

Deliverable 2.2: MetaInnovations – Second version

WP2

Date of document

28/02/2026 (M26)

Deliverable Version: D2.2, V2.0

Dissemination Level: Public

Deliverable Lead: MaaSLab (MLAB)



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 – Second version
Deliverable No.	2.2
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	MaaS Lab
Lead Authors	Antonis Lentzakis, Anne Patricio, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MLAB)
Main Contributor Authors	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA), Hassan Mahdavi (VED), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT), Aleks Tepponen (REMOTED), Manos Barmounakis (MOBILYSIS), Santiago Alvarez (TUM), Leonor Gomes, Jiri Spitzer (TUD), Julian Garbiso (FLWR), Jeroen Steenbakkers (ARGALEO)
Other Contributors	TRASPORTI E TERRITORIO SRL (TRT), HC Linear Muszaki Fejlesztő Kft (HCL), TAMPEREEN KORKEAKOULUSAATIO SR (TAU)
Due date	28/02/2026
Submission date	28/02/2026
Reviewers	Konstantinos Gkiotsalitis, Dimitrios Rizopoulos, Androniki Dimitriadou, Alexandra Vasileiadou, Marilena Merakou (NTUA), Leonor Gomes, Jiri Spitzer (TUD)
Quality Assurance Reviewer	Loha Hashimy (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 or the European Climate, Infrastructure and Environment Executive Agency (CINEA). 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
31/10/2025	0.1	Antonis Lentzakis, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MLAB)	First draft of deliverable and initialization of metadata.
17/11/2025	0.2	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT), Mika Rytönen Julian Garbiso (REMOTED), (FLWR), Santiago Alvarez-Ossorio Martinez (TUM), Hassan Mahdavi, Idoudi Hassan (VED), Manos Barmounakis (MOBILYSIS), Leonor Gomes, Jiri Spitzer (TUD)	First input by partners after request for input (by WP leader) at the WP2 monthly meeting.
30/11/2025	0.3	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT), Mika Rytönen Julian Garbiso (REMOTED), (FLWR), Santiago Alvarez-Ossorio Martinez (TUM), Hassan Mahdavi, Idoudi Hassan (VED), Manos Barmounakis (MOBILYSIS), Leonor Gomes, Jiri Spitzer (TUD)	Completion of several deliverable sub-sections by partners.
22/12/2025	0.4	Antonis Lentzakis (MLAB), Jeroen Steenbakkens (ARGALEO)	Input received and integrated for Task 2.7.
10/02/2026	0.5	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA), Alessandra A. Improta, Andrea Papola, Giovanni Albano (UNINA), Alessandro Busato (NEXT), Mika Rytönen Julian Garbiso (REMOTED), (FLWR), Santiago Alvarez-Ossorio Martinez (TUM), Hassan Mahdavi, Idoudi Hassan (VED), Manos Barmounakis (MOBILYSIS), Leonor Gomes, Jiri Spitzer (TUD)	Finalization of input by WP2 partners.

11/02/2025	0.6	Antonis Lentzakis, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MLAB), Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA)	Formatting homogenization.
13/02/2026	0.7	Konstantinos Gkiotsalitis, Androniki Dimitriadou, Marilena Merakou, Dimitrios Rizopoulos, Alexandra Vasileiadou (NTUA), Leonor Gomes, Jiri Spitzer (TUD)	First review and addressing of reviewers' comments.
23/02/2026	0.8	Loha Hashimy (FACTUAL)	Quality check.
28/02/2026	1.0	Antonis Lentzakis, Maria Kamargianni, Kleanthis Kapsas, Theodora Betsidou (MLAB)	Reviewers' comments addressed and submission to project coordinator.
28/05/2026	2.0	Anne Patricio (MLAB)	Provide an updated version based on PO's comments in the rejection letter.
26/06/2026	2.1	Anne Patricio (MLAB)	Provide an updated version with a more comprehensive Table of MetaInnovations in Section 1.3

Table of Contents

Table of Contents	V
Table of Tables.....	VII
Table of Figures	VIII
Abbreviations and Acronyms.....	XI
Background: About the metaCCAZE project	XV
Executive Summary.....	XVI
1. Introduction	1
1.1. Objectives of the Deliverable	1
1.2. Structure of the Document.....	2
1.3. Relation to Project Documents	4
1.4. Overall Approach	5
2. Task 2.1 Align: AI optimisation and incentivisation engine for grid-fleet-demand	6
2.1. Subtask 2.1.1 Decentralised AI optimisation for grid-fleet-demand	7
The Grid-Fleet-Demand Management	7
Grid Management.....	12
2.2. Subtask 2.1.2 AI recommendation and incentivization engine.....	15
Sustainable Charging Recommendation Engine.....	16
Renewable Energy Sources electricity mix contribution forecasting	16
Decision support platform to achieve sustainable charging practices	18
3. Task 2.2 Harmonise: AI-Data warehouse and APIs.....	18
3.1. Subtask 2.2.1 AI-Data Models and Warehouse	19
3.2. Subtask 2.2.2 APIs and Connectors	20
4. Task 2.3 Charge: Automated charging & V2G protocols.....	20
4.1. Assessment of Automated Charging Technologies.....	21
4.2. Automated Charging System.....	22
Architecture	22
Automated Charging Process.....	24
Next Steps.....	25
5. Task 2.4 Automate: Advanced driving assistance systems and remote-control centers for AVs	25
5.1. Subtask 2.4.1 Remote Control Centre for AVs	26
Main Activities	27
5.2. Subtask 2.4.2 Advanced Driver Assistance System	30
The Robot: Next Modular Vehicles' Pod (Third Generation).....	30
Advanced Driver Assistance System.....	33
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 ..	42
6.1.	Subtask 2.5.1 Traffic management & condition assessment	43
6.2.	Subtask 2.5.2 V2X-based support of transportation services	48
	Use Cases: Definition and Objectives	49
	Conceptual Architecture of Smart Mobility Hubs	50
7.	Task 2.6 Manage & Control: Electric Vehicle (re-) Scheduling, Low emission-based traffic management, Smart parking	56
7.1.	Subtask 2.6.1 Electric Vehicle Scheduling	56
	Electric Bus Charging Station Location Selection Problem with Slow and Fast Charging	56
	Electric Bus Charging Station Location Model considering Travel Time and Energy Consumption Uncertainties	60
	Combined Line Planning Modification and Vehicle Scheduling for a fleet of Electric Buses	65
	Vehicle Scheduling component of the Electric Bus Multi-Depot Vehicle Scheduling Problem with Time Windows	71
	Analyzing the Effects of Pick-Up and Drop-Off (PUDO) Duration and its Stochasticity on Mobility-On-Demand Services	72
	Optimal Dynamic Curb-side Management for mobility-on-demand services	77
	Multimodal Waste Collection Logistics	83
7.2.	Subtask 2.6.2 Low emission-based traffic management	85
	Predictive modeling with multi-source data	86
	TMC (Tradable Mobility Credits) assignment method for multiple population segments	91
7.3.	Subtask 2.6.3 Supply-demand matching platform for on-demand shared zero-emission services	95
8.	Task 2.7 Integrate + Plan: Digital Twin platform for optimisation	100
9.	Conclusion	105
	References	107
	Annex	115
	Annex sub-chapter for Chapter 2 and Task 2.1	115
	Annex sub-chapter for Chapter 3 and Task 2.2	116
	Annex sub-chapter for Chapter 4 and Task 2.3	117
	Annex sub-chapter for Chapter 5 and Task 2.4	118
	Annex sub-chapter for Chapter 6 and Task 2.5	123
	Annex sub-chapter for Chapter 7 and Task 2.6	124
	Annex sub-chapter for Chapter 8 and Task 2.7	134

Table of Tables

<i>Table 1: The results of the application of the bi-level model to Limassol, Cyprus case study</i>	<i>15</i>
<i>Table 2: Optimal charging station physical location (V), and charging types ($N=N1 \cup N2$) across the deterministic and all $\beta(\%)$ variation approaches</i>	<i>64</i>
<i>Table 3: 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>64</i>
<i>Table 4: Best solution of the combined line planning and vehicle scheduling problem for a fleet of 52 electric buses</i>	<i>69</i>
<i>Table 5: Overall Performance Comparison</i>	<i>88</i>
<i>Table 6: Mode-specific Metrics.....</i>	<i>89</i>
<i>Table 7: Resulting Clusters</i>	<i>94</i>
<i>Table 8: Pricing policies</i>	<i>99</i>
<i>Table 9: Sensors Listing</i>	<i>120</i>
<i>Table 10: Lidar Specifications</i>	<i>120</i>
<i>Table 11: Design (555 scenarios)</i>	<i>127</i>

Table of Figures

Figure 1: Limassol case study (a) bus network, including eight bus lines operating within the limits of the urban environment, and (b) candidate charger locations and the end stop for each bus line service.....	10
Figure 2: Pareto optimal solutions derived for the Limassol case study according to the two objectives	11
Figure 3: Comparison of Single-Level formulations (initial model) versus Bi-Level models (extended model during M19 to M26).	12
Figure 4: The candidate charging station locations of the set N as blue dots and the ending locations of bus lines for the Limassol case study.	14
Figure 5: seq2seq model diagram.....	17
Figure 6: Conceptual framework architecture	19
Figure 7: Charging components for Inductive charging (Source: Sawilla & Schuett, 2017)	22
Figure 8: Automated charging system architecture.....	23
Figure 9: ROC flowchart.....	27
Figure 10: Distributed Architecture and Resilience Mechanisms	29
Figure 11: Simulated Navigation.....	31
Figure 12: Next pod in Milan	33
Figure 13: Schematic representation of a docking manoeuvre	34
Figure 14: Problem Statement	35
Figure 15: Trend of the variable Relative Speed at Docking with respect to the input variables Leader Speed (a), Delta V (b), and Delta Distance (c).....	37
Figure 16: GSA indexes	37
Figure 17: Case Study Network (both directions)	41
Figure 18: Passenger load variations when using modular autonomous vehicles with short-turning options and when implementing the as-is scenario with conventional vehicles.....	42
Figure 19: Identifying occlusion areas from trajectories of pedestrians.....	48
Figure 20: Producer Layer multi-source data acquisition ecosystem with eight primary data streams .	51
Figure 21: Technical architecture of V2X Communication System.....	53
Figure 22: Treatment Layer preprocessing: validation, exploratory analysis, and spatial-temporal aggregation.....	54
Figure 23: Treatment Layer ST-GAT predictive modeling framework for multimodal demand forecasting	54
Figure 24: Consumer Layer service delivery pipeline from model outputs to end-user applications.....	55
Figure 25: 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.	59
Figure 26: A generic representation of the time-slice modeling 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.	68
Figure 27: A flowchart of the two-stage process including the E-VSP and LPM models	69
Figure 28: 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.....	70
Figure 29: Impacts of increasing PUDO duration considering fully-informed operators.....	75
Figure 30: Effects of Predicting PUDO Duration for Each Request.....	76
Figure 31: Overview of our proposed solution approach of the CA-PUDO-NDP based on the ALNS metaheuristic.....	79
Figure 32: Pareto frontier between Ncity and the other objectives.	80
Figure 33: Link speed distribution by scenario. SQ: Status Quo; LB: Lower Bound.....	81

Figure 34: PUDO policies and link speeds for 7.5% AMOD demand under different Ncity. (DP: Double Parking.)	82
Figure 35: Limassol Speed Loop.....	90
Figure 36: Paris Speed Loop	90
Figure 37: Score of each generation	94
Figure 38: Decision Tree Interpretation.....	95
Figure 39: User app interface	96
Figure 40: Operator and driver app interface.....	96
Figure 41: School, home, activity locations selected in Limassol.....	98
Figure 42: Results for a fleet size of 10 Minibuses with 17-seat capacity	98
Figure 43: ECDF for the total daily student fare	100
Figure 44: Digital Twin Amsterdam: Demographic data visualizations.	101
Figure 45: Digital Twin Amsterdam: Waste logistics location information.....	102
Figure 46: Selected views of the Limassol Digital Twin interface	103
Figure 47: Initial draft of LUTI model flow patterns illustrating zone-to-zone interactions in the Limassol area	104
Figure 48: The conceptual framework architecture	115
Figure 49: Example of LSTM Unit	115
Figure 50: Wind, Distributed, Conventional Generation, and Total Power Demand (Sourced from Cyprus TSOC)	116
Figure 51 Forecasting model results for 6h, 12h and 24h prediction horizons	116
Figure 52: IoT Module using an ESP32 microprocessor	116
Figure 53: Screenshot of received IoT data in Data Lake.....	117
Figure 54: Screenshot of received static data in Data Lake.....	117
Figure 55: Phase 0 table/lab tests on-going with the University of Applied Science Lapland. This robot hand will be installed to Remoted depot in phase 1	117
Figure 56: Charging process steps	118
Figure 57: RCC Functional Block Diagram - Key Interactions.....	118
Figure 58: Sensors Placement on our prototype.....	119
Figure 59: Sensors Coverage bird view from the real robot	120
Figure 60: Next Modular Vehicles' Third Generation Pod.....	121
Figure 61: Standard Sensor Coverage for AMVs. Source: Wireless Communication and Sensors in Self-Driving Cars - Pisarov, Jelena & Mester, Gyula. (2020). The Future of Autonomous Vehicles. FME Transactions.	121
Figure 62: Graphical representation of the Minimum Safety Distance (MSD)	122
Figure 63: O-D Pair of Case Study.....	122
Figure 64: Flowchart of Traffic Management Framework - From Sensors to Services: A Scalable Pipeline for Predictive Traffic Management.....	123
Figure 65: Athens network of the electric bus lines, along with the proposed charging station physical locations.....	124
Figure 66: The three bus lines examined under this case study	124
Figure 67: A generic representation of the modelled problem	125
Figure 68: PDFs of Log-normal distributions with pivoting mean and constant variance	127
Figure 69: Operational impact of using different PUDO duration prediction models	128
Figure 70: Impact on the Absolute Drop-Off time Deviation	128
Figure 71: Impact on the Absolute Drop-Off time Deviation	128
Figure 72: Operational impacts using different Constant Insertion and Prediction Models	129
Figure 73: Impact on the Absolute Drop-Off time Deviation	129
Figure 74: Impact on the Late Drop-Off share	129
Figure 75: Branch-and-Price Schema.....	133

<i>Figure 76: Tree Representation</i>	<i>133</i>
<i>Figure 77: Cluster Separation</i>	<i>134</i>

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
LPM	Line Planning Modification
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
OASA	Athens Urban Transport Organisation
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

SAA	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
TRL	Technology Readiness Level
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-TD TT	Two-Echelon Vehicle Routing Problem with Time Windows and Time-Dependent Travel Times

Background: About the metaCCAZE project

metaCCAZE 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 metaCCAZE 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 – metaCCAZE 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.

MetaCCAZE 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 presents the innovations' adjustments and further developments that happened after implementing, testing and started demonstrating them in WP3, related to T2.1 to T2.7 of Work Package 2 (WP2) of the metaCCAZE project during the months (M19–M26). 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 methodological approach of this deliverable is to present the scientific contributions developed for each task according to the UCs and Assets of the MetaCCAZE 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 D2.2 indicate how metaCCAZE's smart technologies can create sustainable, zero-emission transport for passengers and freight. Some key 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 metaCCAZE aiming for smarter transportation, with reduced environmental footprint.

The work described in this deliverable is aligned with metaCCAZE'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, D2.2 presents the adjustments and further development of metaCCAZE's metaInnovations, which will be updated in M26 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 refinements and developments on open innovations developed during D2.2 in WP2. Specifically, D2.2 will document refinements and developments of the metaInnovations following their implementation and testing for the Use Cases (UCs) in the Trailblazer 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 metaCCAZE framework, the metaInnovations Toolkit showcases the project's cutting-edge technological advancements across seven key domains, which form the backbone of metaCCAZE:

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. Digital twin optimisation.

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 (UCs), along with collaborative business and governance models. This toolkit represents the second objective (O2) of the metaCCAZE 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 adjustments and further development carried out across these Tasks from July 2025 to January 2026, offering a comprehensive overview of the technological innovations developed up to this point.

More specifically, notable milestones achieved during this period include the adjustments and further development 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 adjustments and further 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 had adjustments and further development for traffic management purposes, and supply-demand matching tools for on-demand shared electric services have been finalized and are currently in use 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 D2.2 is structured as follows.

Chapter 2 presents the adjustments and further development 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 adjustments and further development 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 the adjustments and further development 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 adjustments and further development 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 adjustments and further development 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 adjustments and further development 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 adjustments and further development 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 D2.2 lists the review work and technical progress conducted up to now under Task 2.7.

Chapter 9 concludes Deliverable D2.2 by providing a synthesis of the technical progress and development activities carried out across all tasks within Work Package 2. The chapter summarises the main findings from Tasks 2.1 through 2.7, highlighting the advancements made in AI-based optimisation, data infrastructure, automated charging, autonomous vehicle systems, V2X connectivity, zero-emission fleet management, and digital twin integration.

1.3. MetaInnovations to Living Labs Map

To further supplement this introductory material, this subsection provides a consolidated colour-coded map of the WP2 metaInnovations and their expected application across the metaCCAZE Living Labs. The table links each metaInnovation with its owner, the project subtask responsible for its development, the Living Lab(s) where it is expected to be applied, and the corresponding Use Case(s). This mapping is intended to make the connection between the technical developments of WP2 and their implementation context in WP3 and WP4 explicit from the beginning of the deliverable.

The MetaInnovations are divided into seven broader thematic categories: (1) electrification, including charging, grid-fleet-demand optimisation, and electric vehicle scheduling; (2) automation and connectivity, covering automated charging, remote control, advanced driver assistance, autonomous fleet scheduling, traffic condition assessment, and V2X-based support; (3) curbside and logistics solutions; (4) demand and supply management, including mobility credits and supply-demand matching; and (5) horizontal MetaInnovations, such as the AI data warehouse, APIs, and the Digital Twin platform, which support multiple technical developments across WP2.

	MetaInnovation	Owner	Responsible Project Subtask	Living Lab(s)	Use Case(s) within the Living Lab(s)	MetaInnovation Category
1	Decentralised AI optimisation for grid-fleet-demand	NTUA	ST2.1.1 Decentralised AI optimisation for grid-fleet-demand	Limassol	Limassol UC04	Electrification
2	AI recommendation and incentivization engine	MLAB	ST2.1.2 AI recommendation and incentivisation engine	Limassol	Limassol UC04	Electrification
3	AI-Data warehouse and APIs	MLAB	Task 2.2: ST2.2.1 AI-Data Models and Warehouse + ST2.2.2 APIs and Connectors	Limassol	Limassol UC04	Horizontal Technical Innovations
4	Automated charging & V2G protocols	REMOTED	Task 2.3	Tampere	Tampere UC01 & UC02	Automation and connectivity
5	Remote control center for AVs	REMOTED	ST2.4.1 Remote control center for AVs	Tampere	Tampere UC01 & UC02	Automation and connectivity
6	Advanced Driver Assistance System	NEXT	ST2.4.2 Advanced Driver Assistance System	Milan	Milan UC01	Automation and connectivity
7	AI-based real-time planning of autonomous fleets	NTUA	ST2.4.3 AI-based real-time planning of autonomous fleets	Milan	Milan UC01	Automation and connectivity

8	Traffic management & condition assessment	FLOWARE	ST2.5.1 Traffic management & condition assessment	Paris-Poissy	Poissy UC01	Automation and connectivity
9	V2X-based support of transportation services	VEDECOM	ST2.5.2 V2X-based support of transportation services	Paris-Poissy	Poissy UC01	Automation and connectivity
10	Electric Vehicle Scheduling	NTUA	ST2.6.1 Electric Vehicle Scheduling	Athens; Limassol	Athens UC01 & UC02; Limassol UC04	Electrification
11	Low emission-based traffic management	VEDECOM	ST2.6.2 Low emission-based traffic management	Paris-Poissy; Krakow	Poissy UC01; Krakow UC01	Electrification
12	Mobility Credits	TUD	ST2.6.2 Low emission-based traffic management — TMC assignment method	Amsterdam	Amsterdam UC04	Demand and supply management
13	Supply-demand matching platform for on-demand shared zero-emission services	MLAB	ST2.6.3 Supply-demand matching platform for on-demand shared zero-emission services	Limassol	Limassol UC01	Demand and supply management
14	Reverse logistics	TUD	ST2.6.1 Reverse logistics routing and Scheduling	Amsterdam	Amsterdam UC03	Curb-side management and logistics
15	Curb-side Management	TUM	ST2.6.1 Electric Vehicle Scheduling	Munich	Munich UC01	Curb-side management and logistics
16	Digital Twins	MLAB	Task 2.7 Integrate + Plan: Digital Twin platform for optimisation	Limassol; Amsterdam	Limassol UC04; Amsterdam UC03	Horizontal Technical Innovations

Overall, the map shows that the WP2 metaInnovations are not developed as isolated technical components, but as tools connected to specific Living Lab applications and Use Cases. It also highlights the cross-cutting role of the horizontal metaInnovations, which provide common data, integration, planning, and simulation capabilities for the wider WP2 toolkit. The detailed description of each metaInnovation, including its methodological background, technical progress, and implementation status, is provided in the following task-specific chapters.

1.4. Relation to Project Documents

Deliverable D2.2 is the second report produced under Work Package 2 (WP2) and is directly linked to the adjustment and further development updates submitted by WP2 partners in the project's second periodic report. As such, D2.2 builds upon the progress reported by the various WP2 task teams during the initial reporting period.

At a secondary level, D2.2 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 metaInnovations 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 and D2.2

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 D2.2 and adjustments and further development of D2.2. 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 and subsequently D2.2, 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 D2.2 also plays a role in shaping upcoming project deliverables. Most directly, it informs the future outputs of WP2, namely D2.3, which will present the finalized versions of these metaInnovations, as informed by their application in the WP3 and WP4 LLs.

Beyond WP2, D2.1 and subsequently D2.2 are 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 metaInnovations described herein.

Finally, the metaInnovations, as documented in D2.2, and further detailed and refined 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.

1.5. 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 MetaCCAZE 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 operations research and artificial intelligence models that optimise and coordinate energy distribution, bus fleet energy consumption, and charging demand. The aim is to lower the net carbon intensity of electricity used to charge electric vehicles, with a particular focus on electric bus fleets. These models, designed specifically for transportation planning, operate across strategic, tactical, and operational time horizons. Using optimisation, simulation, and various digital tools, they integrate charging-schedule flexibility, energy flows for bus-fleet charging, and relevant grid constraints to enable “low-carbon decisions” within the combined grid, fleet, and demand ecosystem.

Task 2.1 defines, designs and contributes to two core metaCCAZE technical assets: Technical Asset 1, the Multi-Depot Electric Vehicle Scheduling Problem (MD-E-VSP), and Technical Asset 13, the Moby App. Within WP2, this task directly supports the project’s second objective (O2), which centers on building the metaInnovations toolkit. Task 2.1 consists of Subtask 2.1.1, “Decentralised AI Optimisation for Grid-Fleet-Demand,” and Subtask 2.1.2, “AI Recommendation and Incentivisation Engine.”

This paragraph summarises the progress of Task 2.1 during Months M19 to M26, corresponding to the period following the submission of the preceding WP2 deliverable, D2.1. For Subtask 2.1.1, led by the NTUA team, the previously developed transport planning model has been extended to incorporate the perspective of an additional stakeholder. The initial model, presented in D2.1, was a bi-objective Mixed Integer Linear Programming (MILP) formulation that reflected the objectives of the electric bus operator when planning a charging station network for an electric bus fleet. Beginning in M19, this planning model was expanded to include parameters and decision variables associated with the energy management authority. The resulting bi-level MILP model now generates optimal solutions for charging network planning while accounting for both stakeholder perspectives.

For Subtask 2.1.2, led by MaaSLab (MLAB), efforts have focused on developing an AI-driven recommendation engine that identifies optimal charging times based on renewable energy availability, using datasets obtained from the Electricity Authority of Cyprus (EAC). Since the submission of D2.1, the subtask has completed the acquisition of all required datasets for the WeePlatform, including pricing data from EAC, electric bus operational data from the bus operator, and micromobility data from NextBike. The conceptual architecture has been finalised and facilitates the collection of charging profiles and monthly energy costs for electric buses. These datasets are now being prepared for forecasting applications that will inform recommendations on optimal charging schedules and potential opportunities for feeding energy back into the grid.

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

As part of Subtask 2.1.1, the project team has concentrated on developing two transport network planning models that support the design of sustainable and efficient electric bus networks. As already discussed in the previous WP2 deliverable, D2.1, a key aim of these models is the modelling of real-world public transport operations (using General Transit Feed Specification data) from the Limassol Buses Company (EMEL), in combination with the main parameters that define how the energy grid interacts with public transport service providers.

The research team of Subtask 2.1.1 attempts to provide a holistic perspective that ensures that network planning is not carried out in isolation, but instead aligns decisions across the supply, fleet and demand dimensions. In that regard, within Subtask 2.1.1, the work is structured around two primary research directions:

1. **Grid-Fleet-Demand Management:** This direction focuses on optimizing decisions made by the public transport services operator (EMEL), considering the parameters that shape the operation of the energy grid.
2. **Grid Management:** This direction aims to optimize decisions made by the energy management authority (such as energy pricing schemes), while accounting for the energy requirements of the bus fleet.

The following sections present the developments achieved under Subtask 2.1.1 during the first 26 months of the metaCCAZE project.

The Grid-Fleet-Demand Management

As detailed in Deliverable D2.1, covering the M1–M18 project period, the work undertaken under the first research direction resulted in the development of a novel bi-objective Mixed-Integer Linear Programming (MILP) model to support strategic planning of electric-bus charging infrastructure. Within the Grid–Fleet–Demand Management research stream, the proposed model assists public-transport operators (e.g., EMEL) in determining charger siting and bus charging schedules by explicitly balancing two competing aims: (i) minimizing total cost, including both infrastructure investment and day-to-day operating expenditures, and (ii) minimizing bus deadhead time. Accordingly, the multi-objective formulation derives planning decisions for public-transport service operators with respect to two primary 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 provide decision-makers with operationally relevant guidance, the project team applied the ϵ -constraint method to generate a set of Pareto-optimal solutions, thereby enabling stakeholders to

systematically examine trade-offs between monetary cost and operational performance. The formulation was evaluated using data from the Limassol Living Lab, demonstrating both the model's adaptability and its practical applicability in a real-world setting. Overall, the proposed framework offers urban planners and public-transport authorities a structured basis for supporting sustainable, cost-effective transitions toward electric bus operations.

The following subsections present the model developed under the Grid-Fleet-Demand Management research direction in detail. The exposition begins with the MILP objective functions, proceeds to the full set of operational and infrastructure constraints, and concludes with the resulting solution outcomes and their interpretation.

Objective functions

First, within this research direction, the model is formulated around two primary objectives for the management of an electric bus charging network. The first objective is to minimize deadhead time, defined as non-revenue travel between route terminals and charging facilities, which incurs additional energy consumption and driver time. Reducing deadhead improves overall fleet utilization by increasing productive service time and limiting inefficient vehicle repositioning, thereby supporting more cost-effective and environmentally sustainable operations. The corresponding objective function is provided 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 $k \in \mathcal{K}$, where $\mathcal{K}$ 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 the primary objective of the MILP formulation, a second objective is introduced to incentivize the minimization of the total monetary cost associated with both the expansion and operation of the charging infrastructure. This objective encompasses: (i) capital expenditures for installing new chargers, (ii) electricity procurement costs determined under time-of-use (ToU) tariffs, and (iii) peak demand charges imposed by the energy provider based on the maximum power drawn within the relevant billing periods. The mathematical expressions defining this second objective are provided in Equations (2) - (5).

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

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

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

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

The second objective function, presented in Equation (5), represents the total monetary cost and is decomposed into three constituent terms. Equation (2) captures the Time-of-Use Charging Cost (TOUC), computed as the sum over all time periods of the energy consumed (kWh) multiplied by the corresponding time-dependent electricity tariff. Equation (3) defines the Demand Charge Cost (DCC), which reflects the maximum peak power demand (kW) observed within the billing horizon, multiplied by the demand-charge rate set by the energy provider. Equation (4) quantifies the Charging Station Installation Cost (CSI) as the aggregate capital cost of deploying new chargers, obtained by summing the fixed installation cost associated with each charger selected for installation.

If one pays close attention to mathematical expressions (1) and (5) closely, they will probably notice that these two objectives, O_1 and O_2 often conflict. This means that locating chargers to minimize deadhead time can necessitate a larger number of installations and/or placement at sites that are suboptimal from a grid-integration perspective, thereby increasing both capital and operating expenditures. Conversely, a strict focus on minimizing monetary cost may favour a smaller set of grid-favourable locations, which can increase the distance and time buses must travel to access charging services. To manage this inherent tension, the proposed framework adopts a bi-objective optimization formulation that enables systematic exploration of Pareto-optimal alternatives, each representing a distinct prioritization of the competing objectives and the associated trade-offs. By explicitly co-optimizing operational performance and economic cost, the model captures key dimensions of electric fleet planning and supports the design of charging networks that are both operationally viable and financially efficient. In addition, the formulation incorporates constraints that reflect real-world bus operations and the coupling between transit operator decisions and the tariff and pricing mechanisms imposed by the energy-management authority (as discussed in Rizopoulos and Gkiotsalitis (2025)).

To solve this formulation, we employ the ϵ -constraint method, a classical technique for generating Pareto-optimal solutions in bi-objective optimization. In this framework, one objective (the “primary” objective) is optimized directly, while the second objective is reformulated as a constraint and bounded by a parametric threshold ϵ (epsilon). By systematically varying parameter ϵ across a prescribed range of feasible values, this procedure generates a family of solutions that explicitly capture the trade-off between the two objectives. For each fixed ϵ level, the ϵ -constraint modifies the feasible set, thereby inducing a distinct optimal solution. Aggregating the resulting solutions yields an approximation of the full Pareto frontier. This approach enables decision-makers to systematically examine a spectrum of efficient trade-offs and to identify the alternative that best aligns with their operational preferences and constraints. Accordingly, the original formulation is reformulated as: *minimize O_2 , subject to $O_1 \leq \epsilon_m$.*

Model application to Limassol Living Lab

Within the broader evaluation of real-world charging-infrastructure planning strategies, the Limassol Living Lab provides a practical testbed for implementing the proposed bi-objective optimization model. Limassol, a coastal city with an expanding public-transport network, reflects many of the operational and planning complexities encountered by urban systems transitioning to electric bus fleets. Spatial and geographic characteristics, distribution-grid capacity limitations, and heterogeneous travel-demand patterns collectively increase the realism of the infrastructure planning problem. By representing these conditions explicitly, the Limassol Living Lab enables a rigorous assessment of the trade-offs between charging-network expansion decisions and downstream operational performance.

Model implementation integrated spatial data describing candidate charging locations with detailed inputs on bus routes, operating timetables, and the city's time-of-use electricity tariff structure. The formulation further incorporated site-specific and utility-driven constraints, including upper bounds on the number of chargers per location and applicable demand-charge mechanisms. This setting therefore supports the systematic evaluation of alternative charging strategies within a single optimization framework, spanning combinations of slow and fast chargers as well as operational policies such as partial recharging. In doing so, the Limassol Living Lab allows the robustness and flexibility of the proposed model to be examined under realistic conditions and yields actionable insights into how operators may expand charging infrastructure while managing energy-related costs and reducing deadhead time. The bus network considered in

the Limassol Living Lab is illustrated in Figure 1(a), while the candidate charging locations are shown in Figure 1(b).

Applying the bi-objective model to the Limassol Living Lab indicates that there is no uniquely optimal strategy for expanding the electric-bus charging network. Instead, the optimization identifies a set of Pareto-optimal solutions, each embodying a distinct compromise between minimizing total monetary cost and reducing bus deadhead time. Rather than forcing these competing objectives into a single aggregated outcome, the Pareto framework makes the underlying trade space explicit: greater investment in strategically sited charging infrastructure can mitigate operational inefficiencies (via reduced non-revenue travel) but typically increases capital requirements and energy-related operating expenditures, and conversely, cost-minimizing designs tend to tolerate higher deadhead burdens. These inherent tensions reflect the legitimacy of both objectives and motivate alternative configurations of slow and fast chargers that achieve different degrees of performance across the two criteria.

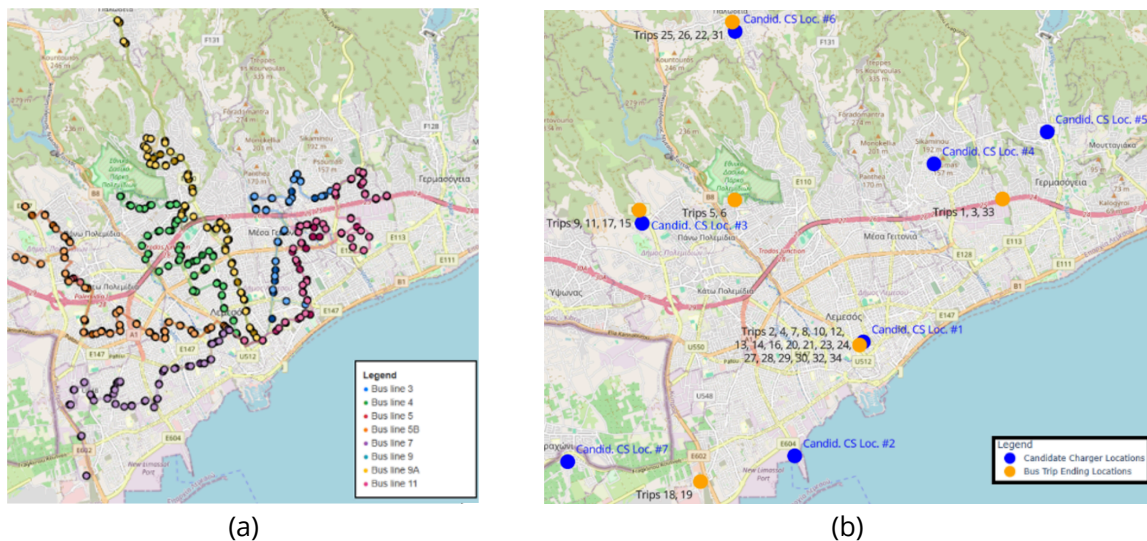


Figure 1: Limassol case study (a) bus network, including eight bus lines operating within the limits of the urban environment, and (b) candidate charger locations and the end stop for each bus line service.

Enumerating the Pareto-optimal set provides decision-relevant transparency for stakeholders, including transit authorities and policy makers, by clarifying how marginal improvements in one objective (e.g., lower system cost) are associated with corresponding degradations in the other (e.g., increased deadhead time and potentially longer non-revenue route extensions). In the Limassol Living Lab setting, this explicit characterization supports more informed and context-sensitive planning, enabling authorities to select solutions consistent with organizational priorities or public-policy aims, whether these emphasize cost-effectiveness, operational efficiency, or a balanced compromise. Figure 2 presents the Pareto-optimal solutions obtained for the Limassol charging-network design problem.

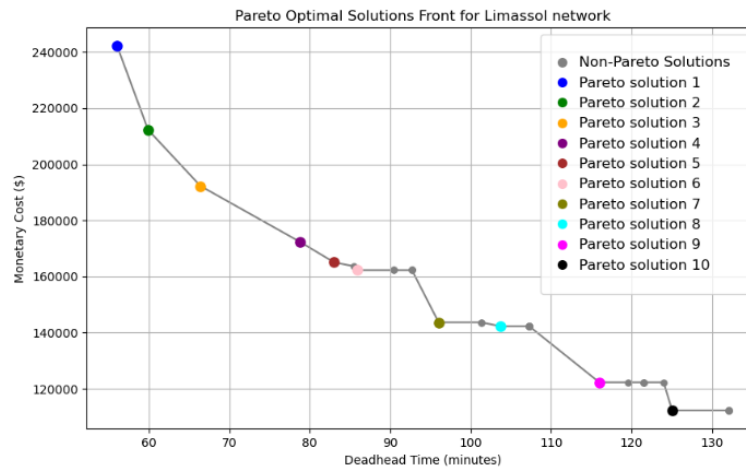


Figure 2: Pareto optimal solutions derived for the Limassol case study according to the two objectives. These ten Pareto-optimal solutions, distinguished by colour, correspond to ten distinct, implementable configurations of charger deployment and the associated charging schedules that assign buses to available time slots. For the Limassol Living Lab, Pareto Solutions 1 and 10 delineate opposite extremes of the cost–deadhead trade-off frontier, each reflecting a different operational and investment emphasis. Pareto Solution 1 minimizes total monetary expenditure related to charging-network installation and operation, albeit at the expense of increased bus deadhead time. Accordingly, it is well-suited to stakeholders for whom capital and operating budget constraints outweigh the marginal gains associated with reduced non-revenue vehicle movements. In contrast, Pareto Solution 10 entails a comparatively denser and more strategically sited charger network, yielding substantial reductions in electric bus deadhead travel, but requiring higher capital investment and elevated energy-related operating costs. This configuration is therefore more appropriate for operators seeking to prioritize operational efficiency and the minimization of non-revenue travel, even under higher infrastructure and electricity expenditures.

Intermediate solutions provide additional feasible trade-offs between these endpoints. Pareto Solution 5 constitutes a representative compromise within the Limassol Living Lab, combining a moderate level of investment in new charging infrastructure with a marked reduction in deadhead time. While the deadhead reductions are less pronounced than those achieved by the most infrastructure-intensive solutions, both installation and operational costs remain substantially below the levels observed in the highest-investment configurations. This balance is particularly advantageous for operators aiming to reconcile fiscal limitations with meaningful improvements in operational performance. Consequently, Solution 5 demonstrates the model's capacity to generate decision-relevant alternatives that support informed planning under competing objectives.

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” and has also been presented at the *hEART 2025* conference as a podium presentation.

Grid Management

Moving on to the second research direction of Subtask 2.1.1, it has been the main focus of research activity from Month 19 to Month 26 of the project within this Subtask. Building upon the previous work with the Grid-Fleet-Management direction, the research team has aimed at extending the modelling scope and including the perspective of the energy authority in the planning models. In more detail, while the model developed up to M18 and Deliverable D2.1 included only the perspective of the bus authority in Limassol (i.e. EMEL), the model also been extended to include the perspective of the energy authority. This has been achieved by mainly methodologically expanding the previous model. While the previous model included a bi-objective MILP model, in this second iteration the research team has adopted a bi-level mathematical programming approach.

Bi-level mathematical programs are two-level optimization models arranged in two hierarchical layers, where the upper-level decision maker (or stakeholder) sets choices that the lower-level decision maker must respond to optimally. In this context, the upper level represents the bus operator as the leader, while the lower level represents the energy authority as the follower reacting to those decisions. This structure is especially useful for modelling systems where decision-makers have interconnected objectives but operate with different priorities and levels of control. It does not only capture the different priorities (like a single-level bi-objective model does), but it can also derive the mathematical optimal one when different constraints are considered for the two levels of decision-makers (or stakeholders). This is an extra capability of bi-level models, when compared single-level models that only capture a single set of constraints. In Figure 3, this difference is highlighted.

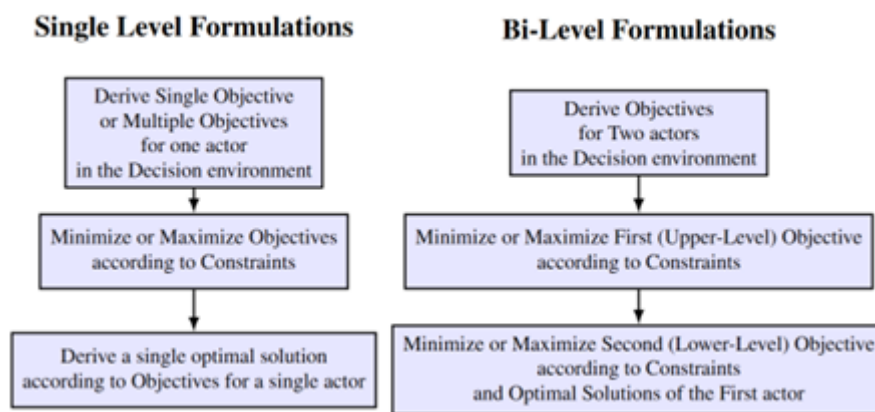


Figure 3: Comparison of Single-Level formulations (initial model) versus Bi-Level models (extended model during M19 to M26).

Objectives

In the model developed in during the period M19 up to M26 of the metaCCAZE project, Subtask 2.1.1 team has focused on solving the Charging Stations Location Problem for Electric Buses (EB-CSLP) when integrating both Objectives and Constraints of two stakeholders: the Electric Bus operator and the Energy or Energy Grid Management Authority.

These two stakeholders have different perspective on the same problem (EB-CSLP) and their objectives can be contradictory. For example, as it obvious, the energy management authority may

want to maximize its profit, which is contradictory with the objective of the buses operator who wants to minimize charging costs. This leads to a problem in the EB-CSLP, where the price that the energy authority charges for each kilowatt-hour (kWh) during the day, can lead to different charging scheduling for the electric bus fleet operator. If the price is too high, then a sub-optimal charging scheduling may emerge for the buses which can in turn ruin the optimal setup of chargers in the problem.

To solve this problem, the project team considers a bi-level MILP approach, where in the upper level, the planning problem is considered from the perspective of the bus operator. For that upper-level planning problem we consider the following objective function:

$$\min ds \cdot (DHC + TOUCs + TOUCH) + CSI \quad (6)$$

In objective function (6), the electric bus operator is seeking to minimize the following quantities:

- DHC Which stands for the deadheading cost of the buses, that emerges from the distance they cover from the last stops of their trips to the charger locations,
- $TOUC^s$ which stands for the total daily charging cost that comes from slow chargers,
- $TOUC^h$ which stands for the total daily charging cost that comes from fast chargers,
- CSI Which stands for the total up-front charger installation cost.
- ds Which represents the number of days considered in the modelling.

So overall, as it may be noticed from this objective function the electric bus operator is mainly seeking to provide the public transport services with the minimal costs that concern both the strategic part of installing chargers (CSI) but also making sure that the daily charging costs remain as low as possible. On the other side, the energy management authority has a totally different objective which contradicts objective function (6) presented above. The objective function for the energy management authority is given in mathematical expression (7):

$$\max \sum_{f_3}^{F_3} RPP_{f_3} \cdot EAREv_{f_3} - \sum_{f_3}^{F_3} PCP \cdot PC_{f_3} \quad (7)$$

In objective function (7), the energy management authority seems to maximize profits from renewable energy sources, while trying to not change an initial price scheme for its Time-of-Use (ToU) Tariffs by a substantial amount. Here is a more detailed view on the terms and quantities of the objective function:

- RPP_{f_3} is the Renewable Source Participation per time period f_3 ,
- $EAREv_{f_3}$ is the revenue that the energy authority has based on the price ToU_{f_3} that it uses and the charging scheduling that electric bus operators decide,
- PCP Is the cost per percentage change of the initial ToU tariffs considered by the authority,
- PC_{f_3} Is the percentage change that the energy authority decides based on its initial ToU Tariffs in order to derive an optimal final ToU tariffs.

One may notice from this objective function (7) that the energy authority does not just maximize profit but it has to also consider how much the price changes according to a penalty PCP that is applied. That penalty can be part of a Business-to-Business (B2B) contract with the energy operator or it can be perceived as a penalty that is applied due to the dissatisfaction of the other stakeholder of the problem. To understand the problem better, it is also important to mention that the energy authority is also seeking to keep its overall profits from renewable and non-renewable sources above a specific threshold. This is why Constraint (8) is introduced in the lower-level model (that of the energy authority):

$$EAREv \geq EAREv_{min} \quad (8)$$

Where $EAREv$ is a variable of the problem and is the profit of the energy authority, calculated based on the decision variable for ToU Tariffs and $EAREv_{min}$ is a parameter of the problem and represents the lower overall profit that the energy management authority is willing to accept.

Model application to Limassol Living Lab

The aforementioned model has been tested on real-world data from the Limassol LL. In more detail an eight-bus line network has been examined with Bus lines 3, 4, 5, 5B, 7, 9, 9A and 11. Seven candidate charger locations are considered. Both can be viewed in Figure 4.

The bi-level model has been tested for its ability to provide cooperative solutions, meaning solutions where the electric bus charging network is set-up in such a way that is optimal for both stakeholders. To achieve that the model had to be solved in an exact mathematical way, meaning that a global optimal solution to the modelling had to be achieved.

To calculate such a solution, the subtask research team has adopted a solution approach based on the Karush-Kuhn-Tucker conditions and the reformulation of the bi-level model to a single-level mathematical model with complementarity constraints (MPCC). This results in a substantial easier problem to solve which still holds the properties of the bi-level formulation.

To evaluate the contributions of the proposed Bi-Level model, this case study investigates and compares three planning models: (1) the standalone upper-level problem which resembles a non-cooperative framework where the upper-level formulation is solved without considering the lower-level. This resembles solving the problem from the perspective of the electric bus fleet operator without considering the perspective of the energy authority. The ToU tariffs are constant in this scenario, and it is not part of a negotiation between the two stakeholders; (2) the bi-level model solved by a heuristic nested algorithm; and (3) the bi-level model solved by the exact KKT reformulation to a single-level formulation as an MPCC (solved to global optimality).

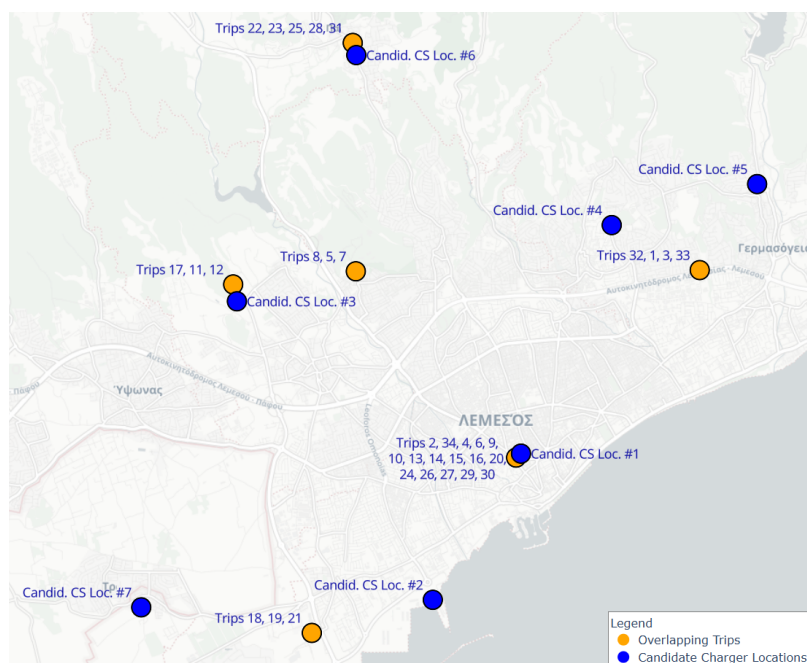


Figure 4: The candidate charging station locations of the set N as blue dots and the ending locations of bus lines for the Limassol case study.

Across all three planning models, the resulting infrastructure designs are largely consistent in terms of the number, type, and placement of chargers. The major variation appears in the optimal charging schedules for the thirty-four electric buses, showing that jointly optimizing the Time-of-Use tariffs has a strong impact on the operator's daily operational decisions. Table 1 compares the aggregated outcomes across the three scenarios, examining the operator's total cost, the energy authority's profit, daily renewable energy usage, and daily charging costs to illustrate the trade-offs introduced by each modelling framework. The upper-level solution with fixed ToU tariffs performs worst for the bus operator but yields the highest profit for the energy authority, and it also results in the lowest daily renewables consumption. The nested bi-level algorithm produces the most favourable outcome for the bus operator, although it weakens the objective of the energy authority. The exact bi-level solution, obtained through the KKT-based MPCC reformulation, achieves a notably more balanced outcome, improving results for both stakeholders.

These findings indicate that the exact bi-level approach delivers the most robust and mutually beneficial solution, successfully capturing the interactions and objectives of both decision-makers. The resulting insights provide valuable decision-support material for the Limassol Living Lab, offering stakeholders a clearer understanding of how infrastructure choices, charging strategies, and tariff structures interact when shaping an electric bus charging network. Although the model already operates on real-world data, the implications presented here can naturally evolve as additional information or constraints become available. Even so, the framework remains technically solid: it is fully implemented in Python with contemporary optimization libraries, and it can be calibrated to explore alternative planning scenarios or stress-test strategic assumptions relevant to the Living Lab. Finally, it is important to note at this point that the model developed under the second research directing of subtask 2.1.1 is an important scientific contribution. This means that the research conducted has been described in a research paper which is currently under first stage review at an accredited peer-reviewed journal with title "Computers and Industrial Engineering". A short paper version of this research article has also been accepted and presented as a poster at the Transport Research Board Annual Meeting for 2026.

Table 1: The results of the application of the bi-level model to Limassol, Cyprus case study

	COST FOR BUS OPERATOR (\$/YEAR)	PROFIT FOR ENERGY AUTHORITY (\$/DAY)	DAILY RENEWABLES CONSUMPTION (KWH)	DAILY CHARGING COSTS (\$)
Upper-level solution	1,377,322.74	529.25	2,709.25	24.31
Nested Algorithm: Bi-level model	1,360,071.23	387.8	2,831.08	14.86
Exact solution via KKT:Bi-level model	1,364,125.02	525.17	2,725.9	17.08

2.2. Subtask 2.1.2 AI recommendation and incentivization 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 metaCCAZE project.

Sustainable Charging Recommendation Engine

This subtask involves 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.

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 where 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 5 presents a diagram of the seq2seq model using GRU networks for the encoder and decoder.

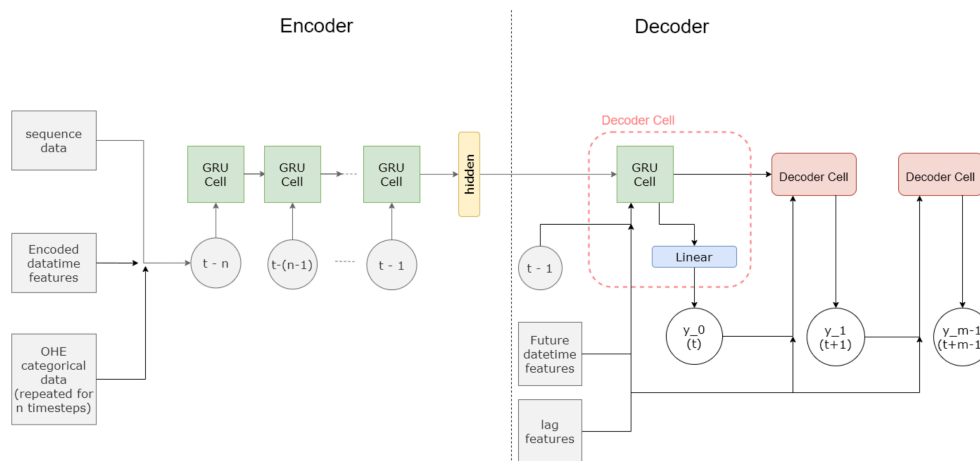


Figure 5: seq2seq model diagram

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 50 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.

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 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. We have also obtained Curtailments in maximum Power generation from Renewable Energy Sources over the years, from 2024 until Sep 2025, and are in the process of estimating the Actual Renewable Energy Sources Contribution to the Electricity Mix. We are developing a multi-horizon sequence-to-sequence (Seq2Seq) forecasting system using LSTM networks for univariate time series prediction, specifically designed for high-frequency data (15min intervals). It trains separate encoder-decoder models for 6h, 12h, and 24h forecast horizons, employs log-transformation for stability, and uses realistic rolling-window backtesting with metrics which include sMAPE (symmetrical Mean Absolute Percentage Error) and MASE (Mean Absolute Scaled Error), with early results described in Figure 51 in the Annex of this deliverable, to provide 6h, 12h and 24h prediction horizons of the Actual Renewable Energy Contributions to the Electricity Mix.

The project team will take into consideration the ToU tariffs, peak demand charges, charging policies and, in combination with the actual 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 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 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 6. 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.

In the Data component of the architecture, 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

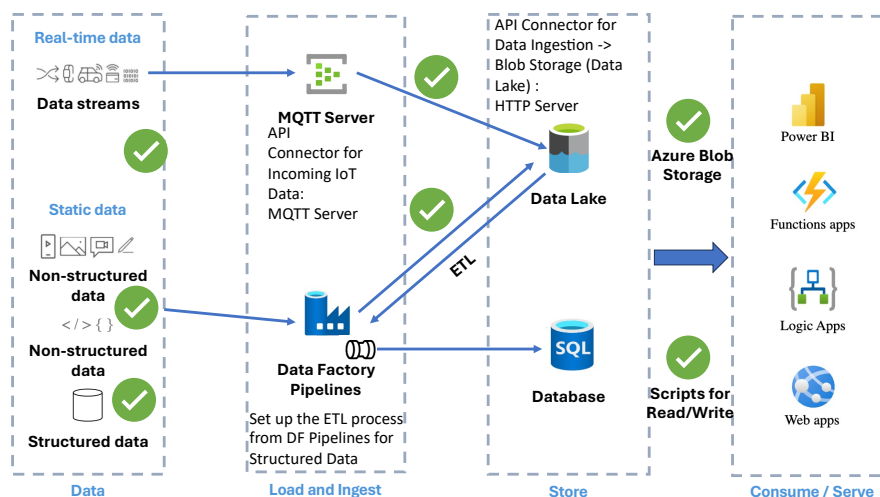


Figure 6: Conceptual framework architecture

In the Load and Ingest component of the architecture, an MQTT server has been set up for real-time data stream ingestion. An MQTT server, or broker, is a central hub in a publish/subscribe messaging system, acting as a lightweight, efficient intermediary for IoT devices and applications to exchange data over low-bandwidth or unreliable networks. Key functions include managing client connections, authentication, and routing messages based on specific topics. 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 allows us to host web applications.

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

- **Real-time data ingestion:**
 - Resources needed: Event Hub, Data Lake (Azure Blob Storage)
 - Criteria to set as complete:
 - Ability to ingest: real-time structured data (e.g. IoT data) and real-time images or video (e.g. bird-eye photos from Drones, or videos from Traffic-light cameras)
- **Static data ingestion without transformation**
 - Resources needed: Data Factory, PostgreSQL (+postGIS plugin)
 - Criteria to set as complete:
 - Ability to ingest: Static structured data
- **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
- **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
- **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 6. 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. The data was successfully received and stored in the Data Lake. 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. Implementation of a predefined set of data access permission levels was completed using Role Based Access Control (RBAC). The Annex of this deliverable includes an example of the ESP32 microprocessor (Figure 52) and screenshots of the received IoT and static data in the Data Lake (Figure 53 and Figure 54).

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. Assessment of Automated Charging Technologies

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.

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 7), 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).

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.

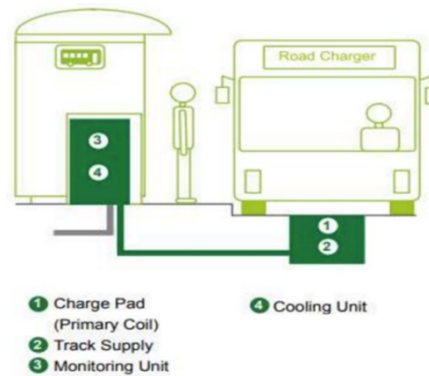


Figure 7: Charging components for Inductive charging (Source: Sawilla & Schuett, 2017)

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 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. Automated Charging System

Architecture

The automate charging architecture system was defined to facilitate robust and efficient automated charging, particularly for automated vehicles, ensuring their seamless interaction and a holistic approach support to the automated charging ecosystem (see a visual representation of the automated charging system architecture in Figure 8).

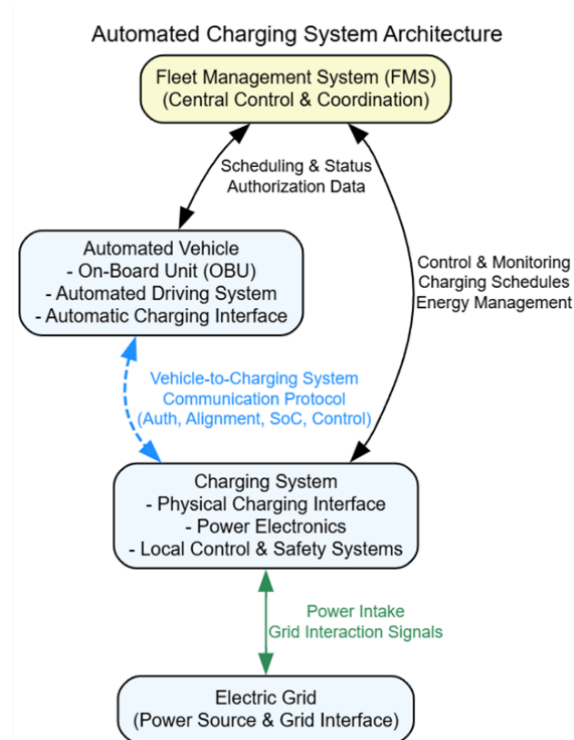


Figure 8: Automated charging system architecture

This architecture is structured around five main elements:

- Automated Vehicles:** the system 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.
- Charging system:** The 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.

Automated Charging Process

In order to ensure clarity and a standardized approach for the interaction between automated vehicles and charging infrastructure, an automated charging process was defined. The sequential steps of this process are described below and a visual representation of the process is included in Figure 56 in the Annex of this deliverable.

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. 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. 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.

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. 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. 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. 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.

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. 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. 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.

Next Steps

Remoted developed this concept in 2025 based on detailed research activities and practical work, yielding three key observations. First, while several charging technologies are available—including inductive charging, physical connections, and pantographs—some require significant implementation efforts by vehicle manufacturers and Remoted currently lacks control over selecting the most optimal automated charging solution due to legal liabilities; however, the company continues to collaborate closely with selected partner OHMIO. Second, the most effective charging strategy aligns with business objectives, with the primary goal of enabling on-road vehicle charging that necessitates long-term contracts to offset implementation costs for charging stations. Third, safety remains paramount when dealing with electricity. Taking these observations into account, Remoted has adopted an incremental development approach structured across four phases: Phase 0 (Development & Early Testing, Aug–Dec 2025) achieved Technology Readiness Level (TRL) 4 through integration of the Universal Robots UR3e AC charging arm (see Figure 55 in the Annex of this deliverable) with the OHMIO vehicle and Proof-of-Concept (PoC) development involving RTDE telemetry, OpenCV, and URScripts in partnership with Lapland University of Applied Sciences; Phase 1 (Validation, Jan–Apr 2026, ongoing) targets TRL 5 with PoC validation for automated charging in depot conditions, enhancing electricity intake from 3x16A to 1x32A in collaboration with facility owners and local providers, alongside real-environment testing and MVP planning based on lessons learned; Phase 2 (MVP Validation, starting Mar/Apr 2026) aims for TRL 6 by developing a DC rapid charging MVP and defining technical and market-fit requirements; and Phase 3 (Market Fit, 2026 onward) will finalize the commercial solution, including infrastructure, permits, and validation with the city of Tampere, reaching an as of yet undetermined Technology Readiness Level.

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

Task 2.4 examines 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 (RCC), led by REMOTED, which has outlined the RCC flowchart and functionalities enabling fully driverless operation through a distributed architecture featuring multiple RCCs with primary/secondary supervision responsibilities, failover mechanisms, and emergency service connectivity addressing legislative requirements for Level 4 AV supervision, edge case interventions, and public transport service needs while supporting scalable Vehicle-to-Operator ratios from 1:1 toward 4:1 with AI assistance; REMOTED continues to refine these developments with updates in progress. This establishes an initial process for remote control operations across scenarios requiring operator interventions. Subtask 2.4.2 investigates modular bus driving assistance systems, where NEXT has completed Navigation Testing, Braking, and Gearing Control, and UNINA solved the Docking maneuver with ADAS controller assistance. Subtask 2.4.3 features NTUA's real-time planning model for modular bus fleets that skip low-demand stops to increase efficiency.

Regarding future steps, NEXT has completed mapping and platform navigation testing and plans to implement higher-level behavioral planning algorithms for end-to-end summoning/parking operations alongside hardware updates. NTUA is advancing toward flexible near-real-time rescheduling of the modular bus system to dynamically adapt operations to demand and network conditions.

5.1. Subtask 2.4.1 Remote Control Centre for AVs

While Level 5 Automated Driving Systems (ADS) are intended to operate in all conditions and environments, no current 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 from outside and intervene when necessary to maintain safety and ensure continuous travel.

Introduction

A Remote Operation Center (ROC/RCC) structure, as presented in Figure 9, 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 (remote driving) based on operational needs and technological constraints such as latency and sensor reliability. Emphasis is placed on ergonomic workstation design, situational awareness (SA) 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 for teleoperation requiring low latency while monitoring and assistance allow for greater flexibility. The ROC should 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.

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 emergency scenarios 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.

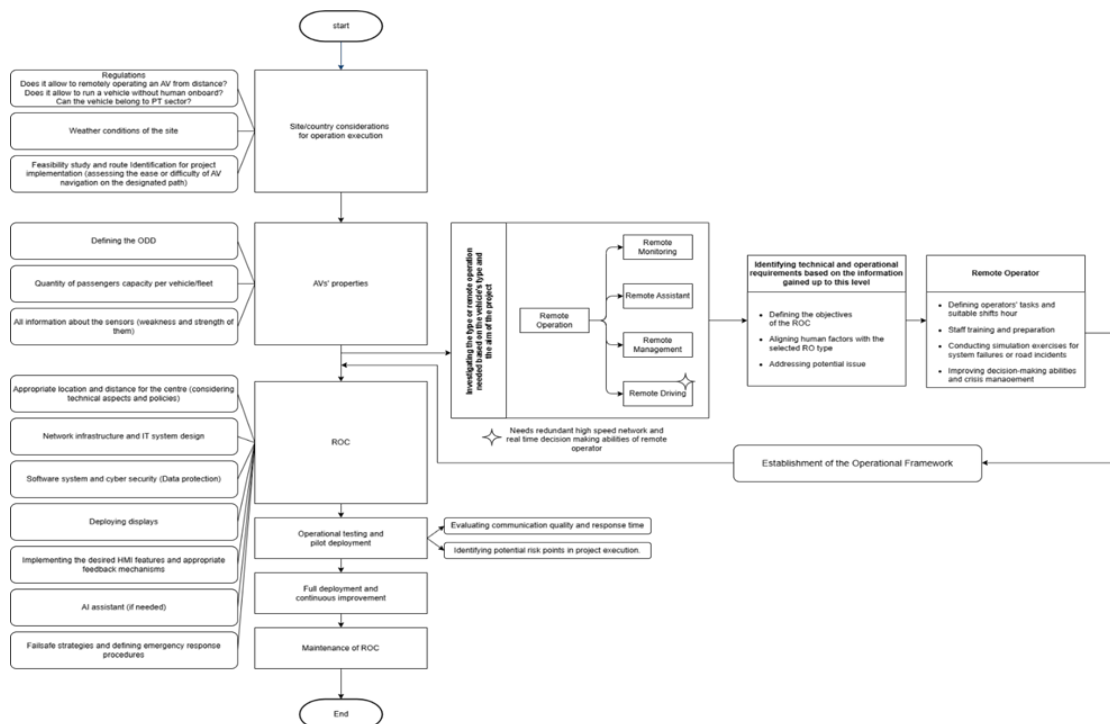


Figure 9: ROC flowchart

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.

Distributed Architecture and Resilience Mechanisms

The distributed architecture visualized in Figure 10 (a), 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.

The sequence diagram presented in Figure 10 (b), 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.

In addition, 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 of handover and maintains a direct line with emergency services to ensure safety and service continuity. Figure 57 in the Annex of this deliverable illustrates the key information flows between its core functions and stakeholders.

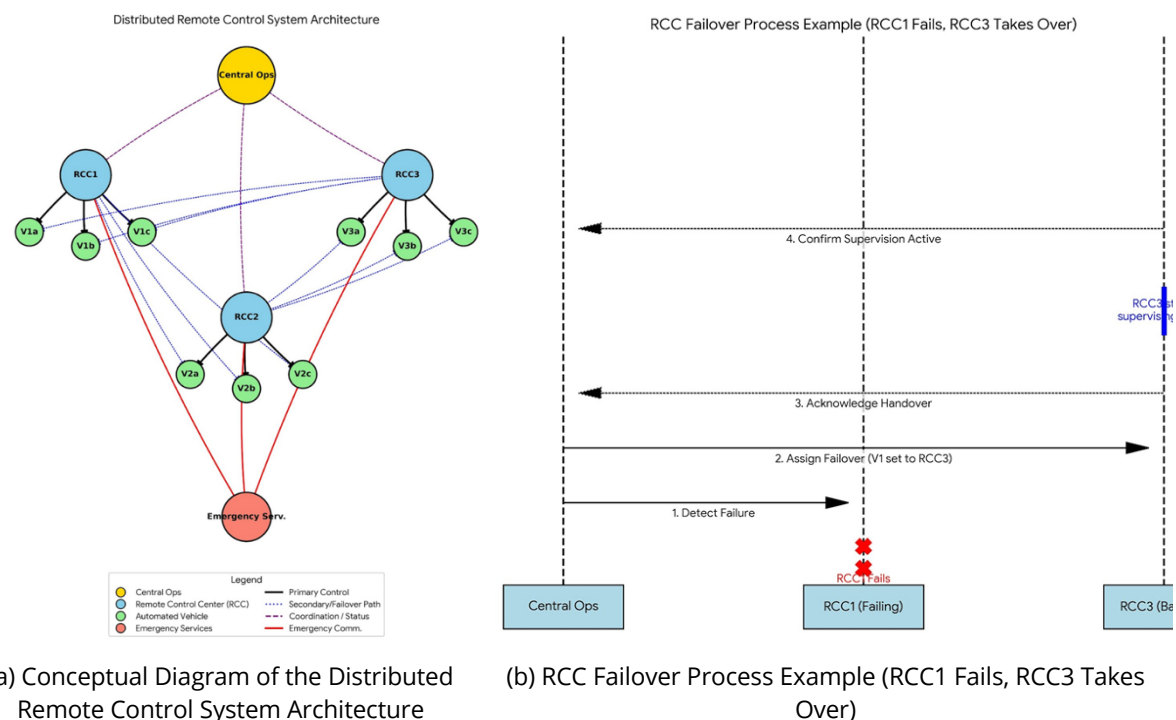


Figure 10: Distributed Architecture and Resilience Mechanisms

Operational Scalability and Field Interaction

The economic feasibility of deploying automated vehicles in public transport heavily relies on achieving a favourable 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. metaCCAZE 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.

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. Details on the sensing characteristics of the Next vehicle are provided in Subtask 2.4.2 in the Annex of this deliverable.

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 11. 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

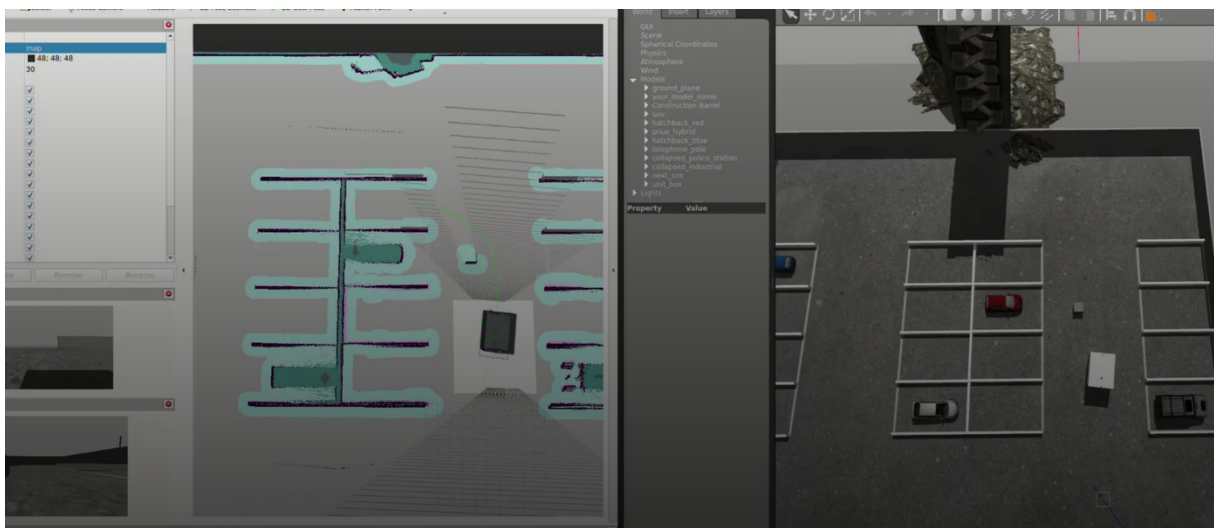


Figure 11: Simulated Navigation

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.

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.

Platform evolution: Transition to Fifth Generation Next Modular Pod

Following the successful development and testing phase on the third-generation pod, the project is transitioning to Next Modular Vehicles' fifth-generation pod platform. This evolution represents a significant advancement in vehicle design, offering improved mechanical precision and vehicle dynamics while maintaining the core autonomous navigation capabilities.

The fifth-generation pod presents slightly larger dimensions compared to the third generation (length: 3.90 m, width: 2.46 m, height: 3.00 m). Despite the increased footprint, the vehicle maintains excellent maneuverability suitable for urban parking scenarios, while providing greater structural rigidity and improved long-term reliability.

The fifth-generation pod will feature its drive-by-wire system with custom-developed add-on actuators for steering, throttle, braking and gear control, in contrast to the native drive-by-wire hardware of the previous vehicle. This new setup ensures compliance with the existing hardware on the new model. This architecture provides enhanced control precision compared to the third-generation pod, thanks to improved mechanical response and superior vehicle dynamics. These improvements will lead to more predictable behavior during low-speed parking maneuvers, critical for autonomous operations.

The sensor suite architecture remains conceptually identical to that described previously, maintaining the multi-layered approach with LiDARs, stereo cameras with IMU, and GPS. However, sensor placement will be adapted to the new pod's geometry and mounting points. The revised sensor placement strategy will maintain the coverage of the previous sensors and is currently under implementation.

A critical design goal for the fifth-generation pod is to minimize software migration effort. The autonomous navigation stack developed on the third-generation pod, including localization

algorithms, path planning, and high-level control logic, will be fully compatible with the new platform, under minimal adaptations. Required modifications are limited to updated kinematic parameters, recalibration of sensor fusion matrices, and adjusted control gains. The core algorithms, ROS2 architecture, and navigation stack remain unchanged, significantly reducing development time and risk.

Upon completion of the new vehicle integration, a comprehensive validation campaign will be conducted to quantitatively assess control precision improvements through standardized autonomous parking manoeuvres, perform comparative benchmarking against the third-generation pod measuring trajectory tracking accuracy, stopping precision, repeatability, and overall system robustness, and validate drive-by-wire system performance under various operational conditions and edge cases. These tests will provide empirical evidence of the performance enhancements enabled by the fifth-generation platform, demonstrating the continued evolution of the Next Modular Vehicles pod (see Figure 12) towards greater reliability and scalability for real-world autonomous mobility deployments.



Figure 12: Next pod in Milan

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. This allows for 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 maneuver 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 13 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. The steps of the problem and its possible outcomes are presented in Figure 14.



Figure 13: Schematic representation of a docking manoeuvre

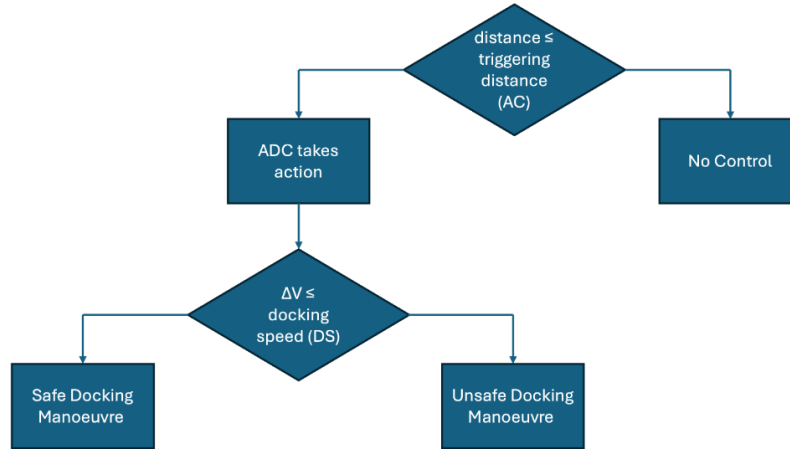


Figure 14: Problem Statement

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. In addition, 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 for the safety condition to be satisfied. We refer to this minimum distance as the “Minimum Safety Distance (MSD)”. Subtask 2.4.2 in the Annex of this deliverable includes a brief description on the MSD calculation.

The simulation framework we designed allows for the implementation of an ADC and the evaluation of its effectiveness in ensuring a safe merging manoeuvre. 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 (9):

$$a = (k_p * e) + (k_i * i) + (k_d * d) \quad (9)$$

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: $e = DV - DS$, $i = \sum e * dt$, and $d = \frac{De}{dt}$.

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.

A Global Sensitivity Analysis (GSA) technique was used to evaluate how input variables impact model outputs. This technique 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 (10) and (11) respectively:

$$S_j = \frac{V_{X_j}(E_{X_{\sim j}}(Y|X_j))}{V(Y)} \quad (10)$$

$$ST_j = \frac{E_{X_{\sim j}}(V_{X_j}(Y|X_{\sim j}))}{V(Y)} \quad (11)$$

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; and $E_{X_{\sim j}}(V_{X_j}(Y|X_{\sim j}))$ 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 ($AC = MSD + DeltaDistance$). For each of the above mentioned 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.

First Simulation Results

The results of the GSA performed on the outputs of the conducted simulations are presented in Figure 15 and Figure 16. Figure 15 presents how the output variable Y , e.g. the relative speed at docking, varies with respect to the input variables. In each of the diagrams, 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 15 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 15. Vice versa, the influence of Delta Distance is evident, as depicted in Figure 15c. 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.

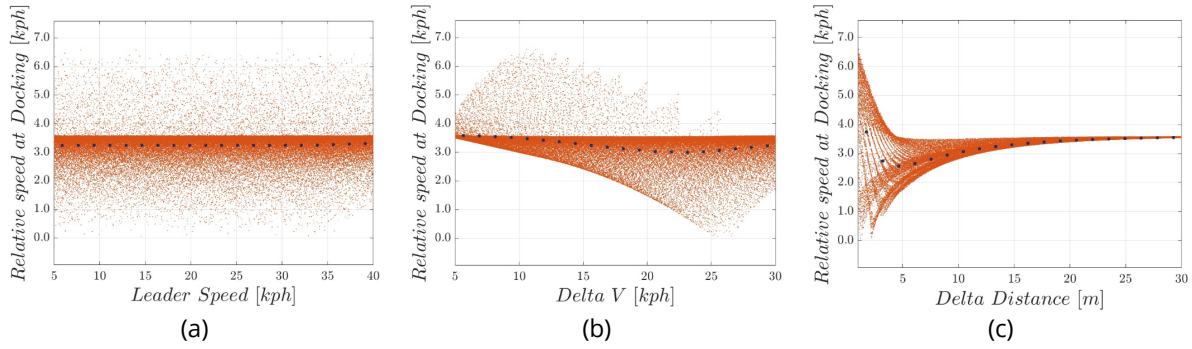


Figure 15: Trend of the variable Relative Speed at Docking with respect to the input variables Leader Speed (a), Delta V (b), and Delta Distance (c).

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 Figure 16. Indeed, these indices reach their highest values precisely for Delta Distance.

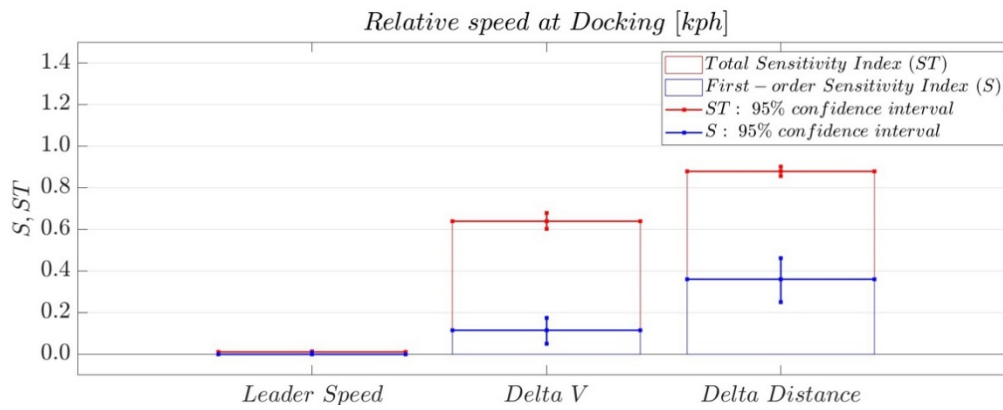


Figure 16: GSA indexes

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. 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. Recently, we've 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 testing to date indicates that it allows seamless 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. 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, and 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. 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. 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. This paper has been submitted to a scientific journal, and it is under review. Also, it will be presented as a poster at the TRB Annual Meeting during the session Research in Transit Operations: Bus Operations on 12 January 2026.

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:

$$\sum_{u \in U} \sum_{s \in S} x_{s,u,t} = 1, \forall t \in T \quad (12)$$

$$\widetilde{b}_{t,k} \doteq \sum_{t \in T} w_{t,k,y} \sum_{u \in U} \sum_{s \in S} x_{s,u,t} \beta_{s,k,y} \forall t \in T, k \in K \quad (13)$$

$$l_{1,1} = \min \left\{ \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} u; \widetilde{b}_{1,1} \right\} \quad (14)$$

$$l_{1,1} \leq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} u \quad (15)$$

$$l_{1,1} \leq \widetilde{b}_{1,1} \quad (16)$$

$$l_{1,1} \geq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} U - d_{1,1} M \quad (17)$$

$$l_{1,1} \geq \widetilde{b}_{1,1} + (d_{1,1} - 1) M \quad (18)$$

$$a_{1,1} \doteq \sum_{y \in K, y \geq 1} w_{1,1,y} - l_{1,1} \quad (19)$$

$$z_{1,1} \doteq \frac{l_{1,1}}{\sum_{y \in K, y \geq 1} w_{1,1,y}} \quad (20)$$

$$g_{1,1,y} = (1 - z_{1,1}) w_{1,1,y} \forall y \in K, y \geq 1 \quad (21)$$

$$l_{1,k} = \min \left\{ l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}; \gamma \sum_{s \in S} \sum_{u \in U} x_{s,u,1} u \right\} \quad (22)$$

$$l_{1,k} \leq l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k}, \forall k \in K \setminus \{1\}, \forall y \in K, y < k \quad (23)$$

$$l_{1,k} \leq \gamma \sum_{s \in S} \sum_{u \in U} x_{s,u,1} U, \forall k \in K \setminus \{1\} \quad (24)$$

$$l_{1,k} \geq l_{1,k-1} + \widetilde{b}_{1,k} - \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{s \in S} \sum_{u \in U} x_{s,u,1} \beta_{s,y,k} (d_{1,k} - 1) M \approx \forall k \in K \setminus \{1\} \quad (25)$$

$$l_{1,k} \geq \gamma \sum_{u \in U} \sum_{s \in S} x_{s,u,1} U - (d_{1,k}) M, \forall k \in K \setminus \{1\} \quad (26)$$

$$z_{1,k} = \frac{l_{1,k} - l_{1,k-1} + \sum_{y \in K, y < k} z_{1,y} w_{1,y,k} \sum_{u \in U} \sum_{s \in S} x_{s,u,t} \beta_{s,y,k}}{\sum_{y \in K, y \geq k} w_{1,k,y}} \forall k \in K \setminus \{1\} \quad (27)$$

$$a_{(1,k)} = \sum_{y \in K, y \geq k} w_{1,k,y} - (1 - z_{1,k}) \forall k \in K \setminus \{1\} \quad (28)$$

$$g_{1,k,y} = (1 - z_{1,k}) w_{1,k,y}, \forall k \in K, \forall y \in K: y \geq 1 \quad (29)$$

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 (12). Constraint (13) 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 (15), (16), (17), and (18), where $d_{1,1}$ is a binary variable. In more detail, these constraints 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 (19) 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 (20) computes the percentage of passengers willing to board at the first trip at the first stop and are able to do so. Constraint (21) computes the number of passengers who could not 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 (22), 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: (1) the passenger load of trip 1 when departing from the previous stop, $l_{1,k-1}$, plus (2) the number of passengers of trip 1 willing to board at stop k , $\widetilde{b}_{1,k}$; minus (3) 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 (22) can be substituted by the following set of linear inequality constraints (23), (24), (25), and (26), where $d_{1,k}$ is a binary variable. Constraint (28) 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 (29) computes the number of passengers who could not board trip 1 at stop $k \in K \setminus \{1\}$ and alight at the stop $y \in K$. (29) 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 17. 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. 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.

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 $s=4$.



Figure 17: Case Study Network (both directions)

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 18 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.

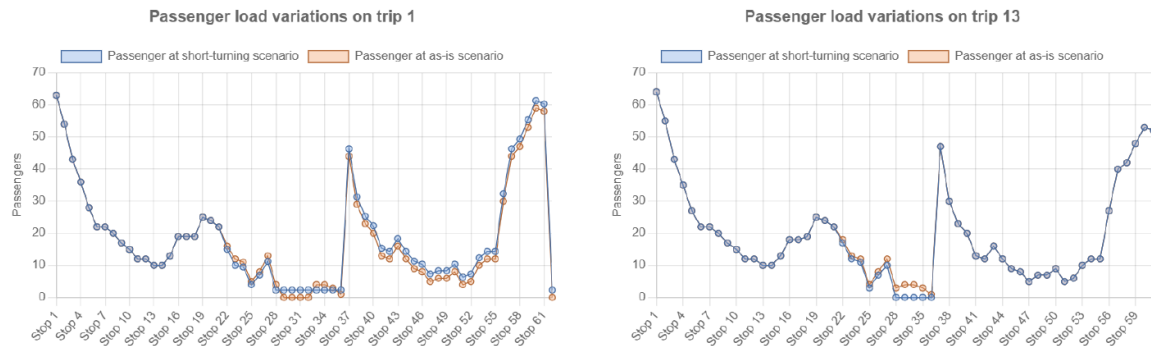


Figure 18: Passenger load variations when using modular autonomous vehicles with short-turning options and when implementing the as-is scenario with conventional vehicles

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-travelled, and 52% in terms of overall operational costs compared to the as-is scenario. Our implementation of the model produced promising results, while the number of passengers who were not able to board their first trip was relatively low (12 passengers in total). In future research, our approach can be expanded by receiving real-time information from passengers about their willingness to board the bus at specific stops and performing real-time rescheduling. Finally, other flexible approaches, such as interlining, can be considered as an addition to short-turning.

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 intelligent roads supporting public and eco-responsible transportation means like carpooling. Up until M26, for subtask ST2.5.1 with the title "Traffic management and condition assessment", Floware designed, prototyped, and validated a sensor-based data acquisition infrastructure for the Producer Layer of the Smart Mobility Hub framework, featuring improvements for low-light conditions, energy supply monitoring, and alternative configurations, including a motorized camera prototype with enhanced waterproofing and upgraded edge AI hardware. Mobilysis is in the process of developing a framework for collecting video data from traffic streams using swarms of drones and processing these data to detect traffic congestion issues and provide evidence-based solutions (related to Asset 3).

In ST2.5.1, Floware has progressed from the D2.1 prototype stage towards a deployable and interoperable sensing unit. Consolidation work covered hardware simplification (in the aim of moving away from the former Dual Control approach towards a more integrated architecture while preserving fault-tolerance principles), reinforced power management and autonomy validation, and strengthened embedded reliability (monitoring and recovery logic). In parallel, Floware implemented secure device-to-cloud operations and cloud ingestion, and advanced interoperability with VEDECOM through the definition of a cloud-to-cloud data exchange approach. The shared API is currently being developed to expose processed indicators and associated metadata in a best-effort near real-time fashion, enabling technical validation and progressive integration with VEDECOM's data environment and V2X-related streams (e.g., CAM/DENM) within the Smart Mobility Hub framework. Upcoming work will focus on final deployment packaging, first end-to-end integration tests with VEDECOM, and the operationalisation of unified mobility indicators for the Poissy-Triel-sur-Seine living lab.

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 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.

6.1. Subtask 2.5.1 Traffic management & condition assessment

As reported in Deliverable D2.1 (M1–M18), 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 are further analysed in the following sections of this deliverable.

Since M19 (D2.2 reporting period, M19–M26), Floware has continued and expanded work across all four domains while shifting focus from exploratory prototyping to deployment readiness, enhanced perception/analytics, and external interoperability. Hardware improvements progressed through iterative, test-driven refinements (robustness, packaging and field suitability), while also advancing a simplification path from the Dual Control model (Jetson Orin Nano + Raspberry Pi 4) towards a more integrated and cost-effective architecture that preserves fault-tolerance principles through dedicated components. Embedded reliability and device operations were further strengthened (monitoring, autonomous recovery logic, secure connectivity and remote operations) to support sustained field testing and maintainability. In parallel, Floware strengthened the data and analytics layer by testing and improving computer vision models and enhancing the processing pipeline to support multi-source data fusion between anonymised Bluetooth signals, computer-vision outputs, and a pilot license plate detection/recognition (LPR) capability (introduced as a complement to the initial Bluetooth + Computer Vision plan). This fusion supports a unified representation of how people and vehicles use station areas, parking facilities and corridors between hubs, enabling indicators such as OD counts and travel times (e.g., Poissy–Triel-sur-Seine), station inflow/outflow, parking usage/rotation/dwell times, and waiting-time proxies around mobility services. Regarding interoperability with VEDECOM, technical exchanges have taken place, and the integration principles have been defined (cloud-to-cloud exchange, scope separation between raw data and processed indicators, and best-effort near real-time delivery compatible with the intended operational use). The API specification is currently being finalised to move into implementation; however, full API finalisation is not on the critical path for deployment. Overall, the sensing stack and analytics pipeline are aligned and ready to support the planned sensor deployment at the end of March 2026 (M27).

MobiLysis's contributions to ST2.5.1 were limited from M18 onwards per schedule, pending Floware sensor installation to resume drone flight activities supporting the task. This complemented earlier MobiLysis work on drone swarm-based video data collection from traffic streams with advanced computer vision algorithms for congestion detection linked to metaCCAZE Asset 3.

Multimodal sensing unit: architecture and integration approach

Floware's work in ST2.5.1 targets a modular and resilient urban sensor platform for multimodal mobility monitoring, capable of capturing rich data while operating autonomously in outdoor environments. The development is organised around four technical domains that structure the section: hardware prototyping, embedded software development, power supply analysis, and system integration. An initial phase concentrated on identifying the most effective combination of sensing, computing, and communication components and establishing a reliable embedded stack (monitoring and recovery, secure communications). The more recent phase focuses on consolidating the prototype into a deployable unit and preparing external interoperability, including planned technical validation with VEDECOM, ongoing API work for data exchange, and hardware simplification from the Dual Control architecture (Jetson Orin Nano + Raspberry Pi 4) towards a more integrated solution with dedicated fault-tolerance components.

Hardware prototyping and field hardening

Floware conducted acquisition and test campaigns to source and validate key components for an outdoor, autonomous sensing unit. The component space explicitly covered: high-resolution RGB/IR-capable cameras, Jetson Orin Nano for on-edge computer vision inference, Raspberry Pi 4 for auxiliary functions/energy management, solid-state LiDAR for volumetric detection/validation, BT/BLE scanners for passive device detection, 4G modems with fallback management logic, IR illuminators for low-light enhancement, and supporting mechanical and power elements (housings, mounts, converters, thermal components).

These configurations were iteratively tested under diverse (simulated and real) environmental conditions against concrete performance criteria: image quality, Bluetooth detection range, LiDAR stability, bandwidth usage, and system temperature under load, with special attention to low-light performance and synchronisation between sensing and transmission subsystems. This benchmarking also included variations in camera optics, BLE antennas, and microcontroller firmware, to optimise trade-offs between accuracy, cost, power usage, and reliability across different mission profiles (from low-traffic anonymous flow detection to denser multimodal junction monitoring).

On the engineering execution side, Floware also coordinated with component suppliers to address practical constraints that affect deployability and long-term maintainability (delivery lead times, firmware incompatibilities, mechanical integration constraints). More recently, the hardware line of work explicitly shifts from exploration to deployable-unit consolidation, including simplification of the Dual Control architecture (Jetson Orin Nano + Raspberry Pi 4) towards a cost-effective integrated solution while retaining the fault-tolerance principles through dedicated components replicating the existing fault-tolerance logic.

Power supply and autonomy validation

Power stability and environmental adaptation are treated as key requirements for real-world deployment of the edge units. Floware implemented and tested a hybrid power strategy covering

three primary configurations: grid-connected mains supply with industrial-grade AC/DC converters, solar systems with MPPT controllers, and battery-only operation for short-term autonomous deployment. These setups were assessed under different conditions including fluctuating solar exposure, power outages, and irregular municipal grid voltage input. Multiple lithium-ion battery packs were evaluated with trade-offs in autonomy, temperature stability, and recharge cycles, and the work converged to battery choices supporting 48–72 hours continuous operations in worst-case scenarios.

The testing campaign surfaced a set of concrete failure modes relevant to outdoor deployments: voltage drop and inverter instability during high-load image processing, overheating of passive converters in sealed enclosures, charge controller malfunctions in cloudy/shaded environments, as well as moisture-related connector corrosion and EMI-related noise affecting sensor accuracy. Floware then implemented iterative mitigations including temperature-activated ventilation in test enclosures, weather-sealed industrial connectors, 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 are being integrated into the reference design. In parallel, continuous power/energy health monitoring is supported by a custom battery monitoring system (based on I2C communication with battery management components), collecting voltage/current/charge-discharge statistics in real time to detect abnormal patterns and optimise power usage strategies.

Secure connectivity and device operations

Floware deployed Azure IoT Hub from scratch and implemented secure device-to-cloud communications to ensure authenticated, encrypted data exchange between edge devices and cloud infrastructure, enabling large-scale data ingestion and remote logging for field testing and maintenance. Remote configuration was supported through device-side configuration mechanisms (device twins) allowing updates of sensor configuration parameters.

To support operational supervision, Floware developed monitoring tooling for system health metrics such as CPU load, memory usage, temperature, and storage availability, combined with remote logging to support debugging during field tests. More recently (and very concretely), Floware validated cloud-to-device command capability to trigger remote checks/actions on the device and monitored device replies/events from the cloud side; this was then moved from ad-hoc tests to a more operational setup by running a persistent command listener on the device to receive and execute cloud-originating instructions.

Edge perception stack and embedded reliability foundations

On the embedded side, Floware developed a robust software stack spanning low-level microcontroller code and higher-level software on embedded microcomputers, with focus on resilience and secure communication. A central achievement of the earlier phase was a multilevel supervision and recovery mechanism: the units detect failures such as network disconnection, inference process crashes, or power dips, and autonomously trigger corrective actions (controlled reboots, fallback procedures, watchdog resets).

This embedded reliability layer is tightly coupled with the hardware exploration: the platform uses Jetson Orin Nano for on-edge computer vision inference and integrates low-light enhancement components (IR illumination), while keeping auxiliary functions and energy management in the overall design (notably via Raspberry Pi 4 in the Dual Control configuration). Long-run testing explicitly evaluated temperature under load and bandwidth usage alongside sensing performance, reflecting the “operational CV on edge” constraint (sustained inference + connectivity + outdoor

thermal envelope). More recently, the direction of work is to consolidate these elements into a deployable unit and simplify the architecture while preserving the fault-tolerance logic, alongside continued work on power management strategies and firmware modularity for context-based adaptation (as stated in the task's forward path).

Bluetooth analytics for mobility patterns (station, parking, corridor)

Bluetooth sensing is used as a continuous, low-intrusion signal to reconstruct mobility flows between sensing points (e.g., movements between stations and along the corridor). The core technical challenge addressed is that raw Bluetooth detections can generate spurious trajectories when sensors are close enough to “see” the same device, or when a device briefly appears as a weak detection on a neighbouring sensor (typical “A-B-A-B...” patterns). This was approached with a filtering methodology designed to be explainable and parametric (with tunable parameters), rather than relying on a black-box model.

Work progressed from observation and experimentation (including controlled tests on how RSSI behaves with distance and obstacles) to an implemented filtering strategy combining:

- time-interval reasoning (what transitions are physically plausible given sensor distances and typical speeds), and
- signal-strength behaviour (distinguishing persistent proximity vs short, weak cross-detections).

The current filtering approach reports a reasonable filtering rate (~7–15%), explicitly justified by physical constraints and empirical observations, and supported by sanity checks (counts before/after filtering, duration per sensor, revisit patterns, etc.). For the living lab, this enables robust production of corridor-level outputs such as OD (origin-destination) counts and travel time distributions between Poissy and Triel, as well as station-area flows when combined with the other sensing modalities (computer vision and LPR).

Pilot license plate recognition (LPR) and robustness measures

In parallel to the base plan (Bluetooth + computer vision), Floware introduced license plate recognition as a pilot/experimental layer to strengthen vehicle-oriented analyses when conditions allow (e.g., corridor travel times and OD reconstruction), and to support cross-checks with Bluetooth-derived dynamics. The work focused on two practical aspects required for field credibility:

(a) Measuring quality and isolating failure modes. A dedicated evaluation was performed to verify recognition behaviour and to separate model-level performance from pipeline/configuration effects. In particular, the analysis highlights that LPR quality can degrade depending on how the detection/recognition stages are integrated, creating observable artefacts (e.g., repeated plate labels) that must be corrected before deployment use.

(b) Stability-oriented post-processing. Because OCR outputs fluctuate frame-to-frame, the processing was designed to be conservative and stable in time: applying confidence gating, selecting stable hypotheses across repeated observations, and filtering anomalies (e.g., strings dominated by repeated characters). These “stability filters” are shown to preserve usable detections while reducing noise and limiting false identifications.

This pilot layer is therefore treated as an additional signal (not a replacement), with a strong emphasis on quality control and stability before it is used for KPI-grade outputs.

Multi-source fusion, consistency checks, and KPI-ready datasets

A key recent technical focus is the objective of producing an integrated view of mobility by combining signals that have different coverage and biases (Bluetooth, computer vision outputs, and the pilot LPR layer). Two research tasks are open in this regard:

(a) Cross-technology validation (LP vs BT trajectories). A dedicated LP-BT trajectory analysis compares trajectories on shared sensor subsets and at multiple aggregation levels (daily, hourly, 15-min). The findings are important for fusion design:

- Bluetooth yields significantly higher volumes, while LP is structurally sparser (often an order of magnitude lower).
- Despite lower volume, LP follows the same temporal dynamics (daily trends, peak hours, intra-day cycles), which supports its use as a validation/enrichment signal under suitable conditions.
- Edge-level heatmap comparisons show that LP can over/under-represent specific edges due to camera positioning and detection constraints. This is used to motivate conservative use and quality flags in fusion.
- LP detection performance is sensor-dependent and sensitive to conditions (e.g., weather/visibility), reinforcing the need to manage LP as a quality-variable signal rather than assuming uniform coverage.

(b) Producing session-based parking datasets (entry/exit, dwell time) with privacy controls.

For parking-oriented indicators (occupancy proxies, rotation, dwell times), the work includes a processing chain that transforms raw detections into session records: cleaning, normalisation of near-duplicate plates caused by OCR errors, pairing entry/exit events into sessions, and applying pseudonymisation so that analysis can be performed without keeping raw identifiers. To reduce OCR noise, the plate normalisation relies on string similarity with a dynamic acceptance threshold (tolerating small errors while avoiding over-merging).

Together, these tasks support the KPI family Floware prepares for the living lab (corridor OD and travel times; station inflow/outflow; parking usage/rotation/dwell; and car-access proxies), with explicit attention to data quality, comparability across modalities, and privacy-preserving outputs.

Interoperability with VEDECOM and readiness for deployment

Floware is preparing technical validation with VEDECOM and the definition of a shared data exchange interface intended to connect Floware-produced outputs with VEDECOM's ecosystem and V2X streams (e.g., CAM/DENM). The integration direction is cloud-to-cloud exchange (rather than direct Floware sensor-to-VEDECOM cloud ingestion), with a best-effort near real-time delivery logic compatible with the intended exploitation (monitoring/analysis; small delays acceptable). The interface definition has been the subject of technical exchanges and is being finalised to move into implementation. Importantly, this interface is treated as an interoperability enabler and does not block the sensor deployment path (deployment readiness is driven primarily by the device consolidation and field robustness work). The sensor deployment is scheduled to take place at the end of March 2026.

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. MobiLysis in 2.5.1 was limited from M18 onwards as they are (according to schedule) on hold based on the advancements of the UC in Yvelines. Specifically, when the installation of the Floware sensors is completed, they will be focusing on conducting the drone flights that can then further support technically the activities of the specific task.



Figure 19: Identifying occlusion areas from trajectories of pedestrians

6.2. Subtask 2.5.2 V2X-based support of transportation services

The second subtask ST2.5.2, with the title "V2X-based support of transportation services" and led by Vedecom, is inter-connected with Assets 9 and 10 of metaCCAZE. During the first 26 months of the metaCCAZE project, Vedecom primarily focused on developing and refining 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 infrastructure for shared services. This provides

seamless integration of traffic management and condition assessment solutions developed in ST2.5.1.

Use Cases: Definition and Objectives

The metaCCAZE project implements two complementary use cases in the Île-de-France region, specifically at Gare de Poissy and Gare de Triel-sur-Seine. These use cases operationalize the Smart Mobility Hub framework through real-world pilot deployments, demonstrating the integration of V2X technologies, sensor infrastructure, and advanced traffic management for sustainable urban mobility.

Use Case 01: Smart Road and Intelligent Traffic Management

Objective:

Use Case 01 establishes the technical foundation of the real-time supervision system for Smart Mobility Hubs by integrating V2X capabilities to improve traffic flow, fleet management, and energy consumption. This use case serves as the data acquisition and communication backbone that enables intelligent decision-making across the mobility ecosystem.

Key Components and Activities:

- **Sensor Infrastructure Deployment:** Installation of V2X-compatible sensors and cameras within selected hubs (Gare de Poissy and Gare de Triel-sur-Seine) developed by Floware, including motorized cameras with enhanced low-light capabilities and edge AI processing.
- **Drone-Based Traffic Monitoring:** Integration of MobiLysis drone swarms for aerial surveillance, capturing traffic trajectories, detecting congestion patterns, and providing high-resolution video data for traffic analysis.
- **Data Integration Framework:** Development of API systems to integrate data producers including sensors, cameras, drones, public transport systems, parking facilities, and V2X-enabled vehicles.
- **V2X Communication Infrastructure:** Deployment of roadside units (RSUs) and establishment of V2X communication channels for real-time data exchange between infrastructure and vehicles.
- **Infrastructure Interoperability:** Provision of standardized data formats and interfaces to enhance interoperability across different systems and enable seamless data fusion.

Use Case 02: Shared Vehicle Services for Enhanced Mobility

Objective:

Use Case 02 aims to operationalize the traffic management framework by testing vehicle-centric supervision functions through integration of shared electric vehicle services. This use case demonstrates how V2X-enabled carpooling services can leverage the infrastructure developed in UC01 to optimize first/last-mile connectivity and off-peak travel while reducing emissions and congestion.

Key Components and Activities:

- **V2X-Enabled Carpooling Service:** Deployment of shared electric vehicles equipped with V2X on-board units operating in the periurban zone connecting to Gare de Poissy, enabling real-time communication with infrastructure and central management system.

- **Smart Parking Integration:** Connection with intelligent parking systems to provide real-time availability information, optimize parking space utilization, and facilitate seamless park-and-ride operations.
- **First/Last-Mile Optimization:** Targeting trips between residential areas and Gare de Poissy during morning/evening peaks and providing flexible mobility solutions during off-peak hours when public transport frequency is reduced.
- **Real-Time Data Integration:** Testing integration of both real-time data (vehicle positions, traffic conditions, parking availability) and offline data (historical patterns, weather forecasts, scheduled events) from connected infrastructure and vehicles.
- **Deep Learning Application:** Implementation of machine learning models for traffic prediction, energy-efficient routing, demand forecasting, and dynamic vehicle repositioning based on predicted demand patterns.

These use cases demonstrate the practical implementation of the Smart Mobility Hub framework, validating the integration of detection (sensors, cameras, drones), communication (V2X infrastructure), treatment (ML-based prediction), and application (carpooling services, parking management) layers. The results will inform scalability strategies and provide evidence-based recommendations for broader deployment across the Île-de-France region and other European cities.

Conceptual Architecture of Smart Mobility Hubs

The evolution of urban mobility requires seamlessly combining sustainable transport practices with intelligent infrastructure and real-time data systems. A key advancement lies in integrating V2X technologies to support intelligent road infrastructure for public and shared transportation services. This subsection presents the technical framework that supports the use cases outlined above, centered on Smart Mobility Hubs (SMHs) and describing the system architecture and enabling technologies.

The Smart Mobility Hub Concept

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.

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. Practical implementation guidance is offered by CoMoUK (2019), which outlines critical design factors such as visibility, intuitive user experience, consistent branding, and inclusive accessibility. Bueno (2021) 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.

Four-Layer Architecture: Detection, Communication, Treatment, and Application

The Smart Mobility Hub framework is organized into four interconnected layers that collectively enable comprehensive traffic supervision and multimodal mobility optimization:

Layer 1: Producer Layer (Detection - Sensors and Cameras)

The Producer Layer as in Figure 20 encompasses the complete spectrum of data generation sources embedded within the urban mobility environment, serving as the foundational layer for both real-time and historical data collection.

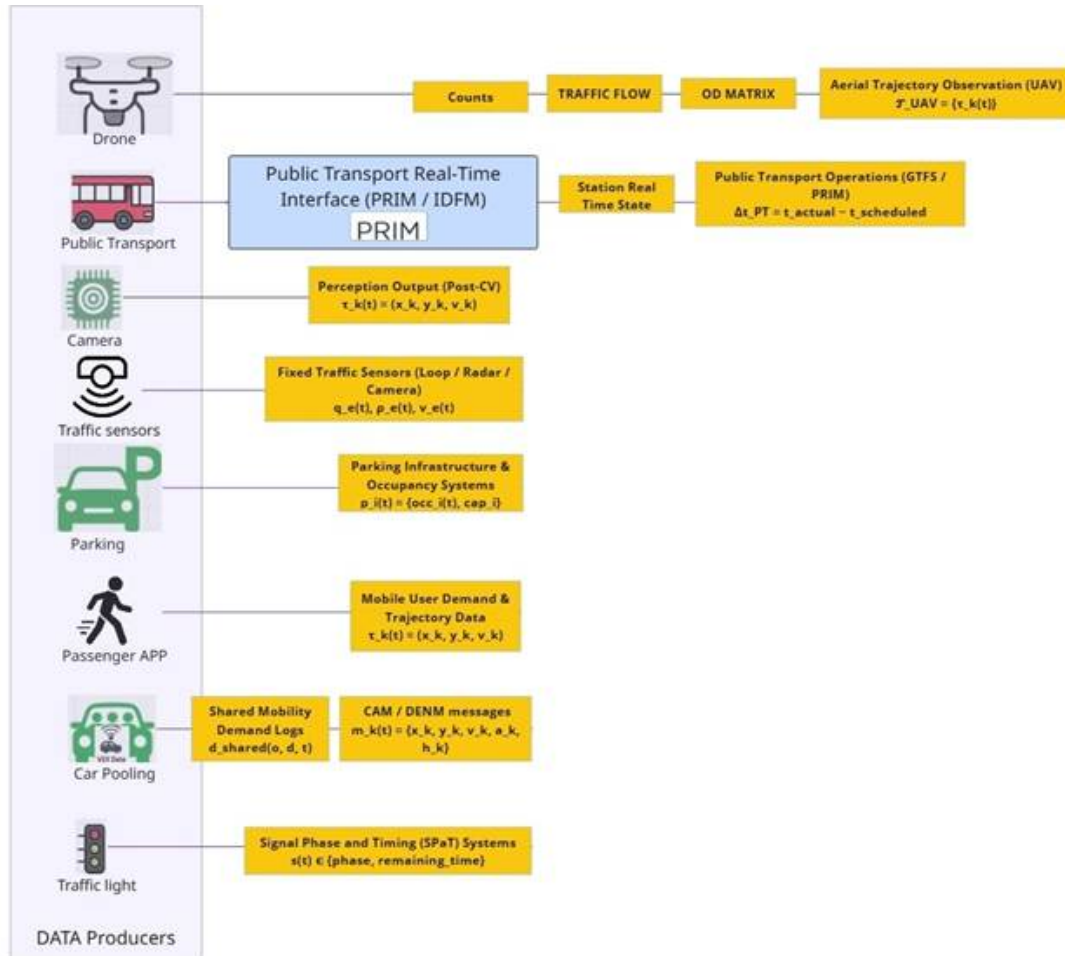


Figure 20: Producer Layer multi-source data acquisition ecosystem with eight primary data streams
This layer integrates diverse sensing platforms and data sources:

- Drone Systems deploy aerial surveillance capabilities, capturing high-resolution imagery and video streams. These generate traffic flow counts, origin-destination (OD) matrices, and aerial trajectory observations $\tau_{UAV} = \{x_k(t), y_k(t)\}$ that enable dynamic traffic pattern analysis and incident detection.
- Public Transport Integration leverages real-time interfaces including PRIM (Plateforme Régionale d'Information Multimodale) and IDFM (Île-de-France Mobilités) APIs. These systems provide station real-time state data and public transport operations information in GTFS/PRIM formats, capturing actual versus scheduled service performance $\Delta t_{PT} = t_{actual} - t_{scheduled}$.
- Camera and Computer Vision Systems employ perception outputs from post-computer vision processing $\tau_k(t) = (x_k, y_k, v_k)$, enabling vehicle and pedestrian detection, behavioral analysis.

- Fixed Traffic Sensors including inductive loops, and roadside cameras record critical traffic metrics: volume, occupancy rates, and classifications $q_e(t)$, $\rho_e(t)$, $v_e(t)$.
- Parking Infrastructure contributes real-time occupancy data $p_i(t) = \{occ_i(t), cap_i\}$ from smart parking systems, supporting dynamic parking management and availability prediction.
- Mobile User Applications collect trajectory and demand data $\tau_k(t) = (x_k, y_k, v_k)$ from passenger apps, capturing travel patterns, preferences, and real-time demand signals.
- Shared Mobility Services provide demand logs $d_{shared}(o, d, t)$ and vehicle telemetry through carpooling and shared vehicle platforms, integrated with V2X communication capabilities that generate Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM): $m_k(t) = \{x_k, y_k, v_k, a_k, h_k\}$.
- Traffic Signal Systems report signal phase and timing (SPaT) data $s(t) \in \{phase, remaining_time\}$, enabling coordination between infrastructure control and vehicle guidance.

Layer 2: Communication Layer (V2X Infrastructure)

The Communication Layer as shown in Figure 21 enables secure, efficient transmission, synchronization, and initial formatting of data collected from the Producer Layer, acting as the bridge between distributed data sources and centralized processing systems.

- **Standardized Mobility APIs and Data Interfaces** serve as the primary integration point, implementing standardized protocols (MQTT, AMQP) with data frequency specifications $\alpha_{\{1:n_1\}} = z$ to ensure consistent temporal alignment across heterogeneous sources.
- **Data Integration and Harmonization** processes incoming streams by normalizing data frequency and granularity, ensuring compatibility between high-frequency sensor data (e.g., 10Hz from vehicles) and lower-frequency sources (e.g., 30-second public transport updates).
- **Distributed Stream Processing Layer** utilizes AMQP/MQTT messaging protocols to handle real-time data ingestion at scale, supporting fault-tolerant, low-latency communication across geographically distributed sensors and systems.
- **Scalable Mobility Data Lake and Time-Series Storage** maintain both historical datasets and real-time streams in a schema-aligned, multi-source spatio-temporal dataset $z(t) = [x, y, q, p, d, \Delta t]$, enabling efficient querying and retrieval for analytical applications.
- **Roadside and on-board communication units** offer dual connectivity supporting both 5G and C-V2X communication technologies for integration into road infrastructure and connected vehicles. These units interface with the RabbitMQ broker through the V2X gateway, enabling bidirectional message flow between physical infrastructure and cloud-based analytics. Besides, roadside units are interconnected with roadside perception sensors and traffic light controller to transmit real-time information on traffic conditions at intersections to connected vehicles.

The V2X gateway architecture integrates seamlessly with RabbitMQ, generating messages from traffic supervision via REST APIs and consuming messages through subscription to RabbitMQ queues and topics. The gateway acts as both producer and consumer: publishing sensor data, predictions, and alerts to the broker while subscribing to vehicle state updates and infrastructure commands. At its core, a V2X software stack implements various C-ITS services and network management functions to reliably and efficiently deliver data messages over 5G networks and V2X roadside units (RSUs) providing the necessary safety information of critical areas (e.g. intersection) to the connected vehicles. Connected vehicles are equipped with on-board unit interfacing with the vehicle information and display for the driver and passengers.

Figure 21 illustrates the overall architecture of the V2X communication system. Concretely, this architecture diagram illustrates a comprehensive V2X (Vehicle-to-Everything) communication system for connected mobility services. The infrastructure consists of three main layers: roadside infrastructure including perception sensors and traffic light controllers that feed data to a V2X

Station and Generation API, an edge server layer hosting V2X message generation and mobility service operations that connects to both a traffic management system and a treatment layer, and the vehicle layer equipped with an on-board V2X unit, 5G connectivity, and display interface. Communication flows through multiple protocols including SPATEM/MAPEM and CPM over V2X links for roadside-to-vehicle messages, and CAM/DENM/POIM-PA messages over 4G/5G for vehicle supervision and broader connectivity. The system enables real-time exchange of road events, parking occupation data, detected objects from perception sensors, and traffic signal states to support enhanced situational awareness and mobility services for connected and autonomous vehicles.

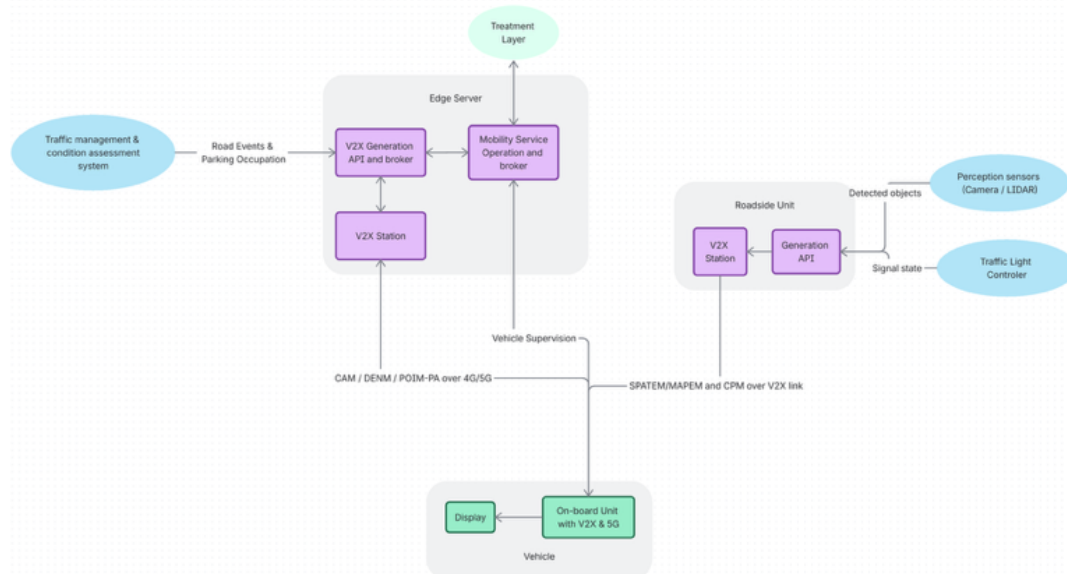


Figure 21: Technical architecture of V2X Communication System

The RabbitMQ broker ensures message ordering, prevents data loss through persistence, and enables parallel processing through consumer groups.

The V2X Communication system is supporting the assessment of the following criteria for the operation of mobility services with connected vehicles:

- Increasing percentage of road alerts informed to the vehicle users
- Increasing the number of intersection crossing realized in safe condition (i.e., without stop)
- Reducing the time to park car near mobility hub

Layer 3: Treatment Layer (Machine Learning and Predictive Models)

The Treatment Layer serves as the computational core of the system, transforming raw, heterogeneous data into structured, actionable intelligence through cloud-hosted advanced analytics as depicted in Figure 22.

- Data Validation and Preprocessing begin with noise filtering and missing value imputation ($Z_{raw} \rightarrow Z_{clean}$), ensuring data quality and consistency across all sources. This step corrects errors, handles outliers, and standardizes formats.
- Exploratory Data Analysis and Statistical Characterization employ descriptive spatio-temporal statistics $X = D(Z_{clean})$ to uncover patterns, identify trends, detect anomalies, and establish baseline performance metrics across the mobility network.
- Spatial-Temporal Aggregation combines data streams by zone, segment, and time window, integrating both historical datasets and real-time streams into unified spatio-temporal representations that capture network-wide dynamics.

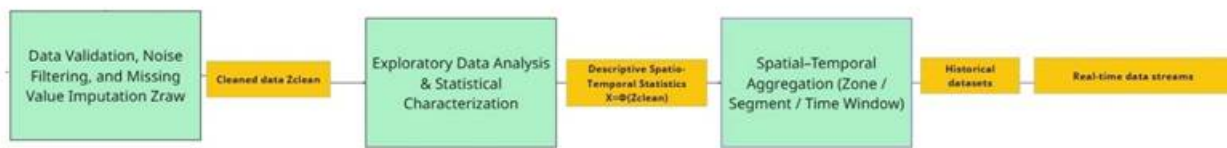


Figure 22: Treatment Layer preprocessing: validation, exploratory analysis, and spatial-temporal aggregation.

Advanced Predictive Modeling leverages sophisticated analytical frameworks as in Figure 23, Dynamic Spatio-Temporal Graph Construction models the mobility network using spatial demand nodes (zones/stations), spatial adjacency relationships based on distance and travel time, and functional connectivity reflecting actual mobility patterns.

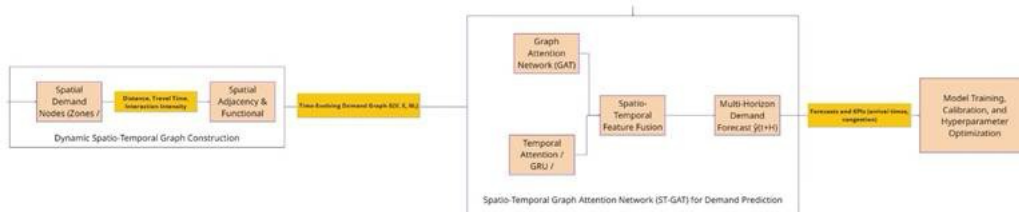


Figure 23: Treatment Layer ST-GAT predictive modeling framework for multimodal demand forecasting. This produces time-evolving demand graphs $G(V, E, W, t)$ that capture network topology and temporal evolution. Spatio-Temporal Graph Attention Network (ST-GAT) employs:

- Graph Attention Network (GAT) layers that learn which spatial connections are most relevant for demand propagation.
- Temporal Attention/GRU mechanisms that capture sequential dependencies and seasonal patterns.
- Spatio-Temporal Feature Fusion that integrates spatial structure with temporal dynamics.
- Multi-Horizon Demand Forecasting capabilities that generate predictions $\hat{y}(t + H)$ for multiple future time steps.
- Model Training and Optimization includes calibration, hyperparameter tuning, and validation to ensure robust performance across diverse operational conditions, weather scenarios, and special events. Models are trained to forecast arrival times, congestion levels, parking demand, and cross-modal transfer patterns.

Application Example: Multimodal Demand Prediction

To demonstrate the capabilities of this framework, we have developed a multimodal demand prediction application that exemplifies how the Treatment Layer's analytical infrastructure can be leveraged for real-world mobility management challenges. This application, which was presented at the Transportation Research Board (TRB) Annual Meeting 2026, addresses the complex problem of forecasting demand across integrated bike-sharing and subway systems.

Problem Formulation

The demand prediction challenge is formally defined through a spatio-temporal framework. The core objective is to learn a mapping function F that forecasts multimodal transportation demand for subsequent time horizons. Given historical demand tensors spanning T time steps, along with weather features, holiday indicators, node embeddings, and graph structure, the model predicts demand for the next H steps across all N network nodes.

This formulation addresses five interconnected challenges inherent to multimodal urban mobility systems. First, heterogeneity arises from modeling fundamentally different transportation modes to a unified predictive framework. Second, interdependency requires capturing demand spillover

effects, where fluctuations in one mode directly influence usage patterns in another. Third, scalability demands computational feasibility over large networks while maintaining real-time prediction capability. Fourth, temporal complexity encompasses daily, weekly, and seasonal cycles, as well as disruptions from holidays and special events. Finally, environmental sensitivity reflects the significant influence of weather and other exogenous factors on mode choice decisions.

Methodology: Spatio-Temporal Graph Attention Network (ST-GAT)

To address these challenges within our application, we employ a Spatio-Temporal Graph Attention Network (ST-GAT) framework. This methodology explicitly captures the spatial and temporal interdependencies between subway and bike-sharing systems within integrated urban environments. The approach models the multimodal network as a heterogeneous graph $G = (V, E, A, E)$, where nodes represent service access points—subway stations and bike-sharing docks—and edges encode both intra-modal (subway-to-subway) and inter-modal (bike-to-subway) spatial relationships. At each time step t , the traffic state is represented as a spatio-temporal graph that maintains temporal continuity by linking each entity across successive time frames. Each node is associated with a feature vector encoding attributes such as location, velocity, and historical demand. A transformer-based encoder then applies multi-head spatio-temporal self-attention to refine these embeddings. This attention mechanism computes relevance weights over neighboring nodes within defined temporal windows, enabling the model to dynamically prioritize the most informative spatial connections and temporal patterns for accurate forecasting.

Layer 4: Consumer Layer (Applications)

The Consumer Layer as shown in Figure 24 delivers refined insights and real-time services from the Treatment Layer to relevant stakeholders and end-user applications, translating analytical outputs into actionable mobility services.

- Standardized Mobility APIs and Data Interfaces provide optimized model parameters and analytical results through well-defined interfaces that support third-party integration and application development.
- User-Level Demand-Aware Guidance delivers personalized routing recommendations, real-time travel time predictions, and mode choice suggestions based on current network conditions and individual preferences.
- Adaptive Signal Control implements dynamic traffic light timing adjustments informed by real-time demand forecasts and congestion predictions, optimizing traffic flow and reducing delays at critical intersections.
- Predictive Parking Demand Management supports real-time parking availability forecasts, dynamic pricing strategies, and guidance to available spaces, reducing search time and improving parking utilization.

This end-to-end architecture ensures seamless integration across all system components, from raw data acquisition through advanced analytics to actionable service delivery. The modular design enables independent scaling of each layer while maintaining system-wide coherence and data integrity.

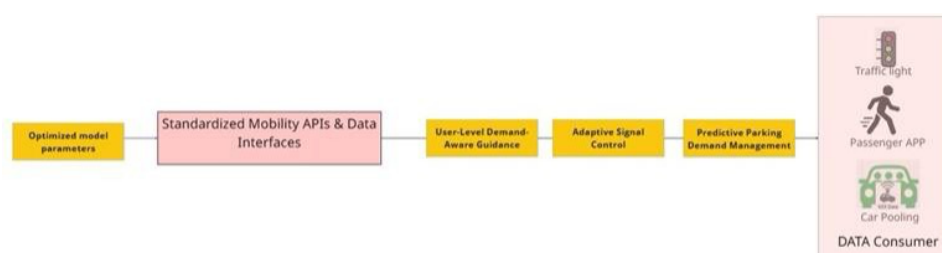


Figure 24: Consumer Layer service delivery pipeline from model outputs to end-user applications

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

Task 2.6 comprises three subtasks addressing the management and operation of zero-emission shared mobility services. Subtask 2.6.1, “Electric Vehicle Scheduling,” is linked to metaCCAZE Technical Assets 1, 8, and 11. Subtask 2.6.2 concerns “Low-Emission Traffic Management” and is associated with Technical Assets 3, 9, and 10. Subtask 2.6.3 focuses on developing a “Supply-Demand Matching Platform for On-Demand Shared Zero-Emission Services,” connected to Technical 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. Since M18, NTUA has directed its research efforts toward extending the existing work to a new planning problem, namely the tactical Vehicle Scheduling Problem for Electric Bus networks with Multiple Depots and Time Windows (EB-MD-VSP-TW). 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 supported the city of Amsterdam in its effort to reduce litter and minimize the presence of heavy-duty vehicles in the city center. The research focuses on multimodal urban logistics optimization under realistic traffic conditions and demand uncertainty.

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 (30):

$$y_k = \sum_{j \in N} t_{kj} q_{kj}, \forall k \in \mathcal{K} \quad (30)$$

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:

$$\min \sum_{k \in K} y_k \quad (31)$$

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:

- Each bus k gets assigned to at least one slow or one fast charger, and their respective charging time slots.
- The energy levels of the buses remain above a specific level SoC_{min} at all times.
- 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).
- The overall charging station installation cost is below a maximum budget.
- The charging station installation selection takes place according to a maximum radius around each last stop of the buses.
- 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 25.

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_k^s and p_k^h 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_{f1}^s or c_{f2}^h 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.

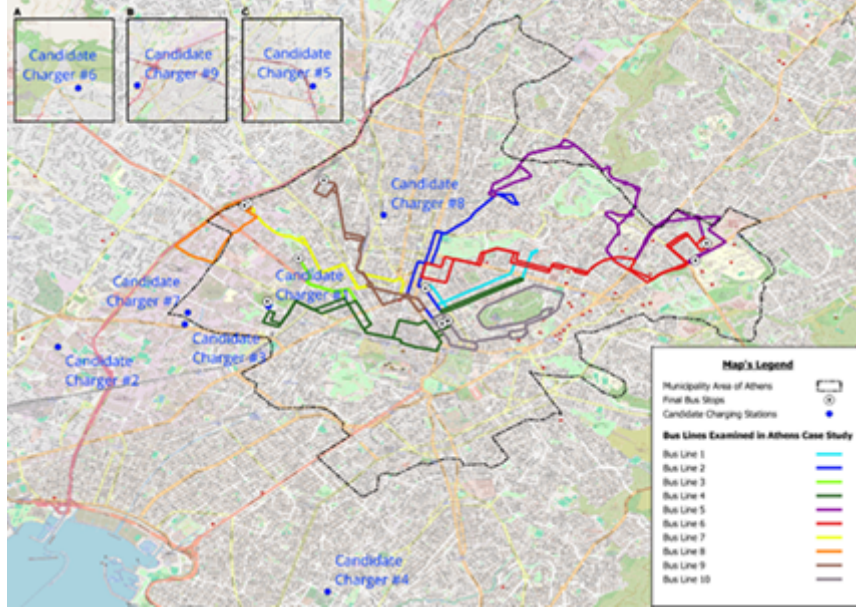


Figure 25: Athens network of the electric bus lines considered for the study, together with the physical locations candidates for the charging stations. Locations A, B and C on the upper left corner represent depots in the Ano Liosia, Nea Philadephia, and Anthousa regions.

Regarding the values of the three-dimensional binary variables u_{k,jf_1}^s and u_{k,jf_2}^h that handle the assignment of bus trips to charging slots, the non-zero values are: $u_{1,15,3}^s = u_{2,15,2}^s = u_{3,1,3}^s = u_{6,15,4}^s = u_{7,1,5}^s = u_{8,1,4}^s = u_{9,15,6}^s = u_{4,2,7}^h = u_{5,16,7}^h = u_{10,16,10}^h = 1$. 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.

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 Annette and Knot, 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 Problem

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_l , 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.

$$(\tilde{Q}): \min \gamma \sum_{k \in K} y_k + \sum_{k \in K} w_k \quad (32)$$

subject to:

$$y_k = \sum_{j \in N} t_{kj} q_{kj} \quad \forall k \in K \quad (33)$$

$$w_k = \sum_{j \in N} [f_{kj} - (\tau_k + t_{kj})] q_{kj} \quad \forall k \in K \quad (34)$$

$$\sum_{j \in N} a_{kj} x_j \geq 1 \quad \forall k \in K \quad (35)$$

$$\sum_{j \in N} q_{kj} = 1 \quad \forall k \in K \quad (36)$$

$$q_{kj} \leq x_j \quad \forall k \in K, j \in N \quad (37)$$

$$q_{kj} \leq a_{kj} \quad \forall k \in K, j \in N \quad (38)$$

$$\sum_{k \in K} q_{kj} \geq x_j \quad \forall j \in N \quad (39)$$

$$\sum_{j \in N_1} x_j \leq \lambda_{1v} \quad \forall v \in V \quad (40)$$

$$\sum_{j \in N_2} x_j \leq \lambda_{2v} \quad \forall v \in V \quad (41)$$

$$\sum_{j \in N} x_j b_j \leq b_{max} \quad (42)$$

$$f_{kj} \geq (\tau_k + t_{kj}) q_{kj} \quad \forall k \in K, j \in N \quad (43)$$

$$l_{kj} \leq c_l q_{kj} \quad \forall k \in K, j \in N \quad (44)$$

$$l_{kj} = f_{kj} + q_{kj} \frac{\omega_k SoC_k^{max} - (SoC_k - ed_{kj})}{r_1} \quad \forall k \in K, j \in N_1 \quad (45)$$

$$l_{kj} = f_{kj} + q_{kj} \frac{\omega_k SoC_k^{max} - (SoC_k - ed_{kj})}{r_2} \quad \forall k \in K, j \in N_2 \quad (46)$$

$$SoC_k - ed_{kj} \geq SoC_k^{min} \quad \forall k \in K, j \in N \quad (47)$$

$$q_{ij} + q_{kj} \leq 1 \text{ if } f_{ij} \leq f_{kj} \wedge f_{kj} \leq l_{ij} \quad \forall k \in K, i \in K \setminus \{k\}, j \in N \quad (48)$$

The objective function (32) 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 (33) define the deadheading time (t_{kj}) of each bus $k \in K$, while constraints (34) define the queue waiting time for each bus k at charger $j \in N$. Constraints (35) 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 (36) assign each bus k to exactly one charging location ($q_{kj} = 1$). Constraints (37) and (38) restrict assignments to construct ($x_j = 1$) and accessible ($a_{kj} = 1$) chargers, respectively. Constraints (39) ensure a charger j is constructed only if at least one bus is assigned to it ($q_{kj} = 1$). Constraints (40) and (41) limit the number of charging types per location $v \in V$, while constraints (42) ensure total installation costs remain within budget b^{max} . Constraints (43) enforce charging sequence of every bus k at the assigned charger j , and constraints (44) ensure that the charging process completes before the predefined end time c_l . Constraints (45) and (46) calculate the charging end time l_{kj} for bus k , based on charging start time (f_{kj}) summed up with the required charging duration. Constraints (45) apply to slow chargers ($j \in N_1$), while constraints (46) apply to fast chargers ($j \in N_2$). Constraints (47) 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 (48) 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.

$$E \left[\gamma \sum_{k \in K} y_k + \sum_{k \in K} w_k \right] \quad (49)$$

which can equivalently be written as:

$$\gamma \sum_{k \in K} \sum_{j \in N} E[t_{kj}] q_{kj} + \sum_{k \in K} \sum_{j \in N} f_{kj} - E[(\tau_k + t_{kj})] q_{kj} \quad (50)$$

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 (50), which can now be written as:

$$\frac{1}{|S|} \left[\gamma \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} t_{s,kj} q_{kj} + \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} f_{kj} - (\tau_k + t_{kj}) q_{kj} \right] \quad (51)$$

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 (43), (45), (46), and (47), 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:

$$(\hat{Q}): \min \frac{1}{|S|} [\gamma \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} t_{s,kj} q_{kj} + \sum_{s \in S} \sum_{k \in K} \sum_{j \in N} f_{kj} - (\tau_k + t_{kj}) q_{kj}] \quad (52)$$

s.t.: Equations (35) - (48)

$$C_{s,kj} = f_{kj} - (\tau_{s,k} + t_{s,kj}) q_{kj} \quad \forall k \in K, j \in N, s \in 1, \dots, S \quad (53)$$

$$\sum_{s \in S} \Pr(C_{s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (54)$$

$$C_{1,s,kj} = l_{kj} - f_{kj} - q_{kj} \frac{\omega_k SoC_k^{max} - (SoC_k - e_s d_{kj})}{r_1} \quad \forall k \in K, j \in N_1, s \in 1, \dots, S \quad (55)$$

$$C_{1,s,kj} = l_{kj} - f_{kj} - q_{kj} \frac{\omega_k SoC_k^{max} - (SoC_k - e_s d_{kj})}{r_2} \quad \forall k \in K, j \in N_2, s \in 1, \dots, S \quad (56)$$

$$\sum_{s \in S} \Pr(C_{1,s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (57)$$

$$C_{2,s,kj} = SoC_k - e_s q_{kj} d_{kj} - SoC_k^{min} \quad \forall k \in K, j \in N, s \in 1, \dots, S \quad (58)$$

$$\sum_{s \in S} \Pr(C_{2,s,kj} \geq 0) \geq \beta |S| \quad \forall k \in K, j \in N \quad (59)$$

where constraints (53) - (59) modify the deterministic model by addressing inter-station travel time and energy consumption uncertainties. Constraints (53) 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 (54) require the sum of probabilities of $C_{s,kj} \geq 0$ across all scenarios to exceed a percentage β of total scenarios. Similarly, constraints (55) and (56) define $C_{1,s,kj}$ to account for charging time variability and the remaining battery level, with constraints (57) ensuring the probability of $C_{1,s,kj} \geq 0$ meets β . Constraints (58) and (59) 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 the transition of 30 bus lines ($|K| = 30$) from the greater Athens metropolitan network from 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. Each bus is assumed to require a single charging session per operational day, based on data from the Athens Urban Transport Organization (OASA) as of July 2024. 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). 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. 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 ($\lambda_{2v} = 2$). Figure 65 in the Annex of this deliverable presents the detailed layout of the bus network utilized in this case study.

The model framework was examined under both deterministic and stochastic approaches. Table 2 presents the optimal physical locations and types of chargers selected for installation in each approach, while Table 3 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 2: Optimal charging station physical location (V), and charging types ($N=N_1 \cup N_2$) across the deterministic and all $\beta(\%)$ variation approaches

Deterministic Approach			$\beta(\%)$ 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 3: 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

A commonality among the deterministic, stochastic, and chance-constrained approaches 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 $V = \{3, 7\}$. 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 3). 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.

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 3. 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 2, 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 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.

The aforementioned stochastic EB-CSLP modelling 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”.

Combined Line Planning Modification and Vehicle Scheduling for a fleet of Electric Buses

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 modification, which arises from the elimination of stops,

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 select the optimal the modification of each bus. Our model, a mixed-integer linear program (MILP), indicates the optimal bus line layout, while considering the passengers' demand for each stop and the available number of buses. The goal of the overall two-stage approach is to minimize the unserved passenger demand that arises from the possible modifications. The paper regarding the model, which is described above, is currently under review to a scientific journal and hence some changes have been conducted compared to the version presented in D2.1. On the following pages we present the updated version of the combined line planning modification and vehicle scheduling problem.

Related studies

The 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-MDVSP). 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 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 LPM. 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 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^k . 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 , 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 $Time_k$. The $o_k \in O$ 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. A generic graphical representation of the modelled problem is provided in Figure 67 in the Annex of this deliverable.

The formulation presented in Figure 26 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 complete mathematical formulation is described in Subtask 2.6.1 in the Annex of this deliverable.

For the modification line planning formulation, we consider a number of lines $\mathcal{L}$. 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 LPM is pre-determined. The mathematical formulation for this is as follows:

$$\min \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{O}} x_{ij} d_{ij} \quad (60)$$

Subject to the constraints:

$$\sum_{j \in \mathcal{O}} x_{ij} = 1 \quad \forall i \in \mathcal{L} \quad (61)$$

$$\sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{O}} x_{ij} o_{ij} \leq B \quad (62)$$

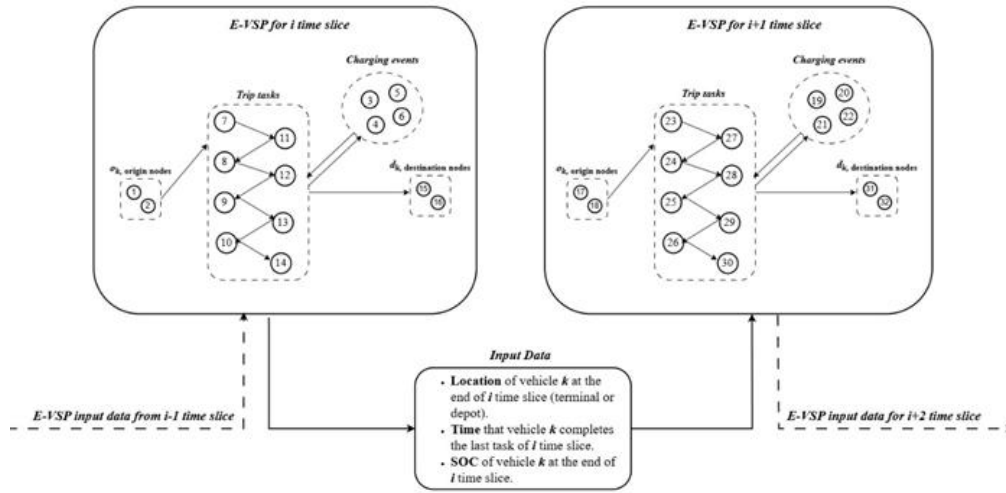


Figure 26: A generic representation of the time-slice modeling approach. Each time slice solves the E-VSP problem based on the location, energy level and time status of a vehicle k provided by the previous time-slice.

The objective function (60) aims to minimize the total number of unserved passengers due to the modification of the line plans. Constraints (61) and (62) 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 model 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 modification line planning 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 LPM models presented in this section, the two-stage approach can be depicted in Figure 27.

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 66 in the Annex of this deliverable. 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.

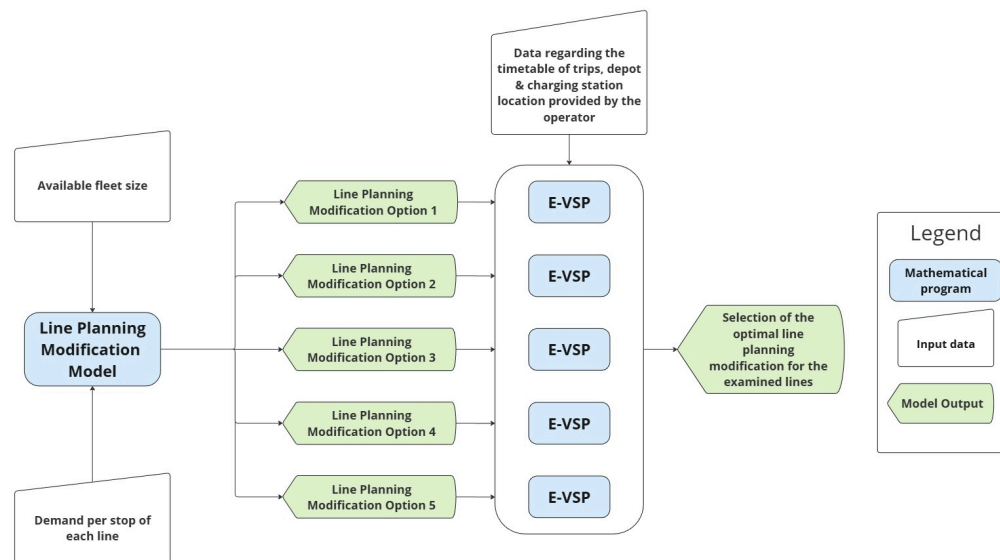


Figure 27: A flowchart of the two-stage process including the E-VSP and LPM models

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 modification line planning 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 4, we showcase the results of our model for an available fleet consisting of 52 electric buses

Table 4: 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^s 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 $23 \rightarrow 7 \rightarrow 32 \rightarrow 13 \rightarrow 38 \rightarrow \text{c.e.} \rightarrow 81 \rightarrow 63 \rightarrow 88 \rightarrow 70 \rightarrow 94 \rightarrow 76$. 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 28 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.

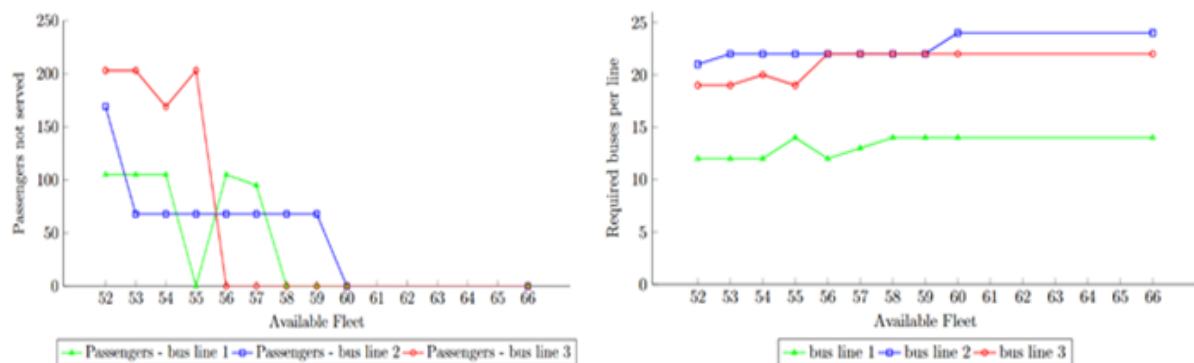


Figure 28: Graphic representations of the number of passengers not served when considering a range of available vehicles of the number of required buses and the number of passengers not served considering the size of the available fleet for every bus line examined

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.

Vehicle Scheduling component of the Electric Bus Multi-Depot Vehicle Scheduling Problem with Time Windows

This subtask has expanded previous research works to create an Vehicle Scheduling for Electric Bus networks with Multiple Depots and Time Windows (EB-MD-VSP-TW) model that can account for real-world networks of electric buses. To achieve this, the project team has developed a continuous time representation at the chargers while previously a discrete time slots formulation was adopted. In addition to that, while traditional models assume that buses can enter and exit the depots that are modelled in the mathematical formulation, the newly proposed model has a maximum capacity in the number of buses per depot, and the buses that leave from one depot have to return to the same depot. These extra requirements that are described above, have emerged through the several collaborative workshops that the subtask team has engaged with OASA, which is a project partner and an electric bus operator in the Athens LL. Through these discussions the necessary trips and depot data have been acquired, as well as other necessary data for the application of the model.

Objective functions and modelling

The newly derived EB-MD-VSP-TW model is a Mixed Integer-Linear Model (MILP) that assumes an objective function and several constraints for the problem. The objective function is modelled after the goals of the electric bus operator which are considered to be two-fold: i) the first is the minimization of the deadhead distances covered by the buses, and ii) the second is the minimization of waiting times of vehicles at either of the nodes of the network. For that reason, the objective function for the EB-MD-VSP-TW can be formulated as follows:

$$\min \sum \sum C_{ij}^k \cdot x_{ij}^k + \sum \sum \lambda \cdot w_{ij}^k \quad (63)$$

In mathematical expression (63), one may notice two terms with the following parameters and decision variables:

- C_{ij}^k which the cost of traversing a distance from point i to point j by bus k,
- x_{ij}^k which is the decision variable of whether a bus k should traverse a distance from point i to point j,
- λ Which is the cost of waiting at one location, and,
- w_{ij}^k which is the waiting time at point j when a bus k traverses the distance from i to j.

Given this objective function, the model seeks to find the optimal scheduling of vehicles from depots to the several trip that need to take place within the electric bus network.

Application to Athens LL and prospective implications for the LL

While this model is still under development, the research team has not yet applied it to the real-world data acquired by OASA. Despite this fact, the model has been tested on a series of well-received test instances for EB-MD-VSP-TW, and the first results that were acquired show that the new model can provide solutions to the problem that are better solution performance, as

compared to previous scientific model that assumed a discrete time slot modelling at the charging stations. This mean that in the real-world application in metaCCAZE the solution can potentially have more value for the electric bus operator. However, the continuous time model has demonstrated a caveat which is that it requires more computation time, when compared to the discrete time model.

With that being said, it is expected that the subtask team will be able to derive an actual scheduling solution for OASA, which can be applied and tested. This solution will potentially include better scheduling solutions (to-be scenario) which will lead to less vehicles (i.e., electric buses) being required for the execution of the same number of trips as in the current scenario (as-is).

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:

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right), \quad x > 0 \quad (64)$$

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 68 in the Annex of this deliverable. 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-León 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-León et al. (2024), 130 s.

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}).

$$\widehat{t}_i^{\text{PUDO}} = \mathcal{U}\left((1 - \delta) \cdot t_i^{\text{PUDO}}, (1 + \delta) \cdot t_i^{\text{PUDO}}\right) \quad (65)$$

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 metaCCAZE'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 metaCCAZE 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

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.

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 11 in the Annex of this deliverable. Simulations were conducted on the Linux Cluster of the Leibniz Supercomputing Centre.

The impact of the PUDO duration is assessed based 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.

We start by exploring 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 29 (a) and (b), 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.

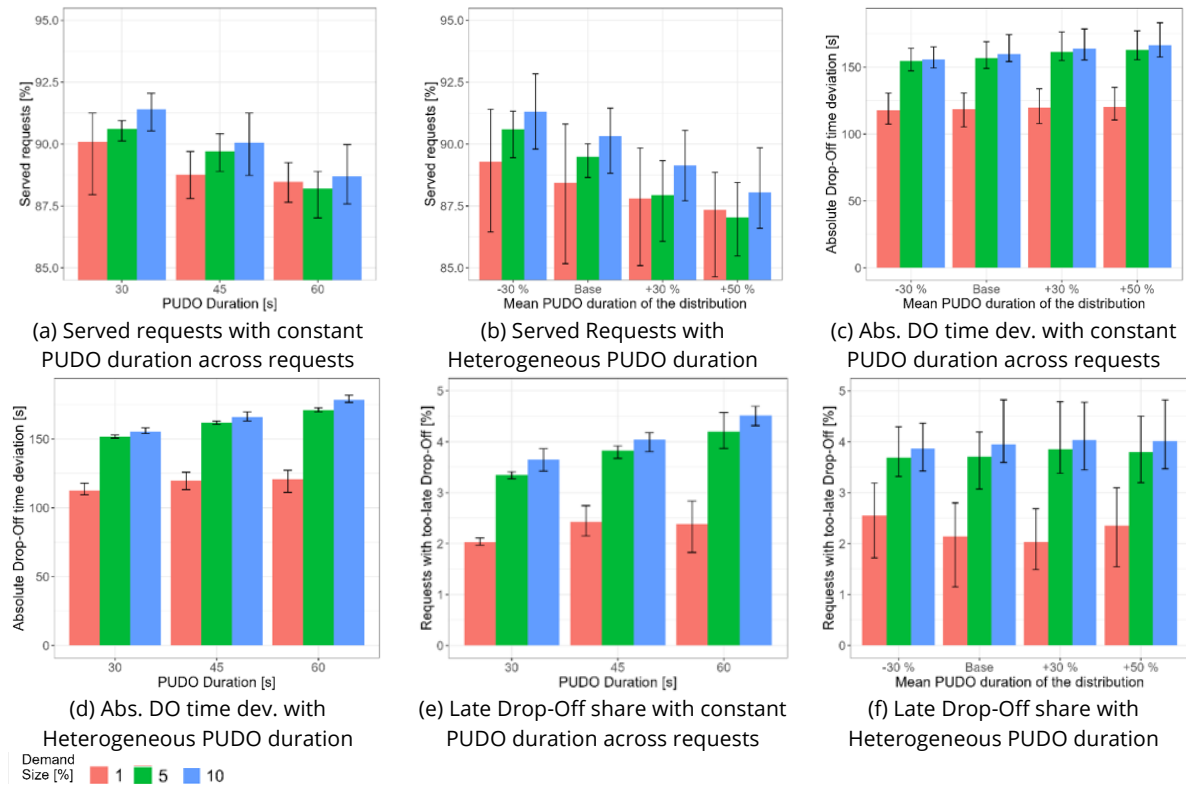


Figure 29: Impacts of increasing PUDO duration considering fully-informed operators

As shown in Figure 29, 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.

Regarding the effect of PUDO duration stochasticity, 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 30 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. 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%.

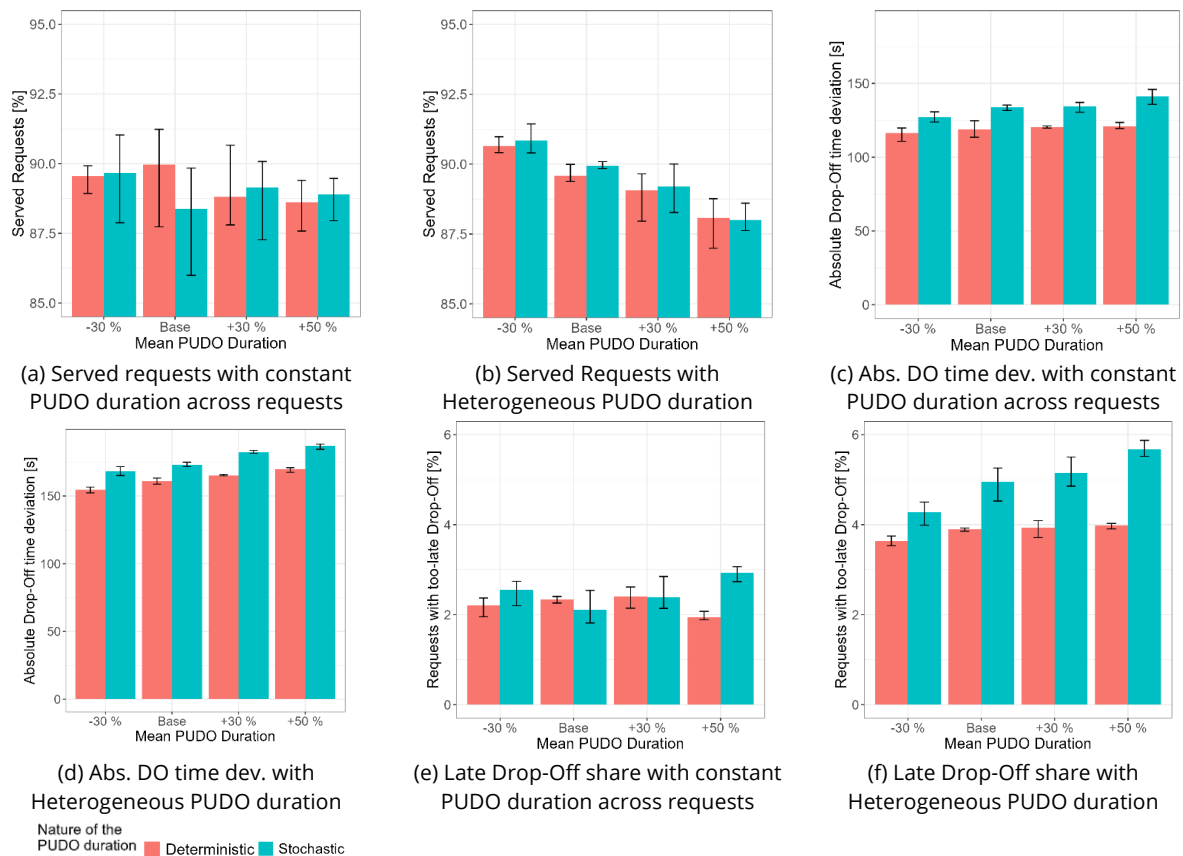


Figure 30: Effects of Predicting PUDO Duration for Each Request

From the perspective of the metaCCAZE 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. The results discussed in this subsection are a short subset from this research effort. The impact of using different prediction models for PUDO duration, as well as the effects of using biased versus informed constant PUDO duration values, are briefly described in Subtask 2.6.1 in the Annex of this deliverable, and a more thorough presentation of the results is available in the Álvarez-Ossorio et al. (2025).

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

Introduction and Related Studies

A unique feature of Mobility-On-Demand (MOD) vehicles compared to regular traffic is that they perform frequent short-term traffic stops for passenger pick-up and drop-off (i.e., PUDO processes). These typically occur either at the curbside (using dedicated bays, similar to taxi stands, or regular parking spots) or directly on the carriageway through double parking (DP). In the former case, MOD vehicles must leave and rejoin the main traffic stream, potentially causing disturbances and delays; in the latter, a driving lane is (partially) obstructed, resulting in a temporary bottleneck (Vishnoi and Simoni 2025). While empirical evidence on the impact of these processes on the overall transport system performance remains limited (Hanekamp et al. 2023; Liu et al. 2023; Stueger et al. 2023), PUDOs share characteristics with other traffic disruptions such as public transport stops, delivery vehicle halts, and double parking—all of which have been shown to negatively impact network capacity and exacerbate congestion (Cao et al. 2016; Cao and Menendez 2015; Johari et al. 2020; Kladeftiras and Antoniou 2013). As vehicle automation advances, leading to a reduction in operational costs, the demand for Autonomous Mobility-On-Demand (AMOD) services is anticipated to grow substantially. Consequently, the movement of AMOD vehicles and their PUDO processes are likely to have a critical impact on traffic flow across significant portions of the transport network.

In the context of metaCCAZE, 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).

This subsection summarises the results of a study that investigates PUDO processes in an Autonomous Mobility-On-Demand (AMOD) service with walking legs, where passengers are asked to walk to designated stop locations rather than being offered a door-to-door service. We provide a methodology aimed at supporting city planners in making informed decisions regarding: (i) how much curbside space should be allocated to PUDO processes across the network; (ii) how this space should be distributed among different streets; and (iii) where double parking for PUDOs should be prohibited. The remainder of this chapter presents a concise description of the research described in Álvarez-Ossorio Martínez et al. (2025). First, we present our proposed solution approach, followed by its application to a specific case study. Due to space constraints in this deliverable, we will not present the extensive formulation and solution approach in detail. Instead, we provide a higher-level description and refer the interested readers to the scientific manuscript.

Methodology

This research applies a modular framework with three components: an AMOD model incorporating PUDO location and regulation constraints, a static traffic assignment sensitive to link-

level PUDO frequencies, and an upper-level optimization that employs a metaheuristic to identify optimal PUDO regulation strategies. Given the novel nature of this problem, a general description of the problem (CA-PUDO-NDP) is defined in the Annex of this deliverable.

Considering the objectives and decision variables of our framework, the optimization problem can be categorized as an integer, non-linear, multi-objective optimization problem. Additionally, the objectives cannot be evaluated easily but require simulation-based methods. Given the challenge of simultaneously considering conflicting objectives of such a diverse nature, we propose to aggregate the driving times (O1) and walking times (O2) objectives into a single objective function, considering weights that represent the relative valuation of in-vehicle and walking times (VOT^{drive} and VOT^{walk}), respectively. An additional objective (O4) is included in the objective function of the upper optimization problem (multiplied by a large factor M) to act as a penalty that guarantees that the number of simultaneous PUDO processes allocated to a specific link does not exceed its maximum available capacity (which is given by the length of the link and whether double parking is allowed or not). Our proposed solution approach results in the following problem:

$$\min_{x_l, n_l} O(x_l, n_l) = \left[\sum_{r=1}^R t_r^{drive}(t_l) + \sum_{d=1}^D t_d^{drive}(t_l) \right] VOT^{drive} + \left[\sum_{r=1}^R t_r^{walk} \right] VOT^{walk} + M \quad (66)$$

$$s.t. \quad (n_{od}^{AMOD}, f_l^{curb}, f_l^{dbl}, t_r^{walk}, \delta_p^l, \omega_p^t) \leftarrow VRP(x_l, n_l; t_l^{exp}; R) \quad (67)$$

$$t_l \leftarrow TA(n_{od}^{AMOD}, n_{od}^{PV}, f_l^{curb}, f_l^{dbl}) \quad (68)$$

$$\sum_{l \in L} n_l = N_{city} \quad (69)$$

$$0 \leq n_l \leq n_l^{max} \quad \forall l \in L \quad (70)$$

$$n_l \in \mathbb{N} \quad \forall l \in L \quad (71)$$

$$x_l \in \{0,1\} \quad \forall l \in L \quad (72)$$

Conversely, we propose to account for the objective of minimizing the curbside space dedicated to PUDOs (O3) by including it as a scenario variable. Additionally, constraints define n_l as an integer, lower or equal than the maximum physical number of curbside stop areas in each link, n_l^{max} . We chose not to include (O3) in the objective function as reliable monetary estimates for the value of curbside space are difficult to obtain. Moreover, this objective differs in nature from the other two, as it represents an externality of the transport system. Instead, through the exploration of different values of N_{city} , it will be possible to approximate the value from which the marginal improvement in the objective function is irrelevant. Alternatively, policymakers can assess how pursuing a given “social” (O3) objective impacts the objective of transport system participants. The above problem has two relevant characteristics. First, the objective function does not have a closed mathematical form, but it depends on the computationally expensive evaluation of the AMOD and Traffic Assignment modules. Second, the solution space, S , is very large, since, $|S| = \prod_{l=1}^L 2(n_l^{max} + 1)$ and L can be large for real networks. Therefore, exploring exhaustively the whole space through brute force is computationally not feasible. To face these challenges, we propose modifying the Adaptive Large Neighborhood Search (ALNS) metaheuristic (Ropke and Pisinger, 2006) to the context of our problem. In particular, we adapt the notion of *neighborhood* of a solution and develop tailored destroy and repair methods.

As illustrated in Figure 31, the ALNS metaheuristic is built around the two other modules of the CA-PUDO-NDP. In the following, we provide a high-level overview of the framework. To guide readers through the flowchart, we map its key components with the explanation in the next paragraph using numbers in **blue**.

Given a network (its topology and n_l^{max} values), the iterative process for a certain *problem*—defined by a given N_{city} constraint, the AMOD and Private Vehicle (PV) demands—takes the following steps.

First [1], at the initialization of the process, the AMOD and PV demand are combined into a total trip demand, and given to an (initial) traffic assignment, which calculates the expected link travel times (t_l^{exp}), used by the AMOD module during the whole optimization process. Then [2], from the solution space of the problem—determined by the network—an initial *feasible* solution is sampled. Feasibility is given by a link-level constraint, the allocation of up to n_l^{max} curbside spaces to every link, and a network-level constraint, the allocation of a total of N_{city} spaces in the whole network. Next [3], at iteration 0, the objective value of this initial feasible solution is calculated and this initial solution is destroyed by one of the implemented destroy methods.

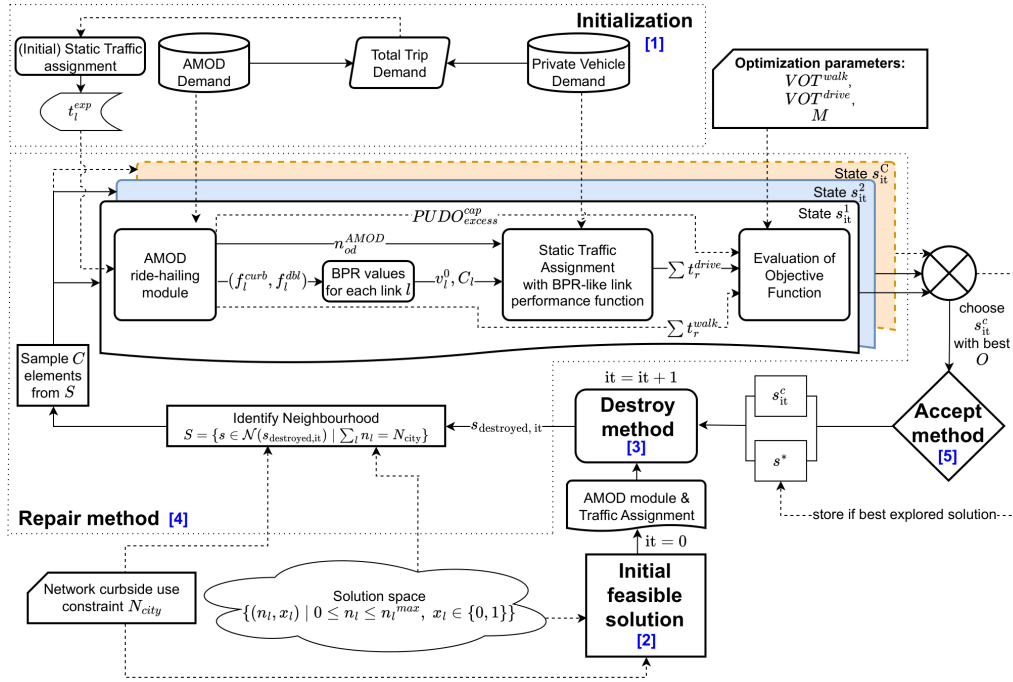


Figure 31: Overview of our proposed solution approach of the CA-PUDO-NDP based on the ALNS metaheuristic.

The destroyed solution is then given to one of the repair methods [4]. Each of these identifies the neighborhood of the destroyed solution following a different approach. In most cases, the neighborhood contains a large number of elements, making exhaustive exploration impossible. For this reason, depending on the available computation resources (e.g., the number of CPUs), a number C of candidate neighbors are sampled from the neighborhood, and their objective function is evaluated concurrently. The evaluation of the objective function (also for the above-described initial feasible solution) consists of solving the VRP problem by the AMOD module, calculating the value of $PUDO_{excess}^{cap}$ (used for the calculation of the penalty term in O), updating the link performance function of each link based on their PUDO frequencies (in the curbside and as double parking), and conducting the static traffic assignment. Then, given the study-specific VOT^{drive} and VOT^{walk} , the objective value for each candidate solution is obtained, and the best candidate is chosen as repaired solution. Finally [5], following the standard ALNS approach, depending on the implemented acceptance criterion, either the repaired solution, s_{it}^c , or the best already-visited solution, s^* , will be given to the destroy method for the next iteration. This iterative process will continue until a pre-defined stop condition is met (e.g., maximum number of iterations or running time, or the best solution has not been improved after a number of iterations). Further notes on the methodology are provided in Subtask 2.6.1 in the Annex of this deliverable.

Case study

This section presents a selection of relevant results discussed in the original manuscript. The methodology was applied to 33 different scenarios considering different AMOD demand level values (1%, 2.5%, 5%, and 7.5% of the total trips), as well as maximum N_{city} values. The values of travel time were adapted from Schmid et al. (2018), Alvarez-Ossorio Martínez et al. (2025), and Kolarova et al. (2019). The Sioux Falls network (B Stabler 2025), a standard benchmark in transport research, was considered. The duration of PUDO processes was defined as 30 seconds.

Given the nature of the CA-PUDO-NDP the optimal solution cannot be known without exhaustively exploring the solution space. Moreover, the objective value can never reach zero—objective $O1$ will have contributions as soon as there are travelers in the network—making it difficult to assess the quality of the solutions produced by the ALNS. To address this, we define two benchmarks for each AMOD demand level: the Status Quo (SQ) and the Lower Bound (LB). The former, which represents a scenario in which AMOD services meet the specified demand, but no dedicated curbside space is allocated for PUDO activities and double parking is (implicitly) allowed everywhere, can be loosely interpreted as an upper bound, although, strictly speaking, it is not. The latter aims to isolate the effects of PUDOs on traffic from those of AMOD vehicle movements and fleet management.

The analysis of the evolution of the objective value across iterations for each studied scenario revealed that, after a rapid improvement within the first few iterations, most cases converge asymptotically. Although the predefined maximum runtime cannot guarantee that the global optimum was reached, the stabilization of objective values suggests that quasi-optimal solutions were obtained and supports the notion that our proposed solution approach is valid.

In Figure 32, we show the Pareto frontier between citywide curbside provision ($O3$) and the combined system objectives ($O1$, $O2$, $O4$). Within a given AMOD demand level, the objective value consistently improves as the number of allocated curbside spots in the network increases. The curves are not perfectly monotonic, suggesting that some scenarios may have become trapped in local optima before the running time limit was reached. In real-world applications, where precisely mapping the Pareto frontier might be more important, additional N_{city} values and longer computation times could be explored. Notably, the marginal improvement from adding curbside spots diminishes: the first allocations bring substantial gains, but the benefit then becomes small or even negligible. A key policy-related takeaway is that, for the studied network, providing more than about 100 curbside spots offers no further advantage regardless of the AMOD demand.

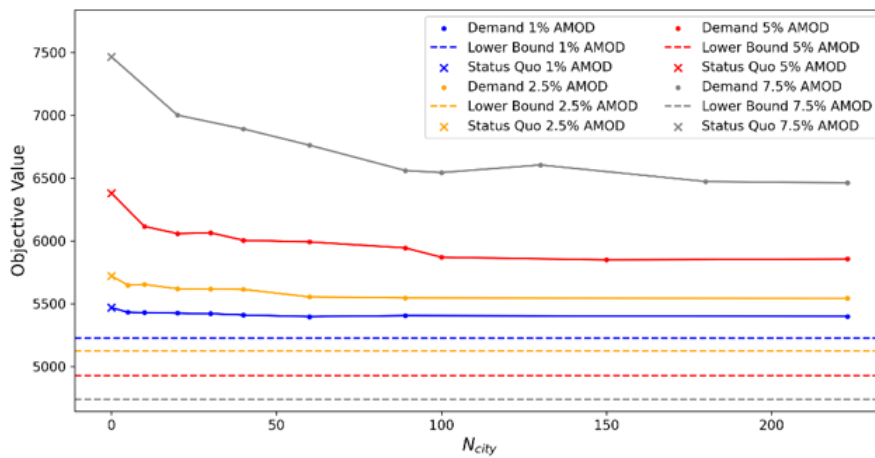


Figure 32: Pareto frontier between N_{city} and the other objectives.

When comparing results across demand levels, low AMOD penetrations (1—2.5%) have limited PUDO-related traffic impacts, the gap to the LB is small, and increasing N_{city} yields minimal

improvement. In contrast, higher AMOD demands, with their greater number of PUDO processes, result in much higher objective values. However, in these cases, allocating a higher N_{city} brings substantial improvements. These findings suggest that, for low AMOD demand levels, authorities could forgo dedicated curbside PUDO locations with minimal system-wide cost, whereas at higher penetration rates, curbside provision can deliver significant benefits.

Figure 33 illustrates the distribution of network speeds for different combinations of AMOD demand and total number of curbside spaces. As AMOD demand increases, average speeds decline, driven by the growing number of PUDOs and their impact on the link performance functions. Conversely, allocating more curbside spaces increases average speeds by reducing the share of PUDOs handled through double parking. Overall, the results in the figure show that providing no curbside spaces and allowing double parking everywhere, as in the SQ scenario, can significantly reduce network speeds. Interestingly, higher demand not only reduces the average network speed—for example, in the SQ scenario, increasing AMOD demand from 1% to 7.5% lowers the average speed by almost 8 km/h, while the corresponding reduction in the scenarios with $N_{city} = 223$ is 5.7 km/h, but also affects the speed range. Specifically, higher demand slightly lowers the maximum speeds and substantially reduces the minimum speeds, which can drop by more than 10 km/h when demand rises from 1% to 7.5%. When examining each demand group individually, we observe that increasing the number of allocated curbside spaces in the network generally raises speeds. This effect is minor for low AMOD demand but becomes more pronounced at 5% and 7.5% demand levels. For instance, in the 7.5% AMOD scenario, providing $N_{city} = 100$ curbside spaces increases average and minimum speeds by 2 km/h and 2.8 km/h, respectively, compared to the SQ scenario.

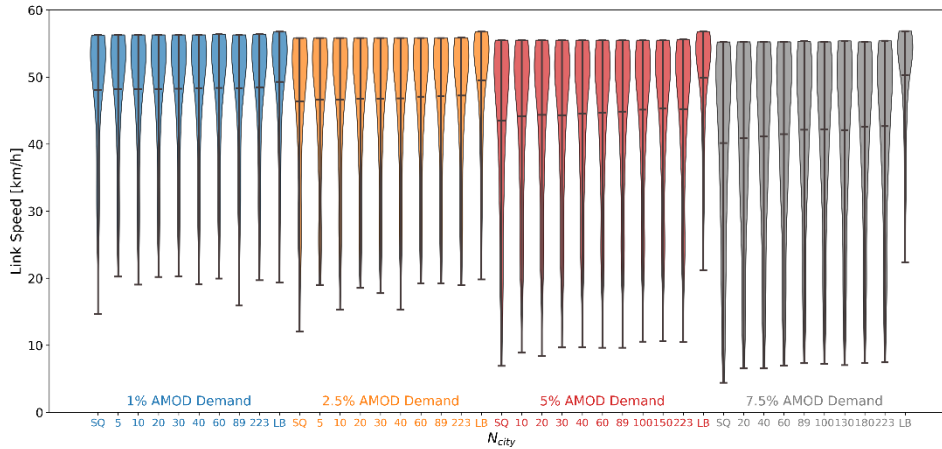


Figure 33: Link speed distribution by scenario. SQ: Status Quo; LB: Lower Bound.

To conclude the analysis of results, we discuss the resulting PUDO policies and their relationship with network speeds. Figure 34 A and Figure 34 B display the policies for $N_{city} = 223$ and $N_{city} = 40$, respectively. Intuitively, the $N_{city} = 223$ case assigns higher n_l values across a greater number of links. In addition, providing more curbside spaces increases the number of links where double parking is prohibited. This reflects the trade-off between objectives $O1$ and $O2$: since the value of time for walking is significantly higher than for driving, the optimization seeks to minimize $O2$ without substantially worsening $O1$. Thus, under low N_{city} values, more links allow double parking with none or only a few curbside spaces—ensuring that AMOD requests can be served without walking—whereas at higher N_{city} levels, double parking (which has larger adverse impacts on surrounding traffic) is avoided whenever possible. In both scenarios, there is only one link—located at the bottom right of the network—where PUDOs are entirely forbidden (i.e., $x_l = n_l = 0$).

Figure 34 C illustrates the link speeds for the $N_{city} = 223$ scenario. Compared with Figure 34 A, it becomes clear that links with lower speeds (darker colors in the lower plot) tend either to be allocated more curbside spaces (lighter colors in the upper plot) or to be completely forbidden for PUDOs (the aforementioned $x_l = n_l = 0$ case). Figure 34 D shows the differences in link speeds between the $N_{city} = 223$ and $N_{city} = 40$ scenarios. Blue shades indicate speed improvements under $N_{city} = 223$, while red shades indicate reductions. The majority of links experience faster speeds when more curbside spaces are provided, with only a small number showing decreases. This results from the higher prevalence of double-parking PUDOs in the $N_{city} = 40$ scenario, which alters link performance functions and subsequently affects the traffic assignment. Moreover, speed increases (darker blue) are generally larger in magnitude than speed decreases (lighter red). Interestingly, links with higher speeds in Figure 34 C tend to show little variation in Figure 34 D (their colors remain close to white), and these are also the links that receive lower n_l allocations in Figure 34 A.

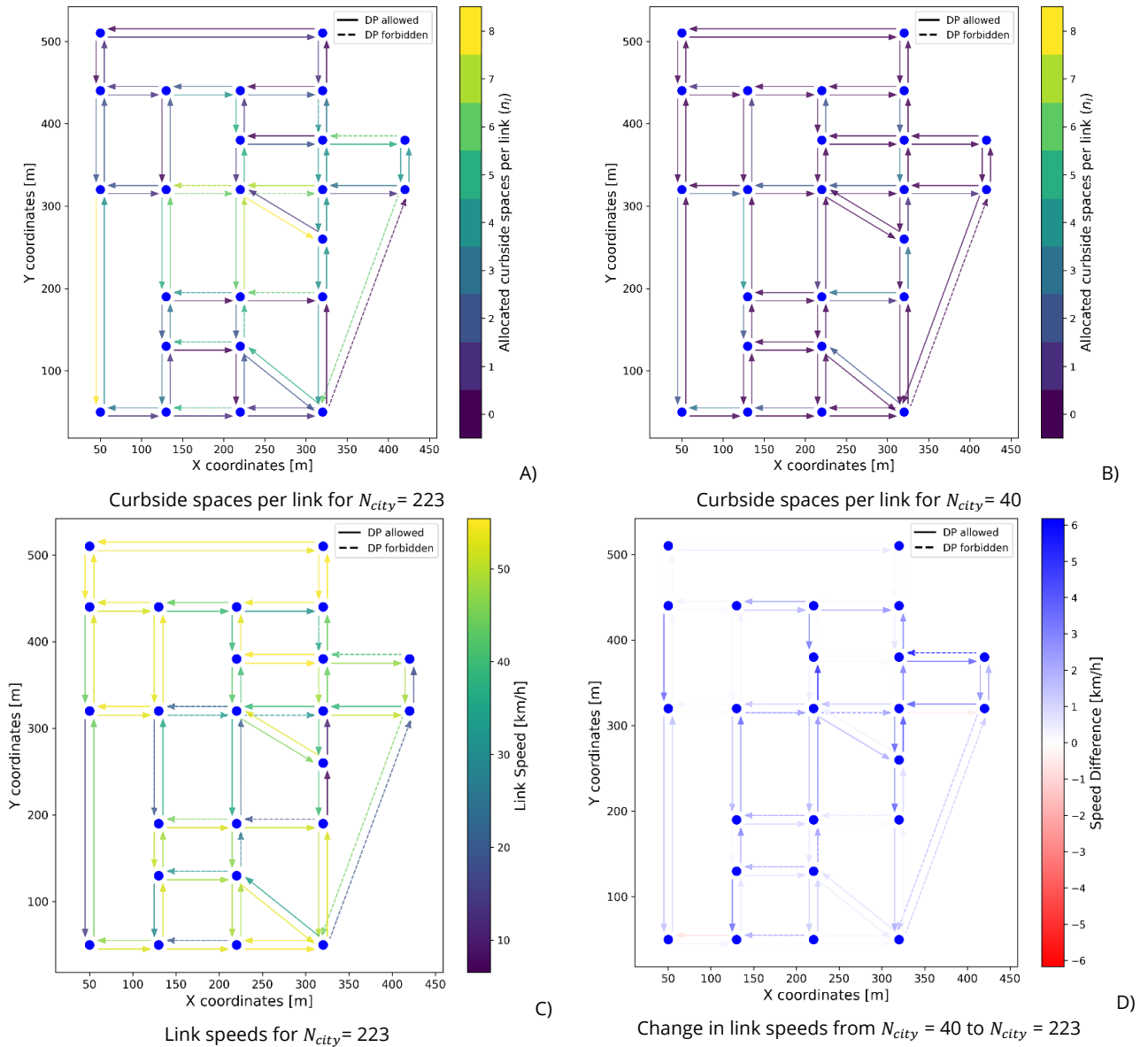


Figure 34: PUDO policies and link speeds for 7.5% AMOD demand under different N_{city} . (DP: Double Parking.)

The work described in this subsection introduced a new network design problem: the CA-PUDO-NDP. In the context of the worldwide expansion of mobility-on-demand services, this problem is

both timely and relevant. It also speaks to the interests of multiple stakeholders, including AMOD users, private vehicle travelers, and society at large. More details can be found in Álvarez-Ossorio Martínez et al. (2025).

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

The system integrates a road network with low-capacity electric vehicles and waterways with medium- and large-capacity barges. This can be modeled as a multi-echelon (multi-level) routing problem. We consider time-dependent travel times to model varying traffic levels throughout the day. 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. Thus, vehicle charging can be disregarded in the model.

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 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 $V = V_0 \cup V_s \cup V_c$ consists of a central depot and processing facility $V_0 = 0$, satellites V_s and customers V_c . The set of arcs A contains two subsets, $A = A_1 \cup A_2$, where $A_1 = A(V_0 \cup V_s)$ 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. $u_s^k \in \mathbb{Z}^+$ is position of satellite s in route of vehicle $k \in K_1$. $w_s^k \in \mathbb{R}^+$ 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 $f_{ij}^s \in \mathbb{R}^+$ denotes load on arc $(i, j) \in A_2$ to satellite s . $a_i^k \in \mathbb{R}^+$ denotes arrival time of vehicle $k \in K_2$ at customer i . δ_{ij}^{sh} 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-TDIT is formulated as follows:

$$\min \sum_{(i,j) \in A_1} \sum_{k \in K_1} t_{ij}^k \cdot x_{ij}^k + \sum_{(i,j) \in A_2} \sum_{s \in S} \sum_{h \in H} t_{ij}^{2h} \cdot \delta_{ij}^{sh} \quad (73)$$

$$\sum_{(i,j) \in A_1} x_{ij}^k = \sum_{(j,i) \in A_1} x_{ji}^k \quad \forall i \in V_0 \cup V_s, \forall k \in K_1 \quad (74)$$

$$\sum_{(i,j) \in A_1} x_{ij}^k \leq 1 \quad \forall i \in V_0 \cup V_s, \forall k \in K_1 \quad (75)$$

$$u_i^k - u_j^k + |S| x_{ij}^k \leq |S| - 1 \quad \forall (i, j) \in V_s, i \neq j, \forall k \in K_1 \quad (76)$$

$$w_s^k \leq Q_1 \cdot \sum_{(i,s) \in A_1} x_{is}^k \quad \forall s \in S, \forall k \in K_1 \quad (77)$$

$$\sum_{s \in S} w_s^k \leq Q_1 \quad \forall k \in K_1 \quad (78)$$

$$\sum_{k \in K_1} w_s^k = \sum_{(s,j) \in A_2} f_{sj}^s \quad \forall s \in S \quad (79)$$

$$\sum_{(i,j) \in A_2} y_{ij}^s = \sum_{(j,i) \in A_2} y_{ji}^s \quad \forall i \in V_s \cup V_c, \forall s \in S \quad (80)$$

$$\sum_{(s,j) \in A_2} y_{sj}^s \leq K_2 \quad \forall s \in S \quad (81)$$

$$\sum_{s \in S} \sum_{(i,j) \in A_2} y_{ij}^s = 1 \quad \forall j \in V_c \quad (82)$$

$$\sum_{(j,i) \in A_2} f_{ji}^s - \sum_{(i,j) \in A_2} f_{ij}^s = D_i \quad \forall i \in V_c \quad (83)$$

$$f_{ij}^s \leq Q_2 \cdot y_{ij}^s \quad \forall (i, j) \in A_2, \forall s \in S \quad (84)$$

The objective function (73) minimizes total travel time. Constraints (74) - (78) are related to the first echelon. Constraints (74) ensure flow conservation. Constraints (75) impose that each vehicle visits each satellite at most once. Constraints (76) prevent subtours. Constraints (77) and (78) ensure that vehicle capacity is not exceeded. The first and second echelons are linked by constraints (79), which ensure flow balance at satellite. Constraints (80) - (84) are related to the second echelon. Constraints (80) ensure flow conservation. Constraints (81) limit the number of second echelon vehicles at each satellite. Constraints (82) ensure that each customer is visited exactly once. Constraints (83) ensure that demand is met. Constraints (84) limit vehicle capacity.

To account for time windows, constraints (85) and (86) below are added to the model. Constraints (85) ensure that arrival time at customer i is within the allowed time interval. Constraints (86) define arrival at customer i as the sum of arrival at customer $(i - 1)$, service time, and travel time on arc $(i - 1, i)$.

$$e_i \leq a_i^s \leq l_i \quad \forall i \in V_c, \forall s \in S \quad (85)$$

$$a_j^s \geq a_i^s + s + t_{ij}^s - M(1 - y_{ij}^s) \quad \forall i, j \in V_c, s \in V_s, i \neq j \quad (86)$$

To account for time-dependent travel times, constraints (87) - (89) are added to the model, and constraint (86) is replaced by (87). Constraints (87) 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 (88) ensure that exactly one time period is selected for each trip between i and j . If time interval h is active, constraints (89) enforce that departure time from customer i is within the time interval.

$$a_j^s \geq a_i^s + s + \sum_{h \in H} t_{ij}^h \cdot \delta_{ij}^{s,h} - M(1 - y_{ij}^s) \quad \forall i \in V_c, \forall j \in V_c, \forall s \in S, i \neq j \quad (87)$$

$$\sum_{h \in H} \delta_{ij}^{s,h} = y_{ij}^s \quad \forall (i, j) \in A_2, \forall s \in S \quad (88)$$

$$\theta_h \cdot \delta_{ij}^{s,h} \leq a_i^s + s \leq \theta_{h+1} \cdot \delta_{ij}^{s,h} + M(1 - \delta_{ij}^{s,h}) \quad \forall h \in H, \forall (i, j) \in A_2, \forall s \in S \quad (89)$$

The model is implemented in Python using gurobipy to access Gurobi Optimizer and validated on a small test instance. The Branch-and-price framework (see Figure 75 in the Annex of this deliverable), is being adopted to solve the two-level vehicle routing problem with time windows under realistic traffic conditions (time-dependent travel times).

Pilot

In parallel with the development of the 2E-VRP-TW-TDTP, TUD developed an algorithm to optimize a waste collection pilot currently being conducted by the city of Amsterdam in the 9 Straatjes neighborhood. The objective is to evaluate a new waste collection system that combines collection by appointment with a scheduled waste collection service.

An extensive analysis of the current needs and challenges of the city in the area of waste collection and the 9 Straatjes pilot operations was conducted in the initial phase of the project. Based on the initial analysis and with respect to the capacity of the city to deploy and/or integrate new solutions within the timeline of the project, TUD developed a prototype algorithm to optimize the scheduled waste collection service in the 9 Straatjes pilot area. A timeline for testing and demonstration is being finalized. The demonstration is expected in 2026.

7.2. Subtask 2.6.2 Low emission-based traffic management

In subtask 2.6.2, Vedecom develops and validates a comprehensive traffic management model that integrates the communication technologies introduced in subtask 2.5.2. This subtask focuses on demonstrating the efficiency and effectiveness of the predictive modeling approach through rigorous scientific validation, presenting results from the Transportation Research Board (TRB) Annual Meeting 2026, and establishing the framework for V2X-related evaluation based on key performance indicators (KPIs).

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.

TUD advanced equitable Tradable Mobility Credit (TMC) schemes by developing an unsupervised tree structure framework with contextual layers to create personalized user profiles, overcoming limitations of classical clustering methods that failed due to feature dominance issues. The approach sequentially clusters travelers using context-specific feature sets—distance traveled, sociodemographic factors (income, children), and travel mode usage—from Netherlands Mobility Panel 2022 Amsterdam data, enabling targeted credit allocation that differentiates sustainable travelers from behavior-change candidates for more equitable mobility interventions.

Predictive modeling with multi-source data

Literature Review Multimodal Transportation Demand Prediction Using Deep Learning

The advancement of deep learning methodologies has substantially enhanced transportation demand forecasting capabilities, with particular emphasis on multimodal urban systems. Recent developments have introduced sophisticated approaches that integrate public transit and micromobility services within unified predictive frameworks (Hua et al. 2025). Empirical evidence indicates strong operational coupling between subway ridership and micromobility usage, particularly within first-mile and last-mile contexts, as documented in metropolitan areas including New York City, Singapore, and Paris (Di et al. 2025; Ma et al. 2025).

Contemporary research demonstrates that deep learning architectures effectively capture temporal and spatial interdependencies in transportation networks. Hua et al. (2025) pioneered transfer learning methodologies for cross-modal demand prediction between bike-sharing and public transit systems, achieving robust performance in cross-city generalization scenarios. However, their framework presupposes infrastructural similarity across urban contexts, which may limit broader applicability. Multi-task learning frameworks have proven particularly effective for complex spatio-temporal modeling. Ke et al. (2021) developed a multi-task, multi-graph learning approach for ride-hailing systems that successfully models spatio-temporal dependencies across diverse service types. Despite computational efficiency concerns and extensive data requirements, this methodology captures critical interactions within heterogeneous mobility networks. Similarly, Huang et al. (2024) applied multi-task learning techniques to metro demand forecasting, effectively capturing inter-station dependencies, though performance deteriorates under non-stationary demand conditions such as major events. Addressing data privacy considerations, Li and Liu (2024) introduced federated learning frameworks for multimodal demand forecasting in rail systems, eliminating centralized data aggregation requirements. While this approach maintains privacy standards, convergence challenges emerge with non-independent and identically distributed data distributions across participating entities.

Hybrid architectures combining convolutional and recurrent neural networks have demonstrated enhanced forecasting accuracy. Noursalehi and Koutsopoulos (2022) proposed a CNN-LSTM architecture incorporating wavelet transforms for dynamic origin-destination prediction, delivering high accuracy for systems with automated fare collection infrastructure. Zhang et al. (2024) expanded this paradigm by integrating contextual multimodal features—including weather conditions and road typologies—into graph-based deep learning models, though performance sensitivity to missing or inconsistent inputs remains a constraint. Environmental variables exert substantial influence on transportation demand patterns and mode choice decisions. Weather conditions, particularly precipitation and temperature extremes, significantly affect micromobility usage while potentially increasing subway ridership as travelers seek enclosed transportation alternatives (Gebhart and Noland 2014; Thomas et al. 2012). Scott and Ciuro (2019) documented

that temperature variations alter walking distances to subway stations, consequently affecting multimodal integration patterns.

Proof of Concept: New York City Case Study

The ST-GAT framework has been validated through a comprehensive proof-of-concept study using a full-year (2022) dataset from New York City. This large-scale validation provides empirical evidence of the model's capabilities and establishes baseline performance metrics that inform the metaCCAZE deployment strategy. The empirical foundation combines 30.2 million City Bike trips from 1,704 stations, 4-hourly MTA subway turnstile entries from 517 stations, and hourly weather observations from the Central Park station. This harmonized panel spans 365 days with 6 uniform 4-hour periods per day (2,190-time steps), enabling consistent origin-based demand measures across both modes.

The NYC case study demonstrates pronounced seasonal fluctuations, with bike volume peaking 23% above winter baseline in summer and dropping 31% below peak during winter, while subway ridership remains more stable but shows heightened sensitivity to adverse weather. The diversity of weather conditions, service levels, and transfer accessibility within this dataset validates the framework's capability to model non-stationarity, spatial spillovers, and cross-modal elasticities across varying operational contexts. Within the heterogeneous graph construction, a 200m walking radius reveals 2,847 potential bike-subway transfer pairs across the network. The connectivity architecture implements a two-tier design: intra-modal subway connections using distance-weighted adjacency with a Gaussian kernel (threshold 2.5km), and inter-modal transfer connections with binary weighting within comfortable walking distance (threshold 0.2km). This design captures over 90% of observed bike-subway transfer patterns while maintaining computational efficiency.

The ST-GAT methodology and its validation through proof-of-concept have been presented to the international scientific community through a paper submitted to the Transportation Research Board (TRB) 104th Annual Meeting 2026. The paper, titled "A Spatio-Temporal Graph Attention Network for Joint Multimodal Demand Forecasting," addresses a critical gap in transportation forecasting literature by jointly modeling subway and micromobility demand within a unified framework. The paper demonstrates that the ST-GAT approach provides a scientifically validated foundation for the metaCCAZE Smart Mobility Hub implementation. The methodology presented in the TRB paper directly informs the predictive models that will be deployed in the Poissy and Triel-sur-Seine use cases, adapted from the NYC proof-of-concept to the Île-de-France context.

Evaluation Methods for Deep Learning in Traffic Prediction

Statistical error metrics serve as the primary tools for evaluating deep learning models in traffic prediction applications. These metrics quantify the divergence between predicted outputs and observed actual values, providing quantitative assessments of model performance. The most widely utilized metrics in contemporary literature include 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). Researchers favor these particular metrics due to their interpretable nature and their capacity to capture diverse dimensions of predictive accuracy.

A notable characteristic of these metrics is their inverse relationship with training dataset size. As the volume of training data expands, prediction errors tend to diminish, thereby strengthening the model ability to generalize unseen data. The mathematical formulations and interpretive roles of these evaluation metrics are presented below.

- **Mean Absolute Error (MAE)** : MAE ($MAE = (1/n) \sum |y_i - \hat{y}_i|$) quantifies the average absolute magnitude of prediction errors across all observations, independent of error direction. This metric has gained widespread adoption owing to its straightforward interpretation and expression in the same units as the target variable, making it particularly accessible for practical applications.
- **Mean Square Error (MSE)**: MSE ($MSE = (1/n) \sum (y_i - \hat{y}_i)^2$) amplifies the penalty for larger prediction errors through its squaring operation, rendering it particularly sensitive to outliers in the dataset. This characteristic makes MSE valuable for applications where substantial deviations between predicted and actual values warrant special attention.
- **Root Mean Square Error (RMSE)**: RMSE ($RMSE = \sqrt{(1/n) \sum (y_i - \hat{y}_i)^2}$) represents the square root of MSE and provides error measurements in identical units to the target variable. Practitioners frequently prefer RMSE in scenarios where large prediction errors carry particularly significant consequences, as it maintains MSE sensitivity to outliers while offering more intuitive interpretability.
- **Mean Absolute Percentage Error (MAPE)**: MAPE ($MAPE = (100/n) \sum |y_i - \hat{y}_i| / |y_i|$) articulates prediction accuracy as a percentage, facilitating performance comparisons across disparate datasets or model architectures. However, researchers should exercise caution when applying MAPE, as it exhibits numerical instability when actual values approach zero.
- **Coefficient of Determination (R²)**: R² ($R^2 = 1 - [\sum (y_i - \hat{y}_i)^2 / \sum (y_i - \bar{y})^2]$) quantifies the proportion of variance in the target variable that the model successfully explains. Higher R² values indicate superior model fit to the observed data, establishing this metric as fundamental in regression analysis. In these formulations, y_i and $\hat{y}_i$ represent the actual and predicted values respectively, while n denotes the total number of samples in the evaluation set.

Results: Comparative Performance Analysis: ST-GAT vs. Baseline Models

Comprehensive experimental results from the NYC proof-of-concept demonstrate the ST-GAT model's superior performance compared to established baseline methods across multiple evaluation metrics. The evaluation framework encompasses traditional statistical models (ARIMA, VAR), machine learning approaches (MLP, CNN), and state-of-the-art graph neural networks (GNN-LSTM), providing a thorough assessment of the model's capabilities and establishing its readiness for deployment in the metaCCAZE use cases. The results presented in Table 5 indicate that the ST-GAT model achieves exceptional performance with the lowest MAE (0.553) and MAPE (39.29%), representing a 60.1% MAPE improvement over the best baseline model (VAR at 98.40%). This demonstrates clear advantages for resource allocation and operational planning. The model's MAE is 8% better than GNN-LSTM (0.600) and 11% better than CNN (0.620), while its MAPE is 58% better than GNN-LSTM (93.97%), confirming that graph-based transformer architectures can simultaneously deliver superior proportional and absolute accuracy compared to all tested alternatives.

Table 5: Overall Performance Comparison

Model	TYPE	MAE	MAPE (%)	RMSE	R ²
VAR	Statistical	0.761	98.40	1.026	-0.031
ARIMA	Statistical	0.775	154.76	0.923	0.006
CNN	Neural Net	0.620	112.63	0.931	0.157
GNN-LSTM	Graph NN	0.600	93.97	0.918	0.023
ST-GAT (Proposed)	Transformer	0.553	39.29	0.974	0.462

Key Performance Insights:

- 12–18% reduction in RMSE over leading baselines across all evaluation scenarios.
- Most significant gains during high-variance conditions (extreme weather, holidays, special events).
- Strong correlation (0.749) between predicted and actual demand across full range of usage levels.
- Robust performance in low-to-medium demand ranges (0-3000 units) representing majority of real-world scenarios.

The ST-GAT model establishes overall dominance by achieving the best MAE (0.553) and MAPE (39.29%), representing a 60.1% MAPE improvement over the best baseline (VAR) and thus offering clear advantages for resource allocation. Its lower MAE than all neural baselines confirms that graph-based architectures can simultaneously deliver superior proportional and absolute accuracy.

Results: Mode-Specific Performance of Bike-Sharing vs. Subway Predictions

The ST-GAT model exhibits sophisticated mode-specific capabilities, adapting to the distinct demand characteristics of bike-sharing and subway systems. The heterogeneous graph design processes intra-modal and inter-modal connections independently, enabling effective capture of mode-specific traits mentioned in Table 6.

The higher bike-station activity (89.2% non-zero demand versus 76.3% for subway) provides richer training signals, explaining the slightly better MAE for bike-sharing predictions. The model demonstrates strong absolute accuracy for both modes while maintaining competitive proportional accuracy, validating its capability to handle heterogeneous multimodal networks. The slightly higher MAPE for subway stations (43.8% vs. 38.8%) reflects the greater variability and lower baseline demand at some subway stations, particularly during off-peak periods. The model demonstrates superior absolute accuracy for bike stations while maintaining competitive proportional accuracy across both modes, validating its capability to handle heterogeneous multimodal networks.

Table 6: Mode-specific Metrics

Metric	Bike-sharing Stations	Subway Stations
MAPE (%)	38.8	43.8
RMSE	0.943	0.887
R ²	0.421	0.394
Non-zero demand (%)	89.2	76.3

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 MaaSLab 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. The high-quality dataset that was produced has been provided

to the local partners for further analysis, while a second drone data collection campaign is planned for October 2026 that will further support the local partners.

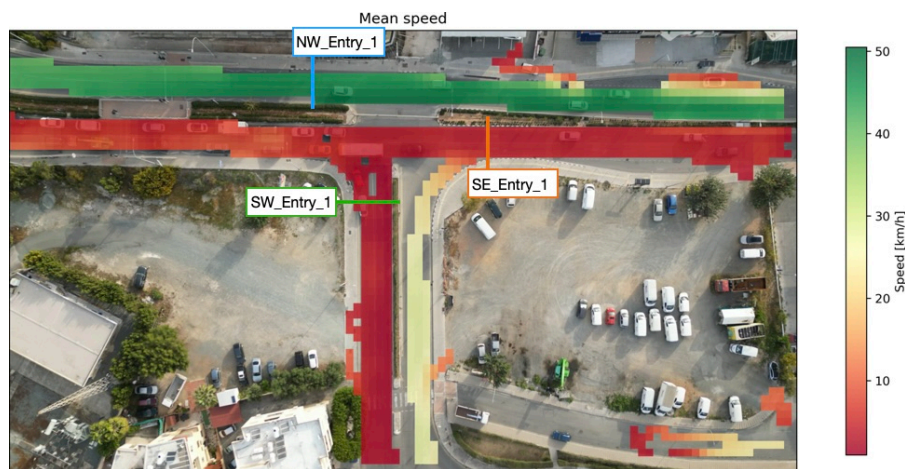


Figure 35: Limassol Speed Loop

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. The flight operations and ability to conduct large-scale drone campaigns were evaluated with success after this campaign, and the collected data is used as a benchmark to continue improving the data quality. A second drone data collection campaign is being planned for spring 2026 and a third later in the project, in collaboration with the local partners.



Figure 36: Paris Speed Loop

In parallel, the Computer Vision team of MobiLysis 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.

While MobiLysis has been advancing its processing pipeline to meet the demanding accuracy requirements of the metaCCAZE use cases, it also conducted a drone data collection campaign in Krakow, the first out of two experiments to be conducted for the specific UC. After discussions with ZTP, six (6) drones were deployed for three (3) days over various points of interest, with emphasis

on the surrounding area of the new Kraków Grzegórzki multimodal hub. Flights were repeated for two sessions, morning and afternoon, to capture variability in bicycle flows and interactions with motorized traffic. This campaign will provide a rich dataset that will support ZTP in evaluating bicycle demand, identifying safety-critical segments, and designing evidence-based improvements for the station surroundings and the nearby zones.

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. However, there are 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 characteristics. 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 user profiles by combining travel behaviour with individual characteristics. These profiles estimate both mobility needs and the potential for behavioural 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. On the other hand, 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. Classic clustering methods in the context of transportation include k-means, hierarchical clustering and latent class analysis. 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, the classical approach that fits this use case 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 cluster number. Furthermore, it is flexible to different cluster shapes depending on the linkage method and provides a dendrogram that can be used for further analysis. However, this classic method failed to create meaningful profiles. The numerical distance features were dominated by the categorical ones. The method created profiles that were similar in terms of children and income but not in terms of distance travelled. Therefore, these profiles can't be used for credit allocation.

The nature of the use case makes it harder to create profiles. This can happen because these features have different contexts. A traveler's income might provide information on the needs they have for traveling, such as cheaper transportation options. However, it isn't a direct translation to

their travel behavior. Therefore, TUD proposes a new approach based on a tree structure framework with contextual layers, where clustering will be applied at each node. Based on this, we want to understand how to create this contextual framework and answer the question “What methodologies can combine travel data with sociodemographic data of travelers to create contextual user profiles based on travel needs and behavior?”.

Methodology

The first step to answer the previous question is to understand which features are relevant to include. These were created based on a dataset of travelers from the city of Amsterdam. For sociodemographic features, income (low, medium, high), and if people have children below the age of 12 are being included. Currently, disability is not being considered because it will be later applied as a restriction in the credit allocation stage. In addition, the methodology also considers travel needs. The information that is being used to create these features comes from travel distances, the main mode of transport, frequency of mode usage, and travel duration. This information can be leveraged to extract features that showcase travel behavior related to commuting and other daily activities. The distance travelled for each traveler is then calculated based on this information. The ratio of different travel modes for each traveler is also estimated. These include active modes, such as biking and walking, public transport and car. The main mode for commuting is also considered. This inclusion of travel mode usage serves to separate travelers who potentially have more opportunities to change behavior, such as those who take the car, from those who are already being sustainable.

As previously mentioned, the clustering approach adopted to create these user profiles is based on unsupervised trees with contextual layers. To apply this idea, the features are grouped into feature sets according to the context of the use case. The unsupervised tree has layers, and each layer is associated with a feature set. As an example of the use case, the feature sets can be distance travelled as one, representing travel distances. Income and kids are another feature set that capture sociodemographic features. Finally, including main transport mode for commuting with bike, public transport and car ratio as a feature set groups travel mode usage.

The algorithm starts in the first layer, where it clusters the sample based only on the feature set associated with it. The resulting clusters will be separated again individually in the second layer. Only that this time they will only be separated based on the second feature set. The process is repeated until the algorithm reaches the end of the layers, and the result will be the resulting clusters of the final layer. Clustering on a single level implies using the agglomerative clustering method with a different number of cluster and linkage method options. The algorithm has a parameter of maximum number of clusters that it can divide at each point. This means that, for example, in a single moment, the algorithm can separate the data up until the maximum number of clusters with a linkage method from the available ones. For mixed or categorical feature sets, the Gower distance is applied, and three linkage methods can be used, the single, average and complete methods. However, for numeric feature sets, the Euclidean distance is used, and four linkage methods can be chosen, the previous ones and the ward method.

For example, an unsupervised tree solution where the maximum number of clusters is three can start by dividing the sample in two clusters (C1 and C2) in the first layer with the ward method. The second layer separates cluster C1 in three clusters (C11, C12, C13) with the complete linkage method and cluster C2 in two clusters (C21, C22) with the average linkage method. Finally, the third layer separates cluster C11 in two clusters (C111, C112), cluster C12 in two clusters (C121, C122) with average linkage method for both and cluster C13 in two clusters (C131, C132) with the complete linkage method. Cluster C21 is separated into three clusters (C211, C212, C213) with the complete linkage method and cluster C22 is separated into two clusters (C221, C222) with the

average linkage method. In the end, the resulting clusters are C111, C112, 121, C122, C131, C132, C211, C212, C213, C221, C222, in total 11 clusters. The compact representation of a solution is [[A-2-w], [A-3-c, A-2-a, 0], [A-2-a, A-2-a, A-2-c, A-3-c, A-2-a, 0, 0, 0, 0]]. The tree representation and the cluster separation can be seen in Figure 76 and Figure 77 in the annex.

Generating a tree solution requires checking for two different validations. The number of clusters must match between lists, and the linkage method must be according to the feature set data type. The ward method can only be used with numeric features, so in case the feature set contains categorical features, it cannot use it. To evaluate the solution, the average of the silhouette score is calculated for each layer. The final score is the average of the silhouette score. This score can go from -1 to 1, where 1 is the best result. In cases where a layer doesn't do any clustering, the score attributed is 0. Furthermore, some configurations are unfeasible. For example, it can happen that the algorithm is supposed to separate a cluster of 2 data points in 3 clusters. This means that the configuration is invalid and the score is -1000.

This concept of different tree solutions was used with a genetic algorithm. For mutation and crossover, the different tree solutions are modified based on the feature layer order, the number of clusters and linkage methods. The algorithm starts by generating a population based on the population size set as a parameter. It then iterates for a certain number of generations, also determined by a parameter. At each generation, the algorithm creates offspring of the size of the population. To generate two trees based on two existing ones, the algorithm selects one random parent and the other based on randomly sampling the population with a determined tournament size and choosing the best solution. Crossover is applied to generate two new trees and there is additionally a mutation probability for each of these new trees. The next generation is decided applying the steady state strategy, by selecting the best half of either the current population, and the offspring. After a certain number of generations not improving the best score, the elements of the population are mutated as an attempt to escape local optimum. The initial results of this approach are presented in the next subsection.

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 required 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. 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 included 396 data points, each representing an individual traveller.

As a starting point, the problem was defined with maximum number of clusters as 3 and the feature sets as distance travelled, income and kids as another feature set and the travel modes features as another one. The population size of the algorithm was defined as 30, the tournament size as 20 and mutation probability as 0.3. The scores of each generation were recorded to generate the convergence graph shown in Figure 37.

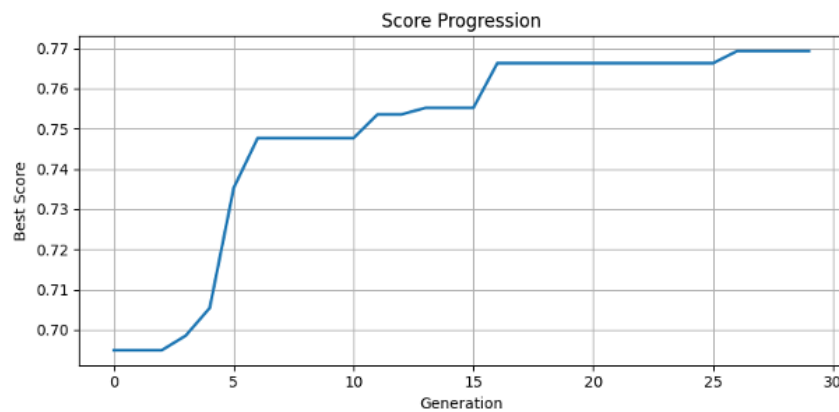


Figure 37: Score of each generation

The algorithm returned a tree solution with a score of 0.769. The resulting clusters can be found in Table 7. Given the contextual layer structure of the algorithm, the interpretation of the decisions taken at each layer is clear as can be seen in Figure 38. By analyzing this separation, it is possible to interpret what each profile represents. As an example, clusters 1 to 7 correspond to travelers who travel less than 245 kilometers over a three-day period. Clusters 1 and 2 represent travelers who prefer active modes of transport: cluster 1 includes travelers without children, while cluster 2 includes those with children.

Returning to our use case of encouraging travelers to adopt more sustainable travel modes, these profiles help us identify which travelers potentially have more opportunities to do so. Some of the clusters identified represent travelers who already exhibit sustainable behaviours. However, for example, clusters 6 and 7 group travelers who don't have children, use the car, and don't travel for long distances compared to the sample. These travelers could be good candidates to receive fewer mobility credits than they need to maintain their current travel behaviour, thereby encouraging a shift towards sustainable travel modes.

Table 7: Resulting Clusters

	CLUSTER	1	2	3	4	5	6	7	8	9	10	11
Indicators	Size	112	9	3	33	16	146	44	7	2	22	2
Distance	Mean	22.24	25.75	110.1	75.38	78.12	68.14	92.68	359.27	363.82	322.58	324
Commuting Mode	Active	103	8	0	10	1	26	5	0	1	6	0
	Car	7	0	0	0	13	120	38	0	0	15	2
	PT	2	1	3	23	2	0	1	7	1	1	0
Car Ratio	Mean	0.07	0.17	0.06	0.05	0.93	0.93	0.92	0.05	0.03	0.92	1
PT Ratio	Mean	0.02	0.05	0.87	0.87	0	0	0	0.94	0.52	0.04	0
Active Ratio	Mean	0.92	0.78	0.07	0.08	0.07	0.07	0.08	0.01	0.44	0.04	0
Kids	Yes	0	9	3	0	16	0	0	0	0	0	2
	No	112	0	0	33	0	146	44	7	2	22	0
Income	Mean*	0.9	1.22	1.33	1.21	1.25	0.75	2	1	2	1.32	1.5

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

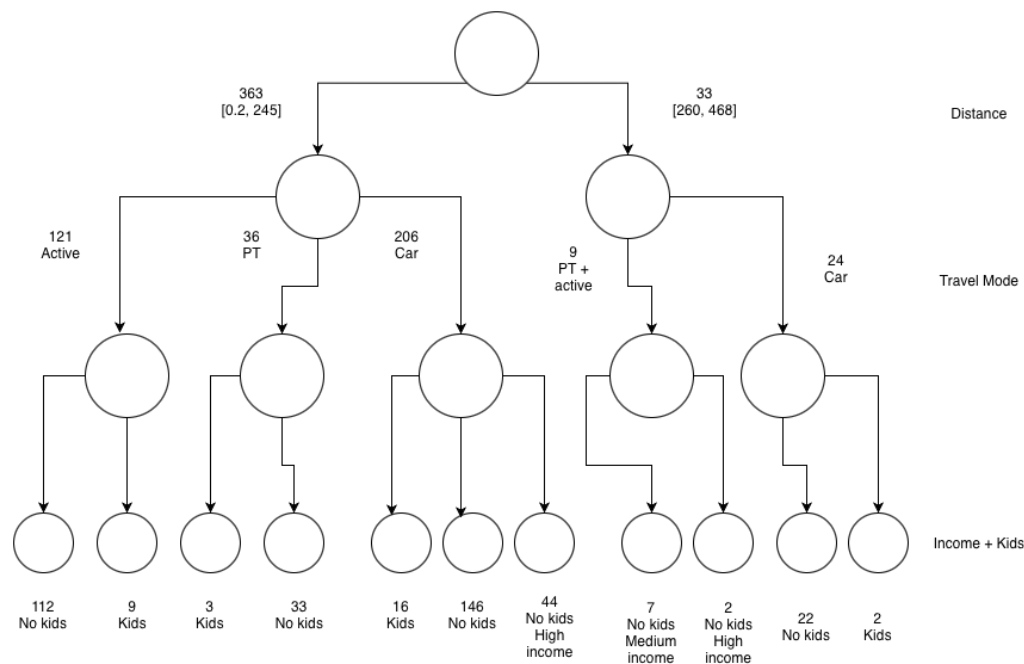


Figure 38: Decision Tree Interpretation

In terms of limitations for these profiles, the algorithm separates the distances in two profiles. For the use case of tradable mobility credits, it makes more sense to separate the travelers with more precision for the distance feature. A more in-depth analysis is required to validate the proposed methodology. The unsupervised tree structure should be better explored and analyse the effects of different maximum number of clusters, different feature sets for the use case and different orders of the feature sets. Additional experiments with different parameter configurations for the genetic algorithm should be carried out. Finally, the results of the algorithm should be compared with the classic methods used in the literature.

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, Vehicle-Driver Assignment, Scheduling, Reporting and Analytics

Sample images from the user app, as well as the operator and driver interfaces, can be found in Figure 39 and Figure 40 respectively.

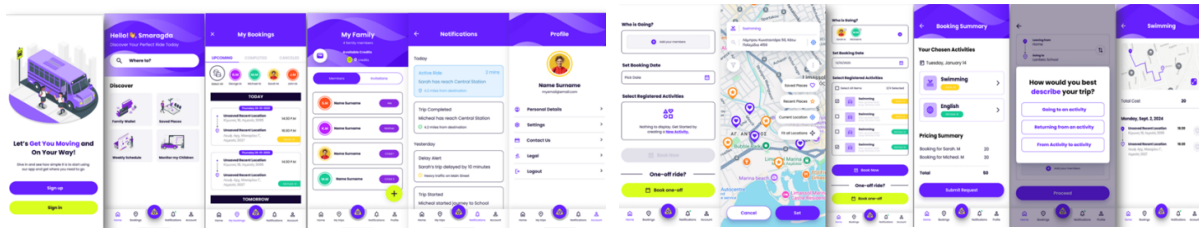


Figure 39: User app interface

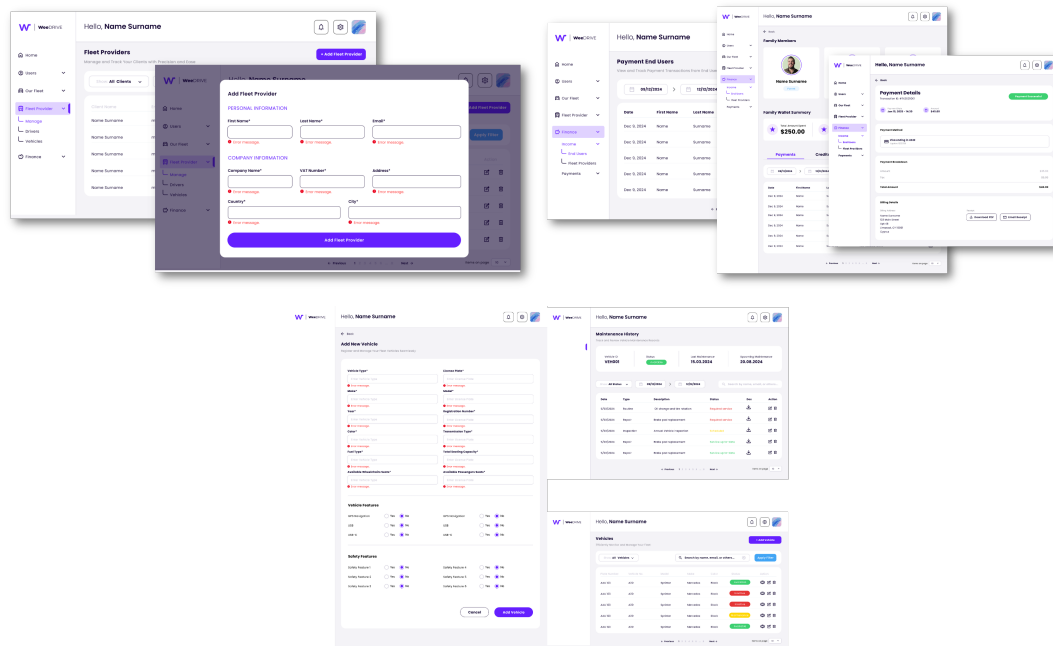


Figure 40: Operator and driver app interface

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 JSRIT, 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 the Simulated Annealing Algorithm (SAA) 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 41.

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.

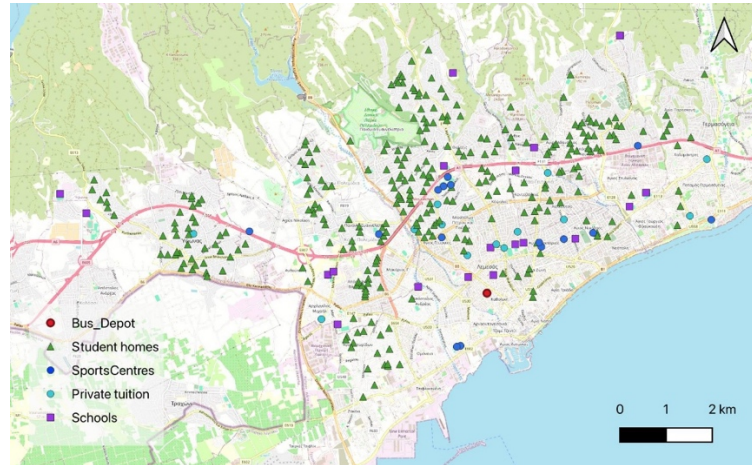


Figure 41: School, home, activity locations selected in Limassol

The preliminary results and operational analysis for the on-demand shared e-service is presented in Figure 42, 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.

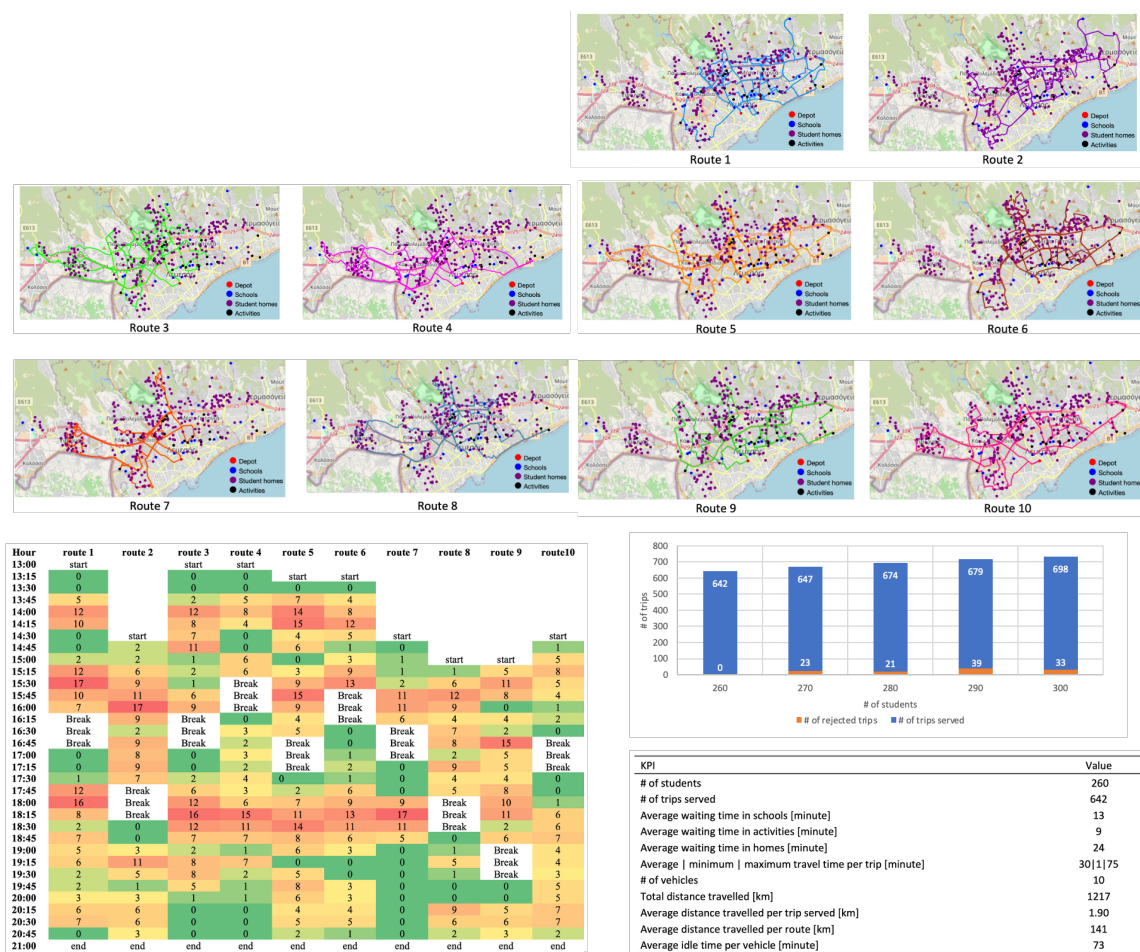


Figure 42: Results for a fleet size of 10 Minibuses with 17-seat capacity

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 42). 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 42 (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.

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 (€)
<i>TaxiF</i>	$3.42 + D0.73$
<i>Pol_1</i>	$\min\{4.5 + D0.5, \textit{TaxiF}\}$
<i>Pol_2</i>	$\min\{6 + D0.3, \textit{TaxiF}\}$
<i>Pol_3</i>	$\min\{16.6/\textit{AOpv} + D0.6, \textit{TaxiF}\}$

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 43. 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.

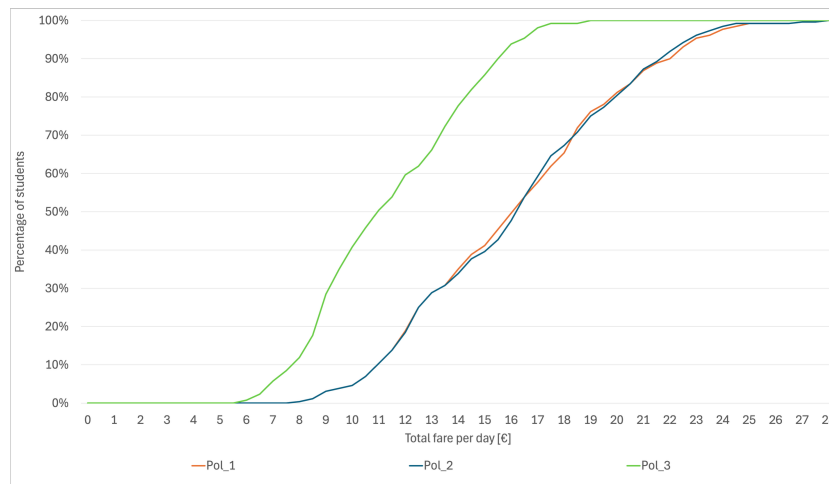


Figure 43: ECDF for the total daily student fare

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.

After an extensive testing period, the WeeDRIVE app went live in October 6 with an initial fleet size of 5 Minibuses. Through the WeeDRIVE platform monitoring & tracking components, we have been collecting route-level and stop-level operational data with stop attributes (location, type: pick-up/drop-off/break, times: prescribed arrival/departure time, tracked arrival/departure time). With data processing, we are able to extract usage indicators, demand patterns and driver & vehicle availabilities. The trip generation stage of our methodology is being updated with the datasets collected to improve realism for our results, e.g. regarding the number of trip requests per day we are able to accommodate.

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

Task 2.7 aims to deliver an integrated planning and simulation platform that consolidates datasets, algorithms, and analytical frameworks developed across WP2. Platform development is pursued through two aligned implementations, one led by MaaSLab and one by Argaleo, which follow a shared conceptual architecture and are designed to support interactive access to geospatial data, scenario visualization, and the integration of modelling outputs. The Digital Twin platforms constitute dynamic digital representations of real-world assets, networks, and selected components of the urban environment, grounded in metaCCAZE datasets, mathematical models, and simulation and experimentation results.

The task addresses the growing need for integrated planning environments capable of jointly addressing transport and energy challenges in urban and metropolitan areas. Advances in real-time data platforms, geospatial analytics, and simulation technologies have accelerated the digitalization of planning and decision-support tools, while policy frameworks at the EU and national levels, increasingly promote integrated approaches across mobility, energy, and sustainability domains. Within this context, Digital Twinning is employed as a mechanism to support the systematic assessment of multidimensional impacts of policies and infrastructure interventions on accessibility, emissions, energy demand, and overall system performance, enabling comparison across alternative planning and operational scenarios.

The conceptualization of Task 2.7 explicitly accounts for persistent challenges associated with Digital Twin implementation, including heterogeneous data availability, interoperability constraints, and the long-term maintenance of live-data connections. These challenges informed the design of the metaCCAZE Digital Twin platforms, which are intended to support transport planners, researchers, operators, and decision-makers in integrating independent models and exploring portfolios of urban and regional interventions for both passenger and freight mobility. Increased emphasis is placed on modularity and extensibility, allowing new datasets, models, and indicators to be incorporated progressively without requiring major structural changes.

Digital Twinning in the Amsterdam Living Lab

In the Amsterdam Living Lab, the digital twin implementation has been further extended and refined compared to the previous reporting period, with a specific focus on urban waste logistics and population-driven waste production modelling. Building on the established spatial foundation of the Amsterdam digital twin, additional datasets, modelling logic, and analytical capabilities were integrated to support more detailed planning and optimisation of waste collection systems. The digital twin continues to rely on a comprehensive geospatial base, including detailed road and street networks, street typologies, bridges, waterways, and physical constraints that shape vehicle accessibility and routing possibilities within the city. In the current reporting period, this network representation was further utilised to explicitly model waste collection routing, accounting for street-level restrictions, vehicle accessibility, canal segments, and other infrastructural barriers that influence operational feasibility and efficiency.

A key extension of the Amsterdam digital twin concerns the systematic use of demographic data to model waste generation patterns. Neighbourhood-level demographic indicators such as population density, household composition, and age distribution were incorporated to estimate spatial differences in waste production (see

Figure 44). These data were linked to residential and mixed-use areas to provide a more realistic representation of waste volumes and temporal generation patterns, enabling scenario-based analysis of collection demand across the city. In parallel, building-level data were further operationalised for waste logistics modelling. Information on building function (e.g. residential, commercial, public), occupancy characteristics, and spatial distribution was used to refine assumptions on waste production intensity. This allowed the digital twin to move beyond static bin locations and towards a population- and activity-driven representation of waste flows, supporting more accurate demand estimation for collection routes and service frequencies.

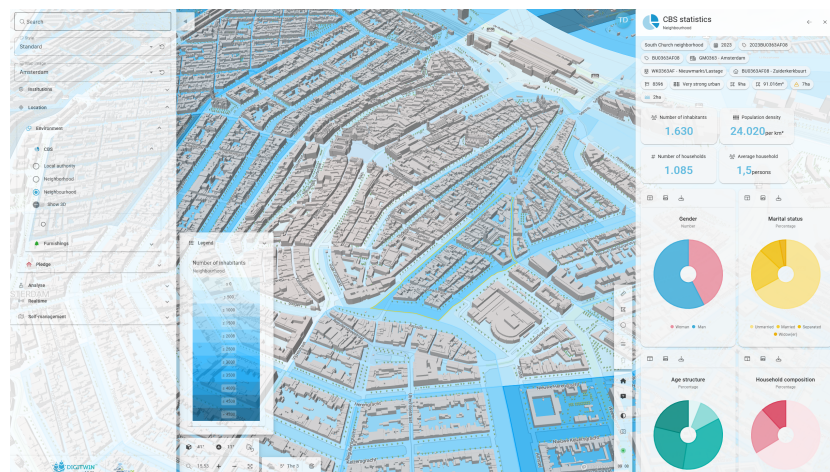


Figure 44: Digital Twin Amsterdam: Demographic data visualizations.

Specific waste-related infrastructure layers were integrated and harmonised within the platform, including the spatial location of above- and underground waste containers, collection schedules, and known routing constraints for service vehicles (see Figure 45). These layers enable the evaluation of alternative routing strategies, assessment of network bottlenecks, and exploration of optimisation scenarios under realistic urban constraints. The combined network and demand modelling supports analysis of trade-offs between service efficiency, operational costs, and spatial coverage.



Figure 45: Digital Twin Amsterdam: Waste logistics location information

All datasets and model outputs are visualised and managed within Argaleo's proprietary digital twin environment, which supports the integration of both static datasets and model-driven results from WP2 activities. The platform offers an interactive geospatial interface through which municipal stakeholders can explore waste logistics scenarios, compare routing strategies, and assess the spatial impacts of policy or operational changes. Overall, the extended Amsterdam digital twin demonstrates how demographic data, urban networks, and waste logistics models can be combined within a single decision-support environment. This evolution strengthens the role of the digital twin as a planning and analysis tool for urban services, supporting more data-driven, adaptive, and context-aware governance of waste collection systems in dense urban environments.

Geoportal: Digital Twin hybrid for Limassol

During the second reporting period, Task 2.7 focused on integrating the datasets and real-time feeds required for the operational Geoportal – Digital Twin hybrid of Limassol and on establishing a coherent environment for planning and simulation activities. The work concentrated on assembling available spatial, mobility, and energy-related information into a single interactive interface capable of supporting evidence-based assessments and exploratory analysis of urban and transport systems. A unified data structure was created to host three-dimensional building information obtained through Mapbox's 3D tiles, enabling a city-scale representation of the built environment with consistent geometry and spatial reference. Public transport supply was integrated through EMEL's GTFS feed, including both route geometries and stop locations, allowing detailed visualisation of the bus network across the metropolitan area. Electric vehicle charging infrastructure was incorporated using datasets provided by the Electricity Authority of Cyprus, including station locations and socket-level attributes, enabling spatial assessment of charging availability in relation to transport and land-use patterns. datasets were formally requested from

the charging infrastructure providers, including charging-usage profiles describing temporal patterns of charger utilisation and detailed pricing information covering cost structures by time of day, charging mode, and tariff scheme. However, these datasets could not be made available during the reporting period due to data-access and provider-level limitations, and therefore the current implementation is limited to static spatial and technical information. In parallel, population density grids based on national census statistics were integrated to support demographic analysis, while land-use zoning layers from the Cyprus Department of Land and Surveys were added to represent functional urban structure and planning designations. Figure 46 shows examples of the visualisation interfaces available in the Limassol Digital Twin.

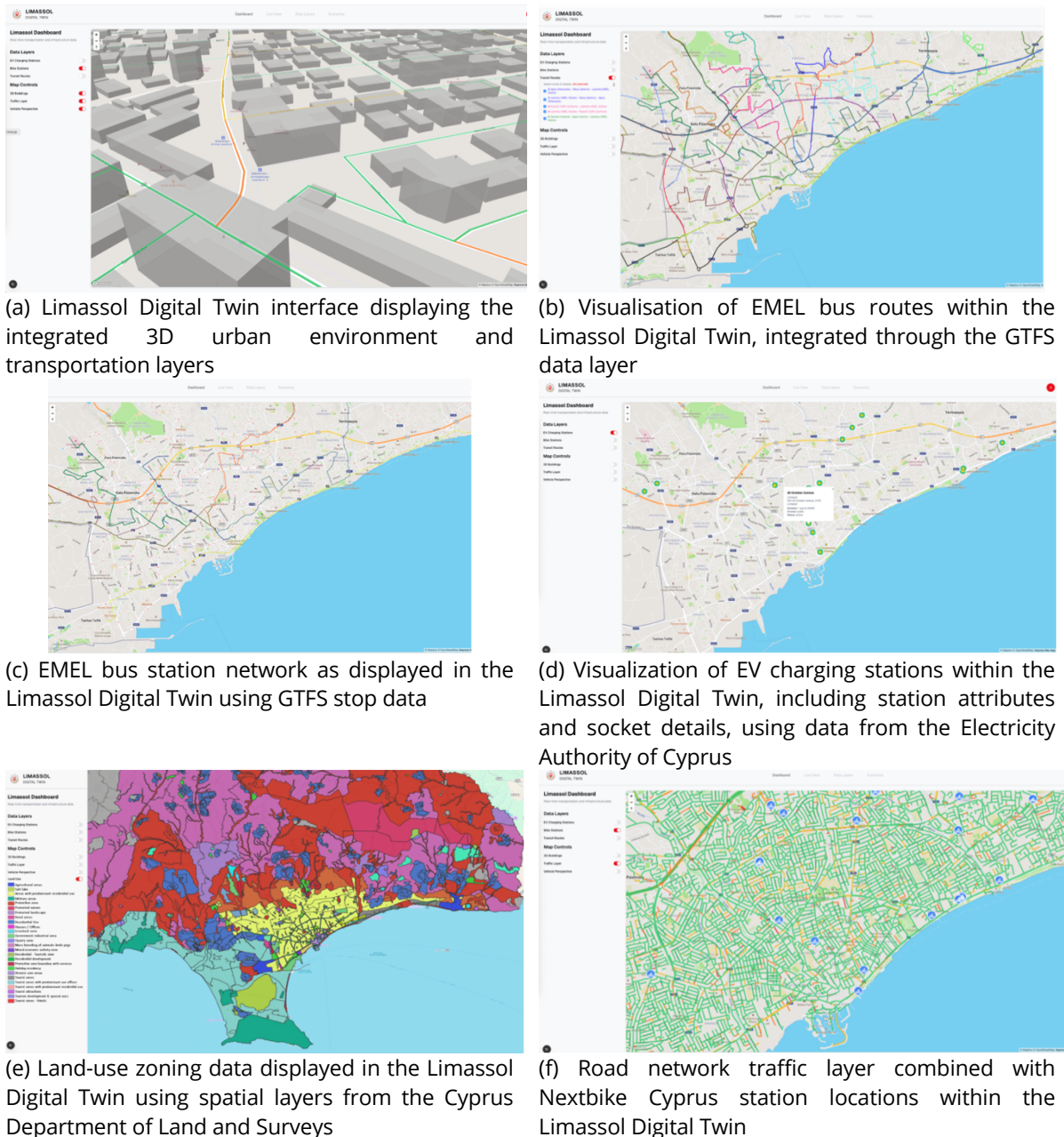


Figure 46: Selected views of the Limassol Digital Twin interface

Multiple live data sources were connected to support near real-time monitoring of selected urban systems. Bike-sharing availability was linked through the Nextbike Cyprus API and visualised together with the road network, enabling joint inspection of micromobility services and traffic conditions. Real-time parking occupancy and status information were ingested from municipal

parking sensors and displayed within the Digital Twin interface to support monitoring of parking supply and utilisation patterns. Environmental and weather data streams were also connected; however, several of these sensor datasets exhibit reliability issues due to inconsistent maintenance and calibration. These limitations result in data gaps, unavailable historical records, and irregular update frequencies, which directly affect the continuity and robustness of real-time monitoring within the Digital Twin environment.

In addition to the datasets already integrated, live traffic events are planned to be incorporated through the [DATEX II Traffic Events](#) feed provided by the Cyprus Department of Public Works. This dataset includes information on road closures, lane reductions, construction works, traffic accidents, temporary diversions, and scheduled maintenance activities. Integrating these feeds will allow the Digital Twin to reflect dynamic network conditions more accurately and support more detailed operational analyses, including re-routing, incident awareness, and scenario testing under disrupted network conditions.

All datasets were harmonised to ensure consistency in coordinate reference systems, data formats, metadata structures, and update routines, enabling seamless visualisation and analytical processing within the Digital Twin environment. The integration process also revealed several operational constraints affecting long-term Digital Twin maintenance, including sensor calibration problems, API outages, inconsistent data standards across providers, and limited access to selected municipal datasets. These constraints create interruptions in the synchronisation between real-world assets and their digital representations and were systematically documented to inform improvements in future data governance and maintenance processes.

Finally, preparatory steps were undertaken for the analytical components to be developed in the next phase of Task 2.7. Planning activities focused on the potential integration of land-use and transport interaction (LUTI) models, accessibility metrics, and micromobility demand exploration within the Digital Twin environment. In particular, initial specifications considered the possibility of modelling spatial interaction flows related to daily activity patterns, such as home-to-work commuting as well as home-to-school and home-to-extra-curricular activities, subject to data availability and quality. These flows could be represented using alternative transport modes, including private vehicles, public transport, and on-demand transport services for school and activity trips, drawing on selected elements and data structures of the WeeDrive system where feasible. Initial LUTI flow visualisations were produced to illustrate zone-to-zone interactions within the Limassol area (Figure 47), serving as a preliminary exploration rather than a validated model.

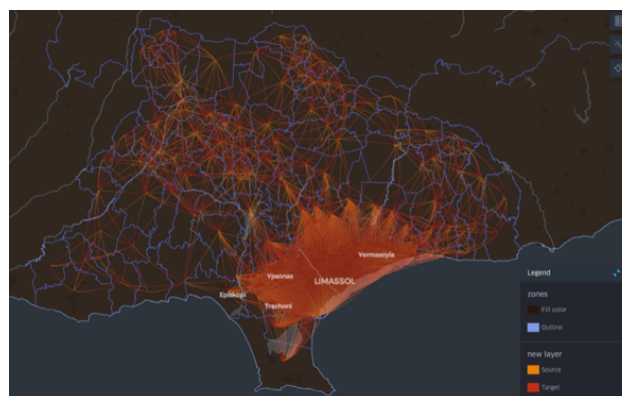


Figure 47: Initial draft of LUTI model flow patterns illustrating zone-to-zone interactions in the Limassol area

In parallel, the incorporation of additional static and historical datasets was explored to enrich the analytical basis of the Digital Twin. This includes the planned integration of sex- and age-disaggregated demographic data to support more differentiated accessibility and equity analyses, as well as historical traffic statistics, including records on fatal accidents, collisions, and victims published by the Cyprus Police. These datasets are expected to support future road-safety assessments, spatial risk mapping, and scenario exploration, subject to data accessibility, spatial resolution, and consistency with the existing data architecture. Initial specifications were defined to ensure that any such analytical components and datasets can be integrated into the existing Digital Twin environment without requiring major structural changes.

9. Conclusion

Deliverable D2.2 has documented the refinements and further developments of the metaInnovations Toolkit carried out in WP2 between July 2025 and January 2026. Across the seven technological domains that constitute the backbone of the metaCCAZE project — Align, Harmonise, Charge, Automate, Connect, Manage & Control, and Integrate + Plan — the work reported in this deliverable consolidates the technical foundations required for the deployment of zero-emission shared mobility (ZESM) Use Cases in the Trailblazer Living Labs and Follower cities.

Under Task 2.1 (Align), the decision-support models for grid-fleet-demand optimisation and the AI-based recommendation and incentivisation engine were further advanced. The grid-fleet-demand optimisation models, which support the design of sustainable and efficient electric bus networks, were expanded into a bi-level MILP framework that incorporates not only the perspective of the electric bus operator, but also that of the energy management authority. Regarding the incentivisation engine, efforts focused on finalising data collection from multiple stakeholders and preparing the conceptual architecture of the model. This model enables forecasting the contribution of Renewable Energy Sources (RES) to the electricity mix, thereby supporting the reduction of carbon emissions through the prediction of sustainable charging practices.

Under Task 2.2 (Harmonise), the architecture of the AI-Data Warehouse was delivered together with the first generation of APIs and connectors. The warehouse is positioned as the horizontal data backbone of metaCCAZE, enabling the data flows required by the electrification, automation, connectivity, and digital-twin innovations developed elsewhere in WP2. Initial integration tests with partner systems have confirmed the viability of the chosen data models and interface specifications.

In Task 2.3 (Charge), the research phase on automated charging and V2G protocols has been completed, and the development phase has progressed to a proof-of-concept stage covering both stationary and dynamic charging for autonomous electric vehicles. The concept design, system design, and integration choices documented in this deliverable establish the technical envelope within which the Living Lab validation phase will operate.

Task 2.4 (Automate) consolidated the design of the Remote-Control Centre for autonomous vehicles, the Advanced Driver Assistance System, and the real-time planning model for autonomous modular bus fleets. The concept of the Remote-Control Centre was further developed to effectively coordinate fleet operations while addressing the legislative, technical, and service-related challenges associated with autonomous mobility systems. Significant progress was also achieved in the development of the Advanced Driver Assistance System for the design, simulation, and deployment of a full-stack autonomous navigation system for a modular pod vehicle, leading to its initial validation in a real-world controlled test environment. Additionally, a scheduling model for autonomous modular buses incorporating short-turning options to reduce operational costs

was developed and applied. The results demonstrated the potential to improve fleet efficiency by up to 50%.

Task 2.5 (Connect) has advanced on both fronts of V2X technology development and integration. The sensor-based data acquisition infrastructure has progressed from a prototype to a deployable and interoperable sensing unit, with improvements in hardware consolidation, power management, and embedded reliability. Secure cloud operations and a shared API for near real-time data exchange are underway, alongside a drone-based framework for traffic congestion detection. Work has also focused on defining C-ITS message profiles and preparing an intelligent road supervision infrastructure to support shared mobility services.

Task 2.6 (Manage & Control) has progressed across its three areas of focus. On electric vehicle scheduling, modelling tools have been developed to address optimal charging station placement, vehicle scheduling under uncertainty, and line planning for electric bus fleets. This work has been extended to tackle the tactical Vehicle Scheduling Problem for electric bus networks with multiple depots and time windows. On low-emission traffic management, a comprehensive traffic management model integrating V2X communication technologies has been developed and validated. Aerial drone-based data collection has been employed to support multimodal traffic analysis and congestion mitigation strategies. Additionally, an equitable Tradable Mobility Credit scheme has been advanced through a personalised user profiling framework applied to real mobility panel data. On supply-demand matching, the development of the three-component platform — covering user-facing, operator-facing, and backend interfaces — is completed, with all key modules of the user application already live, including planning, bookings, payments, and scheduling functionalities.

Finally, Task 2.7 (Integrate & Plan) has advanced the development of digital twin platforms consolidating datasets, algorithms, and frameworks from across WP2. Progress spans two living labs: in Amsterdam, the digital twin has been extended to incorporate urban waste logistics and population-driven waste production modelling, with all datasets and model outputs visualised within an interactive geospatial interface supporting scenario exploration and policy impact assessment. In Limassol, efforts have focused on integrating spatial, mobility, and energy-related datasets into a coherent Geoportal-Digital Twin environment, establishing the foundation for evidence-based planning and simulation activities.

Taken together, the developments reported across Chapters 2 to 8 confirm that the metaInnovations Toolkit has reached a level of maturity sufficient to support the implementation activities planned for WP3 and WP4. The technological progress described in this deliverable provides a solid and replicable foundation for the deployment of flexible, open, and robust ZESM solutions, and for the collaborative business and governance models pursued by metaCCAZE.

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. Samaranayake, 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 Martínez, Santiago and Dandl, Florian and Loder, Allister and Bogenberger, Klaus, Congestion-Aware Pick-Up and Drop-Off Network Design Problem (December 01, 2025). Available at SSRN: <http://dx.doi.org/10.2139/ssrn.5853954>
- 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. <https://doi.org/10.1177/0361198120917666>
- Biometrics, J. G.-, and 1971. 1971. "A general coefficient of similarity and some of its properties." *JSTOR* J C Gower *Biometrics*, 1971•JSTOR.
- Bliemer, M. C. J., Raadsen, M. P. H., Brederode, L. J. N., Bell, M. G. H., Wismans, L. J. J., and Smith, M. J. (2016). Genetics of traffic assignment models for strategic transport planning. *Transport Reviews*, 37(1):56–78
- 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>.
- Boyles, S. D., Lownes, N. E., and Unnikrishnan, A. (2025). *Transportation Network Analysis*, volume 1. edition 1.0.
- Bueno, A. P. 2021. "Identifying and Quantifying Mobility Hubs."
- Cao, J. and Menendez, M. (2015). Generalized effects of on-street parking maneuvers on the performance of nearby signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2483(1):30–38.
- 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.
- Cao, J., Menendez, M., and Nikias, V. (2016). The effects of on-street parking on the service rate of nearby intersections. *Journal of Advanced Transportation*, 50(4):406–420.
- 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." *ElsevierS Chen, R Seshadri, CL Azevedo, AP Akkinipally, R Liu, A Araldo, Y Jiang, ME Ben-AkivaTransportation 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.comYT Chen, A Liu, C Li, S Li, X YangScientific 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>.
- Dial, R. B. (2006). A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological*, 40(10):917–936
- Engelhardt R., and Dandl F. 2025. "Input data for munich case study." *Fleetsy*.
- 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., 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, 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."
- Engelhardt, R., F. Dandl, and K. Bogenberger. 2020. "Speed-up Heuristic for an On-Demand Ride-Pooling Algorithm."

- 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. Álvarez-Ossorio, 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*.
- Hanekamp, G., Tilg, G., Dandl, F., and Bogenberger, K. (2023). On the effects of ride-hailing and ride-pooling stop processes on urban network capacities. In 2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pages 1–6.
- 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." *Elsevier* Hoogendoorn-Lanser, NTW Schaap, MJ OldeKalter *Transportation 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." *Transportation Research Record*, 2017, 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., Keyvan-Ekbatani, M., and Ngoduy, D. (2020). Impacts of bus stop location and berth number on urban network traffic performance. *IET Intelligent Transport Systems*, 14(12):1546–1554.
- 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>
- 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.
- Kladeftiras, M. and Antoniou, C. (2013). Simulation-based assessment of double-parking impacts on traffic and environmental conditions. *Transportation Research Record: Journal of the Transportation Research Board*, 2390(1).
- Kolarova, V., Steck, F., and Bahamonde-Birke, F. J. (2019). Assessing the effect of autonomous driving on value of travel time savings: A comparison between current and future preferences. *Transportation Research Part A: Policy and Practice*, 129:155–169.

- Ku, D., M. Choi, D. Lee, and S. Lee. 2022. "The effect of a smart mobility hub based on concepts of metabolism and retrofitting." *ElsevierD Ku, M Choi, D Lee, S LeeJournal of Cleaner Production, 2022•Elsevier*, 379. Elsevier Ltd. <https://doi.org/10.1016/J.JCLEPRO.2022.134709>.
- Kucharski, R. and Cats, O. (2020). Exact matching of attractive shared rides (exmas) for system-wide strategic evaluations. *Transportation Research Part B: Methodological*, 139:285–310
- Lab, S. (2025). tap-b: C implementation of algorithm b. Accessed: 2025-05-02.
- 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.orgM Li, Z ZhuProceedings 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>.
- Liu, X., Qian, S., Teo, H.-H., and Ma, W. (2024). Estimating and mitigating the congestion effect of curbside pick-ups and drop-offs: A causal inference approach. *Transportation Science*, 58(2):355–376.
- 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.orgM Lv, Z Hong, L Chen, T Chen, T Zhu, S JiIEEE 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." *Transportation* 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 TechnologyStevenage, 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 KoningsveldSustainable 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 RanjitkarTransport 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 WitteJournal of Transport Geography, 2022•Elsevier*, 104. Elsevier Ltd. <https://doi.org/10.1016/J.JTRANGE.2022.103419>.
- Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.
- 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>.
- Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences of the United States of America*, 111(37):13290–13294
- Schmid, B., Jokubauskaite, S., Aschauer, F., Peer, S., H`ossinger, R., Gerike, R., Jara-Diaz, S. R., and Axhausen, K. W. (2019). A pooled rp/sp mode, route and destination choice model to investigate mode and user-type effects in the value of travel time savings. *Transportation Research Part A: Policy and Practice*, 124:262–294
- 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 DueckJournal 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."
- Stabler, B., Bar-Gera, H., and Sall, E. (2025). Transportation networks for research. <https://github.com/bstabler/TransportationNetworks>.

- 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.
- Stüger, P. N., Fehn, F., and Bogenberger, K. (2023). Minimizing the effects of urban mobility-on-demand pick-up and drop-off stops: A microscopic simulation approach. *Transportation Research Record: Journal of the Transportation Research Board*.
- 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>.
- Vishnoi, S. C. and Simoni, M. D. (2025). Surrogate-based real-time curbside management for ride-hailing and delivery operations. *Transportmetrica B: Transport Dynamics*, 13(1):2496823
- 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>.
- Wouda, N. A. and Lan, L. (2023). ALNS: a Python implementation of the adaptive large neighbourhood search metaheuristic. *Journal of Open Source Software*, 8(81):5028.

- 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>.
- Yang, H. and H. Bell, M. G. (1998). Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3):257–278
- 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>.

Annex

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.

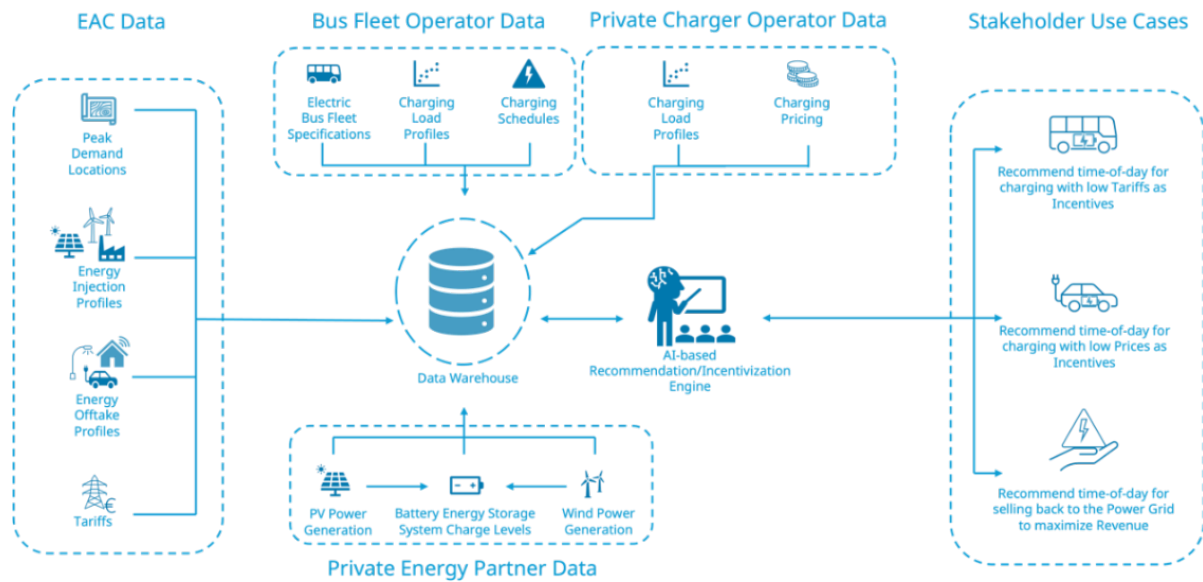


Figure 48: The conceptual framework architecture

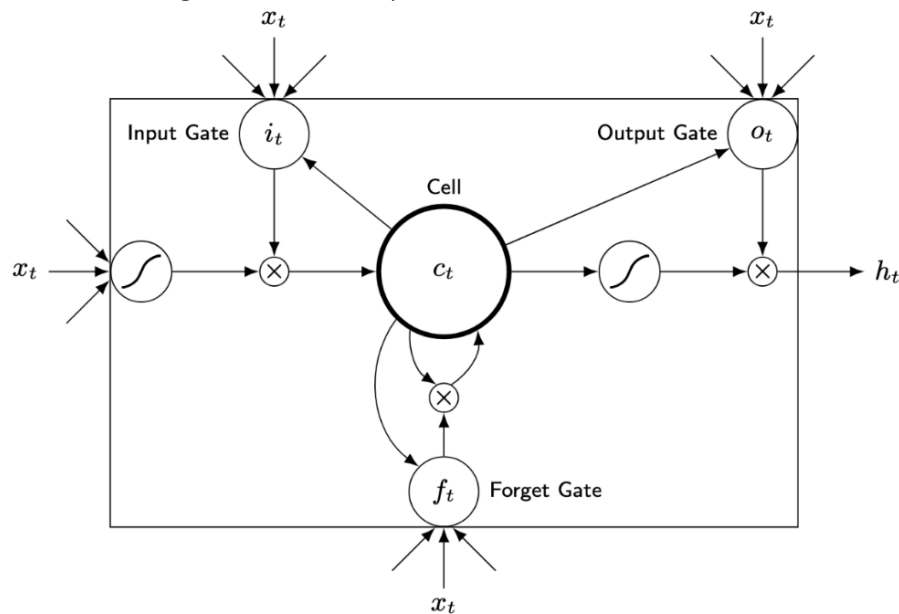


Figure 49: Example of LSTM Unit

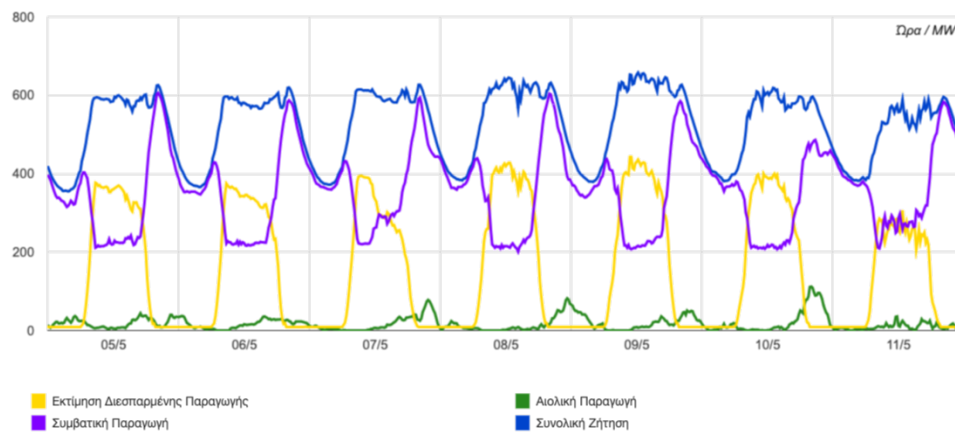


Figure 50: Wind, Distributed, Conventional Generation, and Total Power Demand (Sourced from Cyprus TSOC)

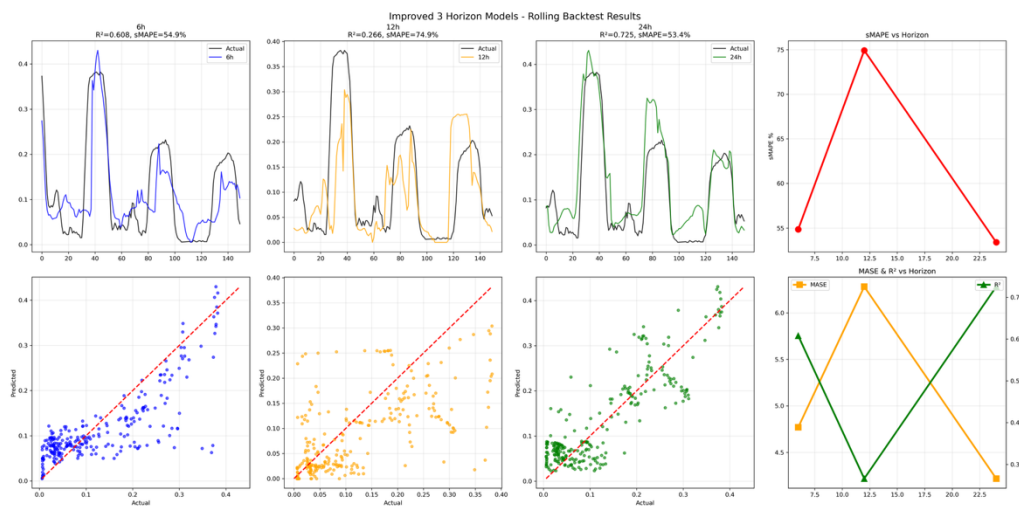


Figure 51 Forecasting model results for 6h, 12h and 24h prediction horizons

Annex sub-chapter for Chapter 3 and Task 2.2

Subtask 2.2.2



Figure 52: IoT Module using an ESP32 microprocessor

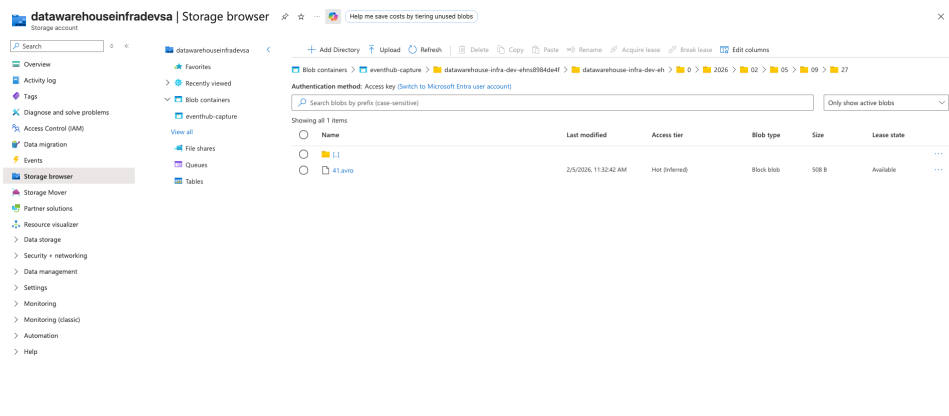


Figure 53: Screenshot of received IoT data in Data Lake

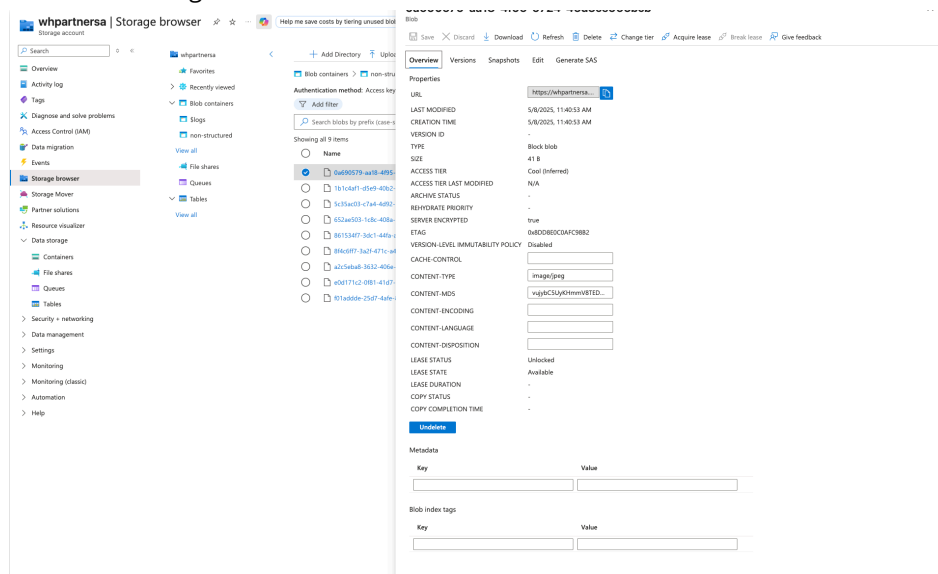


Figure 54: Screenshot of received static data in Data Lake

Annex sub-chapter for Chapter 4 and Task 2.3



Figure 55: Phase 0 table/lab tests on-going with the University of Applied Science Lapland. This robot hand will be installed to Remoted depot in phase 1

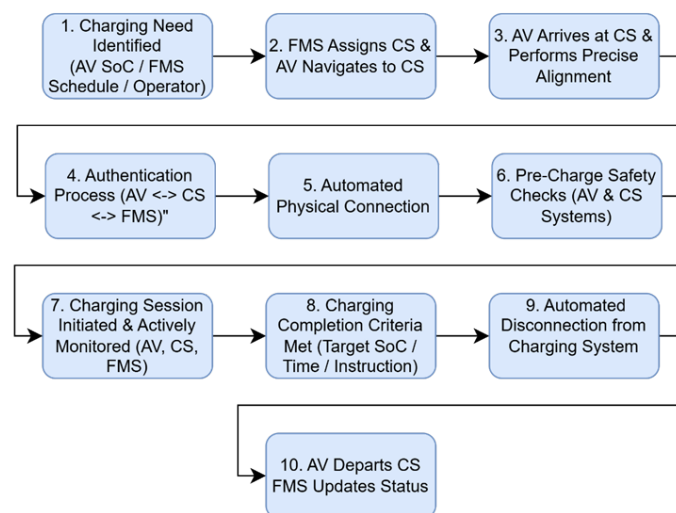


Figure 56: Charging process steps

Annex sub-chapter for Chapter 5 and Task 2.4

Subtask 2.4.1

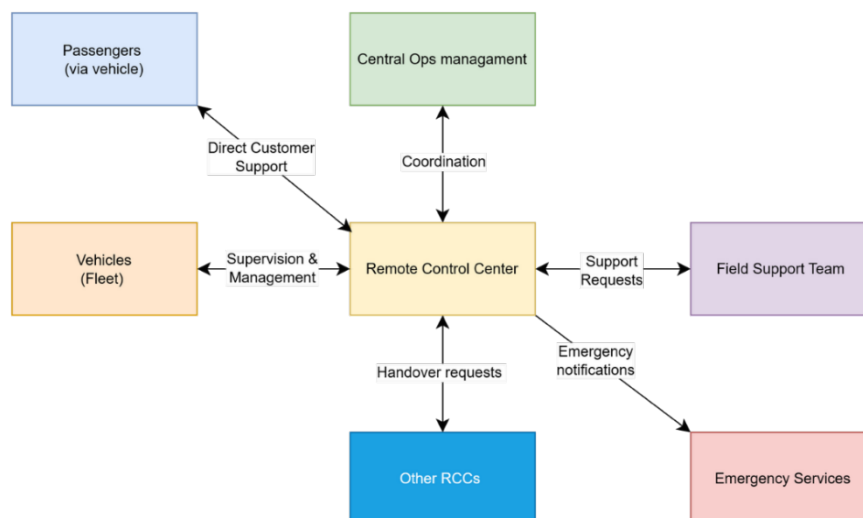


Figure 57: RCC Functional Block Diagram - Key Interactions

Subtask 2.4.2

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. Multiple simulation tests have been conducted 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 58.

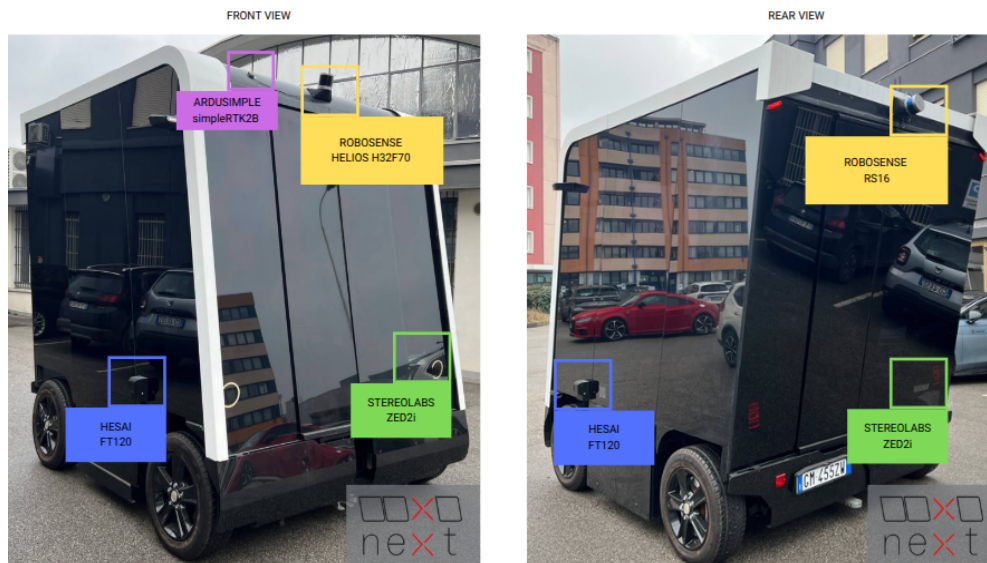


Figure 58: Sensors Placement on our prototype

The Next vehicle uses different types of Lidar for every side in order to have complete coverage of the area surrounding the vehicle with minimal blind area (see details in Table 7 in the Annex of this deliverable). Besides that, we use two wide-angle stereo cameras (ZED2i) with integrated IMU, capable of producing high resolution point clouds up to 100Hz, and with proprietary VIO algorithms that can produce high precision Odometry. 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 61 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 59. 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.

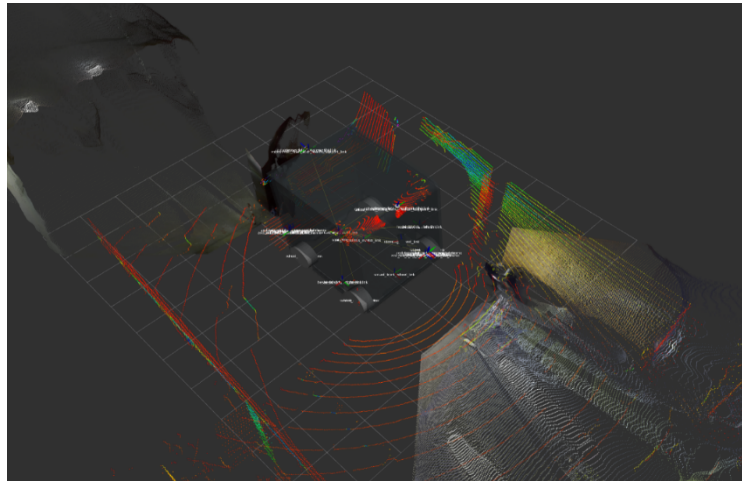


Figure 59: Sensors Coverage bird view from the real robot

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.

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 60: Next Modular Vehicles' Third Generation Pod

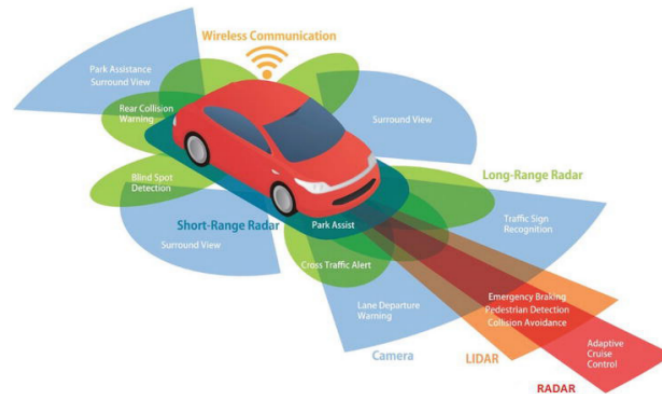


Figure 61: Standard Sensor Coverage for AMVs. Source: Wireless Communication and Sensors in Self-Driving Cars - Pizarov, Jelena & Mester, Gyula. (2020). The Future of Autonomous Vehicles. FME Transactions.

Minimum Safety Distance (MSD) calculation

Figure 62 shows an illustration of the MSD in the speed-time diagram. 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 (90), whereas Equation (91) express the term t with:

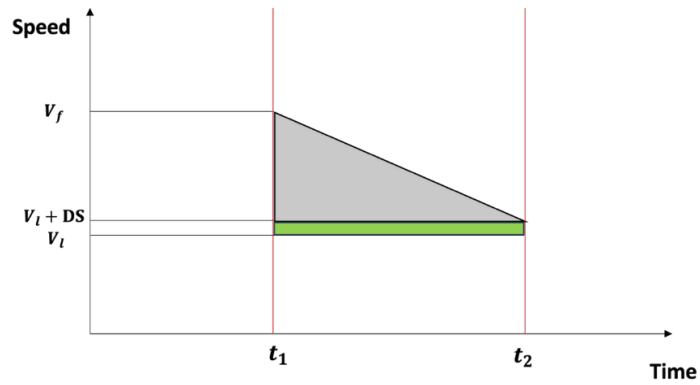


Figure 62: Graphical representation of the Minimum Safety Distance (MSD)

$$MSD = -DV * t - BC - \frac{1}{2} * d_{max} * t^2 \quad 90$$

$$t = \frac{DS - DV}{d_{max}} \quad 91$$

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.

Subtask 2.4.3

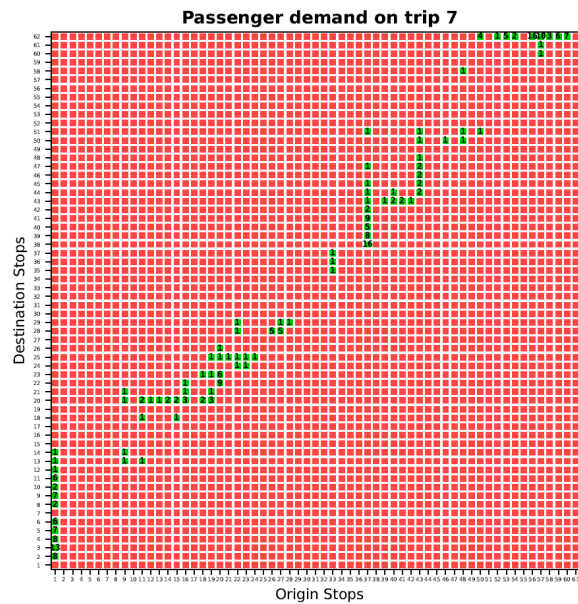


Figure 63: O-D Pair of Case Study

Annex sub-chapter for Chapter 6 and Task 2.5

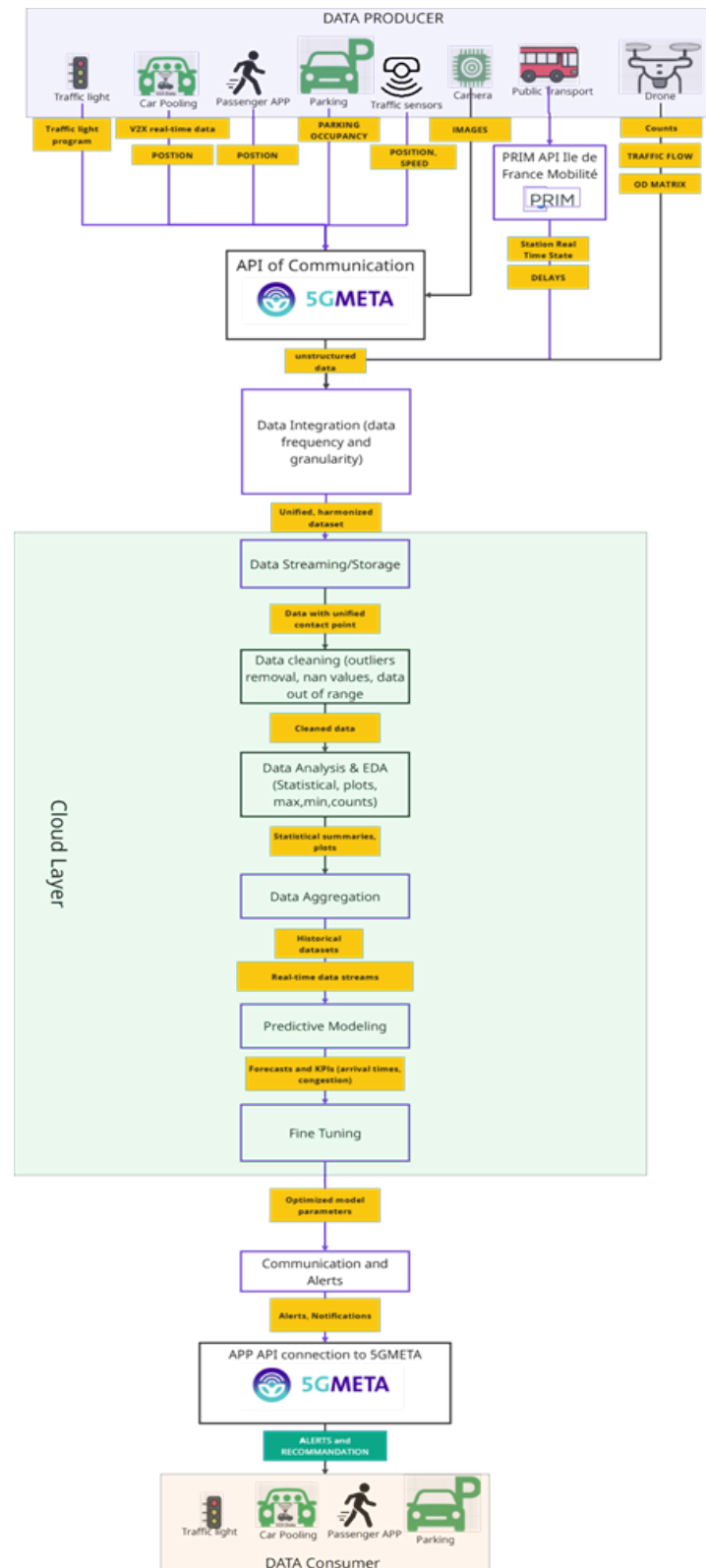


Figure 64: Flowchart of Traffic Management Framework - From Sensors to Services: A Scalable Pipeline for Predictive Traffic Management

Annex sub-chapter for Chapter 7 and Task 2.6

Subtask 2.6.1

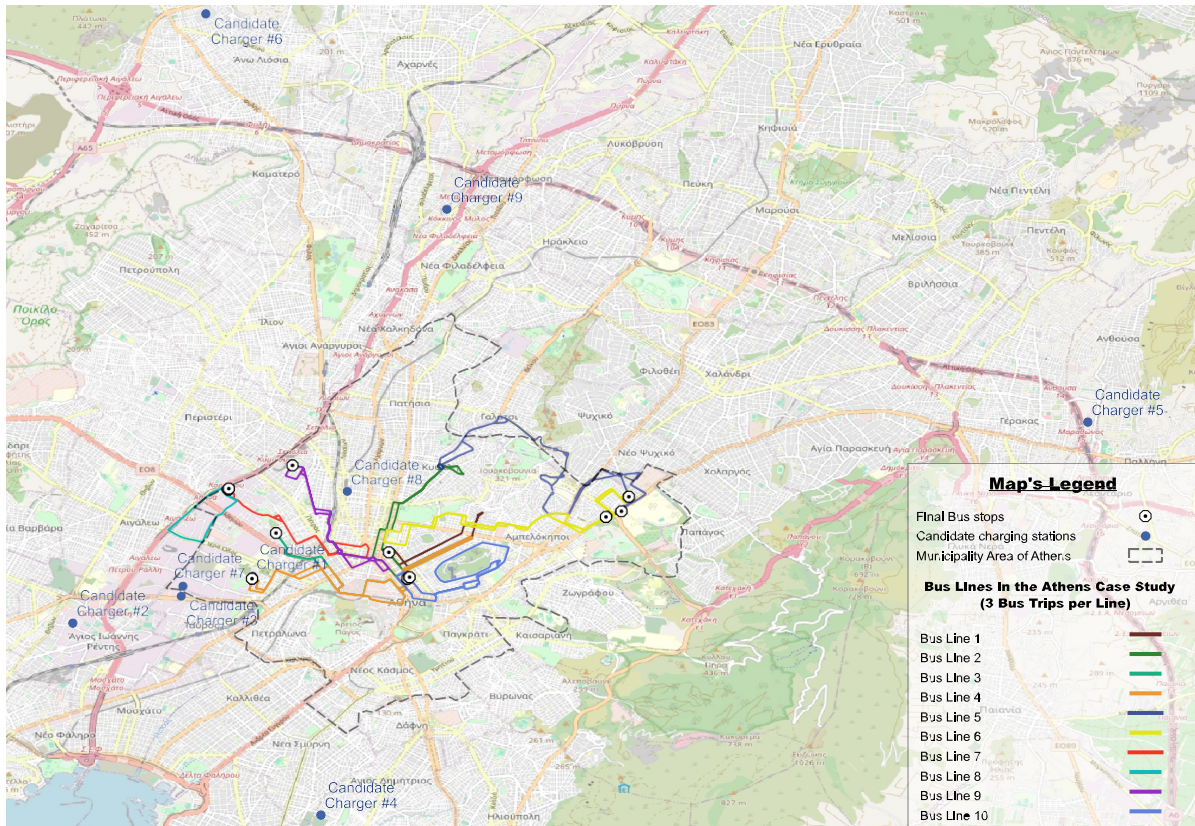


Figure 65: Athens network of the electric bus lines, along with the proposed charging station physical locations

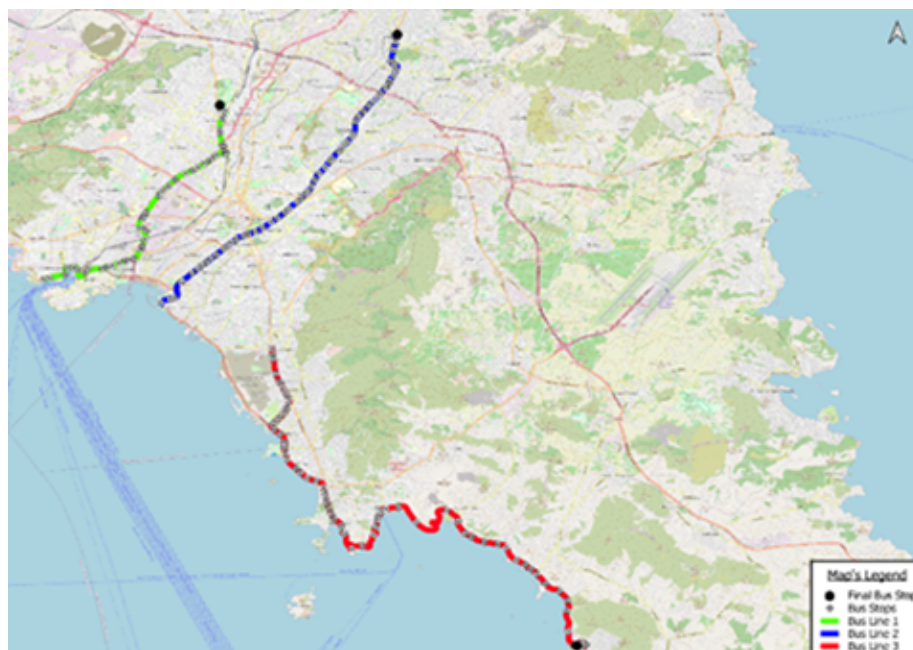


Figure 66: The three bus lines examined under this case study

Combined Line Planning Modification and Vehicle Scheduling for a fleet of Electric Buses

MODELED NETWORK REPRESENTATION

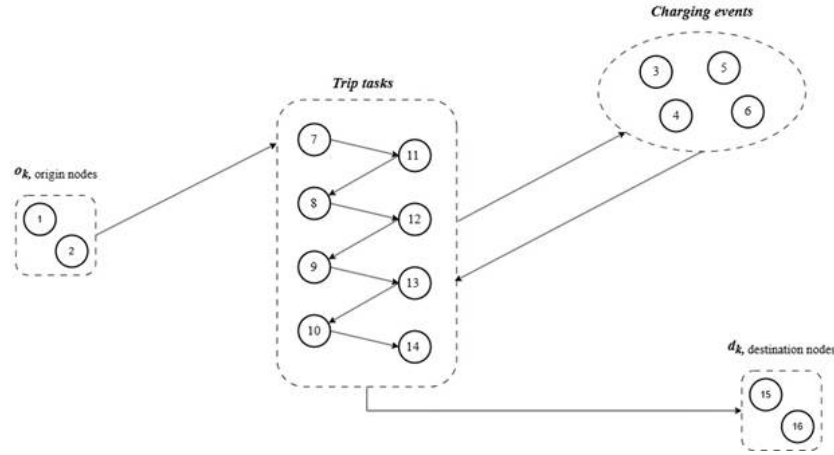


Figure 67: A generic representation of the modelled problem

Mathematical formulation

The mathematical model for the electric vehicle scheduling model (E-VSP) is as follows:

$$\min \left(\sum_{i \in \mathcal{O}} \sum_{j \in (\mathcal{F} \cup \mathcal{V} \cup \mathcal{D})} \sum_{k \in \mathcal{K}} x_{ijk} + p_1 \sum_{i \in (\mathcal{F} \cup \mathcal{V})} \sum_{j \in \mathcal{D}} \sum_{k \in \mathcal{K}} T_{jk} x_{ijk} + p_2 \sum_{i \in (\mathcal{F} \cup \mathcal{V})} \sum_{j \in \mathcal{D}} \sum_{k \in \mathcal{K}} t_{ij} x_{ijk} - p_3 \sum_{i \in \mathcal{D}} \sum_{k \in \mathcal{K}} \text{SOC}_{ik}^s \right) \quad (92)$$

s.t.:

$$\sum_{j \in \mathcal{N}} x_{ojk} \leq 1 \quad \forall k \in \mathcal{K}_1 \quad (93)$$

$$\sum_{j \in \mathcal{N}} x_{ojk} = 0 \quad \forall k \in \mathcal{K}_1 \quad (94)$$

$$\sum_{j \in \mathcal{N}} x_{ojk} \leq 1 \quad \forall k \in \mathcal{K}_2 \quad (95)$$

$$\sum_{j \in \mathcal{N}} x_{ojk} = 0 \quad \forall k \in \mathcal{K}_2 \quad (96)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{O}} x_{ijk} = 0 \quad \forall k \in \mathcal{K} \quad (97)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}, \forall j \in \mathcal{O}, \forall k \in \mathcal{K} \quad (98)$$

$$\sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \leq 1 \quad \forall j \in \mathcal{D}, \forall k \in \mathcal{K} \quad (99)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{D}, \forall j \in \mathcal{N}, \forall k \in \mathcal{K} \quad (100)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}} x_{ijk} = \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} \sum_{j \in \mathcal{D}} x_{ijk} \quad \forall k \in \mathcal{K} \quad (101)$$

$$x_{ijk} = 0 \quad \forall i \in \mathcal{F}, \forall j \in \mathcal{F}, \forall k \in \mathcal{K} \quad (102)$$

$$\sum_{i \in \mathcal{N}} x_{ijk} = \sum_{i \in \mathcal{N}} x_{jik} \quad \forall j \in \mathcal{F} \cup \mathcal{V}, \forall k \in \mathcal{K} \quad (103)$$

$$\sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} x_{ijk} \leq 1 \quad \forall j \in \mathcal{F} \cup \mathcal{V} \quad (104)$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} = 1 \quad \forall j \in \mathcal{V} \quad (105)$$

$$\sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D}} \sum_{k \in \mathcal{K}} x_{ijk} \leq B \quad (106)$$

$$x_{ijk} = 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{N}, \forall j \in \mathcal{N} : i = j \quad (107)$$

$$T_{jk} \leq M(1 - x_{ijk}) + dt_j \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (108)$$

$$T_{jk} \geq -M(1 - x_{ijk}) + dt_j \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (109)$$

$$T_{jk} \leq M \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (110)$$

$$T_{ik} + t'_i + t_{ij} - T_{jk} + \sigma_{ijk} \leq 0 \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (111)$$

$$T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} - T_{jk} + \sigma_{ijk} \leq 0 \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{V} \cup \mathcal{D} \quad (112)$$

$$\sigma_{ijk} \leq M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (113)$$

$$\sigma_{ijk} \geq -M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (114)$$

$$T_{ik} \leq Time_k + M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{O}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (115)$$

$$T_{ik} \geq Time_k - M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{O}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (116)$$

$$T_{jk} \leq T_{ik} + t'_i + t_{ij} + M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{V}, \forall j \in \mathcal{D} \quad (117)$$

$$T_{jk} \geq T_{ik} + t'_i + t_{ij} - M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{V}, \forall j \in \mathcal{D} \quad (118)$$

$$T_{jk} \leq T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} + M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{D} \quad (119)$$

$$T_{jk} \geq T_{ik} + (SOC_{ik}^e - SOC_{ik}^s)/c + t_{ij} - M(1 - x_{ijk}) \forall k \in \mathcal{K}, \forall i \in \mathcal{F}, \forall j \in \mathcal{D} \quad (120)$$

$$SOC_{jk}^s \geq SOC_{min} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{D} \quad (121)$$

$$SOC_{jk}^s \leq M \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (122)$$

$$SOC_{ik}^e \leq E_k + M \left(1 - \sum_{j \in \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \quad (123)$$

$$SOC_{ik}^e \geq E_k - M \left(1 - \sum_{j \in \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \quad (124)$$

$$SOC_{ik}^e \leq M \sum_{j \in \mathcal{N}} x_{ijk} \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V} \quad (125)$$

$$SOC_{jk}^s \leq SOC_{ik}^e - \theta_{ij} + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (126)$$

$$SOC_{jk}^s \geq SOC_{ik}^e - \theta_{ij} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}, \forall j \in \mathcal{F} \cup \mathcal{V} \cup \mathcal{D} \quad (127)$$

$$g_{jk} \leq SOC_{jk}^s - SOC_{jk}^e + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (128)$$

$$g_{jk} \geq SOC_{jk}^s - SOC_{jk}^e - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (129)$$

$$g_{jk} \leq e_j + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{V} \quad (130)$$

$$g_{jk} \geq e_j - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{F} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{V} \quad (131)$$

$$g_{jk} \leq SOC_{jk}^s - SOC_{max} + M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \quad (132)$$

$$g_{jk} \geq SOC_{jk}^s - SOC_{max} - M \left(1 - \sum_{i \in \mathcal{O} \cup \mathcal{V}} x_{ijk} \right) \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \quad (133)$$

$$g_{jk} \leq M \sum_{i \in \mathcal{N}} x_{ijk} \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (134)$$

$$g_{jk} \geq -M \sum_{i \in \mathcal{N}} x_{ijk} \forall k \in \mathcal{K}, \forall j \in \mathcal{F} \cup \mathcal{V} \quad (135)$$

$$y_k M \geq \sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{N}} x_{ijk} \forall k \in \mathcal{K} \quad (136)$$

$$y_k \leq \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} x_{ijk} \forall k \in \mathcal{K} \quad (137)$$

Objective function (92) is used to minimize the number of vehicles required to cover the scheduled trips. Constraints (93) to (107), 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 (108) to (118) make certain that all the tasks a vehicle can perform comply with the timetable given by the public transport operator. Furthermore, constraints (119) to (137) 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.

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

Methodology

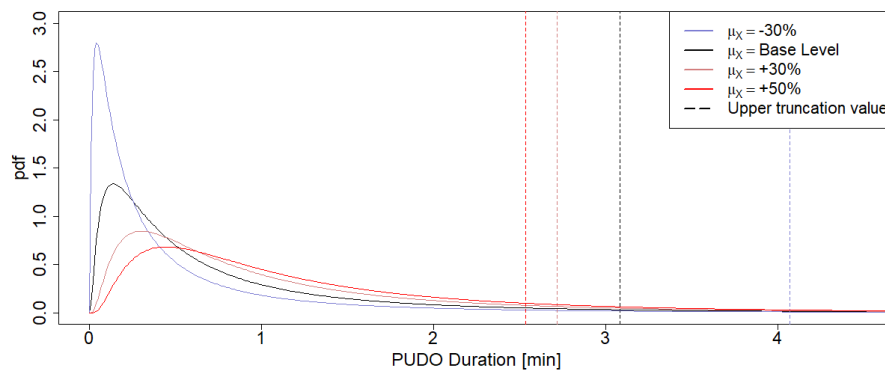


Figure 68: PDFs of Log-normal distributions with pivoting mean and constant variance

Case Study

Table 11: 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
Uninformed stochastic scenarios (108)			Stochastic	Four combinations of Log-Normal distribution parameters	Constant	{30 s, 45 s, 60 s}
Naive-informed stochastic scenarios (36)					Constant	Empirical mean duration of all requests
Model-informed stochastic scenarios (324)					Predicted	Max. relative error: {100%, 75%, 50%, 40%, 30%, 20%, 15%, 10%, 5%}
Fully-informed stochastic scenarios (36)					Predicted	Max. relative error: {0%}

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 69 reveal only a marginal operational improvement trend with better prediction models (e.g., with a lower maximum relative error δ).

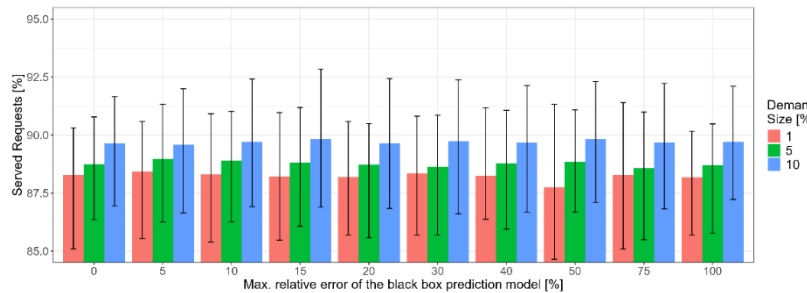


Figure 69: Operational impact of using different PUDO duration prediction models

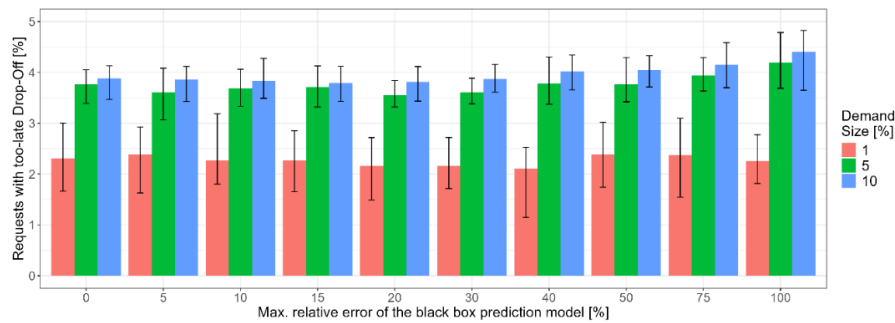


Figure 70: Impact on the Absolute Drop-Off time Deviation

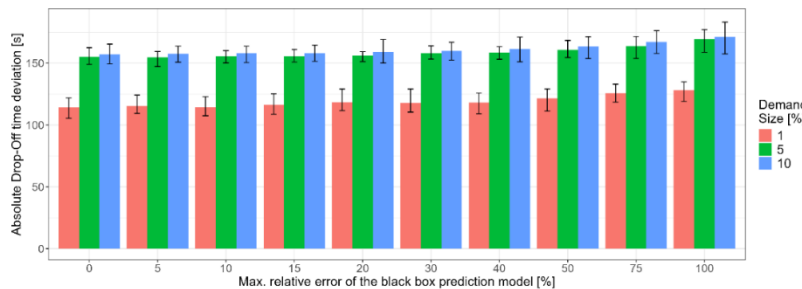


Figure 71: Impact on the Absolute Drop-Off time Deviation

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 70 and Figure 71 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 72, Figure 73, and Figure 74 also display the results for $\delta = 0\%$ and $\delta = 100\%$ from the *Fully-informed stochastic* families.

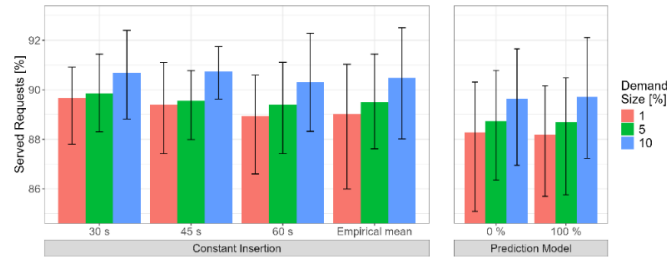


Figure 72: Operational impacts using different Constant Insertion and Prediction Models

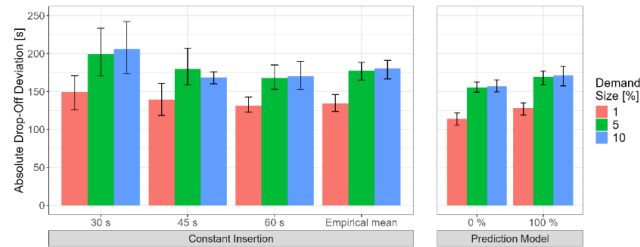


Figure 73: Impact on the Absolute Drop-Off time Deviation

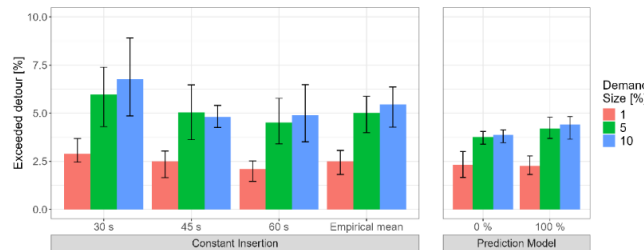


Figure 74: Impact on the Late Drop-Off share

Regarding the share of served request, in Figure 72, 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 72 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 73 and Figure 74 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.

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

Problem Description

The planning decision—or PUDO policy—of how many curbside spaces to provide and whether to allow double-parking PUDO in a street corresponds to the link-level decision variables of our proposed CA-PUDO-NDP. These influence the AMOD and private vehicle routing, the number and type of PUDO processes in the links, and, ultimately, the total traffic flow. The objectives of the associated optimization problem are to minimize (01) congestion on the streets (and thereby implicitly both the in-vehicle travel times of private vehicles and AMOD users) and (02) walking duration of AMOD users. Additionally, city planners will want to limit or even minimize (03) the curbside space dedicated to PUDOs.

Let $G = (N, L)$ with nodes N and links L be a graph describing the street network of a transport system. We define decision variables $x_l \in \{0,1\}$ and $n_l \in N_{\mathbb{0}}$, which describe, for each link $l \in L$, whether double parking is allowed ($x_l = 1$) and how many curb spaces n_l should be dedicated to PUDOs, respectively. We denote the combination of x_l and n_l values for all links in a network as a state. In its most general form, the CA-PUDO-NDP can be described as a multi-level multi-objective problem:

$$\begin{aligned} & \min_{x_l, n_l} 01, 02, 03 \\ & s. t. \quad 01, 02, 03 \leftarrow TM(x_l, n_l; \dots) \end{aligned}$$

Hereby, $\leftarrow$ symbolizes that the variables on the left are determined by the problem/model on the right side. $TM(x_l, n_l; \dots)$ refers to a transport model (in equilibrium), which determines the objectives for a given setting of x_l and n_l . This model must represent travel demand and supply, as well as their interaction, and capture the operation of an AMOD service, including vehicle routing. The selection of a suitable transport model depends on the requirements. For the CA-PUDO-NDP,

the transport model needs to reflect changes in the decision variables and is informative in the objectives. In other words, the travel times in the network need to be sensitive to the number and type (double parking or curbside; the former having a greater impact on congestion) of PUDO processes happening in the links, and the walking durations to the available PUDO locations need to be estimated. Importantly, it should be noted that the solution space grows exponentially with the number of links in the network, $|L|$. Arguably, the CA-PUDO-NDP is a planning problem and does not have real-time requirements, but even for relatively small $|L|$, the size of the solution space requires efficient methods for both optimization and the transport model $TM(x_l, n_l; \dots)$.

Given a series of assumptions—see Álvarez-Ossorio Martínez et al. (2025) for details—the transport model needs to incorporate two key components: a vehicle routing problem (VRP) for the AMOD provider, and a traffic assignment (TA) model based on user equilibrium that accounts for PUDO (pick-up/drop-off) processes. The VRP used here is more complex than standard formulations and requires a custom, ad-hoc approach, which is not detailed in this deliverable. As a result, the CA-PUDO-NDP can be presented in the following form, where R is the set of requests that the AMOD provider has to serve and D the set of trips conducted by private vehicle:

$$\begin{aligned}
 \min_{x_l, n_l} O_1 &= \sum_{p \in R \cup D} t_p^{drive}(t_l), O_2 = \sum_{r \in R} t_r^{walk}, O_3 = \sum_{l \in L} n_l \\
 s. t. (n_{od}^{AMOD}, f_l^{curb}, f_l^{dbl}, t_r^{walk}) &\leftarrow \text{VRP}(x_l, n_l; t_l; R) \\
 t_l &\leftarrow \text{TA}(n_{od}^{AMOD}, n_{od}^{PV}, f_l^{curb}, f_l^{dbl}) \\
 0 \leq n_l &\leq n_l^{max} \forall l \in L \\
 n_l &\in \mathbb{N}_0 \forall l \in L \\
 x_l &\in \{0,1\} \forall l \in L
 \end{aligned}$$

As shown above, O_1 is computed by summing up all the in-vehicle travel times of all AMOD and private vehicle (PV) passengers. O_2 collects all the walking times of AMOD users. Finally, O_3 aims at minimizing the space that has to be dedicated to AMOD processes in the whole network, also denoted by $N_{city} = \sum_{l \in L} n_l$. A VRP with walking legs considering travel times t_l and PUDO constraints is solved to find the stop sequence and the PUDOs (including walking legs) of AMOD vehicles. From these sequences, the AMOD trips, n_{od}^{AMOD} , as well as the number of PUDOs per link can be determined. These can be split into frequencies for both curbside (f_l^{curb}) and double parking (f_l^{dbl}) PUDOs. Finally, a TA sensitive to these PUDO frequencies assigns both AMOD and private vehicle demands ($n_{od}^{AMOD}, n_{od}^{PV}$) to find the user equilibrium in order to determine the link travel times in the network, t_l , and thereby also the in-vehicle travel times t_p of all travelers $p \in R \cup D$. The final two equations describe the domains of the binary (x_l) and non-negative integer (n_l) decision variables, where n_l^{max} represents the physical limit of curbside spots that can be dedicated for PUDOs in link l .

Methodology: notes on the individual modules

This section introduces some fundamentals of the AMOD, TA, and upper-level optimization modules. We refer the reader to Álvarez-Ossorio Martínez et al. (2025) for a more comprehensive explanation.

The AMOD module solves a VRP problem with three additional challenges: i) fleet size is unknown, ii) PUDO processes are not necessarily at request locations, iii) PUDOs need to be coordinated to stay within PUDO capacities. Since the CA-PUDO-NDP is a planning problem, we can look at the static version of the problem. Given the overall scope of the upper-level optimization, we look for a computationally efficient heuristic approach. Hence, we divide the overall problem into several easier to solve sub-problems.

Our strategy is as follows. First, we simplify the vehicle routing problem into a trip finding problem for the requests and select a subset of these trips such that all requests are served—similar to Santi et al. (Santi et al. 2014) and Kucharski and Cats (Kucharski and Cats 2024). In this step, we can include the search for different PUDOs on the trip level. For this study, we assume vehicle capacity $Q = 1$, as this reflects the predominant ride-hailing mode in current MOD services, and severely reduces the solution space and speeds up computation. Next, we discretize time to check the PUDO capacity constraints and compute $PUDO_{excess}^{cap}$ for the selected trips. Finally, we develop a method that puts vehicle routes through the selected trips with the primary objective of minimal fleet size.

The TA module calculates the link travel times and flows for a given network configuration, car travel demand, AMOD travel demand, and PUDO frequencies (both curb and double parking). We use a static, deterministic traffic assignment to address the strategic planning aspect of the PUDO network design problem, given the long tradition of using a static and deterministic traffic assignment in network design problems (e.g., Bliemer et al. 2017; Yang and Bell 1998). We assume that the link travel times are determined by the popular Bureau of Public Roads (BPR) function, which depends on the free flow speeds (v_0) and capacities (C_l). We capture the influence of PUDO processes on congestion by making the parameters v_0 and C_l dependent on the PUDO frequencies as well as the design of the road section, i.e., a two-lane layout, where parking or maneuvering vehicles can be overtaken, or a one-lane layout, where the possibilities for this are limited. The algorithmic implementation uses “Algorithm B” (Dial 2006), implemented in Lab (2025), to accommodate the requirement of ultra-fast computation times of the user equilibrium in the overall optimization problem (Boyles et al. 2025).

The upper-level optimization model consists of an adaptation of the ALNS metaheuristic. Particularly, we reinterpret the concept of (destroying a) neighbourhood and implement diverse destroy and repair operators that leverage domain and CA-PUDO-NDP-specific knowledge. We built our framework around Python’s ALNS implementation by Wouda and Lan (2023).

Multimodal Waste Collection Logistics

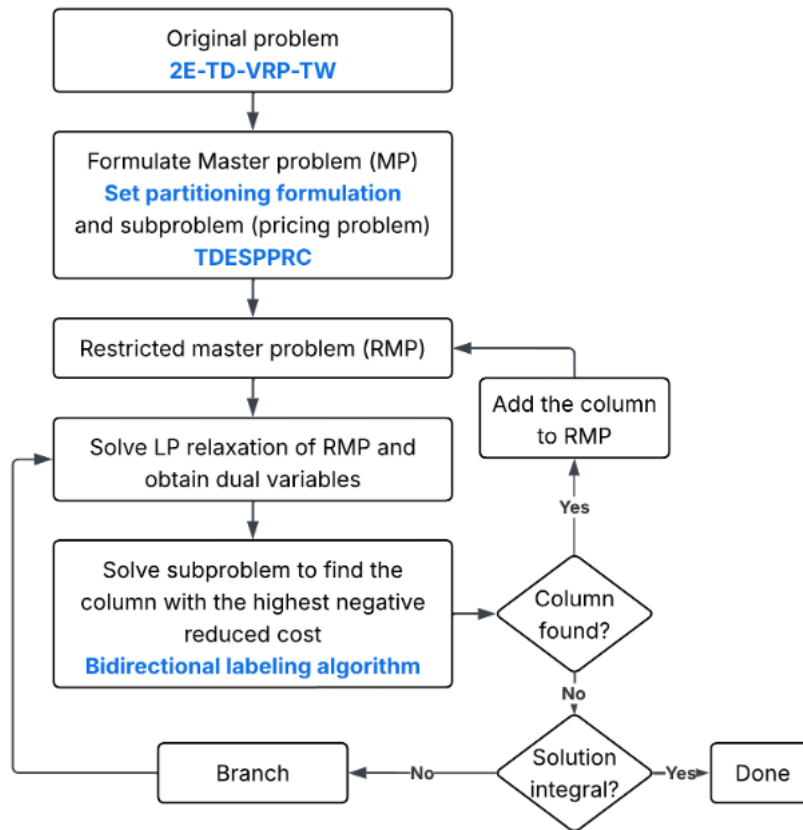


Figure 75: Branch-and-Price Schema

Subtask 2.6.2

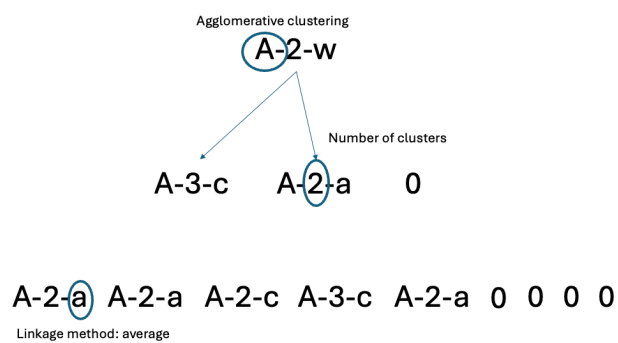


Figure 76: Tree Representation

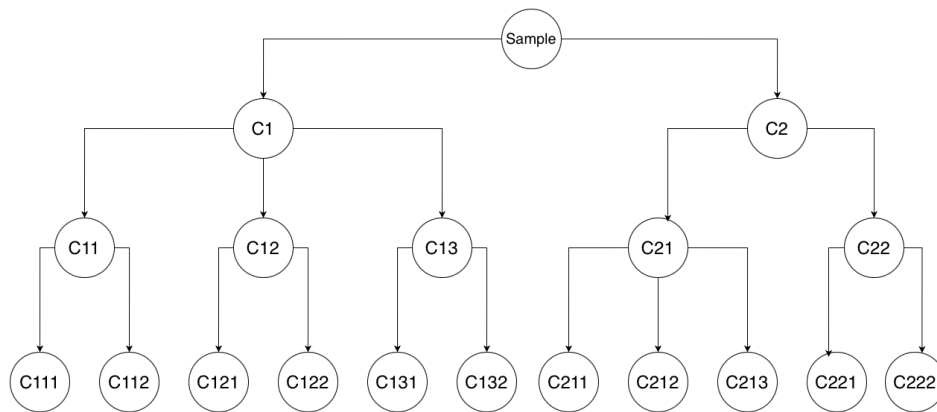


Figure 77: Cluster Separation

Annex sub-chapter for Chapter 8 and Task 2.7

No specific material is referenced here for Chapter 8 and Task 2.7.

www.metaCCAZE-project.eu



@metaCCAZE



metaCCAZE-project