

Deliverable 2.1: MetaInnovations: Align, Charge, Automate, Connect, Manage and Integrate

WP2

Date of document

30/06/2025 (M18)

Deliverable Version: D2.1, V1.0

Dissemination Level: Public

Deliverable Lead: National Technical University of Athens (NTUA)



Co-funded by
the European Union

Deliverable administrative information

Overview	
Project Acronym	metaCCAZE
Project Title	Flexibly adapted MetaInnovations, use cases, collaborative business and governance models to accelerate deployment of smart and shared Zero Emission mobility for passengers and freight
Project Coordinator	ERTICO – ITS Europe
Project Duration	48 months
Deliverable Title	MetaInnovations: Align, Charge, Automate, Connect, Manage and Integrate
Deliverable No.	2.1
Dissemination Level	Public
Work Package	WP2: MetaInnovate: flexibly adjusted, open smart systems for zero emission and self driving shared mobility for passengers & freight
Deliverable Lead	National Technical University of Athens
Lead Authors	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA)
Main Contributor Authors	Antonis Lentzakis, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MaaSLab), Hassan Mahdavi (VED), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT), Aleksi Tepponen (REMOTED), Manos Barmponakis (MOBILYSIS), Santiago Alvarez (TUM), Leonor Gomes, Jiri Spitzer (TUD), Julian Garbiso (FLWR), Jeroen Steenbakkens (ARGALEO)
Other Contributors	TRASPORTI E TERRITORIO SRL (TRT), HC Linear Muszaki Fejlesztó Kft (HCL), TAMPEREEN KORKEAKOULUSAATIO SR (TAU)
Due date	30/06/2025
Submission date	30/06/2025
Reviewers	Hassan Idoudi (VED), Aleksi Tepponen (REMOTED)
Quality Assurance Reviewer	Lamprini Papafoti (FACTUAL)

Copyright Notices

©2024-2027 metaCCAZE Consortium Partners. All rights reserved.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

metaCCAZE is a Horizon Europe project supported by the European Commission under grant agreement No 101139678. metaCCAZE is a project under the CIVITAS Initiative, an EU-funded programme working to make sustainable and smart mobility a reality for all, and contributes to the goals of the EU Mission Climate-Neutral and Smart Cities. All information in this deliverable may not be copied or duplicated in whole or part by any means without express prior agreement in writing by the metaCCAZE partners. All contents are reserved by default and may not be disclosed to third parties without the written consent of the metaCCAZE partners, except as mandated by the Grant Agreement with the European Commission, for reviewing and dissemination purposes. All trademarks and other rights on third party products mentioned in this document are acknowledged and owned by the respective holders. The metaCCAZE consortium does not guarantee that any information contained herein is error-free, or up-to-date, nor makes warranties, express, implied, or statutory, by publishing this document. For more information on the project, its partners and contributors, please see the metaCCAZE website (www.metaccaze-project.eu).

Version History

Date	Version	Author	Comment
01/04/2025	0.1	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasiliadou (NTUA)	First draft of deliverable and initialization of metadata.
14/4/2025	0.2	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasiliadou (NTUA), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT)	First input by partners after request for input (by WP leader) at the WP2 monthly meeting.
9/5/2025	0.3	Aleksi Tepponen (REMOTED), Julian Garbiso (FLWR), Santiago Alvarez (TUM)	Completion of several deliverable sub-sections by partners.
27/5/2025	0.4	Antonis Lentzakis, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MaaSLab), Hassan Mahdavi (VED), Manos Barmounakis (MOBILYSIS), Leonor Gomes, Jiri Spitzer (TUD)	Finalization of input by WP2 partners.
30/5/2025	0.5	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasiliadou (NTUA)	Formatting homogenization.
16/06/2025	0.6	Jeroen Steenbakkens (ARGALEO), Antonis Lentzakis (MaaSLab), Konstantinos Gkiotsalitis, Dimitrios Rizopoulos (NTUA)	Input for T2.7 by Argaleo, finalization of Task 2.7 description in the deliverable.
25/06/2025	0.7	Hassan Idoudi (VED), Aleksi Tepponen (REMOTED)	First review and addressing of reviewers' comments.

27/6/2025	0.8	Lamprini Papafoti (FACTUAL)	Quality check.
30/6/2025	1	Konstantinos Gkiotsalitis, Alexandra Vasiliadou, Androniki Dimitriadou, Dimitrios Rizopoulos, Marilena Merakou (NTUA)	Reviewers' comments addressed and submission to project coordinator.

Table of Contents

Abbreviations and Acronyms	IX
Background: About the metaCCAIZE project	XIII
Executive Summary	XIV
1. Introduction	1
1.1. Objectives of the Deliverable	1
1.2. Structure of the Document.....	2
1.3. Relation to Project Documents	3
1.4. Overall Approach	3
2. Task 2.1 Align: AI optimisation and incentivisation engine for grid-fleet-demand	4
2.1. Subtask 2.1.1 Decentralised AI optimisation for grid-fleet-demand.....	5
2.1.1. The Grid-Fleet-Demand Management	5
2.1.2. Grid Management.....	11
2.2. Subtask 2.1.2 AI recommendation and incentivisation engine	11
2.2.1. Sustainable Charging Recommendation Engine	12
2.2.2. Renewable Energy Sources electricity mix contribution forecasting.....	12
2.2.3. Decision support platform to achieve sustainable charging practices	14
3. Task 2.2 Harmonise: AI-Data warehouse and APIs.....	14
3.1. Subtask 2.2.1 AI-Data Models and Warehouse.....	14
3.2. Subtask 2.2.2 APIs and Connectors	16
4. Task 2.3 Charge: Automated charging & V2G protocols.....	17
4.1. Literature Review Activities of Task 2.3.....	18
4.2. Development Activities of Task 2.3.....	19
5. Task 2.4 Automate: Advanced driving assistance systems and remote control centers for AVs	22
5.1. Subtask 2.4.1 Remote control center for AVs	22
5.2. Subtask 2.4.2 Advanced Driver Assistance System	27
5.3. Subtask 2.4.3 AI-based real-time planning of autonomous fleets	38
6. Task 2.5 Connect: V2V + V2X platform, protocols and traffic management at large scale ..	43
6.1. Subtask 2.5.1 Traffic management & condition assessment.....	44
6.2. Subtask 2.5.2 V2X-based support of transportation services.....	48
7. Task 2.6 Manage & Control: Electric Vehicle (re-)Scheduling, Low emission-based traffic management, Smart parking	52
7.1. Subtask 2.6.1 Electric Vehicle Scheduling	52
7.1.1. Electric Bus Charging Station Location Selection Problem with Slow and Fast Charging	53

7.1.2.	Electric Bus Charging Station Location Model considering Travel Time and Energy Consumption Uncertainties.....	56
7.1.3.	Combined Line Planning and Vehicle Scheduling for a fleet of Electric Buses (CLP-E-VSP) 64	
7.1.4.	Analyzing the Effects of Pick-Up and Drop-Off (PU DO) Duration and its Stochasticity on Mobility-On-Demand Services	72
7.1.5.	Optimal Dynamic Curb-side Management for mobility-on-demand services.....	82
7.1.6.	Multimodal Waste Collection Logistics	85
7.2.	Subtask 2.6.2 Low emission-based traffic management.....	88
7.2.1.	Predictive modeling with multi-source data	88
7.2.2.	TMC (Tradable Mobility Credits) assignment method for multiple population segments93	
7.3.	Subtask 2.6.3 Supply-demand matching platform for on-demand shared zero-emission services	97
8.	Task 2.7 Integrate + Plan: Digital Twin platform for optimisation	103
8.1.1.	Digital Twin implementations around the world	104
8.1.2.	Digital Twinning in the Amsterdam Living Lab	106
8.1.3.	Geoportal – Digital Twin hybrid for Limassol.....	107
9.	References.....	109
10.	Annex.....	116
10.1.	Annex sub-chapter for Chapter 2 and Task 2.1	116
10.2.	Annex sub-chapter for Chapter 3 and Task 2.2	117
10.3.	Annex sub-chapter for Chapter 4 and Task 2.3	117
10.4.	Annex sub-chapter for Chapter 5 and Task 2.4.....	117
10.5.	Annex sub-chapter for Chapter 6 and Task 2.5	120
10.6.	Annex sub-chapter for Chapter 7 and Task 2.6	121
10.7.	Annex sub-chapter for Chapter 8 and Task 2.7	121

Table of Tables

<i>Table 1: Optimal charging station physical location (V), and charging types (N=N1 ∪ N2) across the deterministic and all β(%) variation approaches</i>	<i>62</i>
<i>Table 2: Objective function value, total queue waiting time (in minutes), number of buses assigned to their closest charger, and computation time (in minutes) of the optimal solutions across all approaches</i>	<i>62</i>
<i>Table 3: Best solution of the combined line planning and vehicle scheduling problem for a fleet of 52 electric buses.....</i>	<i>70</i>
<i>Table 4: Design (555 scenarios).....</i>	<i>75</i>
<i>Table 5: Initial Results</i>	<i>96</i>
<i>Table 6: Commuting kms per week Analysis.....</i>	<i>96</i>
<i>Table 7: Total kms in 3 days Analysis</i>	<i>96</i>
<i>Table 8: Pricing policies.....</i>	<i>102</i>
<i>Table 9: Sensors Listing.....</i>	<i>117</i>
<i>Table 10: Lidar Specifications</i>	<i>117</i>

Table of Figures

<i>Figure 1: The bus network studied for Limassol, including eight bus lines operating within the limits of the urban environment.....</i>	<i>8</i>
<i>Figure 2: The candidate charger locations and the end stops for each bus line service considered for the Limassol case study</i>	<i>9</i>
<i>Figure 3: Pareto optimal solutions derived for the Limassol case study according to the two objectives</i>	<i>10</i>
<i>Figure 4: seq2seq model diagram</i>	<i>13</i>
<i>Figure 5: Conceptual framework architecture.....</i>	<i>15</i>
<i>Figure 6: IoT Module using an ESP32 microprocessor</i>	<i>16</i>
<i>Figure 7: Screenshot of received IoT data in Data Lake.....</i>	<i>17</i>
<i>Figure 8: Screenshot of received static data in Data Lake</i>	<i>17</i>
<i>Figure 9: Charging components for Inductive charging (Source: Sawilla & Schuett, 2017).....</i>	<i>18</i>
<i>Figure 10: Automated charging system architecture</i>	<i>20</i>
<i>Figure 11: Charging process steps.....</i>	<i>21</i>
<i>Figure 12: ROC flowchart</i>	<i>23</i>
<i>Figure 13: Conceptual Diagram of the Distributed Remote Control System Architecture</i>	<i>25</i>
<i>Figure 14: RCC Failover Process Example (RCC1 Fails, RCC3 Takes Over)</i>	<i>26</i>
<i>Figure 15: RCC Functional Block Diagram - Key Interactions.....</i>	<i>26</i>
<i>Figure 16: Sensors Placement on our prototype</i>	<i>28</i>
<i>Figure 17: Sensors Coverage bird view from the real robot</i>	<i>30</i>
<i>Figure 18: Simulated Navigation</i>	<i>31</i>
<i>Figure 19: Schematic representation of a docking manoeuvre</i>	<i>33</i>
<i>Figure 20: Problem Statement.....</i>	<i>33</i>
<i>Figure 21: Graphical representation of the Minimum Safety Distance (MSD)</i>	<i>34</i>
<i>Figure 22: Trend of the variable Relative Speed at Docking with respect to the input variables Leader Speed (a), Delta V (b), and Delta Distance (c).....</i>	<i>37</i>
<i>Figure 23: GSA indexes</i>	<i>37</i>
<i>Figure 24: Case study network</i>	<i>41</i>
<i>Figure 25: Case Study- Opposite direction.....</i>	<i>41</i>
<i>Figure 26: Passenger load variations when using modular autonomous vehicles with short-turning options and when implementing the as-is scenario with conventional vehicles</i>	<i>43</i>
<i>Figure 27: Identifying occlusion areas from trajectories of pedestrians.....</i>	<i>48</i>

Figure 28: Overview of V2X based support of mobility service	50
Figure 29: Architecture of the V2X Gateway components	51
Figure 30: 5G and V2X on-board unit for the connected vehicles.....	51
Figure 31: Athens network of the electric bus lines considered for the study, together with the physical locations candidates for the charging stations. Locations A, B and C on the upper left corner represent depots in the Ano Liosia, Nea Philadephia, and Anthousa regions.	55
Figure : Athens network of the electric bus lines, along with the proposed charging station physical locations	61
Figure 33: A generic representation of the modelled problem	66
Figure 34: A generic representation of the time-slice modelling approach. Each time slice solves the E-VSP problem based on the location, energy level and time status of a vehicle k provided by the previous time-slice.	66
Figure 35: A flowchart of the two-stage process including the E-VSP and LPP models	69
Figure 36: The three bus lines examined under this case study	70
Figure 37: Graphic representations of the number of passengers not served when considering a range of available vehicles of the number of required buses and the number of passengers not served considering the size of the available fleet for every bus line examined.....	71
Figure 38: PDFs of Log-normal distributions with pivoting mean and constant variance	73
Figure 39: Served requests with constant PUDO duration across requests	77
Figure 40: Served Requests with Heterogeneous PUDO duration	77
Figure 41: Abs. DO time dev. with constant PUDO duration across requests	77
Figure 42: Abs. DO time dev. with Heterogeneous PUDO duration.....	77
Figure 43: Late Drop-Off share with constant PUDO duration across requests	77
Figure 44: Late Drop-Off share with Heterogeneous PUDO duration	77
Figure 45: Impacts of increasing PUDO duration considering fully-informed operators.....	77
Figure 46: Served requests with constant PUDO duration across requests	78
Figure 47: Served Requests with Heterogeneous PUDO duration	78
Figure 48: Abs. DO time dev. with constant PUDO duration across requests	78
Figure 49: Abs. DO time dev. with Heterogeneous PUDO duration.....	78
Figure 50: Late Drop-Off share with constant PUDO duration across requests	78
Figure 51: Late Drop-Off share with Heterogeneous PUDO duration	78
Figure 52: Effects of Predicting PUDO Duration for Each Request	78
Figure 53: Operational impact of using different PUDO duration prediction models	79
Figure 54: Impact on the Absolute Drop-Off time Deviation.....	79
Figure 55: Impact on the Absolute Drop-Off time Deviation.....	79
Figure 56: LoS impacts of using different PUDO duration prediction models	79
Figure 57: Operational impacts using different Constant Insertion and Prediction Models	80
Figure 58: Impact on the Absolute Drop-Off time Deviation.....	80
Figure 59: Impact on the Late Drop-Off share.....	80
Figure 60: LoS impacts using different Constant Insertion and Prediction Models	80
Figure 61: Overview of predictive GNN-based model	90
Figure 62: Limassol Speed Loop	92
Figure 63: Paris Speed Loop	93
Figure 64: User app interface	98
Figure 65: Operator and driver app interface	99
Figure 66: School, home, activity locations selected in Limassol	100
Figure 67: Results for a fleet size of 10 Minibuses with 17-seat capacity.....	102
Figure 68: ECDF for the total daily student fare.....	103
Figure 69: Los Angeles GeoHub (credit (“City of Los Angeles Geohub” 2025))	105
Figure 70: Digital Twin Amsterdam: 3D visualization with building functions.	106

Figure 71: Digital Twin Amsterdam: Visualization of mobility flows	107
Figure 72: Planned Data Requirements	107
Figure 73: 3D map indicating EV charging locations (preliminary work on Geoportals-Digital Twin hybrid)	107
Figure 74: 2D maps indicating public transit routes and bus stop locations (at an aggregated level) .	108
Figure 75: The conceptual framework architecture	116
Figure 76: Example of LSTM Unit	116
Figure 77: Wind, Distributed, Conventional Generation, and Total Power Demand (Sourced from Cyprus TSOC)	117
Figure 78: Next Modular Vehicles' Third Generation Pod	118
Figure 79: Standard Sensor Coverage for AMVs. Source: Wireless Communication and Sensors in Self-Driving Cars - Pizarov, Jelena & Mester, Gyula. (2020). The Future of Autonomous Vehicles. FME Transactions.....	118
Figure 80: O-D Pair of Case Study	119
Figure 81: Flowchart of Traffic Management Framework - From Sensors to Services: A Scalable Pipeline for Predictive Traffic Management	120
Figure 82: Bathurst Spatial Digital Twin (credit ("Virtual Singapore - a 3D city model platform for knowledge sharing and community collaboration" 2025))	121
Figure 83: Paris Open Data (credit ("Open Data Paris" 2025))	122

Abbreviations and Acronyms

ACRONYM	Description
ADS	Automated Driving System
AMoD	Autonomous Mobility-on-Demand
AMVs	Autonomous Modular Vehicles
AOpv	Average Occupancy per vehicle
APIs	Application Programming Interfaces
AVs	Autonomous Vehicles
BEBs	Battery Electric Buses
BLE	Bluetooth Low Energy
BT	Bluetooth
BIGMs	Business Innovation and Governance Models
CAM	Cooperative Awareness Messages
C-ITS	Cooperative Intelligent Transport Systems
CPU	Central Processing Unit
CSI	Charging Station Installation Cost
C-V2X	Cellular V2X
CSLP	Charging Station Location Problem

DCC	Demand Charge Cost
DCM	Dynamic Curbside Management
DDS	Data Distribution Service
DENM	Decentralized Environmental Notifications
DS	Docking Speed
DTAP	Departure-Time Adjustment Procedure
EAC	Electricity Authority of Cyprus
EB	Electric Bus
EBCSLP	Electric Bus Charging Station Location Problem
EDCF	Empirical Cumulative Distribution Function
EDA	Exploratory Data Analysis
EI	Emergency Index
ELT	Extract-Load-Transform
EMI	Electromagnetic Interference
ESS	Energy Storage Systems
ETL	Extract-Transform-Load
EV	Electric Vehicle
E-VSP	Electric Vehicle Scheduling Problem
F-LLs	Follower Living Labs
FMS	Fleet Management System
FTE	Full-Time Equivalent
GDPR	General Data Protection Regulation
GMAN	Multi-graph Attention Networks
GNNs	Graph Neural Networks
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GSA	Global Sensitivity Analysis
GHG	Greenhouse Gas Emissions
HAVs	Highly Automated Vehicles

HES	Hybrid Energy Storage System
HMI	Human-Machine Interaction
IDFM	Île-de-France Mobilités
ITS	Intelligent Transportation Systems
KPIs	Key Performance Indicators
LCA	Latent Class Analysis
LLs	Living Labs
LoS	Level of Service
LPP	Line Planning Problem
LS	Local Search
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MCA	Multiple Correspondence Analysis
MEA	Mean Absolute Error
MD-E-VSP	Multi-Depot Electric Vehicle Scheduling Problem
MD-E-VSPTW	Multi-Depot Electric Vehicle Scheduling Problem with Time Windows
MILP	Mixed-Integer Linear Program
MINLP	Mixed-Integer Non-Linear Program
MIQP	Mixed-Integer Quadratic Program
MoD	Mobility-on-Demand
MPN	Mobility Panel of Netherlands
MPPT	Maximum Power Point Tracking
MRC	Minimal Risk Conditions
MSD	Minimum Safety Distance
MSE	Mean Square Error
OBU	On-Board Unit
OD	Origin-Destination
ODD	Operational Design Domains
PCA	Principal Component Analysis

PESS	Photovoltaic and Energy Storage Systems
PUDO	Pick-Up and Drop-Off
RBAC	Role Based Access Control
RCC	Remote Control Center
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
ROC	Remote Operations Center
ROS	Robot Operating System
RSU	Roadside Units
SAA	Sample Average Approximation
SA	Situational Awareness
SA	Simulated Annealing (algorithm)
SCSPs	Stochastic Constraint Satisfaction Problems
SMHs	Smart Mobility Hubs
SoC	State of Charge
TMC	Tradable Mobility Credit
ToU	Time-of-Use
ToUC	Time-of-Use Charging Cost
TSOC	Transmission Systems Operator of Cyprus
T-LLs	Trailblazer Living Labs
UART	Universal Asynchronous Receiver-Transmitter
UAVs	Unmanned Aerial Vehicles
UC	Use Case
UPS	Uninterruptible Power Supply
VIO	Visual Inertial Odometry
V/O	Vehicle-to-Operator
VRPPDTW	Vehicle Routing Problem with Pick-up, Drop-off and Time Windows
V2X	Vehicle-to-Everything
WP	Work Package
ZESM	Zero-emission Shared Mobility

2E-VRP-TW-TDTT

Two-Echelon Vehicle Routing Problem with
Time Windows and Time-Dependent Travel
Times

Background: About the metaCCAIZE project

metaCCAIZE is a Horizon Europe MISSION project co-funded by the 2Zero, CCAM-and Cities' Mission partnerships. It participates in the CIVITAS Initiative, an EU-funded programme working to make sustainable and smart mobility a reality for all and contributes to the goals of the EU Mission Climate-Neutral and Smart Cities.

The metaCCAIZE project aims to revolutionise mobility in European cities, serving both passengers and freight, with innovative electric, automated, and connected solutions designed to make transportation smarter, net zero, and more efficient for all. It builds on the expertise of 44 partners from 12 different European countries and contributes to the green metamobility era that the Green Deal, 2ZERO, CCAM, Cities Mission, CIVITAS and other EU initiatives aim to reach by 2030. In the vibrant streets of four trailblazer cities – Amsterdam, Munich, Limassol, and Tampere – metaCCAIZE implements, tests and demonstrates cutting-edge technologies and services that support shared zero emission mobility solutions for people and goods, contributing to climate neutrality. Successful technologies and activities are transferred and implemented to six Follower Cities – Athens, Krakow, Gozo, Milan, Miskolc, and Poissy, Paris.

MetaCCAIZE organises a series of metaDesign activities and develops a toolkit called metaInnovations. This toolkit provides innovations in passenger and freight services (public transport, on-demand minibuses, bike and scooter sharing, deliveries) and related infrastructure (mobility and logistics hubs, traffic management centres, charging infrastructure, transport and energy integration), and is widely demonstrated in our four trailblazer cities for a whole year. Successful metaInnovations and metaServices are transferred, implemented and demonstrated in the 6 follower cities for up to 8 months, to ensure their transferability and resilience potentials.

Executive Summary

This deliverable outlines the activities and progress achieved in Work Package 2 (WP2) of the metaCCAIZE project during the first eighteen months (M1–M18), with particular emphasis on the design and implementation of the developed tools. The technologies and solutions developed in the context of WP2 are aligned with the Use Cases (UCs) of Living Labs (LLs), aiming to accelerate the adoption of smart, shared, and zero-emission mobility solutions. The UCs are carried out in four Trailblazer Living Labs (T-LLs) located in Amsterdam, Munich, Limassol, and Tampere, as well as six Follower Living Labs (F-LLs) situated in Athens, Krakow, Gonzo, Milan, Miskolc, and Paris.

The technical approach and methodology of this deliverable is the presentation of the scientific tools developed for each Task according to the UCs and Assets of the MetaCCAIZE project. The outcome of WP1 resulted in designing and prototyping twelve innovative UCs. These UCs are designed to address critical urban mobility challenges and facilitate the transition towards sustainable, zero-emission transport for both passengers and freight in these four cities. This Deliverable aims to present the incorporation of the results from WP1 into applied tools and technologies that are going to create smart, shared, zero-emission mobility solutions for all LLs. All partners of WP2 developed scientific innovations that resulted from literature review in relevant topics and outcomes deriving from previous European projects and scientific research, as well as mathematical model formulation resulting in applications and tools that can be implemented in the network of the LLs.

The outcomes of D2.1 indicate how metaCCAIZE's smart technologies can create sustainable, zero-emission transport for passengers and freight. Some highlight results of the work described above are the AI optimisation and incentivisation tool for grid-fleet-demand, the AI-Data warehouse and APIs, the automated charging tool, the remote control center for electric autonomous vehicles, the driver assistant system, the AI-real time planning of autonomous vehicles, the traffic management technology for connected and autonomous vehicles, the electric vehicle scheduling and line planning tool, the waste management tool for electric fleets, and a digital twin platform for optimisation considering various data such as congestion levels, public transport schedules, ridership levels, service punctuality, energy consumption, and environmental indicators. All these findings and innovations are aligned with the objectives and goals of metaCCAIZE aiming for smarter transportation, with reduced environmental footprint.

The work described in this deliverable is aligned with metaCCAIZE's Major Milestone 2 which requires the draft version of the metaInnovations of WP2 delivered and communicated with the Trailblazer Living Labs of Work Package 3 (WP3). With respect to that, the work presented in this document is not final, but has reached a considerable level of maturity as required per GA requirements. In more detail, D2.1 presents the current stage of metaCCAIZE's metaInnovations, which will be updated in M26 by D2.2 and will take their final form in M40 by D2.3.

1. Introduction

1.1. Objectives of the Deliverable

The deliverable presents a comprehensive overview of the preliminary work on open innovations developed in WP2. Specifically, it outlines the first version of the metaInnovations toolkit, shaped through the metaDesign approach based on the Use Cases (UCs) provided by the Living Labs (LLs) and cities. The primary focus of the innovations described in this deliverable is their implementation in the LLs (WP3 and WP4).

Within the metaCCAIZE framework, the metaInnovations Toolkit showcases the project's cutting-edge technological advancements across seven key domains, which form the backbone of metaCCAIZE:

1. Align: Grid supply-fleet demand,
2. Harmonise: AI-Data warehouse,
3. Charge: Automated charging,
4. Automate: i) Remote control center, ii) Advanced driver assistance system, iii) AI-based real-time planning for autonomous fleets,
5. Connect: V2V and V2I protocols,
6. Manage & Operate: i) Electric vehicle scheduling, ii) Low-emission traffic management, iii) Supply demand matching for on-demand shared zero emission services, and,
7. Integrate and plan, through Digital twinning platforms.

Through these advancements, flexible, adaptable, open, robust, and replicable technological and infrastructural solutions are being developed to support the deployment of metaDesigned zero-emission shared mobility (ZESM) Use Cases, along with collaborative business and governance models. This toolkit represents the second objective (O2) of the metaCCAIZE project, and this deliverable serves as an introduction to it.

In WP2, the seven aforementioned technological domains are addressed through specific tasks, which are further divided into subtasks. This document details the foundational work carried out across these tasks up to June 2025, offering a comprehensive overview of the technological innovations developed up to this point.

More specifically, notable milestones achieved during this period include the development, and in many cases, application, of algorithms addressing various aspects of electric bus transportation networks, autonomous vehicles, on-demand services, e-fleet operation, tradable mobility credits, and electric waste collection optimization. Collectively, these algorithms tackle problems at the strategic, planning, and operational levels, providing valuable tools for public transport operators and policymakers to facilitate an efficient transition to shared, zero-emission mobility solutions for both people and goods; thus, contributing to climate neutrality.

Other important milestones include the development of an architecture for Data Warehouses designed to host a series of AI-based data models, as well as the completion of research into charging technologies that enable autonomous electric buses to recharge while parked or in motion. Significant progress has also been made in designing initial processes for operating a Remote Control Center (RCC) to manage and control Autonomous Vehicles (AVs). A simulation framework has been finalized for AVs' summoning and parking processes, incorporating sensor installation and real-world experiments in a test environment.

Moreover, a drone-based data collection model has been developed for traffic management purposes, and supply-demand matching tools for on-demand shared electric services have been finalized for both the user and operator interfaces.

These innovations demonstrate the significant technological progress achieved by the metaCCAZE project to date and lay a solid foundation for its future innovation developments.

1.2. Structure of the Document

The remainder of Deliverable D2.1 is structured as follows.

Chapter 2 presents the progress in Task 2.1, the “Align: AI optimisation and incentivisation engine for grid-fleet-demand”, which consists of two subtasks: 1) The Grid-Fleet-Demand Management and 2) the Grid Management direction. The efforts under Task 2.1 have been focused on developing decision support models for optimising the grid, the electric fleet, and the demand for charging to minimise net grid carbon intensity. It consists of two subtasks.

Chapter 3 presents the progress in Task 2.2, the “Harmonise: AI-Data warehouse and APIs”, which is described in two subtasks: 1) the AI-Data Models and Warehouse, and 2) the APIs and Connectors. Task 2.2 is responsible for the development of a sustainable AI-based data warehouse for storing and processing data to enable vehicle charging, integration with the grid, automation, and connectivity, and for building APIs and adaptors for the data warehouse to receive or give data required for systems and services operations.

Chapter 4 includes the description of progress conducted within Task 2.3, which is the “Charge: Automated charging & V2G protocols” and consists of two subtasks: 1) the Research and 2) the Development. The objective of this task is to develop a solution for charging automated vehicles without direct human intervention. The scope includes concept design, system design and integration, procurement of vehicles and charging infrastructure, and the development of a proof-of-concept (PoC) implementation.

Chapter 5 presents the progress in Task 2.4, which is the “Automate: advanced driving assistance systems and remote control centres for AVs”, which consists of three subtasks: 1) the Remote control center for AVs, 2) the Advanced Driver Assistance System, and 3) the AI-based real-time planning of autonomous fleets. Task 2.4’s objective is to examine advanced driver assistance systems and AI-driven real-time planning for autonomous vehicle fleets.

Chapter 6 focuses on the progress in Task 2.5, which is “Connect: V2V + V2X platform, protocols and traffic management at large scale”, and it consists of two subtasks: 1) the Traffic management & condition assessment, and 2) the V2X-based support of transportation services. Task 2.5 is devoted to the development & integration of V2X technologies for the support of intelligent roads dedicated to public and/or eco-responsible transportation means (carpooling).

Chapter 7 presents the progress in Task 2.6, which is “Manage & Control: Electric Vehicle (re-) Scheduling, Low emission-based traffic management, Smart parking” and consists of three subtasks: 1) the Electric Vehicle Scheduling, 2) the low-emission traffic management, and 3) the supply-demand matching platform for on-demand shared zero-emission services. Task 2.6. focused on the management and operation of zero-emission shared mobility services.

Chapter 8 presents the progress in Task 2.7, which is “Integrate + Plan: Digital Twin platform for optimisation”. The main goal is to deliver an integrated digital-twin planning and simulation platform, integrating the algorithms and frameworks developed by consortium partners. The digital twin is based on the HARMONY Model Suite. This final chapter of Deliverable D2.1 lists the review work and technical progress conducted up to now under Task 2.7.

1.3. Relation to Project Documents

Deliverable D2.1 is the initial report produced under WP2 and is directly linked to the progress updates submitted by WP2 partners in the project's first periodic report. As such, D2.1 builds upon the progress reported by the various WP2 task teams during this initial reporting period.

At a secondary level, D2.1 is also connected to several other deliverables that should be considered in parallel for a comprehensive understanding. Notably, Deliverable D1.1 provided an assessment of the Trailblazer Living Labs (LLs), Munich, Amsterdam, Limassol, and Tampere, and their respective Use Cases. In developing the metalInnovations for these LLs, WP2 partners were required to consider and analyze the Capability and Empathy Maps from D1.1, along with an analysis of initial barriers and existing services. In this way, the work presented in D2.1 not only reflects the current state of practice within the LLs but also aims to build upon it in alignment with stakeholder needs.

Furthermore, the work undertaken in D1.4 under WP1 has been a key input for WP2 activities and for the development of D2.1. D1.4 elaborated on the LL UCs and introduced the Prototype Use Case Template as well as the Business Innovation and Governance Models (BIGMs). These outputs offer a more standardized framework for Use Case development, which may also inform the creation of Use Cases for the Follower Cities' LLs. For that reason, it has been a central project document, that has been considered by WP2 partners, and is connected to the work described in the current document.

Another significant input for D2.1 has been Deliverable D3.1 of WP3, which details critical aspects of Trailblazer LL processes, such as management structures, roles, and responsibilities. Additionally, the Trailblazer LLs' work plans and Gantt charts have been central to guiding WP2's implementation activities and will continue to inform WP2 efforts throughout the remainder of the project timeline.

Deliverable D2.1 also plays a foundational role in shaping upcoming project deliverables. Most directly, it informs the future outputs of WP2, namely D2.2 and D2.3. D2.2 will document refinements and developments of the metalInnovations following their implementation and testing in the Trailblazer LLs. Subsequently, D2.3 will present the finalized versions of these metalInnovations, as informed by their application in the WP3 and WP4 LLs.

Beyond WP2, D2.1 is expected to influence a number of other forthcoming deliverables, particularly where it defines technologies to be deployed in both Trailblazer and Follower LLs. Specifically, D3.3 (impact evaluation) and D4.2 (monitoring aspect) are anticipated to be directly shaped by the implementation and outcomes of the metalInnovations described herein.

Finally, the metalInnovations and their evolution, as documented in D2.1, and to be further detailed in D2.2 and D2.3, are potential inputs to Deliverable D5.7, which will offer recommendations to CCAM and 2ZERO, as well as to D5.8, concerning WP6's exploitation and commercialization strategies.

1.4. Overall Approach

This deliverable was developed through close collaboration among the WP2 Task leaders, particularly those involved in Task 2.1, which was led by NTUA and co-led by MaaSLab (developing AI-based models for optimising the grid, the electric fleet and the demand for charging to minimise net grid carbon intensity). Furthermore, Task 2.2, which is responsible for the development of a sustainable AI-based data warehouse (storing and processing data) and for building APIs and adaptors was also led by MaaSLab. Task 2.3 focusing on developing a rapid charging infrastructure for electric and autonomous vehicles is conducted by Remoted. Moreover, Remoted along with

Next and NTUA, are responsible for Task 2.4 dealing with advanced driving assistance systems and control centers for AVs. Additionally, Task 2.5 which focuses on connecting: V2V + V2X is led by Vedecom, Task 2.6 dealing with electric vehicle (re-)scheduling and low emission-based traffic management is led by NTUA and Task 2.7 developing a digital twin platform for optimisation is conducted by MaaSLab. The overall orchestration of the deliverable was managed by NTUA, the partner leading WP2. NTUA also consulted key project documents, including the Grant Agreement and Deliverables 1.1 & 1.3 - Project Handbook (Inception, Quality, and Risk Management), to ensure that all descriptions and processes outlined here are aligned with these key documents.

The partners involved in WP2, both leaders and supporters, contributed to drafting the document sections relevant to the activities carried out in their respective LLs. Their contributions included descriptions of the Status Quo and UCs specific to their LLs. These inputs were then refined to ensure uniformity and comparability across all LLs. The final outcomes, such as the Status Quo Maps, Prototype UCs, and BIGMs, were further developed and interpreted by the WP2 core partners. These were then reviewed and fine-tuned by the LL partners and evaluated by consortium experts specializing in the topic.

This collaborative approach was structured to ensure that all descriptions within this deliverable align with the vision of the LLs and the other partners who contribute to the design or influence of the MetaServices and MetaInnovations that will be developed and showcased throughout the MetaCCAIZE project. Additionally, this deliverable for WP2, which is built upon previous research outcomes of WP1, enables the LLs to demonstrate their zero-emission shared mobility (ZESM) use cases in WP3 and WP4.

The document is organized around the seven Tasks of WP2 corresponding to Trailblazer and Follower cities and their UCs and Assets as noted in the Grant Agreement. In the following sections, an analysis of the WP2 Tasks is presented, describing their main objectives and the resulting tools. The outcome from these tasks, as presented in this deliverable, initiates the development of the metaInnovations that are built upon CCAM, 2ZERO and Horizon projects. The innovations are constantly fed by the outcomes of the metaDesign activities and are grouped into seven main categories as described below.

2. Task 2.1 Align: AI optimisation and incentivisation engine for grid-fleet-demand

Task 2.1 develops models based on Operations Research and Artificial Intelligence to optimise and align energy distribution, bus fleet energy use, and charging demand, with the goal of reducing the net carbon intensity of electricity used for charging electric vehicles, particularly electric bus fleets. The models developed in the context of Task 2.1 are tailored to transportation planning and operate across three planning horizons: strategic, tactical, and operational. Through optimisation, simulation and other digital artifacts (code scripts, data analysis, etc.), they incorporate charging scheduling flexibility, energy flows to towards the charging of the electric bus fleets, and other grid constraints to support low-carbon decision-making within the grid-fleet-demand framework.

Task 2.1 actively contributes towards the definition, design and development of metaCCAIZE Technical Asset 1, the Multi-depot electric vehicle scheduling problem (MD-E-VSP), and technical Asset 13, the Moby App. As a part of WP2, it supports metaCCAIZE project's second objective (O2), which is the development of the metaInnovations toolkit. Task 2.1 is comprised of Subtask 2.1.1 with title "Decentralised AI optimisation for grid-fleet-demand" and Subtask 2.1.2 with title "AI recommendation and incentivisation engine".

As an initial summary, during these first 18 months of the project, efforts under Task 2.1 have been focused on developing decision support models to optimize grid usage, electric fleet operations,

and charging demand with the goal of minimizing net grid carbon intensity. In Subtask 2.1.1, NTUA developed a digital tool that identifies optimal charging station locations by accounting for fleet-specific factors and energy pricing structures, including Time-of-Use (ToU) Tariffs and peak demand charges. The tool uses a bi-objective mathematical model and epsilon-constraint method, and its validation includes case studies on Limassol, Cyprus (Trailblazer LL). Looking ahead, NTUA plans to shift focuses on co-planning charging station networks when considering both the perspectives of the public transport operator and the energy grid management authority. Subtask 2.1.2, led by MaaSLab (MLAB), involves creating an AI-driven recommendation engine to identify optimal charging times based on renewable energy availability, based on data from Electricity Authority of Cyprus (EAC). Additionally, Subtask 2.1.2 includes the ongoing development of the WeePlatform, its alignment with the legal framework as well as its extension for accounting for energy storage systems.

2.1. Subtask 2.1.1 Decentralised AI optimisation for grid-fleet-demand

As part of Subtask 2.1.1, the project team has focused on developing two strategic planning models aimed at supporting the design of sustainable and efficient electric bus networks. A central feature of these models is their capacity to incorporate real-world public transport operations (based on General Transit Feed Specification data) of the Limassol Buses Company (EMEL) and the key parameters that govern the interaction between the energy grid and public transport service providers. This integrated approach ensures that network planning is not conducted in isolation but rather promotes alignment across the supply–fleet–demand triangle.

In that regard, the work within Subtask 2.1.1 is divided in two main research directions:

- The Grid-Fleet-Demand Management direction, which seeks to optimize the decisions of the *public transport services operator* when accounting for the parameters that define the operation of the energy grid.
- The Grid Management direction, where we seek to optimize decisions made by the energy management authority (e.g. energy pricing scheme), while accounting for energy needs of the bus fleet.

With respect to these two research and development directions, we proceed to discuss developments under subtask 2.1.1 in the first 18 months of the metaCCAZE project.

2.1.1. The Grid-Fleet-Demand Management

For this first research direction, NTUA focused on developing a novel bi-objective mixed-integer linear programming (MILP) model for the optimal extension of electric bus charging infrastructure, specifically designed to address both operational efficiency and economic feasibility. Recognizing the increasing global shift toward electrified public transportation, the research work tackles the dual challenge of strategic charger placement and charging schedule optimization for electric bus fleets. The developed multi-objective optimization model determines the planning decisions of public transport services operators according to two main objectives:

- Minimizing the total monetary cost (covering installation and daily operational costs) and,
- Minimizing deadhead time (the extra travel time required for buses to reach charging stations).

To offer actionable insights for decision-makers, the project team has applied the ϵ -constraint method to generate Pareto optimal solutions, enabling stakeholders to explore trade-offs between cost and operational performance. The model has been tested based on data from the Limassol Living Lab, showcasing its adaptability and practical relevance. Through this comprehensive framework, the study provides valuable tools for urban planners and public transport authorities to support sustainable, cost-efficient transitions to electric bus systems.

In the next subsections, a detailed presentation of the model developed under the Grid-Fleet-Demand Management direction will be given. The presentation will start with the objective functions of the MILP model, and then it will continue with the constraints of the model and then the results will be given.

Objective functions

To begin with, the model in this first research direction focuses on two main objectives for managing an electric bus charging network. The first is to minimize deadhead time (non-revenue travel between route endpoints and charging stations) which consumes energy and driver resources. Reducing it improves fleet efficiency by increasing service time and reducing idle repositioning. This supports more cost-effective and sustainable operations. The corresponding objective function is shown in Equation (1):

$$\text{minimize } \mathcal{O}_1 = \sum_{j \in N} t_{kj} q_{kj} \quad \text{for all } k \in \mathcal{K} \quad (1)$$

In Equation (1), t_{kj} represents the travel time from a bus $k \in K$, where K is the total set of buses to any candidate $j \in N$ (set of candidate chargers to be installed) and q_{kj} represents the decision to assign a bus k to any charger j (binary variable). In addition to this first objective for the MILP model, a second objective is considered in order to stimulate the model towards minimizing the monetary cost associated with extending and operating the charging infrastructure. This includes the capital cost of installing new chargers, the operational cost of electricity consumption, calculated using ToU tariffs, and the peak demand charges levied by energy providers based on the highest power drawn during billing periods. In equations (2), (3), (4) and (5) the mathematical expressions for the second objective functions are given.

$$\text{TOUC} = \sum_{f_3}^{F_3} (T_{f_3} \cdot DEC_{f_3}) \quad (2)$$

$$\text{DCC} = \text{DCR} \cdot \text{IP}^{\max} \quad (3)$$

$$\text{CSI} = \sum_{j \in N_5 \cup N_6} x_j b_j \quad (4)$$

$$\mathcal{O}_2 = \text{TOUC} + \text{DCC} + \text{CSI} \quad (5)$$

The second objective function is given in Equation (5) and reflects the total monetary cost. This cost is captured through three components. Equation (2) represents the *Time-of-Use Charging Cost (TOUC)*, calculated as the sum of energy consumed across all time periods (measured in kWh) multiplied by the corresponding electricity price (tariff) at each period. Equation (3) defines the *Demand Charge Cost (DCC)*, which accounts for the highest peak power demand (in kW) during any billing period, multiplied by the demand charge rate imposed by the energy provider. Equation (4) calculates the *Charging Station Installation Cost (CSI)* as the total cost of installing new chargers, summing the fixed cost of each charger that is selected for installation.

Equations (2), (3) and (4) are calculated based on a few parameters and decision variables of the model. In more detail, In Equation (2), the total Time-of-Use Charging Cost (TOUC) is calculated by summing the product of two key components over all billing time periods. The first component is T_{f_3} , a parameter representing the Time-of-Use electricity tariff, which defines the price per kilowatt-hour (kWh) of electricity during each time slot f_3 . The second component is DEC_{f_3} , a variable representing the total amount of electrical energy consumed at all charging stations, both slow and fast, during each respective time period f_3 . Equation (3) addresses the Demand Charge Cost

(DCC), which accounts for the cost imposed by energy providers for the maximum level of power drawn from the grid during the day. Here, DCR is a parameter denoting the demand charge rate, typically expressed in monetary units per kilowatt (kW). IP_{max} is a variable representing the peak power demand observed across all time periods. This component ensures that the model considers not just the total energy consumed, but also the impact of peak loads on the cost.

Finally, Equation (4) captures the Charging Station Installation (CSI) cost. It sums the product of x_j , a binary decision variable indicating whether charger j is installed (1 if installed, 0 otherwise), and b_j , a parameter denoting the fixed installation cost of charger j . This part of the model reflects the capital investment required to expand the charging network. Together, these three equations form the basis of the model's second objective: minimizing the total monetary cost of infrastructure deployment and energy use.

By inspecting mathematical expressions (1) and (5) closely, one may notice that these two objectives, O_1 and O_2 often conflict: placing chargers in locations that minimize deadhead time may require installing more stations or deploying them in less grid-efficient areas, thereby increasing costs. Conversely, minimizing monetary costs might concentrate chargers in fewer, grid-friendly locations, potentially increasing the travel time for buses to access charging. To balance this trade-off, the model uses a bi-objective optimization approach, allowing decision-makers to explore a range of Pareto-optimal solutions that reflect different priorities underlined by different trade-offs. This dual-objective design captures the operational and economic dimensions of electric fleet management, making the model a valuable tool for planning sustainable and cost-effective electric bus networks. The model also includes a number of constraints which formulate the real-world operation of the bus network and the interactions between decisions of the public transport operator and the pricing imposed by the energy management authority (see the published work Rizopoulos and Gkiotsalitis (2025)).

To solve this formulation, we apply the ϵ -constraint method, which is a classical approach for generating Pareto-optimal solutions in bi-objective optimization problems. Under this method, one of the objectives, often referred to as the "primary" objective, is optimized directly, while the other is converted into a constraint bounded by a parameter ϵ (epsilon). By systematically varying parameter ϵ over a range of feasible values, we obtain a family of solutions that trade-off between the two objectives. Each fixed ϵ -constraint yields a different feasible region and thus a different optimal solution, and collecting these solutions provides an approximation of the entire Pareto front. This technique allows decision-makers to explore a diverse set of optimal trade-offs and select a solution that best aligns with their operational priorities. Based on this, the problem to be solved is transformed into:

$$\begin{aligned} & \text{minimize } O_2 \\ & \text{subject to:} \\ & O_1 \leq \epsilon_m \end{aligned}$$

Model application to Limassol Living Lab

In the context of evaluating real-world charging infrastructure strategies, the Limassol Living Lab serves as a practical test site for applying the proposed bi-objective optimization model. Limassol, a coastal city with a steadily growing transportation network, embodies many of the operational complexities faced by modern urban environments making a switch to electric bus services. Geographical factors, grid capacity constraints, and diverse travel patterns all add layers of realism to the planning problem. By modelling these intricate conditions, the Limassol Living Lab facilitates

a comprehensive assessment of the trade-offs between infrastructure expansion and operational outcomes.

To apply the model, spatial data on candidate charger locations were combined with information on bus routes, operational timetables, and the city's ToU electricity tariffs. Additionally, known local constraints, such as maximum permissible chargers at each site and the demand charges imposed by the utility, were built into the formulation. The Limassol Living Lab thus provides a detailed environment where different charging scenarios, ranging from slow and fast chargers to partial recharging strategies, could be evaluated in a unified optimization framework. By doing so, the model's robustness and adaptability could be examined under realistic conditions, providing insights into how bus operators might expand their charging infrastructure while balancing energy costs and minimizing deadhead times.

The network of buses studied for Limassol LL is depicted in Figure 1, while in Figure 2 one may notice the candidate charger locations considered.

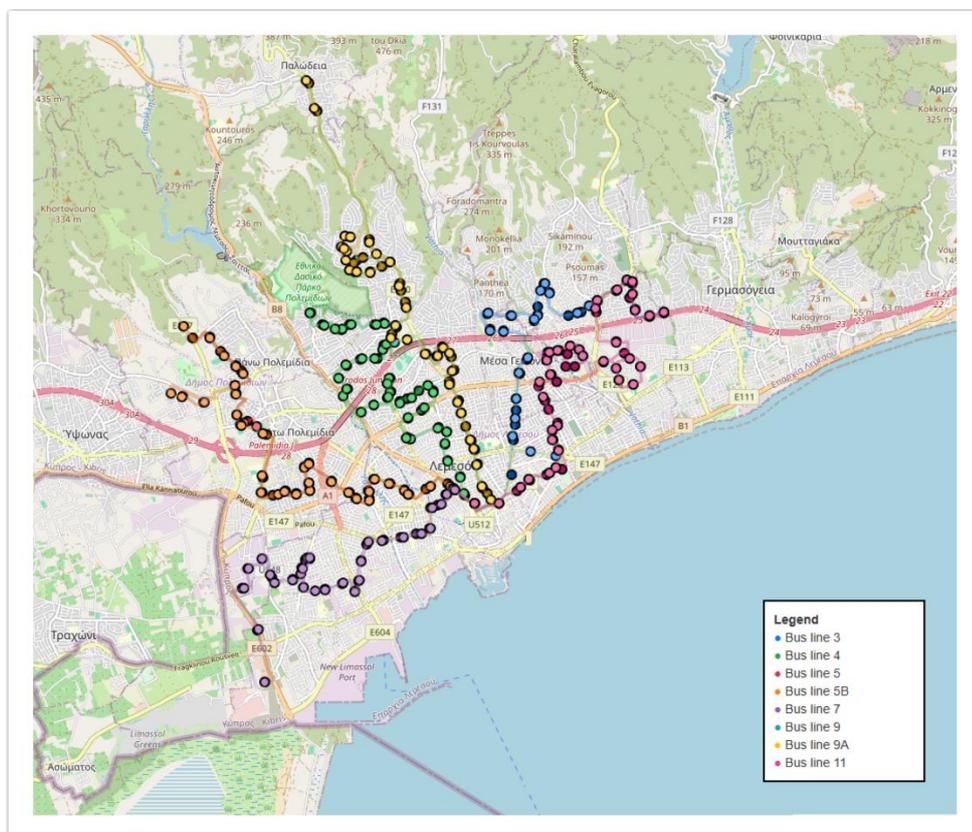


Figure 1: The bus network studied for Limassol, including eight bus lines operating within the limits of the urban environment

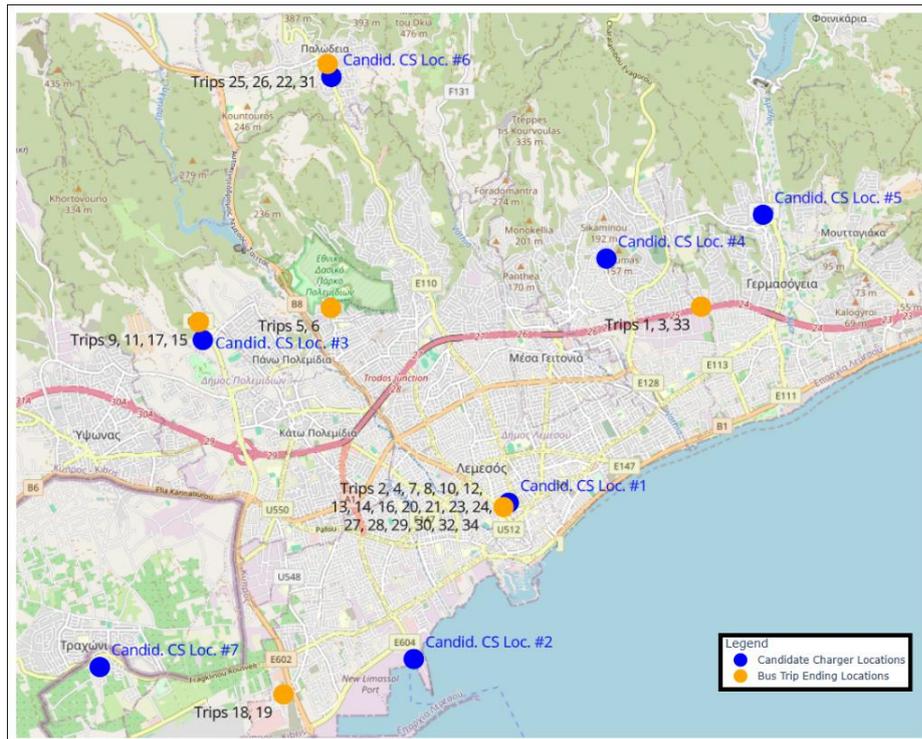


Figure 2: The candidate charger locations and the end stops for each bus line service considered for the Limassol case study

In applying the bi-objective model to the Limassol Living Lab, it became clear that there is no single “best” way to extend the electric bus charging network. Instead, the optimization process revealed multiple Pareto optimal solutions, each reflecting a unique compromise between minimizing monetary cost and reducing deadhead time. Rather than collapsing all trade-offs into a single outcome, this approach illustrates how investing more in strategically located chargers can reduce operational inefficiencies, but may raise the capital and energy-related expenses, and vice versa. These trade-offs underscore that both objectives are valid and often conflicting, leading to different configurations of slow and fast chargers that satisfy the two objectives in varying degrees.

By enumerating these Pareto optimal solutions, stakeholders such as transit authorities and policy makers gain valuable insight into how incremental shifts in one objective (e.g., lowering costs) can impact the other (e.g., additional route extensions for more efficient bus operations). Within the Limassol Living Lab context, such an understanding promotes more informed decision-making, allowing authorities to align their choices with broader organizational or civic goals, whether they prioritize cost-effectiveness, reduced idle travel, or a balanced combination of both. In Figure 3, the Pareto optimal solutions derived for the network of Limassol are given.

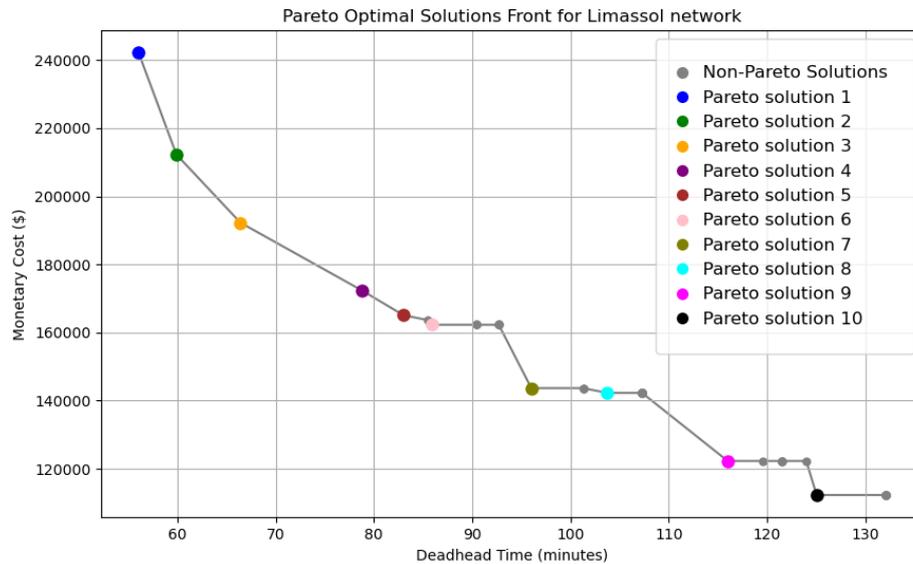


Figure 3: Pareto optimal solutions derived for the Limassol case study according to the two objectives

These 10 Pareto Optimal solutions highlighted with different colours, represent 10 different real-world configurations of chargers, and respective charging scheduling of buses to slots. Among the identified Pareto optimal solutions for the Limassol Living Lab, Pareto Solution 1 and Pareto Solution 10 stand at opposite ends of the cost-deadhead spectrum while each offering distinct advantages. Pareto Solution 1 leans toward lower overall monetary costs for installing and operating the charging network, but at the expense of higher deadhead times for buses. It can therefore be attractive for stakeholders prioritizing budget constraints over operational efficiency. Conversely, Pareto Solution 10 emphasizes a more extensive network of chargers, strategically placed to significantly reduce the deadhead travel of electric buses, yet it entails a higher capital outlay and increased energy-related expenditures. This solution is potentially ideal for operators who wish to minimize buses' non-revenue travel, even if that means committing more resources to infrastructure and power.

Pareto solutions that offer a middle ground exist. Pareto solution 5 occupies a balanced middle ground between the minimal-cost and maximal-efficiency extremes in the Limassol Living Lab. By incorporating a moderate level of investment in new chargers, it achieves a significant reduction in deadhead times, though not as pronounced as in the highest-investment scenarios, while keeping installation and operational costs substantially lower than the most charger-intensive configurations. This trade-off particularly benefits operators looking for a compromise that respects budget limitations yet still addresses inefficient travel. Consequently, solution 5 demonstrates the model's capacity to generate practical outcomes for decision-makers who must balance fiscal considerations with acceptable service performance.

Initial Implications from model application to Limassol Living Lab

By applying our bi-objective optimization model to the Limassol Living Lab, we illustrate how a systematic, data-driven approach can substantially enhance decision-making in designing electric bus charging networks. One crucial insight derived from our work is the model's effectiveness in quantifying trade-offs across multiple real-world constraints, such as local grid capacity and spatial limitations. Rather than simply proposing a singular, one-size-fits-all solution, our framework identifies several viable infrastructures and charging schedules that reflect different priorities, from minimizing upfront investment to minimizing service disruptions. This enables local

authorities and transport operators to select an option that resonates with both their budgetary realities and broader sustainability objectives.

For more detail, we note that the model described above has been published to a *Q1-level peer reviewed journal* (Transportation Research Part C: Emerging Technologies) with the title “Extending electric bus charging infrastructure considering charging scheduling and energy pricing” [Κάντε κλικ ή πατήστε εδώ για να εισαγάγετε κείμενο.](#) and has also been presented at the *hEART 2025* conference as a podium presentation.

2.1.2. Grid Management

In the Grid Management direction, our primary focus during the first 18 months of the project has been to identify the most effective modelling approaches from the perspective of the energy grid operator. To this end, we conducted a thorough review of both internal and external scientific literature, emphasizing peer-reviewed journal articles. This groundwork enabled us to establish a foundational basis for the digital mathematical model that will be further developed in the coming months.

Within the scope of electric bus charging schedules, one of the key insights from our research is the importance of treating energy prices as decision variables rather than fixed parameters. By incorporating dynamic or TOU tariffs, peak/off-peak rates, and other flexible pricing structures directly into the model, we can help operators respond to real-time cost signals and adapt charging routes to minimize expenses. Previous literature studies [Κάντε κλικ ή πατήστε εδώ για να εισαγάγετε κείμενο.](#) confirm that this approach not only reduces operational costs but also mitigates peak-demand pressures on the grid.

A second major insight involves the use of bi-level programming to coordinate decision-making between electric bus operators and energy management authorities. In this hierarchical setup, the upper level typically represents the energy authority’s objectives, such as balancing demand and alleviating congestion, while the lower level focuses on the operator’s cost minimization objectives, accounting for fleet schedules, passenger needs, and route planning. As demonstrated by previous research [Κάντε κλικ ή πατήστε εδώ για να εισαγάγετε κείμενο.](#), such a bi-level model provides a structured way for both stakeholders to optimize their outcomes simultaneously without jeopardizing grid stability.

Building on these foundational elements, the next phase of the project will be devoted to creating a MILP framework that integrates variable electricity pricing and leverages bi-level coordination. By explicitly incorporating energy price variability and synchronizing operator scheduling with grid constraints, we anticipate a robust, cost-effective, and sustainable charging strategy for electric bus fleets, one that not only advances the operational objectives of transport providers but also supports a more resilient overall power system.

2.2. Subtask 2.1.2 AI recommendation and incentivisation engine

As part of Subtask 2.1.2, the project team is developing an interactive platform to support time-of-day charging decisions using real-time recommendations based on personalized patterns, energy pricing, and sustainability goals. The platform will use visual analytics and predictive insights to raise awareness of charging strategies and show estimated carbon emission reductions, helping EV owners and fleet operators make informed choices. In that regard, the work within Subtask 2.1.2 is divided in two main parts:

- The forecasting of Renewable Energy Sources contribution to the electricity mix, accounting for potential growth projection in the Renewable Energy production sector.

- The development of an interactive platform to support user decisions with the ultimate goal of reducing carbon emissions achieved through sustainable charging practices.

We proceed to discuss the developments under subtask 2.1.2 in the first 18 months of the metaCCAIZE project.

2.2.1. Sustainable Charging Recommendation Engine

This subtask involves one activity: developing an AI-based recommendation engine. Its main goal is to reduce emissions by identifying times when renewable energy supply to the grid is highest and advising fleet operators and users on optimal vehicle charging without overloading the grid. The engine will be integrated into WeePlatform. The conceptual architecture is complete and can be viewed in Figure 75 in the Annex of this deliverable. The engine will recommend ideal charging times for electric vehicles and when to sell energy back to the grid. In Cyprus, current laws do not permit Hybrid Energy Storage (HES) Systems alongside renewable energy, but short-term government plans include HES system licensing.

2.2.2. Renewable Energy Sources electricity mix contribution forecasting

In this phase, the project team focused on forecasting the contribution of Renewable Energy Sources (RES) to the electricity mix by adapting a framework based on a Sequence-to-Sequence (Seq2Seq) Neural Network architecture. This architecture consists of two main components:

1. **Encoder:** The encoder processes the input sequence and encodes it into a fixed-length context vector or series of hidden states. The encoder is typically implemented using Recurrent Neural Networks (RNNs) or its variants, the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. It processes the input sequence one element (word, token, or time step) at a time, updating its internal hidden state as it moves through the sequence. After processing the entire sequence, the encoder outputs a context vector that summarizes the information from the input sequence. This context vector serves as the memory for the decoder.
2. **Decoder:** The decoder uses this encoded information (context vector) to generate the output sequence. The decoder is another RNN, LSTM or GRU network that generates the output sequence, one token at a time. The context vector from the encoder provides the initial information that the decoder needs to start generating the output. In some configurations, the decoder is trained through a process where the actual previous token in the output sequence is used as input for the next step, instead of the model's previous prediction.

The LSTM is a variant of RNN that is capable of learning long-term dependencies. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using backpropagation. LSTM also implements the recurrent structure of simple RNNs but adds four gates to all cell states. A common LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Three main gates for each input, hidden and output state, called forget, input and output.

- **Forget gates:** Decide what information to discard from the previous state by mapping the previous state and the current input to a value between 0 and 1. A rounded value of 1 means to keep the information, and a value of 0 means to discard it.
- **Input gates:** decide which pieces of new information to store in the current cell state, using the same system as forget gates.

- **Output gates:** control which pieces of information in the current cell state to output by assigning a value from 0 to 1 to the information, considering the previous and current states.

An example of LSTM Unit is given in Figure 76 in the Annex of this deliverable report. Selectively outputting relevant information from the current state allows the LSTM network to maintain useful, long-term dependencies to make predictions, both in current and future time steps. LSTMs can have a very complex structure with higher computational cost. To overcome this, GRU were introduced which uses LSTM architecture by merging its gating mechanisms offering a more efficient solution for many sequential tasks without sacrificing performance. The core idea behind GRUs is to use gating mechanisms to selectively update the hidden state at each time step allowing them to remember important information while discarding irrelevant details. GRUs aim to simplify the LSTM architecture by merging some of its components and focusing on just two main gates: the update gate and the reset gate.

- **Update Gate:** This gate decides how much information from previous hidden state should be retained for the next time step.
- **Reset Gate:** This gate determines how much of the past hidden state should be forgotten.

These gates allow GRU to control the flow of information in a more efficient manner compared to traditional RNNs which solely rely on hidden state. GRUs are more computationally efficient because they combine the forget and input gates into a single update gate. GRUs do not maintain an internal cell state as LSTMs do, instead they store information directly in the hidden state making them simpler and faster. Figure 4 presents a diagram of the seq2seq model using GRU networks for the encoder and decoder.

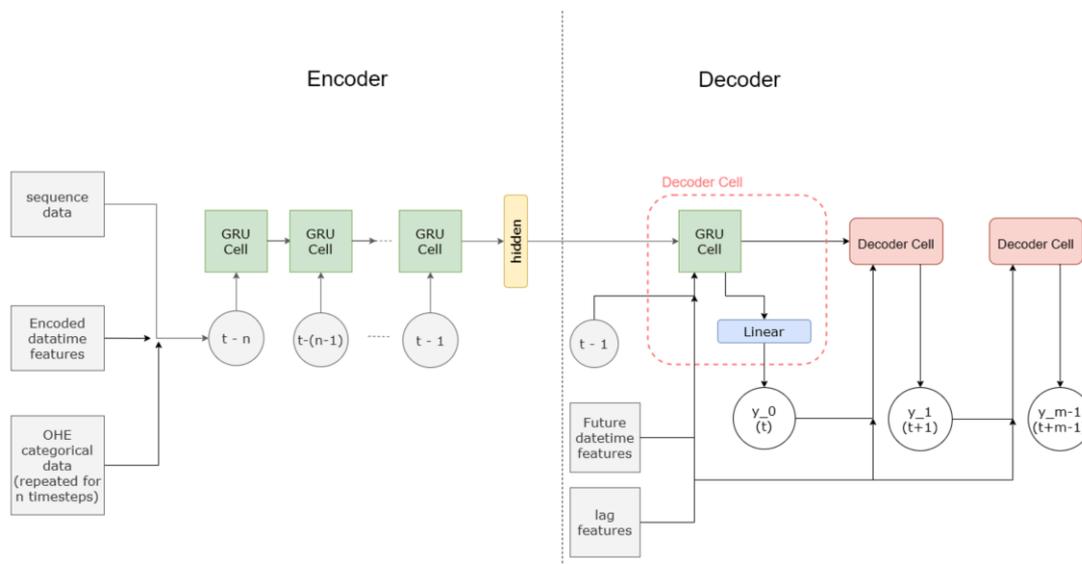


Figure 4: seq2seq model diagram

The project team has acquired time series data from TSOC, including Wind Generation, Distributed Power Generation Estimates (Solar Power, Biomass), Conventional Power Generation (Fossil Fuel) and Total Power Generation as can be seen in Figure 77 in the Annex. We have also observed the differences in maximum Power generation from Renewable Energy Sources over the years, from 2018 until 2024, and are in the process of estimating the Renewable Energy Sources Growth projection.

2.2.3. Decision support platform to achieve sustainable charging practices

For the second part, the project team has been developing an engine to support EV user charging decisions, based on the forecasted contribution to the electricity mix from Renewable Energy Sources. Specifically, recommendations will be provided to the EV fleet operator/private user for which time-of-day should they commence charging their vehicles so as to reduce carbon emissions from Grid Power Generation, while taking the ToU tariffs and peak demand charges imposed by the EAC into consideration.

As mentioned in the previous subsection, EAC and Transmission Systems Operator of Cyprus (TSOC) datasets have been acquired. Data requests for the charging load profiles of fleet operators and EV charger operators have been sent, however, in the case of EV charger operators, no responses have been forthcoming. Charging load profiles from EMEL (Limassol Buses Company) as EV fleet operator are in the process of being calculated implicitly, based on the operator's policy of charging buses during the drivers' breaks.

The project team will take into consideration the ToU tariffs, peak demand charges, charging policies and, in combination with the electricity mix contribution forecasting results, will be able to derive in Gantt chart format the recommended time-of-day for charging, so as to maximize emission reduction in Power Generation.

3. Task 2.2 Harmonise: AI-Data warehouse and APIs

Task 2.2 is responsible for the development of a sustainable AI-based data warehouse for storing and processing data that will enable WP2 partners to develop digital tools for vehicle charging, integration of bus fleets with the energy grid, further automation and connectivity, as well as building APIs and adaptors for the data warehouse to receive or give data required for systems and services operations.

The work within task 2.2 is divided in two main subtasks:

- **2.2.1 titled AI-Data Models and Warehouse:** The main goal of this activity is to provide a comprehensive and extendable space able to host static and streamed heterogeneous data, ranging from structured connected vehicles and grid data to semi-structured video streams of traffic, to bird-eye (drone) traffic data, to variable-size unstructured qualitative data from social innovation instruments.
- **2.2.2 titled APIs and Connectors:** The main goal of this activity is to ensure interoperability and secure sharing by developing application programming interfaces (APIs) and connectors for streaming and queuing message exchanges and linking data infrastructures supporting data capture and event-based data flows.

With respect to these two subtasks, we proceed to discuss developments under subtasks 2.2.1 and 2.2.2. For subtask 2.2.1 a conceptual framework architecture for the AI-Data Models and Warehouse has been implemented and for subtask 2.2.2 APIs and Connectors have been set up between the components of the architecture framework presented in 2.2.1.

3.1. Subtask 2.2.1 AI-Data Models and Warehouse

Under subtask 2.2.1, the project team has developed the conceptual framework architecture in Figure 5. It should be noted that certain elements within this architecture are part of the following subtask 2.2.2. The project team selected to use the Microsoft Azure public cloud computing platform, since it can provide access to diverse, flexible, cloud-based services involving infrastructure, networking, storage, analytics and AI-based solutions. Users can choose from these services to develop and scale new applications, or run their existing applications, in the cloud.

Azure helps businesses and organizations manage and deploy applications globally with ease and flexibility.

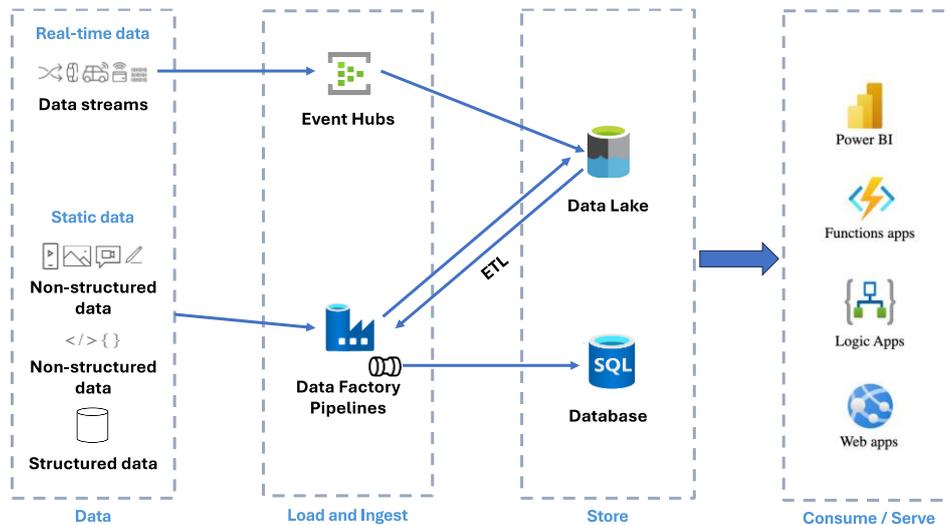


Figure 5: Conceptual framework architecture

In the Data component of the architecture (as seen in Figure 5), real-time data streams will be used, as well as static data which can be classified as:

- Non-structured data in video format, image format, pdf format, presentation slides etc.
- Non-structured data in text format (docx, xml, json, geojson)
- Structured data in table format

In the Load and Ingest component of the architecture, Event Hubs, a fully managed, real-time data ingestion service, part of the Microsoft Azure ecosystem is being considered for real-time data stream ingestion. Pipelines have been set up in Azure Data Factory, a managed cloud service built for complex hybrid extract-transform-load (ETL), extract-load-transform (ELT), and data integration projects, where Big data processing needs to be orchestrated and operationalized in order for the data to be refined into actionable insights.

In the Store component of the architecture, Azure Blob Storage, a scalable and cost-efficient cloud storage service, has been set up to act as the Data Lake, wherein all non-structured data will be stored, initially without any processing and if necessitated by business requirements, after undergoing ETL or ELT processing in the Azure Data Factory Pipelines.

In the Consume/Serve component of the architecture, several services are used to provide actionable insights from the ingested and transformed data. Azure Analytics and Power BI together provide insights at scale. Azure Functions is a serverless solution used to build robust apps while using less code, and with less infrastructure and lower costs. Azure Logic Apps allow to create and run automated workflows. The Azure App Service allows to run web applications, mobile back ends, and RESTful APIs without worrying about managing the underlying infrastructure. Finally, Azure Web Apps are a specific type of Azure App Service that allow us to host web applications.

The Data Warehouse implementation consists of 5 blocks, 3 of which have been tested using sample datasets and deemed to be complete.

1. Real-time data ingestion:

- Resources needed: Event Hub, Data Lake (Azure Blob Storage)
- Criteria to set as complete:
Ability to ingest:

- a. Real-time structured data (e.g. IoT data)
 - b. Real-time images or video (e.g. bird-eye photos from Drones, or videos from Traffic-light cameras)
- 2. Static data ingestion without transformation**
 - Resources needed: Data Factory, PostgreSQL (+postGIS plugin)
 - Criteria to set as complete:
Ability to ingest: Static structured data
 - 3. Data Factory ETL pipeline to Data Warehouse**
 - Resources needed: Data Factory, Data Factory pipeline, Postgres (+postGIS plugin)
 - Criteria to set as complete:
Ability to ingest: Static structured data
 - 4. Data Factory ETL pipeline from Data Lake to Data Warehouse**
 - Resources needed: All the above
 - Criteria to set as complete:
Ability to ingest: Structured data from Data Lake to Database
 - 5. Consume data (API)**
 - Explore the ways to consume the data, with permission levels.

3.2. Subtask 2.2.2 APIs and Connectors

Under subtask 2.2.2 the project team works on setting up APIs and Connectors between the components of the architecture presented in Figure 5. An API connector for real-time data streaming to the Data Lake was implemented using an MQTT server. Streaming data was generated from a homebrewed IoT module, using an ESP32 microprocessor similar to the one seen on Figure 6. The data was successfully received and stored in the Data Lake, see Figure 7. An API connector for static data ingestion in the Data Lake was implemented using an HTTP server. The data was successfully received and stored in the Data Lake, when tested with image files but works with any file type, see Figure 8. Implementation of a predefined set of data access permission levels was completed using Role Based Access Control (RBAC).



Figure 6: IoT Module using an ESP32 microprocessor

Data Explorer

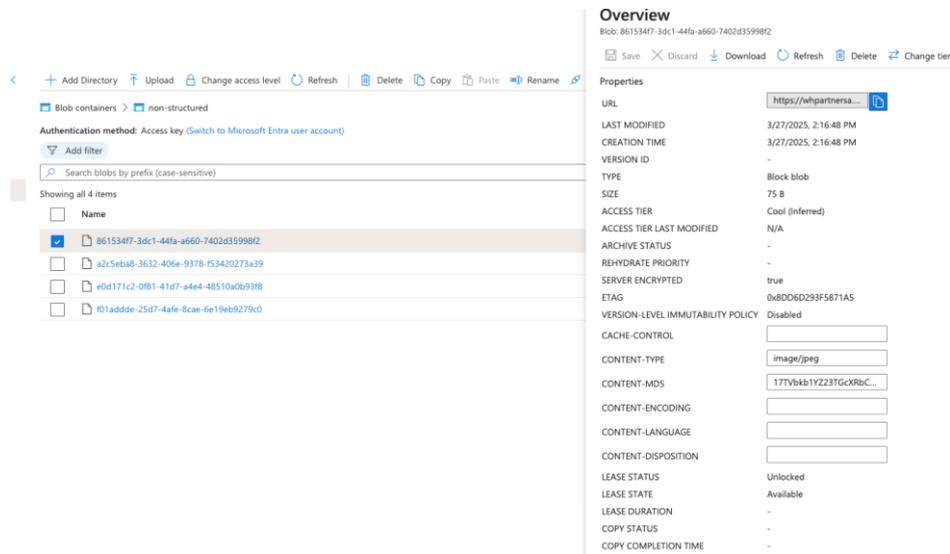
Clear all

Total received events: 159 → View next events

Sequence Number	Offset	Partition ID	Enqueued Time	Content Type	Message ID	Event Body
1	17179869184	0	Thu, Feb 06, 25, 01:51:58 PM GM...		b2c7056f-155e-42d3-ba...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=32.6 1738842708639970639
2	17179869394	0	Thu, Feb 06, 25, 01:51:58 PM GM...		c39bf3a9-9573-45f6-a8...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=25.4 1738842713637477218
3	17179869584	0	Thu, Feb 06, 25, 01:52:18 PM GM...		ad1d87d8-e934-4695-a...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=25.2 1738842728649585391
4	17179869784	0	Thu, Feb 06, 25, 01:52:18 PM GM...		5bfbf4ee-fd49-40d5-89...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=27.7 1738842733655546187
5	17179869984	0	Thu, Feb 06, 25, 01:52:38 PM GM...		56d1e87e-f650-41c6-a2...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=23.4 1738842748678620507
6	17179870184	0	Thu, Feb 06, 25, 01:52:38 PM GM...		6ffc01bf-fccf-4cdc-acad...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=23.3 1738842753679821525
7	17179870384	0	Thu, Feb 06, 25, 01:52:58 PM GM...		a4d10267-46fb-4abb-8...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=28.7 1738842768696556460
8	17179870584	0	Thu, Feb 06, 25, 01:52:58 PM GM...		c5abbd62-dfec-4e57-89...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=25.8 1738842773702014552
9	17179870784	0	Thu, Feb 06, 25, 01:53:18 PM GM...		7cc600fb-157e-4849-b5...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=25.8 1738842788723414687
10	17179870984	0	Thu, Feb 06, 25, 01:53:18 PM GM...		2cdeb02d-9e37-482d-9...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=20.8 1738842793729774983
11	17179871184	0	Thu, Feb 06, 25, 01:53:38 PM GM...		2ae371e2-8d3d-4630-8...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=31.2 1738842808745176996
12	17179871384	0	Thu, Feb 06, 25, 01:53:38 PM GM...		6a0e3fdd-723f-4a4c-ae...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=22.6 1738842813754219307
13	17179871584	0	Thu, Feb 06, 25, 01:53:58 PM GM...		edadd902-a93b-40a3-8...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=31.8 1738842828268971968
14	17179871784	0	Thu, Feb 06, 25, 01:53:58 PM GM...		4b14a161-e0ae-4cfc-bf...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=23.6 1738842833775030208
15	17179871984	0	Thu, Feb 06, 25, 01:54:18 PM GM...		07c5ad66-14ab-40aa-a3...	mqtt_consumer:host=adc98ba144b,topic=sensors/sensor-001 value=24.9 1738842848790140583

Event body: mqtt_consumer,host=adc98ba144b,topic=sensors/sensor-001 value=32.3 173884264923686044

Figure 7: Screenshot of received IoT data in Data Lake



Overview
 Blob: 8615347f-3dc1-44fa-a660-7402d35998f2

Save Discard Download Refresh Delete Change tier

Properties

- URL: https://whpartnersa...
- LAST MODIFIED: 3/27/2025, 2:16:48 PM
- CREATION TIME: 3/27/2025, 2:16:48 PM
- VERSION ID: -
- TYPE: Block blob
- SIZE: 75 B
- ACCESS TIER: Cool (Inferred)
- ACCESS TIER LAST MODIFIED: N/A
- ARCHIVE STATUS: -
- REHYDRATE PRIORITY: -
- SERVER ENCRYPTED: true
- ETAG: 0x8DD6D293F5871A5
- VERSION-LEVEL IMMUTABILITY POLICY: Disabled
- CACHE-CONTROL: []
- CONTENT-TYPE: image/jpeg
- CONTENT-MD5: 177Vbkb1Y2Z3TGcRbC...
- CONTENT-ENCODING: []
- CONTENT-LANGUAGE: []
- CONTENT-DISPOSITION: []
- LEASE STATUS: Unlocked
- LEASE STATE: Available
- LEASE DURATION: -
- COPY STATUS: -
- COPY COMPLETION TIME: -

Figure 8: Screenshot of received static data in Data Lake

4. Task 2.3 Charge: Automated charging & V2G protocols

Rapid automated charging is essential for enabling fully electrified services and autonomous electric vehicle (AV) operations. The objectives of T2.3 focus on developing an automated charging concept and a proof of concept for rapid charging, which can be inductive or robot-based, either in depots or on-route. By optimizing charging strategies, AVs can extend operational hours, maximizing financial and socioeconomic benefits. Key objectives include defining optimal on-route charging time, reviewing movable and stand-alone charger options, selecting suitable EV connectors (Type2, CCS, CHAdeMO) with required power capacity, assessing AV battery demands across seasons, and determining the level of automation. This requires testing across all seasons of the year, since the temperature affects electricity/battery consumption.

To satisfy the key points and objectives of T2.3, detailed research concerning the charging methods, safety of charging infrastructure and influence on the grid has been conducted and an automated charging architecture system has been developed to optimize the process of automated charging.

4.1. Literature Review Activities of Task 2.3

The current charging process for autonomous electric vehicles still requires human involvement. To enable truly autonomous operation, charging must also be automated. A study on autonomous electric vehicle charging examines key factors shaping charging strategies. It evaluates automated charging technologies using techno-economic criteria and assesses their impact on public transportation through a case study in Tampere, Finland. The study uses multiple methods, including a literature review and a techno-economic framework, to evaluate automated charging technologies based on efficiency, infrastructure needs, power capacity, and cost. A scenario-based analysis compares two public transport cases, examining the impact of different charging solutions on total charging time, service interruptions, costs, and overall cycle time.

Factors affecting the charging strategy of electric vehicles, Safety and Security of Charging Infrastructure and Their Influence on the Grid

A smart charging strategy is essential for running EVs and electric bus fleets efficiently and without disruptions. It needs to be customized to suit operational demands, which helps prevent delays. Important factors, like where, when, and how charging occurs, plus the cost, depend on the fleet's routines, battery specs, current charge levels, temperature, and electricity rates. For automated electric buses, charging must also be fully automated, limiting the types of charging setups and affecting how big the batteries need to be. Addressing all these elements properly ensures smooth and timely fleet operations. EVs use three main charging methods: conductive charging, inductive charging (Figure 9), and battery swapping (Nkembi et al. 2023). Charging power varies: electric cars range from 1 kW (home) to 150 kW, while battery electric buses typically charge between 50 kW and 600 kW (CUTRIC 2020).

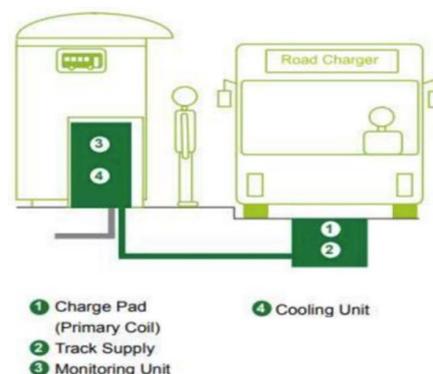


Figure 9: Charging components for Inductive charging (Source: Sawilla & Schuett, 2017)

Inductive charging is a suitable technology for automatically charging autonomous buses, offering power levels between 50–300 kW. It requires minimal installation space and performs reliably in harsh weather. However, it faces challenges such as lower efficiency compared to conductive charging, which can decrease further with poor alignment or increased air gaps between coils. Additionally, its high initial cost is a significant consideration before deployment. In contrast, battery-swapping offers a highly time-efficient solution for charging heavy-duty vehicles, with swap times comparable to refuelling a diesel bus. However, its implementation involves substantial upfront costs and considerable land requirements, which pose key challenges.

High-power charging for autonomous electric buses significantly impacts the electricity grid, consuming much more energy than residential demand. Depot charging, usually conducted at night, minimizes stress, stabilizes voltage, and reduces power losses due to medium-voltage connections. Conversely, en-route charging presents challenges such as voltage fluctuations, harmonic emissions, and grid stability concerns, requiring larger transformers that suffer greater power losses and thermal stress. Safety risks include vandalism, theft, unauthorized charger use, disrupting service, and increasing costs. Inductive charging systems pose electromagnetic risks, necessitating safe distance guidelines. Additionally, security concerns include cyber threats, as charging stations store sensitive data and could be targeted by hackers. Lastly, thermal runaway in lithium-ion batteries is a fire hazard, emphasizing the need for strict monitoring and safe charging practices.

Cybersecurity risks in charging stations stem from handling sensitive data like payment details and vehicle identifiers, making them targets for hackers. Cyber-attacks can allow unauthorized access, leading to data theft, service disruptions, or system manipulation. Securing these systems is crucial for privacy and operational integrity. Additionally, thermal runaway in lithium-ion batteries poses a fire hazard, where excessive heat triggers flammable and toxic gas release. Overcharging or improper discharging are primary causes, necessitating strict monitoring and control measures for safe charging. Typically, mitigation strategies include the following:

- **Physical threats** are addressed with lighting, surveillance cameras, geofencing, and immobilization systems. Restricted zones help safeguard pedestrians from electromagnetic exposure in inductive setups.
- **Cybersecurity** is strengthened through authentication protocols and access controls, ensuring only verified entities can interact with the charging network.
- **Fire hazards** are mitigated by maintaining safe battery charge levels (20–80%) and incorporating effective thermal and battery management systems.

4.2. Development Activities of Task 2.3

Architecture

To facilitate robust and efficient automated charging, particularly for automated vehicles, the following system architecture was defined. This architecture is structured around five main elements, ensuring their seamless interaction and supporting a holistic approach to the automated charging ecosystem. The designed system architecture encompasses the following:

The Automated Vehicles

The design outlines automated vehicles equipped with advanced sensors and interfaces to support seamless automatic charging. Key vehicle-side technologies include high-precision positioning systems (like GPS RTK, LiDAR, and cameras) for accurate navigation to charging stations. Charging interfaces may include pantograph systems, inductive coils, or conductive connectors designed for automated operation. Central to this setup is the On-Board Unit (OBU), which manages the vehicle's state of charge (SoC), initiates charging requests, and handles authentication and coordination with the charging infrastructure.

The Charging System

The charging system, designed for infrastructure, ensures reliable and safe energy transfer to automated vehicles. It includes physical components like retractable pantographs, inductive pads, or automated connectors, paired with power electronics for efficient energy delivery. Local control units manage the charging process, enforce safety, and communicate with vehicles. Importantly,

the system also integrates grid-responsive features to manage power consumption and respond to signals from the electric grid.

Communication Protocol (Vehicle-to-Charging System)

A dedicated communication protocol underpins the interaction between the autonomous vehicle and the charging system. The design includes both hardware interfaces (like transceivers and antennas) and software layers that ensure secure, reliable data exchange. This supports key functions such as vehicle ID, state of charge negotiation, alignment guidance, charging control, and fault reporting, all of which are vital for automating and securing the charging process.

The Electric Grid Interface

The system architecture intentionally integrates with the electric grid, treating it as a vital external component. Key infrastructure, such as power electronics and station controllers, is engineered to ensure safe, compliant, and well-regulated interaction with the grid.

Fleet Management System

The Fleet Management System (FMS) acts as the central coordinator for automated charging across an autonomous vehicle fleet. Through bidirectional communication with both vehicles and charging infrastructure, the FMS oversees strategic operations, scheduling charging, authorizing sessions, authenticating vehicles, monitoring system status, and collecting data for performance analysis. This centralized orchestration is essential for efficient asset use and seamless integration with fleet logistics.

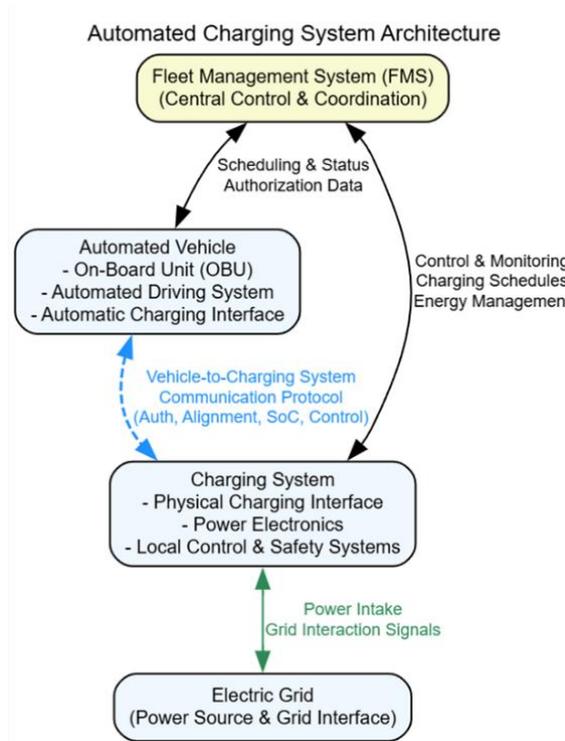


Figure 10: Automated charging system architecture

Automated Charging Process Flow

To ensure clarity and a standardized approach for the interaction between automated vehicles and charging infrastructure, the following sequence (Figure 11) for the automated charging process was developed. The sequential steps of this process are described below.

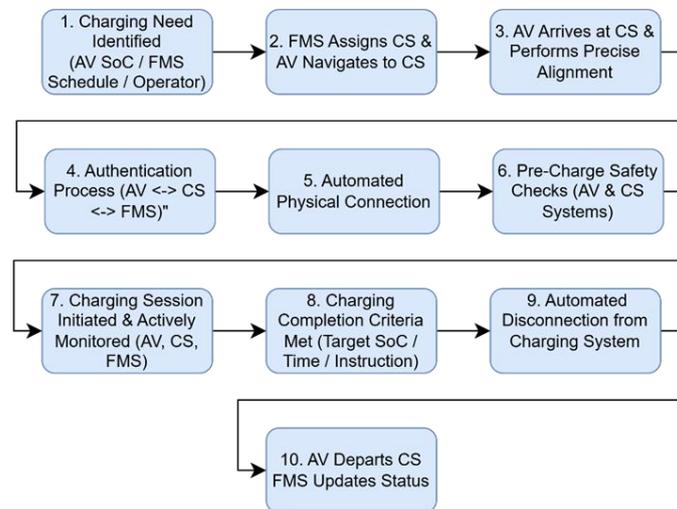


Figure 11: Charging process steps

Charging Need Identified (AV SoC / FMS Schedule / Operator): The process is initiated when a requirement for charging is determined. This trigger can originate from the AV itself (e.g., its State of Charge, falling below a predefined threshold), from a pre-set FMS operational schedule, or through a directive from a remote operator monitoring the fleet via the FMS.

FMS Assigns CS & AV Navigates to CS: Upon identification of the charging need, the FMS assumes responsibility for selecting an appropriate and available Charging System (CS). Once assigned, the FMS communicates the charging task to the AV, which then autonomously navigates to the specified charging station location.

AV Arrives at CS & Performs Precise Alignment: Upon arrival at the designated charging station, the AV signals its presence to the local CS. Subsequently, the AV executes precise manoeuvring routines, to achieve accurate alignment with the physical charging system.

Authentication Process (AV <-> CS <-> FMS): Before any physical connection or power transfer occurs, a secure authentication sequence is performed. This involves a communication exchange between the AV and the CS, with the FMS acting as a validation authority or identity provider.

Automated Connection: Following successful authentication, the connection between the AV and the CS is established. This could involve the deployment of a pantograph, the engagement of a robotic arm with a conductive connector, or the confirmation of optimal proximity for inductive power transfer, depending on the specific charging technology employed.

Pre-Charge Safety Checks (AV & CS Systems): With the physical connection established, both the AV's onboard systems and the CS controllers conduct a series of safety checks. These verifications ensure system integrity, proper grounding, insulation levels, and overall readiness for safe power transfer.

Charging Session Initiated & Actively Monitored (AV, CS, FMS): At this stage the charging session is initiated, and electricity begins to flow to the AV's battery. Throughout the session the voltage, the current, power, the energy transferred, and battery temperature are actively monitored by the AV, the CS, and reported to the FMS for oversight and data logging.

Charging Completion Criteria Met (Target SoC / Time / Instruction): The charging session continues until a predefined completion criterion is met. This is commonly the AV's battery reaching its target SoC, the expiry of a scheduled charging duration, or a specific instruction from the FMS.

Automated Disconnection from Charging System: Upon meeting the completion criteria, the power transfer is safely ramped down and stopped. The automated physical connection between the AV and the CS is then disengaged in a controlled sequence.

AV Departs CS; FMS Updates Status: With the charging process complete and the vehicle disconnected, the AV signals its readiness to depart. The FMS updates the AV's status (new SoC and availability for operational tasks), and the AV autonomously exits the charging spot.

5. Task 2.4 Automate: Advanced driving assistance systems and remote control centers for AVs

Task 2.4 is examining advanced driver assistance systems and AI-driven real-time planning for autonomous vehicle fleets. It consists of three subtasks.

Subtask 2.4.1 develops the state-of-the-art Remote Control Center, including integrating various technological elements into the RCC control room at a fundamental stage. This task is led by REMOTED to develop the flowchart of the Remote Control Center (RCC) for the AVs and its functionalities to enable fully driverless operation of automated vehicles. This is an initial process for remote control operations for several scenarios in which operator interventions are required. Subtask 2.4.2 focuses on investigating the modular bus driving assistance system. NEXT has completed the Navigation Testing, and the Braking and Gearing Control. Also, in this subtask, UNINA solves the Docking manoeuvre problem with the assistance of an ADAS controller. In subtask 2.4.3, NTUA developed a model for the real-time planning of autonomous fleets. The model considers a bus fleet of modular units that are autonomous and can skip stops with low passenger demand to increase efficiency.

Regarding the future steps, NEXT plans to implement higher-level behavioural planning, particularly for end-to-end summoning and parking operations, and perform a hardware update. NTUA plans to shift its focus to flexible real-time rescheduling of the modular bus system.

5.1. Subtask 2.4.1 Remote control center for AVs

While Level 5 Automated Driving Systems (ADS) are intended to operate in all conditions and environments, no current automated vehicle (AV) has yet been able to meet this goal. As Level 4 and 5 AVs are introduced, situations may occur that exceed their programmed capabilities (Goodall 2020). Remote operation has emerged as a practical interim solution (Kettwich et al., n.d.,2024), enabling a human operator to monitor the vehicle and intervene when necessary to maintain safety and ensure continuous travel.

Remote operation center for AVs

A Remote Operation Center (ROC) structure, as presented in Figure 12, aims to support the safe and efficient oversight of Highly Automated Vehicles (HAVs), breaking down its complex, multidisciplinary design into key elements such as infrastructure, human-machine interaction (HMI), operator roles, and regulatory compliance. The flowchart highlights that ROC development is a collaborative, human-centered process requiring domain-specific expertise in areas like data protection and network management, and organizes tasks according to levels of remote monitoring, assistance, management, and teleoperation based on operational needs and

technological constraints such as latency and sensor reliability. Emphasis is placed on ergonomic workstation design, SA (situational awareness) management, and seamless control transitions, all essential for maintaining reliable vehicle support. It further underlines the need for robust IT infrastructure, real-time communication, and well-defined operator protocols, enabling the ROC to monitor fleets, manage control transitions, respond to failures, and communicate with passengers and authorities. Depending on network conditions, different levels of remote engagement are feasible teleoperation requiring low-latency while monitoring and assistance allow for greater flexibility. The ROC must also support dynamic fleet management, hazard alerts, and external coordination, with operator training in HMI systems, driving principles, and high-pressure multitasking deemed critical. Ultimately, the flowchart ensures that ROC development aligns with standards such as SAE levels, GDPR, and national regulations, promoting safety, trust, and operational efficiency.

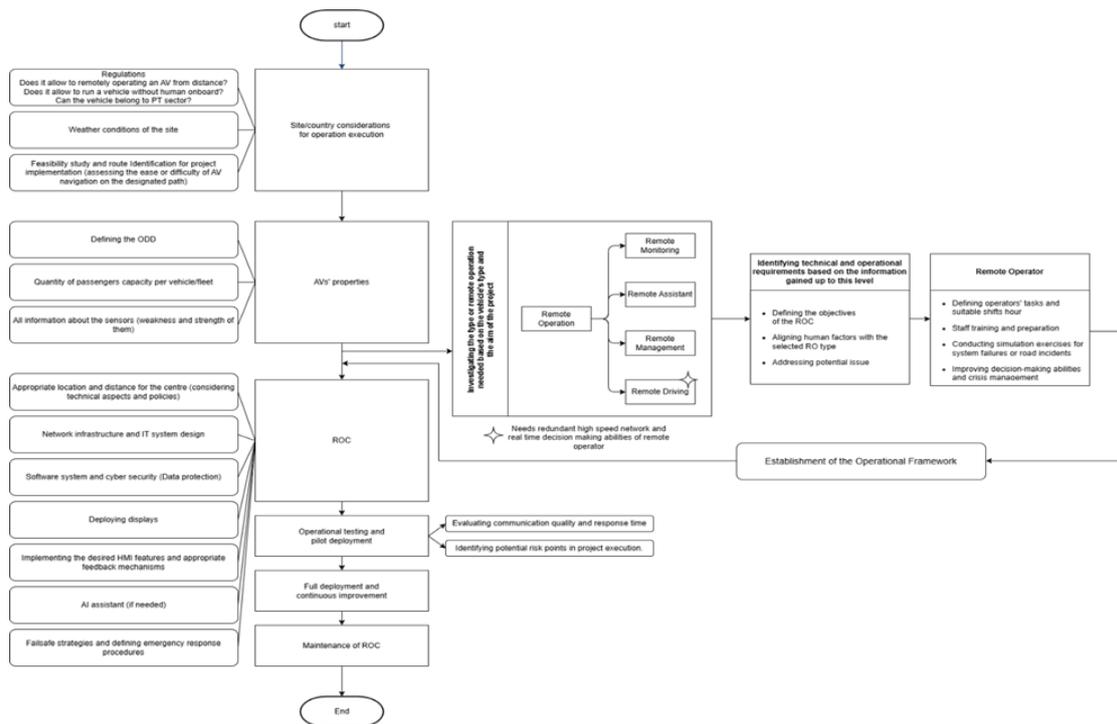


Figure 12: ROC flowchart

Advanced Driver Assistance System

As noted by (Veitch & Andreas Alsos, 2022), the rapid advancement of AI highlights the need for a system-level approach in remote operations, particularly as AI becomes central to managing AV fleets. While no global standard exists for how many vehicles a single operator can oversee, experts support harmonized regulations at the UNECE level. With AI assistance, one operator may eventually manage up to 40 AVs under routine conditions and up to 10 in real-time scenarios. In emergencies scenarios is requiring direct control, remaining AVs should be immediately reassigned to maintain safety. Efficient handling of such situations depends on intelligent system design and strong operator collaboration. To ease workload and improve oversight, expert-backed recommendations include prioritizing alerts, providing intuitive 360-degree video, dynamically adjusting supervision based on traffic or route complexity, and using AI to categorize assistance requests by urgency. Ergonomic workstations, streamlined procedures, and robust communication infrastructure are also essential, while fallback strategies and AI-driven autonomy ensure safety during network disruptions.

5.1.1. Main Activities

The RCC concept is developed to support the operation of vehicles equipped with Level 4 automated driving features, as defined by the Society of Automotive Engineers in Standard SAE J3016. While such vehicles are capable of managing the entire dynamic driving task within their Operational Design Domain (ODD), human oversight remains necessary to ensure safe, reliable, and compliant operations in public transport settings.

The need for an RCC stems from three primary factors. First, legislative requirements in many jurisdictions, including EU countries like Finland, mandate continuous human supervision of automated vehicles in public spaces, even when direct intervention is seldom required. Second, despite the high level of technological maturity, current Automated Driving Systems (ADS) may still encounter edge cases or complex scenarios beyond their ODD, where human intervention is essential to maintain safety and service continuity. Third, in a public transport context, the RCC plays a critical role in customer service by enabling communication with passengers, providing assistance, and managing unexpected events to enhance the overall user experience. The RCC concept developed within the metaCCAZE project is designed to effectively coordinate fleet operations while addressing the specific challenges posed by these legislative, technical, and service-related demands.

Concept overview

The system concept adopts a distributed architecture composed of multiple remote control centres to supervise the automated vehicle fleet, aligning with EU legal requirements for continuous human supervision of the automated vehicle fleet. This approach was chosen primarily in response to legal requirements in most EU countries, which mandate that automated vehicles must remain under continuous human supervision.

To ensure resilient and uninterrupted supervision of automated vehicle fleets, the system adopts a distributed architecture with multiple RCCs operating in parallel rather than relying on a single centralized unit, which would pose significant risks in the event of technical failures or connectivity disruptions. Each RCC is primarily responsible for a subset of the fleet while also holding secondary responsibility for vehicles from another center, enabling swift failover and continuity of service if one RCC becomes unavailable. A central operations management function coordinates the overall system, managing real-time task allocation, load balancing, and operational integrity. Additionally, each RCC is required to maintain direct communication with emergency services to enable a timely response to critical incidents, although the precise technical setup of these connections remains under development.

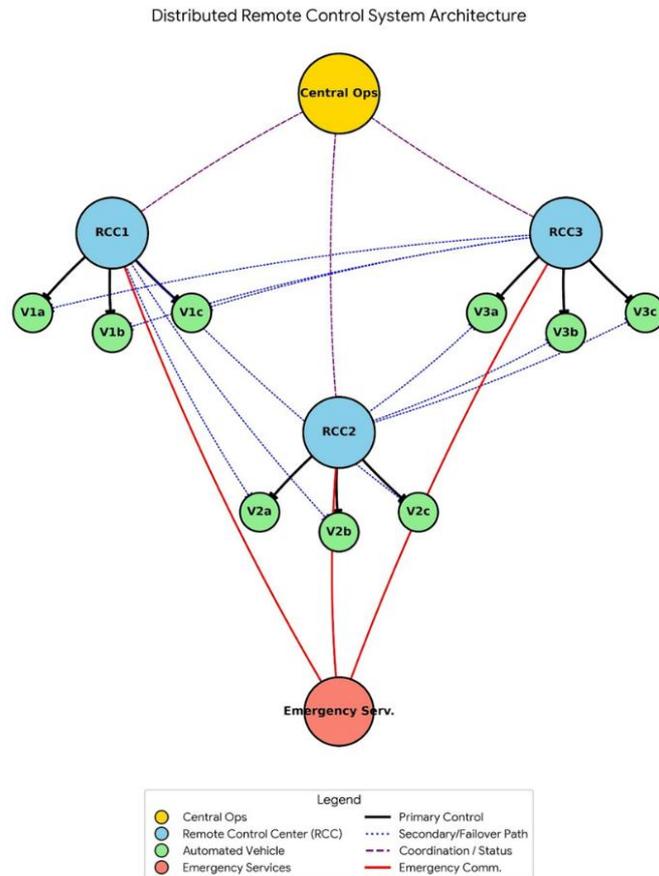


Figure 13: Conceptual Diagram of the Distributed Remote Control System Architecture

Distributed Architecture and Resilience Mechanisms

The distributed architecture, visualized conceptually in Figure 14, illustrates the conceptual structure of the distributed RCC architecture, which is central to the system’s operational resilience. Unlike a centralized control model, this approach involves multiple independent RCCs operating in parallel, directly mitigating the critical risk of a single point of failure. Each RCC is assigned a primary group of vehicles and serves as a secondary (backup) supervisor for another set, typically managed by a different RCC. This dual-responsibility setup, visualized through solid black lines (primary control) and dotted blue lines (secondary control), enables a predefined failover mechanism: if one RCC becomes non-operational, its supervisory duties are automatically reassigned to its designated backup. To avoid overloading a single backup center, supervisory responsibilities can be further distributed across multiple RCCs, with a specific backup RCC coordinating the transition. The Central Operations Management function oversees the entire RCC network, dynamically allocating responsibilities and balancing resources, both in regular and failover conditions. This coordination is shown through dashed purple lines, representing bidirectional communication between Central Ops and each RCC. Additionally, the solid red lines represent direct communication channels between each RCC and emergency services, ensuring a timely response to critical incidents.

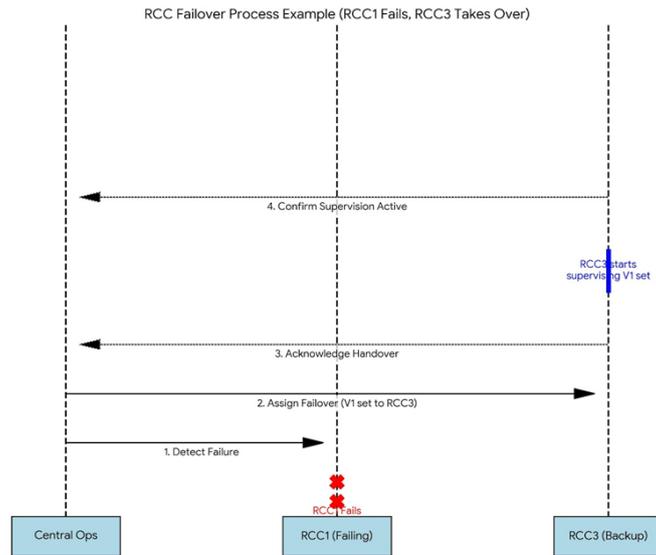


Figure 14: RCC Failover Process Example (RCC1 Fails, RCC3 Takes Over)

The sequence diagram presented in Figure 15 provides a sequence diagram depicting the RCC failover mechanism, a critical part of the system’s resilience strategy that ensures continuous autonomous vehicle supervision in the event of an RCC failure. The diagram outlines a scenario in which RCC1, responsible for the V1 vehicle set, becomes non-operational due to a technical issue or connectivity disruption. The failure is first detected by Central Operations via heartbeat monitoring or alerts, which then issues a directive to RCC3, the designated backup, to take over supervision. RCC3 acknowledges the command, initiates its preparation phase by loading relevant configurations and activating communication systems, and upon readiness, confirms the handover back to Central Operations. This process, represented through numbered message arrows and lifelines, demonstrates a smooth and timely transfer of supervisory responsibilities, maintaining uninterrupted service, ensuring passenger safety, and upholding compliance, all core tenets of the system’s robust distributed architecture.

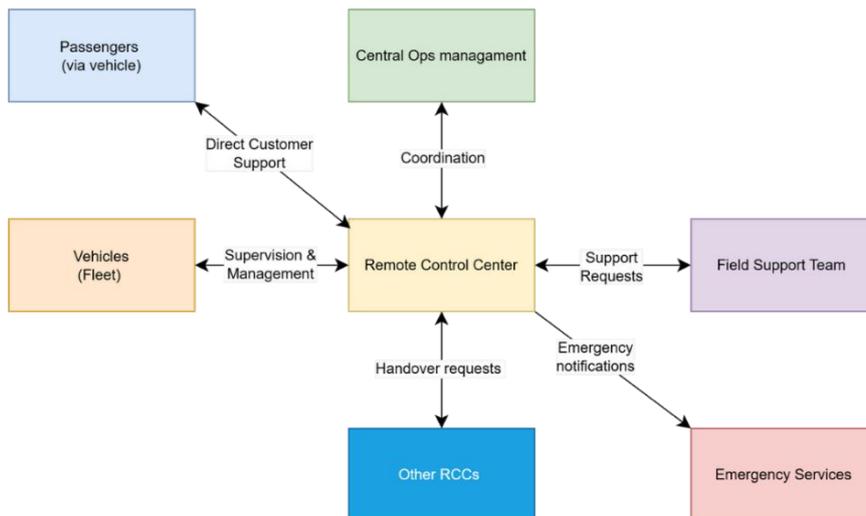


Figure 15: RCC Functional Block Diagram - Key Interactions

In addition, Figure 15 illustrates the RCC's central role in autonomous vehicle operations, showing key information flows between its core functions and stakeholders. The RCC exchanges real-time data with vehicles, manages passenger communication, coordinates with Central Operations for

strategic oversight, and interacts with field teams for on-site support. It also connects with other RCCs for supervision handover and maintains a direct line with emergency services to ensure safety and service continuity.

Operational Scalability and Field Interaction

The economic feasibility of deploying automated vehicles in public transport heavily relies on achieving a favorable Vehicle-to-Operator (V/O) ratio. While legislation mandates supervision, the specific requirements may allow a single Remote Operator to oversee multiple vehicles simultaneously. Based on technical input, ratios up to 10:1 might be achievable under ideal conditions with mature technology. However, a pragmatic, phased approach is necessary. Initial operations, especially with new vehicle types or in complex environments like Tampere during winter, will likely start at 1:1. The operational target is to reach a minimum V/O ratio of 4:1 within approximately six months of starting operations to ensure the service's scalability and cost-effectiveness. This ratio is not static; it will be dynamically assessed and adjusted based on accumulated operational data, system performance improvements, prevailing environmental conditions, and any evolving legislative constraints.

Although the RCC manages remote operations, it functions in close collaboration with the Field Support team. Certain situations inevitably require physical presence near the vehicle. These include scenarios where the ADS encounters an unresolvable issue requiring manual intervention, response to accidents, or specialized maintenance tasks. Field Support personnel also typically handle routine depot activities like vehicle preparation, charging supervision, and cleaning. An initial estimate suggests requiring one field support person per four vehicles, but this figure is subject to refinement based on real-world operational needs. Seamless communication and coordination between the RCC operators directing remote actions and the Field Support operators executing physical tasks are paramount for efficient service delivery.

Infrastructure and Resource Summary

Establishing and operating the distributed RCC network necessitates specific technical and human resources. Each RCC requires a dedicated physical facility equipped with specialized operator workstations, including necessary hardware (computers, large displays, communication equipment) and the core RCC software platform. High-availability, low-latency, and redundant network connectivity to the vehicles and other operational centers is critical. Furthermore, uninterruptible power supply (UPS) systems are mandatory to ensure continuous operation during power disruptions.

Human resources include trained Remote Operators and Backup Operators, staffed to meet the target V/O ratios and cover operational hours, breaks, and potential failover scenarios. The possibility exists for personnel to have dual roles, potentially serving as field operators when not needed in the RCC. Oversight is provided by the Central Operations Management function, estimated to require 0.5 to 1 Full-Time Equivalent (FTE) for supervising fleets larger than four vehicles. Fleet support staffing depends on fleet size and operational parameters, estimated at 1 FTE for fleets of 1-4 vehicles and 2 FTEs for fleets of 4-10 vehicles under typical operating conditions (e.g., 6 hours/day, within ODD). RCC operator staffing is estimated at 1 FTE per fleet under similar typical conditions. These estimates form a baseline for resource planning.

5.2. Subtask 2.4.2 Advanced Driver Assistance System

This task aims to design, simulate, and deploy a full-stack autonomous navigation system for a modular pod vehicle. The goal is to enable autonomous parking, summoning, and docking manoeuvres in a controlled parking lot environment.

This activity directly supports the roadmap for scalable self-driving solutions in last-mile logistics and passenger transport. metaCCAIZE contributes significantly to higher-level objectives by delivering a robust, real-world autonomous driving system that adheres to safety standards and accurately executes task objectives.

5.2.1. The Robot: Next Modular Vehicles' Pod (Third Generation)

The third-generation Next Modular Vehicles pod was the main implementation choice, as illustrated in Figure 78 in the Annex of this deliverable report. This entirely custom-built vehicle, developed three years ago, is homologated for urban driving. With dimensions of 2.80 m in length, 2.40 m in width, and 2.90 m in height, it offers a compact yet versatile form factor ideal for urban environments. It maintains the footprint of a small urban car while allowing passengers to stand upright.

The vehicle's hardware specifications include a custom electric drivetrain with dual Elaphe in-wheel motors, delivering 2×45 kW of nominal power and a top speed of 65 km/h. The power system consists of a custom-made 40 kWh LiFePO₄ (Lithium Iron Phosphate) battery pack operating at 400 V, along with a 12 V service battery. A DC-DC converter with a maximum power output of 2.5 kW supplies energy to all onboard systems. The chassis features four-wheel independent suspension and a tight turning radius of just 3.7 m, ensuring exceptional manoeuvrability in confined spaces, a critical requirement for parking and docking operations.

Sensors supply and placement

Autonomous vehicles require a comprehensive sensing suite to perceive their environment accurately and safely. The sensor selection process focused on creating redundant perception capabilities with complementary strengths, ensuring the system can operate reliably in various environmental conditions. The sensor placement strategy was designed to minimize blind spots while protecting sensitive equipment from environmental factors. We conducted multiple simulation tests to verify coverage patterns before finalizing the physical installation, as illustrated in Table 9 in the Annex of this deliverable report. The sensors placed in our vehicle are presented in Figure 16.



Figure 16: Sensors Placement on our prototype

Long-range Infrared Sensors: Lidars

Due to the shape of the Next vehicle and the requirements of the project, it would have been very impractical to use a single 360° Lidar and have complete coverage of the vehicle surroundings. Therefore, we used different types of Lidar for every side, as illustrated in Table 10 in the Annex of this deliverable report.

This allowed us to have complete coverage of the area surrounding the vehicle with minimal blind area.

Visual and Inertial Sensors: Cameras & IMU

The ZED2i Camera is a wide-angle stereo camera with integrated IMU, capable of producing high resolution point clouds up to 100Hz. Its proprietary VIO algorithms can produce high precision Odometry. We use two, one in the front and one in the rear of the vehicle. The one in the rear is useful for reverse parking, with or without the aid of fiducial markers.

Global Localization: GPS

The vehicle also sports an ArduSimple simpleRTK2B, although we are not currently using GPS for navigation. In the future we plan to integrate it in the navigation stack. Furthermore, the ZED camera VIO (Visual Inertial Odometry) natively supports GPS sensor fusion to enhance precision and greatly reduce odometry loss.

Sensors Coverage

For Autonomous Modular Vehicles (AMVs), comprehensive environmental sensing is critical to safe operation. The initial requirements specified complete 360-degree coverage with sufficient redundancy to handle sensor failures and varying environmental conditions, as illustrated in Figure 79 in the Annex of this deliverable report. The target specifications included:

- Detection of obstacles as small as 10cm in height at distances up to 30 meters in all directions.
- Minimum 30% overlap between adjacent sensors to ensure no blind spots.
- Reliable performance in various lighting conditions (day/night) and weather scenarios (clear/rain/fog).
- Sufficient vertical field of view to detect both ground-level obstacles and overhead hazards.
- Real-time processing capabilities with maximum latency under 100ms.

Through iterative testing and placement optimization, the actual coverage exceeded these initial requirements in most areas, as illustrated in Figure 17. Current performance metrics show:

- Complete 360-degree coverage with 25% sensor overlap in critical zones (front and rear).
- Reliable detection of obstacles down to 8cm height at distances up to 35 meters in the forward field.
- Complete coverage of the parking space envelope during manoeuvring.
- Vertical detection from ground level up to 2.5 meters in height.
- Processing latency averaging 75ms under normal operating conditions.

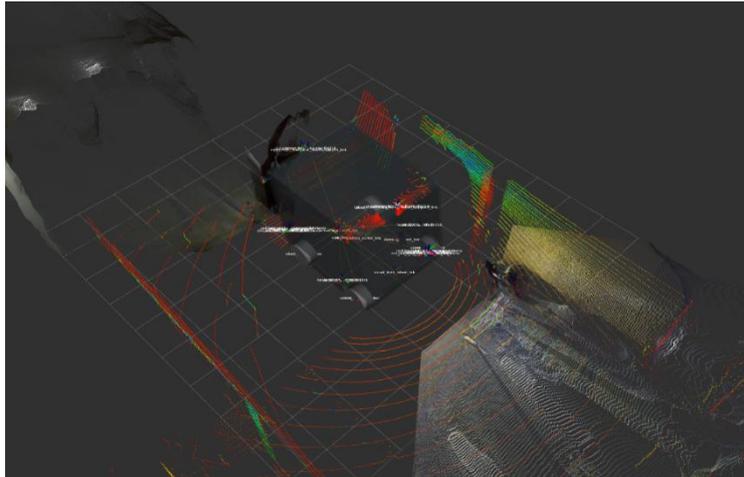


Figure 17: Sensors Coverage bird view from the real robot

The multi-sensor approach has proven particularly valuable in edge cases, where certain sensors may be compromised (e.g., cameras in direct sunlight or lidars in heavy precipitation). The redundant coverage ensures the vehicle maintains situational awareness even when individual sensors operate sub-optimally.

Initial Simulation Campaign

Before deploying our autonomous system on the physical vehicle, we conducted extensive simulations to validate our approach and minimize risks. Our simulation strategy was two-pronged: first, creating an accurate digital twin of our vehicle and operating environment, and second, developing proof-of-concept navigation capabilities in this simulated world.

Our digital twin implementation accurately simulates our test parking lot with precise dimensions, surface properties, obstacles, and parking spaces. The simulated vehicle model incorporates exact physical parameters from our real-world pod, including weight distribution, motor torque curves, steering geometry, and tire friction characteristics. Each sensor is modelled with appropriate noise profiles and failure modes to ensure our algorithms are robust against real-world imperfections. The digital twin serves multiple purposes throughout our development lifecycle, like algorithm development and testing, parameter tuning, and edge case testing.

Proof of Concept for Navigation

With our digital twin environment in place, we developed an initial navigation capability to validate our approach, as presented in Figure 18. The proof-of-concept demonstration included:

- Basic localization using simulated sensor fusion
- Simple path planning with obstacle avoidance
- Simulated parking manoeuvres in designated spaces
- Speed control and emergency stopping capabilities

These initial tests helped identify critical challenges in our design, particularly around sensor placement and occlusion handling. By addressing these issues in simulation, we avoided costly redesigns after hardware integration. The proof of concept also served as a valuable demonstration tool for stakeholders, visualizing the intended capabilities before physical implementation.

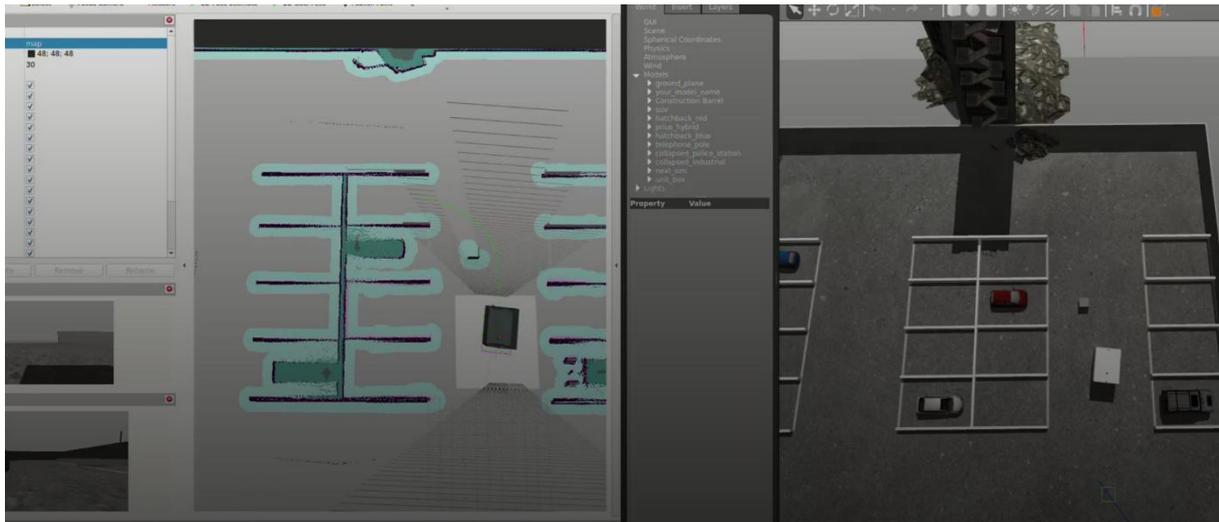


Figure 18: Simulated Navigation

Kinematics and physical interpretation

Understanding and modelling the vehicle's physical behaviour is essential for precise control and accurate prediction. Our kinematics implementation considers the unique characteristics of our pod vehicle, including its front-wheel steering geometry and rear-wheel drive configuration. We developed a comprehensive vehicle model incorporating an Ackermann steering geometry with variable turning radius, acceleration and deceleration profiles accounting for motor torque curves and vehicle inertia properties affecting cornering behaviour.

This physics-based approach enables our control algorithms to accurately predict the vehicle's response to control inputs, allowing for smoother trajectories and more precise manoeuvres. The model parameters were initially estimated from CAD data and subsequently refined through calibration tests on the physical vehicle. Our motion controllers use this kinematic model to convert high-level trajectory commands into appropriate steering angles and throttle/brake commands, with continuous feedback loops ensuring adherence to the planned path. Special consideration is given to low-speed manoeuvres where non-linearities in steering response become more pronounced.

Localization and Mapping

Precise localization is fundamental to autonomous navigation. Our approach combines multiple techniques to achieve robust position estimation regardless of environmental conditions.

After many trials and errors, we confirmed the unreliability of the wheel odometry. In order to provide robust localization, our to-go strategy was to use the precise Visual Inertial Odometry provided by our ZED2i depth cameras, and to further fuse it with 1cm-precision level RTK GPS. RTK service was needed to improve the GPS precision and relies on a web service (Hexagon' HxGN SmartNet) that retrieves calibration data from the surrounding antennas in open spaces. To fuse the VIO and RTK GPS, we implemented an extended Kalman filter to optimally combine these measurements, weighting each source according to its estimated reliability in different conditions. The system achieves sub-10cm positioning accuracy in our test environment, sufficient for precise parking manoeuvres.

Regarding mapping, Occupancy Grid maps of the parking lot were generated using Graph Slam algorithms comprised in the ROS2 environment. Map maintenance occurs continuously in the background, with detected changes evaluated for permanence before incorporation into the

reference map. This allows the system to adapt to environmental changes while rejecting temporary obstacles. Global and Local Costmaps are created real-time to update the free space around the vehicle.

5.2.2 Advanced Driver Assistance System

Research efforts have focused on the development of a simulation framework capable of replicating the approach and docking manoeuvres between two Next pods, one acting as the leader and the other as the follower. The framework was coded in the Python programming language and designed to be as general and scalable as possible to allow testing of any type of Advanced Driver Assistance System controller, referred to as “ADC” in this section, and replication of various dynamic scenarios. With minimal and intuitive modifications to the code settings, it is possible to adjust the speed of the simulated pods, their braking capabilities, as well as the distance at which the ADC begins to operate. Additionally, the framework allows for the simulation of disturbances introduced by the leader during the approach manoeuvre. This feature is particularly relevant for studying the responsiveness of the ADC responsible for docking, especially in scenarios involving sudden braking by the leading pod. These scenarios are very common under real-world conditions, especially in urban settings with heavy traffic, and therefore allow for testing the ADC's safety performance in realistic situations. Overall, the framework enables the assessment of the robustness of ADCs under various operating conditions using Sobol sampling, as well as the evaluation, through sensitivity analysis, of the factors that are most critical to the safety of the docking manoeuvre.

Docking manoeuvre with the assistance of an ADAS controller

The goal of the ADC is to support the human drivers during a docking manoeuvre between two pods. Specifically, the controller manages the dynamics of a follower pod, travelling at a speed v_f , aiming to dock with a leader pod, travelling at a speed v_l , with $v_l < v_f$. Given the critical nature of this manoeuvre, the assistance of an autonomous controller could ensure smoother execution and enhanced safety. It is assumed that the ADC takes full control of the follower pod's dynamics once it reaches a certain distance from the leader and maintains this control until the docking manoeuvre is completed and the two pods move as a single unit.

Figure 19 provides a schematic representation of the problem. In the figure, the follower pod is depicted as a red block, while the leader pod is shown as a green block. The line connecting the two blocks resembles the distance between the pods, whereas the blue arrow on the right represents the driving direction. The three vertical lines indicated as A, B and C represent reference points used to identify key variables in the problem. Specifically, when the net distance between the 2 pods (bumper-to-bumper distance) is lower than the distance AC, the ADC activates and begins to control the follower pod's dynamics to perform the docking manoeuvre. We refer to the AC distance as “triggering distance”. At the current stage, this distance is upper-bounded by the sensing range of the sensors installed on the pod. Therefore, if the sensors can detect a vehicle at a maximum distance of x meters, the triggering distance AC cannot exceed x meters. Conversely, BC is referred to as the 'docking distance'. It is a design parameter of the pods, as it represents the length of the retractable hook that physically enables the connection between two pods. When the distance between the two pods is smaller than BC, the pods are considered docked. However, reducing the inter-pod distance below BC is not sufficient to ensure that the docking manoeuvre has been completed safely. A key parameter is the speed at which docking occurs or, more precisely, the relative speed (ΔV) between the two pods at the moment of docking. This ΔV must be controlled and kept below a safety threshold, which is also a design parameter and will be referred to as the “docking speed” (DS). It follows that the ADC's task is precisely to control the

follower’s speed in order to ensure an approach to the leading pod such that the distance BC is reached with a relative speed to the leader that is less than or equal to the docking speed. If this condition is not met, the manoeuvre must be considered unsafe.



Figure 19: Schematic representation of a docking manoeuvre

The steps of the problem and its possible outcomes are presented in Figure 20.

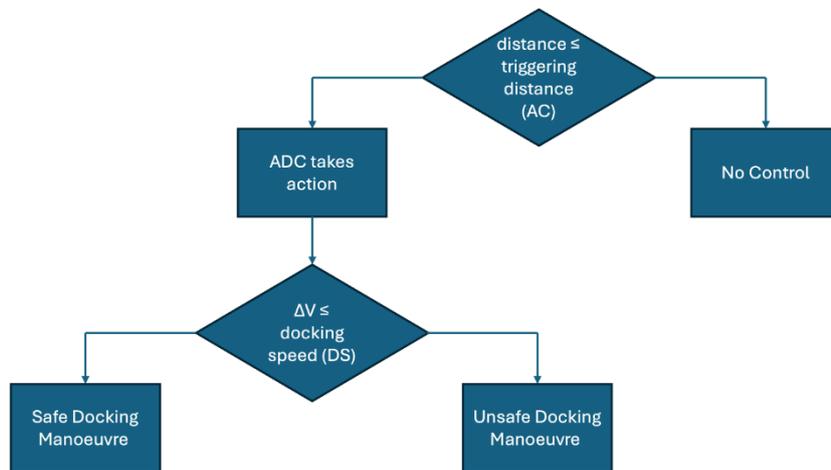


Figure 20: Problem Statement

Problem Constraints

As mentioned in the previous subsection, there are technical and design constraints related to the sensor range and the length of the retractable hook, which respectively impose an upper bound on the triggering distance and determine the docking distance. However, there are also physical constraints. Specifically, to ensure a good level of comfort, the acceleration/deceleration of the pods is limited to 1.5/-1.5 m/s². Indeed, it should be noted that the pods are vehicles intended for passenger transport, and passengers might even be standing inside the pod. Therefore, overly abrupt accelerations/decelerations could compromise passenger safety.

Following the introduction of the deceleration constraint, limited to -1.5 m/s² (d_{max}), another constraint on the triggering distance emerges, this time as a lower bound. Indeed, since safety condition implies that the follower must reach distance BC with a ΔV lower than DS, the maximum deceleration the follower can apply defines the minimum distance respect to the leader at which braking must begin in order for the safety condition to be satisfied. We refer to this minimum distance as the “Minimum Safety Distance (MSD)”. Figure 21 shows an illustration of the MSD in the speed–time diagram.

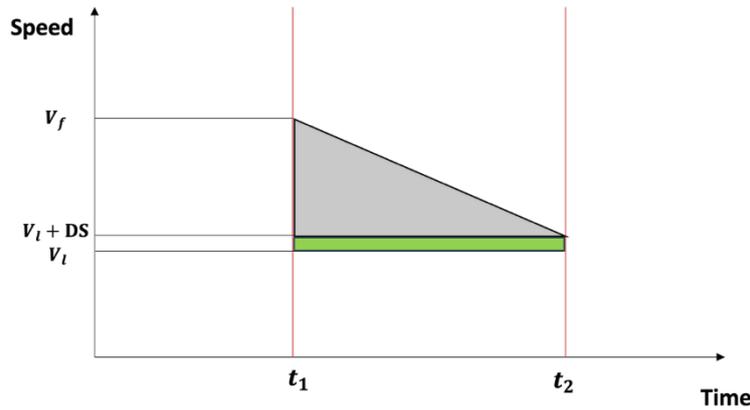


Figure 21: Graphical representation of the Minimum Safety Distance (MSD)

In Figure 21, t_1 represents the instant when the follower, initially traveling at speed v_f , begins to apply the maximum deceleration d_{max} . t_2 is the time when the follower reaches the distance BC from the leader, with a speed equal to $v_l + DS$. It follows that the MSD corresponds to the area of the grey triangle. By applying the equations of motion, the MSD can be calculated, whose formulation is presented in Equation (1), whereas Equation (2) express the term t with:

$$MSD = -DV * t - BC - \frac{1}{2} * d_{max} * t^2 (1)$$

$$t = \frac{DS - DV}{d_{max}} (2)$$

where: ΔV is the difference $v_f - v_l$, BC is the docking distance, d_{max} is the maximum deceleration the pod can apply, and DS is the docking speed.

Simulation Framework

The simulation framework we designed allows for the implementation of an ADC and the evaluation of its effectiveness in ensuring a safe merging manoeuvre. Below, we present the implementation of a PID controller as the ADC, along with the corresponding sensitivity analysis.

PID Controller

The first ADC proposed for the pods' dynamic control was a PID controller, where PID stands for Proportional, Integral and Derivative. The structure of the PID controller implemented in the present analysis, along with the values of its coefficients, was designed by the Next partners. This PID controller defines the acceleration of the follower pod and its formulation is given in Equation (3):

$$a = (k_p * e) + (k_i * i) + (k_d * d) (3)$$

where k_p , k_i , k_d are the coefficients, whereas e , i and d are the error, integral and derivative term respectively, whose expressions are:

$$e = DV - DS (4)$$

$$i = \sum e * dt \quad (5)$$

$$d = \frac{De}{dt} \quad (6)$$

In this initial phase of the study, the PID controller was implemented within our simulation and analysis framework, and its effectiveness and robustness were tested. Particularly, the simulations were conducted under a baseline scenario in which the leader did not introduce any disturbance during the merging manoeuvre. The design of alternative ADCs, as well as the simulation of different traffic scenarios, is deferred to a later stage of the study.

Global Sensitivity Analysis

Global Sensitivity Analysis (GSA) is a technique to evaluate how input variables impact model outputs. It investigates the influence of input uncertainty across the full range of their possible values, offering a deeper insight into the model's behaviour (Saltelli et al. 2010). By systematically varying inputs based on their probability distributions and analysing the corresponding output responses, GSA allows to understand how input variability propagates through the model. The outcomes of such analyses can support the identification of the most influential factors. The GSA approach adopted in this framework is a variance-based method, which relies on variance decomposition to quantify the sensitivity of model outputs to changes in input variables. This method examines how the overall output variance can be attributed to the uncertainty in individual inputs and decomposes it accordingly (Saltelli et al. 2010). Input variables are sampled using a Monte Carlo strategy based on Sobol's quasi-random sequences (Sobol, 1967).

Given a model in the form $Y = f(X_1, \dots, X_N)$, to quantify the contribution of each input factor X_j to the output variance, two sensitivity indices are introduced: the first-order sensitivity index (S_j) and the total sensitivity index (ST_j). The first-order index captures the main effect of input X_j , representing the portion of output variance attributable solely to variations in that input. In contrast, the total sensitivity index accounts for the overall contribution of input X_j , including both its individual effect and its interactions with other inputs. Thus, S_j reflects the direct influence of input X_j , while ST_j encompasses both direct and interaction effects on output variance. The two indexes S_j and ST_j are presented in Equations (7) and (8) respectively:

$$S_j = \frac{V_{X_j}(E_{X_{\sim j}}(Y|X_j))}{V(Y)} \quad (7)$$

Where $V_{X_j}(E_{X_{\sim j}}(Y|X_j))$ is the variance based first order effect for a generic factor X_j , $X_{\sim j}$ denotes the matrix of all factors but X_j , and $V(Y)$ is the variance of the dependent variable.

$$ST_j = \frac{E_{X_{\sim j}}(V_{X_j}(Y|X_{\sim j}))}{V(Y)} \quad (8)$$

where $E_{X_{\sim j}}(V_{X_j}(Y|X_{\sim j}))$ is the total effect contribution of the factor j to the output variance.

In the present analysis, the output variable Y considered is the relative speed between the follower and the leader when the distance BC is reached. This relative speed is indicated as "Relative Speed at Docking". Vice versa, the three input variables X considered are: (i) the leader speed (v_l), (ii) the

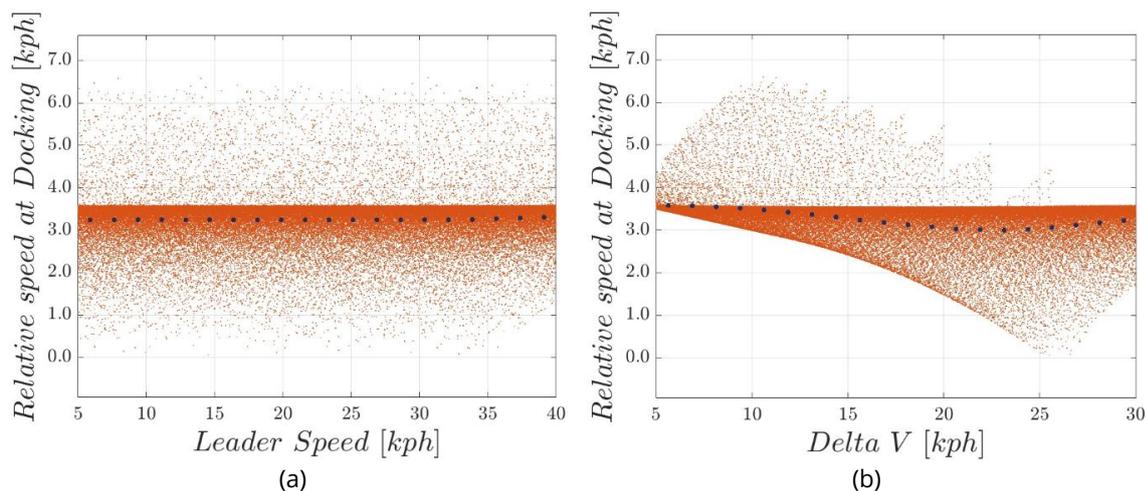
initial relative speed between leader and follower (DV); (iii) the extra distance to add to the MSD, indicated as “Delta Distance”. This latter variable, together with the MSD, defines the triggering distance AC, e.g. the distance at which the PID controller starts operating, as shown in equation 9:

$$AC = MSD + DeltaDistance \quad (9)$$

For each of the abovementioned X variables, the following range of values has been set: $v_l \hat{I}[5,40]$ km/h, $DV \hat{I}[5,30]$ km/h, $DeltaDistance \hat{I}[5,30]$ m. Then, the value of each variable to be fed into our simulation framework was selected using the Monte Carlo strategy based on Sobol’s quasi-random sequences. Specifically, a Sobol sequence of length $L = 2^{14} * 5$ has been used. Finally, knowing the input variables v_l and DV , MSD is calculated according to Equation (1).

First Simulation Results

The results of the GSA performed on the outputs of the conducted simulations are presented in Figures 22 and 23. Figure 22 presents how the output variable Y , e.g. the relative speed at docking, varies with respect to the input variables. In each of the diagram 22a, 22b and 22c, the red dots represent a simulation run, while each blue dot represents the average value of Y within a specific range of the input variable shown on the x-axis. These blue dots are included in the plots to better visualize the trend of Y with respect to the inputs. For example, Figure 22a shows that the relative speed at docking is not influenced by the speed at which the leader pod is traveling. Conversely, the relative speed (ΔV) appears to have a slight effect on Y , as highlighted by the trend of the blue dots in Figure 22. Vice versa, the influence of Delta Distance is evident, as depicted Figure 22c. Specifically, as Delta Distance increases, the relative speed at docking tends to approach the desired value $DS = 1$ m/s (3.6 km/h in the figure). This result can be explained by the fact that a larger Delta Distance implies a greater distance at which the ADC starts controlling the pod’s dynamics. On the other hand, when the Delta Distance takes on small values, a safe approach is not always guaranteed, as the recorded relative speed at docking exceeded 3.6 km/h in several simulations.



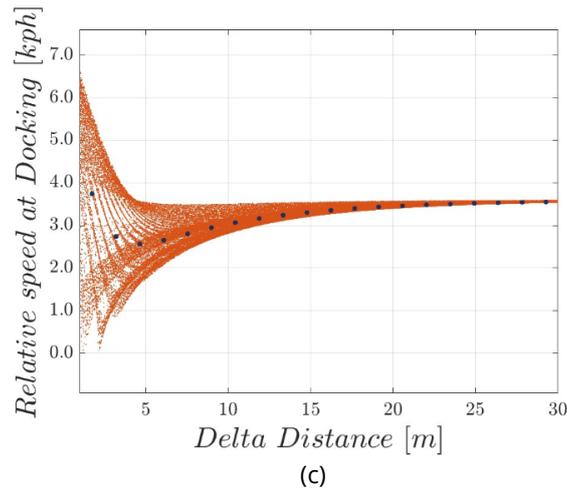


Figure 22: Trend of the variable Relative Speed at Docking with respect to the input variables Leader Speed (a), Delta V (b), and Delta Distance (c).

The fact that the Delta Distance variable is the most significant in determining the safety of the docking manoeuvre is confirmed by the sensitivity indices S and ST shown in the Figure 23. Indeed, these indices reach their highest values precisely for Delta Distance.

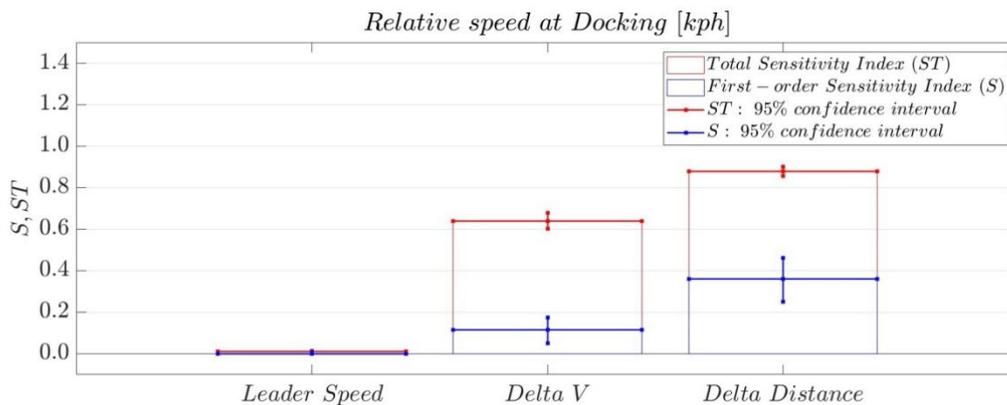


Figure 23: GSA indexes

Achievements and Future Steps

Our project has reached several significant milestones that demonstrate the viability of our approach and provide a foundation for future development.

Navigation Testing

We have successfully completed autonomous navigation testing in our controlled parking environment. The vehicle has demonstrated reliable path following with average lateral errors even before the RTK GPS fusion for localization. Our latest testing phase included introducing dynamic obstacles, with the system correctly detecting and avoiding boxes and other vehicles with appropriate safety margins. The vehicle maintains appropriate speeds based (up to 5km/h) on visibility conditions and proximity to obstacles, demonstrating context-aware decision making.

Braking and Gearing Control

We've recently completed integration of our advanced braking control system, which provides precise deceleration profiles for passenger comfort while maintaining stopping accuracy. The system is under testing, but it is proved that it allows seamlessly transitions between forward and reverse operations during complex parking manoeuvres, with proper safety interlocks preventing

inappropriate gear engagement. This capability is essential for autonomous parking and retrieval operations where multiple directional changes are often required.

While we have been able to develop a self-driving vehicle, several key areas remain for development.

Behavioural Planning for Summoning, Parking and Docking

The next development phase will focus on higher-level behavioural planning, particularly for end-to-end summoning and parking operations. This includes:

- User interface development for remote summoning requests
- Intelligent parking spot selection algorithms
- Advanced docking behaviors for charging stations
- Cooperative navigation with other autonomous vehicles

These capabilities will transform the system from a navigation demonstration to a practical mobility solution with real-world utility.

System Security for ISO Compliance

As we move toward production readiness, we're implementing comprehensive security measures aligned with automotive standards, like encrypted communications for all external interfaces and regular security scanning and penetration testing. The main goal is to make the vehicle processes compliant with ISO 21434 (Road Vehicles - Cybersecurity Engineering). These measures will ensure that the system is safeguarded against both accidental interference and intentional malicious attacks.

Hardware Upgrade

For next prototypes, we're planning to transition to automotive-grade computing platforms with enhanced reliability while increasing sensor redundancy in critical areas. Moreover, we will add weather-resistant enclosures for all external components. These improvements will enhance system robustness while preparing for deployment in less controlled environments.

By focusing on these three areas, we aim to progress from our current proof-of-concept system to a production-ready autonomous solution suitable for commercial applications in controlled environments.

5.3. Subtask 2.4.3 AI-based real-time planning of autonomous fleets

Modular bus scheduling problem with short-turning options

In subtask 2.4.3, NTUA developed a model for scheduling autonomous modular buses, incorporating short-turning options to reduce vehicle operational costs. In the first 18 months of the project, NTUA developed and tested the model using data from bus line 54 in Milan to explore the improvement potential in terms of operational costs when deploying modular autonomous vehicles and performing short-turning policies to avoid serving bus stops with low passenger demand.

The use of modular buses for short-turning enables more efficient resource utilization compared to conventional fixed-line bus services. As in past studies (Li et al. 2024), we take into consideration the fluctuating passenger demand patterns across various times of the day. In our approach, we consider bi-directional bus lines operated by autonomous modular buses. The developed scheduling model for autonomous modular buses with short-turning options can be applied to bi-directional bus lines.

We will now describe the mathematical formulation of the developed model. Each modular bus can use a number of modular units defined by the set of $u \in U$. These modular units are capable of coupling/decoupling at the depot. Each bus line has a number of stops K and is operated by a number of bus trips defined by the set T . Each autonomous bus can adopt a short-turning policy, where the eligible short-turning policies are defined by the set S .

We also introduce the parameters $c_{s,u,t}$ which indicate the incurred operational costs when performing bus trip t with u modular units while adopting short-turning policy s . Another parameter is the demand data, between any two stops k and y of a bus trip t , indicated by parameter $w_{t,k,y}$. Finally, each modular unit u has a passenger capacity γ .

With respect to variables, our binary decision variables $x_{s,u,t} = 1$ indicate the selected short-turning policy s and the selected number of modular units u for performing bus trip t . If $x_{s,u,t}$ is 1, bus trip t will be comprised of u modular units and will implement the short-turning policy. If not, $x_{s,u,t} = 0$. The passenger load of each vehicle trip t when departing from stop $k \in K$ is indicated by the continuous variable $l_{t,k}$. Also, the unserved demand by trip t at stop k is denoted by $a_{t,k}$ and is a non-negative continuous variable.

The objective of our scheduling problem for autonomous buses with short-turning options is to minimize the incurred travel costs. To do so, for each bus trip $t \in T$ we aim to select the minimum possible number of modular units and the respective short-turning policy that will result in traveling the shortest possible distance. Formally, this can be expressed as:

$$\min \sum_{t \in T} \sum_{s \in S} \sum_{u \in U} x_{s,u,t} c_{s,u,t}$$

For the first trip, and any other odd trip, we also enforce the following constraints which ensure the conservation of flow and calculate the in-vehicle load after departing from each bus stop:

$$\sum_{u \in U} \sum_{s \in S} x_{s,u,t} = 1, \forall t \in T \quad (1)$$

$$\widetilde{b}_{t,k} \doteq \sum_{t \in T} w_{t,k,y} \sum_{u \in U} \sum_{s \in S} x_{s,u,t} \beta_{s,k,y} \forall t \in T, k \in K \quad (2)$$

$$l_{1,1} = \min \left\{ \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} u; \widetilde{b}_{1,1} \right\} \quad (3)$$

$$l_{1,1} \leq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} u \quad (4)$$

$$l_{1,1} \leq \widetilde{b}_{1,1} \quad (5)$$

$$l_{1,1} \geq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} U - d_{1,1} M \quad (6)$$

$$l_{1,1} \geq \widetilde{b}_{1,1} + (d_{1,1} - 1) M \quad (7)$$

$$a_{1,1} \doteq \sum_{y \in K, y \geq 1} w_{1,1,y} - l_{1,1} \quad (8)$$

$$z_{1,1} \doteq \frac{l_{1,1}}{\sum_{y \in K, y \geq 1} w_{1,1,y}} \quad (9)$$

$$g_{1,1,y} = (1 - z_{1,1})w_{1,1,y} \forall y \in K, y \geq 1 \quad (10)$$

$$l_{1,k} = \min\{l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}; \gamma \sum_{s \in S} \sum_{u \in U} x_{s,u,1} u\} \quad (11)$$

$$l_{1,k} \leq l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}, \forall k \in K \setminus \{1\}, \forall y \in K, y < k \quad (12)$$

$$l_{1,k} \leq \gamma \sum_{s \in S} \sum_{u \in U} x_{s,u,1} U, \forall k \in K \setminus \{1\} \quad (13)$$

$$l_{1,k} \geq l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k} (d_{1,k} - 1)M \approx \forall k \in K \setminus \{1\} \quad (14)$$

$$l_{1,k} \geq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} U - (d_{t,k})M, \forall k \in K \setminus \{1\} \quad (15)$$

$$z_{1,k} = \frac{l_{t,k} - l_{t,k-1} + \sum_{y \in K, y < k} z_{t,y} w_{t,y,k} \sum_{u \in U} \sum_{s \in S} x_{s,u,t} \beta_{s,y,k}}{\sum_{y \in K, y \geq k} w_{t,k,y}} \forall k \in K \setminus \{1\} \quad (16)$$

$$a_{(1,k)} = \sum_{y \in K, y \geq k} w_{1,k,y} - (1 - z_{1,k}) \forall k \in K \setminus \{1\} \quad (17)$$

$$g_{1,k,y} = (1 - z_{1,k})w_{1,k,y}, \forall k \in K, \forall y \in K: y \geq 1 \quad (18)$$

The aforementioned objective function aims to minimize the overall vehicle running costs. With respect to the constraints of our problem, a first constraint is related to the selection of one, and only one, short-turning policy and modular unit composition per bus trip t . This is expressed in constraint (2). Constraint (3) indicates how many passengers are expected to board the modular bus of trip t at stop k . This depends on the selected number of modular units and short-turning policy of trip t , $x_{s,u,t}$, and the respective passenger demand $w_{t,k,y}$. The passenger load of the first trip at the first stop is defined by constraints (4), (5), (6), (7), where $d_{1,1}$ is a binary variable.

In more detail, constraints (4), (5), (6), (7) enforce that the load of the first trip at the first stop is equal to the minimum of the number of passengers willing to board trip 1 at stop 1 and the capacity of trip 1 which is derived based on the number of modular units u assigned to this trip and the capacity γ of each modular unit.

Constraint (8) computes the unserved demand of the first trip at the first stop, and it is equal to the total number of passengers willing to board trip 1 at stop 1, minus the passenger load of trip t when it departs from stop 1. Constraint (9) computes the percentage of passengers willing to board at the first trip at the first stop, and are able to do so. Constraint (10) computes the number of passengers who were not able to board trip 1 at stop 1 and alight at stop $y \in K$. The passenger load of the first trip at all other stops $k > 1$ is described by constraint (11), where:

$$\gamma \sum_{s \in S} \sum_{u \in U} x_{s,u,1} u$$

is the capacity of trip 1 and

$$l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}$$

is comprised of:

- the passenger load of trip 1 when departing from the previous stop, $l_{1,k-1}$
- plus the number of passengers of trip 1 willing to board at stop k , $\widetilde{b}_{1,k}$
- minus the number of passengers of trip 1 who boarded at any previous stop $y < k$ and alight at stop k , $\sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}$.

The non-convex constraint (11) can be substituted by the following set of linear inequality constraints (12), (13), (14) and (15), where $d_{1,k}$ is a binary variable.

Constraint (17) computes the unserved demand of the first trip at each stop $k \in K \setminus \{1\}$, and it is equal to the total number of passengers willing to board trip 1 at stop k , $\sum_{y \in K, y \geq 1} w_{1,k,y}$, multiplied by the percentage of passengers who were not able to board trip 1 at stop k , $(1 - z_{1,k})$. Constraint (18) computes the number of passengers who were not able to board trip 1 at stop $k \in K \setminus \{1\}$ and alight at the stop $y \in K$.

For the even trips, where $t = \{2,4,6, \dots\}$, we do not allow our vehicles to perform short-turning in order to avoid passengers being unserved by two consecutive vehicles.

The developed mathematical model for the scheduling of autonomous modular buses with short-turning options presented in the aforementioned constraints is a mixed-integer program. In the following sections, we present the implementation of the model to a real-world case study.

Case study

This case study analyzes Line 54, which operates in Milan, Italy. Line 54 serves bus stops numbered (1,2,3,4, 5, 6, 7, 8, 9, 10, 11,...,31), as shown in Figure 24. During the model implementation, trips are run in both directions, resulting in additional stops (32, 33, 34, 35, 36, ..., 62) to represent the movement in the reverse direction, as shown in Figure 25. Stop 32 corresponds to the opposite direction stop of 31, stop 33 corresponds to the opposite-direction stop of 30, and so on. Furthermore, we consider a set of 20 bus trips, denoted as $T = \{1, 2, 3, \dots, 20\}$. The topology of bus line 54 in both directions, connecting the city center of Milan with its suburban areas, is presented in Figures 24 and 25.

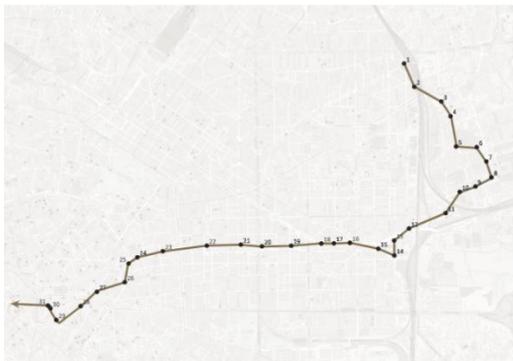


Figure 24: Case study network



Figure 25: Case Study- Opposite direction

Firstly, we consider the following short-turning policy scenarios to determine their applicability for each trip:

- Scenario 1: All stops are served (no short-turning). This scenario represents the short-turning policy $s=1$.
- Scenario 2: A short-turning is performed at stop 30, thus stops 31 and 32 are not served. This scenario presents the short-turning policy $s=2$.
- Scenario 3: A short-turning is performed at stop 29, thus stops 30, 31, 32 and 33 are not served. This scenario presents the short-turning policy $s=3$.
- Scenario 4: A short-turning is performed at stop 28, thus stops 29, 30, 31, 32, 33 and 34 are not served. This scenario presents the short-turning policy $s=4$.

For our study, we assume that each modular unit has a capacity of $\gamma = 15$ passengers, with 9 seated and 6 standing. To validate our model, we use passenger demand data from bus Line 54 in Milan provided by the AMAT Operator, an Italian Mobility planning and monitoring company specializing in urban transportation. The passenger demand of Line 54 for the first trip is illustrated in Figure 80 in the Annex of this deliverable report.

Solving the mixed-integer nonlinear problem for this case study resulted in a model with 3369 continuous and 1876 integer (1156 binary) variables. The mixed-integer nonlinear problem was solved using the commercial solver Gurobi. The computation time for solving this problem on a computer with an Intel Core i3-1215U CPU was 1.18 seconds. The optimal solution to this problem for the 10 trips of the bus line is as follows:

$$x_{4,1,1} = 1, x_{1,1,2} = 1, x_{1,1,3} = 1, x_{1,1,4} = 1, x_{2,1,5} = 1, x_{1,1,6} = 1, x_{1,1,7} = 1, x_{1,1,8} = 1, x_{2,1,9} = 1, x_{1,1,10} = 1,$$

and the trips 11 to 20:

$$x_{1,1,11} = 1, x_{1,1,12} = 1, x_{4,1,13} = 1, x_{1,1,14}, x_{2,1,15}, x_{1,1,16}, x_{2,1,17} = 1, x_{1,1,18} = 1, x_{1,1,19} = 1, x_{1,1,20} = 1$$

resulting in an objective function value of 20. Our optimal solution results in 16 unserved passengers. Analyzing the values of our decision variables, trip 1 uses 1 modular unit and short-turning option $s=4$, meaning that it does perform a short-turn at stop 28. Trip 2 uses 1 modular unit and short-turning option 1, meaning that it serves all stops. Trip 3 uses 1 modular unit and short-turning option 1. Trips 4, 6, 7, and 8 use 1 modular unit and do not use the short-turning option ($s=1$). Trip 9 uses 1 modular unit and short-turning option $s=2$, meaning that it does perform a short-turn at stop 30. Finally, trip 13 uses 1 modular unit and short-turning option $s=4$.

The passenger loads of these trips are adjusted according to the number of modular units and the short-turning policy we have selected for each bus trip. To explore the improvement potential of our model, we run the experiments for the passenger loads at the As-Is scenario associated with no short-turning policy and the use of all available modular units for every bus trip (2 modular units per trip). Comparing our approach with the As-Is scenario, we observe that in the first trip of our approach, we skipped 6 stops, resulting in a zero load for the modular bus. In trip 13 from stops 29 to 32, there was no demand, leading to no passenger load being recorded at these stops. To visualize the difference between our approach and the As-Is scenario at some stops, Figure 26 presents variations in passenger loads when implementing each approach. This figure represents scenario 4 of the short-turning policy, followed by Trips 1 and 13, which exhibit the greatest variations in passenger load.

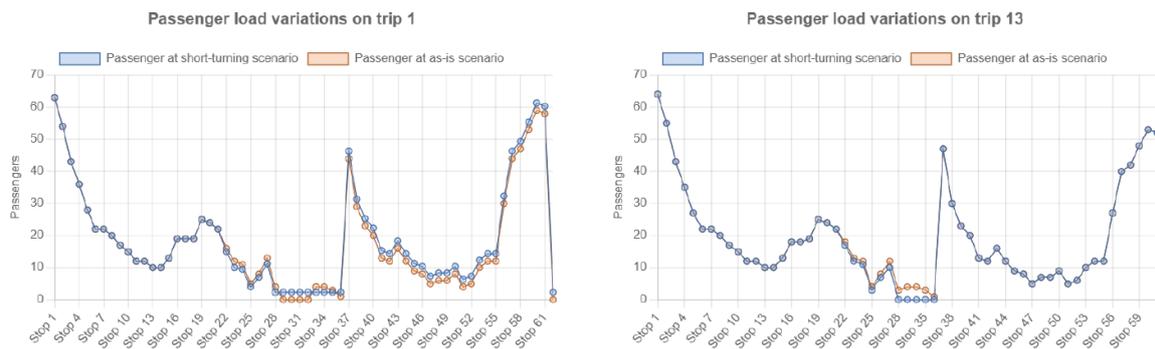


Figure 26: Passenger load variations when using modular autonomous vehicles with short-turning options and when implementing the as-is scenario with conventional vehicles

To conclude, in our case study of bus line 54 in Milan, we demonstrated an improvement potential of 50% in terms of the number of deployed vehicle capacity, 5% in terms of km-traveled, and 52% in terms of overall operational costs compared to the as-is scenario. Our implementation of the model on this data produced promising results, while the number of passengers who were not able to board their first trip was relatively low (12 passengers in total).

6. Task 2.5 Connect: V2V + V2X platform, protocols and traffic management at large scale

Task 2.5 consists of two subtasks devoted to the development and integration of Vehicle-to-Everything (V2X) technologies for the support of intelligent roads dedicated to public and/or eco-responsible transportation means (carpooling). In the first subtask ST2.5.1, with the title “Traffic management and condition assessment”, Floware focused on designing, prototyping, and validating of a sensor-based data acquisition infrastructure to support the Producer Layer of the Smart Mobility Hub framework to improve functionality in low-light conditions, monitor energy supply, and assess alternative configurations. A motorized camera prototype was developed, alongside enhancements to waterproofing and an upgrade to a new generation of edge AI hardware. In addition, MobilLysis complemented this effort by developing a framework for collecting video data from traffic streams through swarms of drones and processing these data to detect traffic congestion and provide evidence-based solutions, which is related to Asset 3 of the metaCCAZE project. The main focus was on understanding and addressing the specific requirements with an emphasis on enhancing computer vision algorithms for improved tracking and data analysis.

The second subtask ST.2.5.2, with the title “V2X-based support of transportation services” and led by Vedecom, is inter-connected with the Assets 9 and 10 of metaCCAZE. During the first 18 months of the metaCCAZE project, within this subtask, Vedecom and MobilLysis primarily focused on developing a framework to explore V2X-enabled support for transportation services. This included defining cooperative intelligent transport systems (C-ITS) message profiles tailored for various scenarios and preparing a sensor-equipped, intelligent road supervision to shared services. This will provide seamless integration of Traffic management & condition assessment solutions developed in ST2.5.1.

Future steps in ST2.5.1 include Floware’s effort to consolidate the prototype into a deployable unit and ensure interoperability with external systems in the metaCCAZE ecosystem. A key milestone is planned for July 2025, when Floware will deliver a test sensor unit to Vedecom, enabling technical

validation and collaborative integration with Vedecom’s V2X infrastructure. In parallel, development will begin on a shared API to enable data exchange between Floware sensors and Vedecom’s V2X data streams (e.g., cooperative awareness messages (CAM), decentralized environmental notifications (DENM)). This aims to improve the situational awareness and data fusion within the Smart Mobility Hub framework. On the hardware side, Floware will progressively simplify the current architecture, transitioning from the Dual Control model (Jetson Orin Nano + Raspberry Pi 4) to a more integrated and cost-effective solution. This will involve dedicated components replicating the current fault-tolerance logic while reducing hardware complexity, power use, and overall cost. Future research will also address power management strategies, and firmware modularity for context-based adaptation.

6.1. Subtask 2.5.1 Traffic management & condition assessment

Floware concentrated its efforts on identifying optimal hardware components and developing a reliable embedded software stack for multimodal mobility sensing. This included extensive experimentation with different combinations of microcomputers, microcontrollers, cameras, LiDAR sensors, Bluetooth modules, 4G modems, and infrared illuminators. The goal was to evaluate performance, power consumption, and resilience in preparation for real-world deployment scenarios.

In parallel, significant developments were made on the embedded software side, including the implementation of a battery monitoring system, autonomous recovery logic, and a secure connection to the Azure IoT Hub platform, which was deployed from scratch. Power supply options, such as mains and solar energy, were also analyzed and tested, with particular attention paid to potential instability issues and their resolution. During the first 18 months of the metaCCAZE project, the work in Subtask 2.5.1 was carried out across the following domains: (1) hardware prototyping, (2) embedded software development, (3) power supply analysis, and (4) system integration. These areas will be further analysed in the following sections of this deliverable.

Hardware exploration and prototyping

During the reporting period, Floware undertook a comprehensive exploration of hardware configurations to support the development of a modular and resilient urban sensor platform for multimodal mobility monitoring. The objective was to identify the most effective combination of sensing, computing, and communication components capable of capturing rich, real-time data while operating autonomously in outdoor environments.

A series of acquisition campaigns were carried out to source and test components including: High-resolution cameras (RGB and IR-compatible), NVIDIA Jetson Orin Nano modules for on-edge computer vision inference, Raspberry Pi 4 modules for auxiliary functions and energy management, solid-state LiDAR sensors for volumetric detection and validation, bluetooth (BT) and bluetooth low energy (BLE) scanners for passive device detection, 4G modems with fallback management logic, IR illuminators for low-light enhancement, and various housings, mounts, power converters, and thermal components.

Each configuration was subjected to iterative testing under diverse simulated environmental conditions to assess performance criteria such as image quality, Bluetooth detection range, LiDAR stability, bandwidth usage, and system temperature under load. Special attention was given to low-light performance and synchronization between sensing and transmission subsystems.

The modular architecture of the sensor platform allows the hardware to adapt to various mission profiles, from anonymous flow detection in low-traffic zones to dense multimodal junction monitoring. In this context, different combinations of camera optics, BLE antennas, and

microcontroller firmware were benchmarked to ensure optimal trade-offs between accuracy, cost, power usage, and reliability.

Floware also established coordination with several component suppliers to address delivery lead times, firmware incompatibilities, and mechanical integration constraints. These collaborations were essential to mitigate bottlenecks in prototyping and to guarantee long-term maintainability and sourcing of critical components.

This hardware exploration phase enabled the convergence towards a validated design architecture that will serve as the baseline for sensor deployments during the next phase of the project.

Embedded software development and IoT reliability

In parallel with hardware prototyping, Floware developed a robust embedded software stack to ensure the reliability, autonomy, and maintainability of the sensor units deployed in the field. This effort covered both low-level microcontroller code and higher-level software running on embedded microcomputers, with a particular focus on system resilience and secure communication.

A central achievement during the period of the first 18 months of the metaCCAIZE project was the implementation of a multilevel supervision and recovery mechanism. The sensor units are capable of detecting system failures, such as network disconnection, inference process crashes, or power dips, and autonomously triggering corrective actions, including controlled reboots, fallback procedures, or watchdog resets. These capabilities are essential to minimize on-site maintenance and ensure uninterrupted data collection over long deployments.

To support continuous power and energy health monitoring, Floware developed and integrated a custom battery monitoring system based on I2C communication with external battery management chips. This system collects voltage, current, and charge/discharge statistics in real time, enabling proactive detection of abnormal patterns and optimization of power usage strategies.

From a connectivity standpoint, all sensor units now support secure communication with Azure IoT Hub, which was deployed from scratch during the reporting period. The platform ensures authenticated, encrypted data exchange between edge devices and cloud infrastructure, facilitating large-scale data ingestion, remote logging, and future integration with the metaCCAIZE ecosystem. Custom device twins were implemented to allow remote updates of sensor configuration parameters. Furthermore, Floware developed dedicated monitoring tools to track system health metrics (CPU load, memory usage, temperature, storage availability) and implemented remote logging to support debugging and maintenance during field testing.

The software stack is designed to be modular and hardware-agnostic, enabling future adaptation to additional sensor modules (e.g., radar, environmental sensors) and easy integration into urban infrastructure. These developments represent a critical step toward achieving high system availability in the complex operational contexts foreseen for metaCCAIZE deployments.

Power supply and environmental adaptation

Ensuring power stability in varying environmental and infrastructural conditions is a key requirement for the deployment of Floware's edge sensing units in real-world urban settings. During the reporting period, Floware conducted a detailed analysis of power supply options and implemented a hybrid power strategy designed to accommodate diverse urban constraints.

Three primary power configurations were tested: (1) Grid-connected mains supply with industrial-grade AC/DC converters, (2) Solar panel systems with MPPT (maximum power point tracking) controllers, and (3) Battery-only operation for short-term autonomous deployment

Each setup was assessed under simulated real-life conditions, including fluctuating solar exposure, power outages, and irregular voltage input from municipal grids. Multiple types of lithium-ion battery packs were evaluated, with trade-offs considered in terms of autonomy, temperature stability, and recharge cycles. The team identified optimal battery chemistries and capacities to support continuous operations between 48 and 72 hours in worst-case scenarios.

Several critical challenges emerged during the experimentation phase, including: voltage drop and inverter instability during high-load image processing tasks, overheating of passive converters in sealed enclosures, charge controller malfunctions in cloudy or shaded environments, and moisture-related connector corrosion and EMI (electromagnetic interference)-related noise affecting sensor accuracy.

To address these issues, Floware applied iterative improvements such as: addition of temperature-activated ventilation modules in test enclosures, selection of weather-sealed industrial connectors, use of anti-surge components and voltage regulators, and firmware-level energy consumption management (e.g., idle modes for non-critical components).

The learnings from this testing campaign are being integrated into the reference design for upcoming pilot deployments. This includes robust support for automatic power source switching and fallback protocols, allowing sensor units to operate reliably in hybrid or degraded power conditions. These environmental adaptation strategies reinforce the resilience of the sensing infrastructure and enable its deployment in constrained urban contexts, including infrastructure-poor areas or temporary monitoring setups.

Interfacing and system integration

A central focus of Floware's work during the reporting period has been the design and integration of a modular, fault-tolerant architecture capable of supporting robust and autonomous sensing operations in diverse urban contexts. The current prototype is built around a Dual Control architecture consisting of a Jetson Orin Nano and a Raspberry Pi 4, which mutually supervise and regulate each other to maximize system stability and responsiveness.

This dual-unit setup allows each board to independently monitor the other's status, trigger safe restarts when necessary, and recover from critical failures without human intervention. This architecture significantly increased system uptime, reaching nearly 100% under stable power supply conditions, and has proven critical in early field test campaigns.

Inter-process communication between the units is implemented through a lightweight messaging protocol over UART (universal asynchronous receiver-transmitter) and GPIO (general purpose input/output) interfaces, supporting the exchange of health diagnostics, restart commands, and sensor control signals. The system was designed with modularity in mind, allowing for the flexible integration of auxiliary components, including:

- BLE scanning modules with dynamic mode switching between classic Bluetooth and BLE
- IR illuminators with synchronized triggering for night operation
- Solid-state LiDARs for detection validation during early experiments
- GPS modules with PPS output for optional time synchronization

Importantly, not all evaluated modules will be retained in the final deployed version. LiDAR was excluded due to its low cost-efficiency ratio and overlap with the existing camera-based computer vision capabilities. GPS will not be present in the initial rollout, as the sensors are intended for fixed-location deployment. BLE remains an optional feature, with the system supporting both Bluetooth and BLE modes depending on use-case-specific requirements.

To future-proof the platform and enable smoother evolution towards mass deployment, a plug-in software architecture was developed. This design allows for rapid updates and integration of new sensors or communication protocols without overhauling the full system. Looking ahead, the Dual Control architecture will be further refined and gradually simplified. Future iterations will explore dedicated and customized electronic boards capable of replicating the current control logic with greater cost efficiency and reduced complexity, while preserving the system's robustness and autonomy.

Complementing Sensor-Based Mobility Data through Drone-Based Aerial Observations

Significant early efforts from MobiLysis focused on preparing for phase 1 of the Paris pilot. This included a detailed assessment of drone regulations in France and an analysis of data collection requirements, both in terms of technical feasibility and alignment with the activities of other project partners. A central goal was to ensure that drone-based data collection would strategically fill gaps in existing sensing networks and avoid unnecessary redundancy, thereby maximizing the added value of aerial monitoring in urban mobility analysis.

Initial efforts centered on implementing the StrongSORT algorithm by (Xu et al. 2025), which links fragmented trajectories using a model trained on trajectory data and a set of rule-based filters. After adapting this approach to aerial traffic scenes, they introduced additional rules tailored to drone-captured mobility data and studied descriptors that better capture pedestrian appearance in aerial footage. These enhancements improved tracking continuity with minimal computational overhead.

Recognizing the limitations of rule-based matching in complex environments, they also explored a second approach. By clustering previously extracted trajectories into typical "standard paths", they were able to guide and refine the linking process for fragmented tracks, especially in congested or occluded areas. This allowed more accurate reconstruction of pedestrian and vehicle movement even in challenging contexts. To further improve system robustness, they developed methods to identify failure-prone zones within the drone footage. By statistically analyzing where and why tracking failed, due to occlusions, environmental complexity, or camera limitations, they identified specific spatial regions in each scene requiring improved sensing or algorithmic adaptation. These advancements are being validated especially after the drone operations in Yvelines that took place in April 2025.



Figure 27: Identifying occlusion areas from trajectories of pedestrians

6.2. Subtask 2.5.2 V2X-based support of transportation services

The future of urban mobility depends on the ability to merge sustainable transport practices with intelligent infrastructure and real-time data systems. A critical step forward involves the integration of V2X technologies to support intelligent road infrastructure dedicated to public and shared transportation services. In this subsection a robust framework centered around Smart Mobility Hubs (SMHs) is proposed. In the following section the conceptual foundation, systemic architecture, and enabling technologies that comprise this next-generation smart mobility hub approach are outlined.

Smart Mobility Hubs (SMHs)

A Smart Mobility Hub (SMH) is defined as a transport node that integrates physical, digital, and democratic dimensions (Geurs et al. 2023). This sets it apart from traditional mobility hubs emphasizing user-friendly design that support intermodality (physical integration), real-time travel information, service booking, and open data platforms (digital integration), as well as participatory governance that includes diverse stakeholders (democratic integration). The concept for SMH has emerged in response to the increasing complexity of urban mobility systems. Traditional hubs, intersection of transport services, have historically aimed to improve modal shift and reduce car dependence (Rongen et al. 2022), but often lacked the technological and inclusive planning necessary for fully integrated systems. Geurs et al. (2023) argue that the societal value is maximized when integration spans all three domains, such as hubs enabling seamless transfer between modes, providing digital travel tools, and involving communities in hub design.

Further support for the SMH framework comes from Ku et al. (2022), who apply the concept of urban metabolism, showing how adaptive hubs can optimize infrastructure for sustainability, health, and efficiency. Their study in Seoul demonstrates how retrofitting dispersed services into centralized hubs can reduce car use and encourage healthier mobility behavior. Practical implementation guidance is offered by CoMoUK (CoMoUK 2019), which outlines critical design factors such as visibility, intuitive user experience, consistent branding, and inclusive accessibility,

aligning closely with the SmartHubs model that co-locates shared and public transport options, supported by digital tools. Pelaez Bueno (2021) also offers empirical evidence from Zurich, where GPS and data analytics have been used to identify optimal intermodal transfer points, highlighting the importance of data-driven planning in aligning SMHs with real travel patterns and user needs.

Traffic Management Framework for SMH

To operationalize the Smart Mobility Hub concept, a comprehensive, multi-layered traffic management architecture is required. The pipeline integrates real-time and historical data from multiple stakeholders and sensors, enabling proactive decision-making and optimization of mobility flows. This framework illustrated in Figure 81 (Annex) represents an end-to-end intelligent traffic data pipeline, designed to collect, integrate, and analyze multi-source mobility data to enable real-time predictions and actionable supervision for connected vehicles and devices. This framework is structured into four layers: Producer, Communication, Treatment, and Consumer; each representing a logical stage in the smart mobility data lifecycle.

The system begins with data acquisition from various sources (e.g., sensors, cameras, drones), followed by transmission to the cloud of all collected data and in the cloud layer, data is cleaned, analyzed, and used for predictive modeling. Finally, outputs are communicated back to consumers through alerts and recommendations, enabling data-driven mobility supervision and management.

Producer Layer

The Producer Layer encompasses the full spectrum of data generation sources embedded within the urban mobility environment and serves as the foundational layer for both real-time and historical data collection. It integrates fixed and mobile sensing platforms to ensure comprehensive data acquisition. Drones, in this case, are deployed for aerial surveillance, capturing high-resolution imagery and video streams used in traffic flow analysis, event detection, incident reporting, and dynamic object recognition. Complementing this, ground-based sensors such as inductive loops, radar units, typically installed along roadways and intersections, record critical traffic metrics, including vehicle speed profiles, volume, occupancy rates, and classifications. Camera sensors (fixed and rotating) support vehicle/pedestrian detection, behavioral analysis, and illegal maneuver monitoring (see subtask 2.5.1). Smart parking systems contribute data on space availability, turnover, and occupancy duration for dynamic parking management. Public transport vehicles are equipped with GPS and telemetry systems, offering real-time location data, while mobile applications facilitate station-to-station carpooling through booking, tracking, and route sharing. Additionally, V2X technologies and carpooling platforms enrich the data ecosystem with information on vehicle locations, routes, availability, and occupancy.

To evaluate the effectiveness and efficiency of the proposed framework, Vedecom will integrate a carpooling service that supports V2X communication capabilities. This service will not only serve as a source of real-time vehicular data but will also act as a testbed for assessing the practical impact of developed V2X communication system and gateway on dynamic traffic environments. By analyzing the data generated through this platform, the main aim is to measure improvements in key performance indicators (KPIs), define in WP1, such as vehicle speed, congestion levels, and road safety. This real-world validation will help demonstrate how the proposed framework can enhance supervision-based traffic management, mobility efficiency, reduce traffic bottlenecks, and contribute to safer, smarter urban transportation systems.

Communication Layer

The Communication Layer enables secure, efficient transmission, synchronization, and initial formatting of data collected from the Producer Layer, acting as the bridge between different data sources and central processing units. A core component is the PRIM API, developed by Île-de-France Mobilités (IDFM), which provides access to real-time and scheduled public transport data, such as vehicle locations, timetables, service disruptions, and estimated arrival times. With collection of high-frequency GPS data from buses and trams, updated every 30 seconds, and structured data feeds (GTFS and SIRI), the PRIM API ensures data consistency and interoperability across multimodal transport systems and operators.

Complementing this, the 5GMETA API, developed under the European project 5GMETA, supports real-time V2X communication between vehicles and other connected devices. The system leverages Cooperative Intelligent Transport Systems (C-ITS) services and integrates Cellular V2X (C-V2X) and 5G network for low-latency data exchange. As illustrated in Figure 28, the platform's core component is C-ITS based data sharing software, which connects vehicles, roadside sensors, and infrastructure services using standard ETSI C-ITS messages. Finally, the platform relies on the C-V2X communication protocols for low latency data communication with vehicle using both telecommunication networks from mobile operators (e.g. 4G/5G) and V2X roadside units (RSU) deployed by the infrastructure operators.

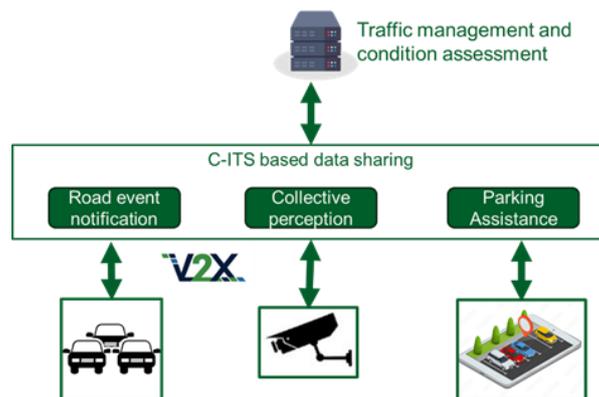


Figure 28: Overview of V2X based support of mobility service

To support the mobility services in the project, three services have been primarily identified: (1) Decentralized event notification service (ETSI EN 302 637-3(Its, 2023)), (2) Collective perception service (ETSI TS 103 324 (Its 2024a)), and (3) Parking Availability Service (ETSI TS 103 916 (Its 2024b)).

The implementation of such platform is done through a so called V2X gateway which supports the generation of messages from traffic supervision by rest APIs and message consumption by the subscription to a data broker. At the core of this V2X gateway is a V2X software stack that implements various C-ITS services and network management functions to reliably and efficiently deliver data messages over multiple radio communication links, using either the 5G network or C-V2X RSUs. Hence, connected vehicles can be informed and alerted of the different traffic and safety messages. This platform will support the data exchange with all types of connected vehicles to offer road safety in dangerous areas as well as deliver information in order to enhance traffic management such as parking notification for the shared mobility services.

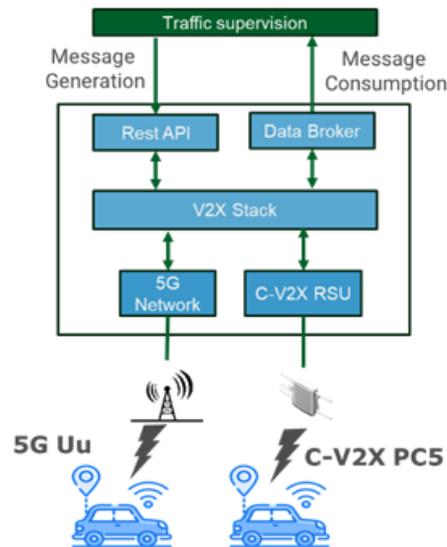


Figure 29: Architecture of the V2X Gateway components

Finally, roadside and on-board communication units are being developed for integration into the road infrastructure and connected vehicles. These units will offer dual connectivity, supporting both 5G and C-V2X communication technologies. Figure 30 illustrates a sample of an on-board unit which is intended to be installed in the connected vehicles. The Communication Layer thus serves as the infrastructural backbone for consistent and scalable data exchange.



Figure 30: 5G and V2X on-board unit for the connected vehicles

Treatment Layer

The Treatment Layer serves as the computational core of the system, transforming raw, heterogeneous data into structured, and actionable information. Hosted in the cloud, it supports advanced data processing and analytics. This process begins with data cleaning, which ensures accuracy and consistency by correcting errors, handling outliers, and filling in missing values across different sources. This is followed by exploratory data analysis (EDA), using statistical and visual to uncover patterns, regular trends, unusual behaviors, and relationships between different types of mobility data, ultimately enabling informed analysis and decision-making.

Following this, data aggregation is performed by combining various data streams based on time and location. For instance, merging traffic counts, vehicle trajectories, and public transport activity can create richer and more informative datasets. This step allows for deeper insights into mobility dynamics by placing different types of information into a shared context. With this integrated dataset, predictive modeling will then be carried out. This involves using machine learning models to estimate future traffic conditions, identify areas at risk of congestion, and detect potential safety concerns.

Consumer Layer

The Consumer Layer delivers refined insights and real-time services from the Treatment Layer to relevant stakeholders and end-user applications. As the system's main point of interaction is with external actors, it ensures the processed data is translated into actionable services. In carpooling services, the Consumer Layer supports optimized routing, real-time ride-driver matching, and demand forecasting to enhance vehicle occupancy and reduce passenger wait times. Parking systems benefit from predictive analytics for space availability, dynamic pricing, and centralized management via dashboards and mobile apps. For pedestrian applications, it enables navigation assistance, safety alerts at intersections, and real-time crowd density mapping, enhancing walkability and urban safety.

Validation Methodology: V2X Carpooling Service

To validate the proposed four-layer framework, a V2X-enabled carpooling service serves as an ideal test case, traversing all layers of the pipeline while providing measurable outcomes. The validation methodology consists of the following components:

Producer Layer Validation: Collecting real-time GPS data from participating vehicles, capturing user requests via the mobile application (including origin, destination, and time constraints), and monitoring hub occupancy and traffic conditions.

Communication Layer Validation: Measuring data transmission latency between vehicles and the central system, evaluating the reliability of V2X communications under varying traffic conditions, and testing API request completion rates during peak usage periods

Treatment Layer Validation: Assessing both the performance and accuracy of the system's core analytical components. This includes evaluating the computational efficiency of the route optimization algorithms by measuring processing times and resource utilization under varying demand levels.

Consumer Layer Validation: User engagement metrics – such as app opens, booking completions, and cancellations – are tracked alongside response times to service notifications. This validation methodology enables a quantitative assessment of the framework's performance across all layers while providing actionable insights for iterative improvements to the Smart Mobility Hub infrastructure.

7. Task 2.6 Manage & Control: Electric Vehicle (re-)Scheduling, Low emission-based traffic management, Smart parking

Task 2.6 consists of three subtasks related to the management and operation of zero-emission shared mobility services. Subtask 2.6.1, entitled "Electric Vehicle Scheduling", is linked to Assets 1, 8, and 11 of the metaCCAIZE project. Subtask 2.6.1 focuses on "Low emission-based traffic management", associated with Assets 3, 9, and 10. Finally, subtask 2.6.3 develops a "Supply-demand matching platform for on-demand shared zero emission services", related to Assets 12, 13, and 14.

7.1. Subtask 2.6.1 Electric Vehicle Scheduling

This section outlines the methodological framework and models developed in subtask 2.6.1 to support scheduling strategies for electric vehicle networks, specifically focusing on electric bus transit systems. NTUA proposes a series of modelling tools that address: (1) the optimal selection of charging station locations for electric buses, considering both slow and fast charging types; (2) an extended version of this tool that incorporates inter-station travel time and energy

consumption uncertainties; and (3) a line planning and vehicle scheduling framework tailored for electric bus fleets.

In addition, TUM has extended the e-fleet operation framework, FleetPy, to better account for user-operator interactions for Pick-Up and Drop-Off (PUDO) processes in autonomous mobility-on-demand (AMoD) services. This work highlights the importance of considering PUDO characteristics, such as duration, exact location, and boarding/alighting infrastructure) when planning and deploying AMOD services. Additionally, this aligns with the remaining initiative of subtask 2.6.1: developing algorithms to determine the optimal network of dynamically managed curbside areas, which will also be analysed in the current deliverable.

Finally, TUD focused on developing multimodal waste collection logistics in the city center of Amsterdam by creating a multimodal reverse urban logistics optimization model. This model aims to support the municipality in its effort to reduce litter and minimize the presence of heavy-duty vehicles in the city center.

7.1.1. Electric Bus Charging Station Location Selection Problem with Slow and Fast Charging

One of the metaInnovations that is fully developed in the first 18 months of the project is the charging station location selection model for fixed-line public transport services that utilize electric buses. The model considers the deadheading time of electric buses between the final stop of their trip and the locations of the potential charging stations with the objective of minimizing vehicle running costs. The problem is solved at a strategic level; therefore, several parameters of day-to-day operations, such as deadheading distances, are included as aggregate data considering their average values. In addition, it considers different charger types (slow and fast), which are subject to a day-ahead scheduling of the charging sessions of the buses. The developed model is a mixed-integer nonlinear program, which is reformulated as a mixed-integer linear program. The model has been tested based on data from the Athens Living Lab (see the related published work (Gkiotsalitis et al. 2025)), demonstrating its potential as a decision support tool for selecting charging station locations and charger types in large public transport networks.

A brief literature review will be given in the context of the Deliverable report D2.1 to introduce readers to the broader topic of decision support models for the Charging Station Selection Problem for Electric Buses (EB-CSLP). The EB-CSLP involves strategically locating static charging infrastructure for a bus network. The main objective is to efficiently provide electric power to buses under considerations related to the installation cost and the energy requirements of electric vehicles. Several studies have been conducted in this topic. Considering the charging station location and charging scheduling problem several research have been conducted. A branch of the related literature has formulated comprehensive models that optimize both location and scheduling decisions for electric bus charging. In this line of research, (Rogge et al. 2018) jointly examined vehicle scheduling and charger location under depot charging, presenting a grouping GA and a MILP model for each task, minimizing the TCO for an electric fleet. Liu et al. (2020) introduced a model based on the deficit function theory, along with a mathematical programming model, to establish schedules for an electric bus fleet while optimizing the number of fast-chargers deployed at terminal stations. Stumpe et al. (2021) presented a MILP to concurrently optimize both the opportunity charging infrastructure and schedules for an electric fleet and presented a Variable Neighbourhood Search-based (VNS) algorithm to solve the model. Using a VNS algorithm, (Olsen and Kliewer 2022) explored the integration of depot charging planning and electric bus scheduling with the aim of minimizing the overall cost, accounting for depot charger installation, vehicle expenses, and operational costs. In contrast to the above, our developed method decides simultaneously the locations of charging stations and the charger types.

Mathematical Formulation

The problem at hand entails determining the optimal charging installation options, where a charging installation option is associated with a physical location and a charger type (slow/fast). The problem also entails the optimal scheduling of vehicle charging requests at the selected charging installation options.

In this problem definition, we have a pre-selected pool of potential charging station locations V . At each charging station location, we may have several chargers of different types (slow/fast). This results in the set of all possible charging installation options N . Note that $V \subseteq N$ because multiple charging installation options might be offered at the same physical location. The set N of possible charger installations results in an expanded set of potentially available charging slots F . Given the above, we have to allocate $K = \{1, 2, 3, \dots\}$ vehicle trips that require charging to the available charging slots F such that the deadheading times of all trips are minimized subject to budgetary constraints. The deadheading times are average values and the distance between the final stop and the potential charging station.

The set of potential charger installations N can be decomposed into the set of slow charging installation options N_1 and the set of fast charging installation options N_2 , respectively. These charger installation options can be located at any point of the set V , which represents the set of all possible physical locations of charging stations and is a sub-set of N . In addition, each charger $j \in N_1$ can be used multiple times during the day resulting in a set of time slots F_1 . The same holds for any charger $j \in N_2$, resulting in time slots F_2 .

Using the above notation, variable q_{kj} is introduced that model the assignment decision of each vehicle k to exactly one charger j . With respect to q_{kj} , the deadheading time of a trip k from its last stop to its charger is defined as y_k that takes values according to Equation (1):

$$y_k = \sum_{j \in N} t_{kj} q_{kj}, \forall k \in K \quad (1)$$

where t_{kj} is a parameter indicating the estimated deadhead time from the end stop of trip k to charger j . Because we need to minimize the overall deadheading time of all trips when traveling to the locations of their chargers, the objective function of our problem is:

$$\text{MIN } \sum_{k \in K} y_k \quad (2)$$

which is a linear function that minimizes the deadheading cost for all buses $k \in K$.

Considering the above, the charging station location selection and scheduling problem with slow and fast charging options, which strives to minimize the overall deadheading times, is further defined by a set of constraints that guarantee:

1. Each bus k gets assigned to at least one slow or one fast charger, and their respective charging time slots.
2. The energy levels of the buses remain above a specific level SoC_{min} at all times.
3. That the charging scheduling of buses to the charging slots of slow and fast chargers takes place according to the bus schedule and parameter t_{kj} (as it has been defined above).
4. The overall charging station installation cost is below a maximum budget.

5. The charging station installation selection takes place according to a maximum radius around each last stop of the buses.
6. Depending on the type of charger, the buses charge for different durations.

All these constraints and their mathematical formulation can be viewed at the published peer-reviewed article for this specific model by Gkiotsalitis et al. (2025).

Case study on Athens Living Lab

Next, the case study on central Athens is presented, focusing on the planning of the electric bus network of this study area. In the case study, nine candidate physical locations for charging stations (set V) are considered, each having two charger options: one slow and one fast. This results in 18 candidate charger installation options (set N), split into slow chargers (N_1) and fast chargers (N_2). The candidate locations are primarily bus depots, but the model can consider other locations as well. The study examines 10 bus lines from Athens' network, selected because their conventional buses will be replaced by electric ones in 2024. These lines, all within Athens Municipality, require one recharge per day, resulting in 10 trips needing charging. Before solving the model, travel distances between each bus line's terminal stop and each candidate charging station are calculated. The network described above is presented in Figure 31.

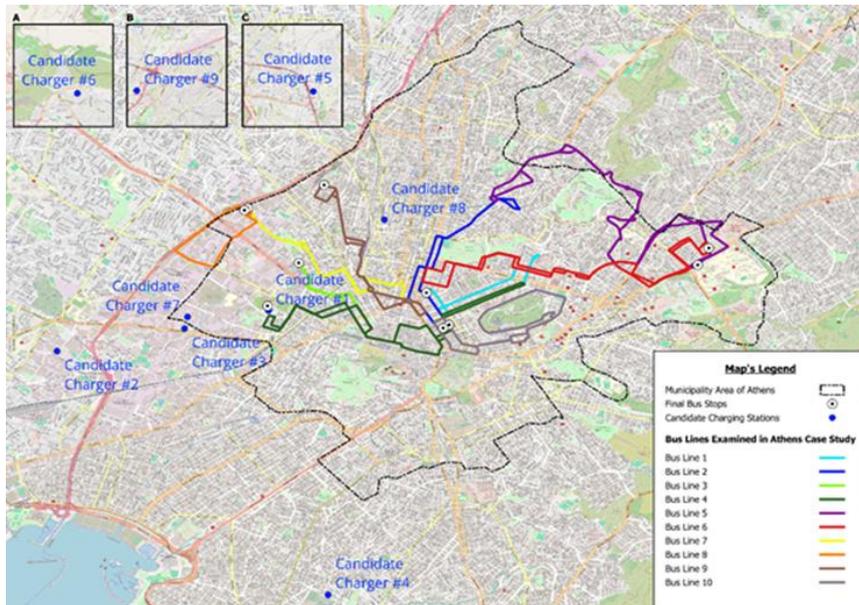


Figure 31: Athens network of the electric bus lines considered for the study, together with the physical locations candidates for the charging stations. Locations A, B and C on the upper left corner represent depots in the Ano Liosia, Nea Philadephia, and Anthousa regions.

The bus arrival times τ_k indicate when bus trip k finishes its itinerary and is ready to head to a charger location v to charge at a candidate charger $j \in N$. The parameters p^{s_k} and p^{h_k} refer to two time limits that define the period during which a bus can start charging at a charger $j \in N$ at either charging slots $c^{s_{j1}}$ or $c^{h_{j2}}$ depending on its type (fast or slow). As in the Athens toy network, all times are represented in continuous time representation in minutes past midnight (e.g. 1114.0 represents 18:34).

Based on this input data, the model's optimal solution indicates that four charging options should be selected (e.g. #1, #2, #15, #16) and chargers should be installed at locations #1 and #8. The total deadhead time for this optimal solution is 50.23 minutes for the ten bus trips of the respective lines.

Regarding the values of the three-dimensional binary variables $u^s_{kjj_1}$ and $u^h_{kjj_2}$ that handle the assignment of bus trips to charging slots, the non-zero values are: $u^s_{1,15,3} = u^s_{2,15,2} = u^s_{3,1,3} = u^s_{6,15,4} = u^s_{7,1,5} = u^s_{8,1,4} = u^s_{9,15,6} = u^h_{4,2,7} = u^h_{5,16,7} = u^h_{10,16,10} = 1$. One can notice that only two charging station locations are selected out of the 9 available locations: locations #1 and #8. Bus trips 3, 4, 7 and 8 are assigned to charging station location #1, while the rest of the bus trips are assigned to charging station location #8. Given the optimal solution derived and the respective assignments of bus trips k to charging options $j \in N$, as well as time slots $f_1 \in F_1$ and $f_2 \in F_2$, one can notice that in the Athens Municipality area, all ten bus trips can charge at the nearest candidate charging location.

An additional experiment was conducted on the Athens network to compare the proposed model to existing approaches. This experiment considered only slow chargers ($N \equiv N_1$) as candidate options at each location, unlike the initial application, which included both slow and fast chargers. With this limitation, the model selects an optimal solution with a total deadhead time of 55.13 minutes and identifies the need for an extra charger at location #3.

The model successfully assigns bus trips to potential charging options $j \in N$ of nine candidate charging station locations. Following the same approach, one can easily extend the application of our model in several medium-sized cities and parts of wider metropolitan regions. We finally note that the 10 selected lines in Athens are short, and the electric buses operating in these lines will have to recharge only once during their daily operations, resulting in 10 trips that need recharging. Notwithstanding this, our model can also be applied to longer lines for which electric buses might require multiple recharging sessions during their daily operations.

The contribution of this tool is twofold: first, we introduce a mathematical model which is formulated as a MILP and can be solved to global optimality for instances with considerable size, thereby enabling the provision of support towards strategic decisions related to the transition of medium-sized cities to public transport networks that include electric buses. On a second level, the study provides insights from the real-world network of Athens, Greece, where the model is applied to select specific locations for charging stations, and respective charging station types, out of a wider set of charging station installation options.

This mathematical modelling tool is also part of a peer-reviewed publication, conducted in the context of WP2 of metaCCAZE with title “*Electric bus charging station location selection problem with slow and fast charging*” by Gkiotsalitis et al. (2025). Additional analysis is included in the deliverable, such as a computational complexity analysis for this proposed mathematical model.

7.1.2. Electric Bus Charging Station Location Model considering Travel Time and Energy Consumption Uncertainties

NTUA also developed an exact model for the Electric Bus Charging Station Location Problem (EB-CSLP) in urban areas, addressing uncertainties in inter-station travel times and energy consumption. As a strategic-level problem (Kchaou-Boujelben 2021), EB-CSLP precedes decisions like timetabling, frequency setting, and vehicle scheduling (Gkiotsalitis 2023; Gkiotsalitis et al. 2023a). Prior studies have explored optimal placement of charging infrastructure at depots (Hsu et al. 2021; Uslu and Kaya 2021), en-route bus stops (Wu et al. 2021), or turnaround locations (Randhahn Annetteand Knote, 2020), with the latter two providing opportunity charging. As BEB adoption grows, mitigating operational disruptions becomes critical to ensure consistent service, minimize deadheading, and reduce queue waiting time costs. In the remainder of this section, *deadheading* refers to the time electric buses travel empty from their final stop to a charging station, and *queue waiting time* refers to the period between bus arrival at the station and the start of charging (Gkiotsalitis 2021). The proposed model optimizes charging station locations, charger type selection (slow/fast), and charging schedules under operational uncertainties.

Mathematical Program

The problem aims at determining the optimal locations of charging stations from a set of V potential charging station sites. The charging station installation options are N ; slow or fast based on the frequency of charging requests assigned to each charging location. At each location, either type of charger (slow/fast) can be selected. The installation budget of these chargers is considered to be fixed.

The objective of this model is twofold. The first objective is the minimization of the overall deadheading costs. The second objective is related to the reduction of the queue waiting times for the K electric buses requiring charging, defined as the elapsed time between their arrival at the charger and the start of the charging process. In this model, the predefined set V represents the potential physical locations for charging stations, which can host up to N types of chargers. The exact number depends on the outcome of the specific optimization problem, considering charging demand and deployment costs. The set N is divided into two subsets; $N_1 (\subseteq N)$, representing slow chargers, and $N_2 (\subseteq N)$, representing fast chargers. Any location in V can accommodate either charging type, meaning $V \subseteq N$. Each charger $j \in N$ can be used multiple times daily as long as it serves only one bus at a time. A predetermined set M includes all electric buses, with a subset $K \subseteq M$ indicating the buses requiring charging.

The state of charge for electric bus k , denoted as SoC_k , refers to the battery level after completing operations and before charging. SoC_k must remain within predefined charging level boundaries (SoC_k^{min}, SoC_k^{max}) and reach a sufficient charge level (SoC_k^{suf}) upon charging completion, defined as a percentage ω_k of SoC_k^{max} ($SoC_k^{suf} = \omega_k SoC_k^{max}$). This percentage varies throughout the service day to minimize downtime and potential delays. Electric bus k reaches its final stop at τ_k , with a deadheading time t_{kj} from its last stop to the potential charging location j , following the minimum travel distance d_{kj} . Charging is required after completing a number of trips, which deplete its battery.

The battery consumption per unit distance, e , lies within the electric bus energy consumption range (e_{min}, e_{max}). The fixed cost of installing charger $j \in N$ is represented by b_j , with a total budget available for charging station installations, b_{max} . Charger power is defined by r_1 (slow chargers) and r_2 (fast chargers). The maximum number of chargers a location $v \in V$ can accommodate is given by λ_{1v} (slow charging types) and λ_{2v} (fast charging types). Chargers are available throughout the operational day, with c_i , indicating the latest time by which all charging operations must be completed.

The problem has the following seven variables, the first two of which are binary, and the rest are continuous and nonnegative:

- x $x = [x_1, \dots, x_j, \dots, x_{|N|}]^T$, where $x_j = 1$ if we decide to construct the charger $j \in N$, and $x_j = 0$ if not
- q_{kj} $q_{kj} \in 0,1$, where $q_{kj} = 1$ if the trip $k \in K$ is assigned to charger j , and 0 otherwise,
- f_{kj} continuous variable, which indicates the time when the bus k starts charging at charger j ,
- l_{kj} continuous variable, which indicates the time when the bus k finishes charging at charger j
- y $y = [y_1, \dots, y_i, \dots, y_{|K|}]^T$, continuous variable, which indicates the deadheading time of bus $k \in K$
- w $w = [w_1, \dots, w_i, \dots, w_{|K|}]^T$, continuous variable, which indicates the queue waiting time of bus $k \in K$, from its arrival at the charger j until the start time of its charging.

Considering the above, the mathematical model of the charging station location problem is presented below.

(\tilde{Q}):

$$\min \gamma \sum_{k \in K} y_k + \sum_{k \in K} w_k \quad (1)$$

subject to:

$$y_k = \sum_{j \in N} t_{kj} q_{kj} \quad \forall k \in K \quad (2)$$

$$w_k = \sum_{j \in N} [f_{kj} - (\tau_k + t_{kj})] q_{kj} \quad \forall k \in K \quad (3)$$

$$\sum_{j \in N} a_{kj} x_j \geq 1 \quad \forall k \in K \quad (4)$$

$$\sum_{j \in N} q_{kj} = 1 \quad \forall k \in K \quad (5)$$

$$q_{kj} \leq x_j \quad \forall k \in K, j \in N \quad (6)$$

$$q_{kj} \leq a_{kj} \quad \forall k \in K, j \in N \quad (7)$$

$$\sum_{k \in K} q_{kj} \geq x_j \quad \forall j \in N \quad (8)$$

$$\sum_{j \in N_1} x_j \leq \lambda_{1v} \quad \forall v \in V \quad (9)$$

$$\sum_{j \in N_2} x_j \leq \lambda_{2v} \quad \forall v \in V \quad (10)$$

$$\sum_{j \in N} x_j b_j \leq b^{\max} \quad (11)$$

$$f_{kj} \geq (\tau_k + t_{kj}) q_{kj} \quad \forall k \in K, j \in N \quad (12)$$

$$l_{kj} \leq c_l q_{kj} \quad \forall k \in K, j \in N \quad (13)$$

$$l_{kj} = f_{kj} + q_{kj} \frac{\omega_k SOC_k^{\max} - (SOC_k - ed_{kj})}{r_1} \quad \forall k \in K, j \in N_1 \quad (14)$$

$$l_{kj} = f_{kj} + q_{kj} \frac{\omega_k SOC_k^{\max} - (SOC_k - ed_{kj})}{r_2} \quad \forall k \in K, j \in N_2 \quad (15)$$

$$SOC_k - ed_{kj} \geq SOC_k^{\min} \quad \forall k \in K, j \in N \quad (16)$$

$$q_{ij} + q_{kj} \leq 1 \text{ if } f_{ij} \leq f_{kj} \wedge f_{kj} \leq l_{ij} \quad \forall k \in K, i \in K \setminus \{k\}, j \in N \quad (17)$$

The objective function (1) minimizes total deadheading and queue waiting times for charging, with weight γ prioritizing deadheading time relative to the queue waiting time, because, during this period, the bus incurs high operational costs. Constraints (2) define the deadheading time (t_{kj}) of each bus $k \in K$, while constraints (3) define the queue waiting time for each bus k at charger $j \in N$. Constraints (4) ensure each bus k has at least one accessible charger j , based on the binary parameter a_{kj} , which equals 1 if the state of charge of the bus upon arrival at the charger j remains above the minimum threshold (SOC^{\min}), and 0 otherwise.

Constraints (5) assign each bus k to exactly one charging location ($q_{kj} = 1$). Constraints (6) and (7) restrict assignments to constructed ($x_j = 1$) and accessible ($a_{kj} = 1$) chargers, respectively. Constraints (8) ensure a charger j is constructed only if at least one bus is assigned to it ($q_{kj} = 1$). Constraints (9) and (10) limit the number of charging types per location $v \in V$, while constraints (11) ensure total installation costs remain within budget b^{max} . Constraints (12) enforce charging sequence of every bus k at the assigned charger j , and constraints (13) ensure that the charging process completes before the predefined end time c_l .

Constraints (14) and (15) calculate the charging end time l_{kj} for bus k , based on charging start time (f_{kj}) summed with the required charging duration. Constraints (14) apply to slow chargers ($j \in N_1$), while constraints (15) apply to fast chargers ($j \in N_2$). Constraints (16) ensure that the remaining battery level of bus k at the charger j , if assigned to it, remains above SoC_k^{min} . Charger $j \in N$ can be used multiple times daily without predefined time slots, allowing the charging start time of each bus $k \in K$ at charger j (f_{kj}) to be any positive continuous value. To avoid overlaps, constraints (17) ensure that no charger serves more than one vehicle at a time: if another bus $i \in K \setminus \{k\}$ is assigned to charger j and starts charging before bus k ($f_{ij} \leq f_{kj}$), then bus k must wait until bus i finishes ($f_{kj} < l_{ij}$), enforcing $q_{ij} + q_{kj} \leq 1$. In some of the aforementioned constraints, the big- M approach was utilized. This involved setting M as a large positive number ($M \gg 0$), greater than the relevant terms in the constraints, to ensure their enforcement or relaxation as needed.

Charging Station Location Selection Problem Under Inter-Station Travel Time and Energy Consumption Uncertainties

During daily operations, bus trips may experience variability in inter-station travel times, deadheading times to charging stations, and energy consumption. Factors like traffic, roadworks, weather, and passenger occupancy can cause disruptions. Thus, it is essential to account for these uncertainties in decision-making. Specifically, we consider uncertainties in the following parameters for each bus $k \in K$: (i) completion time τ_k , (ii) deadheading time t_{kj} , and (iii) energy consumption e . These parameters are treated as random variables, and the goal is to minimize the expected value of the objective function by optimizing based on their probability distributions.

$$E \left[\gamma \sum_{k \in K} y_k + \sum_{k \in K} w_k \right] \quad (18)$$

which can be equivalently be written as:

$$\gamma \sum_{k \in K} \sum_{j \in N} E[t_{kj}] q_{kj} + \sum_{k \in K} \sum_{j \in N} f_{kj} - E[(\tau_k + t_{kj})] q_{kj} \quad (19)$$

where, τ_k and t_{kj} are uncertain parameters drawn from probability distributions. To solve this Stochastic Optimization Problem, we use the Sample Average Approximation (SAA) method (Gkiotsalitis et al., 2022; Kim et al., 2015), which samples travel time scenarios $\tau_{1,k}, \tau_{2,k}, \dots, \tau_{|S|,k}$ and $t_{1,kj}, t_{2,kj}, \dots, t_{|S|,kj}$ from the distributions of τ_k and t_{kj} , respectively. This approximates the objective function in (32), which can now be written as:

$$\frac{1}{|S|} \left[\gamma \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} t_{s,kj} q_{kj} + \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} f_{kj} - (\tau_k + t_{kj}) q_{kj} \right] \quad (20)$$

Incorporating uncertain parameters τ_k , t_{kj} , and e_s into the optimization problem may cause infeasibilities due to excessive completion times, deadheading times, and energy consumption. These uncertainties affect the charging start time f_{kj} , the state of charge, and the charging end time l_{kj} , potentially violating constraints. To address this, we relax constraints (12), (14), (15), and (16), allowing a small number of buses to remain uncharged or have incomplete charging in

scenarios with unexpected delays or high energy consumption. This results in an optimization model (\hat{Q}) that accounts for these uncertain elements, as follows:

(\hat{Q}):

$$\min \frac{1}{|S|} \left[\gamma \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} t_{s,kj} q_{kj} + \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} f_{kj} - (\tau_k + t_{kj}) q_{kj} \right] \quad (21)$$

$$\text{s.t.: Equations (4) - (17)} \quad (22)$$

$$C_{s,kj} = f_{kj} - (\tau_{s,k} + t_{s,kj}) q_{kj} \quad \forall k \in K, j \in N, s \in 1, \dots, S \quad (23)$$

$$\sum_{s \in S} \Pr(C_{s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (24)$$

$$C_{1,s,kj} = l_{kj} - f_{kj} - q_{kj} \frac{\omega_k SOC_k^{max} - (SOC_k - e_s d_{kj})}{r_1} \quad \forall k \in K, j \in N_1, s \in 1, \dots, S \quad (25)$$

$$C_{1,s,kj} = l_{kj} - f_{kj} - q_{kj} \frac{\omega_k SOC_k^{max} - (SOC_k - e_s d_{kj})}{r_2} \quad \forall k \in K, j \in N_2, s \in 1, \dots, S \quad (26)$$

$$\sum_{s \in S} \Pr(C_{1,s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (27)$$

$$C_{2,s,kj} = SOC_k - e_s q_{kj} d_{kj} - SOC_k^{min} \quad \forall k \in K, j \in N, s \in 1, \dots, S \quad (28)$$

$$\sum_{s \in S} \Pr(C_{2,s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (29)$$

where constraints (23) – (29) modify the deterministic model by addressing inter-station travel time and energy consumption uncertainties. Constraints (23) introduce the variable $C_{s,kj}$, representing the difference between f_{kj} and $(\tau_{s,k} + t_{s,kj})q_{kj}$ for each bus and charger. Constraints (24) require the sum of probabilities of $C_{s,kj} \geq 0$ across all scenarios to exceed a percentage β of total scenarios. Similarly, constraints (25) and (26) define $C_{1,s,kj}$ to account for charging time variability and the remaining battery level, with constraints (27) ensuring the probability of $C_{1,s,kj} \geq 0$ meets β . Constraints (28) and (29) introduce $C_{2,s,kj}$, ensuring the remaining battery level stays above SOC_k^{min} , with a similar probability requirement.

We recognize that solving the stochastic version of the problem requires a considerably larger number of binary variables compared to its deterministic counterpart ($3 \times (|K| \times |N| \times |S|)$ more binary variables). This results in exploring a larger solution space and creating larger rooted trees when utilizing Branch-and-Cut to solve the problem. As a result, addressing the stochastic version is computationally more complex than addressing the deterministic one.

Case Study on the Bus Network of Central Athens

We focus on an area serviced by 280 bus lines, representing the greater Athens metropolitan network. From this total, a subset of 30 in total bus lines is chosen ($|K| = 30$) for the shift of the existing conventional bus fleet to an electric one. These bus lines primarily function between start and end stops within Athens Municipality and neighboring municipalities of its central administrative region, since the electric bus fleet will be confined to these areas. For the candidate charging station physical locations V , which are comprised of nine locations ($|V| = 9$), we consider that each one can host multiple chargers, which in total are twenty ($|N| = 20$); ten slow chargers (set N_1) and ten fast chargers (set N_2). Specifically, charging options #1, #3, #5, #7, #9, #11, #13, #15, #17, and #19 represent slow chargers (set N_1), while options #2, #4, #6, #8, #10, #12, #14, #16, #18, and #20 correspond to fast chargers (set N_2). Thus, $N_1 = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$ and $N_2 = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$. The nine sites (set V) proposed for charger installations

align with the actual bus depots in the Attica region, as Athens’ public transport authority plans to install electric bus charging infrastructure at some of these locations. It is acknowledged that some bus depots are located far from the borders of the municipality of Athens; however, it was deemed appropriate for the model’s candidate charging locations to include all potential options. Each candidate location can typically host up to two charging options (one slow, $\lambda_{1v} = 1$, and one fast charger, $\lambda_{2v} = 1$), except for locations #1 and #2, which can accommodate up to three charging options. Specifically, location #1 can host up to one fast charger ($\lambda_{2v} = 1$) and two slow chargers ($\lambda_{1v} = 2$), while location #2 can host up to one slow charger ($\lambda_{1v} = 1$) and two fast chargers ($\lambda_{2v} = 2$).

In compliance with the Athens Urban Transport Organization (OASA) as of July 2024, some buses have shown no need for charging at the end of their operational day. To align with this data, the frequency of the charging requirements for each bus throughout the day has been determined accordingly, with each bus requiring a single charging session per operational day. Figure 32 presents the detailed layout of the bus network utilized in this case study.

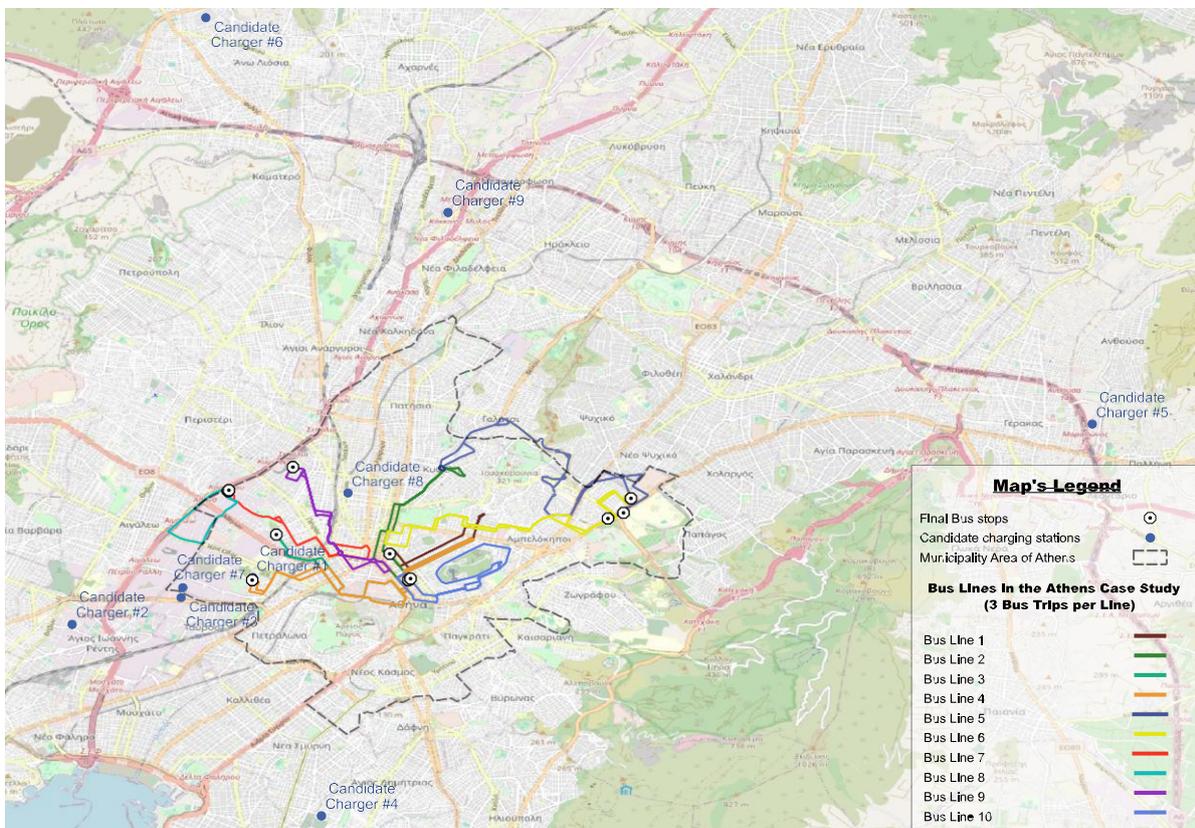


Figure 32: Athens network of the electric bus lines, along with the proposed charging station physical locations

The model framework was examined under both deterministic and stochastic approaches. Table 1 presents the optimal physical locations and types of chargers selected for installation in each approach, while Table 2 provides additional data from the optimal solutions, including the best objective value, total queue waiting time, the number of buses assigned to the nearest charger, and total computation time. Evidently, these solutions differ from each other. It is also noteworthy that the sequence in the charging process for buses assigned to the same charger is successfully achieved across all the approaches.

Table 1: Optimal charging station physical location (V), and charging types (N=N₁ ∪ N₂) across the deterministic and all β(%) variation approaches

Deterministic Approach		β(%) percentage variations															
		100(%)			95(%)			80(%)			75(%)			60(%)			
V	N ₁	N ₂	V	N ₁	N ₂	V	N ₁	N ₂	V	N ₁	N ₂	V	N ₁	N ₂	V	N ₁	N ₂
1	1,3	2	1	1,3	2	1	1,3	2	1	1,3	2	1	1,3	2	1	1,3	2
8	17	18	3	7	8	7		16	8	17	18	3	-	8	8	17	18
			7	-	16	8	17	18				7	15	16			
			8	17	18							8	17	18			

Table 2: Objective function value, total queue waiting time (in minutes), number of buses assigned to their closest charger, and computation time (in minutes) of the optimal solutions across all approaches

	Deterministic Approach	100(%)	95(%)	80(%)	75(%)	60(%)
Objective Function Value:	234.38	1754.79	1265.07	825.04	749.61	572.08
Total Queue Waiting Time (min):	0	1.60	23.49	1.06	3.78	0.77
Buses assigned to their closest charger:	28	22	23	27	25	27
Computation Time (min):	3.62	6.68	16.99	255.98	684.38	5216.55

Under the deterministic approach, the model's optimal solution suggests installing chargers at two physical locations, $V = \{1, 8\}$, out of the nine available. A total of five chargers are allocated across these sites, three slow chargers $N_1 = \{1, 3, 17\}$ and two fast chargers $N_2 = \{2, 18\}$. The solution resulted in an optimal total deadheading time of 156.25 minutes and a zero total queue waiting time for all thirty buses across the ten examined distinct lines. Approximately 53% of the buses are assigned to charging location #8, while the remaining 47% are assigned to location #1. From a cost-efficiency perspective, the objective of minimizing installation costs is practically achieved, as all thirty buses are covered using only two charging sites. Additionally, only two out of the thirty buses are not assigned to their nearest charger. Instead, they are allocated to the second closest site, an outcome that reflects the model's dual objective of minimizing both deadheading and queue waiting times.

A commonality among the deterministic, stochastic, and chance-constrained approaches, as observed in Table 1, is that most solutions recommend installing chargers at two specific physical locations $V = \{1, 8\}$ out of the nine available. Complementary to charging locations #1 and #8, the stochastic approach and several chance-constrained approaches suggest installing chargers at $V = \{3, 7\}$. Table 1 also shows that both the stochastic approach and the chance constrained approach with $\beta = 75\%$ recommend installing a total of eight chargers, while the chance-constrained approach with $\beta = 95\%$ a total of six chargers. It is also worth mentioning that chance-constrained approaches with $\beta = 60\%$ and $\beta = 80\%$ recommend the same five chargers at the same locations as the deterministic approach.

The assignment of each bus $k \in K$ to charger $j \in N$, varies significantly across these approaches. In the deterministic approach and the chance-constrained approaches with $\beta = 60\%$ and $\beta = 80\%$, all buses k are assigned to the charging physical locations $V = \{1, 8\}$. For the remaining chance-constrained approaches, the percentage of buses allocations to stations #1 and #8 is as follows: 90% in the stochastic approach, $\approx 93\%$ in the chance-constrained approach with $\beta = 95\%$, and $\approx 87\%$ in the chance-constrained approach with $\beta = 75\%$. Despite the cost-inefficiency of installing additional charging sites for a small percentage of bus charging demands, the stochastic and chance-constrained approaches allow this to happen.

All solutions in chance-constrained approaches resulted in higher objective function values compared to the deterministic approach, though these values diminish as the chance-constrained

$\beta(\%)$ value decreases. Regarding the total queue waiting time for all thirty buses, it is negligible in most approaches, except for the chance-constrained approach with $\beta = 95\%$, where it reaches 11.59 minutes (see Table 2). However, even in this case, where the total queue waiting time is the highest, the maximum delay a bus experiences before commencing its charging process is approximately seven minutes. This duration is considered acceptable given the various uncertainties that are taken into consideration at these case studies.

Table 2 also shows that in the chance-constrained approaches with $\beta = 60\%$ and $\beta = 80\%$, the minimum number of buses not assigned to their nearest charging station is three, while the maximum (eight buses) occurs, as expected, under the stochastic approach ($\beta = 100\%$). In all approaches, buses that are not assigned to their closest charger $j \in N$ are instead allocated to the second, third, or fourth nearest option. This outcome reflects the model's dual objective of minimizing both deadheading time (t_{kj}) and queue waiting time.

It is important to note that under the stochastic approach and chance constraints nearing $\beta = 100\%$, buses tend to start their charging process (f_{kj}) significantly later in their operation day compared to the deterministic approach. This delay is expected due to the stringent requirement for constraints (24) to be satisfied across the vast majority of the $|S|$ scenarios. As we progress to the chance-constrained approach with $\beta = 60\%$, the f_{kj} values approximate those of the deterministic approach. Based on the observed f_{kj} values, the sequence of the charging process of bus $k \in K$ assigned to the same charger $j \in N$, and the charging completion time (l_{kj}) of all buses within the designated time horizon of $c_l = 12$ a.m. were confirmed. Furthermore, by analyzing the remaining battery levels of all buses k upon arrival at their assigned charging location j across all $\beta(\%)$ variations, along with the downtime resulting from both deadheading and charging under the most stringent approach ($\beta = 100\%$), it is evident that none of the stochastic approaches violate the energy threshold SOC_k^{min} . The most elevated downtimes are observed in buses assigned to slow charging types, as observed.

Besides the initial sampling of $\tau_{s,k}$, $t_{s,kj}$, and e_s , an additional, and independent set of samples is drawn to evaluate the performance of the optimal solutions derived by the stochastic and chance-constrained approaches. These new samples of $\tau_{s,kj}$, $t_{s,kj}$, and e_s , which differ from the original scenarios $|S|$, are used to evaluate the performance of the above optimal solutions.

Regarding the total queue waiting times, the simulation results show non-zero values higher than those in the respective solutions of Table 2. Among all tested approaches, the optimal solution derived from the chance-constrained method with $\beta = 75\%$ provides the best results in terms of total excessive time and queue waiting time. As detailed in Table 1, this approach recommends the same set of charging station locations, $V = \{1, 3, 7, 8\}$, and the same total number of chargers $j \in N$ (8 in total) as the stochastic solution ($\beta = 100\%$). However, the two solutions differ in both the allocation and types of chargers installed at certain charging sites. Notably, the solution for $\beta = 75\%$ recommends a more uniform distribution of the electric bus charging demands across charging sites.

In contrast, the simulation of the $\beta = 95\%$ solution results in the poorest total excessive time and highest total queue waiting time, despite recommending the second most extensive infrastructure configuration. Interestingly, even the simulation results of the chance-constrained approaches with $\beta = 60\%$ and $\beta = 80\%$, which recommend only two, yet highly critical charging locations (#1 and #8), and fewer charging types (five in total), outperform the simulation results of the solution for $\beta = 95\%$ due to a more effective distribution of charging demand at these key locations.

These findings suggest that under time and energy-related uncertainties, the variety of charging types at each charging location is more crucial than the number of charging locations themselves, particularly for key charging sites handling a large share of the total charging demand. Therefore,

it can be inferred that, under time and energy-related uncertainties, the diversity of charging types and the balanced distribution of charging demand across charging locations are more influential factors than the number of charging locations themselves, especially at critical charging sites that accommodate a large share of the total demand.

Modelling Framework Overview & Publications

The aforementioned stochastic EB-CSLP modeling framework, developed during the first year of the metaCCAZE project, was accepted for a Poster Session at the 104th TRB Annual Meeting, held in Washington, D.C., on January 5-9, 2025. Additionally, the model has been expanded into a full-length paper, published in a Q1 peer-reviewed journal *Transportation Research Part C: Emerging Technologies*, under the title: “An exact optimization model for the electric bus charging station location problem under inter-station travel time and energy consumption uncertainties”.

7.1.3. Combined Line Planning and Vehicle Scheduling for a fleet of Electric Buses (CLP-E-VSP)

NTUA developed a two-stage model that addresses the issue of the electric vehicle scheduling problem and modifies the bus line routes according to the charging demand. First, we are scheduling the fleet for each bus line, taking into consideration the charging needs by solving the Electric Vehicle Scheduling Problem (E-VSP), which is solved as a mixed-integer quadratic program (MIQP). The modelling considers the real-world data from the living lab of Athens such as the timetables of the line, the autonomy of the buses, the number of trips performed per day, the trip distances, and the deadheading distances. Then, in a second stage, we examine the modification of each bus line in terms of stops elimination, while considering the passengers' demand for each stop, resulting in a mixed-integer linear program (MILP). The goal of the overall two-stage approach is to minimize the unserved passenger demand that arises from the possible modifications.

Related studies

The LPP (Line Planning Problem) is considered to be a fundamental problem in the field of strategic planning for Public Transport Services. Several approaches and mathematical formulations have been proposed, considering conventional bus fleets. An in-depth analysis of past studies related to basic line planning models can be found in the seminal work of Schöbel (2012). Borndörfer et al. (2008) presented two new multi-commodity flow integer linear models, which compute optimal line routes and passengers' paths. In the work of Lyu et al. (2019) a Multinomial Logit model (MNL) was developed that investigates the probability of a passenger choosing a Custom Bus service over a set of alternative modes. The Vehicle Scheduling Problem (VSP) is a fundamental step in the tactical level of public transport network planning. Chao and Xiaohong (2013) proposed a modification of the existing VSP model for the single depot vehicle scheduling problem, with route time constraints. Teng et al. (2020) developed a multi-objective model with an integrated approach to the VSP for electric buses, for a single bus line. The objectives concern the smoothing of the vehicle departure intervals, the minimizing of the required number of vehicles and total charging costs. Gkiotsalitis, Iliopoulou, & Kepaptsoglou, (2023b) introduced an exact approach on the multi-depot electric vehicle scheduling problem, considering time windows in which vehicles can recharge at charging stations located at any point of the service operation area (EB-MDVSPTW). Some other studies combine optimization models with heuristic methods. Wu et al. (2022) introduced a bi-objective multi-depot electric vehicle scheduling problem (MDVSP). The objectives were to minimize the total operation cost and to minimize the peak load resulting from recharging activities.

Formulation

In this sub-section the formulation of the CLP-E-VSP problem is presented. It is modelled as a two-stage problem, firstly formulating the E-VSP as a MIQP model, and then, on a second stage, formulating a MILP model for the LPP. The problem concerning the electric vehicle scheduling model (E-VSP) is described as follows:

In this problem definition, we consider all the activities that need to be performed by vehicles as nodes in a graph. Firstly, we have a pre-defined number of trips V according to the timetable of each line. Secondly, we consider a set F , which represents all of the possible charging events at the depot. Each possible charging event $i \in F$ can start within the time window $[a, b]$. In this time window, a is considered as the time point where bus $k \in K$ has performed trip $i \in V$ and has travelled from the final bus stop to the charging destination, and b is the time point after the mentioned travel time, plus the time needed to complete the charging procedure. In addition, we consider two sets O, D , representing the depot and the last stop on each line. Evidently, all the tasks that a bus $k \in K$ can perform belong to the set N , where $N = V \cup F \cup O \cup D$ and $V \cap F \cap O \cap D = \emptyset$.

The set of potentially required buses $k \in K$ to perform tasks $i \in N$ consists of two sub-sets: K_1 and K_2 . While set K_1 represents the vehicles that did not perform any tasks $i \in N$ in a specific time-slice, set K_2 includes the vehicles that did. This separation of set $k \in K$ is necessary, because, in the formulation of the E-VSP, we divide the operating hours of each line in parts consisting of time-slices. Therefore, it becomes crucial to keep track of the required buses per slice. The E-VSP is solved in each time-slice with the use of a quadratic model. Using a heuristic approach, we assign buses to sets K_1, K_2 and we also succeed to transfer the time and battery level state of a bus from one slice to another. The reason behind using a heuristic and solving the E-VSP in slices is that the E-VSP is an NP-hard problem. With this approach and formulation, we manage to solve the problem without computational limitations, finding the best solution per time-slice and a comprehensive solution for the entire operating day of a line.

Considering the parameters of the problem, the duration of a task $i \in N$ that a bus k performs is τ_i' . In addition, the travel time between the end location of task $i \in N$ and the start location of task $j \in N$ is τ_{ij} . The energy consumption caused by the distance covering between the locations of nodes (i, j) is θ_{ij} and the distance covered is d_{ij} . The battery consumption per travelled distance is e and the charging rate is c . In addition, the minimum allowed state of charge of a bus $k \in K$ is SOC_{min} and the maximum allowed state of charge is SOC_{max} . In order to successfully transfer the data regarding the location, battery level and time of each bus $k \in K$ performing task $i \in N$ from one time-slice to another, we have established some parameters. The state of charge for the buses at

the beginning of a time-slice is $E_{k,t}$, the location of the bus (either the first or final stop of the line or the depot) is L_k and the time that each bus k is available in each time-slice is $Time_{k,t}$.

MODELED NETWORK REPRESENTATION

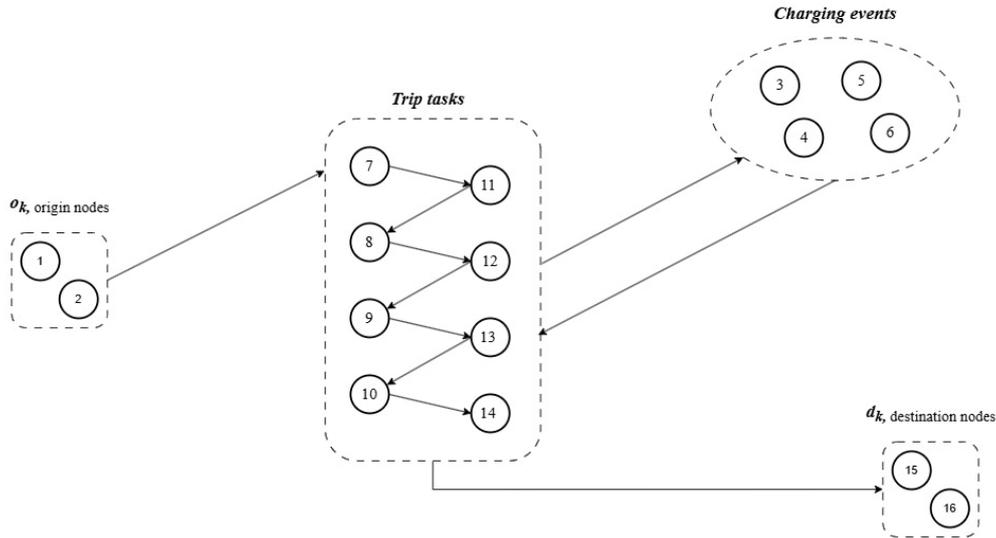


Figure 33: A generic representation of the modelled problem

In Figure 33, a generic graphical representation of the modelled problem is provided. The $o_k \in \mathcal{O}$ is the origin point, consisting of two nodes. The node indicated with "1" (i.e. first node) is the depot and the node indicated with "2" (i.e. second node) is the last bus stop of every trip for every bus line examined. The d_k is the destination point, which consists of the two last nodes of the model. These two last nodes, they also represent the depot and the last bus stop, respectively. This formulation keeps track of the location of bus k when transferring from one time-slice to another. For example, if a bus k is located at the final stop of its route at the end of the time slice, then, at the next time-slice, it is located at the second node of the origin point o_k . Nonetheless, all buses have to complete their whole trip at the end of a time-slice, meaning they are either located at the last stop of their route or they managed to transfer to their corresponding depot.

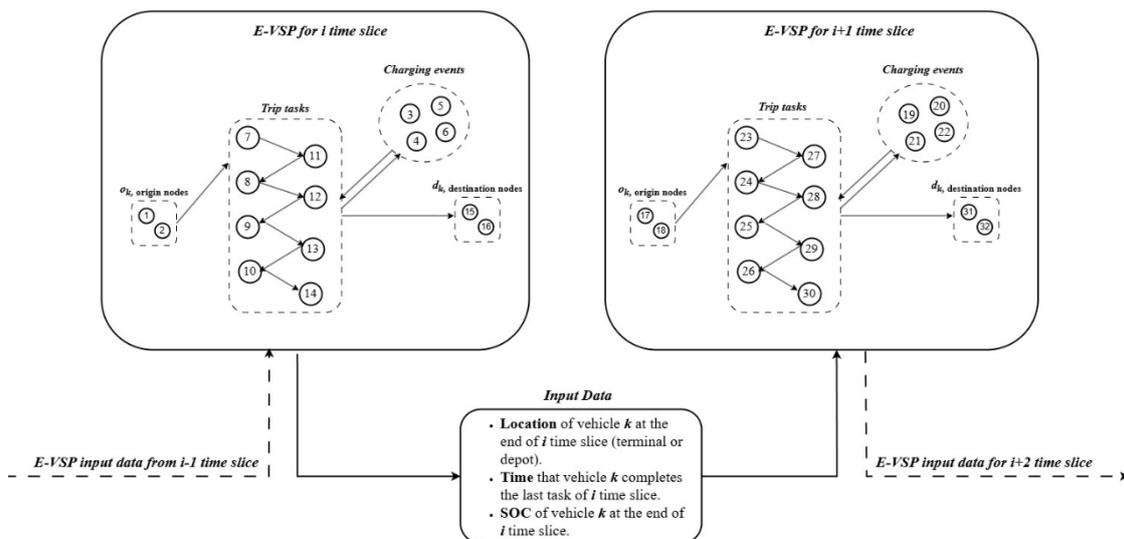


Figure 34: A generic representation of the time-slice modelling approach. Each time slice solves the E-VSP problem based on the location, energy level and time status of a vehicle k provided by the previous time-slice.

The formulation presented in Figure 34 is being solved in each time slice with a mixed-integer quadratic programming model, giving us the best solution for each time slice. That is, the minimum buses required for the service of the trips in each time slice. This result comes considering the deadhead distance, the timetable and charging limitations. The mathematical model for this is as follows:

$$\min \left(\sum_{i \in \mathcal{O}} \sum_{j \in (\mathcal{F} \cup \mathcal{V} \cup \mathcal{D})} \sum_{k \in \mathcal{K}} x_{ijk} + p_1 \sum_{i \in (\mathcal{F} \cup \mathcal{V})} \sum_{j \in \mathcal{D}} \sum_{k \in \mathcal{K}} T_{jk} x_{ijk} + p_2 \sum_{i \in (\mathcal{F} \cup \mathcal{V})} \sum_{j \in \mathcal{D}} \sum_{k \in \mathcal{K}} t_{ij} x_{ijk} - p_3 \sum_{i \in \mathcal{D}} \sum_{k \in \mathcal{K}} SOC_{ik}^s \right) \quad (1)$$

s.t.:

$$\sum_{j \in \mathcal{N}} x_{ojk} \leq 1 \quad \forall k \in \mathcal{K}_1 \quad (2)$$

$$\sum_{j \in \mathcal{N}} x_{ojk} = 0 \quad \forall k \in \mathcal{K}_1 \quad (3)$$

$$\sum_{j \in \mathcal{N}} x_{okj} \leq 1 \quad \forall k \in \mathcal{K}_2 \quad (4)$$

$$\sum_{j \in \mathcal{N}} x_{okj} = 0 \quad \forall k \in \mathcal{K}_2 \quad (5)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{O}} x_{ijk} = 0 \quad \forall k \in \mathcal{K} \quad (6)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}, \forall j \in \mathcal{O}, \forall k \in \mathcal{K} \quad (7)$$

$$\sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \leq 1 \quad \forall j \in \mathcal{D}, \forall k \in \mathcal{K} \quad (8)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{D}, \forall j \in \mathcal{N}, \forall k \in \mathcal{K} \quad (9)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}} x_{ijk} = \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} \sum_{j \in \mathcal{D}} x_{ijk} \quad \forall k \in \mathcal{K} \quad (10)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{F}, \forall j \in \mathcal{F}, \forall k \in \mathcal{K} \quad (11)$$

$$\sum_{i \in \mathcal{N}} x_{ijk} = \sum_{i \in \mathcal{N}} x_{jik} \quad \forall j \in \mathcal{F} \cup \mathcal{V}, \forall k \in \mathcal{K} \quad (12)$$

$$\sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} x_{ijk} \leq 1 \quad \forall j \in \mathcal{F} \cup \mathcal{V} \quad (13)$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} = 1 \quad \forall j \in \mathcal{V} \quad (14)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}} \sum_{k \in \mathcal{K}} x_{ijk} \leq B \quad (15)$$

$$x_{ijk} = 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}, \forall j \in \mathcal{N}: i = j \quad (16)$$

$$T_{jk} \leq M(1 - x_{ijk}) + dt_j \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (17)$$

$$T_{jk} \geq -M(1 - x_{ijk}) + dt_j \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (18)$$

$$T_{jk} \leq M \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (19)$$

$$T_{ik} + t'_i + t_{ij} - T_{jk} + \sigma_{ijk} \leq 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (20)$$

$$T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} - T_{jk} + \sigma_{ijk} \leq 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{V} \cup \mathcal{D} \quad (21)$$

$$\sigma_{ijk} \leq M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (22)$$

$$\sigma_{ijk} \geq -M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (23)$$

$$T_{ik} \leq Time_k + M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (24)$$

$$T_{ik} \geq Time_k - M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (25)$$

$$T_{jk} \leq T_{ik} + t'_i + t_{ij} + M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{V}, \forall j \in \mathcal{D} \quad (26)$$

$$T_{jk} \geq T_{ik} + t'_i + t_{ij} - M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{V}, \forall j \in \mathcal{D} \quad (27)$$

$$T_{jk} \leq T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} + M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{D} \quad (28)$$

$$T_{jk} \geq T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} - M(1 - x_{ijk}) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{D} \quad (29)$$

$$SOC_{jk}^s \geq SOC_{min} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{D} \quad (30)$$

$$SOC_{jk}^s \leq M \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (31)$$

$$SOC_{ik}^e \leq E_k + M \left(1 - \sum_{j \in \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \quad (32)$$

$$SOC_{ik}^e \geq E_k - M \left(1 - \sum_{j \in \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \quad (33)$$

$$SOC_{ik}^e \leq M \sum_{j \in \mathcal{N}} x_{ijk} \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V} \quad (34)$$

$$SOC_{jk}^s \leq SOC_{ik}^e - \theta_{ij} + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (35)$$

$$SOC_{jk}^s \geq SOC_{ik}^e - \theta_{ij} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (36)$$

$$g_{jk} \leq SOC_{jk}^s - SOC_{jk}^e + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (37)$$

$$g_{jk} \geq SOC_{jk}^s - SOC_{jk}^e - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (38)$$

$$g_{jk} \leq e_j + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{V} \quad (39)$$

$$g_{jk} \geq e_j - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{V} \quad (40)$$

$$g_{jk} \leq SOC_{jk}^s - SOC_{max} + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \quad (41)$$

$$g_{jk} \geq SOC_{jk}^s - SOC_{max} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{V}} x_{ijk} \right) \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \quad (42)$$

$$g_{jk} \leq M \sum_{i \in \mathcal{N}} x_{ijk} \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (43)$$

$$g_{jk} \geq -M \sum_{i \in \mathcal{N}} x_{ijk} \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (44)$$

$$y_k M \geq \sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{N}} x_{ijk} \quad \forall k \in \mathcal{K} \quad (45)$$

$$y_k \leq \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} x_{ijk} \quad \forall k \in \mathcal{K} \quad (46)$$

Objective function (1) is used to minimize the number of vehicles required to cover the scheduled trips. Constraints (2) to (16), make sure that all the conditions for the proper operation of the network, that is the activities a vehicle can perform represented by nodes in graph, are being met. Constraints (17) to (27) make certain that all the tasks a vehicle can perform comply with the

timetable given by the public transport operator. Furthermore, constraints (28) to (46) ensure that all vehicles during their operation have a battery level greater than SOC_{min} and less than SOC_{max} . In addition, they make sure that the energy level of each vehicle is being updated after performing a task, which could be a trip or a charging event.

For the LPP formulation we consider a number of lines L . For every line we propose different modification options $o \in O$. Each option is developed through the elimination of bus stops. Thus, every line option has a different line length. This means that the duration of the trips, the number of stops, the charging demand of vehicles, the number of trips they conduct, and essentially the required number of vehicles change. Considering the parameters of the problem, the E-VSP model presented above solves the vehicle scheduling problem for each one of the options per examined line and, accordingly, offers the number of vehicles required. The number of passengers not served per line modification option is d_{ij} . Further, it should be clarified that the available fleet of electric buses B for solving the LPP is pre-determined. The mathematical formulation for this is as follows:

$$\min \sum_{i \in \mathcal{L}} \sum_{j \in O} x_{ij} d_{ij} \tag{47}$$

Subject to the constraints:

$$\sum_{j \in O} x_{ij} = 1 \quad \forall i \in \mathcal{L} \tag{48}$$

$$\sum_{i \in \mathcal{L}} \sum_{j \in O} x_{ij} o_{ij} \leq B \tag{49}$$

The objective function (47) aims to minimize the total number of unserved passengers due to the modification of the line plans. Constraints (48) and (49) ensure that for every line a modification is being selected and that for all the lines according to their modification, the number of required buses is equal or less than the available vehicles. The LPP is a binary linear program which can be solved to global optimality. We note, though, that the input parameter values o_{ij} are determined by solving a series of E-VSP problems for every line modification option j . The combination of the LPP and E-VSP problems allows us to make line modifications that minimize the impact on unserved passenger demand when transitioning from a conventional to an electric bus fleet of the same size. With both the E-VSP and LPP models presented in this section, the two-stage approach can be depicted in Figure 35.

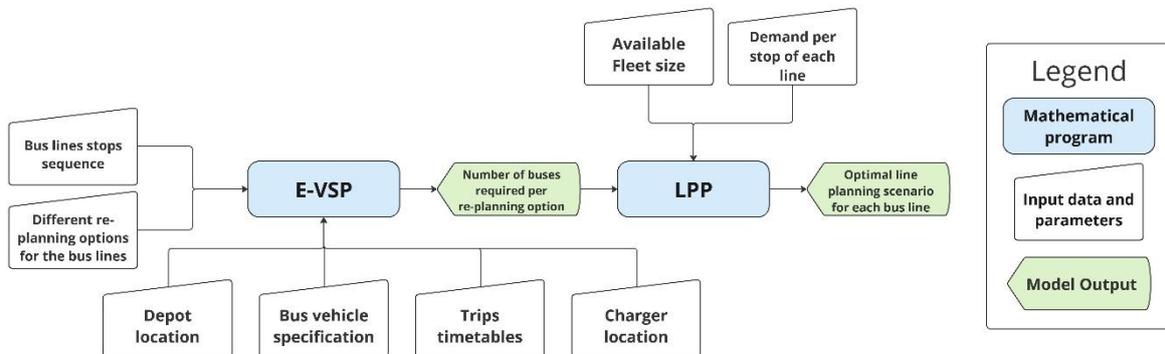


Figure 35: A flowchart of the two-stage process including the E-VSP and LPP models

Case Study

In our numerical experiments, we illustrate the model’s application in three different bus lines spread out in the Athens Metropolitan area, see Figure 36. In each bus line, we examine five different line modification options. The modifications arise from the elimination of bus stops in the existing layout of the bus lines. In all options, the eliminated bus stops are located in the same geographical area, at the end of each direction of the bus line and are equal per direction. For every line and every scenario examined, we get as a result the required number of electric buses for the operation of the line by solving an E-VSP problem. Thus, there is a limit to the number of vehicles that are necessary to operate lines in every scenario. Each modification, according to the length of the bus line, can offer a different result, giving us the opportunity to evaluate which is the best new layout of the line while considering the number of passengers served. The lengths of the three examined lines are 15km, 20km and 28km, respectively, per direction.

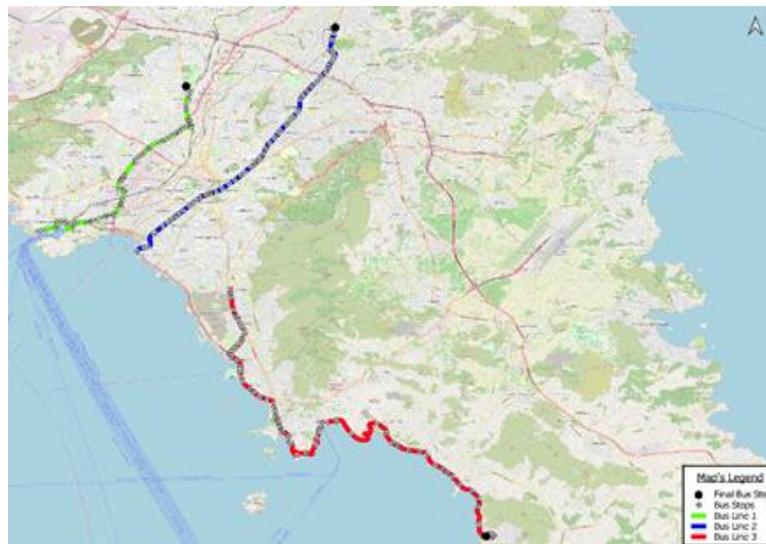


Figure 36: The three bus lines examined under this case study

With the implementation of this two-stage model mentioned above, we try to balance the cost of the electric bus fleet and the unserved passenger demand. Clearly, different numbers of available electric buses offer a different solution to our problem. Specifically, if the number of available vehicles changes, different options are selected, resulting in different line layouts, different numbers of unserved passengers, and essentially different costs for the operation of the line. For this reason we implement a model that offers an optimal solution for all the lines examined, while considering a bounded fleet of electric buses and the passenger demand. This is achieved by applying the E-VSP model in every examined modification of the bus lines and deciding according to the LLP model which is the optimal option for all bus lines, while considering a bounded number of available vehicles that can serve all three bus lines. In Table 3, we showcase the results of our model for an available fleet consisting of 52 electric buses

Table 3: Best solution of the combined line planning and vehicle scheduling problem for a fleet of 52 electric buses

	Bus Line 1 Option 3	Bus Line 2 Option 5	Bus Line 3 Option 5	Total
Buses	12	21	19	52
Unserved Passengers	105	169	203	477

We applied the E-VSP model for every option of the solution, and for each bus line we determined how many buses are needed. For every vehicle we keep track of the energy level of the battery before and after performing any task, SOC^b and SOC^e respectively. A task can be a trip or a charging

event (c.e.) that a vehicle performs during its operation. We also keep track of the time that a vehicle arrives at the node of each task. For bus line 1 and option 3, that is selected in the solution of the problem for a fleet of 52 buses, 12 buses are required. For example, vehicle 1 performs 12 tasks during its day of operations, where one of them is a charging event. It follows the path 23 → 7 → 32 → 13 → 38 → c.e. → 81 → 63 → 88 → 70 → 94 → 76. Its first trip, numbered as 23, starts at 377.73 minutes. We consider midnight as 0 minutes, so consequently 377.73 minutes is approximately 6:30 AM. The numbering of trips is not random; it is matched to the timetable given by the public transport operator. Therefore, trip number 23 is the twenty-third trip of bus line 1 starting at 6:30 in the morning.

All vehicles start their operation fully charged, meaning that their energy level at the depot is 350 kWh. For example, vehicle 1 starts its first trip, number 23 with a battery level of $SOC_s=341.54$ kWh. The difference between 350 kWh and 341.54 kWh equals to the energy consumption caused by the distance covered between the depot and the location of the start of the bus line. The energy level of the bus after performing trip 23 is $SOC_e=316.63$ kWh. Vehicle 1 returns to the first bus stop after performing trip 23 and proceeds to perform trip 7. Consequently, the energy level at the first bus stop of trip 7, $SOC_s=316.63$ kWh, is equal to the energy level at the end of trip 23. After performing trip 38, the vehicle heads to the corresponding depot for charging at 724.78 minutes, approximately after 6 and a half hours of operation. Its energy state when it arrives at the charging depot is 204.6 kWh and after the charging procedure it reaches its full capacity, 350 kWh. In our model we assumed that the minimum energy level for the operation of the lines is 70 kWh. That means that they do not necessarily reach their minimum energy limit when they head for charging. It should also be clarified that when a vehicle is charged, the charging time slot is fully occupied by the vehicle, regardless of the actual charging duration. This helps us match the vehicle to the corresponding charger according to the timetable of the line and the charging timetable, allowing the realistic modelling of charging queues.

Lastly, we provide a sensitivity analysis concerning the behaviour of our model when the size of the available fleet changes. The analysis ranges from the benchmark of 52 vehicles, where below this point the model is infeasible, until the increase of the fleet size does not have an impact on the number of unserved passengers (60 vehicles).

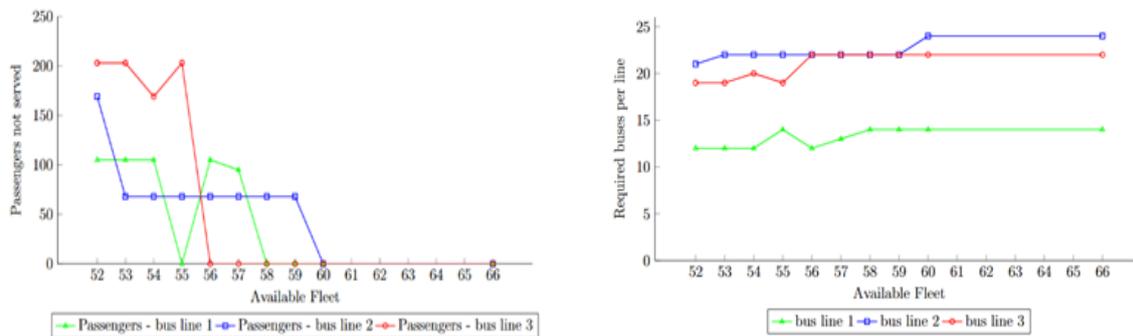


Figure 37: Graphic representations of the number of passengers not served when considering a range of available vehicles of the number of required buses and the number of passengers not served considering the size of the available fleet for every bus line examined

Figure 37 offers a visual representation of how the availability of vehicles affects the number of unserved passengers. The figure shows that for a small fleet, the number of unserved passengers is higher compared to the case of larger fleet sizes. As one can observe, the selection of options for one bus line, therefore the number of unserved passengers, is not independent of the selection of options for the other lines. For example, for the case of 55 vehicles, bus line 1 serves all

passengers, indicating that the model selects option 1. However, bus line 3 serves fewer passengers compared to the case of the fleet of 54 vehicles and bus line 2 serves the same number of passengers for both cases. It also shows, that if we have more available vehicles, these vehicles will be distributed to different lines to reduce the total number of unserved passengers. This distribution of the extra vehicles to the bus lines is not even, as the extra vehicles are distributed so as to satisfy as much passenger demand as possible.

To conclude, we showed that with significant line modifications that will result in 477 unserved passengers one can offer a service by using only 52 electric buses. Increasing the available number of electric buses to 56 will result in only 173 unserved passengers, whereas increasing this number further to 60 electric buses will allow us to operate all lines without performing any modifications resulting in unserved passengers. Here, it is worth to highlight that a conventional fleet of non-electric vehicles could operate all lines without requiring any line modifications with 17 buses less (43 instead of 60). NTUA completed the development of the model and explored potential line planning modifications for the newly introduced electric buses in Athens within the 18 first months of metaCCAZE.

7.1.4. Analyzing the Effects of Pick-Up and Drop-Off (PUDO) Duration and its Stochasticity on Mobility-On-Demand Services

Introduction and Related Studies

Shared mobility solutions, especially Mobility-on-Demand (MoD) services like ride-hailing and ride-pooling, have attracted increasing attention for their potential to reduce car ownership, cut environmental impact, and support public transit. These services could also enhance transportation accessibility and provide economic benefits to users (Shaheen et al., 2016). With automation reducing operational costs, Autonomous MoD (AMoD) services are expected to expand significantly (Bösch et al., 2018; Dandl & Bogenberger, 2019). This motivates a closer examination of various MoD operational aspects, such as the selection of pick-up and drop-off (PUDO) locations (Fielbaum et al. 2021), the aggregation of multiple requests to the same PUDO process (Hunter et al. 2024), and travel time stochasticity (Wolf et al. 2023).

The metaCCAZE team focused on a largely neglected attribute of PUDO processes: the impact of their duration (and variability) on system performance and Level of Service (LoS). Most studies have considered deterministic PUDO durations, often arbitrary and constant across all requests (Dandl et al. 2017; Stiglic et al. 2015; Zwick and Axhausen 2020). However, there are no studies considering heterogeneous and/or stochastic PUDO duration, nor investigating in depth the operational and LoS impacts of the mismatching between the real duration of the processes (t_i^{PUDO}), and the duration considered by the operator for its fleet management ($t_i^{PUDO,assumed}$).

Exploring these aspects is relevant, as there is extensive research on dwelling duration in the Public Transportation field (Rashidi et al. 2023). Moreover, in the context of the metaCCAZE project, this initiative aligns with Munich's Dynamic Curbside Management (DCM) Use Case, as these areas could be used in the future by MOD vehicles for passenger pick-up and drop-off. Therefore, understanding how stop duration impact system performance and passengers' LoS can inform the design of effective DCM strategies and infrastructure.

In the following subsection, we briefly introduce the methodology and results of a simulation study that seeks to isolate and analyse the effects of heterogenous and stochastic PUDO duration processes in a ride-pooling MoD service. Specifically, we seek to determine if stop duration stochasticity degrades service performance and if operators could benefit from predicting stop durations using data-driven models or using constant values derived from actual stop duration distributions. Due to space limitations in this deliverable, we provide only a short overview of the

study here. Interested readers are referred to the full peer-reviewed study, which has been accepted for publication in an academic journal (Álvarez-Ossorio et al. 2025).

Methodology

PUDO Duration Generation

One of the primary challenges in assessing the impact of PUDO durations on MoD services is the scarcity of high-quality empirical data. While datasets like the New York City taxicab and Chicago taxi and ride-sharing datasets are popular in MoD studies, they lack fine-grained temporal data on individual stop durations. The most representative source we found is Machado-León et al. (2024), which provides descriptive statistics and parametric cumulative distribution functions based on several thousand PUDO observations in Seattle, USA. However, this study does not disclose individual stop durations.

To address this data gap, we generated synthetic PUDO duration data using a log-normal distribution function reported by Machado-León et al. (2024). This distribution was chosen because it adequately fits the empirical data. The probability density function (pdf) of a log-normal distribution is defined as:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right), \quad x > 0(1)$$

where μ is the mean and σ is the standard deviation of the natural logarithm of the variable. Machado-León et al. (2024) reported parameters $\mu = -0.69$ and $\sigma = 1.14$, corresponding to a mean of 57 seconds and a variance of 2.46 min² of the stop duration distribution.

To ensure the generalizability of our results, we decided not to rely solely on this specific distribution, which might be describing the local context of the study area, but to also explore different underlying PUDO duration distributions. Thus, we estimated the μ and σ parameters of log-normal distributions that kept the same variance (2.46 min²) and had the mean pivoted by -30%, +30%, and +50%. These distributions are shown in Figure 38. We excluded -50% pivoting as that distribution leads to excessively, unrealistic, low PUDO durations. Given that log-normal distributions exhibit large right arms, to avoid drawing unrealistically large values, we truncated each of the distribution with a minimum and maximum value. The former corresponds to the 5th percentile reported in Machado-Leon et al. (2024), 5 s; whereas the latter was calculated specifically for each distribution based on 15,000 random draws so that, after truncating the distribution, its 95th percentile would correspond to the 95th percentile reported in Machado-Leon et al. (2024), 130 s.

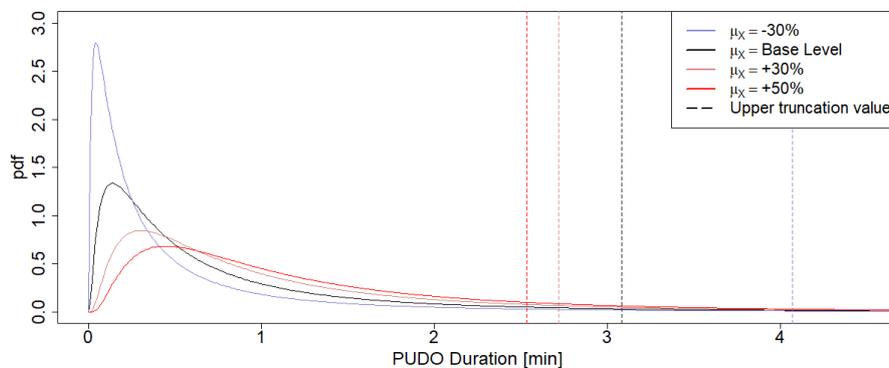


Figure 38: PDFs of Log-normal distributions with pivoting mean and constant variance

Black Box Prediction Model

We aimed to determine whether predicting the duration of each specific request using a data-driven model would benefit operators. Although some MoD operators likely have the necessary data to train such models, e.g., user characteristics, historical data of previous trips by the user, type of stop location, etc., it is not publicly available to researchers. Therefore, we simulated a "black box prediction model" that predicts the duration of a request i with a maximum allowed relative error δ (ranging from 0% to 100%). The predicted duration ($\widehat{t}_i^{\text{PUDO}}$) is drawn from a continuous uniform distribution (\mathcal{U}) with boundaries dependent on δ and the real PUDO duration (t_i^{PUDO}).

$$\widehat{t}_i^{\text{PUDO}} = \mathcal{U}\left((1 - \delta) \cdot t_i^{\text{PUDO}}, (1 + \delta) \cdot t_i^{\text{PUDO}}\right) \quad (2)$$

To ensure realism, we assumed the operator would not deploy a model predicting values outside the solution space defined by the truncation values in the PUDO duration generation process. Thus, predictions (e.g., draws) outside this range were adjusted to the truncation values.

MoD Simulation Tool: FleetPy

We utilized the open-source simulation framework FleetPy, one of metaCCAIZE's assets, to simulate an MoD provider's interactions with customers and fleet control planning (Engelhardt et al. 2022). Customers made app requests specifying their latest pick-up time and maximum in-vehicle time. The operator responded with offers or rejections based on these constraints. The operator batches requests and optimized vehicle-user assignments with the primary objective to serve as many customers as possible while minimizing the system time, e.g., the amount of time all vehicles must drive to serve these customers.

FleetPy separates fleet control and simulation. At the end of a fleet control step, a vehicle plan is handed to the simulation, the vehicle drives to the next stop, and customer boarding/alighting is simulated. During this study, the metaCCAIZE team extended the framework to handle different PUDO duration values for boarding and alighting and to deny the operator access to simulated boarding times (e.g., it only knows the predicted boarding time). Importantly, a solution from a previous time step is guaranteed to remain feasible if the simulation of the system is according to the fleet control plan. However, with stochasticity in the system, delays compared to the original plan may become apparent from one optimization step to the next. It might be possible to salvage infeasible customer constraints by reassigning these customers; however, it is also possible that no feasible solution can be found anymore. In these cases, we followed the approach by Wolf et al. (2023): the vehicle currently containing the request for which no feasible solution can be found "locks" its current plan and does not allow any further reassignments. The idea is to avoid further delays and to serve the respective customers as quickly as it is still possible.

Case Study, Demand Data, Experimental Design, and Simulation Setup

Case Study and Demand Data

We simulated the operation of a MoD service within Munich, Germany. The network includes over 19,000 nodes and 40,000 edges, with travel times updated hourly based on a traffic micro-simulation model (Dandl et al. 2017). Pre-processed travel-time tables were used to reduce simulation time (Engelhardt et al., 2022b). For demand data, we utilized the total private vehicle trip demand in Munich, sampled at various fractions. We focused exclusively on private vehicle trips rather than the full modal demand in Munich because the private vehicle data was available at a finer spatial and temporal granularity. The road network of Munich and the private vehicle trip demand data used in this study are publicly available in an open-access repository (Engelhardt R. and Dandl F. 2025).

To reduce computational efforts, simulations were restricted to the 06:00 a.m. to 10:00 a.m. period, excluding the first and last hour for performance metrics calculation. The number of requests per 1% demand ranged between 100 and 200 per 15-minute interval. The MoD provider operates a **ride-pooling** service, creating offers in 60-second batches and reassigning feasible schedules every 60 seconds (Engelhardt et al. 2019; Wolf et al. 2023). A single operator with vehicles having four passenger seats was considered, with a maximum 8-minute waiting time and a 40% maximum additional detour for assigned requests.

Experimental Design

The experimental design aimed to replicate differences between real (simulated) and assumed (by the operator) PUDO durations. Five scenario families were considered, based on whether the actual duration is deterministic or stochastic and whether the operator's duration value is constant or based on a prediction model. Each family was simulated for three demand levels (1%, 5%, 10%) and three demand seeds per level.

The models were named based on the operator's information type: *fully-informed* (exact PUDO duration known), *model-informed* (predicted duration), *uninformed* (constant arbitrary duration), and *naive-informed* (constant value corresponding to the empirical mean). Fleet sizes were calibrated to accept at least 90% of requests, resulting in fleets of 235, 825, and 1550 vehicles for the 1%, 5%, and 10% demand scenarios, respectively. A total of 555 different scenarios were studied, summarised in Table 4. Simulations were conducted on the Linux Cluster of the Leibniz Supercomputing Centre.

Table 4: Design (555 scenarios)

FAMILY OF SCENARIOS (N. OF SCENARIOS)	DEMAND LEVEL	SEED	REAL PUDO DURATION		PUDO DURATION FOR FLEET CONTROL	
			Type	Values	Type	Values
Fully-informed deterministic scenarios (51)	{1%, 5%, 10%}	{1, 2, 3}	Deterministic	{30 s, 45s, 60s, Empirical Mean of four Log-Normal distributions}	Constant	Same as real duration
Stochastic				Four combinations of Log-Normal distribution parameters		Constant
			Constant		Empirical mean duration of all requests	
			Predicted		Max. relative error: {100%, 75%, 50%, 40%, 30%, 20%, 15%, 10%, 5%}	
Fully-informed stochastic scenarios (36)			Predicted	Max. relative error: {0%}		

Results and discussion

This section provides a short description of some selected results from the study. A more thorough presentation of the results is available in the published scientific article (Álvarez-Ossorio et al. 2025).

Metrics for Evaluating the Impacts of PUDO Duration on MoD Services

MoD services are complex systems with numerous impacts on operators, users, and society. Evaluating their performance involves trade-offs among these perspectives. In this study, we focus on three key performance indicators (KPIs): i) the share of accepted requests, ii) the absolute deviation between the drop-off time provided to the passenger in the operator's offer (e.g., at the time of the request) and actual drop-off time, iii) and the share of requests dropped off later than their latest arrival time constraint (Late Drop-Off share). The second and third KPIs will be referred to as LoS metrics.

Regarding the Absolute Drop-Off Time Deviation metric, it is crucial to recognize that in ride-pooling services, the dynamic nature of fleet management often leads to significant deviations, even with deterministic travel and PUDO duration times (Hyland and Mahmassani 2017). The main reason is that, after an offer is created and a request is assigned to a vehicle, new requests can still be inserted into the vehicle's schedule if the temporal constraints of already-accepted requests are met. Additionally, while travel times in the network are dynamic, fleet control decisions are based on the network conditions at the current simulation step. This means that trips spanning multiple network-state periods may not perfectly align with the operator's initial plans, resulting in delayed arrivals for some requests.

In the following figures, the height of the bars represents the average value of a metric grouped by demand size. Depending on the case, it may also consider the stochastic distribution and the maximum relative error of the black box prediction model. The whiskers indicate the maximum and minimum values for each metric and group.

Effect of Increasing PUDO Duration

We analyzed the impact of PUDO duration magnitude using two scenario families: Fully-informed deterministic and Fully-informed stochastic. In both scenarios, the operator has perfect knowledge of each request's duration, allowing us to isolate the effect of PUDO duration magnitude. Our results, in Figure 39 and Figure 40, indicate that increasing PUDO duration, both in deterministic and heterogeneous scenarios, reduces the share of served requests, with larger drops observed in higher demand scenarios. For instance, a 30-second increase in PUDO duration results in a 1.6% reduction in served requests for 1% demand scenarios and a 2.7% reduction for 10% demand scenarios. This trend aligns with previous studies (Bilali et al., 2020; Zwick & Axhausen, 2020) and highlights the importance of minimizing PUDO durations to optimize vehicle utilization.

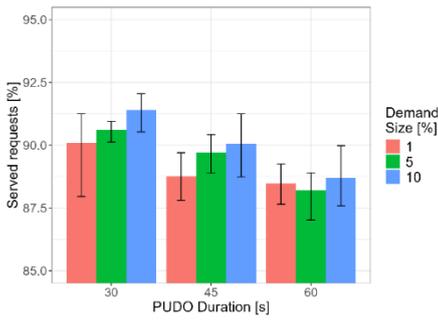


Figure 39: Served requests with constant PUDO duration across requests

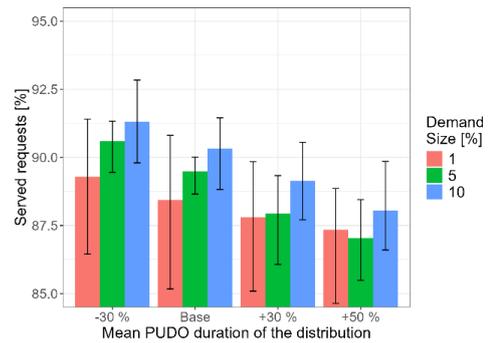


Figure 40: Served Requests with Heterogeneous PUDO duration

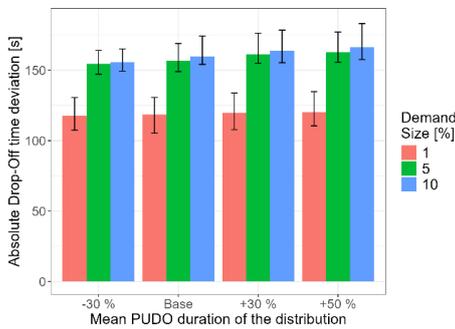


Figure 41: Abs. DO time dev. with constant PUDO duration across requests

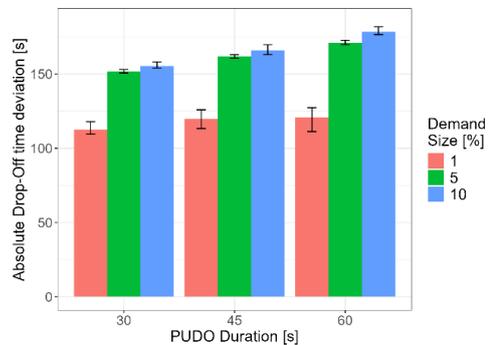


Figure 42: Abs. DO time dev. with Heterogeneous PUDO duration

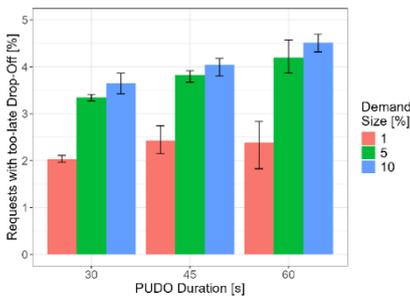


Figure 43: Late Drop-Off share with constant PUDO duration across requests

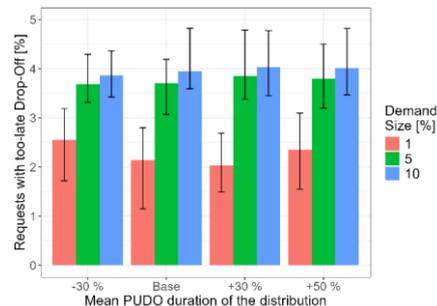


Figure 44: Late Drop-Off share with Heterogeneous PUDO duration

Figure 45: Impacts of increasing PUDO duration considering fully-informed operators

As shown in Figure 39 to Figure 44, longer PUDO durations also lead to greater deviations in drop-off times and higher Late Drop-Off shares, indicating a deterioration in the LoS. The absolute drop-off time deviation increases because the boarding and alighting of requests inserted into a vehicle's schedule after an offer has been generated take more time, increasing the difference from the offered arrival time. Additionally, the observed increase in the share of requests arriving too late is explained by the fact that, despite the operator being fully informed, longer PUDO durations result in more trips spanning multiple network-state periods. Consequently, the differences between planned and realized trips become more significant.

Effect of PUDO Duration Stochasticity

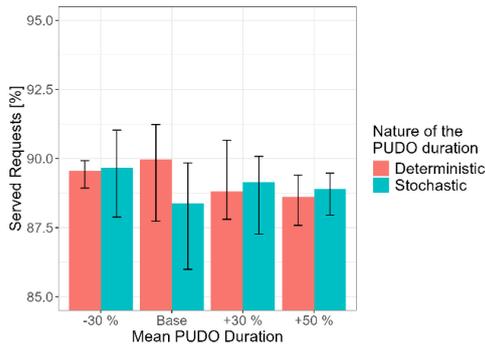


Figure 46: Served requests with constant PUDO duration across requests

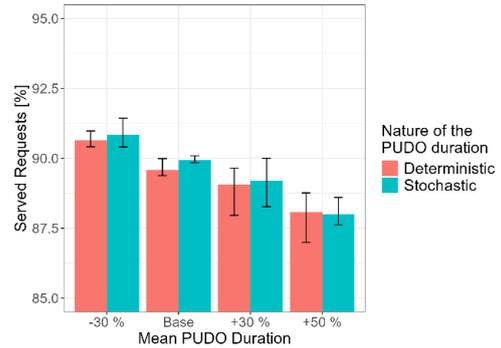


Figure 47: Served Requests with Heterogeneous PUDO duration

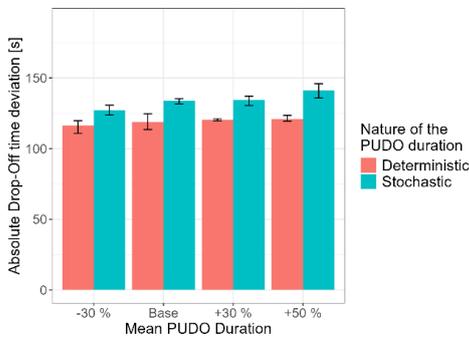


Figure 48: Abs. DO time dev. with constant PUDO duration across requests

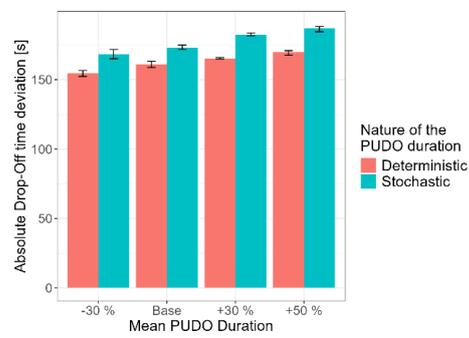


Figure 49: Abs. DO time dev. with Heterogeneous PUDO duration

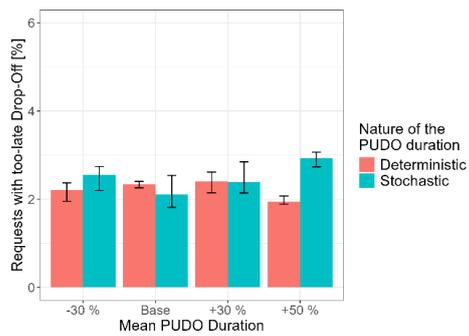


Figure 50: Late Drop-Off share with constant PUDO duration across requests

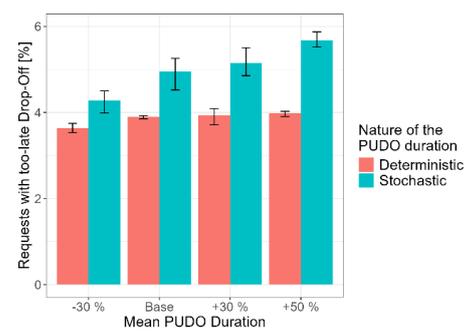


Figure 51: Late Drop-Off share with Heterogeneous PUDO duration

Figure 52: Effects of Predicting PUDO Duration for Each Request

We compared a subset of the deterministic scenarios (in the *Fully-informed deterministic* family, those with the real PUDO duration based on the empirical mean of the four Log-normal distributions) and those stochastic scenarios that consider the same empirical mean duration (e.g., the *Naive-informed stochastic* family). As in both cases the operator has the same level of information, we can isolate the effect of stochasticity.

Results, in Figure 46 and Figure 47, indicate no significant difference in the share of served requests between deterministic and stochastic scenarios (after controlling for the magnitude of the PUDO

duration). However, the situation is notably different for the LoS metrics (Figure 48 to Figure 51). The absolute drop-off time deviation and Late Drop-Off share are systematically worse for stochastic scenarios, with higher demand levels exacerbating the degradation of the LoS. As an example, the presence of stochasticity can increase the share of late drop-off requests by up to 1.7%.

Effects of Predicting PUDO Duration for Each Request

We evaluated the impact of using different prediction models for PUDO duration, focusing on the Fully-informed and Model-informed stochastic scenario families. Results in Figure 53 reveal only a marginal operational improvement trend with better prediction models (e.g., with a lower maximum relative error δ).

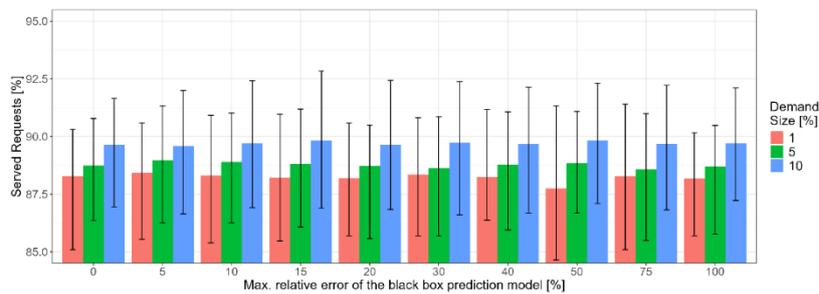


Figure 53: Operational impact of using different PUDO duration prediction models

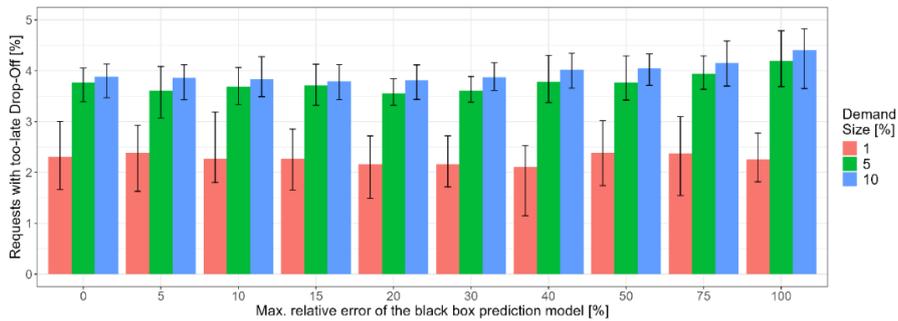


Figure 54: Impact on the Absolute Drop-Off time Deviation

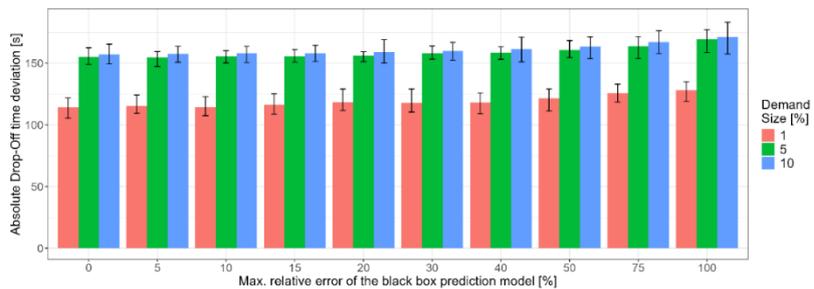


Figure 55: Impact on the Absolute Drop-Off time Deviation

Figure 56: LoS impacts of using different PUDO duration prediction models

We argue that the lack of a clear trend is, to a certain extent, an artifact of the PUDO duration generation process, the right-skewedness of log-normal distributions and the use of truncation. In addition, the proposed black box model in this study is unbiased, meaning that the chances of over- and under-estimating the PUDO duration are the same. Thus, when several PUDOs are

planned one after the other for a vehicle, they tend to compensate over time, mitigating the operational impacts from the operator’s perspective.

Results in Figure 54 and Figure 55 show that better prediction models improve service reliability, reducing absolute drop-off time deviations and Late Drop-Off shares. This is logical, as passengers spend only a short period in the vehicle and share the ride with a limited number of other requests. Consequently, temporal deviations caused by over- or under-predictions during the ride are less likely to be compensated.

Effects of Using Biased vs Informed Constant PUDO Duration Values

In this final subsection, we seek to understand whether, in the absence of a prediction model capable of predicting the duration of every individual PUDO process, it would be beneficial to use a naive-informed constant value for the fleet management tasks, rather than an arbitrary one, when the real PUDO duration is stochastic. To this end, we compare the results of the *Uninformed* and *Naive-informed stochastic* scenario families. To enrich the interpretation of the results, Figure 57, Figure 58, and Figure 59 also display the results for $\delta = 0\%$ and $\delta = 100\%$ from the *Fully-informed stochastic* families.

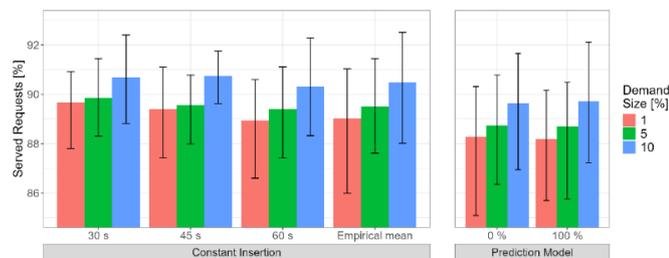


Figure 57: Operational impacts using different Constant Insertion and Prediction Models

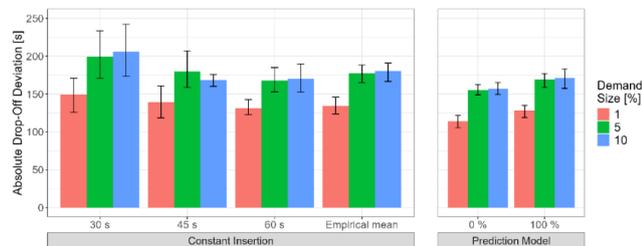


Figure 58: Impact on the Absolute Drop-Off time Deviation

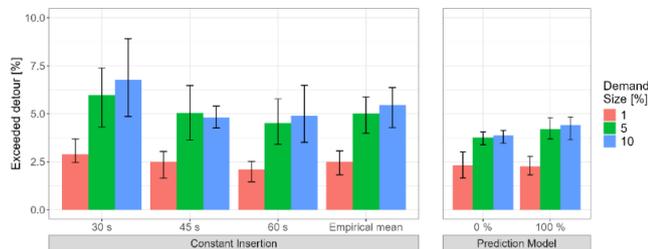


Figure 59: Impact on the Late Drop-Off share

Figure 60: LoS impacts using different Constant Insertion and Prediction Models

Regarding the share of served request, in Figure 57, we observe relatively minor differences between using 30 s, 40 s, or the Empirical Mean value for fleet management, and a lower performance for 60 s. In comparison, using the prediction models also leads to fewer served

requests across all demand size levels. This might suggest that ignoring the real duration and using a fixed 30 s value is the best approach for the operator. However, we argue that the results shown in Figure 57 are an artifact of how the simulation framework models MoD operator's decisions (particularly the assignment algorithm) and user behaviour (users cannot cancel even if the vehicle is delayed). Regarding the former, when the operator decides whether to assign a user request to a vehicle, it considers the feasibility of that combination (Alonso-Mora et al. 2017; Engelhardt et al. 2020). This feasibility depends on the necessary travel time and on the number (and duration) of the intermediate stops for that vehicle before arriving at the origin of the new request. Thus, if the duration, considered by the operator, not necessarily the actual duration, of the intermediate PUDO processes decreases, more combinations of request-vehicles would potentially be feasible, leading to a higher share of served requests. Nevertheless, decreasing the planned PUDO duration would likely cause delays for passengers, degrading the LoS, making the service less attractive, and inducing the cancellation of requests, although this last aspect is not captured by our framework. In contrast, when the operator uses a prediction model, it anticipates whether inserting that request into a vehicle's schedule will cause already-assigned requests to arrive too late at their destinations. It then considers those request-vehicle combinations unfeasible, reducing the number of served requests but improving the service quality for the accepted ones.

Figure 58 and Figure 59 support this argument, as the LoS is always better using the Empirical mean than using a 30 s constant value. However, as already mentioned, this did not affect the number of served requests in our study as we considered exogenous demand and did not allow for user cancellations even when the vehicles were significantly delayed. Interestingly, these figures also reveal that employing a prediction model, even one with $\delta = 100\%$, leads to a better LoS than using any constant insertion value.

Importantly, although the improvements in the Late Drop-Off share using the prediction models might not seem astonishing, for the 10% demand case the reduction around 1%, we believe that the impacts on the long-term acceptance and usage of the MoD service could be considerable in certain contexts. For example, if the MoD service operated as a feeder mode to scheduled public transport services, violating the latest drop-off constraint of a request might mean missing the connection and waiting for the next service (with its high associated disutility). Thus, even small reductions in the Late Drop-Off share could lead to an important improvement in the service's quality.

In summary, we can conclude that using a prediction model or, alternatively, considering a constant naïve-informed value, would lead to more reliable services for the passengers. We believe that this would also lead to a higher share of served requests, but this phenomenon is not realistically captured in our setup of the simulation framework.

Conclusions, Discussion, and relevance for the metaCCAZE project

In this study, we analyzed the impact of Pick-Up and Drop-Off (PUDO) duration on the operational performance and Level of Service (LoS) of a ride-pooling MoD service. We examined how the magnitude of the average PUDO duration, its heterogeneity across requests, and stochasticity impact the service, and explored the benefits of using informed PUDO duration values for fleet management.

The results indicate that even modest changes in the mean PUDO duration significantly affect service performance, also in the presence of stochasticity. For example, a 25-second increase in PUDO duration can reduce the share of served requests by over 3%, highlighting the sensitivity of the system to PUDO duration changes. We also found that PUDO duration stochasticity leads to substantial degradation of the LoS, particularly at higher demand levels. Finally, we noted that the LoS would be improved if operators used an informed PUDO duration value when managing their

fleet (either a specific one for each request based on a data-driven prediction model or, in its simplest form, a constant naive-informed value, instead of the arbitrary 30 s, 45 s, or 60 s typically used in research and practice). This latter approach is particularly interesting for real-life applications, as it would require a very low burden to implement (an operator would only need to calculate the average duration across all historical trips).

While the experimental design aimed to derive generalizable results, some conclusions may be specific to the fleet control algorithms used (e.g., regular reassignment of requests may compensate for schedule deviations caused by PUDO stochasticity). Additionally, the simulation framework does not fully capture the trade-off between LoS and the share of served requests. Further research is needed to develop more realistic approaches to handle stochasticity in both travel and PUDO times and to understand how the discrepancies between the “promised” (offered) and “actual” (provided) LoS impact users’ short-term and long-term behaviour and acceptance of MoD services.

From the perspective of the metaCCAIZE project, the findings of this study highlight the importance of planning MoD infrastructure to minimize both the magnitude and variability of PUDO durations. In this regard, Dynamic Curbside Management zones, whose availability can be checked in advance by approaching vehicles and, potentially, reserved, offer a safe and reliable solution for conducting PUDO operations. By reducing uncertainty and delays at the curbside, such infrastructure can enhance the operational efficiency for mobility service providers and improve the Level of Service (LoS) experienced by users.

7.1.5. Optimal Dynamic Curb-side Management for mobility-on-demand services

Introduction and Related Studies

Traditionally, most Mobility-on-Demand (MoD) services operate in a door-to-door fashion, transporting passengers from the closest possible point to their origin to the closest possible point to their destination. This approach has advantages for passengers, who minimize walking distances, and for operators, who only need to consider one combination of origin and destination during the fleet planning process, which is intrinsically high in computational demand.

However, recent studies have demonstrated the environmental and operational benefits of incorporating walking legs to/from Pick-Up Drop-Off (PUDO) locations. This means passengers are asked to walk short distances, a few hundred meters at most, between their origins/destinations and the PUDO locations. This can be seen as a hybrid between current MoD services and stop-based bus services, with the main differences being that the service is on-demand and requires no transfers (Engelhardt & Bogenberger, 2021; Fielbaum & Alonso-Mora, 2024; Zwick et al., 2021). The main demonstrated benefits of including walking legs in MoD services are: (1) avoiding unnecessary detours, thus decreasing the system’s vehicle-kilometers traveled, which results in reduced congestion, energy consumption, and greenhouse gas emissions (Engelhardt and Bogenberger 2021; Fielbaum et al. 2021), (2) reducing noise emissions (Zwick et al., 2021), (3) increasing the number of served requests (Engelhardt and Bogenberger 2021), and (4) lowering operational costs, which may result in lower prices for riders.

These benefits are enhanced when the MoD operator does not simply ask passengers to walk to/from the nearest PUDO area but instead considers multiple possible combinations of Pick-Up and Drop-Off zones within a reasonable walking distance. However, in practice, accounting for multiple potential PUDO locations per request introduces significant computational complexity for fleet control algorithms due to the expanded solution space (Wang et al. 2022). Establishing a network of PUDO locations, instead of operating door-to-door services, also opens the possibility

of providing infrastructure elements that improve the quality of the service, such as stop shelters, lighting, and weather protection. This can also make finding the pick-up location easier, a common source of delays in MoD operations. Moreover, municipalities could consider providing dedicated space for PUDO processes on the curbside (similar to existing taxi ranks or bus stops), thus reducing the impacts of these processes on surrounding traffic and improving the safety of passengers and other traffic participants.

However, this type of MoD operation also comes with some negative consequences. Primarily, users might experience a high disutility by walking these “small” segments, and the total travel time might slightly increase, as walking is slower than driving. In the context of metaCCAIZE, we seek to explore how, given the availability of curbside-occupancy sensors and communication technologies, PUDO processes could be accommodated in Dynamic-Curbside-Management (DCM) areas. This presents important benefits, as allowing different activities (logistics and service vehicles, but also PUDOs) in the same DCM zone would increase its utilization and reduce the total infrastructure needed for transport-related activities, which could be allocated to other uses (commercial, vegetation, bike infrastructure, recreation) (Valença et al. 2021).

Given the above explanation, a natural question arises: if a city were to roll out DCM zones at a large scale over the network, where should DCM zones be located? For the case of DCM zones used by MoD services, we name this task the *PUDO network design problem*.

Although there are some similarities to traditional public transportation design problems (e.g., the stop location problem or the transit network design problem), the on-demand nature, without predefined routes, schedules, or fleets, of MoD services makes it impossible to adapt existing methods. Moreover, the problem entails obvious trade-offs. On the one hand, providing many DCM locations will minimize the walking distance of passengers and reduce their travel times. On the other hand, it will require a higher use of public space, reduce the benefits of operating with walking legs, and increase the computational complexity for fleet control algorithms (as more DCM zones will fall within the reachable distance from origin and destination).

The introduced PUDO network design problem is closely connected with metaCCAIZE’s Munich **Dynamic Curbside Management Use Case**, in which a limited number of DCM zones are to be implemented in the inner-city area. Specifically, we aim to develop strategies that could guide the expansion of the DCM project beyond the city center after the project.

The metaCCAIZE team is currently working on two different approaches to tackle this problem. Their differences lie in both the methodology and objectives of the optimization process. Both are still under development and do not have final results. For this reason, in this deliverable, we only provide a short description of both approaches.

First approach

Previous studies have employed relatively simplistic approaches to define candidate stop locations, such as selecting all network intersections or placing stops at uniform distances (e.g., (Engelhardt & Bogenberger, 2021; Fielbaum et al., 2021)). These approaches overlook critical factors such as the convenience of stop locations, their potential contribution to congestion, and the presence of quasi-duplicate candidates located in close proximity or along the same street. Moreover, the resulting networks tend to be dense, offering many pick-up and drop-off options per request within a reasonable walking distance. This density renders it computationally infeasible for MoD operators to exhaustively evaluate all possible PUDO combinations within real-time fleet control algorithms. As a result, prior research has largely focused on developing heuristics that explore only a small subset of potential combinations, potentially missing more optimal solutions.

In this research, we propose a different approach: an offline framework for the strategic design of a PUDO location network that optimizes both the number and spatial distribution of candidate stops. The goal is to reduce the solution space to a curated set of high-potential locations, enabling exhaustive (or near-exhaustive) exploration per request within a reasonable computation time. The resulting network is less dense, supports faster operations, and preserves spatial coverage while enhancing trip-sharing potential.

Importantly, this framework also contributes to transport planning by identifying areas where MoD with walking access may be less effective, potentially guiding operators toward offering door-to-door service in those regions. We begin with a broad set of candidate locations (e.g., all intersections) and use metaCCAIZE's asset Fleetpy simulation tool to model MoD operations for large-scale demand samples. Inspired by the work of Fielbaum and Alonso-Mora (2024), each candidate location receives a *hierarchy score*, the number of vehicles flowing through it, which captures the likelihood that a vehicle would serve a request at that location without substantial detours. These scores are then spatially aggregated and smoothed to account for nearby candidate overlap, and thresholds are applied to distinguish zones suitable for walking access versus those that require door-to-door service. Finally, in walking-legs operation, we use an optimization algorithm to select a minimal set of high-scoring PUDO locations that maintain spatial accessibility across all requests.

While full simulation results are not yet available, we are currently working on the implementation and plan to present preliminary results at an upcoming research symposium. A thorough comparison with existing methods is also planned for future work.

Second approach

In this approach, we address the same overarching challenge, the design of a network of PUDO locations, but we also introduce a critical new dimension: the impact on traffic congestion caused by PUDO activities.

PUDO processes share characteristics with other traffic disruptions, such as public transportation stops, delivery vehicle halts, and double parking. These events have been shown to reduce network capacity and intensify congestion (Cao et al. 2016; Johari et al. 2020). As mobility-on-demand (MoD) services continue to grow, the movement of these vehicles and their associated PUDO activities are expected to significantly affect traffic flow throughout large portions of urban transportation networks.

In this work, the metaCCAIZE team is developing a methodology aimed at supporting city planners in making informed decisions regarding: 1. where and how much curb space should be dedicated for PUDO processes 2. where to prohibit double parking PUDO processes. These planning decisions correspond to the link-level decision variables of the proposed "*Congestion-aware Pick-Up and Drop-Off Network Design Problem*", and they influence the MOD and private vehicle routing. The objectives of the associated optimization problem are to minimize (O1) congestion on the streets (and thereby implicitly both the in-vehicle travel times of private vehicles and MOD users) and (O2) walking duration of the MOD users. Additionally, city planners will want to limit or even minimize (O3) the amount of curbside they need to dedicate to PUDOs.

Given the novelty and complexity of this problem, and the prohibitive computational costs of microscopic simulations, a new modelling approach is required. We are developing a modular framework composed of three components: an MOD ride-pooling model with walking legs, a static traffic assignment, and an upper-level optimization.

Both the MoD and traffic assignment models are being extended to account for the location and type of PUDO activity (e.g., curbside versus double parking), as well as regulatory constraints. The

overall multi-objective optimization problem is integer, non-linear, and high in complexity, containing non-trivial interactions of decision variables and MOD and private vehicle routing. Additionally, the objective function does not have a closed mathematical form, but it depends on the evaluation of the AMOD and Traffic Assignment modules, both requiring expensive computational simulations. Consequently, exploring exhaustively the whole space through brute force is computationally not feasible. As an alternative, we propose using a meta-heuristic with problem-specific and adaptive operators for large neighbourhood search, allowing the efficient exploration of the solution space

The metaCCAIZE team is currently developing this framework and plan to submit this work for review later this year.

7.1.6. Multimodal Waste Collection Logistics

This subtask focuses on vehicle routing optimization in multimodal urban logistics operations involving transshipment points, while accounting for time-dependent travel times and demand uncertainty.

Background

In large parts of the city center of Amsterdam, the Netherlands, residents leave their garbage bags on the street in front of their houses. Standard waste containers cannot be used in this area due to limited space. The waste is collected by heavy-duty vehicles twice a week. This is problematic, particularly because the bags often contain edible waste, which attracts rats and seagulls. In addition, heavy-duty vehicles add stress to the infrastructure, such as bridges and quay walls. Recognizing these challenges, the municipality has taken steps to limit the access of heavy vehicles to the city center, aiming to protect its historic infrastructure (“Stricter rules for heavy vehicles” 2024). Pourmohammad-Zia et al. (2024) summarize these challenges, emphasizing the importance of well-functioning city logistics networks. They point out that in cities such as Amsterdam, there is a potential to utilize the inland waterways to enhance city logistics, including waste collection.

The municipality aims to place as many underground waste containers as possible. However, in the city center, this is not possible everywhere due to limited space. Alternative waste collection approaches must be considered for households that cannot be covered by underground containers. One possible approach is on-demand waste collection in combination with a scheduled service that goes around the neighborhood and stops at designated collection points, where the residents can bring their waste according to the schedule. Cargo bikes and light electric vehicles are used in combination with barges that take the waste outside of the city center via the waterways. This approach is currently being tested in a pilot in De 9 Straatjes neighbourhood.

Integration of the inland waterways naturally creates a two-echelon network. Shifting a part of the traffic to the canals and deploying light electric modes of transport seems to be a promising way to relieve the traffic in the narrow streets and reduce the strain on quay walls and bridges in the historic city center. However, given the limited space in the city center, the transshipment points have no storage capacity. In other words, the waste can be transferred only if there is a barge at the transshipment point. This requires seamless interoperability between the two echelons.

Model

Let us first consider the key characteristics of such a system and the implications for modeling. The system integrates a road network with low-capacity vehicles and bikes and waterways with medium- and large-capacity barges. This can be modeled as a multi-echelon (multi-level) routing problem. Given the limited capacity of light vehicles and cargo bikes, it is essential to consider multiple trips between the transshipment points and households. Utilizing both light electric

vehicles and electric cargo bikes implies that the model needs to handle a heterogeneous fleet. Given the limited space in the city center, the road-based vehicles can only be unloaded when there is a barge at the corresponding transshipment point. Thus, spatiotemporal synchronization needs to be considered. Furthermore, hard time windows must be considered in the context of waste collection by appointment. Finally, we assume that the battery of the electric vans can last the whole shift, and the battery of the electric cargo bikes can be easily swapped at the transshipment point.

The first step is to formulate an arc-based mixed integer linear programming model for the two-echelon vehicle routing problem with time-dependent travel times, multiple trips, heterogeneous fleet, time windows, and satellite synchronization with the objective to minimize total travel time. A piecewise-linear travel time function will be adopted to model the time-dependent travel times. Subsequently, a route-based formulation will be proposed, and a column generation algorithm with a branch-and-price method will be designed to solve the problem to optimality.

The arc-based formulation of the two-echelon CVRP proposed by Jepsen et al. (2013) was adopted and extended to account for time windows and time-dependent travel times. The current formulation assumes homogeneous fleets, single trips, and does not consider satellite synchronization.

The two-echelon VRP with time windows and time-dependent travel times (2E-VRP-TW-TD TT) is formulated on a graph $G = (V, A)$, where the set of nodes $V = V_o \cup V_s \cup V_c$ consists of a central depot and processing facility $V_o = 0$, satellites V_s and customers V_c . The set of arcs A contains two subsets, $A = A_1 \cup A_2$, where $A_1 = A(V_o \cup V_s)$ and $A_2 = A(V_s \cup V_c)$. The size of the first and second echelon fleets is given as K_1 and K_2 , respectively. C_1 and C_2 are vehicle capacities of the respective fleets. Each customer $i \in V_c$ has demand $D_i > 0$ and must be served within the time interval $[e_i, l_i]$. Service time, denoted s , is assumed to be the same for all customers. θ_h denotes the start time of time period h , and t_{ij}^h denotes travel time from i to j in time period h .

Binary decision variable x_{ij}^k is 1 if vehicle k traverses arc $(i, j) \in A_1$, and 0 otherwise. $u_s^k \in Z^+$ is position of satellite s in route of vehicle $k \in K_1$. $w_s^k \in R^+$ is amount of waste transported from satellite s to the processing facility by vehicle $k \in K_1$. Binary variable y_{ij}^k is 1 if vehicle $k \in K_2$ from satellite s traverses arc $(i, j) \in A_2$, and 0 otherwise. Variable $f_{ij}^s \in R^+$ denotes load on arc $(i, j) \in A_2$ to satellite s . $a_i^k \in R^+$ denotes arrival time of vehicle $k \in K_2$ at customer i . $\delta_{ij}^{s,h}$ is 1 if vehicle dispatched from satellite s uses $(i, j) \in A_2$ in time interval h , and 0 otherwise. The 2E-VRP-TW-TD TT is formulated as follows:

Objective function

$$\min \sum_{(i,j) \in A_1} \sum_{k \in K_1} t_{ij}^1 \cdot x_{ij}^k + \sum_{(i,j) \in A_2} \sum_{s \in S} \sum_{h \in H} t_{ij}^{2h} \cdot \delta_{ij}^{s,h} \quad (1)$$

Constraints

$$\sum_{(i,j) \in A_1} x_{ij}^k = \sum_{(j,i) \in A_1} x_{ji}^k \quad \forall i \in V_0 \cup V_s, \forall k \in K_1 \quad (2)$$

$$\sum_{(i,j) \in A_1} x_{ij}^k \leq 1 \quad \forall i \in V_0 \cup V_s, \forall k \in K_1 \quad (3)$$

$$u_i^k - u_j^k + |S| x_{ij}^k \leq |S| - 1 \quad \forall (i,j) \in V_s, i \neq j, \forall k \in K_1 \quad (4)$$

$$w_s^k \leq Q_1 \cdot \sum_{(i,s) \in A_1} x_{is}^k \quad \forall s \in S, \forall k \in K_1 \quad (5)$$

$$\sum_{s \in S} w_s^k \leq Q_1 \quad \forall k \in K_1 \quad (6)$$

$$\sum_{k \in K_1} w_s^k = \sum_{(s,j) \in A_2} f_{sj}^s \quad \forall s \in S \quad (7)$$

$$\sum_{(i,j) \in A_2} y_{ij}^s = \sum_{(j,i) \in A_2} y_{ji}^s \quad \forall i \in V_s \cup V_c, \forall s \in S \quad (8)$$

$$\sum_{(s,j) \in A_2} y_{sj}^s \leq K_2 \quad \forall s \in S \quad (9)$$

$$\sum_{s \in S} \sum_{(i,j) \in A_2} y_{ij}^s = 1 \quad \forall j \in V_c \quad (10)$$

$$\sum_{(j,i) \in A_2} f_{ji}^s - \sum_{(i,j) \in A_2} f_{ij}^s = D_i \quad \forall i \in V_c \quad (11)$$

$$f_{ij}^s \leq Q_2 \cdot y_{ij}^s \quad \forall (i,j) \in A_2, \forall s \in S \quad (12)$$

The objective function (1) minimizes total travel time. Constraints (2) - (6) are related to the first echelon. Constraints (2) ensure flow conservation. Constraints (3) impose that each vehicle visits each satellite at most once. Constraints (4) prevent subtours. Constraints (5) and (6) ensure that vehicle capacity is not exceeded. The first and second echelons are linked by constraints (7), which ensure flow balance at satellite. Constraints (8) - (12) are related to the second echelon. Constraints (8) ensure flow conservation. Constraints (9) limit the number of second echelon vehicles at each satellite. Constraints (10) ensure that each customer is visited exactly once. Constraints (11) ensure that demand is met. Constraints (12) limit vehicle capacity. Constraints (13) - (17) define decision variable domains.

Time windows extension

$$e_i \leq a_i^s \leq l_i \quad \forall i \in V_c, \forall s \in S \quad (18)$$

$$a_j^s \geq a_i^s + s + t_{ij}^2 - M(1 - y_{ij}^s) \quad \forall i, j \in V_c, s \in V_s, i \neq j \quad (19)$$

To account for time windows, constraints (18) - (20) are added to the model. Constraints (18) ensure that arrival time at customer i is within the allowed time interval. Constraints (19) define arrival at customer i as the sum of arrival at customer $(i - 1)$, service time, and travel time on arc $(i - 1, i)$. (20) are non-negativity constraints.

Time-dependent travel times extension

$$a_j^s \geq a_i^s + s + \sum_{h \in H} t_{ij}^h \cdot \delta_{ij}^{s,h} - M(1 - y_{ij}^s) \quad \forall i \in V_c, \forall j \in V_c, \forall s \in S, i \neq j \quad (21)$$

$$\sum_{h \in H} \delta_{ij}^{s,h} = y_{ij}^s \quad \forall (i, j) \in A_2, \forall s \in S \quad (22)$$

$$\theta_h \cdot \delta_{ij}^{s,h} \leq a_i^s + s \leq \theta_{h+1} \cdot \delta_{ij}^{s,h} + M(1 - \delta_{ij}^{s,h}) \quad \forall h \in H, \forall (i, j) \in A_2, \forall s \in S \quad (23)$$

To account for time-dependent travel times, constraints (22) - (24) are added to the model, and constraints (19) are replaced by (21). Constraints (21) define arrival at customer i as the sum of arrival at customer $(i - 1)$, service time, and travel time on arc $(i - 1, i)$ at a given time interval. Constraints (22) ensure that exactly one time period is selected for each trip between i and j . If time interval h is active, constraints (23) enforce that departure time from customer i is within the time interval. Constraints (24) define variable domains.

The model is implemented in Python using gurobipy to access Gurobi Optimizer and validated on a small test instance. The next step is to implement the spatiotemporal satellite synchronization, multiple trips, and heterogeneous fleets.

7.2. Subtask 2.6.2 Low emission-based traffic management

In subtask 2.6.2, Vedecom develops a traffic management model that integrates the communication technologies introduced in subtask 2.5.2. To this end, Vedecom has developed a deep learning-based predictive model that integrates data from a wide range of sources, including drones, V2X communication, cameras, parking facilities, charging stations, sensors, public transport systems, traffic signals, and passenger mobile applications. Leveraging these heterogeneous data streams, the model employs a transformer-based architecture capable of capturing complex temporal and spatial patterns to generate accurate traffic predictions.

As part of this subtask, MobilLysis employs aerial data collected via drones to support low-emission, multimodal traffic management strategies. The objective is to enable in-depth analysis of modal interactions and traffic bottlenecks, using video-derived insights to inform congestion mitigation, sustainable route planning, and intersection design improvements aligned with the project's environmental goals. As drone-based data collection expands to new geographic areas, MobilLysis anticipates additional opportunities to refine its traffic management tools. The diversity of urban contexts will reveal varying patterns of congestion, inefficiencies, and multimodal conflicts, thereby enhancing the adaptability of the proposed solutions.

In parallel, under subtask 2.6.2, TUD has been working on the development of Tradable Mobility Credits (TMC) algorithms for optimally allocating credits to distinct population segments within an urban environment. The main objective is to promote the adoption of sustainable transportation modes. The methodology involves first constructing population profiles that encapsulate diverse travel needs and subsequently assigning mobility credits based on these profiles to maximize modal shift while adhering to individual travel constraints.

7.2.1. Predictive modeling with multi-source data

The methodology proposed by Vedecom is graph-based and designed to explicitly capture the spatiotemporal relationships among the various entities that contribute to a smart mobility hub. It integrates drone data provided by MobilLysis, sensor data sourced from Floware, and additional publicly available datasets. Following the design of the model, evaluation metrics are identified and examined to assess the accuracy and overall quality of the predictions. These metrics will not only

quantify the model's predictive performance but also offer actionable insights for further model refinement and enhancement of its capabilities.

This section is structured into two main subsections. The first outlines the overall traffic management methodology, including the graph-based data representation and transformer-based architecture. The second focuses on drone data collection and analysis, emphasizing its role in supporting low-emission, multimodal traffic management strategies.

Literature review

Traffic forecasting is a critical component of Intelligent Transportation Systems (ITS), significantly enhanced by deep learning, particularly Graph Neural Networks (GNNs), which model complex spatiotemporal dependencies. GNNs represent non-Euclidean traffic data through graph structures, in which nodes symbolize traffic sensors or road segments, and edges denote spatial relationships. Jiang and Luo (2022) emphasized GNNs' superiority over conventional models, especially when using multi-graph structures to capture functional and spatial relationships across transport networks. Recent models like MHGNet (Wu et al. 2025), D-MGDCN-CLSTM (Zhang et al. 2025), and DGCN-TRL (Chen et al. 2025) address heterogeneous data, partial data loss, and temporal patterns using advanced embedding, wavelet decomposition, and transformer techniques, achieving strong predictive performance.

Recent research underscores the importance of integrating heterogeneous data to enhance model robustness. Zhang et al. (2025) leveraged multi-graph attention networks (GMAN) to incorporate spatial correlations (distance, flow patterns) and temporal dynamics, significantly reducing error propagation in multi-step forecasting. Additionally, recent models increasingly incorporate real-world events – such as accidents and congestion – using multi-graph configurations. For instance, T-MGCN (Lv et al. 2020) effectively combines non-Euclidean spatial correlations with semantic information on road features. Similarly, STFGNN model (Li et al. 2021) processes multiple spatiotemporal graphs across varying periods, capturing latent dependencies among speed, direction, and flow parameters.

Machine learning models that fuse unmanned aerial vehicles (UAVs) and loop detector data via LSTM and GNNs, such as HiMSNet (Xiong et al. 2025), offer reliable congestion forecasts at both segment and regional levels. For risk assessment, drones support trajectory-based surrogate safety metrics like the Emergency Index (EI) (Chen et al. 2025). Despite these advances, challenges remain. Most studies rely solely on UAV data, limiting robustness, and multimodal data fusion is still underexplored (Telikani et al. 2024). Furthermore, models like GSAN (Ye et al. 2021) decouple spatial and temporal interactions, which may hinder the capture of joint dynamics essential for autonomous driving applications.

Building on the demonstrated value of drone-supported analysis in traffic forecasting, congestion prediction, and risk assessment, this study integrates drone-collected trajectory data with complementary sources, including V2X communication data, fixed roadway sensors (e.g., inductive loops, radar), and public transport feeds. This multi-source integration addresses the identified limitations of single-stream models by combining the spatial coverage of drones, the temporal precision of ground-based sensors, the real-time connectivity of V2X, and the operational context of transit data – forming a robust framework for urban traffic state forecasting.

Methodology

Drone's data is used in the predictive approach that leverages spatiotemporal graphs, as depicted in Figure 61, that model traffic interactions across both space and time. By representing vehicles

and their dynamic relationships as nodes and edges within a graph structure, the model effectively captures complex interaction patterns. These spatiotemporal dependencies are further processed using a transformer encoder, which enhances the model's ability to learn long-range temporal relationships and make accurate, robust predictions for downstream traffic forecasting tasks.

At each time step t , the traffic scene is modeled as a spatio-temporal graph $G_t = (V_t, E_t)$, where V_t denotes the set of vehicles and E_t the set of spatially-directed edges reflecting inter-vehicle relationships (e.g., distance or heading). To preserve temporal continuity, edges $e_{i,j}^{(t,t+1)}$ connect the same vehicle across successive frames.

The full temporal graph over T frames is $G_{st}^T = (V, E)$, where $V = \cup_t V_t$ and $E = \cup_t E_t \cup E_{temporal}$. Each node $v_i^{(t)} \in V$ is associated with a feature vector $v_i^{(t)} \in R^d$ encoding kinematic attributes (e.g., position, velocity). These features are projected into a hidden space via an embedding function $h_i^{(t)} = \varphi(x_i^{(t)})$, where function $\varphi(\cdot)$ is typically an MLP or linear layer.

The transformer model encoder employs multi-head spatio-temporal self-attention to refine these embeddings. For a given node i , attention is computed over its neighbors $j \in N(i)$ across time windows $t' \in t - 1, t, t + 1$ using:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Where $Q_i = W_q h_i^{(t)}$, $K_j = W_k h_j^{(t')}$, and $V_j = W_v h_j^{(t')}$. Multiple attention heads are concatenated and projected:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (2)$$

$$head_k = Attention(QW_k^Q, KW_k^K, VW_k^V) \quad (3)$$

After L transformer layers, the final node representations are denoted as:

$$z_i^{(t)} = TransformerLayer_L \circ \dots \circ TransformerLayer_1(h_i^{(t)}) \quad (4)$$

and all outputs are concatenated as $Z = [z_1^{(1)}, \dots, z_N^{(T)}] \in R^{NT \times d}$.

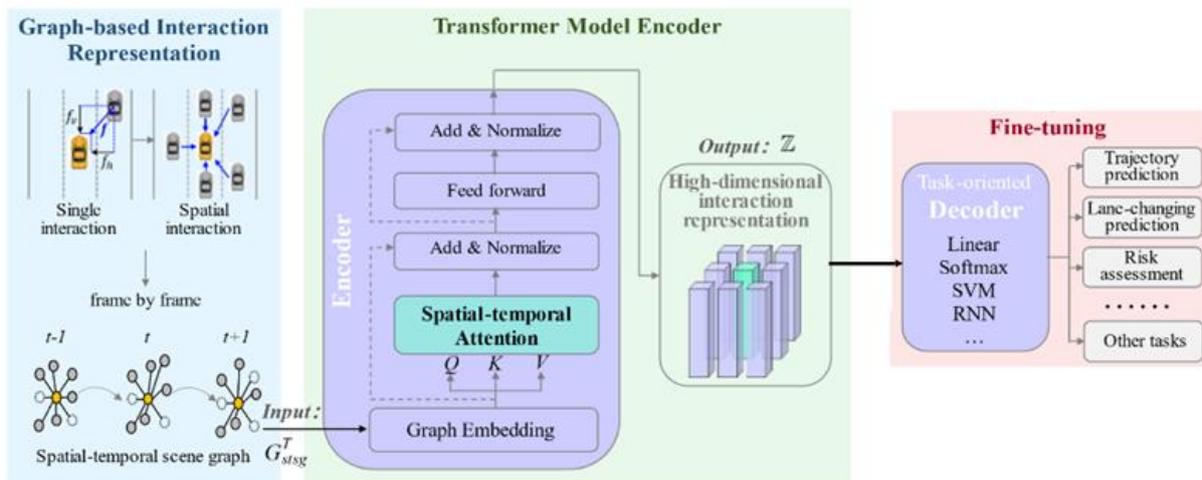


Figure 61: Overview of predictive GNN-based model

The Fine-Tuning and Optimization step involves adjusting key hyperparameters such as learning rate, batch size, and the number of training epochs to improve model performance. These parameters are iteratively refined using methods like cross-validation to ensure the model generalizes well across different data samples and avoids overfitting. During this stage, the high-dimensional representation Z , obtained from the earlier modeling process, is passed into a task-specific decoder. This decoder is then fine-tuned for targeted applications, allowing the model to adapt its learned representations to specific objectives, such as traffic prediction or anomaly detection, with greater accuracy and reliability. For trajectory prediction, a regressor $\hat{y}_i^{(t+\Delta)} = \Psi(z_i^{(t)})$ is applied. For classification tasks like lane-change detection, a *softmax* classifier is used: $\hat{y}_i = \text{softmax}(W_c z_i + b_c)$, while regression tasks (e.g., risk assessment) employ linear mapping: $\hat{y}_i = W_r z_i + b_r$. This unified framework enables flexible adaptation across a variety of traffic forecasting and decision-support tasks.

Through this pipeline, the Treatment Layer converts voluminous and unstructured data into high-resolution, spatiotemporal intelligence used to support strategic and operational decision-making.

Deep learning evaluation methods

Deep learning models for traffic prediction are commonly evaluated using statistical error metrics that quantify the difference between predicted and actual values. Among these, the most frequently adopted in the literature are Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). These metrics are favored for their interpretability and effectiveness in capturing different aspects of prediction accuracy. Importantly, they are generally negatively correlated with the size of the training dataset; as more training data becomes available, the prediction error typically decreases, enhancing the model's generalization capabilities. The formulas and roles of these metrics are outlined below:

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

MAE measures the average magnitude of errors in a set of predictions, without considering their direction. It is widely used for its simplicity and clear interpretability in real-world units.

Mean Square Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (6)$$

MSE penalizes larger errors more heavily due to squaring, making it sensitive to outliers. It is useful for highlighting significant deviations between predicted and actual values.

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7)$$

RMSE is the square root of MSE and provides error metrics in the same units as the target variable. It is often preferred when large errors are particularly undesirable.

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (8)$$

MAPE expresses prediction accuracy as a percentage, making it useful for comparing performance across datasets or models. However, it can be unstable when actual values are close to zero.

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (9)$$

R^2 indicates the proportion of variance in the target variable explained by the model. A higher R^2 suggests a better fit to the data, making it a key metric in regression analysis.

Here, \hat{y}_i and y_i denote the predicted and actual values, \bar{y} is the mean of the actual values, and n represents the number of samples. While these metrics provide a solid foundation for model evaluation, it is also essential to consider the potential for overfitting and underfitting. Traditionally, increasing model complexity or training data was believed to reduce generalization performance. However, recent research (Belkin et al. 2019) has shown that deep learning models can still generalize well even when overfitted, challenging long-held assumptions and calling for further investigation into the theoretical underpinnings of deep model behavior.

Drone-supported analysis

MobiLysis began with a thorough evaluation of the three Use Cases where the technology will be applied – Paris, Limassol, and Krakow – focusing on drone regulations, local data collection requirements, and the potential for drone flights to augment or replace other sensing methods. This preparatory phase was essential in ensuring that the drone campaigns of MobiLysis would add unique value and operate within regulatory boundaries. In Limassol, MobiLysis coordinated closely with the Municipality and the MaaS Lab to plan and execute a successful drone campaign in March 2025. The operation targeted high-traffic roundabouts and intersections during peak periods, aiming to capture detailed interactions and identify potential safety concerns around potential pick-up/drop-off points.



Figure 62: Limassol Speed Loop

In France, a preliminary flight was conducted in Yvelines (April 2025) to assess local drone operations feasibility and begin benchmarking the current vision algorithms for the UC needs. This pilot supports future data collection related to the deployment of autonomous vehicles, demanding high-fidelity trajectory data and interaction analysis.



Figure 63: Paris Speed Loop

In parallel, the Computer Vision team of MobilLysis has been improving tracking performance under real-world conditions. They built upon the earlier StrongSORT-based work and tailored it for urban environments characterized by high density, frequent occlusions, and multiple interacting agents. The resulting improvements have strengthened the ability to extract continuous trajectories across a variety of mobility modes.

7.2.2. TMC (Tradable Mobility Credits) assignment method for multiple population segments

Background

Existing research on Tradable Mobility Credit (TMC) schemes predominantly considers a uniform credit allocation across all travellers, despite significant differences in mobility needs influenced by factors such as commuting distance, caregiving responsibilities, disabilities, and age. Therefore, uniform allocation may disproportionately penalize certain traveller groups. Consequently, a more equitable credit distribution should incorporate these socio-demographic and mobility needs profiles. As highlighted in (Chen et al. 2023; Gao et al. 2016; Provoost et al. 2023), personalized credit allocation represents a critical direction for future research. A comprehensive review by Hamm et al. (2025) categorizes various allocation strategies, including origin-destination (OD)-specific, work-related, income-based, household size-based, and accessibility-based approaches. Their survey results show strong public support, about 70%, for non-uniform allocation, with respondents citing mobility impairments, caregiving duties, and limited transport access as key considerations.

To implement personalized credit allocation, a first step involves using travel data to create used profiles by combining travel behavior with individual characteristics. These profiles estimate both mobility needs and the potential for behavioral change. Based on that, we want to take into account the nuances of the identified profiles. For example, travellers who travel by car but have good sustainable transportation options are candidates for changing behaviour but the travellers who, because of lack of options or other restrictions, require to use the car more shouldn't be disproportionately affected by the scheme based on their circumstances.

One of the most effective techniques for analyzing and segmenting user profiles is clustering, a data-driven approach that groups individuals with similar attributes based on predefined criteria. Clustering methods have been widely employed in transportation studies to identify distinct traveller segments, enabling more targeted and efficient interventions.

Methodology

To create user profiles, the plan is to apply profile segmentation techniques to a dataset that represents a population that would be the users of the TMC scheme living and working in a city, in this case, Amsterdam. This step requires extracting and processing features, such as sociodemographic ones. We're including features that provide information on income (low, medium, high), their occupation (working, studying, retired, unemployed, other), if people have a disability, and if people have children below the age of 12.

In addition, TUD also proposes taking into account the travel needs and activities in the profiles. In this case, information on trips each traveller takes, such as trip purpose (e.g., commuting or leisure), the main mode of transport, day type (weekday or weekend), frequency of mode usage, trip origin and destination, and travel duration can be leveraged to extract travel needs features that showcase travel patterns related to commuting and other daily activities. Additional indicators, such as the annual number of kilometers travelled per person, provide absolute values of individual travel. The final features should aim to capture the travel needs and behaviour that can then be used to distinguish travellers for credit allocation. These can be differences in commuting distance and how many days people require to be at the office, trips related to taking children to school, and different transportation modes used. The inclusion of travel habits according to transportation mode is done so that the different profiles can also be separated into travellers who have more opportunities to change behaviour, such as those who take the car, and those who are already taking the bike and public transport.

To capture accessibility, the TUD team proposes incorporating travel times between home and work across different transportation modes. Additionally, given that sociodemographic characteristics can influence mode choice, the plan is to incorporate these factors into the analysis as well. To account for both of these factors, the plan is to use a mode choice model with travel information and sociodemographics of the traveller that estimates the probability of selecting a car, public transportation, or bicycle for a given trip. By comparing the predicted probabilities across modes, it becomes possible to assess the potential for a mode shift toward more sustainable options.

After the feature extraction process, clustering methods will be applied. The most commonly used approaches for profile segmentation in the literature include k-means, Latent Class Analysis (LCA), and hierarchical clustering, each having associated advantages and limitations.

For this use case, the number of clusters is not yet known, and the dataset is mixed with both categorical, ordinal (such as income) and numeric features. Taking this into account, one promising approach is hierarchical clustering with a mixed data similarity measure. This method doesn't require transforming the features to numerical/categorical, doesn't require an initial k number, is flexible to different cluster shapes depending on the linkage method, and provides a dendrogram that can be used for further analysis. LCA is another option for the use case, by transforming the numerical variables into categorical variables. This might be a good option given that LCA has been used to segment profiles based on travel behaviour.

To perform hierarchical clustering on mixed data types, a similarity measure that supports both numerical and categorical variables is needed. Gower Distance, introduced by Gower, 1971, is well-suited for this purpose. It accommodates numerical, and categorical data, normalizes the output,

handles missing values, and allows for assigning different weights to features. This flexibility enables testing various weighting schemes and possibly fine-tune the similarity measure.

After applying the different clustering algorithms, the results should be evaluated to understand which method works best and if the methodology needs to be fine-tuned. The various evaluation criteria present in the literature will be applied to assess the quality of the resulting clusters, such as the Silhouette score, Dunn's index, chi-squared test, BIC and AIC. This quantitative evaluation is a first step, making it easier to compare algorithms by measuring how well-separated and cohesive the clusters are. In addition, interpretability is crucial in this context, as the ultimate goal of clustering is to segment the population for the allocation of credits. Therefore, a qualitative evaluation of the identified profiles, by examining their relevance within the context of this use case, will support the assessment of which clustering method works best. Given the social acceptability aspect of the policy, if the population is segmented into too many clusters, the credit allocation system will look more complex, and users will have a harder time understanding it. Therefore, the resulting number of clusters is something to take into account when evaluating the results.

Initial Results and Next Steps

The data used is from the Netherlands Mobility Panel (MPN) 2022 (Hoogendoorn-Lanser et al. 2015). The MPN is a household panel that encapsulates the travel behaviour of Dutch individuals and households, including a travel diary to collect trip data and personal questionnaires with sociodemographic information of the survey participants. The MPN contains both travel-based data and sociodemographic data for the Netherlands, therefore, it provides the information required for the use case of Amsterdam. The dataset requires pre-processing steps, including filtering to the Amsterdam area and the removal of incomplete answers to the survey. Income levels were then grouped into three categories: low, medium, and high. Occupation types were simplified into four groups: employed, student, retired, and other. Additionally, the number of children was converted into a binary feature indicating whether a person had children under the age of 12. After this processing, the current sample includes 461 data points, each representing an individual traveller.

Two travel-related features were also derived. The first was the weekly commuting distance, calculated using the distance between home and work and the number of days a person worked at the office. The second feature was the total distance travelled over the three recorded days. For this metric, outliers were identified and removed.

The first algorithm applied as a baseline was Agglomerative Clustering, a type of hierarchical clustering. The Gower distance was used as the similarity measure, as it is well-suited for mixed-type datasets and allows for flexible feature weighting. In this analysis, all features were given equal weight. Initially, the occupation feature was included in the clustering process. However, after removing it, the clustering results showed a higher silhouette score and better alignment with the intended use case. As a result, it was decided not to include it.

Using the dendrogram generated with average linkage distance, various threshold values were tested to determine which produced the highest silhouette score. For this dataset and metric, the optimal result suggested 15 clusters, with a silhouette score of 0.718. However, upon checking the results, it was found that 8 of the 15 clusters contained only 1 to 4 individuals, which is not practical for this use case. Additionally, defining 15 distinct profiles would likely be too detailed and difficult to communicate to the public, especially when considering the importance of social acceptability in policymaking.

The algorithm was applied using 6 clusters instead of 15. This resulted in a silhouette score of 0.539. However, 2 of the 6 clusters contained only a single individual. Using average linkage distance, the algorithm tended to isolate outliers, leading to highly unbalanced clusters (the largest cluster contained 409 out of the 461 data points). The linkage method was changed to complete, which calculates the distance between clusters based on the farthest pair of points, rather than the average distance between all pairs. This adjustment improved the silhouette score to 0.689 and resulted in more balanced cluster sizes. The initial results taken from the Agglomerative Clustering with complete linkage are in the Table 5.

Table 5: Initial Results

	CLUSTER	1	2	3	4	5	6
Indicators	Size	97	241	9	29	4	81
Commuting kms per week	Mean	17.8	9.28	13.53	22.64	7.16	3.49
Total kms in 3 days	Mean	127.37	103.59	85.72	105.2	20.08	59.75
Income	Mean(*)	2	1	0.44	1.41	0	0
Disability	Yes	0	0	9	0	1	0
	No	97	241	0	29	3	81
Kids	Yes	0	0	0	29	4	0
	No	97	241	9	0	0	81

(*) Income was mapped from Low-0, Medium-1, High-2

The table reveals that the first cluster represents high-income individuals without children, while the second cluster consists of people with average income, also without children. The third cluster includes individuals with disabilities and low to medium income, who do not have children. The fourth cluster represents individuals with children and either average or high income. The fifth cluster includes people with children and low income, as well as one individual with a disability. Finally, the sixth cluster represents low-income individuals without children.

The algorithm separated the sample well in terms of disability, having kids and income, however when checking the mean for the travel features, it gets harder to interpret the results. Thus, an analysis for these features was done in Tables 6 and 7.

Table 6: Commuting kms per week Analysis

CLUSTER	COUNT	MEAN	STD	MIN	25%	50%	75%	MAX
1	97	17.8	34.91	0	1	5	19	202
2	241	9.28	20.05	0	0	1	7	146
3	9	13.53	17.27	0	1	2.16	19	49.09
4	29	22.64	35.28	0	2	8.42	23.61	127
5	4	7.16	13.66	0	0	0.5	7.66	27.63
6	81	3.49	13.45	0	0	0	0	103

Table 7: Total kms in 3 days Analysis

CLUSTER	COUNT	MEAN	STD	MIN	25%	50%	75%	MAX
1	97	127.37	160.1	0	16.6	78.35	155.7	840
2	241	103.59	170.7	0	10.3	34.98	118.2	937.3
3	9	85.72	97.72	3	22	30.7	190.3	228.2
4	29	105.2	111.6	0	31.96	63.5	135.6	392
5	4	20.08	28.57	0	4.5	9	24.58	62.3
6	81	59.75	108.9	0	2.5	22.1	61.1	724.64

From the tables, the algorithm hasn't managed to successfully separate the clusters taking into account the travel features as well. Within each cluster, there is a wide range of travel needs, making it challenging to allocate credits based on these clusters.

Therefore, to improve results, it's necessary to study how to make the travel needs within each cluster similar. Several steps can be taken for this. First, a deeper exploration of travel needs and behaviours is required to identify potential new features, such as the frequency of transport modes and the kilometers travelled for different activities (e.g., shopping, leisure). Regarding the accessibility feature, there is additional data that will be shared in the future that includes home and work locations, as well as commuting times using various transportation modes.

The current sample size is relatively small, so it may be beneficial to include travellers from previous travel diaries to expand the dataset. Given the number of features and the potential for high-dimensionality, applying techniques like principal component analysis (PCA) and multiple correspondence analysis (MCA) could help, especially since some travel features may be correlated.

Agglomerative clustering was chosen as the first approach given that it is a hierarchical algorithm. In addition, it's planned to apply LCA and understand which approach works best for this use case. Another algorithm worth exploring is HDBSCAN, as proposed by McInnes et al. (2017). HDBSCAN is an extension of the density-based DBSCAN algorithm that builds a hierarchy of clusters, allowing for clusters of varying densities and more robustness in parameter selection. However, HDBSCAN tends to remove outliers, which is not ideal for this use case. A potential solution is to run HDBSCAN first, identify which travellers are considered outliers, and then evaluate whether it makes sense to reassign these outliers to the nearest cluster.

7.3. Subtask 2.6.3 Supply-demand matching platform for on-demand shared zero-emission services

Under subtask 2.6.3, the main goal of MaaS Lab is to deliver a platform that consists of three components: a B2C app interface to allow users to indicate in advance when they want to travel; a B2B app interface for operators and drivers of the on-demand shared e-service, that will allow them to accept/reject demand requests, and also recommend routes to optimise travel and waiting times, as well as electricity/fuel consumption; and a backend interface that will process the data (utilising also the AI models from Task 2.2), allowing for optimal recommendations to operators and users. For the user app interface, development has been completed on the following modules: **Planning, Infrastructure, Security, Bookings, Payments, Schedules, Calendar, Student Monitoring for Parents, and Student Location Tracking.**

The app interface for operators and drivers comprises of a Fleet Operator web platform and a driver app interface. Work has been completed on the following modules for the operator and driver app interface: **Planning, Infrastructure, Security, Driver Management, Vehicle Management, Driver Navigation, Vehicle Location Tracking, and Routing.**

Work on the following modules for the app interface for operators and drivers is heavily under development:

- Vehicle-Driver Assignment
- Scheduling
- Reporting
- Analytics

Sample images from the user app, as well as the operator and driver interfaces, can be found in Figures 64 and 65 respectively.

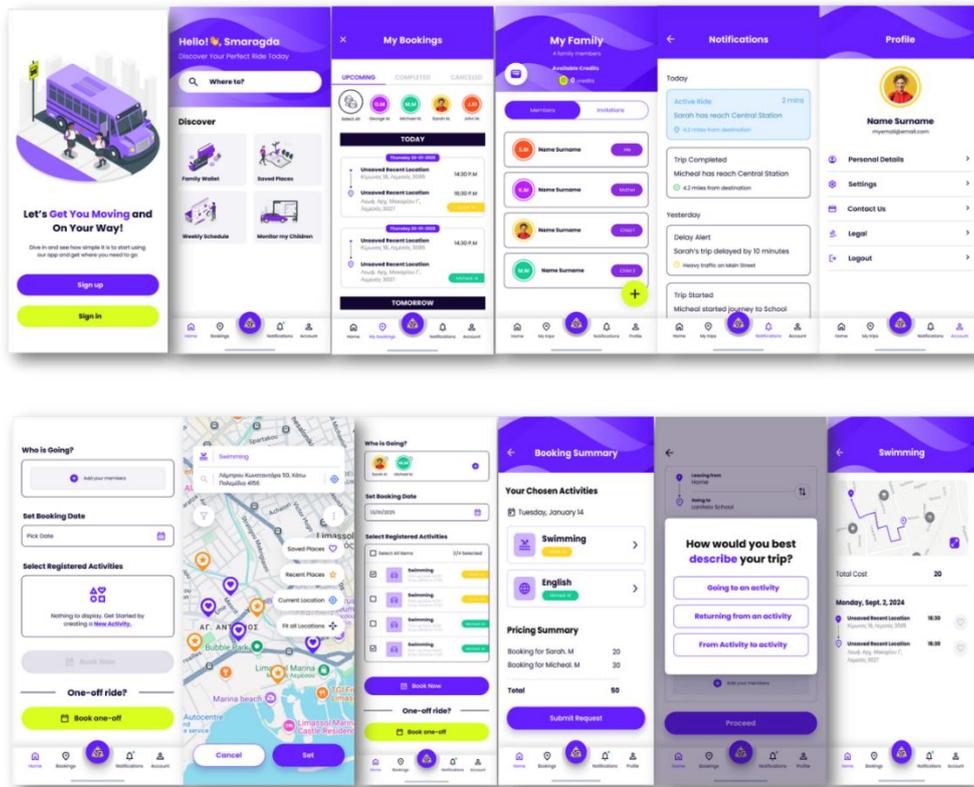
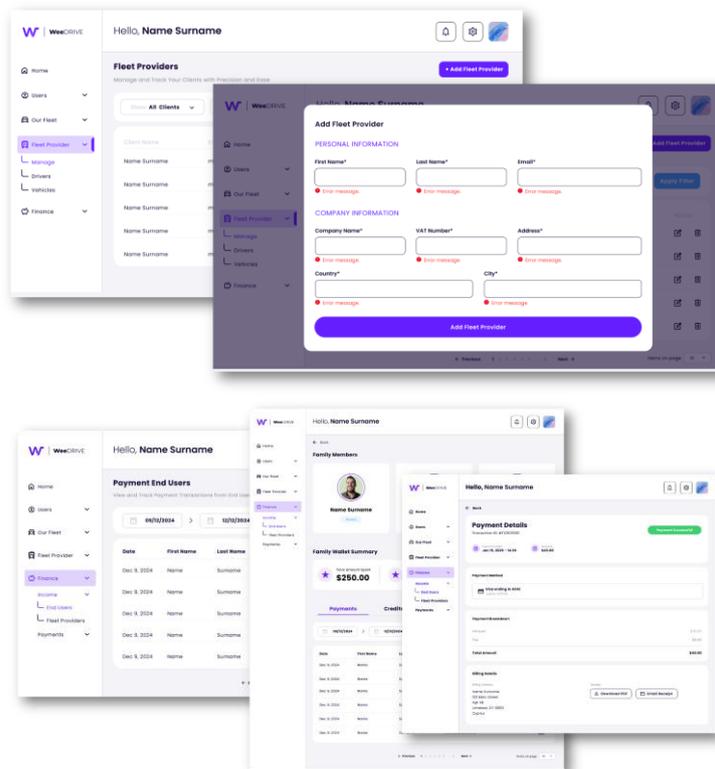


Figure 64: User app interface



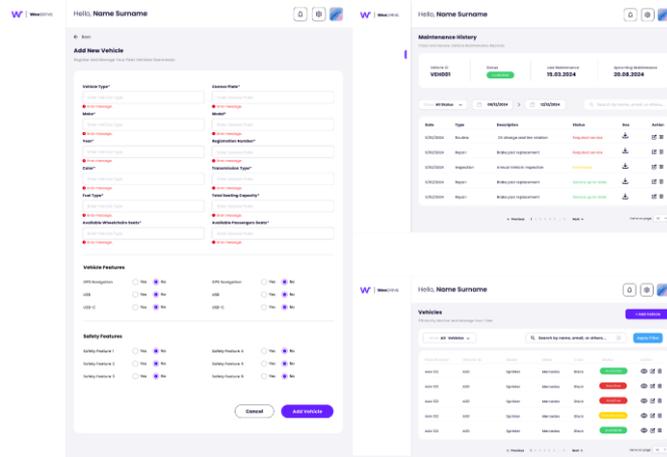


Figure 65: Operator and driver app interface

In the backend interface, tests have been conducted following a two-stage methodology. First, a trip generation process produces a synthetic population of students and their corresponding trip requests. Then, a vehicle routing algorithm is applied to determine optimal routes, considering students' schedules and drivers' break times.

Parents must submit trip requests at least one day in advance using the WeeDrive application. Trips can include three types: school-to-home, school/home-to-activity, activity-to-activity, and activity-to-home. For each trip request, parents must specify the expected pick-up and drop-off times. Parents must pay for each trip a fare that varies with the distance of the trip. Daily routes are generated by the service operator, taking into account the pick-up and drop-off times specified for each student, as well as mandatory driver breaks. The routing process is optimized to minimize unserved travel requests and overall travel distances.

The service is assessed using a two-stage methodology. The first stage aims to generate trip requests. Then, the second stage focus on the supply side, with the application of a Vehicle Routing Problem with Pick-up, Drop-off and Time Windows (VRPPDTW) to determine the routes required to serve the trip requests.

The generation of trip requests relies on two datasets: one detailing origins, such as the locations of student homes and schools, school dismissal times, and the distribution of students across schools; and another containing information about activities, including their locations, schedules, and levels of attractiveness. These datasets, along with a set of parameters specifying the percentage of trips for each activity type and the proportion of students involved in two activities, are used to generate detailed trip requests for students, including origin and destination locations, as well as expected pick-up and drop-off times.

The trip generation process begins with the creation of students based on the distribution of students per school and on the home addresses. Then, a daily plan for each student is defined consisting of at least two trips: one to get to the activity (school/home-activity) and another to return (activity-school/home). Besides those, the daily plans can also include a trip from school to home (school-home) and a trip for an intermediate activity (activity-activity). The daily plan for each student is defined using the following steps: (1) Define whether the daily plan starts from home or from school, (2) Choose first activity, (3) Check if trip school-home is needed, and (4) Determine intermediate activities.

The problem focuses on a passenger service operated using a fixed fleet of minibuses. The route of each minibus must start and finish at the bus depot. The driver associated with each minibus is entitled of a break, which must be taken in any location (not necessarily in the depot) but within a

certain time windows. To guarantee the quality of the service, specific time windows are also defined for the pick-up and drop-off of passengers. The objective is to design routes that maximize the number of trip requests served while minimizing the total travel distance. More specifically, the objective function is defined as the sum of the total distance travelled and of penalties for unassigned trip requests and missed driver breaks. The VRPPDTW is solved using JSPLIT, a Java-based optimization library that employs a meta-heuristic based on the Ruin-and-Recreate principle developed by Schrimpf et al. (2000). This principle is a large neighbourhood search method that combines simulated annealing and threshold-accepting techniques. It begins with an initial solution and follows a "ruin-and-recreate" process. In the ruin step, parts of the solution are dismantled, creating a set of unserved jobs and a partial solution. During the recreate step, the unserved jobs are reintegrated into the partial solution to form a new solution. If the new solution meets quality criteria, it is accepted as the best solution, and the process repeats until a termination condition, such as a time limit or iteration count, is reached.

The service is designed for students attending after-school activities in the municipality of Limassol, Cyprus. It operates on school days between 13:00 and 21:00. At first, we assume the 300 students are served. These students can be picked up either from their schools or homes, with public schools typically finishing at 13:30 and private schools around 15:00. According to a survey conducted by Kamargianni (2013), more than 80% of after-school activities for students in Cyprus involve either tutorial lessons in tuition institutions or sports activities. Among these students, 55% participate in both tutorial lessons and sports activities, while 26% attend only tutorial lessons, and 19% are involved only in sports. For trip generation, it is assumed that students cannot combine two sports activities, and the maximum interval between the end of one activity and the start of the next is 1 hour. Furthermore, student trips to and from activities must fall within a distance range of 750 to 10,000 meters. The service characteristics vary according to the typology of the origin and destination. For trips from home, pick-ups can occur up to 60 minutes before the start of an activity while for trips from schools or other activities, waiting times are limited to 30 minutes after the previous school/activity ends. Students traveling to activities can be dropped off up to 20 minutes before the activity begins, while trips returning home allow for an extended travel time of up to 60 minutes. The locations of the schools, homes, activity facilities and the bus depot are shown in Figure 66.

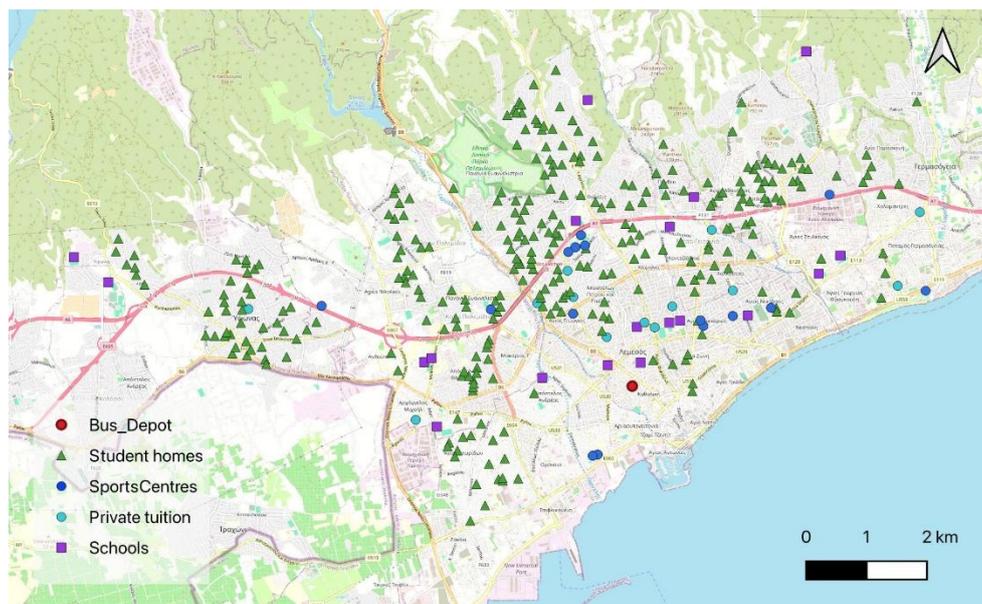
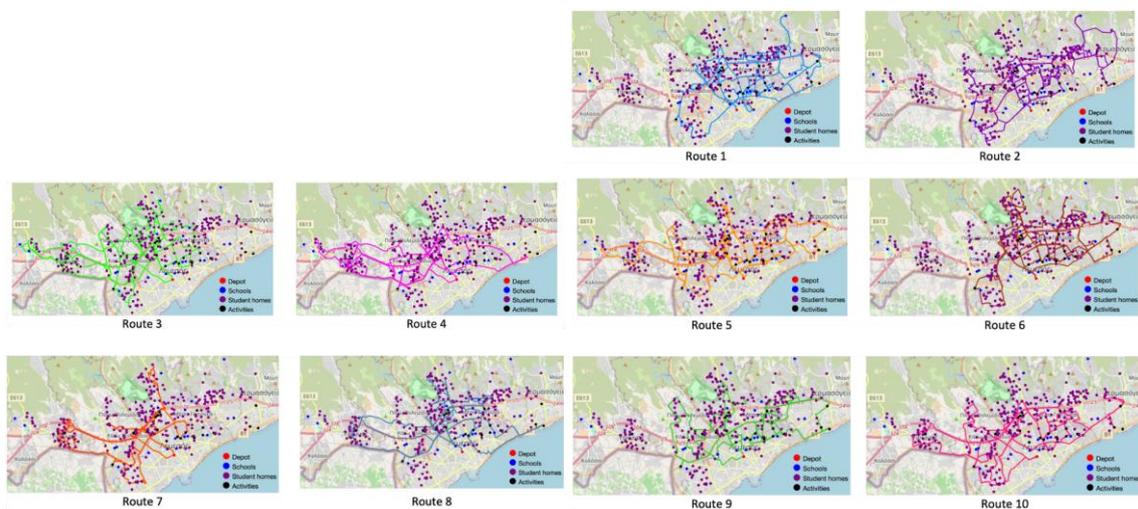


Figure 66: School, home, activity locations selected in Limassol

The WeeDrive service initially deploys a fleet of 10 Ford Transit minibuses, each with 17 seats. Drivers are available from 13:00 to 21:00, with breaks scheduled according to local legislation, whereby a 30-minute break is mandatory between 15:30 and 17:30, while a flexible 15-minute break can be taken between 15:30 and 20:30. Daily operational costs of the service combines the WeeDrive costs and the fleet operator costs.

The preliminary results and operational analysis for the on-demand shared e-service is presented in Figure 68, including the generated routes, the vehicle occupancy matrix, the number of students and their respective number of trips served successfully, as well as several Key Performance Indicators associated with the student population size where all trips could be served successfully (no rejections). We start by exploring the results for the default configuration in terms of service level and characteristics; and then we check the service costs and analyze how different fare scenarios affect profit levels.

In this first analysis, the WeeDrive service with 10 minibuses handles 642 after-school trips from 260 students (2.3 trips per student, on average) with no booking rejections. As the number of students increases beyond this point, trip rejections begin to appear, rising to 23, 21, 30, and 33 for 270, 280, 290, and 300 students, respectively. Almost 80% of these trips are to or from activities, 19% are trips between activities, and only 4% are trips from school to home. The average seating occupancy of minibuses reaches up to 9.7 occupied seats (see the seating occupancy per route for each 15-minute interval in Figure 67). On average, students wait less than 15 minutes before being picked up, except for trips from home, which have an average wait time of 24 minutes. Travel times ranged from 1 to 75 minutes, with an average of 30 minutes. In total, the minibuses travelled 1217 km per day along the routes shown in Figure 67 (118 km per route in average). The daily operation for drivers includes, on average, more than 1h of idle time per day (73 minutes), which is more than 1.5 times the mandatory 45-minute break.



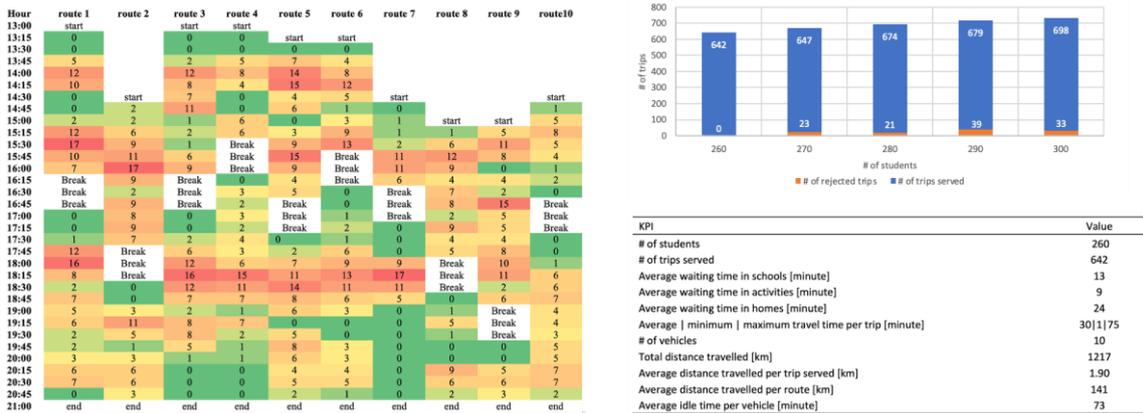


Figure 67: Results for a fleet size of 10 Minibuses with 17-seat capacity

In this methodology, pricing is currently being done ex-post the trip generation and routing steps. The fare charged per trip is calculated based on an initial fee plus a distance-based fare, capped at local taxi fare rates as can be seen in Table 8. Concretely, the pricing structure consists of per trip charges and is calculated ex post the VRPDTW solution, based on 3 distance-based pricing policies. In order to preserve fairness and equity in the pricing, the distance D in the pricing policies is not set as the actual distance \bar{D} traveled during each trip, but as the driving distance from each trip origin to destination, without any detours. In the first and second policies, the fare per trip consists of an initial fee plus a constant distance-based fare. The first policy Pol_1 uses a smaller initial fee of €4.50, with a higher distance-based increment (\geq €0.50), meaning students who travel longer distances are more penalized. In contrast, the second policy Pol_2 uses a higher initial fare of €6 and a smaller distance-based fare ($<$ €0.50), ensuring a more balanced fare for all students. In the third policy Pol_3 , the pricing includes not only the initial fee and distance-based fare but also the average occupancy per vehicle ($AOpv$). Specifically, unless the vehicle carries only one passenger, there is a partial overlap of trips. An equitable way to distribute cost savings is to divide the initial fare by the average occupancy per vehicle ($AOpv$). Therefore, due to this division, this policy is associated with the highest initial fare of €16.6. All trip costs in these policies are capped at local taxi fare rates. The rationale behind this upper bound is that parents are unlikely to pay more than the equivalent taxi fare, given the higher level of comfort taxis offer compared to minibuses.

Table 8: Pricing policies

Name	Total Fare (€)
$TaxiF$	$3.42 + D0.73$
Pol_1	$\min\{4.5 + D0.5, TaxiF\}$
Pol_2	$\min\{6 + D0.3, TaxiF\}$
Pol_3	$\min\{16.6/AOpv + D0.6, TaxiF\}$

For pricing policy Pol_1 , the min, max and average total fares per trip are €3.70, €10, and €6.54 respectively. For pricing policy Pol_2 , the min, max and average total fares per trip are €3.70, €9.30, and €6.54 respectively. For the most equitable pricing policy, Pol_3 , the min, max and average total fares per trip are €3.11, €11.55, and €6.54 respectively. Assuming that the total fare per trip is normally distributed, Pol_3 has longer tails than the distributions of Pol_1 and Pol_2 . The total fares paid by each student per trip are aggregated to derive the ECDF (Empirical Cumulative Distribution Function) for the total student fares per day, shown in Figure 68.

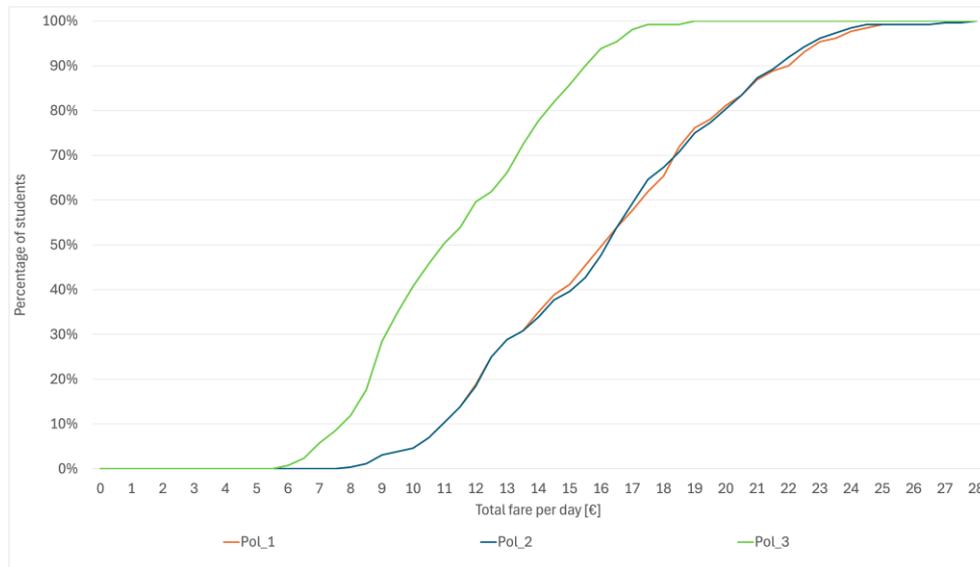


Figure 68: ECDF for the total daily student fare

As is evident from the ECDF, for policies *PoL_1* and *PoL_2*, more than 50% of the student population is charged from €16-€24 per day. On the other hand, *PoL_3* charges daily total student fares not exceeding €18 for 100% of the student population. From a user satisfaction standpoint, *PoL_3* is the best choice.

We have concluded that it is feasible to serve +640 trip requests per day with a fleet of 10, 17-seater minibuses. Comparing various pricing policies at the approximately equal revenue point, we have found that an ex-post pricing policy with initial tariff and distance-based cost divided by the average occupancy per vehicle (AOpv) results in the lowest daily cost for the large majority of the student population.

8. Task 2.7 Integrate + Plan: Digital Twin platform for optimisation

In Task 2.7, the task team's main goal is to deliver a planning and simulation platform integrating the algorithms and frameworks developed by WP2 partners. The platform development is currently being developed in two instances, one by MaaSLab and one by Argaleo. Both platforms' development is aligned and aim for features such as accessing and visualizing geospatial data interactively. The final digital twin platforms will constitute a dynamic, digital representation of real-world assets, networks, and parts of the urban environment based metaCCAZE's datasets, mathematical and digital models, as well as simulation and experimentation results.

Before diving into the actual developments that occurred during the first 18 months of the metaCCAZE, the task team will provide the general background for Task 2.7. The advancement of digital technologies, such as real-time data platforms, geospatial analytics, and simulation models, has accelerated how cities plan, monitor, and manage transport and energy systems, leading to rapid digitalization of planning tools. Cities and regions face interconnected challenges (e.g., congestion, emissions, energy demand), which require planning solutions that consider both mobility and energy infrastructure simultaneously. National and EU-level funding opportunities (e.g. Green Deal, Digital Europe), as well as new calls for proposals have been coming out at an ever-increasing rate with a particular focus on integrated decision-making in transport and energy sectors to address the challenges faced by cities and regions. For that reason, the concept of Digital Twinning will be used to assist metropolitan areas by providing a suite of tools that quantifies the multidimensional impact of various concepts, soft and hard policies on citizens' quality of life,

sustainability, economic growth, while identifying the most appropriate solutions and recommending ways to exploit advances in mobility concepts to achieve their goals.

Despite the emerging opportunities in Digital Twin development, several limitations and challenges remain. These potential barriers have informed the conceptualization of the metaCCAIZE digital twin platforms, designed to enable end-users, such as transport planners, researchers, operators, providers, and decision-makers, to integrate independent models and analyze a portfolio of regional and urban interventions for both passenger and freight mobility, including policies, travel demand management strategies, and new mobility service concepts.

8.1.1. Digital Twin implementations around the world

Datasets that can be used in both Digital Twin implementations spans multiple sectors, including real-time traffic and congestion levels, public transport schedules, usage, and punctuality, as well as energy grid loads and renewable energy production. It also includes building-related data such as floor area, insulation quality, and occupancy, along with environmental indicators like air quality, temperature, and flood risk. Additional layers cover EV charging station locations, usage, and pricing; micromobility flows such as bike-sharing and e-scooters; and population density and demographic trends. Utility network data for water, electricity, and gas, land use and zoning regulations, and detailed 3D city models are also integrated, along with IoT sensor data capturing factors like noise, light, and humidity.

Digital Twin and Geoportal implementations can support a wide range of use cases, including urban planning simulations such as low-emission zones, urban densification, and 15-minute city design. They are also applicable in energy demand forecasting and grid optimization, as well as assessing the impacts of new policies or infrastructure projects. These tools can aid in EV charging network planning by aligning infrastructure with demand and grid capacity, and in disaster risk modelling for events like floods, heatwaves, or energy outages. Additional applications include citizen-facing dashboards to promote transparency and innovation, optimization of public transport and active mobility networks, sustainability and emissions tracking for policy compliance, predictive infrastructure maintenance planning, and visualization of tourism and land use data to support economic development.

The following section provides a brief overview of the types of data commonly used in Digital Twin implementations, along with key use cases for both Digital Twin and Geoportal applications, drawing on insights from existing deployments.

Bathurst Spatial Digital Twin (NSW)

Data Used:

- High-resolution aerial imagery and LiDAR scans,
- 3D building models with 2cm resolution,
- Infrastructure layers including bus stops, parking, and EV stations,
- Zoning and planning overlays.

Use Cases:

- Visualizing proposed buildings in 3D,
- Public consultation and feedback tools,
- Identifying accessibility needs (disabled parking),
- Energy and infrastructure planning.

In Figure 82 of the Annex of this Deliverable, a visual representation of the Bathurst Spatial Digital Twin can be seen.

Los Angeles GeoHub

Data Used:

- Transportation Data: Real-time traffic conditions, public transit routes, and bike lanes.
- Infrastructure Data: Locations of public facilities, utilities, and zoning information.
- Environmental Data: Air quality indices, green spaces, and urban heat maps.
- Public Safety Data: Crime statistics, emergency response times and fire station locations.
- Demographic Data: Population density, income levels, and housing statistics.

Use Cases:

- Urban Planning: Land use and infrastructure development decision making
- Public Engagement: Enabling residents to access and understand city data
- Emergency Response: Real-time visualization of fire, crime, medical emergency locations and GIS layers showing road closures, power outages, damaged buildings.
- Sustainability Initiatives: Data on green spaces and pollution levels.

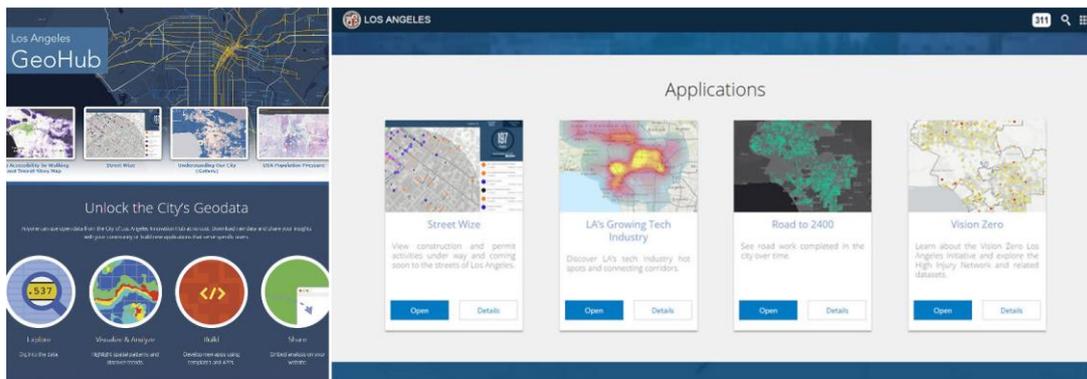


Figure 69: Los Angeles GeoHub (credit ("City of Los Angeles Geohub" 2025))

Paris Open Data portal

Data Used:

- Mobility Data: Real-time traffic conditions, public transport schedules, and cycling infrastructure.
- Environmental Data: Air quality indices, noise pollution levels, and green space distribution.
- Housing and Urban Planning: Building permits, land use, and zoning information.
- Public Services: Locations of schools, libraries, and waste collection schedules.
- Socioeconomic Indicators: Population demographics, employment statistics, and income levels.

Use Cases:

- Urban Planning: Informing infrastructure development and land use decisions.
- Environmental Monitoring: Tracking pollution levels and green space accessibility.
- Public Service Optimization: Enhancing the efficiency of waste collection and public transportation.
- Civic Engagement: Empowering citizens to participate in decision-making processes through accessible data.

In Figure 83 of the Annex of this Deliverable, a visual representation of the Paris Open Data platform can be seen.

8.1.2. Digital Twinning in the Amsterdam Living Lab

In the Amsterdam Living Lab context, the implementation of digital twin technology (see Figures 70 and 71) was supported by the integration of a diverse and comprehensive set of datasets, enabling multidimensional planning and simulation of urban logistics and mobility flows. These datasets included detailed geographical and topographic layers, such as road networks, street typologies, waterway corridors, bridges, and physical terrain boundaries, which form the spatial foundation of the platform. Demographic data was incorporated at the neighbourhood level, offering insight into population density, age distribution, and household compositions that influence local travel demand and waste generation patterns.

A significant addition involved building-level information, including cadastral and planning data, functional use classifications (residential, commercial, public), energy performance indicators, and company registration records. This dataset enabled modelling of user occupancy patterns and the expected intensity of human activity across space and time, which are critical factors in both transport system planning and waste collection logistics. The platform also integrated infrastructure-specific layers such as street capacity constraints, waterborne transport connectivity, and access conditions for different vehicle types. These inputs support nuanced routing, scenario-based optimisation, and identification of potential system bottlenecks.

Specific to waste logistics and operational monitoring, additional data sources included the spatial location of above- and underground waste bins, collection frequency schedules, and barriers such as canal segments or low-clearance passages that restrict routing options for service vehicles. All data layers were harmonised and visualised within a proprietary digital twin software environment developed and operated by Argaleo. The platform is designed to accommodate real-time and static data, and supports modular integration with model outputs from other WP2 tasks.

The resulting system offers a geospatial interface through which local authorities and stakeholders can interactively explore simulation results, compare policy scenarios, and assess the impacts of interventions under real-world constraints. It also enables integration with performance indicators (e.g., KPIs from T1.4) and future extensions to include citizen-facing components for participatory planning. This implementation demonstrates the potential of digital twin environments to bring together data, modelling, and decision support in a transparent and context-sensitive manner, supporting cities in achieving more adaptive and informed urban governance.

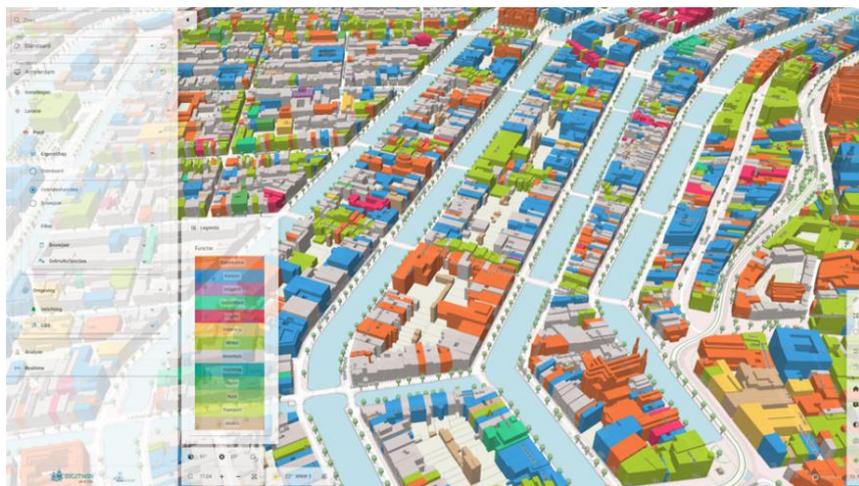


Figure 70: Digital Twin Amsterdam: 3D visualization with building functions.



Figure 71: Digital Twin Amsterdam: Visualization of mobility flows

8.1.3. Geoportal – Digital Twin hybrid for Limassol

The Geoportal – Digital Twin hybrid is amenable to integration of models from willing partners. We have been preparing a preliminary version of the platform. Subject to availability, all possible efforts will be taken for the Geoportal – Digital Twin hybrid to adhere to the data requirements described in Figure 72.

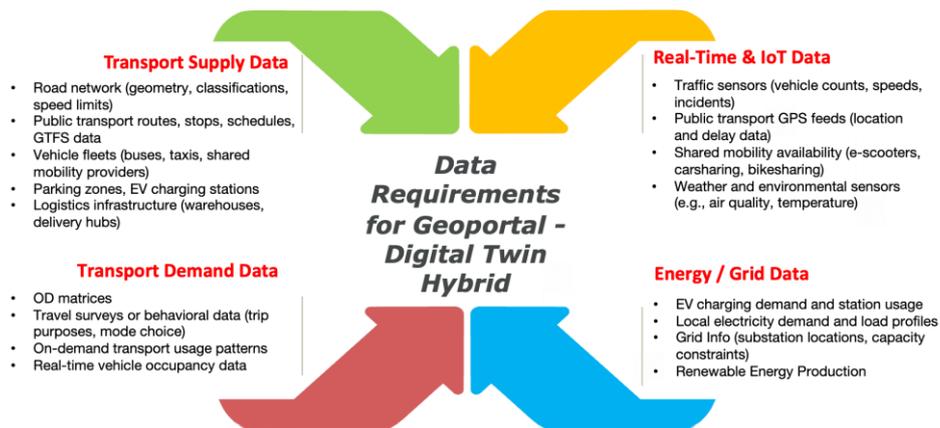


Figure 72: Planned Data Requirements

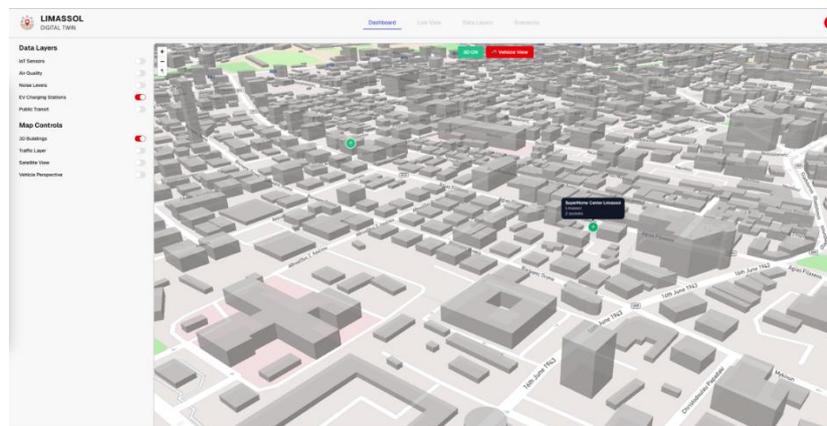


Figure 73: 3D map indicating EV charging locations (preliminary work on Geoportal-Digital Twin hybrid)

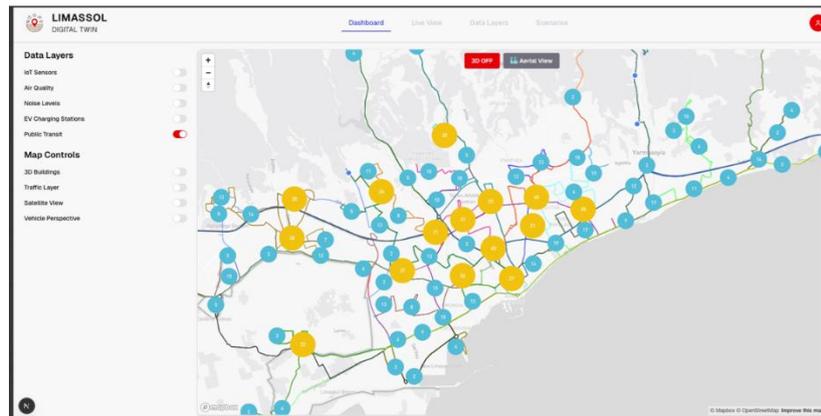


Figure 74: 2D maps indicating public transit routes and bus stop locations (at an aggregated level)

Several layers have been added to our preliminary work on the Geoportal – Digital Twin hybrid which can be activated in various combinations to provide insights to the stakeholders. The list of data layers includes: IoT Sensors, Air Quality, Noise Levels, EV Charging Stations (see Figure 73), and Public Transit (Routes & Bus Stop Locations, see Figure 74).

These can be combined with the Map Controls which include: 3D Buildings, Traffic Layer (Indicating the average speed and levels of congestion on the network), Satellite View, and Vehicle Perspective.

For instance, the Traffic Layer could be combined with the Public Transit Layer to provide an indication of whether the buses are running on time on sections of the network which may be of interest to the stakeholders. Another example would be the combination of the EV Charging Stations Layer and the Public Transit Layer, to provide information about the nearest bus stops to the Charging Stations, facilitating stakeholder transition to alternative modes of transport.

Additional layers are being incorporated to the Geoportal – Digital Twin hybrid and development is ongoing. Documentation and a set of slides are being prepared to be shared with the partners so that an assessment and planning of the integration of their work to the Geoportal – Digital Twin hybrid can take place. The KPIs dashboard of the Geoportal – Digital Twin is currently under development and will include API connectors to the Data Warehouse being developed under Task 2.2. Visualisations of the Standardised Impact Evaluation Framework (T1.4) KPIs are also being considered for the KPIs dashboard.

9. References

- Abdelwahed, A., P. L. van den Berg, T. Brandt, J. Collins, and W. Ketter. 2020. "Evaluating and Optimizing Opportunity Fast-Charging Schedules in Transit Battery Electric Bus Networks." *Transportation Science*, 54 (6): 1601–1615. <https://doi.org/10.1287/trsc.2020.0982>.
- Abdelwahed, A., P. L. van den Berg, T. Brandt, W. Ketter, and J. Mulder. 2021. "A Boost for Urban Sustainability: Optimizing Electric Transit Bus Networks in Rotterdam." *INFORMS Journal on Applied Analytics*, 51 (5): 391–407. <https://doi.org/10.1287/inte.2021.1092>.
- Alonso-Mora, J., S. Samaranyake, A. Wallar, E. Frazzoli, and D. Rus. 2017. "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment." *Proc Natl Acad Sci U S A*, 114 (3): 462–467. National Academy of Sciences. https://doi.org/10.1073/PNAS.1611675114/SUPPL_FILE/PNAS.1611675114.SM01.MP4.
- Álvarez-Ossorio, S., F. Dandl, and K. Bogenberger. 2025. "Analyzing the Effects of Pick-Up and Drop-Off (PUDO) Duration and its Stochasticity on Mobility-On-Demand Service." *Transportation Research Record: Journal of the Transportation Research Board*, In Press.
- Belkin, M., D. Hsu, S. Ma, and S. Mandal. 2019. "Reconciling modern machine-learning practice and the classical bias–variance trade-off." *pnas.org* M Belkin, D Hsu, S Ma, S Mandal *Proceedings of the National Academy of Sciences*, 2019•pnas.org, 116 (32): 15849–15854. National Academy of Sciences. <https://doi.org/10.1073/PNAS.1903070116>.
- Bilali, A., R. Engelhardt, F. Dandl, U. Fastenrath, and K. Bogenberger. 2020. "Analytical and Agent-Based Model to Evaluate Ride-Pooling Impact Factors." *Transp Res Rec*, 2674 (6): 1–12. SAGE Publications Ltd. <https://doi.org/10.1177/0361198120917666;WEBSITE:WEBSITE:SAGE;JOURNAL:JOURNAL:TRR A;WGROU:STRING:PUBLICATION>.
- Biometrics, J. G., and 1971. 1971. "A general coefficient of similarity and some of its properties." *JSTOR* J C Gower *Biometrics*, 1971•JSTOR.
- Borndörfer, R., M. Grötschel, and M. E. Pfetsch. 2008. "Models for line planning in public transport." *Lecture Notes in Economics and Mathematical Systems*, 363–378.
- Bösch, P. M., F. Becker, H. Becker, and K. W. Axhausen. 2018. "Cost-based analysis of autonomous mobility services." *Transp Policy (Oxf)*, 64: 76–91. Pergamon. <https://doi.org/10.1016/j.TRANPOL.2017.09.005>.
- Bueno, A. P. 2021. "Identifying and Quantifying Mobility Hubs."
- Cao, J., M. Menendez, and V. Nikias. 2016. "The effects of on-street parking on the service rate of nearby intersections." *J Adv Transp*, 50 (4): 406–420. John Wiley and Sons Ltd. <https://doi.org/10.1002/ATR.1329;JOURNAL:JOURNAL:20423195;WGROU:STRING:PUBLICATION>.
- Chao, Z., and C. Xiaohong. 2013. "Optimizing Battery Electric Bus Transit Vehicle Scheduling with Battery Exchanging: Model and Case Study." *Procedia Soc Behav Sci*, 96: 2725–2736. Elsevier BV. <https://doi.org/10.1016/j.sbspro.2013.08.306>.
- Chen, S., R. Seshadri, ... C. A. ... R. P. C., and undefined 2023. 2023. "Market design for tradable mobility credits." *Elsevier* S Chen, R Seshadri, CL Azevedo, AP Akkinepally, R Liu, A Araldo, Y Jiang, ME Ben-Akiva *Transportation Research Part C: Emerging Technologies*, 2023•Elsevier.
- Chen, Y., A. Liu, C. Li, S. Li, X. Y.-S. Reports, and undefined 2025. 2025. "Traffic flow prediction based on spatial-temporal multi factor fusion graph convolutional networks." *nature.com* Y Chen, A

- Liu, C Li, S Li, X Yang *Scientific Reports*, 2025•nature.com. <https://doi.org/10.1038/s41598-025-96801-1>.
- “City of Los Angeles Geohub.” 2025. Accessed June 5, 2025. <https://geohub.lacity.org/>.
- CoMoUK, I. S. N. 2019. “Mobility hubs guidance.” Accessed May 30, 2025. <https://www.como.org.uk/documents/comouk-mobility-hubs-guidance>.
- CUTRIC. 2020. *ANNUAL REPORT*.
- Dandl, F., and K. Bogenberger. 2019. “Comparing Future Autonomous Electric Taxis with an Existing Free-Floating Carsharing System.” *IEEE Transactions on Intelligent Transportation Systems*, 20 (6): 2037–2047. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/TITS.2018.2857208>.
- Dandl, F., B. Bracher, and K. Bogenberger. 2017. “Microsimulation of an autonomous taxi-system in Munich.” *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings*, 833–838. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/MTITS.2017.8005628>.
- Engelhardt, R., and K. Bogenberger. 2021. “Benefits of flexible boarding locations in on-demand ride-pooling systems.” *2021 7th International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2021*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/MT-ITS49943.2021.9529284>.
- Engelhardt R., and Dandl F. 2025. “Input data for munich case study.” Fleetpy.
- Engelhardt, R., F. Dandl, A. Bilali, and K. Bogenberger. 2019. “Quantifying the Benefits of Autonomous On-Demand Ride-Pooling: A Simulation Study for Munich, Germany.” *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, 2992–2997. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ITSC.2019.8916955>.
- Engelhardt, R., F. Dandl, and K. Bogenberger. 2020. “Speed-up Heuristic for an On-Demand Ride-Pooling Algorithm.”
- Engelhardt, R., F. Dandl, A.-A. Syed, Y. Zhang, F. Fehn, F. Wolf, and K. Bogenberger. 2022. “FleetPy: A Modular Open-Source Simulation Tool for Mobility On-Demand Services.”
- Fielbaum, A., and J. Alonso-Mora. 2024. “Design of mixed fixed-flexible bus public transport networks by tracking the paths of on-demand vehicles.” *Transp Res Part C Emerg Technol*, 168: 104580. Pergamon. <https://doi.org/10.1016/J.TRC.2024.104580>.
- Fielbaum, A., X. Bai, and J. Alonso-Mora. 2021. “On-demand ridesharing with optimized pick-up and drop-off walking locations.” *Transp Res Part C Emerg Technol*, 126: 103061. Pergamon. <https://doi.org/10.1016/J.TRC.2021.103061>.
- Gao, G., H. Sun, J. Wu, and H. Zhao. 2016. “Tradable credits scheme and transit investment optimization for a two-mode traffic network.” *Wiley Online Library Gao, H Sun, J Wu, H Zhao Journal of Advanced Transportation*, 2016•Wiley Online Library, 50 (8): 1616–1629. <https://doi.org/10.1002/ATR.1418>.
- Geurs, K., A. Grigolon, K. Münzel, K. Gkiotsalitis, D. Duran-Rodas, B. Büttner, C. Kirchberger, J. Pappers, L. Martinez Ramirez, A. Graf, J. Hansel, R. Gkrava, and R. Klementschtz. 2023. “The Smarthubs integration ladder: a conceptual model for the categorisation of shared mobility hubs.” *Taylor & Francis*, 44 (1): 112–139. Routledge. <https://doi.org/10.1080/01441647.2023.2239499>.

- Gkiotsalitis, K. 2021. "Improving service regularity for high-frequency bus services with rescheduling and bus holding." *Journal of Traffic and Transportation Engineering (English Edition)*, 8 (5): 778–794. Chang'an University. <https://doi.org/10.1016/j.jtte.2020.06.002>.
- Gkiotsalitis, K. 2023. *Public Transport Optimization*. Public Transport Optimization. Springer International Publishing.
- Gkiotsalitis, K., C. Iliopoulou, and K. Kepaptsoglou. 2023a. "An exact approach for the multi-depot electric bus scheduling problem with time windows." *Eur J Oper Res*, 306 (1): 189–206. Elsevier B.V. <https://doi.org/10.1016/j.ejor.2022.07.017>.
- Gkiotsalitis, K., C. Iliopoulou, and K. Kepaptsoglou. 2023b. "An exact approach for the multi-depot electric bus scheduling problem with time windows." *Eur J Oper Res*, 306 (1): 189–206. Elsevier B.V. <https://doi.org/10.1016/j.ejor.2022.07.017>.
- Gkiotsalitis, K., D. Rizopoulos, M. Merakou, C. Iliopoulou, T. Liu, and O. Cats. 2025. "Electric bus charging station location selection problem with slow and fast charging." *Appl Energy*, 382: 125242. <https://doi.org/10.1016/j.apenergy.2024.125242>.
- Gkiotsalitis, K., M. Schmidt, and E. van der Hurk. 2022. "Subline frequency setting for autonomous minibusses under demand uncertainty." *Transp Res Part C Emerg Technol*, 135. Elsevier Ltd. <https://doi.org/10.1016/j.trc.2021.103492>.
- Goodall, N. 2020. "Non-technological challenges for the remote operation of automated vehicles." *Transp Res Part A Policy Pract*, 142: 14–26. Pergamon. <https://doi.org/10.1016/j.TRA.2020.09.024>.
- Hamm, L., S. Martínez, A. Loder, K. B.-T. Policy, and undefined 2025. 2025. "Who Gets What? A user perspective on initial credit allocation in tradable mobility credit schemes." *Elsevier*.
- Hoogendoorn-Lanser, S., ... N. S.-T. R., and undefined 2015. 2015. "The Netherlands Mobility Panel: An innovative design approach for web-based longitudinal travel data collection." *ElsevierS Hoogendoorn-Lanser, NTW Schaap, MJ OldeKalterTransportation Research Procedia, 2015•Elsevier*.
- Hsu, Y. T., S. Yan, and P. Huang. 2021. "The depot and charging facility location problem for electrifying urban bus services." *Transp Res D Transp Environ*, 100. Elsevier Ltd. <https://doi.org/10.1016/j.trd.2021.103053>.
- Hunter, C. B., K. M. Kockelman, and S. Djavadian. 2024. "Curb Allocation and Pick-Up Drop-Off Aggregation for a Shared Autonomous Vehicle Fleet." *Int Reg Sci Rev*, 47 (2): 131–158. SAGE Publications Inc. https://doi.org/10.1177/01600176231160498/ASSET/8DF79700-CB7E-40BE-A359-397FDCC2550C/ASSETS/IMAGES/LARGE/10.1177_01600176231160498-FIG7.JPG.
- Hyland, M. F., and H. S. Mahmassani. 2017. "Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility." *journals.sagepub.comMF Hyland, HS MahmassaniTransportation Research Record, 2017•journals.sagepub.com*, 2653: 26–34. National Research Council. <https://doi.org/10.3141/2653-04>.
- Its. 2023. "TS 103 900 - V2.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Cooperative Awareness Service; Release 2."
- Its. 2024a. "TS 103 926 - V2.1.1 - Intelligent Transport Systems (ITS); Testing; Collective Perception Service (CPS); Interoperability tests specification; Release 2."
- Its. 2024b. "TS 103 916 - V2.1.1 - Intelligent Transport Systems (ITS); Parking Availability Service; Release 2."

- Jepsen, M., S. Spoorendonk, and S. Ropke. 2013. "A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem." *pubsonline.informs.org* M Jepsen, S Spoorendonk, S Ropke *Transportation Science*, 2013•*pubsonline.informs.org*, 47 (1): 23–37. INFORMS Inst.for Operations Res.and the Management Sciences. <https://doi.org/10.1287/TRSC.1110.0399>.
- Jiang, W., and J. Luo. 2022. "Graph neural network for traffic forecasting: A survey." *Expert Syst Appl*, 207: 117921. Pergamon. <https://doi.org/10.1016/J.ESWA.2022.117921>.
- Johari, M., M. Keyvan-Ekbatani, and D. Ngoduy. 2020. "Impacts of bus stop location and berth number on urban network traffic performance." *IET Intelligent Transport Systems*, 14 (12): 1546–1554. Institution of Engineering and Technology. <https://doi.org/10.1049/IET-ITS.2019.0860;WEBSITE:WEBSITE:IETRESEARCH;WGROU:STRING:PUBLICATION>.
- Kamargianni, M. 2013. "Development of hybrid models of teenager's travel behavior to school and to after-school activities."
- Kchaou-Boujelben, M. 2021. "Charging station location problem: A comprehensive review on models and solution approaches." *Transp Res Part C Emerg Technol*. Elsevier Ltd.
- Kettwich, C., A. Schrank, H. Avsar, and M. Oehl. n.d. "Teleoperated Driving." ACM. <https://doi.org/10.1145/3473682.3480271>.
- Kim, S., R. Pasupathy, and S. Henderson. 2015. "A Guide to Sample Average Approximation." 207–243.
- Ku, D., M. Choi, D. Lee, and S. Lee. 2022. "The effect of a smart mobility hub based on concepts of metabolism and retrofitting." *Elsevier* D Ku, M Choi, D Lee, S Lee *Journal of Cleaner Production*, 2022•*Elsevier*, 379. Elsevier Ltd. <https://doi.org/10.1016/J.JCLEPRO.2022.134709>.
- Li, M., Z. Z.-P. of the A. conference on artificial, and undefined 2021. 2021. "Spatial-temporal fusion graph neural networks for traffic flow forecasting." *ojs.aaai.org* M Li, Z Zhu *Proceedings of the AAAI conference on artificial intelligence, 2021*•*ojs.aaai.org*.
- Li, X., X. An, and B. Zhang. 2024. "Minimizing passenger waiting time in the multi-route bus fleet allocation problem through distributionally robust optimization and reinforcement learning." *Comput Oper Res*, 164: 106568. Pergamon. <https://doi.org/10.1016/J.COR.2024.106568>.
- Liu, T., and A. (Avi) Ceder. 2020. "Battery-electric transit vehicle scheduling with optimal number of stationary chargers." *Transp Res Part C Emerg Technol*, 114: 118–139. Elsevier Ltd. <https://doi.org/10.1016/j.trc.2020.02.009>.
- Lv, M., Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji. 2020. "Temporal multi-graph convolutional network for traffic flow prediction." *ieeexplore.ieee.org* M Lv, Z Hong, L Chen, T Chen, T Zhu, S Ji *IEEE Transactions on Intelligent Transportation Systems*, 2020•*ieeexplore.ieee.org*. <https://doi.org/10.1109/TITS.2020.2983763>.
- Lyu, Y., C. Y. Chow, V. C. S. Lee, J. K. Y. Ng, Y. Li, and J. Zeng. 2019. "CB-Planner: A bus line planning framework for customized bus systems." *Transp Res Part C Emerg Technol*, 101: 233–253. Elsevier Ltd. <https://doi.org/10.1016/j.trc.2019.02.006>.
- Machado-León, J. L., D. MacKenzie, and A. Goodchild. 2024. "An empirical analysis of passenger vehicle dwell time and curb management strategies for ride-hailing pick-up/drop-off operations." *Springer* J L Machado-León, D MacKenzie, A Goodchild *Transportation*, 2024•*Springer*, 51 (5): 1635–1661. Springer. <https://doi.org/10.1007/S11116-023-10380-6>.
- McInnes, L., J. Healy, S. A.-J. O. S. Softw., and undefined 2017. 2017. "hdbscan: Hierarchical density based clustering." *scholar.archive.org*.

- Nkembi, A. A., P. Cova, I. Kortabarria, E. Sacchi, and N. Delmonte. 2023. "An improved modelling and dynamic control of the dual active bridge converter for fast battery charging of electric vehicles." *IET Conference Proceedings*, 2023 (17): 247–254. The Institution of Engineering and Technology Stevenage, UK. <https://doi.org/10.1049/ICP.2023.2006>.
- Olsen, N., and N. Kliewer. 2022. "Location Planning of Charging Stations for Electric Buses in Public Transport Considering Vehicle Scheduling: A Variable Neighborhood Search Based Approach." *Applied Sciences (Switzerland)*, 12 (8). MDPI. <https://doi.org/10.3390/app12083855>.
- "Open Data Paris." 2025. Accessed June 5, 2025. <https://opendata.paris.fr/pages/home/>.
- Pourmohammad-Zia, N., M. van K.-S. C. and, and undefined 2024. 2024. "Sustainable urban logistics: A case study of waterway integration in Amsterdam." *ElsevierN Pourmohammad-Zia, M van Koningsveld Sustainable Cities and Society, 2024•Elsevier*, 105. Elsevier Ltd. <https://doi.org/10.1016/J.SCS.2024.105334>.
- Provoost, J., O. Cats, S. H.-T. Policy, and undefined 2023. 2023. "Design and classification of tradable mobility credit schemes." *Elsevier*.
- Randhahn Annette and Knotte, T. 2020. "Deployment of Charging Infrastructure for Battery Electric Buses." *Towards User-Centric Transport in Europe 2: Enablers of Inclusive, Seamless and Sustainable Mobility*, G. Müller Beate and Meyer, ed., 169–183. Cham: Springer International Publishing.
- Rashidi, S., S. Ataeian, and P. Ranjitkar. 2023. "Estimating bus dwell time: A review of the literature." *Taylor & FrancisS Rashidi, S Ataeian, P Ranjitkar Transport Reviews, 2023•Taylor & Francis*, 43 (1): 32–61. Routledge. <https://doi.org/10.1080/01441647.2021.2023692>.
- Rizopoulos, D., and K. Gkiotsalitis. 2025. "Extending electric bus charging infrastructure considering charging scheduling and energy pricing." *Transp Res Part C Emerg Technol*, 176: 105141. <https://doi.org/10.1016/j.trc.2025.105141>.
- Rogge, M., E. van der Hurk, A. Larsen, and D. U. Sauer. 2018. "Electric bus fleet size and mix problem with optimization of charging infrastructure." *Appl Energy*, 211: 282–295. Elsevier Ltd. <https://doi.org/10.1016/j.apenergy.2017.11.051>.
- Rongen, T., T. Tillema, J. Arts, M. J. Alonso-González, and J. J. Witte. 2022. "An analysis of the mobility hub concept in the Netherlands: Historical lessons for its implementation." *ElsevierT Rongen, T Tillema, J Arts, MJ Alonso-González, JJ Witte Journal of Transport Geography, 2022•Elsevier*, 104. Elsevier Ltd. <https://doi.org/10.1016/J.JTRANGEO.2022.103419>.
- Saltelli, A., P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. 2010. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index." *Comput Phys Commun*, 181 (2): 259–270. North-Holland. <https://doi.org/10.1016/J.CPC.2009.09.018>.
- Schöbel, A. 2012. "Line planning in public transportation: Models and methods." *OR Spectrum*.
- Schrimpf, G., J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. 2000. "Record breaking optimization results using the ruin and recreate principle." *ElsevierG Schrimpf, J Schneider, H Stamm-Wilbrandt, G Dueck Journal of Computational Physics, 2000•Elsevier*, 159 (2): 139–171. Academic Press Inc. <https://doi.org/10.1006/JCPH.1999.6413>.
- Shaheen, S., A. Cohen, I. Zohdy, and B. Kock. 2016. "Shared mobility: Current practices and guiding principles brief."
- Sobol', I. M. 1967. "The distribution of points in a cube and the approximate evaluation of integrals."

- Stiglic, M., N. Agatz, ... M. S.-... R. P. B., and undefined 2015. 2015. "The benefits of meeting points in ride-sharing systems." *ElsevierM Stiglic, N Agatz, M Savelsbergh, M GradisarTransportation Research Part B: Methodological, 2015•Elsevier*.
- "Stricter rules for heavy vehicles." 2024.
- Stumpe, M., D. Rößler, G. Schryen, and N. Kliewer. 2021. "Study on sensitivity of electric bus systems under simultaneous optimization of charging infrastructure and vehicle schedules." *EURO Journal on Transportation and Logistics*, 10. Elsevier B.V. <https://doi.org/10.1016/j.ejtl.2021.100049>.
- Telikani, A., A. Sarkar, ... B. D.-I. T. on, and undefined 2024. 2024. "Machine learning for uav-aided its: A review with comparative study." *ieeexplore.ieee.orgA Telikani, A Sarkar, B Du, J ShenIEEE Transactions on Intelligent Transportation Systems, 2024•ieeexplore.ieee.org*, 1–19. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/TITS.2024.3422039>.
- Teng, J., T. Chen, and W. "David" Fan. 2020. "Integrated Approach to Vehicle Scheduling and Bus Timetabling for an Electric Bus Line." *J Transp Eng A Syst*, 146 (2). American Society of Civil Engineers (ASCE). <https://doi.org/10.1061/jtepbs.0000306>.
- Uslu, T., and O. Kaya. 2021. "Location and capacity decisions for electric bus charging stations considering waiting times." *Transp Res D Transp Environ*, 90. Elsevier Ltd. <https://doi.org/10.1016/j.trd.2020.102645>.
- Valença, G., F. Moura, and A. Morais de Sá. 2021. "Main challenges and opportunities to dynamic road space allocation: From static to dynamic urban designs." *Elsevier*, 1. Elsevier Ltd. <https://doi.org/10.1016/J.URBMOB.2021.100008>.
- Veitch, E., and O. Andreas Alsos. 2022. "A systematic review of human-AI interaction in autonomous ship systems." *Saf Sci*, 152: 105778. Elsevier. <https://doi.org/10.1016/J.SSCI.2022.105778>.
- "Virtual Singapore - a 3D city model platform for knowledge sharing and community collaboration." 2025. Accessed June 5, 2025. <https://www.sla.gov.sg/articles/press-releases/2014/virtual-singapore-a-3d-city-model-platform-for-knowledge-sharing-and-community-collaboration>.
- Vuelas, J., F. Ruiz, and G. Gruosso. 2021. "A time-of-use pricing strategy for managing electric vehicle clusters." *Sustainable Energy, Grids and Networks*, 25: 100411. <https://doi.org/10.1016/j.segan.2020.100411>.
- Wang, Z., M. F. Hyland, Y. Bahk, and N. J. Sarma. 2022. "On Optimizing Shared-ride Mobility Services with Walking Legs."
- Wolf, F., R. Engelhardt, Y. Zhang, F. Dandl, and K. Bogenberger. 2023. "Effects of Dynamic and Stochastic Travel Times on the Operation of Mobility-on-Demand Services." *ieeexplore.ieee*, 5476–5481. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ITSC57777.2023.10422554>.
- Wu, M., Y. Lin, T. Jiang, and W. Weng. 2025. "MHGNet: Multi-Heterogeneous Graph Neural Network for Traffic Prediction."
- Wu, W., Y. Lin, R. Liu, and W. Jin. 2022. "The multi-depot electric vehicle scheduling problem with power grid characteristics." *Transportation Research Part B: Methodological*, 155: 322–347. Elsevier Ltd. <https://doi.org/10.1016/j.trb.2021.11.007>.
- Wu, X., Q. Feng, C. Bai, C. S. Lai, Y. Jia, and L. L. Lai. 2021. "A novel fast-charging stations locational planning model for electric bus transit system." *Energy*, 224. Elsevier Ltd. <https://doi.org/10.1016/j.energy.2021.120106>.

- Wu, Z., F. Guo, J. Polak, and G. Strbac. 2019. "Evaluating grid-interactive electric bus operation and demand response with load management tariff." *Appl Energy*, 255: 113798. <https://doi.org/10.1016/j.apenergy.2019.113798>.
- Xiong, W., R. Fonod, A. Alahi, and N. Geroliminis. 2025. "Multi-Source Urban Traffic Flow Forecasting with Drone and Loop Detector Data."
- Xu, W., X. Du, R. Li, B. Li, Y. Jiao, and L. Xing. 2025. "Attention-enhanced StrongSORT for robust vehicle tracking in complex environments." *nature.com*, 15 (1). Nature Research. <https://doi.org/10.1038/S41598-025-99524-5>.
- Ye, Z Wang, X Chen, J Wang, K Wu, K Lu. 2021. "GSAN: Graph self-attention network for learning spatial-temporal interaction representation in autonomous driving." *ieeexplore.ieee.org IEEE Internet of Things Journal*, 2021•ieeexplore.ieee.org.
- Zhang, L., S. Xu, S. Li, ... L. P.-S. (Basel, and undefined 2025. 2025. "D-MGDCN-CLSTM: A Traffic Prediction Model Based on Multi-Graph Gated Convolution and Convolutional Long-Short-Term Memory." *pmc.ncbi.nlm.nih.gov*
- Zwick, F., and K. W. Axhausen. 2020. "Impact of Service Design on Urban Ridepooling Systems." *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ITSC45102.2020.9294289>.
- Zwick, F., K. A.-2020 I. 23rd International, and undefined 2020. 2020. "Impact of service design on urban ridepooling systems." *ieeexplore.ieee.org*F Zwick, KW Axhausen2020 *IEEE 23rd International Conference on Intelligent, 2020*•ieeexplore.ieee.org.

10. Annex

10.1. Annex sub-chapter for Chapter 2 and Task 2.1

As described in Chapter 2, Figure 77 below includes a depiction of the three time horizons considered in the model developed under Subtask 2.1.1.

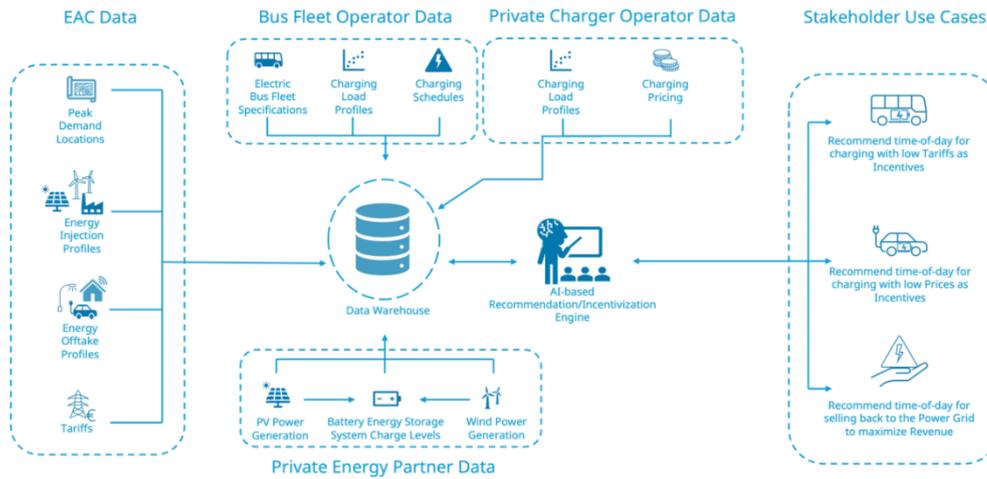


Figure 75: The conceptual framework architecture

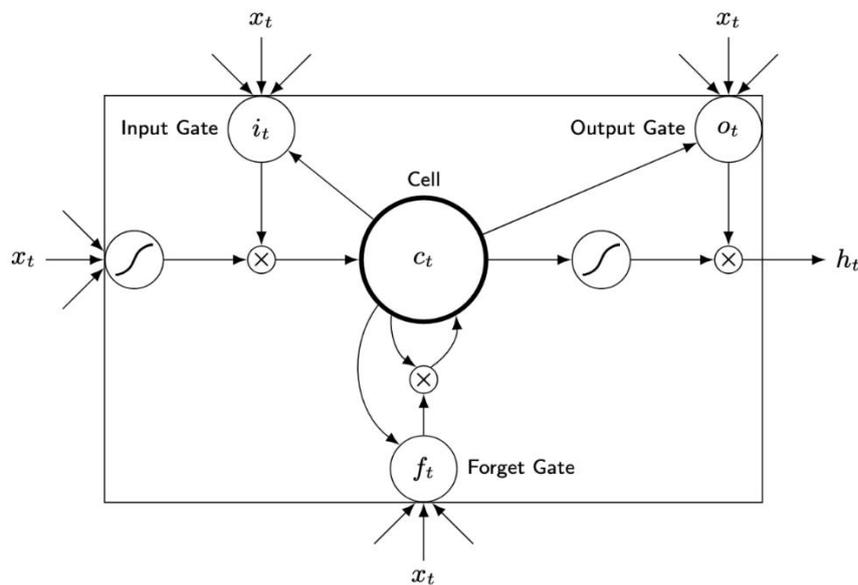


Figure 76: Example of LSTM Unit

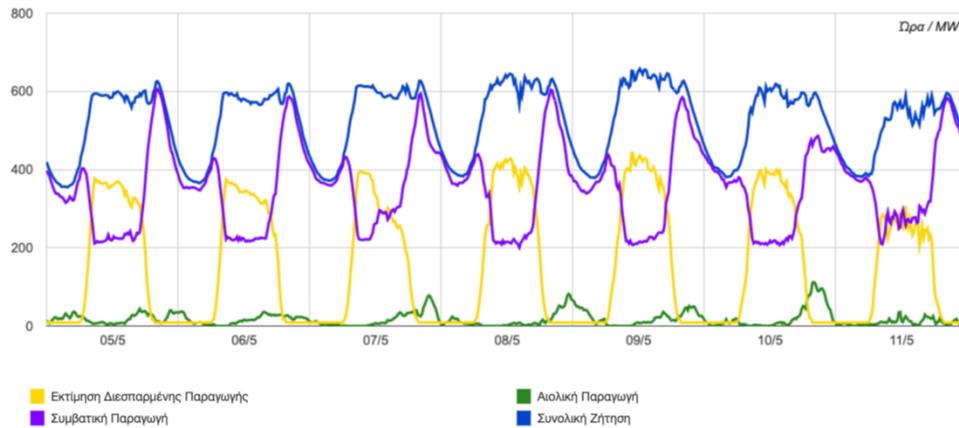


Figure 77: Wind, Distributed, Conventional Generation, and Total Power Demand (Sourced from Cyprus TSOC)

10.2. Annex sub-chapter for Chapter 3 and Task 2.2

No specific material is referenced here for Chapter 3 and Task 2.2.

10.3. Annex sub-chapter for Chapter 4 and Task 2.3

No specific material is referenced here for Chapter 4 and Task 2.3.

10.4. Annex sub-chapter for Chapter 5 and Task 2.4

Subtask 2.4.2

Table 9: Sensors Listing

SENSOR TYPE	MANUFACTURER	MODEL	Q.TY
Depth Camera	StereoLabs	Zed2i	2
3D Lidar	Hesai	FT120	2
3DSolidState Lidar	Robosense	HELIOS H32F70	1
3D Lidar	Robosense	RS-LiDAR-16	1
GPS	ArduSimple	simpleRTK2B	1
RTK Service	Hexagon	HxGN SmartNet	1

Table 10: Lidar Specifications

MOUNTING POSITION	MANUFACTURER	MODEL	SPECIFICATIONS
Front	Robosense	HELIOS H32F70	VFOV:70°, HFOV: 360°, TILT: 10° RANGE: 150m
Left and Right sides	Hesai	FT120	VFOV: 90°, HFOV: 120°, TILT: 0° RANGE: 25m
Rear	Robosense	RS-LiDAR-16	VFOV:30°, HFOV: 360°, TILT: 80° RANGE: 100m (3m after tilting)



Figure 78: Next Modular Vehicles' Third Generation Pod

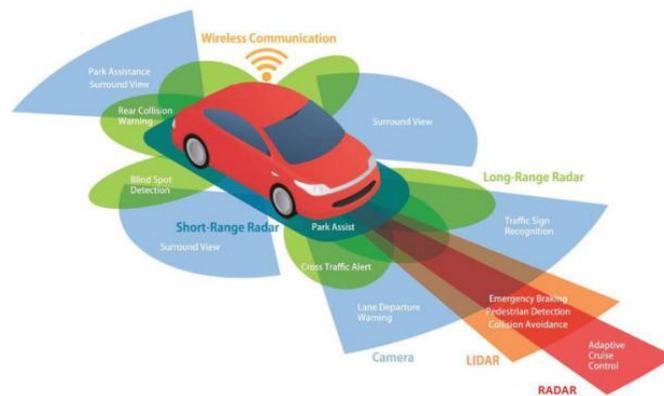


Figure 79: Standard Sensor Coverage for AMVs. Source: Wireless Communication and Sensors in Self-Driving Cars - Pizarov, Jelena & Mester, Gyula. (2020). The Future of Autonomous Vehicles. FME Transactions.

Subtask 2.4.3

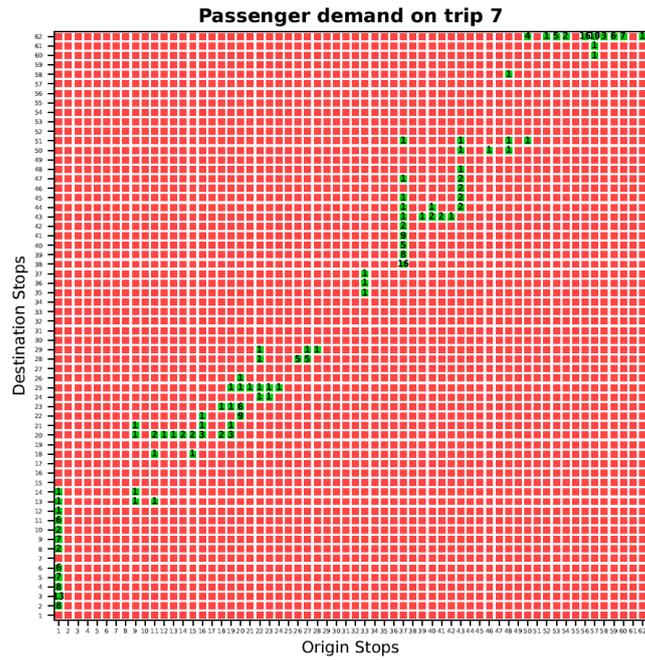


Figure 80: O-D Pair of Case Study

10.5. Annex sub-chapter for Chapter 6 and Task 2.5

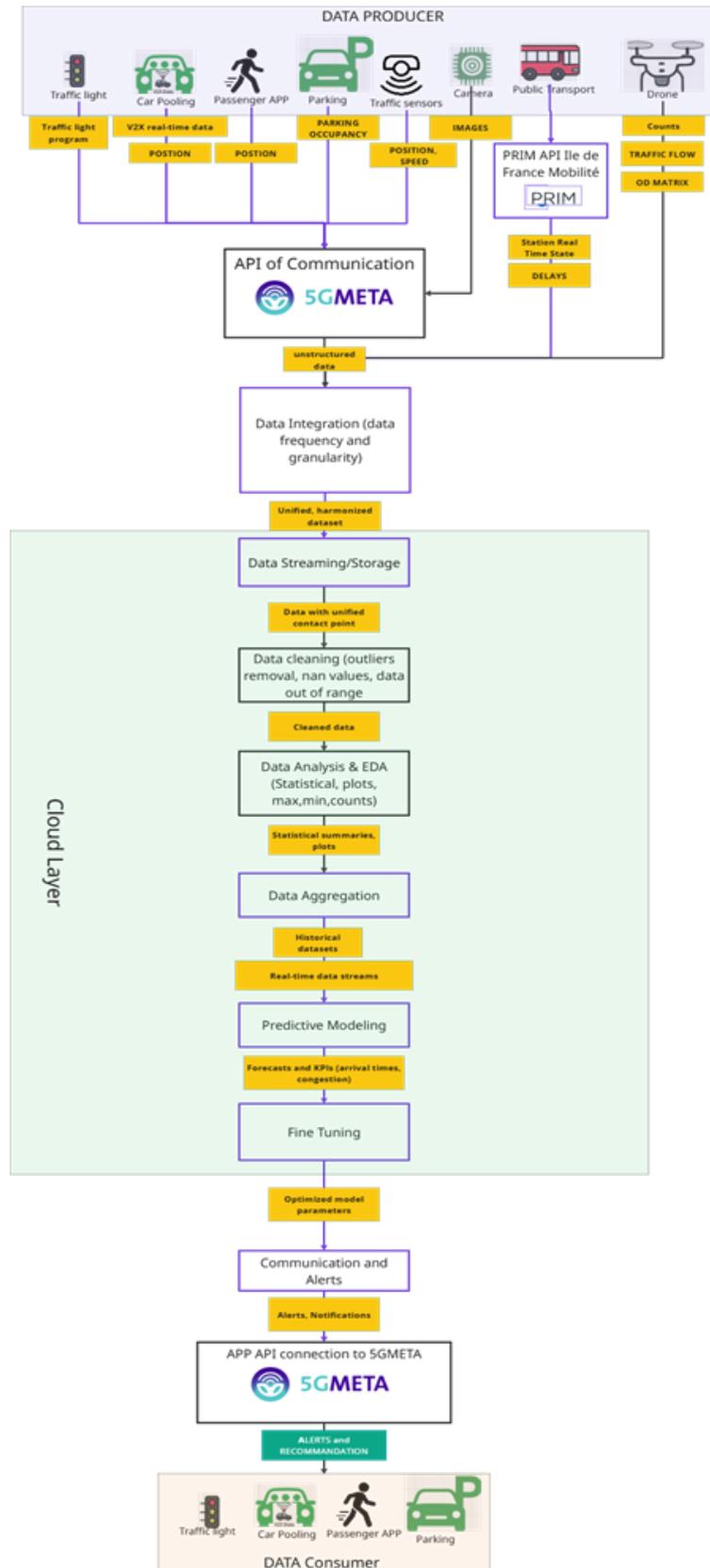


Figure 81: Flowchart of Traffic Management Framework - From Sensors to Services: A Scalable Pipeline for Predictive Traffic Management

10.6. Annex sub-chapter for Chapter 7 and Task 2.6

No specific material is referenced here for Chapter 7 and Task 2.6.

10.7. Annex sub-chapter for Chapter 8 and Task 2.7

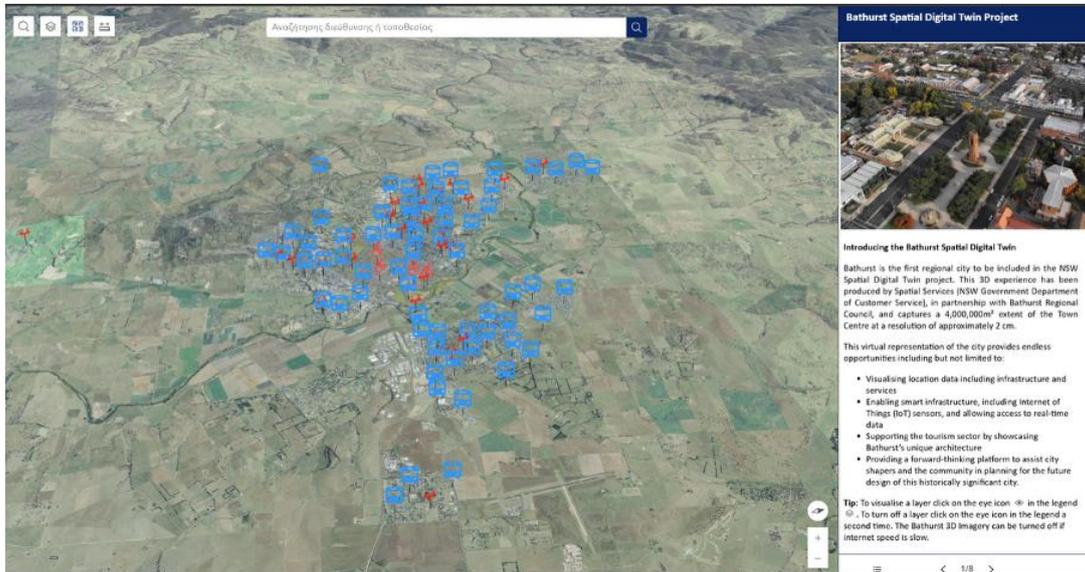


Figure 82: Bathurst Spatial Digital Twin (credit (“Virtual Singapore - a 3D city model platform for knowledge sharing and community collaboration” 2025))

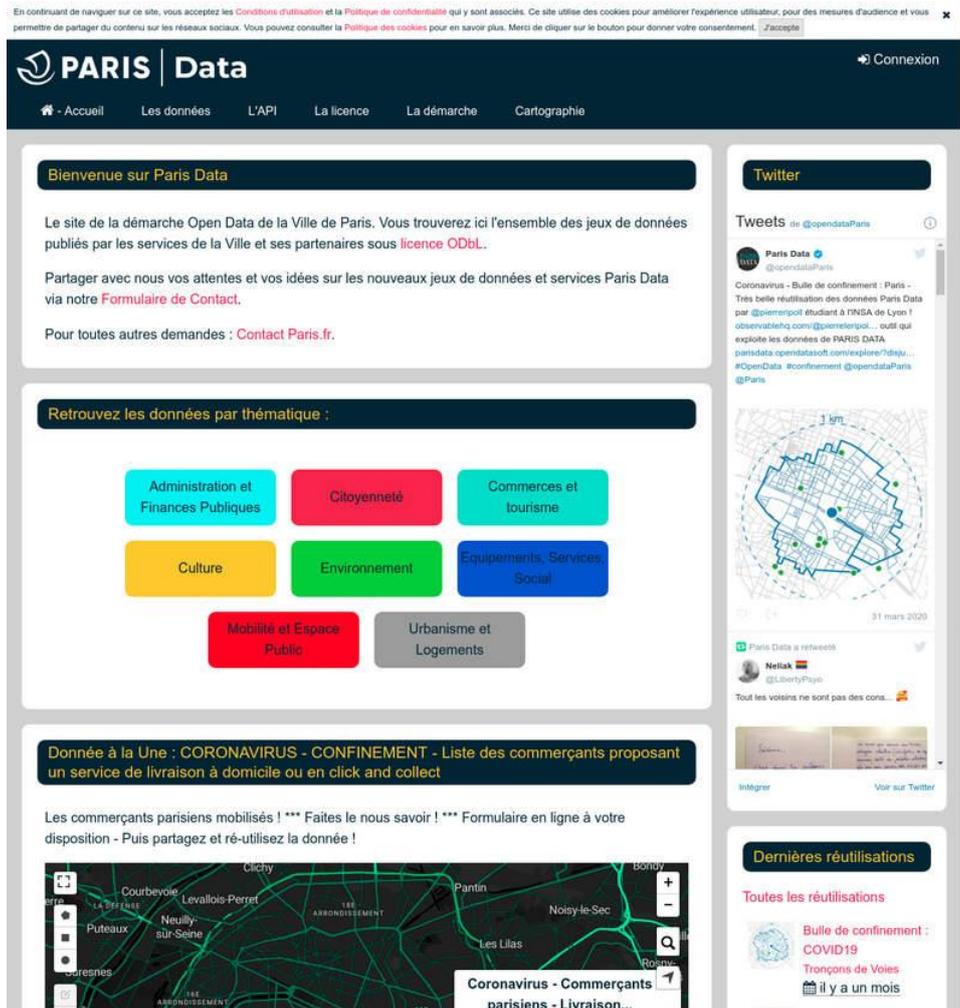


Figure 83: Paris Open Data (credit ("Open Data Paris" 2025))

www.metaCCAZE-project.eu



@metaCCAZE



metaCCAZE-project