Park (LSP) (2025). Deliverable D4.3 – Evaluation of impacts at Starring LLs. DISCO Project.

Ajmar, A., Arco, E., Boccardo, P., & Perez, F. (2019). Floating car data (FCD) for mobility applications. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W13, 1517–1523. https://doi.org/10.5194/isprs-archives-XLII-2-W13-1517-2019

Comi, A., & Polimeni, A. (2021). Forecasting delivery pattern through floating car data: Empirical evidence. Future Transportation, 1(3), 707–719. https://doi.org/10.3390/futuretransp1030038

Maxner, T., Ranjbari, A., Dowling, C. P., & Güneş, Ş. (2023). Simulation-based analysis of different curb space type allocations on curb performance. Transportmetrica B: Transport Dynamics, 11(1), 1384–1405. https://doi.org/10.1080/21680566.2023.2212324

ALICE-ETP. (2021). A framework and process for the development of a Roadmap towards zero emissions logistics 2050. ALICE-ETP. https://www.etp-logistics.eu/wp-content/uploads/2019/12/Alice-Zero-Emissions-Logistics-2050-Roadmap-WEB.pdf

ALICE-ETP, & POLIS. (2021). Cities-Regions and Companies working together Guide for advancing towards zero-emission urban logistics by 2030. ALICE-ETP. https://www.etp-logistics.eu/wp-content/uploads/2021/12/POLIS_ALICE_Guide-Zero-Emission-Urban-Logistics_Dec2021-low.pdf

Ballot, E., Montreuil, B., & Meller, R. (2014). The Physical Internet. La Documentation Française.

Bates, O., Friday, A., Allen, J., Cherrett, T., McLeod, F., Bektas, T., Nguyen, T., Piecyk, M., Piotrowska, M., Wise, S., & Davies, N. (2018). Transforming last-mile logistics: Opportunities for more sustainable deliveries. Conference on Human Factors in Computing Systems - Proceedings, 2018-April. https://doi.org/10.1145/3173574.3174100

BCI. (2023). Handboek multi-client stadshubs.

CIVITAS. (2015). Smart choices for cities: making urban freight logistics more sustainable. https://civitas.eu/sites/default/files/civ_pol-an5_urban_web.pdf

Comi, A., & Savchenko, L. (2021). Last-mile delivering: Analysis of environment-friendly transport. Sustainable Cities and Society, 74. https://doi.org/10.1016/j.scs.2021.103213
Cossu, P., Astegiano, P., Fergadiotou, I., Ayfantopoulou, G., & Franklin, R. (2025). Data-Driven and Dynamic Land Use for Physical Internet-Led Urban Logistics and Planning. In Lecture Notes in Mobility: Vol. Part F658 (pp. 569–575). Springer. https://doi.org/10.1007/978-3-031-95284-5_81
Crainic, T. G., & Montreuil, B. (2016). Physical Internet Enabled Hyperconnected City Logistics. Transportation Research Procedia, 12, 383–398. https://doi.org/10.1016/j.trpro.2016.02.074
de Bok, M., Tavasszy, L., & Sebastiaan Thoen. (2022). Application of an empirical multi-agent model for urban goods transport to analyze impacts of zero emission zones in The Netherlands. Transport Policy, 124, 119–127. https://doi.org/10.1016/j.tranpol.2020.07.010
Demir, E., Syntetos, A., & Van Woensel, T. (2022). Last mile logistics: Research trends and needs. IMA Journal of Management Mathematics, 33(4), 549–561. https://doi.org/10.1093/imaman/dpac006
Desrousseaux, M. (n.d.). Benchmark stedelijke multimodale platformen Globale visie Schaarbeek-Vorming. www.rebelgroup.com
Dobers, K., Perotti, S., Wilmsmeier, G., Mauer, G., Jarmer, J. P., Spaggiari, L., Hering, M., Romano, S., & Skalski, M. (2023). Sustainable logistics hubs: greenhouse gas emissions as one sustainability key performance indicator. Transportation Research Procedia, 72, 1153–1160. https://doi.org/10.1016/j.trpro.2023.11.572
Hesse, M., & Rodrigue, J. P. (2004). The transport geography of logistics and freight distribution. Journal of Transport Geography, 12(3), 171–184. https://doi.org/10.1016/j.jtrangeo.2003.12.004
Jain, A. ;, Heijden, R. E. C. M., Van Der; Marchau, V. A. W. J. ;, Bruckmann, D., Jain, A., Van Der Heijden, R. E. C. M., Marchau, V. A. W. J., & Bruckmann, D. (2021). The role of digital platforms in the modal shift of road to rail freight transport. https://hdl.handle.net/2066/296582
McKinnon, A. (2018). Decarbonizing Logistics - distributing goods in a low-carbon world (1st ed.). KoganPage.
Mohammad, W. A. M., Nazih Diab, Y., Elomri, A., & Triki, C. (2023). Innovative solutions in last mile delivery: concepts, practices, challenges, and future directions. In Supply Chain Forum (Vol. 24, Issue 2, pp. 151–169). Taylor and Francis Ltd. https://doi.org/10.1080/16258312.2023.2173488
Montreuil, B. (2011). Toward a Physical Internet: meeting the global logistics sustainability grand challenge. Logistics Research, 3, 71–87. https://doi.org/https://doi.org/10.1007/s12159-011-0045-x
Nefs, M., & Daamen, T. (2022). Behind the big box: understanding the planning-development dialectic of large distribution centres in Europe. European Planning Studies, March, 1–22. https://doi.org/10.1080/09654313.2022.2057792
POLIS. (2023). LEAD Policy Recommendations. https://www.leadproject.eu/wp-content/uploads/2023/11/LEAD-Policy-Recommendations.pdf
POLIS, & Environmental Defense Fund Europe. (2022). Sustainable URban Freight. https://www.polisnetwork.eu/wp-content/uploads/2022/05/SURF-Final-Report_2022-April_28-standard.pdf
POLIS, & TDA. (2020). How-to Guide Zero-Emission Zones Don’t Wait to Start with Freight! https://www.polisnetwork.eu/wp-content/uploads/2020/12/ZEZ-F_How-to-Guide_low.pdf

Pourrahmani, E., & Jaller, M. (2021). Crowdshipping in last mile deliveries: Operational challenges and research opportunities. In Socio-Economic Planning Sciences (Vol. 78). Elsevier Ltd. https://doi.org/10.1016/j.seps.2021.101063

Prandtstetter, M., Biesinger, B., Hu, B., Nolz, P., Reinthaler, M., Zajicek, J., & Angelini, A. (2019). On the Potentials and Dilemmas of Cooperative/White-Label Deliveries based on Selected Austrian Demonstration Cases. Proceedings of the 6th International Physical Internet Conference IPIC 2019.

Pufahl, L., Ihde, S., Glöckner, M., Franczyk, B., Paulus, B., & Weske, M. (2020). Countering congestion: a white-label platform for the last mile parcel delivery. . In Business Information Systems (Vol. 389, pp. 210–223). Springer.

Russo, F., & Comi, A. (2024). New challenges for city logistics: A unified view of energy and transport systems for addressing sustainability. Transportation Research Procedia, 79, 313–320. https://doi.org/10.1016/j.trpro.2024.03.042

Russo, S. M., Voegl, J., & Hirsch, P. (2021). A multi-method approach to design urban logistics hubs for cooperative use. Sustainable Cities and Society, 69. https://doi.org/10.1016/j.scs.2021.102847

Sweere, S. (2023). The Ex-post-performance evaluation of urban consolidation centers in the Netherlands [Msc]. TU Delft.

TDA. (2019). Zero Emission Urban Freight: A TDA whitepaper on how to reach zero emission urban freight by uniting Countries, Cities/Regions and Companies. https://tda-mobility.org/wp-content/uploads/2019/05/TDA-Zero-Emission-Urban-Freight.pdf

Transport Forum International. (2024). The Final Frontier of Urban Logistics: Tackling the Last Metres. https://www.itf-oecd.org/sites/default/files/docs/final-frontier-urban-logistics.pdf

Annexes

Annex A: Technical API Documentation

Logistics Real Estate Database

Thessaloniki Urban Logistics Dataspace REST API 0.0.1 OAS 3.1

/openapi.json

A collection of REST APIs available for the Thessaloniki Urban Logistics Dataspace API server.

This is a server that can serve various APIs and authenticates users calling its APIs. An initial user with `admin` permissions is required to be inserted by default in the database in order to create more users. A user can have one of the following permissions

- `admin` : The user can call user and permission management APIs, as well as every other API in this server
- `write` : The user can send data to write-only APIs
- `read` : The user can call read-only APIs

Root
Root route for *pinging* purposes.

- GET `/` Root route just for testing purposes

Users
Users operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have `admin` permission.

- GET `/api/v1/users` Get a collection of users
- POST `/api/v1/users` Create a new user
- GET `/api/v1/users/{username}` Get a user by username
- PUT `/api/v1/users/{user_id}` Update an existing user
- DELETE `/api/v1/users/{user_id}` Delete an existing user

Permissions
Users permissions to call specific APIs operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have `admin` permission.

- POST `/api/v1/permissions` Create a new permission for a user
- PUT `/api/v1/permissions/users/{user_id}` Update an existing user's permission
- DELETE `/api/v1/permissions/users/{user_id}` Delete an existing user's permission

Shops
Shops operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `read` permission.

- GET `/api/v1/moth-shops` Get all Municipality of Thessaloniki shops of sanitary interest data

Warehouses
Warehouses operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `read` permission.

- GET `/api/v1/warehouses` Get all warehouses data

Lockers
Lockers operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `read` permission.

- GET `/api/v1/lockers/eett` Get all Hellenic Telecommunications and Post Commission lockers data

Insights
Database Insights. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `read` permission.

- GET `/api/v1/moth/insights/voronoi-geometry` Get the Municipality of Thessaloniki geometry data required to build the Voronoi diagram in the frontend in GeoJSON

Spitogatos
Spitogatos operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `admin` permission.

- POST `/api/v1/spitogatos/data` Get Spitogatos data and insert them into our database

GAIAOSE
GAIAOSE operations. All APIs require authentication via `x-username` and `x-api-key` headers and also the user to have at least `admin` permission.

- POST `/api/v1/gaiaose/data` Get GAIAOSE data and insert them into our database

WareM&O
WareM&O operations. All APIs require authentication via `x-username` and `x-api-key` headers.

- POST `/api/v1/waremo/tif/data` Get TIF warehouse data from WareM&O and save them in the database - Must have "admin" permission
- POST `/api/v1/waremo/fm-logistic/data` Get FM LOGISTIC warehouse data from WareM&O and save them in the database - Must have "admin" permission
- POST `/api/v1/waremo/tif/availabilities` Create TIF compartment availabilities inside WareM&O - Must have at least "write" permission
- GET `/api/v1/waremo/warehouses` Get all WareM&O warehouse data
- POST `/api/v1/waremo/fm-logistic/availabilities` Create FM LOGISTIC compartment availabilities inside WareM&O - Must have at least "write" permission

Schemas

- AuthResponseModel › Expand all `object`
- AvailabilityModel › Expand all `object`
- BaseModel › Expand all `object`
- CompanyModel › Expand all `object`
- CreateResourceRequestBodyModel-Output › Expand all `object`
- CreatedUserModel › Expand all `object`
- Feature › Expand all `object`
- FeatureCollection › Expand all `object`
- GeneralizedUserModel › Expand all `object`
- GeometryCollection › Expand all `object`
- GetGeoJSONResourceResponseModel › Expand all `object`
- GetResourceResponseModel › Expand all `object`
- GetWarehouseCollectionResponseModel › Expand all `object`
- HTTPValidationError › Expand all `object`
- LineString › Expand all `object`
- LocationModel › Expand all `object`
- LockerModel › Expand all `object`
- MothShopAttributeModel › Expand all `object`
- MothShopGeometryModel › Expand all `object`
- MultiLineString › Expand all `object`
- MultiPoint › Expand all `object`
- MultiPolygon › Expand all `object`
- PermissionTypeEnum › Expand all `string`
- Point › Expand all `object`
- Polygon › Expand all `object`
- Position2D › Expand all `array<[number, number], any>` 2 items
- Position3D › Expand all `array<[number, number, number], any>` 3 items
- ResponseModel › Expand all `object`
- ResultModel › Expand all `object`
- ShopModel › Expand all `object`
- UserIdModel › Expand all `object`
- UserModel › Expand all `object`
- ValidationError › Expand all `object`
- app__pydantic_models__availability__CreateResourceRequestBodyModel › Expand all `object`
- app__pydantic_models__availability__CreateResourceResponseModel › Expand all `object`
- app__pydantic_models__compartment__CompartmentModel › Expand all `object`
- app__pydantic_models__lockers__GetCollectionResponseModel › Expand all `object`
- app__pydantic_models__permissions__CreateResourceRequestBodyModel › Expand all `object`
- app__pydantic_models__permissions__CreateResourceResponseModel › Expand all `object`
- app__pydantic_models__permissions__UpdateResourceRequestBodyModel › Expand all `object`
- app__pydantic_models__permissions__UpdateResourceResponseModel › Expand all `object`
- app__pydantic_models__shops__GetCollectionResponseModel › Expand all `object`
- app__pydantic_models__users__CreateResourceRequestBodyModel › Expand all `object`
- app__pydantic_models__users__CreateResourceResponseModel › Expand all `object`
- app__pydantic_models__users__GetCollectionResponseModel › Expand all `object`
- app__pydantic_models__users__UpdateResourceRequestBodyModel › Expand all `object`
- app__pydantic_models__users__UpdateResourceResponseModel › Expand all `object`
- app__pydantic_models__warehouse__GetCollectionResponseModel › Expand all `object`
- app__pydantic_models__warehouse__WarehouseModel › Expand all `object`
- app__pydantic_models__waremo__CompartmentModel › Expand all `object`
- app__pydantic_models__waremo__WarehouseModel › Expand all `object`

WareM&O by DISCOPROXI & DISCOESTATE

Thessaloniki VFC WareM&O by DISCOPROXI & DISCOESTATE REST API

Account

POST /api/services/app/Account/IsTenantAvailable

POST /api/services/app/Account/Register

BookingRequest

GET /api/services/app/BookingRequest/GetAll

GET /api/services/app/BookingRequest/GetBookingRequestsByWarehouseId

POST /api/services/app/BookingRequest/AcceptBooking

POST /api/services/app/BookingRequest/Create

GET /api/services/app/BookingRequest/Get

PUT /api/services/app/BookingRequest/Update

PUT /api/services/app/BookingRequest/UpdateRating

DELETE /api/services/app/BookingRequest/Delete

GET /api/services/app/BookingRequest/GetAllBookingsByMonth

GET /api/services/app/BookingRequest/GetAllBookingsPrices

GET /api/services/app/BookingRequest/GetAllSearchesByMonth

Company

GET /api/services/app/Company/GetAll

POST /api/services/app/Company/Create

PUT /api/services/app/Company/Update

GET /api/services/app/Company/Get

GET /api/services/app/Company/GetByCreatorId

DELETE /api/services/app/Company/Delete

CompanyUser

GET /api/services/app/CompanyUser/GetAll

GET /api/services/app/CompanyUser/GetListFilter

POST /api/services/app/CompanyUser/Create

DELETE /api/services/app/CompanyUser/Delete

Compartment

GET /api/services/app/Compartment/GetListByWarehouseId

GET /api/services/app/Compartment/GetAll

POST /api/services/app/Compartment/Create

PUT /api/services/app/Compartment/Update

GET /api/services/app/Compartment/Get

DELETE /api/services/app/Compartment/Delete

CompartmentAvailability

GET /api/services/app/CompartmentAvailability/GetAll

GET /api/services/app/CompartmentAvailability/GetCompartmentsAvailabilitiesByWarehouseId

GET /api/services/app/CompartmentAvailability/GetCompartmentsAvailabilitiesByCompartmentId

GET /api/services/app/CompartmentAvailability/GetAvailabilyIndexByStorageTypeAndCondition

DELETE /api/services/app/CompartmentAvailability/Delete

POST /api/services/app/CompartmentAvailability/Create

GET /api/services/app/CompartmentAvailability/Get

PUT /api/services/app/CompartmentAvailability/Update

GET/api/services/app/CompartmentAvailability/GetAllCompartmentAvailabilitiesGrouped

GET/api/services/app/CompartmentAvailability/GetAvailabilityIndex

GET/api/services/app/CompartmentAvailability/GetAveragePricesPerUnit

CompartmentTaxonomy

GET/api/services/app/CompartmentTaxonomy/GetAll

POST/api/services/app/CompartmentTaxonomy/Create

DELETE/api/services/app/CompartmentTaxonomy/Delete

Configuration

POST/api/services/app/Configuration/ChangeUiTheme

Country

GET/api/services/app/Country/GetAll

POST/api/services/app/Country/Create

PUT/api/services/app/Country/Update

GET/api/services/app/Country/Get

DELETE/api/services/app/Country/Delete

Event

GET/api/services/app/Event/GetAll

GET/api/services/app/Event/GetDetails

POST/api/services/app/Event/Create

POST/api/services/app/Event/Cancel

POST/api/services/app/Event/Register

POST/api/services/app/Event/CancelRegistration

Location

GET/api/services/app/Location/GetAll

GET/api/services/app/Location/GetCities

POST/api/services/app/Location/Create

PUT/api/services/app/Location/Update

GET/api/services/app/Location/Get

DELETE/api/services/app/Location/Delete

MemberType

GET/api/services/app/MemberType/GetAll

POST/api/services/app/MemberType/Create

PUT/api/services/app/MemberType/Update

GET/api/services/app/MemberType/Get

DELETE/api/services/app/MemberType/Delete

Role

POST/api/services/app/Role/Create

GET/api/services/app/Role/GetRolesAsync

PUT/api/services/app/Role/Update

DELETE/api/services/app/Role/Delete

GET/api/services/app/Role/GetAllPermissions

GET/api/services/app/Role/GetRoleForEdit

GET/api/services/app/Role/Get

GET/api/services/app/Role/GetAll

Session

GET/api/services/app/Session/GetCurrentLoginInformations

StandardFeature

GET/api/services/app/StandardFeature/GetAll

POST/api/services/app/StandardFeature/Create

PUT/api/services/app/StandardFeature/Update

GET/api/services/app/StandardFeature/Get

DELETE/api/services/app/StandardFeature/Delete

Taxonomy

GET/api/services/app/Taxonomy/GetCurrentLanguageFromCookie

GET/api/services/app/Taxonomy/GetAll

POST/api/services/app/Taxonomy/Create

PUT/api/services/app/Taxonomy/Update

GET/api/services/app/Taxonomy/Get

DELETE/api/services/app/Taxonomy/Delete

Tenant

POST/api/services/app/Tenant/Create

DELETE/api/services/app/Tenant/Delete

GET/api/services/app/Tenant/Get

GET/api/services/app/Tenant/GetAll

PUT/api/services/app/Tenant/Update

TokenAuth

POST/api/TokenAuth/Authenticate

GET/api/TokenAuth/GetExternalAuthenticationProviders

POST/api/TokenAuth/ExternalAuthenticate

User

POST/api/services/app/User/Validate

POST/api/services/app/User/Register

POST/api/services/app/User/Create

PUT/api/services/app/User/Update

PUT/api/services/app/User/UpdateUserOwn

DELETE/api/services/app/User/Delete

GET/api/services/app/User/GetRoles

GET/api/services/app/User/GetCurrentLanguage

POST/api/services/app/User/ChangeLanguage

GET/api/services/app/User/Get

GET/api/services/app/User/GetAll

UserAuth

POST/api/services/app/UserAuth/AuthorizeUser

Warehouse

GET/api/services/app/Warehouse/GetFiltered

GET/api/services/app/Warehouse/GetListWithCompartments

GET/api/services/app/Warehouse/GetIfExists

GET/api/services/app/Warehouse/GetAll

GET/api/services/app/Warehouse/GetListByCompanyId

POST/api/services/app/Warehouse/Create

PUT /api/services/app/Warehouse/Update

GET /api/services/app/Warehouse/Get

GET /api/services/app/Warehouse/GetRatingsById

GET /api/services/app/Warehouse/GetByGuid

DELETE /api/services/app/Warehouse/Delete

GET /api/services/app/Warehouse/GetCurrentLanguageFromCookie

GET /api/services/app/Warehouse/GetCurrentLanguage

GET /api/services/app/Warehouse/GetAllSummaries

GET /api/services/app/Warehouse/GetAllSummariesStCond

GET /api/services/app/Warehouse/GetAllSummariesOffSrvs

GET /api/services/app/Warehouse/GetAllSummariesPrdType

GET /api/services/app/Warehouse/GetTota

Annex B: User Manuals for Key Tools

DISCOLLECTION Smart Data Platform Setup guide

Data-driven and Dynamic Space and Assets for Physical Internet-led Urban Logistics and Planning

DISCOLLECTION SMART DATA PLATFORM SETUP GUIDE

IMEC

Alper BASAK

April 2025

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101103954. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

Abstract

Data spaces are a great way for cities to connect to logistics players in their cities, and integrate valuable city data directly into the logistics ecosystem. However, making city data available on the data space can come with some technical challenges, both in transforming the data and connecting to the data space.

The Smart Data Platform reduces these obstacles to offering data on the data space by providing city data engineers with the tools they need to prepare their data for data space sharing and connect it to the data space. To achieve this, The Smart Data Platform makes use of a pre-existing data transformation (ETL) tool, packaged with a data space connector, which it connects through an overarching UI. This provides cities with all the capabilities of an ETL tool, while reducing the barrier to connecting to the data space.

The below document serves as a setup guide for the Smart Data Platform and attached urban Freight data space connector. The setup guide follows the case study of the Ghent Living lab where data was gathered from the Gent Open Data Platform to the Smart Data Platform, and made available to the Disco network via the embedded Sovity dataspace connector (and UI). The setup guide starts by presenting the overall architecture and building blocks of the Smart Data Platform. It provides a comprehensive description of each of these building blocks, as well as a technical installation guide, providing access to source codes for easy downloads.

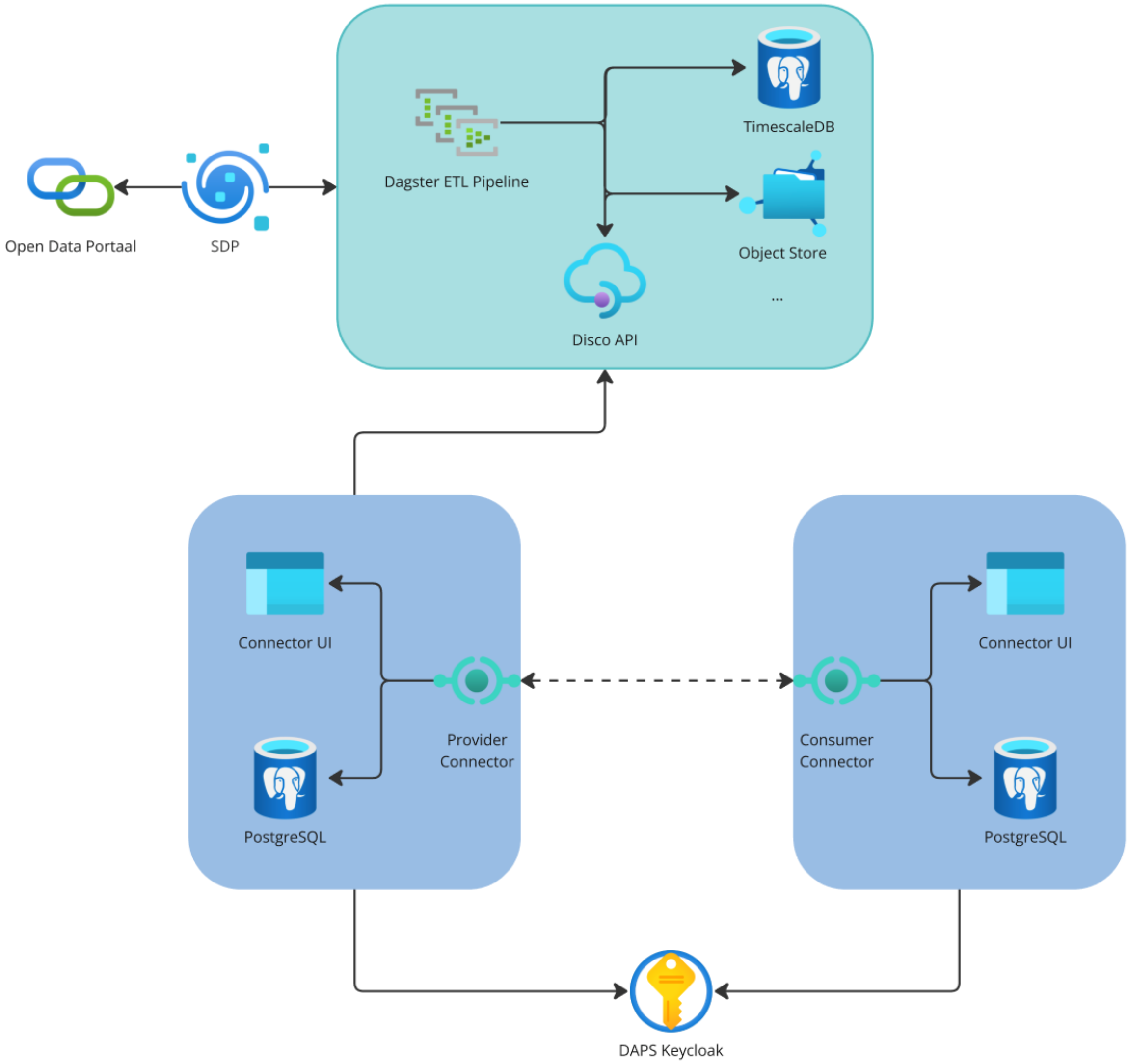
Document history

Version	Date	Organisation	Main area of changes	Comments
0.8	10 April 2025	imec		First version
0.9	18 April 2025	Imec	Added some extra technical docs and refined some text	

List of acronyms

EDS	European Data Space
EDC	Eclipse Dataspace Connector
OAuth2	Open Authorization 2.0
UVAR	Urban Vehicle Access Regulations

1. SDP System Architecture



Full System Architecture overview

The DISCOLLECTION solution consists of key components that interact with various parts of the system to facilitate data flow and integration. This architecture represents the data flow and system integration for delivering standardized, interoperable data to a dataspace environment.

The Smart Data Platform (SDP) serves as the central orchestrator that coordinates various components of the data lifecycle, from retrieval to consumption:

- Data Retrieval:
  DISCOLLECTION interfaces with the Gent Open Data Platform[1] using a REST API to fetch relevant datasets periodically. This platform provides data on public parking spaces, street parking locations, and real-time occupancy. These raw datasets will be processed by an ETL (Extract, Transform, Load) pipeline managed by Dagster[2]. SDP ensures that the data is retrieved in a timely and efficient manner, leveraging its orchestration capabilities to match raw data with the appropriate transformation processes and scheduling.

- Data Transformation:
  The Dagster ETL Pipeline plays a crucial role in transforming raw data into standardized formats such as APDS[3], which is standard in parking spaces or the DATEX II[4], which is a European standard for the exchange of traffic and travel data. This transformation ensures consistency and standardization across different data sources. The pipeline is orchestrated by SDP and may include validation, enrichment, and restructuring operations to meet the schema and semantic requirements of these formats.

- Data Storage:
  Once transformed, data is stored in one or more storage backends. The architecture supports flexibility in storage options, including TimescaleDB[5] for time-series data providing historical data queries and Object Stores[6] for unstructured or bulk data. These stores are part of the same infrastructure layer that includes an API server responsible for making the processed data accessible.

- DISCO Dataspace Connector:
  The DISCOLLECTION API facilitates interaction with the dataspace by exposing standardized data assets to external consumers via the Sovity[7] connectors. This includes Provider Connectors that publish data and Consumer Connectors that retrieve and consume it.

- Identity and Access Management:

---

[1] https://data.stad.gent/explore
[2] https://dagster.io
[3] https://www.allianceforparkingdatastandards.org/
[4] https://datex2.eu/
[5] https://www.timescale.com/
[6] https://en.wikipedia.org/wiki/Object_storage
[7] https://sovity.de/en/sovity-en/

All components involved in data exchange and access via the connectors are integrated with DAPS[8]; a Keycloak[9] implementation for Sovity, for secure authentication and authorization, ensuring trusted interactions within the dataspace.

- Integration with other DISCO-X solutions:
  The data provided by DISCOLLECTION can be integrated with other DISCO-X solutions across various cities and regions. This integration can help in visualizing and analyzing different types of geodata in logistics, such as historical occupation info, parking locations, and logistics real estate.

By clearly defining the interaction between these components, DISCOLLECTION ensures a seamless flow of data from initial collection to final distribution, enhancing the accessibility and utility of city data within the data space.

[8] https://github.com/sovity/sovity-daps
[9] https://www.keycloak.org/

2. Smart Data Platform

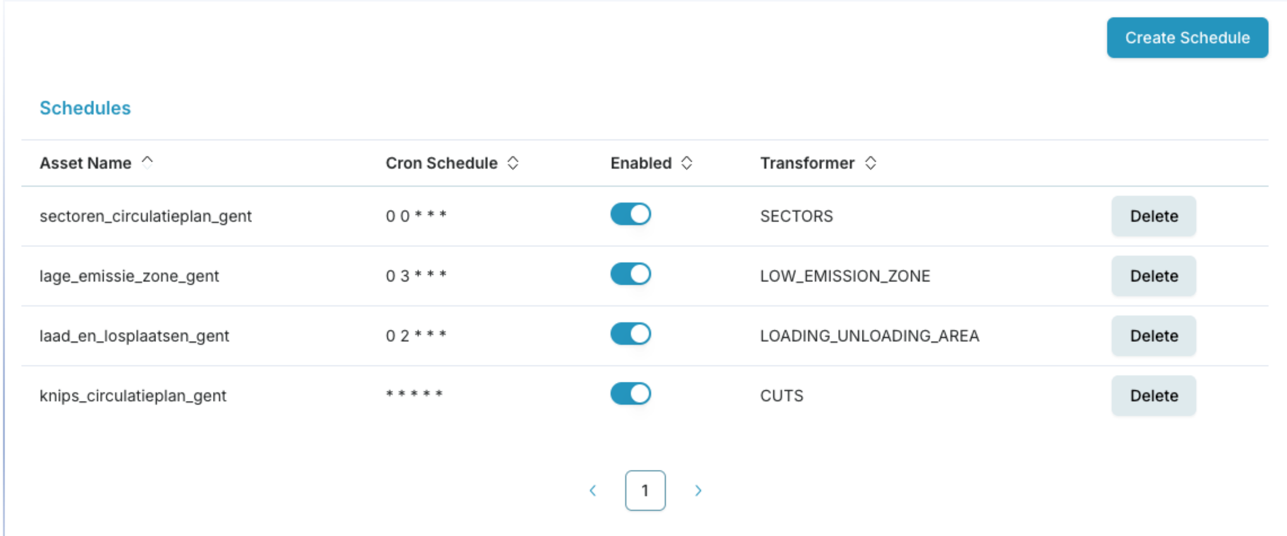
2.1. Component Description

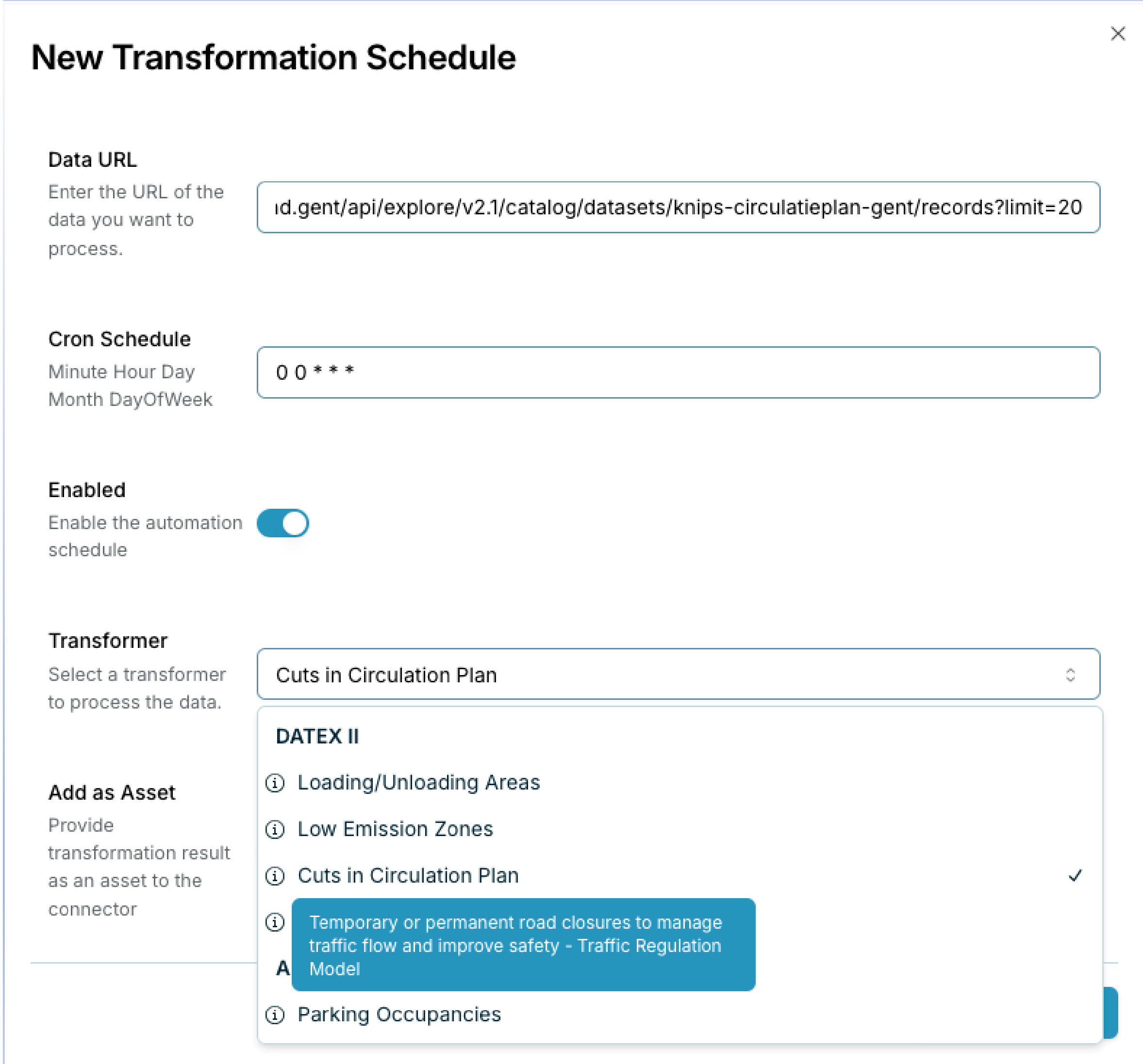
The Smart Data Platform is a data management and analytics platform that provides a unified environment for data ingestion, processing, storage, and analysis. It is designed to handle standardization and normalization of data from various sources. The platform acts as a central hub for data integration, allowing users to setup a pipeline for connecting the data sources, performing data transformations, and storing the data in a structured format. Scheduling capabilities are also included to automate the data ingestion and processing tasks.

2.2. Technical Specifications

Smart Data Platform is a Next.js[10] application that runs on Node.js. It is designed to be deployed in a cloud environment, and can be accessed via a web interface. The platform supports pipeline management for data ingestion and processing, allowing users to define the flow of data from source to destination. Platform uses the PostgreSQL[11] database for storing pipeline metadata and configuration information. User manages pipelines through a web interface, where they can define the data source url, data transformation rules and schedules for data ingestion.

[10] https://nextjs.org/
[11] https://www.postgresql.org/


Create Schedule
Schedules
Asset Name
Cron Schedule
Enabled
Transformer
sectoren_circulatieplan_gent
0 0 * * *
SECTORS
Delete
lage_emissie_zone_gent
0 3 * * *
LOW_EMISSION_ZONE
Delete
laad_en_losplaatsen_gent
0 2 * * *
LOADING_UNLOADING_AREA
Delete
knips_circulatieplan_gent
* * * * *
CUTS
Delete
1


New Transformation Schedule
Data URL
Enter the URL of the data you want to process.
ıd.gent/api/explore/v2.1/catalog/datasets/knips-circulatieplan-gent/records?limit=20
Cron Schedule
Minute Hour Day Month DayOfWeek
0 0 * * *
Enabled
Enable the automation schedule
Transformer
Select a transformer to process the data.
Cuts in Circulation Plan
DATEX II
Loading/Unloading Areas
Low Emission Zones
Cuts in Circulation Plan
Temporary or permanent road closures to manage traffic flow and improve safety - Traffic Regulation Model
Parking Occupancies
Add as Asset
Provide transformation result as an asset to the connector

Amongst the available standard transformers, suitable ones can be selected and will initiate the pipeline in the Dagster via the Schedule Monitor (see 1.3).

For convenience it is possible to automatically create an asset in the connector for the processed data.

2.3. Configuration Details

3 main configuration sets are required for the Smart Data Platform:

- Database configuration: This includes the database connection string, username, password, and other database settings.
- Auth configuration: This includes the Keycloak server URL, client ID, and other authentication settings for the platform.
- Connector configuration: This includes the configuration for the data space connectors, such as the management URL and secret key.

```
# DB
DB_HOST=
DB_PORT=
DB_USER=
DB_PASSWORD=
DB_NAME=
DB_SSL=

# NextAuth
AUTH_SECRET=
AUTH_KEYCLOAK_ID=
AUTH_KEYCLOAK_SECRET=
AUTH_KEYCLOAK_ISSUER=

# EDC CONNECTOR
EDC_HOST=
EDC_API_KEY=
```

2.4. Deployment Information

A Docker and a Helm chart is provided for the Smart Data Platform, which can be deployed in a cloud environment. Helm chart includes services, configmaps and deployment specifications for the platform. The manifest looks like this:

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: disco-schedule-manager
spec:
  releaseName: disco-schedule-manager
  interval: 5m
  chart:
    spec:
      chart: disco-schedule-manager
      version: 0.1.4
      sourceRef:
        kind: HelmRepository
        name: acrdisco-charts
        namespace: flux-system
  values:
    environment:
      DB_HOST: 'timescaledb.etl'
      AUTH_TRUST_HOST: true
      NEXTAUTH_URL: 'https://sdp.${domain}'
      DB_SSL: false
      EDC_HOST: 'https://provider.${domain}'
    secrets:
      - name: DB_PASSWORD
        secret:
disco.timescaledb.credentials.postgresql.acid.zalan.do
        key: password
      - name: AUTH_SECRET
        secret: schedule-manager-secrets
        key: auth-secret
      - name: AUTH_KEYCLOAK_SECRET
        secret: schedule-manager-secrets
        key: auth-keycloak-secret
      - name: EDC_API_KEY
        secret: schedule-manager-secrets
        key: edc-api-key
```

An ingress[12] is also included to expose the platform.

[12] https://sdp.disco.ai-data.imec.be

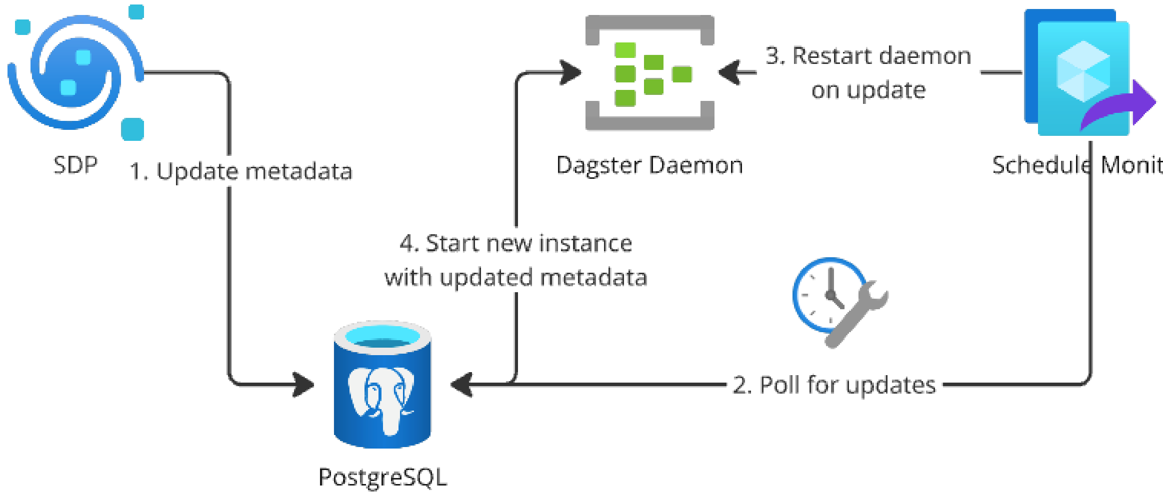
3. Schedule Monitor

3.1. Component Description

The Dagster Kubernetes deployment does not allow for runtime asset or schedule updates. On startup, the Dagster service will load all the definitions from the user code image, which is built and pushed to a container registry. The Dagster service will then use this image to run the user code in a Kubernetes cluster. If the User Code already has schedules and assets defined, they cannot be changed at runtime. This scenario is not really user friendly, as it requires a new image to be built and pushed to the container registry every time a schedule or asset is changed. This is not ideal for a production environment, where schedules and assets may need to be updated frequently.
An alternative method is being used here to allow for runtime updates of schedules and assets. As Dagster bootstraps the user code, it is possible to inject the asset and schedule definitions from external resources, such as the pipeline metadata database used by the Smart Data Platform.
The Dagster Kubernetes deployment uses this custom Kubernetes container image, namely Schedule Monitor, as a sidecar to the Dagster service. This sidecar is responsible for monitoring the schedule and asset changes in the PostgreSQL database that Smart Data Platform uses to store the pipeline metadata. Schedule Monitor polls the Postgres database to check for any changes to the schedules or assets. If a change is detected, the Schedule Monitor will trigger a restart of the Dagster service. This allows for runtime updates of schedules and assets without requiring a full redeployment of the Dagster service.
This means that any changes to the schedule or asset definitions require a restart of the Dagster service, triggered by this component. In the end, the Dagster service will be running with the latest schedule and asset definitions, and the user code image need not be updated.

3.2. Technical Specifications

Schedule Monitor is simple a Python script with sleep loop that checks the PostgreSQL database for any changes to the schedules or assets. If a change is detected, it will trigger a restart of the Dagster service. The script uses 'psycopg2[13]' to connect to the Postgres database and 'kubernetes[14]' to trigger the restart of the Dagster service. The script is run as a sidecar container in the same Kubernetes pod as the Dagster service. The script is configured to run periodically, and the period can be changed by modifying the sleep time in the script.



3.3. Configuration Details

[13] https://pypi.org/project/psycopg2/
[14] https://pypi.org/project/kubernetes/

The Schedule Monitor component is configured using environment variables. The following environment variables are required:

```
# DB
DATABASE_IP=
DATABASE_PORT=
DATABASE_USER=
DATABASE_NAME=
DATABASE_PASSWORD=

# Polling period
MONITOR_PERIOD=30

# Namespace of Dagster services in cluster
NAMESPACE=
```

3.4. Deployment Information

The Schedule Monitor is deployed as a sidecar to the Dagster Daemon pod as indicated in the helm chart values file [15]of the Dagster. It is possible to deploy as a standalone resource but as daemon and this service share similar resources, it was chosen to deploy as a sidecar.

```
      extraContainers:
        - name: dagster-schedule-monitor
          image: "acrdisco.azurecr.io/dagster-schedule-
monitor:0.2.15"
          imagePullPolicy: Always
          envFrom:
            - configMapRef:
                name: disco-user-code-config
          env:
            - name: DATABASE_PASSWORD
              valueFrom:
                secretKeyRef:
                  name:
disco.timescaledb.credentials.postgresql.acid.zalan.do
                  key: password
```

[15] https://github.com/dagster-io/dagster/blob/master/helm/dagster/values.yaml#L1268

4. Dagster ETL Component

4.1. Component Description

Dagster[16] is a data orchestrator that makes it easy to build scalable data pipelines. It allows you to define the data pipelines as a series of tasks that are executed in a specific order. This makes it easy to build complex data pipelines that can be easily scaled and maintained.
It is useful for building ETL (Extract, Transform, Load) pipelines that are used to extract data from various sources, transform it into a format that is suitable for analysis, and load it into a data warehouse or other storage system .
It has been considered as an alternative for Apache Airflow[17] or Prefect[18], which are other popular data orchestrators. Dagster has a more modern architecture and has a lower learning curve compared to others.

It uses Python to define the data pipelines. You can define the tasks as Python functions and then use the Dagster API to define the dependencies between the tasks.
It has a rich set of built-in libraries that make it easy to interact with various data sources and storage systems. It also has a web-based UI that allows you to monitor the progress of the data pipelines and view the logs of the tasks.

4.2. Pipeline Architecture

Pipelines in Dagster are defined as a series of operations that are executed in a specific order. Each task is defined as a python function that takes inputs and produces outputs. The tasks can be connected to form a data pipeline. These operations can be defined using the @asset[19] decorator. It is possible to connect one operation to multiple other operations, allowing for complex data pipelines to be built. For additional information, please refer to the official documentation[16].

16 https://docs.dagster.io/
17 https://airflow.apache.org/
18 https://www.prefect.io/
19 https://docs.dagster.io/guides/build/assets/

```
@asset
def knips_circulatieplan_gent(context: AssetExecutionContext) ->
list[Cut]:
    url =
"https://data.stad.gent/api/explore/v2.1/catalog/datasets/knips-
circulatieplan-gent"

    try:
        context.log.info(f"Fetching {url}")
        json_data = requests.get(url).json()

        response = CutsResponse(cuts=[Cut(**data) for data in
json_data])

        context.log.info(
            "Queried %s results for cuts spots in gent",
            len(response.cuts),
        )

        return response.cuts

    except Exception as e:
        context.log.error(e)
        raise e
```

In Dagster, assets can be chained together to form a directed acyclic graph (DAG) of dependencies[20].

```
# Example of defining assets with dependencies
@asset
def asset_a():
    return "Data from asset A"

@asset
def asset_b(asset_a):
    return f"Data from asset B, dependent on {asset_a}"

@asset
def asset_c(asset_b):
    return f"Data from asset C, dependent on {asset_b}"
```

[20] https://docs.dagster.io/guides/build/assets/defining-assets-with-asset-dependencies

Assets can be considered as the functions that are executed in the pipeline. When a schedule is attached to an asset, it becomes a job[21]. Jobs can be scheduled to run at specific times or intervals[22].

```
cuts_schedule = ScheduleDefinition(
    job=define_asset_job(
        name="knips_circulatieplan_gent_job",
        selection="knips_circulatieplan_gent*",
    ),
    cron_schedule="0 2 * * *",  # every day at 2am
    default_status=DefaultScheduleStatus.RUNNING,
)

defs = Definitions(
    resources={
        "timescale_resource": TimescaleResource(
            database_ip=EnvVar("DATABASE_IP"),
            database_port=EnvVar("DATABASE_PORT"),
            database_user=EnvVar("DATABASE_USER"),
            database_password=EnvVar("DATABASE_PASSWORD"),
            database_name=EnvVar("DATABASE_NAME")
        ),
    },
    assets=load_assets_from_modules(cuts),
    schedules=[cuts_schedule]
)
```

Assets can be connected to other resources such as databases, storage systems, or other data sources[23]. This allows for the data to be extracted, transformed, and loaded into the desired storage system.

[21] https://docs.dagster.io/guides/build/jobs/
[22] https://docs.dagster.io/guides/automate/schedules/defining-schedules
[23] https://docs.dagster.io/guides/build/external-resources/

The data pipelines can be monitored using the web-based UI[24] that is provided by Dagster. This allows you to view the progress of the data pipelines, view the logs of the tasks, and monitor the performance of the data pipelines. All these sources add up to be a definition.

4.3. Deployment

Dagster deployment consists mainly of three components[25]: the Dagster daemon, Dagster web server and the user code that defines the data pipelines.

[24] https://docs.dagster.io/guides/operate/webserver
[25] https://docs.dagster.io/deployment/dagster-instance#k8srunlauncher

Minimally 2 images are enough for a standard setup: one image for daemon and web server, another for the user code. Daemon will be using gRPC to run the user code that is served from the container[26]. It is possible to create more user code images, suitable for the use case.

```
# workspace.yaml
load_from:
  # Each entry here corresponds to a service that exposes user code.
  - grpc_server:
      host: disco_user_code
      port: 4000
      location_name: "app/etl"
```

It is possible to tell the Dagster daemon where to find the user code from the 'workspace.yaml' file[27]. If more user codes are required, they need to be referenced here.

The Dagster daemon is responsible for executing the tasks in the data pipelines, while the dagster web server provides a web-based UI to monitor the progress of the data pipelines. The user code is responsible for defining the data pipelines and the tasks that are executed in the data pipelines.

It is possible to use Kubernetes Helm charts[28] to deploy the Dagster daemon and the Dagster web server. This allows for the deployment to be easily scaled and managed.

[26] https://docs.dagster.io/guides/deploy/code-locations/workspace-yaml#running-your-own-grpc-server
[27] https://docs.dagster.io/guides/deploy/code-locations/workspace-yaml
[28] https://docs.dagster.io/guides/deploy/deployment-options/kubernetes/deploying-to-kubernetes

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta2
kind: HelmRelease
metadata:
  name: dagster
spec:
  releaseName: dagster
  chart:
    spec:
      chart: dagster
      version: 1.10.5
  values:

    dagsterWebserver:
      image:
        repository: "acrdisco.azurecr.io/dagster"
        tag: 0.2.13
      envConfigMaps:
        - name: dagster-config

    dagsterDaemon:
      image:
        repository: "acrdisco.azurecr.io/dagster"
        tag: 0.2.13
      envConfigMaps:
        - name: dagster-config
        - name: disco-user-code-config
      extraContainers:
        - name: dagster-schedule-monitor
          image: "acrdisco.azurecr.io/dagster-schedule-
monitor:0.2.15"

    dagster-user-deployments:
      deployments:
        - name: disco-user-code
          image:
            repository: "acrdisco.azurecr.io/dagster-user-code"
            tag: 0.2.13
          dagsterApiGrpcArgs:
            - "-m"
            - "disco_user_code"
          port: 4000
          envConfigMaps:
            - name: dagster-config
            - name: disco-user-code-config
```

The user code being a separate image allows it to be referenced from the Dagster instance and can be easily updated without affecting the deployment of the Dagster daemon and the web server.

Dagster uses a PostgreSQL included in the Helm chart[29] for jobs and assets history data persistence.

[29] https://github.com/dagster-io/dagster/blob/fb16ea1f8cfb2a7c910d688bcdd138393588dde3/helm/dagster/values.yaml#L799

5. DISCOLLECTION API Server

5.1. Component Description

The DISCOLLECTION API server is a RESTful API that provides access to the DISCOLLECTION data. It is designed to be lightweight and efficient. The API server allows users to query the DISCOLLECTION data using various parameters, including date range, location, and other filters. It also supports pagination and sorting of results, making it easy to retrieve large datasets in manageable chunks.
The API server is designed to be easy to use and integrate with Sovity Data Space Connectors. It provides a HTTP interface for accessing the DISCOLLECTION data.

5.2. Technical Specifications

It is built using the FastAPI[30] framework and Python programming language. It integrates with the PostgreSQL database and with Minio Object Store[31] that is also used to store the DISCOLLECTION data, which was processed and transformed by the ETL pipeline.
Follows OpenAPI[32] standards for API documentation and provides a Swagger UI[33] for easy exploration of the API endpoints. The API server is designed to be stateless and can be easily scaled horizontally to handle increased load.

30 https://fastapi.tiangolo.com/
31 https://min.io/
32 https://www.openapis.org/
33 https://swagger.io/

apds

GET `/v1/apds/places` List and search for elements in the Place hierarchy

tuld

GET `/v1/tuld/shops/` Get Shops

GET `/v1/tuld/warehouses/` Get Warehouses

GET `/v1/tuld/lockers/` Get Lockers

datex2

GET `/v1/datex2/cuts` Get Cuts

GET `/v1/datex2/loading-unloading` Get Loading Unloading

GET `/v1/datex2/sectors` Get Sectors

GET `/v1/datex2/low-emission-zones` Get Low Emission Zones

s3

GET `/v1/s3/` Get Object

From the given query params, retrieve an object from S3. This is generic endpoint to retrieve any object from S3.

Parameters Try it out

Name	Description
bucket * required `string` *(query)*	The S3 bucket name bucket
object_name * required `string` *(query)*	The S3 object name object_name

5.3. Configuration Details

Environment variables are used to configure the API server.

```
# Comment out to enable CORS for all origins
# BACKEND_CORS_ORIGINS="http://localhost:3000"

# Postgres
PG_SERVER=localhost
PG_PORT=5432
PG_DB=disco
PG_USER=disco
PG_PASSWORD=password

# Minio
S3_SERVER=http://localhost:9000
S3_ACCESS_KEY=
S3_SECRET_KEY=
```

5.4. Deployment Information

For deployment, the API server can be run using Docker. A chart is provided for deploying the API server on Kubernetes. The chart includes configmap, deployment, and service resources for the API server. API server does not necessarily have to be publicly accessible, but it needs to be accessed through the Sovity Data Space Connector. The API server can be deployed in a private network and accessed securely through the connector.

```yaml
apiVersion: helm.toolkit.fluxcd.io/v2beta1
kind: HelmRelease
metadata:
  name: disco-api
spec:
  releaseName: disco-api
  interval: 5m
  chart:
    spec:
      chart: disco-api
      version: 0.1.25
      sourceRef:
        kind: HelmRepository
        name: acrdisco-charts
        namespace: flux-system
  values:
    environment:
      PG_SERVER: "timescaledb.etl"
      S3_SERVER: "http://minio"
    secrets:
      - name: PG_PASSWORD
        secret:
disco.timescaledb.credentials.postgresql.acid.zalan.do
        key: password
      - name: S3_ACCESS_KEY
        secret: minio-client
        key: access-key
      - name: S3_SECRET_KEY
        secret: minio-client
        key: secret-key
```

6. Sovity Connectors

6.1. Component Description

The Sovity Connector is chosen to be the Data Space connector implementation for DISCO, to provide a standardized way to share data assets securely and efficiently. It implements the Eclipse Dataspace Connector (EDC)[34] framework, enabling organizations to publish, share, and consume data across different platforms while maintaining data sovereignty.

6.2. Technical Specifications

It is designed to support various data models, and provides REST API endpoints for managing data assets, establishing data contracts, and initiating data transfers. The Sovity Connector extends EDC connector that uses Java, with support for containerization using Docker and orchestration with Kubernetes. The dependencies include PostgreSQL for metadata storage and Keycloak for secure access control.

6.3. Configuration Details

It is highly configurable through environment variables, allowing customization of database connections, API keys, security tokens, and other settings. The Sovity Connector provides API endpoints for managing data assets, establishing data contracts, and initiating data transfers. Protected by OAuth2, it ensures secure access to APIs and data assets, with support for TLS encryption for data in transit. Protocol support includes IDS for secure data exchange[35], with fine-grained access control policies and comprehensive logging for auditing purposes.

A standard connector configuration looks like this:

---

[34] https://projects.eclipse.org/projects/technology.edc
[35] https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol

```
MY_EDC_PARTICIPANT_ID=${connector}
MY_EDC_TITLE=${connector} Connector
MY_EDC_DESCRIPTION=Extended Sovity Community Edition EDC Connector
MY_EDC_CURATOR_URL=https://www.imec-int.com/en
MY_EDC_CURATOR_NAME=imec
MY_EDC_MAINTAINER_URL=https://www.imec-int.com/en
MY_EDC_MAINTAINER_NAME=imec
MY_EDC_FQDN=${connector}.${domain}
MY_EDC_JDBC_URL=jdbc:postgresql://postgres.${connector}:5432/edc
MY_EDC_JDBC_USER=edc
MY_EDC_PROTOCOL=https://
EDC_DSP_CALLBACK_ADDRESS=https://${connector}.${domain}/api/dsp
EDC_DATAPLANE_TOKEN_VALIDATION_ENDPOINT=https://${connector}.${domai
n}/api/control/token
EDC_WEB_REST_CORS_ENABLED=true
EDC_WEB_REST_CORS_HEADERS=origin,content-
type,accept,authorization,x-api-key
EDC_WEB_REST_CORS_ORIGINS=*
EDC_AGENT_IDENTITY_KEY=client_id
EDC_OAUTH_TOKEN_URL=${daps}/realms/DAPS/protocol/openid-
connect/token
EDC_OAUTH_PROVIDER_AUDIENCE=${daps}/realms/DAPS
EDC_OAUTH_PROVIDER_JWKS_URL=${dataspace-
manager}/api/v1/daps/certs/jwks
EDC_OAUTH_TOKEN_EXPIRATION=15
EDC_OAUTH_CLIENT_ID=${connector}
EDC_OAUTH_CERTIFICATE_ALIAS=${connector}
EDC_OAUTH_PRIVATE_KEY_ALIAS=${connector}
WEB_HTTP_PUBLIC_PATH=/public
WEB_HTTP_PUBLIC_PORT=12001
```

Each connector requires a keystore file where the private key is stored. The key will be used for identification purposes, via DAPS[36], which is a customized Keycloak implementation as an identity governance.

DAPS follows similar configuration as a Keycloak instance:

```
KC_HOSTNAME=daps.${domain}
KEYCLOAK_ADMIN=admin
QUARKUS_HTTP_ACCESS_LOG_ENABLED=true
KC_HTTP_ENABLED=true
KC_DB=postgres
KC_DB_URL_HOST=postgres
KC_DB_URL_DATABASE=keycloak
KC_DB_SCHEMA=public
KC_DB_USERNAME=keycloak
KC_PROXY=edge
```

[36] https://github.com/sovity/sovity-daps

And a user interface is also provided for easily interact with the connector. Required configurations are:

```
EDC_UI_ACTIVE_PROFILE=sovity-open-source
EDC_UI_CONFIG_URL=edc-ui-config
EDC_UI_MANAGEMENT_API_URL=https://${connector}.${domain}/api/managem
ent
EDC_UI_CATALOG_URLS=https://${connector}.${domain}/api/dsp
NGINX_ACCESS_LOG=off
```

Most important are the 'management url' and (if set – management api key), to be able to use the connector management API.

6.4. Deployment Information

Packages are available for Docker images for Kubernetes deployment, simplifying the setup and scaling of the Sovity Connector. The deployment process involves either fetching prebuilt Docker images or cloning the community edition GitHub repository[37] and building your own images and configuring environment variables for secure operation.
Community editions are available for testing and development purposes, with enterprise editions offering additional features and support options.
Releases[38] can be used for community edition releases. DISCOLLECTION uses v10.4.2[39] with extended features[40].

A standard deployment of a connector includes the connector, connector UI and a PostgreSQL database. Images are referenced from deployment manifests.

[37] https://github.com/sovity/edc-ce/
[38] https://github.com/sovity/edc-ce/releases
[39] https://github.com/sovity/edc-ce/releases/tag/v10.4.2
[40] https://github.com/imec-int/edc-extensions/pkgs/container/edc-extensions%2Fsovity-xce/354863169?tag=10.4.2

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: connector-ui
spec:
  selector:
    matchLabels:
      app: connector-ui
  template:
    metadata:
      labels:
        app: connector-ui
    spec:
      containers:
        - name: connector-ui
          image: ghcr.io/sovity/edc-ui:4.1.8
          envFrom:
            - configMapRef:
                name: connector-ui-config
          env:
            - name: EDC_UI_MANAGEMENT_API_KEY
              valueFrom:
                secretKeyRef:
                  key: api-auth-key
                  name: connector-secrets
          ports:
            - containerPort: 8080
              name: http
```

For the connector, it is important to have the DSP and the public port accessible from external resources, as they will be used during the negotiation and data transfer between connectors. Management port can be made private, if UI connection is not required. Keystore files should be mounted as volumes.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: connector
spec:
  selector:
    matchLabels:
      app: connector
    spec:
      containers:
        - name: connector
          image: acrdisco.azurecr.io/sovity-xce:latest
          imagePullPolicy: Always
          envFrom:
            - configMapRef:
                name: connector-config
          env:
            - name: EDC_KEYSTORE
              value: keystore.p12
            - name: EDC_KEYSTORE_PASSWORD
              valueFrom:
                secretKeyRef:
                  key: keystore-password
                  name: connector-secrets
                  optional: true
            - name: EDC_API_AUTH_KEY
              valueFrom:
                secretKeyRef:
                  key: api-auth-key
                  name: connector-secrets
            - name: MY_EDC_JDBC_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: edc.postgres.credentials.postgresql.acid.zalan.do
                  key: password
          ports:
            - containerPort: 12001
              name: http-public
            - containerPort: 11001
              name: http
            - containerPort: 11002
              name: management
            - containerPort: 11003
              name: protocol
            - containerPort: 11004
              name: control
          volumeMounts:
            - name: connector-keystore
              mountPath: /app/keystore.p12
              subPath: keystore.p12
              readOnly: true
      volumes:
        - name: connector-keystore
          secret:
            secretName: certs
            items:
              - key: keystore.p12
                path: keystore.p12
```

And the for the DAPS:

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: daps
spec:
  selector:
    matchLabels:
      app: daps
  template:
    metadata:
      labels:
        app: daps
    spec:
      containers:
        - name: daps
          image: ghcr.io/sovity/sovity-daps:sha-32a3033
          envFrom:
            - configMapRef:
                name: daps-config
          env:
            - name: KEYCLOAK_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  key: admin-password
                  name: daps-secrets
            - name: KC_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name:
keycloak.postgres.credentials.postgresql.acid.zalan.do
                  key: password
          ports:
            - containerPort: 8080
              name: http
```

6.5. Security Considerations

To utilize authentication and authorization mechanisms, the Sovity Connector integrates with OAuth2 for secure access control. One custom component is the Sovity DAPS which is Sovity's implementation of the Dynamic Attribute Provisioning Service based on the Keycloak IAM platform. It acts as a central identity provider and attribute authority for the Sovity Connector, providing fine-grained access control policies and user management capabilities.
Connectors obtain certificates from Sovity DAPS for authentication and authorization with other Sovity Connectors. These certificates add custom claims to JWT tokens used in this process. Invalid tokens will cause protocol requests between connectors to fail for security reasons.

6.6. Setup Instructions

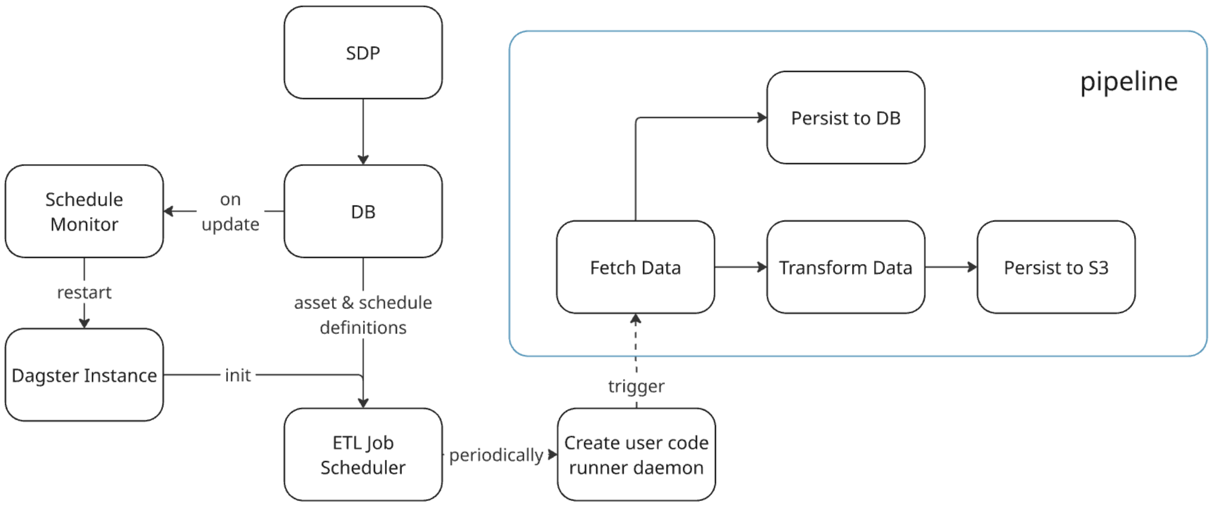
The Sovity Connector can be set up by following these steps:

- Deploy 'sovity-daps'[41]: This will automatically create a 'DAPS' realm on first initialization.
- Login with admin, and create clients and users and generate certificates in the Sovity DAPS for authentication and authorization[42]: Create clients with matching names with the connector IDs. Select 'Signed JWT' from Credentials and from Keys, 'Generate new keys' and save them later to be used as Keystore in the connector.
- Connector docker image for deployment[43] is a customized version referencing `sovity/edc-ce:10.4.2`, the difference being the addition of HTTP Proxy transfer support in the build. This allows the users to be able to use the extra transfer method besides the standard HTTP Push (see transfer methods[44]).
- Configure environment variables for database connections, API keys, and security tokens[45]
- Deploy the Sovity Connector ensuring proper network and storage configurations.
- Register data assets, establish data contracts, and initiate data transfers using the API endpoints provided by the Sovity Connector.

For a more detailed started guide, please check official documentation[46].

[41] https://github.com/sovity/sovity-daps/pkgs/container/sovity-daps/228065319?tag=sha-32a3033
[42] https://github.com/sovity/sovity-daps?tab=readme-ov-file#realm-configuration
[43] https://github.com/imec-int/edc-extensions/tree/feat/sovity-ce-http-proxy
[44] https://github.com/imec-int/edc-extensions/blob/feat/sovity-ce-http-proxy/docs/getting-started/documentation/data-transfer-methods.md
[45] https://github.com/sovity/edc-ce/tree/main/docs/deployment-guide/goals/production
[46] https://github.com/sovity/edc-ce/tree/main/docs/getting-started

7. Integration and Data Flow



Data Flow Pipeline

Preprocessing: Whenever a user code is added, the Dagster instance will schedule a new pipeline run based on configuration. It will trigger the creation of a new Dagster daemon that will run the operation per scheduled process. This allows for parallel processing of multiple pipelines separately. Essentially this is equivalent to a `Job` in Kubernetes.

There are some datasets that are important for DISCO such as parking availabilities and city access rules.

7.1 Straatparkeerplaatsen Gent:

This dataset[47] shows the on-street parking spaces in Ghent. Job schedule is configurable and for now it is configured to run every midnight.

Data Ingestion:
Dataset is not updated regularly, so it was deemed unnecessary to persist the data in a way that allows for historical analysis. Any query will update the data in the database.

[47] https://data.stad.gent/explore/dataset/straatparkeerplaatsen-gent

One example record looks like this:

```
{
  "geometry":{
    "type":"Feature",
    "geometry":{
      "coordinates":[...],
      "type":"Polygon"
    },
    "properties":{
    }
  },
  "pregime":"4",
  "opmerkingen":null,
  "capaciteit":2,
  "datum":"2024-05-06T22:00:00Z",
  "straatcode":71840,
  "pcapcode":4286,
  "wijkcode":null,
  "ptype":400,
  "zone":"TARGROEN",
  "geo_point_2d":{
    "lon":3.7450704829301436,
    "lat":51.03632127319923
  }
}
```

There is a limitation of the records returned for Open Data Portaal API, which is 100. There are 19842 records in total, so either the data was to be fetched in chunks with query API, or the whole dataset was to be exported as JSON[48]. The latter was chosen for simplicity.

Data Transformation:
Transformation is not required for this dataset.

Data Storage:
The data is stored in a Timescale DB instance, with a similar schema to the original dataset.

[48] https://data.stad.gent/api/explore/v2.1/catalog/datasets/straatparkeerplaatsen-gent/exports/json

7.2 Real time bezetting parkeergarages Gent:

This dataset[49] shows the real-time occupancy of parking garages in Ghent. Job schedule is configurable and for now it is configured to run every hour from the SDP.

Data Ingestion:

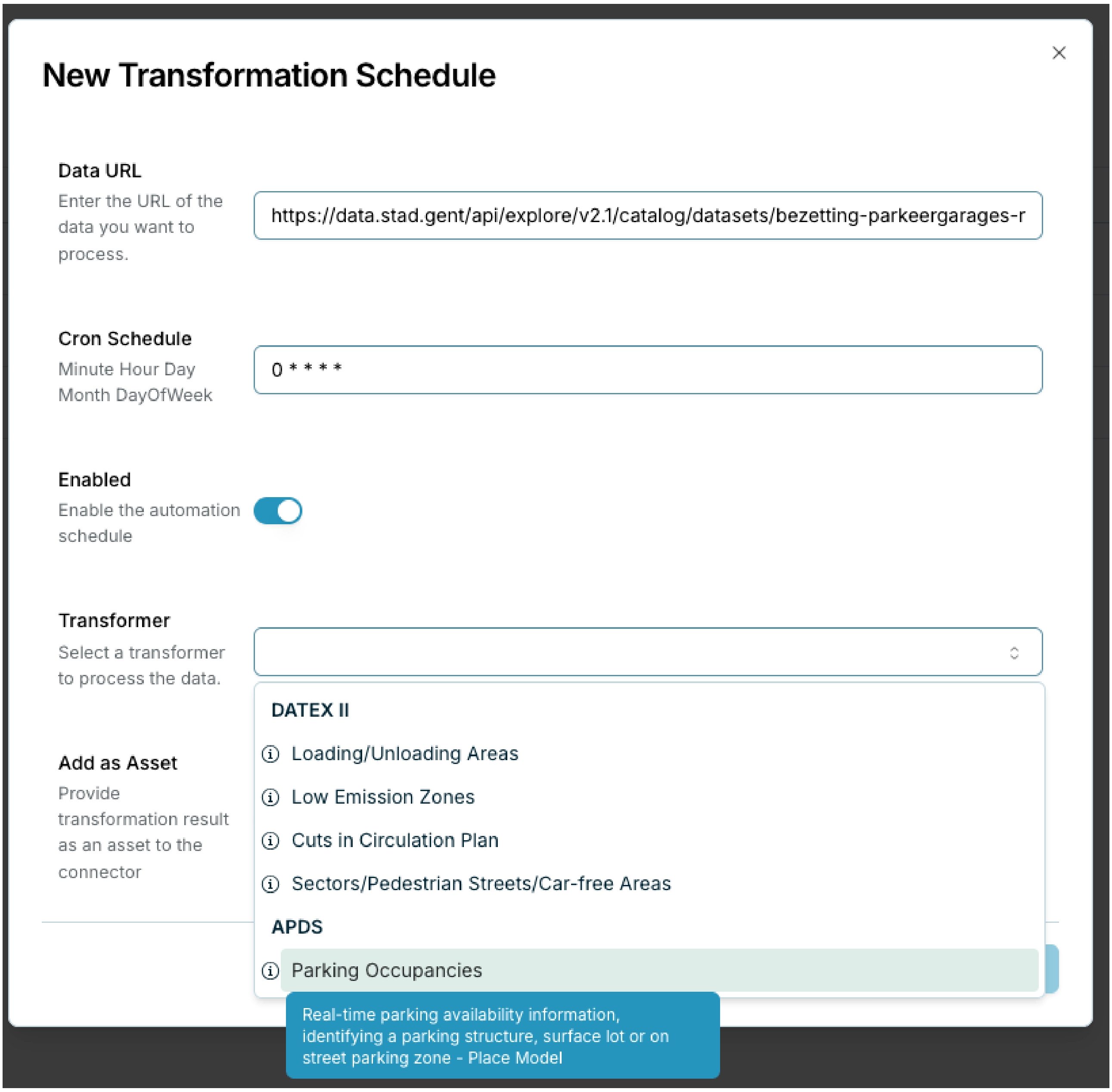
Visit the Open Data Portaal's API details page[50]. There should be the URL of the API call available:

URL of the API call

/api/explore/v2.1/catalog/datasets/bezetting-parkeergarages-real-time/records?limit=20

We will be using this URL as the input in the SDP platform[51]. Create a new transformation schedule with given details. Be sure to select the suitable transformer. As this is a Parking type dataset, APDS format is suitable. SDP will orchestrate the creation of the ETL pipeline where it will schedule a periodic job that fetches the data from the given URL, transforms it and persists the formatted results for queries. It is also possible to view the pipeline details from Dagster web UI[52].

49 https://data.stad.gent/explore/dataset/bezetting-parkeergarages-real-time/information/
50 https://data.stad.gent/explore/dataset/bezetting-parkeergarages-real-time/api/
51 https://sdp.disco.ai-data.imec.be
52 https://etl.disco.ai-data.imec.be/



Dataset is updated in every minute, so it is important to persist the data in a way that allows for historical analysis. Any query will append the data with query timestamp in the database.

One example record looks like this:


```
{
  "name":"Vrijdagmarkt",
  "lastupdate":"2024-07-30T10:00:31+02:00",
  "totalcapacity":593,
  "availablecapacity":452,
  "occupation":23,
  "type":"carPark",
  "description":"Ondergrondse parkeergarage Vrijdagmarkt in Gent",
  "id":"https://stad.gent/nl/mobiliteit-openbare-
werken/parkeren/parkings-gent/parking-vrijdagmarkt",
  "openingtimesdescription":"24/7",
  "isopennow":1,
  "temporaryclosed":0,
  "operatorinformation":"Mobiliteitsbedrijf Gent",
  "freeparking":0,
  "urllinkaddress":"https://stad.gent/nl/mobiliteit-openbare-
werken/parkeren/parkings-gent/parking-vrijdagmarkt",
  "occupancytrend":"unknown",
  "locationanddimension":"{"specificAccessInformation": ["inrit"],
"level": "0", "roadNumber": "?", "roadName": "Vrijdagmarkt 1\n9000
Gent", "contactDetailsTelephoneNumber": "Tel.: 09 266 29
00\n(permanentie)\nTel.: 09 266 29 01\n(tijdens kantooruren)",
"coordinatesForDisplay": {"latitude": 51.05713405953412,
"longitude": 3.726071777876147}}",
  "location":{
    "lon":3.726071777876147,
    "lat":51.05713405953412
  },
  "text":null,
  "categorie":"parking in LEZ",
  "dashboard":"True"
}
```


Same limitation for the API queries exists for this dataset, however as the total count is generally 10-13 records, it is acceptable to use the API[53] instead of exports.

Data Transformation:
Conversion to APDS standards will be applied, as this dataset is a parking information type[54]. For more detailed information on transformations please refer to the **DISCO Data Conversion Documentation.**

[53] https://data.stad.gent/api/explore/v2.1/catalog/datasets/bezetting-parkeergarages-real-time/records
[54] https://gitlab.com/disco-horizon-europe/open-software-repository/discollection-etl/-/blob/main/app/etl/assets/apds/transform.py?ref_type=heads

Data Storage:
As this dataset is updated very frequently, and as the Open Data Portaal does not allow for historical queries; as an alternative, a TimescaleDB instance is used to store this time series data allowing for efficient querying and storage of large amounts of time series data. The dataset is first stored in a table resembling the original payload.

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
CREATE TABLE IF NOT EXISTS parking_occupancies_gent
(
    id                        TEXT        NOT NULL,
    name                      TEXT        NOT NULL,
    lastupdate                TIMESTAMPTZ NOT NULL,
    totalcapacity             INT,
    availablecapacity         INT,
    occupation                INT,
    type                      TEXT,
    description               TEXT,
    openingtimesdescription TEXT,
    isopennow                 INT,
    temporaryclosed           INT,
    operatorinformation       TEXT,
    freeparking               INT,
    urllinkaddress            TEXT,
    occupancytrend            TEXT,
    locationanddimension      JSONB,
    location                  JSONB,
    text                      TEXT,
    categorie                 TEXT,
    dashboard                 TEXT
);

SELECT create_hypertable('parking_occupancies_gent', 'lastupdate',
if_not_exists => TRUE);
CREATE INDEX IF NOT EXISTS idx_occupancy_id_lastupdate ON
parking_occupancies_gent (id, lastupdate DESC);
```

The transformed data also is stored in another table, which is APDS compliant. This table is used to query the ADPS data.

```
CREATE TABLE IF NOT EXISTS apds_parking_occupancies_gent
(
    id                              TEXT        NOT NULL,
    version                         INT         NOT NULL,
    timestamp                       TIMESTAMPTZ NOT NULL,
    name                            JSONB       NOT NULL,
    description                     JSONB,
    layer                           INT,
    type                            TEXT,
    aliases                         JSONB,
    parent_id                       JSONB,
    child_ids                       JSONB,
    operator_defined_reference      JSONB,
    hierarchy_element_record        JSONB,
    right_specifications            JSONB,
    hierarchy_element_reference     JSONB,
    occupancy_level                 JSONB,
    time_zone                       TEXT,
    place_street_address            JSONB,
    place_bounded_zone              JSONB,
    indicative_place_point_location JSONB,
    characteristics                 JSONB,
    contacts                        JSONB,
    marketing                       JSONB,
    operating_restrictions          JSONB,
    rgb_colours                     JSONB,
    times                           JSONB,
    payment_methods                 JSONB,
    safety_standard_classifications JSONB
);

SELECT create_hypertable('apds_parking_occupancies_gent',
'timestamp', if_not_exists => TRUE);
CREATE INDEX IF NOT EXISTS idx_occupancy_id_timestamp ON
apds_parking_occupancies_gent (id, timestamp DESC);
```

With two tables in place, both the historical original data and regularly transformed data are always available. Both have hyper tables created for efficient querying and storage based on timestamp. For a more automated approach, it is possible to create Dagster operations to initialize the TimescaleDB schemas as well.

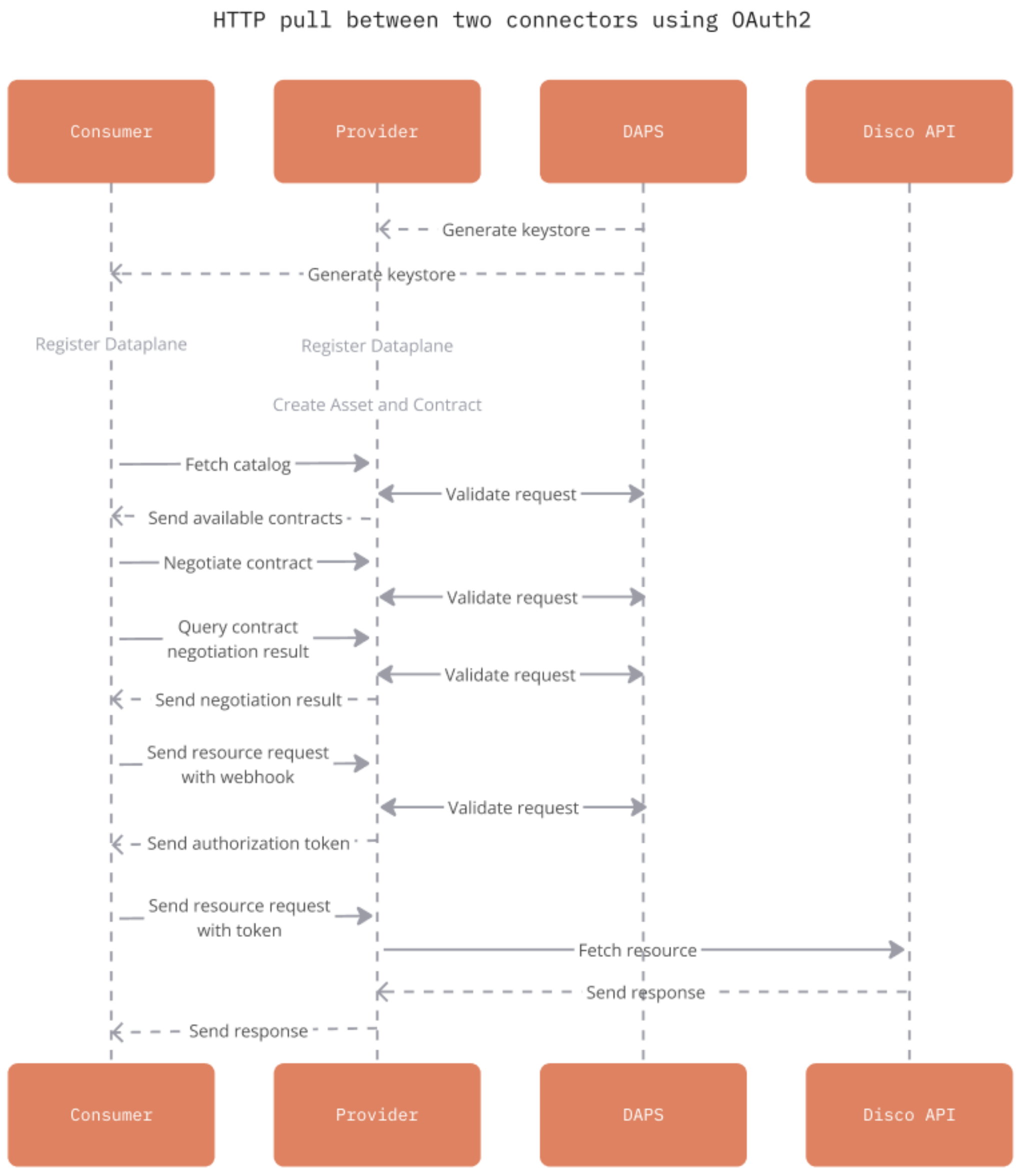
7.3 Connectors

For DISCO data space, we'll be using an extended version (XCE)[55] of the Sovity Community Edition (SCE) connector, forked from their original repository. The SCE extends the base connector with

[55] https://github.com/imec-int/edc-extensions/tree/feat/sovity-ce-http-proxy

some default policies, a user interface, and more. For more information, see the available extensions[56]. The XCE adds support for the Consumer Pull transfer mode[57]. It is important to note Sovity's Connectors (v10.x) are dependent on an older version of the EDC Connector (v0.2.1.4)[58]. When using and EDC Connector, one can choose between two flows when performing a data transfer[59]: Consumer Pull or Provider Push. The Consumer Pull flow allows the Consumer to query the asset's data source through a proxy on the Provider side. This allows for synchronous HTTP-based data transfers that return the data immediately to the Consumer.
Alternatively, the Provider Push flow provides an asynchronous way to transfer data whereby the Provider "pushes" data from the asset's data source to the Consumer's data sink. This method supports more types of data sinks, such as cloud storage services, FTP servers, and more.
The Consumer Pull flow is a simpler approach, allowing the Consumer to send regular HTTP requests through the Provider connector as if it was querying the data source directly.

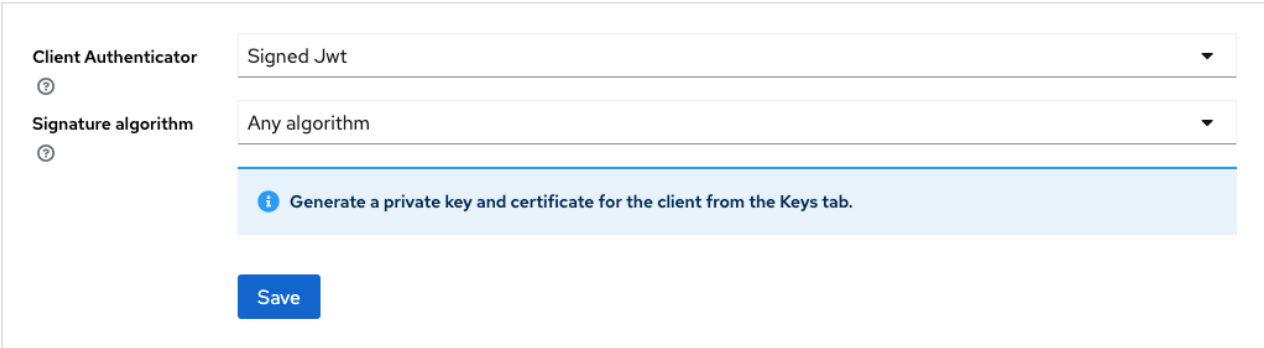
[56] https://github.com/sovity/edc-ce/tree/v10.5.1/extensions
[57] https://github.com/imec-int/edc-extensions/blob/feat/sovity-ce-http-proxy/launchers/common/base/build.gradle.kts#L38-L41
[58] https://github.com/imec-int/edc-extensions/blob/31de454e21c757e10d31f20de8e2d93f6d16103b/gradle/libs.versions.toml#L17
[59] https://github.com/imec-int/edc-extensions/blob/feat/sovity-ce-http-proxy/docs/getting-started/documentation/data-transfer-methods.md



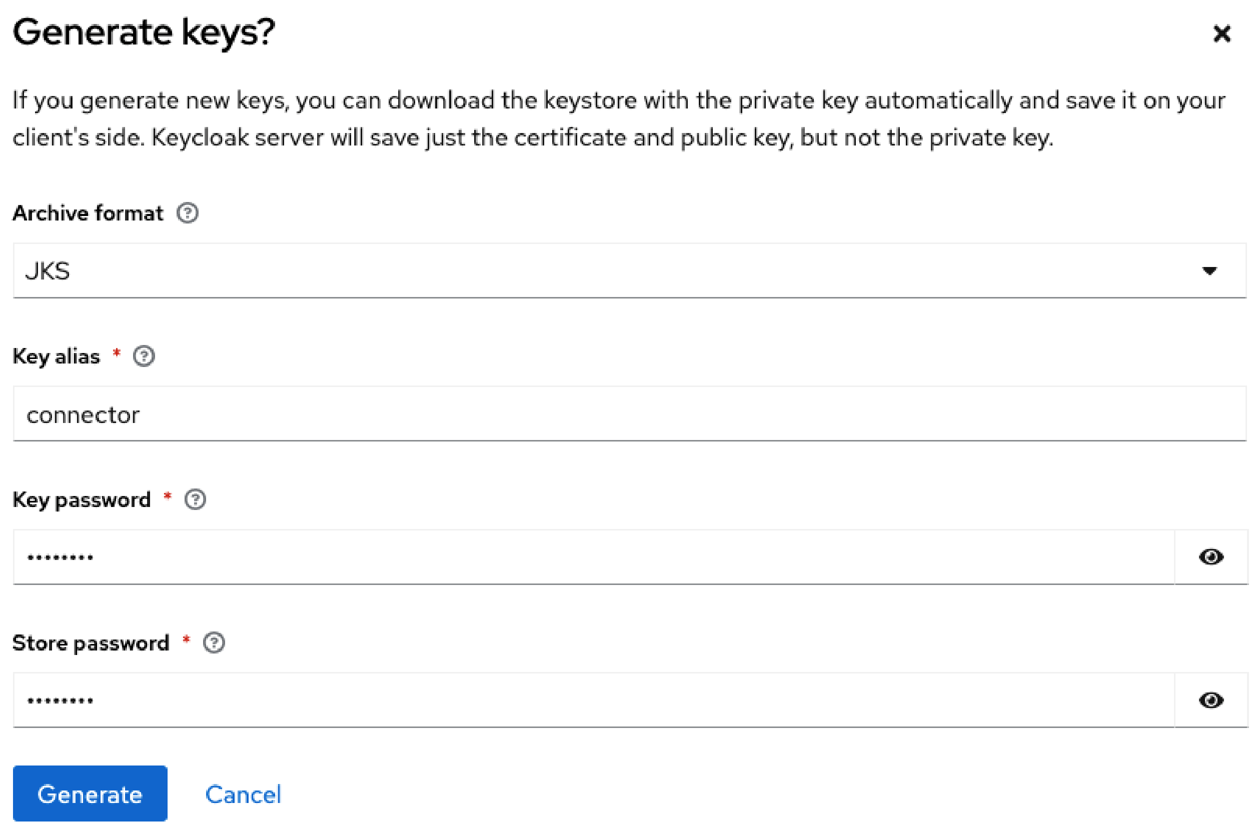
OAuth2 flow

- Create clients from DAPS instance: Be sure to enable Client authentication.

- From Credentials select Signed JWT



- Generate New private and public keys for each connector from DAPS, to be used as keystore for the connectors. Save the passwords as they will be used for connector configuration.



- Register the data planes.

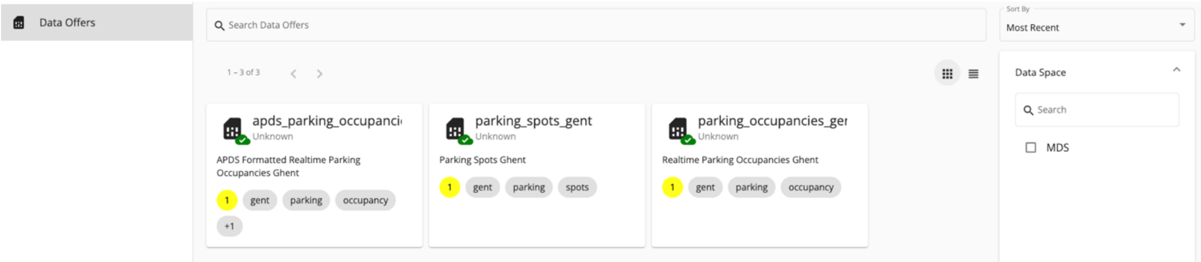
```
curl --location 'https://${connector}/api/management/v2/dataplanes'
\
--header 'Content-Type: application/json' \
--header 'x-api-key: <api-key>' \
--data-raw '{
    "@context": {
        "edc": "https://w3id.org/edc/v0.0.1/ns/"
    },
    "edctype": "dataspaceconnector:dataplaneinstance",
    "id": "http-pull-dataplane",
    "url": "https://${connector}/api/management/transfer",
    "allowedSourceTypes": [
        "HttpData"
    ],
    "allowedDestTypes": [
        "HttpData",
        "HttpProxy"
    ],
    "properties": {
        "edc:publicApiUrl": "https://${connector}/public"
    }
}'
```

- Create Assets which will point to Disco API. Disco API should not be publicly accessible so that it is only available via Provider Connector. You can either not have a public URL for the API or require Authorization for the API. It is possible to add OAuth2 client and secrets in the asset definition, so that the provider will use these while fetching the resource from a public endpoint[60] As this is not available in the community edition of the Sovity connector, the DISCO API is not made publicly available.

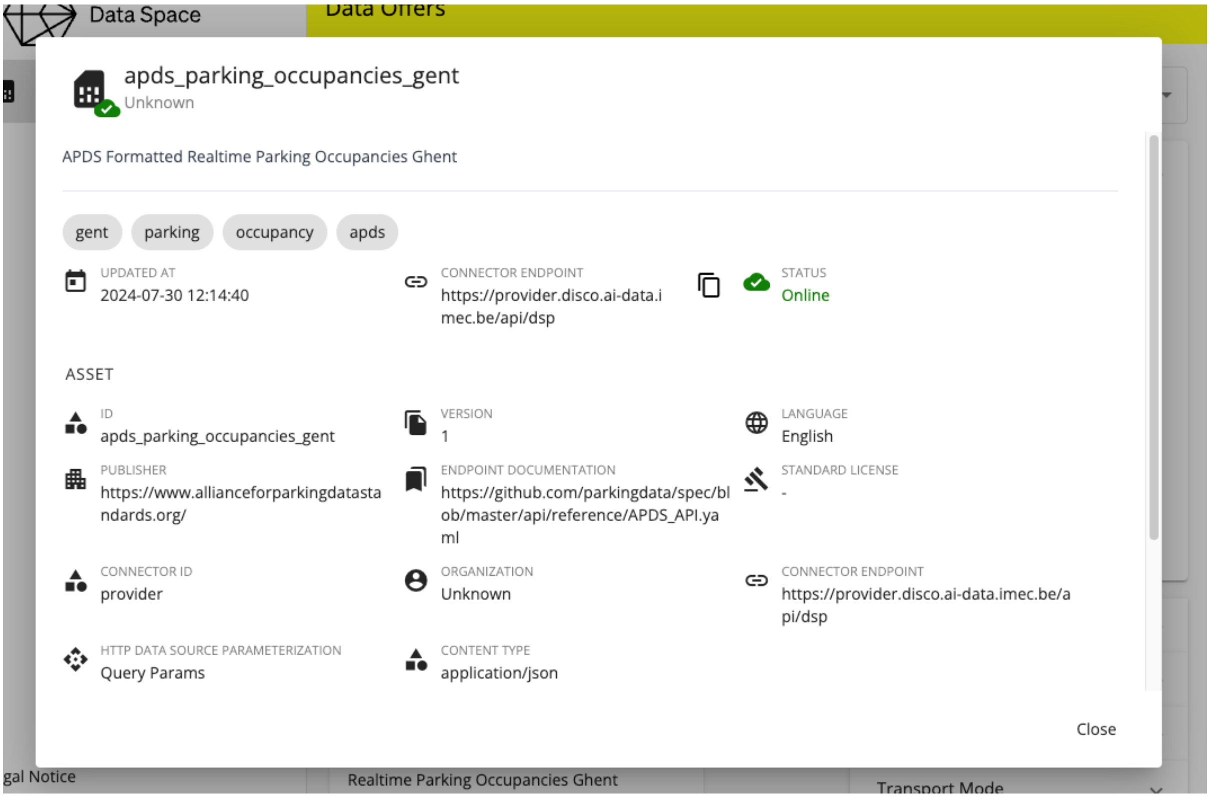
- Create Contracts for these Assets. Connectors provide Management API for these features[61] or also possible via the Connector UI[62]

- Fetch the catalog of the provider, this should return three previously created assets.

[60] https://github.com/sovity/edc-ce/blob/main/docs/getting-started/documentation/oauth-data-address.md
[61] https://github.com/sovity/edc-ce/blob/v10.4.2/docs/getting-started/documentation/parameterized_assets.md
[62] https://github.com/sovity/edc-ce/blob/v10.4.2/docs/getting-started/documentation/parameterized_assets_via_ui.md



Screenshot of Datasets



Screenshot of Datasets Catalog

- As a consumer, negotiate the contract. Once it is approved, it will be possible to start a transfer[63].
- The provider will send a token to the given webhook in the transfer request details. With that token, now it is possible to fetch the resource.

[63] https://github.com/sovity/edc-ce/blob/v10.4.2/docs/getting-started/documentation/pull-data-transfer.md

Assets are configured to have dynamic query parameters, which allows for queries to be passed to Disco API.

For example:
https://provider.disco.ai-data.imec.be/public?name=Vrijdagmarkt of the APDS dataset asset would fetch all historic records for parking spots with given name. To fetch all, any query parameters can be omitted. Even though there is only one public endpoint per connector, correct asset resolution is based on the Authorization token provided.

8. Conclusion

In conclusion, DISCOLLECTION effectively streamlines the process for cities to contribute valuable data to the data space, enhancing accessibility and integration across urban data systems. Using an architecture that includes data retrieval from the cities' open data portals, transformation into a standardized model, storage in various platforms, and dissemination through the Urban Freight Data Space connector, DISCOLLECTION not only standardizes data but also facilitates its broader use and integration with other DISCO-X solutions and existing data platforms. This approach enables cities to participate in a unified data ecosystem, promoting better urban management and planning through the efficient sharing and utilization of critical geospatial and logistical information.

9. References

Gent Open Data Portaal:
https://data.stad.gent/explore/dataset/straatparkeerplaatsen-gent/information/
https://data.stad.gent/explore/dataset/bezetting-parkeergarages-real-time/information/

Dagster:
https://dagster.io/

Sovity Community Edition EDC:
https://github.com/sovity/edc-ce

Keycloak Identity and Access Management:
https://www.keycloak.org/

APDS:
https://www.allianceforparkingdatastandards.org/
https://github.com/parkingdata/spec

imec DISCOLLECTION Repository Deliverables:
DISCOLLECTION API: https://gitlab.com/disco-horizon-europe/open-software-repository/discollection-api
DISCOLLECTION ETL: https://gitlab.com/disco-horizon-europe/open-software-repository/discollection-etl
DISCOLLECTION K8S: https://gitlab.com/disco-horizon-europe/open-software-repository/discollection-k8s
DISCOLLECTION SDP: https://gitlab.com/disco-horizon-europe/open-software-repository/discollection-sdp

imec DISCOLLECTION Deployment URLs:
Dagster User Code: https://etl.disco.ai-data.imec.be/
DAPS: https://daps.disco.ai-data.imec.be/
Broker: https://broker.disco.ai-data.imec.be/
Provider: https://provider.disco.ai-data.imec.be/
Consumer: https://consumer.disco.ai-data.imec.be/
SDP: https://sdp.disco.ai-data.imec.be/

DISCOLLECTION APIs (private network – only accessible via provider connector):

http://disco-api/v1/parking-occupancies/
Retrieve parking occupancies of a given time window.

- `gte`: filter by timestamp greater than or equals
- `lte`: filter by timestamp less than or equals

http://disco-api/v1/parking-spots/
Retrieve parking spots

http://disco-api/v1/apds/places
Lists `Places` and allows searching/filtering by:

- `latitude`, `longitude`, `radius`: to check for elements in the `Place` hierarchy within a geographic area
- a modification instant (`modified_since`): to only display changed resources from that point onwards
- `layer`: maximum hierarchy element layer desired
- `type`: types of hierarchy elements * `right_type`: types of `RightSpecifications` associated with Place hierarchy elements
- `structure_type`: types of structure
- `structure_grade`: grades of structures
- `name`: name of facilities

Additionally, supports providing a comma-separated list of optional object attributes associated to a `Place` hierarchy element that should be included in the result set. (see `expand` attribute). If unused or left blank, `none` will be assumed.

Funded by the European Union

THE CIVITAS INITIATIVE IS CO-FUNDED BY THE EUROPEAN UNION

DISCO is a project under the CIVITAS Initiative.
Read more - civitas.eu

DISCOLLECTION Data Model Conversion Documentation

-

Data-driven and Dynamic Space and Assets for Physical Internet-led Urban Logistics and Planning

DATA MODEL CONVERSION GUIDE: APDS & DATEX II

IMEC

Alper BASAK

Xueying DENG

April 2025

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101103954. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

Abstract

Data-level interoperability plays a vital role in a data space, and leveraging standard data models significantly simplifies this process. As a first proof-of-concept, this document shows how the DISCO project incorporates the APDS and DATEX II standards.

APDS (Alliance for Parking Data Standards) is an international standard designed to establish a common language for sharing parking data. It aims to promote interoperability between parking systems and services, enabling seamless data exchange across various regions and platforms.

DATEX II, on the other hand, is Europe's most widely adopted standard for road situations and traffic regulations. It provides a structured framework for sharing information related to traffic management, road conditions, and regulations, ensuring consistency and compatibility across European transportation systems.

This document outlines mapping data from its sources to these two standards. It provides an overview of the APDS and DATEX II standards, detailing their structure and practical applications. Furthermore, it includes specific data mapping examples from the city of Ghent, demonstrating how real-world data can be transformed to comply with these standards. For an in-depth explanation of the ETL (Extract, Transform, Load) technical implementation, including insights into the code and workflows used for this mapping within the framework of the Smart Data Platform, please refer to the DISCOLLECTION Design and Architecture Documentation.

Document history

Version	Date	Organisation	Main area of changes	Comments
0.9	10 April 2025	imec		First version
1.0	22 April 2025	imec	Further fill out examples and some spelling changes	

List of acronyms

EPA	European Parking Association
API	Application Programming Interface
APDS	Alliance for Parking Data Standards
DATEX II	DAta EXchange second version
EPA	European Parking Association
ETL	Extract, Transform, Load
UVAR	Urban Vehicle Access Regulation
XML	eXtensible Markup Language

1. APDS

To follow a standard for parking spaces data format, the Alliance for Parking Data Standards (APDS)[1] was chosen. It follows a strict API specification interface documented in an Open API document[2]. For more detailed info on the specification, we refer the reader to the ADPS specs folder on their GitHub repository[3].

1.1. Mapping to APDS

Matching the data to the APDS `Places` schema is crucial. Below is an example of how a single record from the city of Ghent's car parking data is mapped into the APDS following format (excluding null values):


```
{
  "name": "Vrijdagmarkt",
  "lastupdate": "2024-07-30T10:00:31+02:00",
  "totalcapacity": 593,
  "availablecapacity": 452,
  "occupation": 23,
  "type": "carPark",
  "description": "Ondergrondse parkeergarage Vrijdagmarkt in Gent",
  "id": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-
vrijdagmarkt",
  "openingtimesdescription": "24/7",
  "isopennow": 1,
  "temporaryclosed": 0,
  "operatorinformation": "Mobiliteitsbedrijf Gent",
  "freeparking": 0,
  "urllinkaddress": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-
vrijdagmarkt",
  "occupancytrend": "unknown",
  "locationanddimension": {
    "specificAccessInformation": ["inrit"],
    "level": "0",
    "roadNumber": "?",
    "roadName": "Vrijdagmarkt 1\n9000 Gent",
    "contactDetailsTelephoneNumber": "Tel.: 09 266 29 00\n(permanentie)\nTel.: 09 266 29
01\n(tijdens kantooruren)",
    "coordinatesForDisplay": {
      "latitude": 51.05713405953412,
      "longitude": 3.726071777876147
    }
  },
  "location": {
    "lon": 3.726071777876147,
    "lat": 51.05713405953412
  },
  "text": null,
```


[1] https://www.allianceforparkingdatastandards.org/
[2] https://github.com/parkingdata/spec/blob/master/api/reference/APDS_API.yaml
[3] https://github.com/parkingdata/spec/tree/master/specs

```
  "categorie": "parking in LEZ",
  "dashboard": "True"
}


{
  "id": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-
vrijdagmarkt",
  "version": 1,
  "name": [{
      "language": "en",
      "string": "Vrijdagmarkt"
    }],
  "description": [{
      "language": "en",
      "string": "Ondergrondse parkeergarage Vrijdagmarkt in Gent"
    }],
  "layer": 0,
  "type": "parkingPlace",
  "aliases": [[{
        "language": "en",
        "string": "parking in LEZ"
      }]],
  "operatorDefinedReference": [{
      "language": "en",
      "string": "Mobiliteitsbedrijf Gent"
    }],
  "hierarchyElementRecord": {
    "creationTime": "1970-01-01T00:00:00Z"
  },
  "hierarchyElementReference": {
    "elementId": {
      "id": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-
vrijdagmarkt",
      "version": 1
    },
    "demandTable": [{
        "timestamp": "2024-07-29T12:37:46Z",
        "demandType": [{
            "count": 32,
            "occupancyCalculation": "verified",
            "recordDateTime": "2024-07-29T12:36:14Z"
          }]}],
    "supply": [{
        "supplyViewType": "spaceView",
        "supplyQuantity": 401
      }]},
  "timeZone": "Europe/Brussels",
  "placeStreetAddress": [{
      "addressLines": [{
          "order": 0,
          "text": [{
              "language": "en",
              "string": "Vrijdagmarkt 1\n9000 Gent"
            }],
          "type": "street"
        },
        {
          "order": 1,
          "text": [{
              "language": "en",
              "string": "?"
            }],
          "type": "poBox"
        }]}],
  "indicativePlacePointLocation": [{
      "type": "Point",
      "coordinates": [
```

```json
        3.726071777876147,
        51.05713405953412
      ]}],
  "characteristics": [{
      "spacesTotal": 593,
      "structureType": "offStreetStructure"
    }],
  "contacts": [{
      "type": "operator",
      "contactDetails": [{
          "urlLinkAddress": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-vrijdagmarkt"
        }],
      "telephoneContacts": [{
          "localNumbers": [
            "Tel.: 09 266 29 00\n(permanentie)\nTel.: 09 266 29 01\n(tijdens kantooruren)"
          ]}],
      "id": "https://stad.gent/nl/mobiliteit-openbare-werken/parkeren/parkings-gent/parking-vrijdagmarkt",
      "version": 1
    }],
  "times": {
    "available24hours": true,
    "openAllYear": true,
    "openingTimesNotSpecified": true,
    "openingTimesUnknown": true,
    "operatingTime": {
      "isOpenNow": true,
      "temporaryClosed": false
    }}
}
```

This is a more detailed and structured format, which is more suitable for further analysis and integration with other datasets. There are more APDS schemas available, such as `Rates`, `Sessions`, `Observations`, etc., which can be used to enrich the data further; however, this is out of scope for this project.

2. DATEX II

DATEX II is a widely used referenceable data model in the European context for road traffic and travel information. The DATEX II models particularly valuable for the DISCO project are the Urban Vehicle Access Regulations (UVAR) models, as the data sources being modelled primarily fall within this scope. These include:

- Low Emission Zone
- Circulation Plan Restrictions
- Loading and Unloading Points
- Sectors, Pedestrian Streets, and Car-Free Areas

We are fortunate to have the official release of DATEX II 3.6 in April 2025, which, for the first time, officially includes UVAR models under the "TrafficRegulation" ontology. DISCO is among the pioneers in mapping these standards.

2.1. DATEX II Schemas

DATEX II provides official schemas for reference (Version 3.6 - DATEX II Docs), including their associated namespaces. Additionally, it offers a DATEX II Online Schema Generation Tool[4] which includes a step-by-step guide to help users obtain the appropriate schemas for profiling.
Using this tool, one can generate .xsd schemas required for their data modelling. For the DISCO project, the following schemas are utilized:

- DATEX II_3_Common.xsd
- DATEX II_3_CommonExtension.xsd
- DATEX II_3_ControlledZone.xsd
- DATEX II_3_D2Payload.xsd
- DATEX II_3_LocationExtension.xsd
- DATEX II_3_LocationReferencing.xsd
- DATEX II_3_Parking.xsd
- DATEX II_3_Situation.xsd
- DATEX II_3_TrafficManagementPlan.xsd
- DATEX II_3_TrafficRegulation.xsd

Python has a package called xsdata[5] which provides a code generator to generate Python data classes. However, the auto-generation can lead to circular import issues. To prevent this when generating data classes from DATEX II, while running xsdata generate, we set the --structure-style option to single-package. However, you are free to use any structure style.

2.2. Low Emission Zone Mapping to DATEX II

The generated Python `dataclasses` can be utilized to map data from its original fields to the DATEX II 3.6 format, adhering to common programming practices. As an example, we will demonstrate how to map a Low Emission Zone (LEZ). The source data is presented in JSON format as shown below (with some coordinates omitted):

```
"geometry": {
  "type": "Feature",
  "geometry": {
    "coordinates": [
      [
        [3.705964812272827, 51.06470247162831],
        ....
        [3.703127948775252, 51.06425954657684],
        [3.707356564956, 51.064941241256584],
```

[4] https://webtool.datex2.eu/wizard
[5] https://xsdata.readthedocs.io/en/latest/

```
            [3.705964812272827, 51.06470247162831]
          ]
        ],
        "type": "Polygon"
        },
        "properties": {}
    },
    "gentid": "LEZ2",
    "urid": "milieu/lez12",
    "geo_point_2d": {
    "longitude": 3.7047251774233394,
    "latitude": 51.06074695800349
    }
}
```

The LEZ corresponds to a `ControlledZoneTablePublication` in DATEX II, with fields matching those in the data class. Cross-validation or referential examples can improve mapping accuracy, as mapping heavily relies on the mapper's semantic understanding of the fields.

Table - Low Emission Zone data

Gent Open Data Portaal	DATEX II	Comments
geometry	GmlMultiPolygon. gmlPolygon	Datex II Docs: GML (Linear and area location)
gentid	UvarZone.name	
urid	ControlledZoneSummary.informal_name	No specific identifier could be found in DATEX, closest was to set this in the summary
geo_point_2d	ControlledZoneSummary.location_for_display	

The XML output is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<ControlledZoneTablePublication
    xmlns:tro="http://datex2.eu/schema/3/trafficRegulation"
    xmlns:loc="http://datex2.eu/schema/3/locationReferencing"
    xmlns:cz="http://datex2.eu/schema/3/controlledZone"
    xmlns:com="http://datex2.eu/schema/3/common"
    xmlns:d2="http://datex2.eu/schema/3/d2Payload"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" lang="nl" modelBaseVersion="3">
    <com:feedType>ControlledZoneTablePublication</com:feedType>
```

```xml
    <com:publicationTime>2025-04-07T14:49:40.502558Z</com:publicationTime>
    <com:publicationCreator>
        <com:country>BE</com:country>
        <com:nationalIdentifier>BE</com:nationalIdentifier>
    </com:publicationCreator>
    <cz:controlledZoneTable>
        <cz:tableVersionTime>2025-04-07T14:49:40.502644Z</cz:tableVersionTime>
        <cz:uvarZone>
            <cz:name>
                <com:values>
                    <com:value>LEZ1</com:value>
                </com:values>
            </cz:name>
            <cz:controlledZoneType>lowEmissionZone</cz:controlledZoneType>
            <cz:status>active</cz:status>
            <cz:controlledZoneSummary>
                <cz:cityName>
                    <com:values>
                        <com:value>Gent</com:value>
                    </com:values>
                </cz:cityName>
                <cz:informalName>
                    <com:values>
                        <com:value>milieu/lez11</com:value>
                    </com:values>
                </cz:informalName>
                <cz:locationForDisplay>
                    <loc:latitude>51.052422</loc:latitude>
                    <loc:longitude>3.725227</loc:longitude>
                </cz:locationForDisplay>
            </cz:controlledZoneSummary>
            <cz:trafficRegulationOrder>
                <tro:issuingAuthority>
                    <com:values>
                        <com:value>Stad Gent</com:value>
                    </com:values>
                </tro:issuingAuthority>
                <tro:regulationId>LEZ1</tro:regulationId>
                <tro:status>madeAndImplemented</tro:status>
                <tro:trafficRegulation>
                    <tro:condition xsi:type="tro:ConditionSet">
                        <tro:conditions xsi:type="tro:LocationCondition">
                            <tro:locationByOrder xsi:type="loc:AreaLocation">
                                <loc:gmlMultiPolygon>
                                    <loc:gmlPolygon>
                                        <loc:exterior>
                                            <loc:posList>3.729674 51.067678 3.729302
51.067686 ...</loc:posList>
                                        </loc:exterior>
                                    </loc:gmlPolygon>
                                </loc:gmlMultiPolygon>
                            </tro:locationByOrder>
                        </tro:conditions>
                        <tro:conditions xsi:type="tro:VehicleCondition">
                            <tro:vehicleCharacteristics>
                                <com:vehicleType>anyVehicle</com:vehicleType>
                            </tro:vehicleCharacteristics>
                        </tro:conditions>
                    </tro:condition>
                </tro:trafficRegulation>
            </cz:trafficRegulationOrder>
        </cz:uvarZone>
    </cz:controlledZoneTable>
</ControlledZoneTablePublication>
```

2.3. Circulation Plan Restrictions Mapping to DATEX II

The source data of Circulation Plan Restrictions is presented in JSON format as shown below (with some coordinates omitted):


```
"geometry": {
        "type": "LineString",
        "coordinates": [
        [3.744528945735961, 51.04586088010989],
        [3.7446984838812, 51.04599642959088]
        ]
},
"properties": {
        "description": "Knip middenberm nieuw "
},
"geo_point_2d": {
        "lon": 3.7446137148085805,
        "lat": 51.045928654850385
}
```


The Circulation Plan Restrictions corresponds to a `TrafficRegulationPublication` in DATEX II, with fields matching those in the data class.

Table - Circulation Plan Restrictions

Gent Open Data Portaal	DATEX II	Comments
geometry	GmlLineString.posList	DATEX II Docs: GML (Linear and area location) DATEX II_3_ControlledZone
description	TrafficRegulationPublication.feedDescription	
geo_point_2d	AreaLocation.coordinatesForDisplay	

The XML output is as follows:


```
<?xml version="1.0" encoding="UTF-8"?>
<TrafficRegulationPublication
    xmlns:tro="http://datex2.eu/schema/3/trafficRegulation"
    xmlns:loc="http://datex2.eu/schema/3/locationReferencing"
    xmlns:cz="http://datex2.eu/schema/3/controlledZone"
    xmlns:comx="http://datex2.eu/schema/3/commonExtension"
    xmlns:com="http://datex2.eu/schema/3/common"
    xmlns:d2="http://datex2.eu/schema/3/d2Payload"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    lang="nl"
    modelBaseVersion="3">
    <com:feedDescription>
        <com:values>
            <com:value lang="nl">Knippen Circulatieplan Gent</com:value>
```

```xml
        </com:values>
    </com:feedDescription>
    <com:feedType>Mobiliteit</com:feedType>
    <com:publicationTime>2025-04-07T14:37:18.605655Z</com:publicationTime>
    <com:publicationCreator>
        <com:nationalIdentifier>BE</com:nationalIdentifier>
    </com:publicationCreator>
    <tro:trafficRegulationsFromCompetentAuthorities>
        <tro:trafficRegulationOrder>
            <tro:description>
                <com:values>
                    <com:value lang="nl">Knip middenberm nieuw</com:value>
                </com:values>
            </tro:description>
            <tro:issuingAuthority>
                <com:values>
                    <com:value lang="nl">Mobiliteitsbedrijf Gent</com:value>
                </com:values>
            </tro:issuingAuthority>
            <tro:reason>trafficOrder</tro:reason>
            <tro:reason>trafficSafety</tro:reason>
            <tro:status>madeAndImplemented</tro:status>
            <tro:trafficRegulation xsi:type="cz:ControlledZoneRegulation">
                <tro:status>active</tro:status>
                <tro:condition xsi:type="tro:LocationCondition">
                    <tro:implementedLocation xsi:type="loc:LinearLocation">
                        <loc:coordinatesForDisplay>
                            <loc:latitude>51.062682643332735</loc:latitude>
                            <loc:longitude>3.703543141565572</loc:longitude>
                        </loc:coordinatesForDisplay>
                        <loc:gmlLineString>
                            <loc:posList xsi:type="GmlPosList">
                                3.703501767409373 51.062723199354345 3.703584515721771
51.06264208731113
                            </loc:posList>
                        </loc:gmlLineString>
                    </tro:implementedLocation>
                </tro:condition>
                <cz:enforcementMethodType>other</cz:enforcementMethodType>
            </tro:trafficRegulation>
        </tro:trafficRegulationOrder>
    </tro:trafficRegulationsFromCompetentAuthorities>
</TrafficRegulationPublication>
```


2.4. Loading and Unloading Points Mapping to DATEX II

The source data of Loading and Unloading Points is presented in JSON format as shown below (with some coordinates omitted):


```json
{
  "geometry": {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [3.718601223830008, 51.04165280243418],
          [3.718705750898138, 51.04174169661648],
          [3.71868284655334З, 51.04175240559054],
          [3.718661818789186, 51.041734531317346],
          [3.718578313197001, 51.04166350747502],
          [3.718601223830008, 51.04165280243418]
        ]
```

```
      ]
    },
    "properties": {}
  },
  "pregime": "Voorbehouden",
  "ptype": 815,
  "opmerkingen": null,
  "type": null,
  "capaciteit": 2,
  "lopendemeter": "12.31000000",
  "zone": "TARGEEL",
  "straatcode": 70779,
  "straatnaam": "Kortrijksesteenweg",
  "datum": "2024-05-23T22:00:00Z",
  "pcapcode": 337,
  "geo_point_2d": {
    "lon": 3.7186420329727152,
    "lat": 51.04170260468722
  }
}
```


The Loading and Unloading Points corresponds to a `TrafficRegulationPublication` in DATEX II , with fields matching those in the data class.

Gent Open Data Portaal	DATEX II	Comments
geometry	GmlMultiPolygon.gml_polygons	Datex II Docs: GML (Linear and area location)
streetnaam + straatcode	GmlMultiPolygon.gml_area_name	No specific identifier could be found in DATEX, closest was to append these values and set as the area name
opmerkingen	TrafficRegulationOrder.description	

ptype	if 814:
	StandingOrParkingControl.STANDING_AND_PARKING_PROHIBITED
	if 815:
	StandingOrParkingControl. permittedParkingTime
		• Dedicated plaatsen voor laden en lossen (Ptype 814)
		• Shop & Go parkeerplaatsen met een parkeersensor (Ptype 815)

capaciteit	PermitInformation.description	No specific parking regulation characteristics could be found for capacity metric, closest chosen

lopendemeter	PermitInformation.description	No specific parking regulation characteristics could be found for length metric, closest chosen
pregime	OtherAccessRestriction.value	
datum	LegalBasis.date	Registration date
zone	IsoNamedArea.areaName	
pcapcode	IsoNamedArea.subdivisionCode	
geo_point_2d	AreaLocation. coordinatesForDisplay	

The XML output is as follows:


```xml
<?xml version="1.0" encoding="UTF-8"?>
<TrafficRegulationPublication
    xmlns:tro="http://datex2.eu/schema/3/trafficRegulation"
    xmlns:loc="http://datex2.eu/schema/3/locationReferencing"
    xmlns:cz="http://datex2.eu/schema/3/controlledZone"
    xmlns:comx="http://datex2.eu/schema/3/commonExtension"
    xmlns:com="http://datex2.eu/schema/3/common"
    xmlns:d2="http://datex2.eu/schema/3/d2Payload"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    lang="nl"
    modelBaseVersion="3">
    <com:feedDescription>
        <com:values>
            <com:value lang="nl">Laad en losplaatsen Gent</com:value>
        </com:values>
    </com:feedDescription>
    <com:feedType>Mobiliteit</com:feedType>
    <com:publicationTime>2025-04-07T14:44:03.449622Z</com:publicationTime>
    <com:publicationCreator>
        <com:nationalIdentifier>BE</com:nationalIdentifier>
    </com:publicationCreator>
    <tro:trafficRegulationsFromCompetentAuthorities>
        <tro:trafficRegulationOrder>
            <tro:issuingAuthority>
                <com:values>
                    <com:value lang="nl">Mobiliteitsbedrijf Gent</com:value>
                </com:values>
            </tro:issuingAuthority>
            <tro:reason>trafficOrder</tro:reason>
            <tro:reason>trafficSafety</tro:reason>
            <tro:status>madeAndImplemented</tro:status>
            <tro:trafficRegulation>
                <tro:status>active</tro:status>
                <tro:typeOfRegulation xsi:type="tro:StandingOrParkingControl">
                    <tro:permittedParkingTime>PT30M</tro:permittedParkingTime>
                </tro:typeOfRegulation>
                <tro:condition xsi:type="tro:AccessCondition">
                    <tro:legalBasis>
```

```
            <tro:date>2024-05-06Z</tro:date>
        </tro:legalBasis>
        <tro:accessConditionType>loadingAndUnloading</tro:accessConditionType>
        <tro:otherAccessRestriction>
            <com:values>
                <com:value lang="nl">Voorbehouden</com:value>
            </com:values>
        </tro:otherAccessRestriction>
        <tro:applicableLocation xsi:type="loc:AreaLocation">
            <loc:coordinatesForDisplay>
                <loc:latitude>51.03687330650958</loc:latitude>
                <loc:longitude>3.7265317473319737</loc:longitude>
            </loc:coordinatesForDisplay>
            <loc:namedArea xsi:type="loc:IsoNamedArea">
                <loc:areaName>
                    <com:values>
                        <com:value lang="nl">TARGEEL</com:value>
                    </com:values>
                </loc:areaName>
                <loc:namedAreaType>carParkArea</loc:namedAreaType>
                <loc:country>BE</loc:country>
                <loc:subdivisionType>district</loc:subdivisionType>
                <loc:subdivisionCode>22193</loc:subdivisionCode>
            </loc:namedArea>
            <loc:gmlMultiPolygon>
                <loc:gmlAreaName>
                    <com:values>
                        <com:value lang="nl">Zwijnaardsesteenweg 72118</com:value>
                    </com:values>
                </loc:gmlAreaName>
                <loc:gmlPolygon>
                    <loc:exterior>
                        <loc:_gmlLinearRingExtension>
                            3.726502652355531 51.03682199367616 3.726530764234032
51.036818556385214
                            3.726533265143784 51.03682739222782 3.726535734612505
51.0368361235661
                            3.726538208572778 51.036844854902725 3.726540682533084
51.03685358849726
                            3.726543156493357 51.03686232209011 3.726545624165451
51.03687105116229
                            3.726548093634172 51.0368797762787 3.726550563102892
51.03688849800417
                            3.72655303346991 51.0368972248119 3.726555506531886
51.03690595218285
                            3.72655797510231 51.03691468294147 3.726559150997011
51.03691883140692
                            3.726560788625753 51.036924618018546 3.726532840240657
51.036928203299716
                            3.726502652355531 51.03682199367616
                        </loc:_gmlLinearRingExtension>
                    </loc:exterior>
                </loc:gmlPolygon>
            </loc:gmlMultiPolygon>
        </tro:applicableLocation>
    </tro:condition>
    <tro:permitInformation>
        <tro:description>
            <com:values>
                <com:value lang="nl">2</com:value>
            </com:values>
        </tro:description>
    </tro:permitInformation>
    <tro:permitInformation>
        <tro:description>
            <com:values>
                <com:value lang="nl">12.02000000</com:value>
            </com:values>
```

```xml
                    </tro:description>
                </tro:permitInformation>
            </tro:trafficRegulation>
        </tro:trafficRegulationOrder>
    </tro:trafficRegulationsFromCompetentAuthorities>
</TrafficRegulationPublication>
```

2.5. Sectors, Pedestrian Streets, and Car-Free Areas Mapping to DATEX II

The source data of Sectors, Pedestrian Streets, and Car-Free Areas is presented in JSON format as shown below (with some coordinates omitted):


```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [3.725599, 51.05739],
            …
          ]
        ]
      },
      "properties": {
        "naam": "Kuip_VGG_Vrijdagsmarkt",
        "zone": "Autovrij Nieuw",
        "urid": "mob/sect63",
        "autovrijgebied": "Autovrij gebied 1",
        "autovrijgebied_codedvalue": "AV10",
        "geo_point_2d": {
          "lon": 3.725621,
          "lat": 51.057726
        }
      }
    }
  ]
}
```


The Sectors, Pedestrian Streets, and Car-Free Areas correspond to a `ControlledZoneTablePublication` in DATEX II, with fields matching those in the data class.

Gent Open Data Portaal	DATEX II	Comments
geometry	GmlPolygon.exterior._ gmlLinearRingExtension	Datex II Docs: GML (Linear and area location)
naam	ControlledZone.name	
zone	ControlledZone.description	

urid	ControlledZoneSummary.informalName	No specific identifier could be found in DATEX, closest was to set this in the summary.
autovrijgebied + autovrijgebied_coded value	AreaLocation.gmlMultiPolygon. gmlAreaName	No specific identifier could be found in DATEX II, closest was to append these values and set as the area name
geo_point_2d	ControlledZoneSummary.locationForDisplay	

The XML output is as follows:


```xml
<?xml version="1.0" encoding="UTF-8"?>
<ControlledZoneTablePublication
    xmlns:tro="http://datex2.eu/schema/3/trafficRegulation"
    xmlns:loc="http://datex2.eu/schema/3/locationReferencing"
    xmlns:cz="http://datex2.eu/schema/3/controlledZone"
    xmlns:comx="http://datex2.eu/schema/3/commonExtension"
    xmlns:com="http://datex2.eu/schema/3/common"
    xmlns:d2="http://datex2.eu/schema/3/d2Payload"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    lang="nl"
    modelBaseVersion="3">
    <com:feedDescription>
        <com:values>
            <com:value lang="nl">Sectoren/Voetgangersstraten/Autovrij Gebied Gent</com:value>
        </com:values>
    </com:feedDescription>
    <com:feedType>Mobiliteit</com:feedType>
    <com:publicationTime>2025-04-07T14:45:37.870605Z</com:publicationTime>
    <com:publicationCreator>
        <com:nationalIdentifier>BE</com:nationalIdentifier>
    </com:publicationCreator>
    <cz:controlledZoneTable>
        <cz:controlledZoneTableName>
            <com:values>
                <com:value lang="nl">Autovrij</com:value>
            </com:values>
        </cz:controlledZoneTableName>
        <cz:controlledZone>
            <cz:name>
                <com:values>
                    <com:value lang="nl">Kuip_VGG_Patershol</com:value>
                </com:values>
            </cz:name>
            <cz:controlledZoneType>limitedTrafficZone</cz:controlledZoneType>
            <cz:description>
                <com:values>
                    <com:value lang="nl">Autovrij Nieuw</com:value>
                </com:values>
            </cz:description>
            <cz:status>active</cz:status>
            <cz:controlledZoneSummary>
                <cz:cityName>
```

```
                    <com:values>
                        <com:value lang="nl">Gent</com:value>
                    </com:values>
                </cz:cityName>
                <cz:informalName>
                    <com:values>
                        <com:value lang="nl">mob/sect64</com:value>
                    </com:values>
                </cz:informalName>
                <cz:locationForDisplay>
                    <loc:latitude>51.0579257373271</loc:latitude>
                    <loc:longitude>3.7224861407265415</loc:longitude>
                </cz:locationForDisplay>
            </cz:controlledZoneSummary>
            <cz:controlledZoneRegulation>
                <tro:typeOfRegulation xsi:type="tro:AccessRestriction">
                    <tro:accessRestrictionType>noEntry</tro:accessRestrictionType>
                </tro:typeOfRegulation>
                <tro:condition xsi:type="tro:LocationCondition">
                    <tro:implementedLocation xsi:type="loc:AreaLocation">
                        <loc:gmlMultiPolygon>
                            <loc:gmlAreaName>
                                <com:values>
                                    <com:value lang="nl">Autovrij gebied 1 AV10</com:value>
                                </com:values>
                            </loc:gmlAreaName>
                            <loc:gmlPolygon>
                                <loc:exterior>
                                    <loc:_gmlLinearRingExtension>
                                        3.72438063672001 51.059444583212645 3.723656821874795
51.0596680109651 ...
                                    </loc:_gmlLinearRingExtension>
                                </loc:exterior>
                            </loc:gmlPolygon>
                        </loc:gmlMultiPolygon>
                    </tro:implementedLocation>
                </tro:condition>
            </cz:controlledZoneRegulation>
        </cz:controlledZone>
        <cz:controlledZone>
            <cz:name>
                <com:values>
                    <com:value lang="nl">Kuip_VGG_Paddenhoek</com:value>
                </com:values>
            </cz:name>
            <cz:controlledZoneType>limitedTrafficZone</cz:controlledZoneType>
            <cz:description>
                <com:values>
                    <com:value lang="nl">Autovrij Nieuw</com:value>
                </com:values>
            </cz:description>
            <cz:status>active</cz:status>
            <cz:controlledZoneSummary>
                <cz:cityName>
                    <com:values>
                        <com:value lang="nl">Gent</com:value>
                    </com:values>
                </cz:cityName>
                <cz:informalName>
                    <com:values>
                        <com:value lang="nl">mob/sect66</com:value>
                    </com:values>
                </cz:informalName>
                <cz:locationForDisplay>
                    <loc:latitude>51.051727190217754</loc:latitude>
                    <loc:longitude>3.7247820400580776</loc:longitude>
                </cz:locationForDisplay>
            </cz:controlledZoneSummary>
```

```xml
            <cz:controlledZoneRegulation>
                <tro:typeOfRegulation xsi:type="tro:AccessRestriction">
                    <tro:accessRestrictionType>noEntry</tro:accessRestrictionType>
                </tro:typeOfRegulation>
                <tro:condition xsi:type="tro:LocationCondition">
                    <tro:implementedLocation xsi:type="loc:AreaLocation">
                        <loc:gmlMultiPolygon>
                            <loc:gmlAreaName>
                                <com:values>
                                    <com:value lang="nl">Autovrij gebied 2 AV20</com:value>
                                </com:values>
                            </loc:gmlAreaName>
                            <loc:gmlPolygon>
                                <loc:exterior>
                                    <loc:_gmlLinearRingExtension>
                                        3.724547646107753 51.052151381493275 3.724525755062584
51.051896952398636 ...
                                    </loc:_gmlLinearRingExtension>
                                </loc:exterior>
                            </loc:gmlPolygon>
                        </loc:gmlMultiPolygon>
                    </tro:implementedLocation>
                </tro:condition>
            </cz:controlledZoneRegulation>
        </cz:controlledZone>
    </cz:controlledZoneTable>
</ControlledZoneTablePublication>
```

2.6. Existing Tools - UVARBox Tool

UVARBox, an EU project described in detail at UVARBox Tool, has already implemented DATEX II for UVAR modelling. This toolbox offers **a user interface for mapping data sources to the DATEX II model and provides validation** methods. However, the toolbox is not designed for machine-to-machine pipelines. It relies on a user interface, allowing only manual input data for DATEX II conversion.

2.7. Existing DATEX II Mappings

2.7.1. The National Road Traffic Data Portal (NL)

The National Road Traffic Data Portal (NL) represents the Dutch implementation of UVAR mapping. However, it uses its own namespaces as it was developed prior to the release of the official DATEX II version 3.6 schema.

- For the low emission zone, please refer to the schema available at DATEX II_3_Emissiezones-2024.zip via Downloads - NDW Docs. The relevant product can be found at NDW open data.
- For school zones, the schema can be accessed through School Zones - NDW Docs.

2.7.2. Mobilitek (DE)

The German portal at Mobilithek.info - Mobilitätsdaten Deutschland showcases DATEX II mappings. As of the document's date, the majority are based on DATEX II version 2 (224 mappings), with only four mappings utilizing DATEX II version 3. DATEX II version 3.6 is not utilized on the German portal.

Funded by
the European Union

THE CIVITAS INITIATIVE
IS CO-FUNDED BY
THE EUROPEAN UNION

DISCO is a project under the CIVITAS Initiative.
Read more - civitas.eu

Funded by
the European Union

THE CIVITAS INITIATIVE IS CO-FUNDED BY THE EUROPEAN UNION

DISCO is a project under the CIVITAS Initiative.
Read more - civitas.eu

