MODULAR SIMULATION PLATFORM AND SIM2REAL TECHNIQUES

Project deliverable D5.2

HORIZON Research and Innovation Action | 101203465 |
Call HORIZON-CL5-2023-D6-01

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

DELIVERABLE ADMINISTRATIVE INFORMATION

TITLE OF THE DELIVERABLE	D5.2 - Modular Simulation Platform and Sim2Real Techniques
WORK PACKAGE	WP5
TYPE OF DELIVERABLE	R — Document, report
DISSEMINATION LEVEL	PU - Public
STATUS – VERSION, DATE	Final – V3, 25/08/2026
DELIVERABLE LEADER	ETH Zürich
CONTRACTUAL DATE OF DELIVERY	31/08/2026
SUBMISSION DATE	31/08/2026
KEYWORDS	orchestration framework, model-agnostic simulation, interoperability, Sim2Real, model-to-application transferability, foresight analysis, traffic signal control, multimodal traffic management

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

LIST OF CONTRIBUTORS

Name	Organisation
Julius Schlapbach, Kevin Riehl, Michail A. Makridis, Anastasios Kouvelas	ETH Zürich
Katerina Vakrinou, Emmanouil Kampitakis, Eleni Vlahogianni	NTUA
Charalambos Menelaou, Tania Panayiotou	UCY

QUALITY CONTROL

	Name	Organisation	Date
Peer review	Emmanouil Kampitakis, Eleni Vlahogianni	NTUA	17/08/2026
Peer review	Csaba L. Hargitai	BME	25/08/26
Quality review	Roozbeh Mohammadi	ERTICO	31/08/2026

VERSION HISTORY

Version	Date	Author	Summary of changes
V1.0	10/07/2026	ETH	First draft
V2.0	06/08/2026	ETH, NTUA, UCY	Consolidated draft circulated for internal peer review
V3.0	25/08/2026	ETH, NTUA, UCY	Revised after NTUA and BME peer review

LEGAL DISCLAIMER

Funded by the European Union's Horizon Europe Research and Innovation Actions under grant agreement No. 101203465. 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.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft / Confédération suisse / Confederazione Svizzera / Confederaziun svizra — Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER — State Secretariat for Education, Research and Innovation SERI

TABLE OF CONTENTS

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

LIST OF FIGURES

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

LIST OF TABLES

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

LIST OF ABBREVIATIONS AND ACRONYMS

Acronym	Meaning
API	Application Programming Interface
CPU	Central Processing Unit
EI	Expected Innovation
ER	Expected Result
ETH	Swiss Federal Institute of Technology Zurich
EU	European Union
FEDORA	Federation of network optimisation services, simulation foresights, and data alchemy for adaptable, agile, secure, and resilient multimodal traffic management.
FSM	Finite State Machine
GAE	Generalized Advantage Estimation
GAT	Grounded Action Transformation
MDP	Markov Decision Process
MFD	Macroscopic Fundamental Diagram
NTUA	National Technical University of Athens
PN	Protected Network
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
Sim2Real	Simulation-to-Reality
SUMO	Simulation of Urban Mobility
TCP	Transmission Control Protocol
TDD	Total Departure Delay
TraCI	Traffic Control Interface
TTT	Total Travel Time
UAV	Unmanned Aerial Vehicle
UCY	University of Cyprus

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

UPP	Urban Priority Pass
VHT	Vehicle Hours Travelled
VKT	Vehicle Kilometres Travelled
WP	Work Package

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

DELIVERABLE EXECUTIVE SUMMARY

These deliverable reports the work carried out under Tasks 5.4, 5.5 and 5.6 of Work Package 5, which develop the modular simulation platform and the model-to-application transferability methodologies of the FEDORA simulation space. It is led by ETH Zürich, while Chapter 3 has been contributed by NTUA as Task 5.5 leader and Chapter 4 by the University of Cyprus as Task 5.6 leader.

Chapter 2 presents the model-agnostic and modular simulation platform developed in Task 5.4. The platform is not a simulator of its own, but a coordinating layer between the modules that determine how the modelled system is acted upon and the environments in which those decisions take effect. The two sides never communicate directly, but they only exchange state information and decisions through a single orchestrator, which brings all connected components into a common lifecycle and keeps them synchronised as a run proceeds. The design has been instantiated end to end in a demonstrator for traffic signal control. SUMO acts as the environment, and four control methodologies of differing character, among them a baseline in which no controller is connected, are handled on the same terms. Every run is recorded and can be assessed through a common evaluation pipeline, and the implementation is available open source together with its documentation.

Chapter 3 reports the Sim2Real techniques for model-to-application transferability developed in Task 5.5, with particular emphasis on reinforcement learning based control strategies. As a proof of concept, a Proximal Policy Optimization perimeter controller combined with a queue balancing framework was trained in a source simulator, CityFlow, and evaluated in a target simulator, SUMO, in a sim-to-sim experiment on a grid test-bed network. The transfer relies on Grounded Action Transformation. A forward dynamics model is fitted on transitions collected in the target simulator and an inverse dynamics model on transitions collected in the source, and the resulting grounded action is applied in the source simulator during training. Direct transfer degraded the network-level indicators with respect to a policy trained directly in the target simulator, whereas the evaluated configurations of the framework outperformed direct transfer in almost all cases.

Chapter 4 reports the foresight analysis scenarios generation and evaluation capability developed in Task 5.6. Its in-silico platform starts from a parameterised scenario description covering demand, network state, UAV service, traveller behaviour and control policy. A configuration studio constructs a reproducible run, SUMO executes the microscopic multimodal traffic while a TraCI-based module coordinates the UAV service layer, a multimodal manager evaluates eligible passenger journeys and reserves capacity, and a dashboard consolidates the outputs into KPI tables and scenario comparisons. For each eligible passenger journey, the best available non-UAV journey is compared with a UAV-assisted chain of access, waiting, scheduled flight and continuation. The UAV option is selected only when it is operationally feasible and reduces the passenger's total estimated travel time. The workflow is currently being validated on the Aglantzia network, before transfer to the calibrated Greater Nicosia network. Chapter 5 draws the three tasks together and relates them to the innovations and results the project set out to achieve.

The three tasks contribute components of the FEDORA simulation space, and the platform of Task 5.4 offers a possible way of integrating them, as it is intended as an interoperability framework for the technical components developed across WP4 and WP5. Each task has produced a working implementation, and the demonstrator shows that methods and environments can be brought together through common interfaces rather than through integration work carried out for each pairing. Consistent with the stage of the project, the content of this deliverable reflects work in progress, and the scope covered by each task is set out in the corresponding chapter.

The purpose of this deliverable is therefore to document the state of these components and of the interfaces between them, ahead of the final version of the FEDORA simulation space that is to be reported in D5.3.

Funded by the European Union

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

1 INTRODUCTION

This deliverable is submitted at Month 15 of the project and reports the work carried out under Tasks 5.4, 5.5, and 5.6 of Work Package 5 (WP5), which develop the modular simulation platform and simulation-to-reality transferability methodologies for the FEDORA simulation space. It is led by the Swiss Federal Institute of Technology Zurich (ETH), while Chapter 3 has been contributed by the National Technical University of Athens (NTUA) as Task 5.5 leader, and Chapter 4 by the University of Cyprus (UCY) as Task 5.6 task leader.

The core of the deliverable is a model-agnostic and modular orchestration framework (Task 5.4) that separates decision logic from the environment in which it is executed, so that the controller or optimisation modules developed in the technical WPs can be connected to different simulation environments or a real pilot site, through a shared communication contract. To facilitate the deployment of logic modules developed in simulation in real-world pilot environments, Task 5.5 introduces a control approach and the Sim2Real techniques needed to bridge the reality gap between simulation and real-world deployment, and Task 5.6 introduces the foresight scenario components and simulation environments developed for network-wide analyses.

The content of this deliverable reflects work in progress. The orchestration framework, the control and transferability techniques, and the foresight components will continue to evolve as the demonstration preparation in WP6 advances and as the FEDORA solutions from WP4 and WP5 are progressively integrated. Where the current implementation covers only part of the intended architecture, this is documented transparently, and the corresponding elements will be consolidated in the final version of the FEDORA simulation space reported in D5.3.

1.1 MAPPING FEDORA OUTPUTS

FEDORA GA COMPONENT	COMPONENT TITLE	RESPECTIVE DOCUMENT CHAPTER(S)	JUSTIFICATION
	DELIVERABLE		
D5.2	Modular Simulation Platform and Sim2Real Techniques	Chapters 2-5	Chapter 2 covers the modular simulation platform (T5.4); Chapter 3 the Sim2Real techniques (T5.5); Chapter 4 foresight components (T5.6); Chapter 5 consolidates the research and scientific innovation, including the cross-task integration.
	TASKS		
T5.4	Model-agnostic and modular simulation platform	Chapters 2.1-2.5	Implemented model-agnostic and modular orchestration framework, covering its architecture, communication contract, traffic signal control demonstrator, evaluation pipeline and user documentation.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

T5.5	Sim2Real techniques for model-to-application transferability	Chapters 3.1-3.5	Control approach and Sim2Real techniques for model-to-application transferability, with a structured account of their integration into the orchestration framework.
T5.6	Foresight analysis scenarios generation and evaluation	Chapters 4.1-4.5	Foresight scenario components and simulation environments, and their compatibility with the orchestration framework.

Table 1. Adherence to FEDORA GA Deliverable & Tasks Descriptions

1.2 DELIVERABLE OVERVIEW AND REPORT STRUCTURE

This deliverable is organised into three chapters that reflect the tasks it covers, a chapter consolidating the research and scientific innovation, and a concluding chapter.

Chapter 2 documents the outputs of Task 5.4, led by ETH. It presents the model-agnostic and modular simulation platform and orchestration framework: its objective and design principles, its architecture and component model, the communication contract through which components exchange information, the mechanisms that allow simulation environments and logic modules to be exchanged, and the accompanying evaluation pipeline and user documentation. A demonstrator of the framework is available as open-source software, linking the “Urban Priority Pass” (UPP), presented in more detail in deliverable D4.1, and a set of baseline traffic signal controllers to the open-source SUMO (Simulation of Urban MObility) simulation environment.

Chapter 3 documents the outputs of Task 5.5, led by NTUA. It introduces a control approach and the Sim2Real techniques for model-to-application transferability, describes the demonstration through which transferability is assessed, and shows, as a structured exercise, how the control components would be integrated into the orchestration framework so that the same controller can be connected to a custom simulator, a baseline microsimulation or a real pilot site.

Chapter 4 documents the outputs of Task 5.6, led by UCY. The FEDORA foresight scenario components and simulation environments developed for network-wide analysis are designed to support the configuration, execution and evaluation of multimodal mobility scenarios and to specify how these components connect to the orchestration framework. The framework defines the information that each component can absorb and process, the commands and decision outputs it can return, and the inputs and logic modules that the simulation environments can accept and host. At the current stage, the complete FEDORA solution pipeline is being tested and validated on a smaller network representing the Aglantzia area, where the University of Cyprus is located, before being transferred to the calibrated Greater Nicosia network that will support the final large-scale pilot evaluation.

Chapter 5 summarizes the research contributions and scientific innovation of the activities within this deliverable. Its first section draws the three tasks together, explaining how they will compose into a single system and how the work relates to subsequent deliverables and to the WP6 pilots; the following sections relate the work to the project's Expected Innovations (EI) and Expected Results (ER). Chapter 6 concludes the deliverable and outlines the next steps, and Chapter 7 lists the references.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

2 MODEL-AGNOSTIC AND MODULAR SIMULATION PLATFORM

2.1 DESIGN PRINCIPLES AND OBJECTIVES

The evaluation of traffic management strategies before their deployment in the real world relies almost entirely on simulation. It is in a simulated environment that a new signal-control policy, a pricing scheme or a demand-management measure is developed, tuned and compared against the status quo. At the same time, the credibility of that simulation-based evidence ultimately determines whether a measure is trusted enough to be tested on real streets. Yet the different modules that are used for this research are currently rarely designed to work together. Optimisation models, control logic and simulation engines are typically developed independently, each with their own assumptions, data formats and interfaces, typically fitting one specialized context. Connecting a novel controller to a given simulator therefore tends to require significant integration work, which typically must be redone whenever the controller is moved to a different simulation environment, or should be deployed in a real-world setting. The result is that promising methods remain confined to the setting in which they were built, are difficult to compare on a common basis, and are hard to carry forward from a research prototype towards a real deployment.

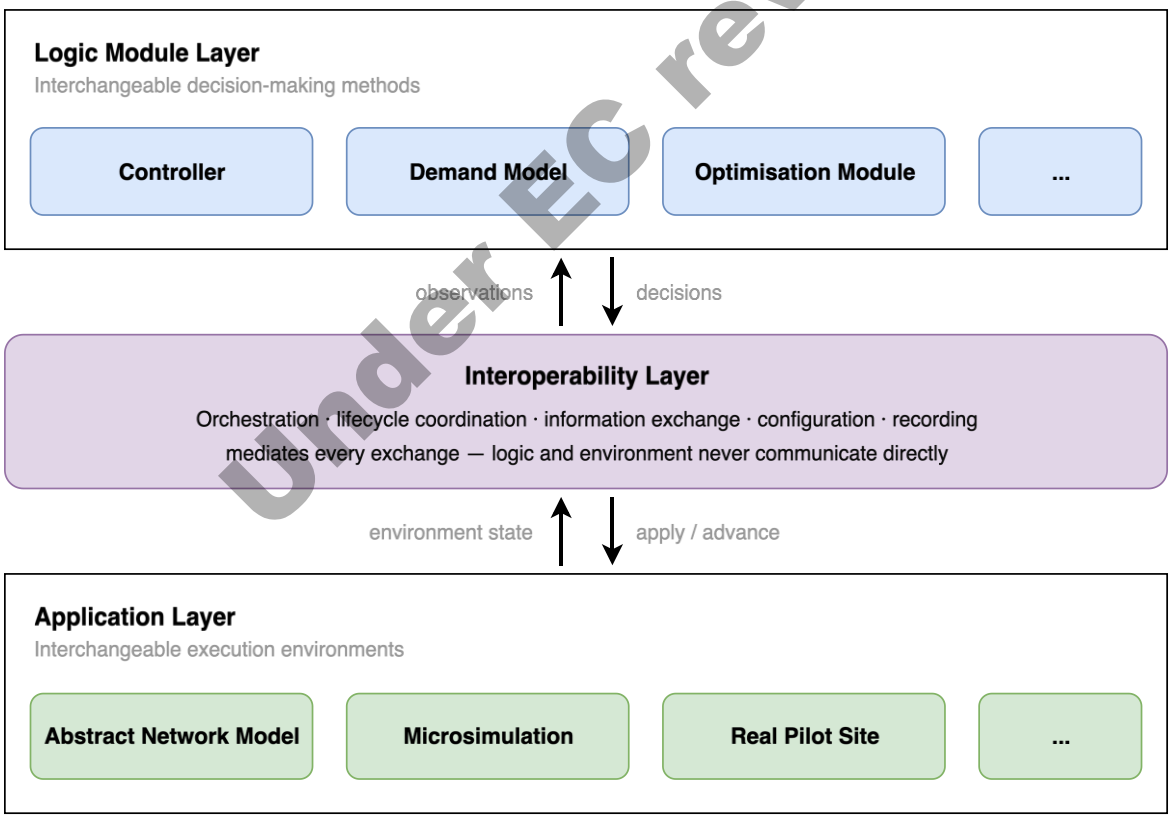

Figure 1. Generic three-layer design of the modular simulation platform. Interchangeable logic modules and execution environments communicate through a central interoperability layer, which coordinates their lifecycle and the exchange of state information and decision outputs. The list of logic modules and application layer environments is not exhaustive.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

This difficulty is not a new observation, and several established frameworks address parts of it. Multi-domain simulation frameworks such as Eclipse MOSAIC (Schrab et al. 2023), whose architecture is inspired by IEEE standard for High-Level Architecture (HLA) and couples simulators of the traffic, communication and application domains through federates and ambassadors, and earlier couplings such as iTETRIS (Rondinone et al. 2013) and Veins (Sommer et al. 2011) coordinate independently developed simulators, but their organising principle is the composition of several simulation domains into one holistic model, whereas the concern taken up here is the separation of the decision-making methods from the environments in which they are applied.

Task 5.4 addresses this challenging fragmentation by developing a model-agnostic and modular simulation platform: an interoperability framework whose purpose is to integrate the various traffic control solutions, demand management approaches, and simulation environments developed and used across the FEDORA technical work packages. Beyond the transferability allowing for simple application-independent execution and comparison of different modules of the same type, this should eventually also enable the combination of components without excessive integration work. Rather than being a single monolithic simulator, the platform is a coordinating layer that sits between two kinds of interchangeable parts, the logic module layer for decision making on one side and the application layer executing the communicated commands on the other, and mediates the exchange of information between them. This separation is visualized in Figure 1 and represents the central design decision of the platform. Both the logic module layer and the application layer hold interchangeable parts represented through a non-exhaustive list of possible modules, joined by an interoperability layer in the middle. The remainder of this chapter describes how the platform is realised, demonstrated and evaluated.

Three design principles follow directly from the Task 5.4 objective and shape the platform throughout.

The first is the **decoupling of decision logic from the environment in which it is applied and the model-agnosticism that follows from it**. A control or optimisation method should not need to know whether it is driving a microscopic traffic simulator, a more abstract network model or, in principle, an interface to a real pilot site. Similarly, an environment should not need to know which specific traffic management methodology is deployed for the computation of a traffic signal plan or which demand prediction model informs the decisions of the controller. As shown in Figure 1, the two sides never communicate directly, but the interoperability layer takes responsibility for the exchange of information between them, so that each side can be developed, replaced or re-used independently. Because logic and application layer communicate only through the platform's mediating layer, and never with one another, neither is tied to the internals of the other. A method developed against the platform can in principle be connected to any environment that speaks the same shared contract, and an environment integrated once becomes available to many methods. This is what makes it meaningful to speak of the platform as *model-agnostic*: it does not privilege a particular simulator, a particular class of logic modules, or a particular application domain.

The second is **composability**. The platform is deliberately structured as a set of cooperating parts with well-defined responsibilities, coordinated by a central orchestrator. This component acts as an *interoperability agent* between the logic and the application layer: it brings the parts into a common lifecycle, mediates the information they exchange, and keeps them synchronised as a run proceeds. At the same time this agent-based framework represents the introduction of a dedicated coordinating element rather than interoperability being built into each method or engine. Composability thereby also extends to the logic side, where more than one module can be combined so that their contributions are merged into a single course of action for the application layer.

The third is **testability and reproducibility**. Because the platform standardises how parts are coordinated and how information is exchanged, a given experiment can be described, configured and repeated in a controlled manner, and its outputs can be captured and analysed on a consistent basis regardless of which specific method with the same type of decision output or which environment was used. This principle is what allows the platform to serve not only as an integration mechanism but also as a basis for fair comparisons and evaluation, a role developed further in Section 2.4.

Taken together, these principles position the platform as the connective layer between the different technical logic modules and simulation environments developed and used throughout WP4 and WP5 within the FEDORA simulation space. It builds directly on

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

the integrated multimodal simulation and prediction framework reported in D5.1, which established the simulation engines and demand models that the platform is designed to orchestrate, and provides the mechanisms through which the logic modules developed elsewhere in the project can be brought into a common application setting. Finally, it also prepares connected methods for the transfer towards real-world operation that motivates Tasks 5.5 and 5.6. The sections that follow cover the platform's architecture and the way its parts interoperate (Section 2.2), the implemented demonstrator (Section 2.3), and its evaluation capabilities, implementation status and availability (Section 2.4).

2.2 ARCHITECTURE AND INTEROPERABILITY

The layered organization introduced in Section 2.1 is realised through a set of cooperating components that exchange information through a common scheme. This section describes that architecture: the way the components communicate (Section 2.2.1), and the lifecycle and stepwise execution through which the orchestrator coordinates them throughout a scenario run (Section 2.2.2). The descriptions focus on the reasoning behind the structure since these choices determine how easily new logic modules or alternative execution environments can later be brought together under the same framework, beyond the implemented demonstrator presented in Section 2.3.

2.2.1 COMPONENT MODEL AND COMMUNICATION

The platform comprises a small number of component types with distinct roles. A single orchestrator is tasked with addressing the responsibilities of the interoperability layer and maintains contact with all others. Any communication among decision-making modules or between them and the execution side is routed through it. On the execution side, one connected environment represents the setting in which decisions take effect and which determines the advancement of time. In the presented demonstrator this is a time-based simulation environment, but it could equally be an event-driven model or a connection to an operational site. On the decision-making side there are the logic modules, the methods that determine how the modelled system is acted upon, such as a traffic signal controller or an optimization routine. Also demand prediction models or other components that could inform decision-making processes can be connected through the logic module layer. A recorder observes the exchange between these components and stores it for later analysis without influencing it. This composition follows the three-layer arrangement of Figure 1, with the orchestrator and the recorder forming the mediating layer between the interchangeable decision-making and execution components.

Communication between the components takes the form of message passing rather than direct invocation. Each component runs as an independent process and exchanges states or decisions with the orchestrator over a Transmission Control Protocol (TCP) connection, so that no component holds a reference to another or calls its code. Running the components as separate processes that communicate over a network keeps the application and logic module layers decoupled at runtime and removes any requirement that they share a programming language, an internal data representation or a common address space. The same arrangement also allows the execution backend to be a simulator running alongside the orchestrator or, without any change to the coordinating logic, a connection to a remote system such as a real-world pilot installation.

The logic modules specify to the orchestrator during configuration and start-up what context information they require to return the desired decision and at each step the orchestrator requests the corresponding states from the environment and passes it on to them. Every message exchanged during execution is self-contained: it carries its origin, its intended recipient layer, the type of information it holds and the information itself, which is enough for the orchestrator to route it. For efficiency only the required state information is requested and then selectively communicated by the orchestrator in one direction and decisions from the logic modules can be collected and combined before being sent to the application layer, keeping the communication needs as low as possible. Because each message is characterised by the type of information it carries rather than by the specific component that produces it, the same mechanism conveys state descriptions, control decisions, or, for other classes of method, demand estimates

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

or optimisation results. How the quantities are reported in a state description or how control decisions are expressed, depends on the participating components and is described for the traffic signal control demonstrator in Section 2.3.

Table 2 summarises the message types currently defined, the information each of them carries, and the form this information takes in that demonstrator. Alongside the routing information described above, every message also carries the time at which it was sent, so that the recorded trace can be ordered independently of the components that produced it.

Table 2. Message types exchanged during a run, with the state and decision quantities of the traffic signal control demonstrator.

MESSAGE TYPE	DIRECTION	INFORMATION CARRIED	ROLE IN THE DEMONSTRATOR
step	Orchestrator to application layer	Releases the environment to carry out its next iteration.	No further content; the iteration reads the network state, applies the decisions returned for it and advances SUMO by one step.
traffic_state	Application layer to logic module layer, routed by the orchestrator	The state description for the current step. A fixed part is always transmitted; the lane measurements, namely queue lengths, weighted queue lengths and priority bids, only where a logic module requires them.	Time and step counter and, per signalised intersection, the number of phases used for the measurements, the index of the signal setting displayed and the signal state of the controlled links. The two state-responsive controllers add the per-phase queue lengths, the Urban Priority Pass also the priority bids, counted within a configurable distance of the stop line.
logic_command	Logic module layer to orchestrator	The decision of one logic module, labelled with a decision type.	Type traffic_light_command, with the time and step it refers to, the index of the next signal setting per signalised intersection and the controller's internal state for each.
apply_and_advance	Orchestrator to application layer	The decisions of all logic modules for one step, merged and passed on to be applied before the environment advances.	The combined assignment of a next signal setting per signalised intersection; empty in the baseline configuration, where the environment keeps its own signal plans.
get_state, state_report	Orchestrator to and from the application layer and the logic module layer	An optional request for a snapshot of a component's internal state and its reply, restricted to the attributes enabled in the configuration.	For the environment the step and time and the identifiers, lanes, positions and priority flags of the vehicles present; for the controllers the controller type, the internal state per intersection and, where computed, the submitted bids and whether a phase was switched.
vehicle_log_meta, vehicle_event	Application layer to orchestrator	The header of a run and the individual entry and exit events observed in the environment, reported rather than written by the environment itself.	Scenario label, signalised intersections, run length, spawn horizon, random seed and total lane length; per vehicle the identifier, event type, time, priority flag and, on leaving, the length of its route.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

environment_stated, environment_stopped	Application layer to orchestrator	Lifecycle signals marking when the environment has become operational and when its run has ended.	The first carries the SUMO label and prompts the first step; the second the time and step at which the run ended.
communication, vehicle_log	Orchestrator to recorder	Copies of the messages routed between the components and of the collected entry and exit events, mirrored without affecting the decision path.	Written to the communication log and to the vehicle log used by the evaluation pipeline.

2.2.2 COMPONENT LIFECYCLE AND ORCHESTRATION

Beyond exchanging information during a run, the components must be brought into and out of operation in a coordinated manner. They differ entirely in their internals and in how long they take to become operational. A microscopic simulator, for instance, must load a network and reach a running state, whereas a rule-based controller is ready almost immediately. The orchestrator therefore requires a uniform way to reason about the status of each component without being aware of their internal working principles. The proposed integration platform achieves this by modelling every component as a finite state machine (FSM) with the same lifecycle, shown in Figure 2. Each component passes through the same explicit states, namely created, configured, ready, running and stopped, together with a single failed state reachable from any of them. Because the lifecycle is explicit and identical across components, each may manage its own readiness and communicate it in a form the orchestrator interprets uniformly.

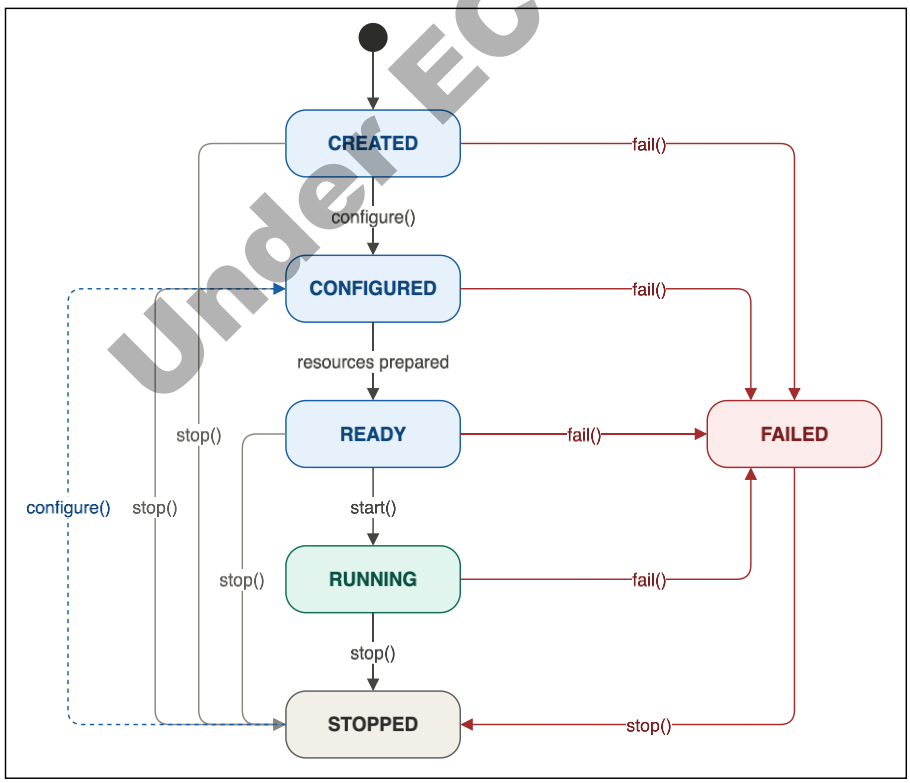

Figure 2. Shared component lifecycle modelled as a finite state machine. Every component managed by the orchestrator passes through the same states (created, configured, ready, running and stopped), with a failed state reachable on error from any stage and a path from stopped back to configured that returns a component to service.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

The separation of the early states is deliberate. Creation instantiates a component but does not launch it yet, configuration injects the scenario-dependent and user-defined parameters passed through a single configuration file, and the transition to ready certifies that the component's resources have actually been prepared, for example a network loaded, an internal model initialised, or a connection to an external system opened. Keeping these states distinct allows the orchestrator to create every component separately, configure them, and only continue towards the scenario run execution once each has independently reached the ready state. This readiness barrier based on internal component states excludes the ill-defined situation in which one component is already acting while another is still being initialised. The single failed state serves the complementary purpose. Since failures may arise at any stage and in any component, whether due a malformed configuration, a simulator that stops responding, or a lost connection to an external site, routing every such case to one explicit state gives the orchestrator a uniform means of detecting it and bringing the run to a controlled stop rather than leaving it in an inconsistent state. Finally, the path from stopped back to configured allows for a component to be reused across successive runs without being destroyed and rebuilt, which is especially relevant when a systematic variation over scenarios or module settings would otherwise repeatedly cause the computationally expensive restarting of a complex component such as a microscopic simulator.

Putting everything together, the orchestrator executes a full scenario according to the control loop shown in Figure 3, consisting of a startup phase, a repeated main loop, and a shutdown phase. In the startup phase the orchestrator reads the user-defined run configuration containing all necessary setup parameters and launches the recorder, the logic modules, and the environment through the lifecycle described above, completing with all components being in the ready state.

Once all components in the pipeline have signalled their readiness, each iteration of the main loop follows a fixed sequence, reflecting the semantic steps coordinated by the orchestrator between each advancement of the environment by one step. The orchestrator first requests all states required by the connected logic modules from the environment to enable decisions grounded in the situation at that step and distributes them to all logic modules at once. As every module receives the same information, their decisions are directly comparable and, where several modules act together, can be combined without ambiguity about the state they refer to. Each module computes its decision independently, preserving interchangeability and extensibility. The orchestrator accumulates the returned decisions and, once every module has replied, merges them into a single course of action and issues it to the environment, which then applies it and advances by one step. Throughout, the orchestrator mirrors the exchanged messages and, where enabled, additional environment state information to the recorder without affecting the decision path, so that a complete trace is produced for post-analysis without perturbing the scenario execution. After each iteration, the loop tests whether the scenario has been completed and either proceeds to the next steps or moves into the shutdown phase.

By collecting the logic modules' decisions and applying them bundled together, the orchestrator synchronises the main execution loop within every cycle, so that every module works from the same view of the state and their decisions take effect jointly, guaranteeing reproducibility given the same scenario configuration and connected logic modules. What follows then depends on the observed state and on the methods only, rather than on incidental differences in the order or speed with which the modules respond. This consistency is what makes it straightforward to connect and combine logic modules of very different character: a rule-based controller, an optimisation routine and a learning-based method are all coordinated through the same mechanism and reason based on the same information, so that any of them can be added, replaced or run alongside the others without having to account for how the others reach their decisions. The advancement of the simulation by one step is executed autonomously, so that under the assumption of operational equivalence the same coordination applies whether that environment is a simulation driving a model or a connection to an operational site that reflects the system as it evolves. As all exchanges of state information and logic module decisions are recorded, a run can be reconstructed and examined during post-processing, supporting the repeatable comparison of alternative configurations.

The orchestrator, as well as the overall framework, also support the case of no connected logic modules, in which the environment proceeds to execute the defined scenario under its own default behaviour and provides a natural reference evolution. When a run ends, the shutdown phase returns the components to the stopped state, releases their resources and finalises the recorded trace.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

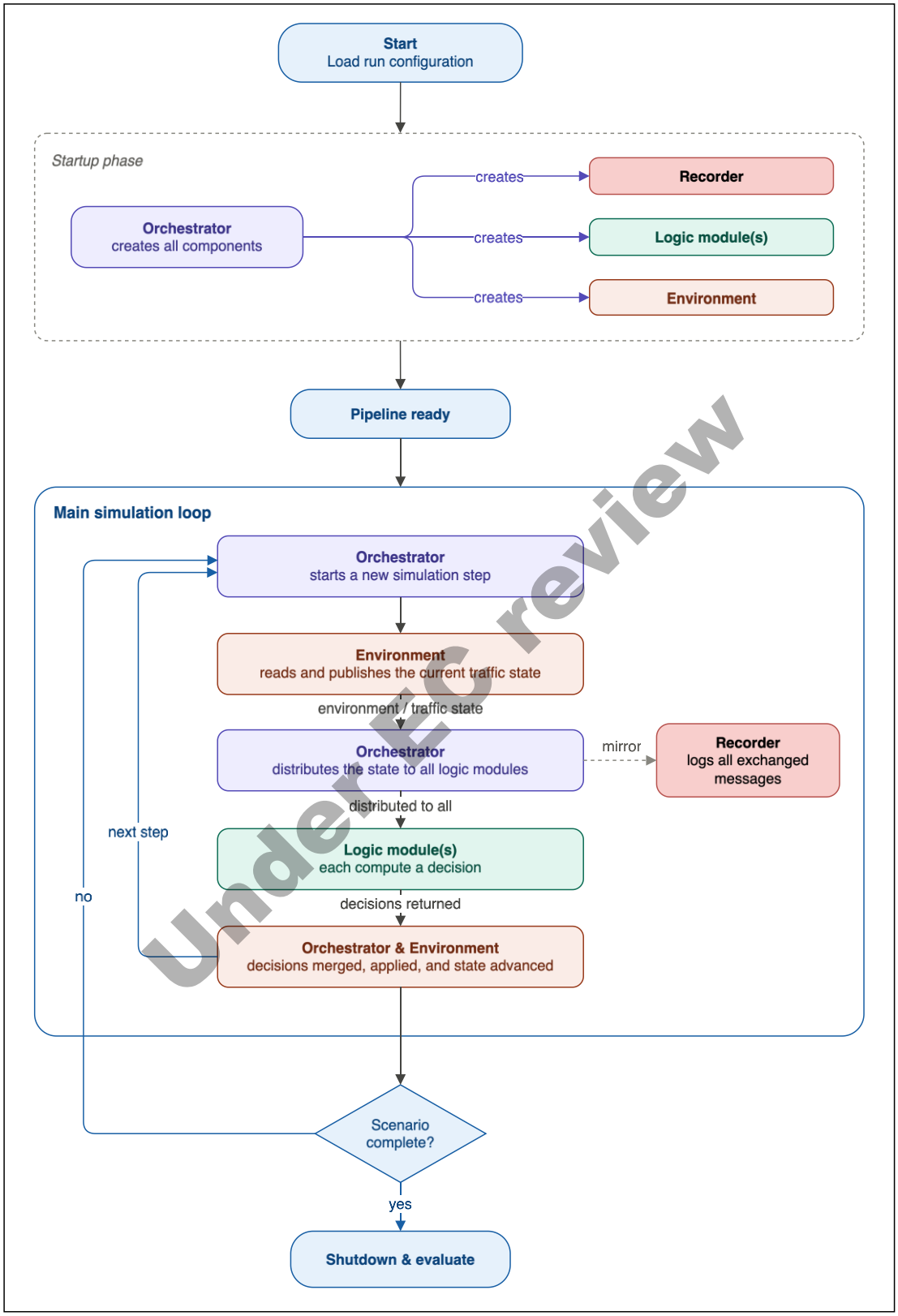

Figure 3. Control loop coordinated by the orchestrator. Each step consists of a publication of the traffic state, the collection of the logic modules' decisions, and the application of the merged decisions to the environment.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

2.3 DEMONSTRATOR: TRAFFIC SIGNAL CONTROL

The architecture described in Section 2.2 is general, since it describes how logic modules and application layer execution environments are coordinated without fixing the exact functionality of either of them. To demonstrate the functionalities of such a modular simulation platform end-to-end and provide a more concrete example for the described behaviours, it has been instantiated in a demonstrator for traffic signal control. Signal control is well suited for this purpose, as it is a well-studied problem with established methods of differing character, ranging from fixed schedules to state-responsive strategies, which lets the platform's handling of interchangeable logic modules be exercised on methods that are genuinely different from one another. Figure 4 illustrates the components of the demonstrator as an instance of the generic three-layer design in Figure 1, with a microscopic traffic simulator on the execution side, a set of signal controllers forming the logic module layer, and the orchestrator mediating between them with the recorder connected for tracking.

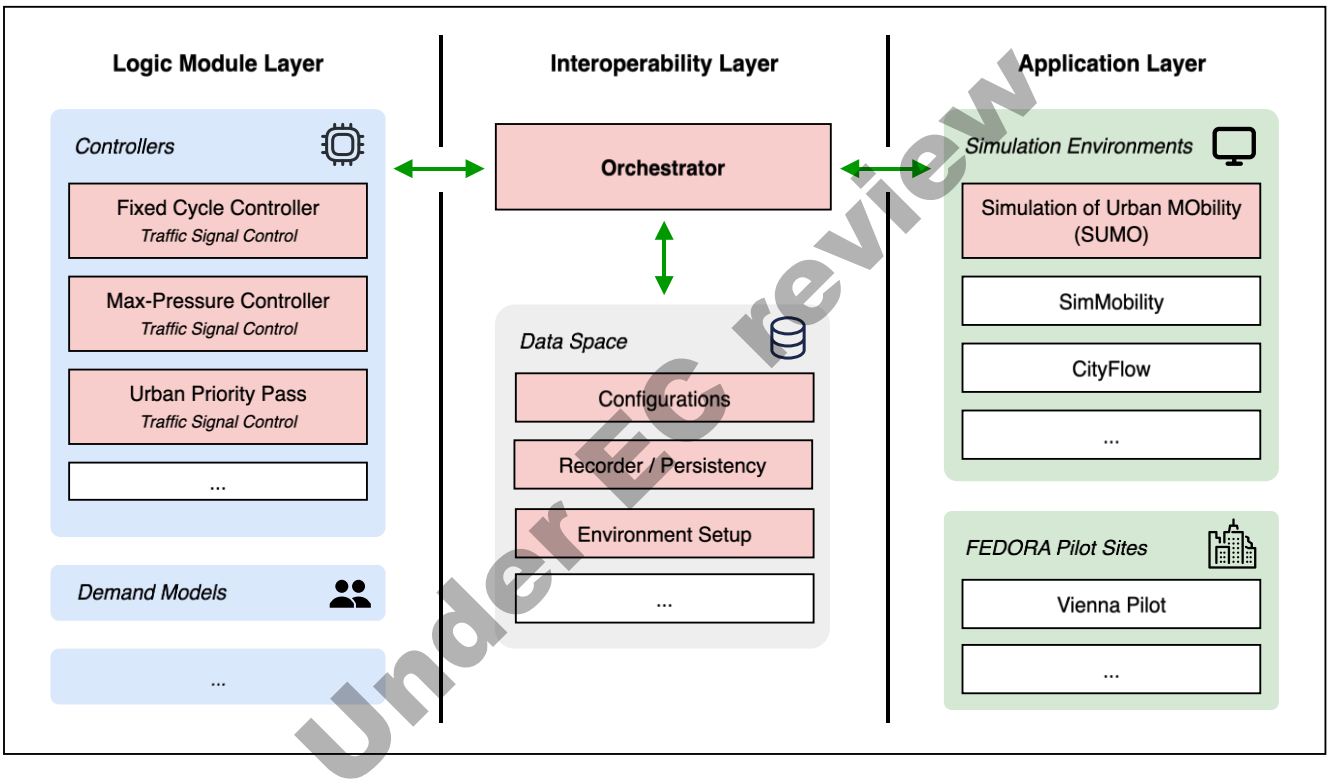

Figure 4. Traffic signal control demonstrator as an instance of the three-layer platform, with the signal controllers as logic modules and SUMO as the application layer environment (highlighted in red). Other simulation environments and pilot connections shown are illustrative and not part of the present implementation.

On the execution side, the environment is provided by SUMO (Simulation of Urban MObility), an established open-source microscopic traffic simulator. Consistent with the functionalities discussed in Section 2.2, SUMO reports selected state information of the modelled network on request, such as the queues forming on each approach and the current signal settings, and applies the given signal plan decisions before advancing the simulation by one step. It is driven through the Traffic Control Interface (TraCI), which the environment component uses to both read states and execute traffic signal changes. The scenario run with the demonstrator is defined to be self-contained, consisting of a road network, a corresponding traffic demand, and the definition of the signalised intersections it contains, which together provide a reproducible setting in which the controllers can be run and examined. Since the environment is only contacted through the orchestrator, the same role could equally be filled by another simulator or, in time, by a connection to a pilot site, as the illustrative entries in Figure 4 suggest.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

The logic modules in the demonstrator are traffic signal controllers, each computing signal decisions from the state information the environment reports, with the active controller being selected and configured through the user-defined scenario file. Four different control methodologies are available as part of the demonstrator. The baseline scenario corresponds to the case where no external controller is selected, which results in the environment running under its own built-in signal plans with default timing and the orchestrator issuing an empty list of decisions at each step. This corresponds to the case of no connected logic modules anticipated in Section 2.2, and serves as a reference configuration against which the active controllers can be assessed. The most basic connected method is a fixed-cycle controller that applies pre-timed signal plans, whose phase sequence and timing, including the green durations, the yellow transition and per-intersection offsets are specified entirely in the user-defined scenario configuration with each intersection cycling through its phases independently. While it is simple and predictable, this type of controller represents a logic module that acts without requesting or relying on state information from the environment.

The remaining two controllers are state-responsive. Max-pressure control derives its signal decisions from the reported state, using the pressure at different approaches to each intersection, understood as the difference in queue lengths between opposing approaches, to decide which movement should receive a green light. The selection is organised as an auction, in which the competing movements bid according to their pressure and the movement with the highest bid is granted the next green phase, so that the controller continually favours the directions under the greatest load. The Urban Priority Pass controller, developed within the FEDORA project and documented in deliverable D4.1, builds on this mechanism. It extends the auction with priority bids for designated vehicles, such as public transport, so that the signal timing can favour these vehicles alongside the general balancing of queues. The strength of this preference is set by a single tuneable parameter that trades off network-wide efficiency against the reliability of the priority service. When setting the strength parameter to its neutral value the priority bids have no effect and the controller reduces to a regular max-pressure approach. The quantitative behaviour of the Urban Priority Pass is reported in deliverable D4.1, while its role within this demonstrator is to show how advanced and new control methodologies developed as part of the FEDORA project can operate through the platform on the same terms as established solutions.

Combined, these controllers showcase the mechanisms introduced in Section 2.2. Although they range from a static schedule to an adaptive, priority-aware strategy, all four controllers present the same behaviour to the rest of the platform, requesting specific state information available in the application layer environment and returning a signal decision, so that switching between them requires no change to the environment, the orchestrator, or the recorder, and is expressed only in the user-defined run configuration. The shared auction basis of the two state-responsive controllers, and the reduction of the Urban Priority Pass to the max-pressure control approach at its neutral setting, further show that functionally similar methods can be examined against one another within a single environment and scenario, which is another practical value of the platform for the development of novel control approaches and optimisation methods elsewhere in FEDORA.

2.4 IMPLEMENTATION, EVALUATION AND AVAILABILITY

The demonstrator presented in Section 2.3 is provided as a working Python implementation of the platform, in which the orchestrator, the recorder, the SUMO environment and the different signal controllers run together as the independent, message-passing components introduced in Section 2.2. A scenario is fully defined by the mentioned user-defined configuration file, containing the scenario selection, the environment, and the active controller. Based on the configuration, the orchestrator launches all relevant components through their lifecycle and drives the main simulation and control loop until completion. As auxiliary files for the SUMO simulation environment, the scenario also supplies a self-contained network, demand information, and a set of signalized intersections, so that the implementation can be executed and reproduced without reliance on external data.

The scenario supplied with the demonstrator is a synthetic grid network with nine signalised intersections, each operating four controllable phases. Vehicles are generated at twelve entry points at 50 vehicles per hour each and assigned one of twelve routes, and a run covers 5 000 steps of one second under a fixed random seed. In the Urban Priority Pass configuration, 30 % of them are additionally designated as priority vehicles.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

During a run, two complementary records are produced. The recorder receives a copy of every message routed by the orchestrator and stores it, so that the information exchange between the components remains traceable after the completion of a run. These records hold the state information passed from the environment to the logic modules, the decisions they return, as well as the control messages that drive the main simulation loop. Where enabled, the orchestrator also periodically requests snapshots of the component's internal states through polling. The content of this communication log can be restricted to selected message types where a smaller log is sufficient. A second record captures the modelled traffic itself: as vehicles enter and leave the network, the environment reports these events to the orchestrator, which forwards them to the recorder. The latter collects them into a dedicated vehicle log, pairing each vehicle's entry and exit with the distance it travelled, which provides the basic for the evaluation pipeline described below and other post-processing.

Building on the vehicle logs, the repository provides an automated evaluation pipeline that is executed once a run has finished and computes standard traffic engineering metrics from it: the mean, median and spread of vehicle travel times, the vehicle hours and vehicle kilometres travelled (VHT and VKT), the traffic flow, the space-mean speed and the network density. Which of these are computed can be configured, and the pipeline returns a concise summary together with a set of standard plots. As the metrics are calculated based on the abstracted vehicle logs rather than simulation output files from any specific environment, the same evaluation applies to any execution backend that reports these events in the expected form, whether a simulator or a future pilot connection. As every run follows the same control loop and can be repeated under a fixed random seed, the metrics for different configurations are directly comparable, so that the baseline and the three controllers described in Section 2.3 can be assessed against one another on a common scenario. Moreover, the unmodified per-vehicle and message records also remain available to further post-processing, for which the repository includes controller-specific analysis scripts, such as a breakdown of the experienced average travel times for prioritised against regular vehicles under the Urban Priority Pass, as well as cross-controller comparisons. Consistent with the scope of this deliverable, quantitative results for developed components such as the Urban Priority Pass are reported in deliverable D4.1.

The entire platform is available open source. The code repository (https://github.com/fedora-horizon/fedora_platform) contains the presented demonstrator, including the SUMO integration, controllers, orchestrator, evaluation pipeline, and post-processing scripts. An accompanying user documentation (https://ivt-baug-ethz.github.io/fedora_platform) covers the platform's technical architecture, configuration, evaluation, as well as the components and their interfaces in more detail. This open availability lets the demonstrator be reproduced and the platform be reused and extended beyond the demonstrator reported here.

2.5 CONCLUSION

This chapter has presented the model-agnostic and modular simulation platform for coordinating interchangeable decision-making methods with the environments in which they are deployed. Section 2.1 set out its purpose and guiding principle to introduce a platform that acts as a coordinating layer and brings decision-making methods and execution environments together under a common scheme, while keeping the two sides independent and allowing several methods to act in combination. Section 2.2 developed these principles into an architecture centred around a single orchestrator through which all communication passes. It described how the components exchange information, how a shared lifecycle brings heterogeneous components into and out of operation in a coordinated manner, and how a stepwise control loop keeps all of them synchronised.

Sections 2.3 and 2.4 applied this design in a working system. The platform was instantiated in a demonstrator for traffic signal control, where a microscopic traffic simulator serves as the environment and a set of signal controllers, from fixed-time control to the Urban Priority Pass, act as the logic modules, each connected through the same interface. At this example, it was described how the recorder retains a complete account of each run and how an evaluation pipeline can process this information into standard traffic engineering measures, allowing different controllers to be assessed on a common basis. Released as an open-source code repository together with a user documentation, the platform offers a foundation on which the control and optimisation methods developed across the project can be integrated, compared, and in time carried towards the pilot environments.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

3 SIM2REAL TECHNIQUES FOR MODEL-TO-APPLICATION TRANSFERABILITY

3.1 TASK DESCRIPTION

One of the main challenges in deploying intelligent traffic management systems is the gap between simulation and real-world operation. Although advanced traffic control strategies often achieve excellent results in simulation, their performance often declines when applied in practice because real traffic does not behave exactly as it does in the training environment. Reducing this simulation-to-reality (Sim2Real) gap is therefore important for developing traffic management systems that can be transferred from research to real-world applications with confidence. Task 5.5 focuses on Sim2Real methodologies that improve the transferability of traffic management models, with particular emphasis on reinforcement learning (RL)-based control strategies. The objective is to reduce the differences between the training and deployment environments by enabling simulated traffic dynamics to better reflect those encountered in more realistic settings. To evaluate the effectiveness of the proposed methodology, the developed traffic management models are tested across microscopic traffic simulators with different levels of fidelity, ranging from computationally efficient but more abstract simulators to more detailed and realistic traffic simulation environments. The resulting Sim2Real framework is designed to support the multimodal traffic simulator developed in Task 5.1.

As a first step towards this objective, a proof-of-concept study was carried out using network-level perimeter control as a representative traffic management application. Perimeter control has received considerable attention in recent years because it regulates traffic at the network level by controlling the flow of vehicles entering a congested urban area. The effectiveness of perimeter control is associated with the Macroscopic Fundamental Diagram (MFD), which captures the relationship between network accumulation and trip-completion rate (or production) (Daganzo, 2005; Gartner & Wagner, 2004; Geroliminis & Daganzo, 2007). Under typical operating conditions, the MFD exhibits a critical accumulation at which network productivity reaches its maximum. Once this threshold is exceeded, congestion develops rapidly and network performance deteriorates. This observation has motivated the development of many MFD-based control strategies, including model predictive control, linear quadratic regulators, and proportional-integral controllers (Aboudolas & Geroliminis, 2013; Geroliminis et al., 2013; Hajiahmadi et al., 2015). These methods have shown strong performance, however they depend on accurate traffic models and reliable MFD estimation, assumptions that become difficult to satisfy under uncertain demand, heterogeneous traffic conditions, or imperfect state estimation.

Recently, Reinforcement learning (RL) has been used as an alternative because it can learn control policies directly through interaction with the traffic environment, reducing the need for explicit traffic models. Many studies have demonstrated that RL can regulate network inflow effectively, achieve performance comparable to model-based approaches, incorporate domain knowledge to improve learning, and scale to large urban networks consisting of multiple interacting regions (Kampitakis & Vlahogianni, 2025; Ni & Cassidy, 2019; Yoon et al., 2020; Zhou & Gayah, 2021, 2023). Nevertheless, the existing studies assume that the environment used for training is the same as the one in which the learned policy will eventually be deployed. As a result, the ability of RL-based perimeter control policies to generalize across environments with different traffic dynamics remains largely unexplored.

This issue is part of the sim-to-real problem and perimeter control is a natural instance of it. RL policies are typically trained in simulation, because collecting data and training policies in real traffic networks is expensive, slow, and often impractical. Differences between simulation environments and the real-world, referred to as the reality gap (Jakobi et al., 1995), can nevertheless lead to a degradation in performance after deployment. To reduce this gap, research mostly in the domain of robotics has proposed several sim-to-real transfer methods, including system identification, domain randomization, adaptive domain randomization, and domain adaptation (Chebotar et al., 2019; Peng et al., 2018; Tobin et al., 2017). For RL-based perimeter control, though, the question of how well a learned policy generalizes across environments with different traffic dynamics remains open.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

Grounded Action Transformation (GAT) is one of the most recent approaches developed to address this kind of problem (Hanna et al. 2021). Instead of modifying the source simulator or learning policies that remain unchanged across environments, GAT learns forward and inverse dynamics models from a small amount of target-domain data and uses them to transform the actions selected by the policy during training. In doing so, the source simulator more closely replicates the dynamics of the target environment, while training continues entirely in the source simulation. However, despite these developments, its potential for perimeter control has not yet been examined. From a traffic management perspective, sim-to-real techniques have been used mainly in the context of RL-based traffic signal control. More specifically, (Da et al., 2023a) have used GAT for traffic signal control, introducing forward and inverse dynamics models to make the simulator actions closer to those of the target environment without changing the simulator itself. Further work built on this approach introduced Uncertainty-aware Grounded Action Transformation (UGAT), which used the confidence of the dynamics models for selective action transformation to increase stability and transfer performance (Da et al., 2023b). Additionally, PromptGAT improved the GAT framework through the addition of large language models to enhance forward dynamics predictions through domain knowledge and thus improving adaptability to differences in weather conditions and road features (Da et al., 2024). Lastly, (Turnau et al., 2025) extended this method to Multi-Agent RL through a Joint-Local GAT (JL-GAT). Their framework uses information from nearby intersections to account for local interactions and maintains the scalability of decentralized learning. In their experiments, JL-GAT was able to reduce the sim-to-real performance gap under adverse weather conditions consistently and managed to perform better that the centralized and the decentralized GAT approaches in large traffic networks.

Despite the aforementioned developments, its potential for perimeter control has not yet been examined. To address this gap, the study proposes a sim-to-real framework for RL perimeter control based on GAT. To the best of the authors' knowledge, this is the first study to investigate sim-to-real transfer for RL-based perimeter control, extending the application of GAT to network-level traffic management. Perimeter control can also be prone to the reality gap as its effectiveness is affected by the network's macroscopic dynamics, which are a result from lower-level processes, e.g., car-following behaviour, route choice, congestion spillback, and demand patterns. Even a detailed simulator can reproduce these phenomena only approximately. Differences between simulation and reality can therefore affect the overall traffic patterns that the perimeter controller depends on. To study this transfer gap in controlled conditions, we train and test a policy across two simulators that have different dynamics, so that the same network and demand can produce different macroscopic states. The proposed framework adapts a perimeter control policy trained in a source simulator to the dynamics of a target simulator. We evaluate it in a sim-to-sim experiment with CityFlow and SUMO as the source and target simulators, respectively. Previous studies mostly assess GAT frameworks against direct policy transfer, we additionally benchmark against RL policies trained directly in the target simulator under an identical interaction budget. Finally, we test various GAT refinement frequencies and target-domain interaction budgets, in order to give some insights on how these two factors can alter transfer performance and data efficiency.

3.1.1 CONCEPTUAL FRAMEWORK

We consider a sub-network, which we refer to as the protected network (PN), embedded within a broader network that needs to be protected from congestion. Under certain conditions, there is a critical accumulation level in which the network operates most efficiently and beyond this level congestion builds rapidly leading to degraded conditions. This phenomenon is captured by the Macroscopic Fundamental Diagram (MFD) which describes the relationship between vehicle accumulation and trip completion rate. Perimeter control is the traffic management strategy adopted here with the objective of regulating the number of vehicles entering the PN in order to maintain vehicle accumulation around the critical value and prevent oversaturation. In this work this is achieved by adjusting the green duration of the phases that advance traffic into the PN where each boundary link that leads into the PN is assumed to be controlled by a traffic signal.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

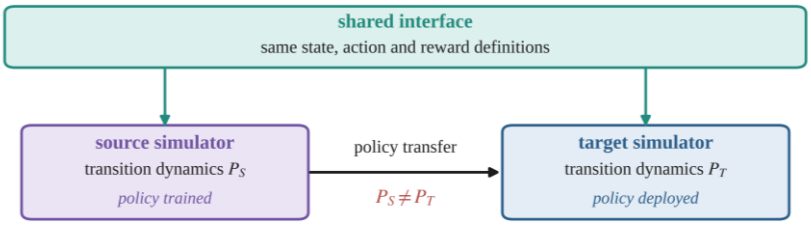

Figure 5. The transfer setting.

In the context of RL, perimeter control can be viewed as a sequential decision-making problem, which we formalize as a Markov Decision Process (MDP) characterized by (S, A, P, R, γ). An RL agent learns a control policy through repeated interaction with a traffic simulation environment. At each control cycle, it observes the current traffic state, selects a control action and receives a reward, after which the environment evolves under its traffic dynamics to the next state. The state summarizes the prevailing traffic conditions relevant to control, the action is a single, network-level quantity and the reward encodes the control objective. Through this interaction the agent learns a policy that maps states to actions so as to maximize the expected cumulative discounted reward. Crucially, we consider this MDP across two distinct simulation environments. The policy is trained in a source simulator, which we denote by S_{env}, but is intended for deployment in a target simulator, which we denote by T_{env}, whose underlying traffic dynamics differ (Figure 5). Note that the two environments share the same state and action definitions and the same reward and they differ only in their transition dynamics ($P_{S_{env}} \neq P_{T_{env}}$). This shared interface with differing dynamics is what makes transfer between them both meaningful and non-trivial, and it is the setting the proposed framework is intended to resolve.

3.1.2 THE RL-BASED PERIMETER CONTROLLER

The RL agent has three main design components: the state, the action, and the reward. At the beginning of each control step, the agent observes a state s_t which is a vector of the form

$$s_t = \left[q_t^{enter}, q_t^{in}, q_t^{out}, n_t, TC_t\right] \in [0,1]^5 \quad (1)$$

where the variables represent the total entrance queue, the total PN inflow, the total PN outflow, the PN's instantaneous accumulation, and total trip completions upon the completion of the control step, respectively. To increase stability all variables are normalized to the interval [0,1]. Also, note that the same state representation is used in both the source and target simulators. Based on the observed state s_t agent selects an action a_t. In this work, the action is the total number of vehicles that are allowed to enter the PN over the upcoming control cycle. As a rule of thumb, $a_t \in [-1,1]$ which is then linearly mapped to the feasible network inflow range according to

$$q_t^{ordered} = \left\lfloor q_{min} + \frac{a_t+1}{2}(q_{max} - q_{min}) \right\rfloor \quad (2)$$

where q_{min} and q_{max} is the minimum and maximum boundary inflow, respectively, which resulted from the minimum and maximum green time settings of the perimeter intersections assuming a saturation flow of 1800 veh/h.

After tha action a_t is applied to the environment, it advances to the next state (s_{t+1}) and a reward signal is returned also to the agent. The reward r_t is defined as the total number of completed trips during the previous control step:

$$r_t = TC_t \quad (3)$$

where r_t is normalized to $[0,1]$ for enhanced stability. This reward encourages the agent to maximize trip completion rate without explicitly given the critical accumulation.

In this study, the RL algorithm used for training the perimeter control agent is the Proximal Policy Optimization (PPO) algorithm. PPO is an on-policy, model-free, policy-gradient algorithm, belongs to the actor-critic family of RL methods (Schulman et al., 2017) and

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft / Confédération suisse / Confederazione Svizzera / Confederaziun svizra — Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER — State Secretariat for Education, Research and Innovation SERI

has shown great results in continuous control tasks. PPO improves the stability of policy-gradient learning by using a clipped objective function. The clipping operation limits how much the new policy is allowed to differ from the previous policy. In this way, PPO avoids overly aggressive policy updates and improves training robustness. The policy π_θ is represented by an actor network, which maps the current state s_t to a control action α_t, while a separate critic network estimates the value of that state, i.e., the expected cumulative reward obtainable from it under the current policy. During training, the critic's estimate acts as a baseline: the actor is updated to make actions that performed better than the critic predicted more likely, and those that performed worse less likely.

The RL-instructed inflow is uniform along the perimeter of the protected network. A queue balancing framework is introduced to optimally distribute the permitted vehicle inflow and to calculate the exact green times for the corresponding phases of the controlled intersections along the perimeter of the protected network, based on real-time observed queues. More precisely, an optimization problem (Integer Quadratic Programming - IQP) is built and solved before the beginning of each control cycle. We define the following notation for the queue balancing optimization problem. Let I be the set of perimeter control (PC) intersections and n_i^e denote the number of entrance phases at the i^{th} PC intersection, while n_i^s is the number of secondary phases (i.e., non-entrance phases). For each entrance phase k of the i^{th} PC intersection, the number of queued vehicles is denoted by q_{ik}^e, and the discharge rate (in vehicles per second) is given by s_{ik}^e. The green time assigned to each entrance phase, gt_{ik}^e, is a decision variable in the optimization and is bounded by minimum and maximum limits $gt_{ik}^{e,min}$ and $gt_{ik}^{e,max}$, respectively. Similarly, for each secondary phase k of the i^{th} PC intersection, the number of queued vehicles is denoted by q_{ik}, and the discharge rate is s_{ik}. The assigned green time for each secondary phase, gt_{ik}, is also treated as a decision variable and is constrained also by minimum and maximum green time bounds, gt_{ik}^{min} and gt_{ik}^{max}, respectively. Finally, the total green time available at PC intersection i is denoted by gt_i^{tot}, and the permitted inflow into the protected region, as instructed by the RL controller, is represented by f_{RL}. The optimization problem is formulated as follows:

$$\underset{gt_{ik}^e,\ gt_{ik}}{\text{minimize}} \quad \theta_1 \sum_{i \in I} \sum_{k=1}^{n_i^e} (q_{ik}^e - gt_{ik}^e * s_{ik}^e)^2 + \theta_2 \sum_{i \in I} \sum_{k=1}^{n_i^s} \left(q_{ik} - (gt_{ik} * s_{ik})\right)^2 \qquad (4)$$

subject to

$$\sum_{i \in I} \sum_{k=1}^{n_i^e} gt_{ik}^e * s_{ik}^e \leq f_{RL} + 5$$

$$\sum_{i \in I} \sum_{k=1}^{n_i^e} gt_{ik}^e * s_{ik}^e \geq f_{RL} - 5$$

$$\sum_{k=1}^{n_i^s} gt_{ik} + \sum_{k=1}^{n_i^e} gt_{ik}^e = gt_i^{tot} \quad \forall i \in I$$

$$gt_{ik}^e \geq gt_{ik}^{e,min} \quad \forall i \in I, k \in \{1, \dots, n_i^e\}$$

$$gt_{ik}^e \leq gt_{ik}^{e,max} \quad \forall i \in I, k \in \{1, \dots, n_i^e\}$$

$$gt_{ik} \geq gt_{ik}^{min} \quad \forall i \in I, k \in \{1, \dots, n_i^s\}$$

$$gt_{ik} \leq gt_{ik}^{max} \quad \forall i \in I, k \in \{1, \dots, n_i^s\}$$

$$gt_{ik}^e \in \mathbb{Z}^+ \quad \forall i \in I, k \in \{1, \dots, n_i^e\}$$

$$gt_{ik} \in \mathbb{Z}^+ \quad \forall i \in I, k \in \{1, \dots, n_i^s\}$$

The solution to the optimization problem in (4) yields the updated traffic signal plans that are implemented in the upcoming control step. The objective function of the optimization problem consists of two terms. The first term minimizes the number of queued vehicles on the entrance phase links, and the second term targets the reduction of queues on the secondary phase links. Each term is weighted by θ_1 (here 0.9) and θ_2 (here 0.1), respectively, allowing the formulation to emphasize these objectives differently. The

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

first and second constraints defined in (4) ensure that the assigned green durations for all entrance phases align with the RL-instructed inflow, within an allowable deviation equivalent to ±5 vehicles. The next constraint guarantees that the total green time at each PC intersection is fully allocated. The following four constraints ensure adherence to the predefined minimum and maximum green time settings for both entrance and secondary phases. Finally, the last two constraints make sure that all green time assignments, are strictly positive integers. An overview of the RL-based perimeter controller is shown in Figure 6.

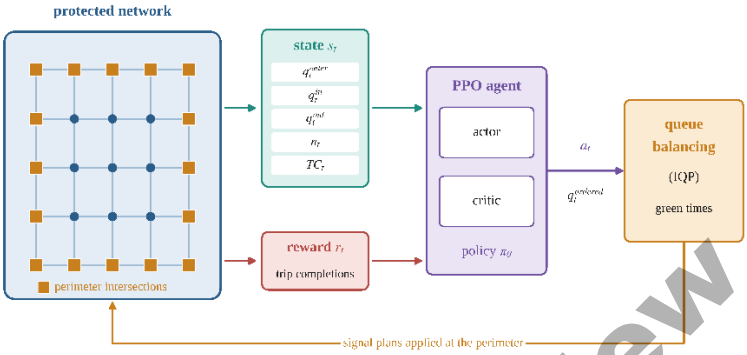

Figure 6. The RL-based perimeter control framework.

3.1.3 GROUNDED ACTION TRANSFORMATION

Under the GAT framework, adapted to our setting, there is a simulator (source) which has an imperfect and modifiable MDP that models the MDP of the environment of interest, i.e., the target simulator. Modifiable means that the source simulator has a parameterized transition probability function of the form $P_\phi(\cdot|s,a) := P_{S_{env}}(\cdot|s,a;\phi)$ where ϕ can be changed in order to produce a different simulator. GAT what effectively tries to do is to find a ϕ^* that minimizes the discrepancy between the transition probability function of the source and the target simulator. Given a dataset $\mathcal{B}_T = \{\tau_i\}_{i=1}^m$, $\tau_i = \left(s_0^i, a_0^i, \dots, s_{T-1}^i, a_{T-1}^i, s_T^i\right)$ which is collected by rolling out policy π_θ in the target simulator then

$$\phi^* = \underset{\phi}{\operatorname{argmin}} \sum_{\tau_i \in \mathcal{B}_T} \sum_{t=0}^{T-1} d\left(P_{T_{env}}\left(s_{t+1}^i | s_t^i, a_t^i\right), P_{S_{env}}\left(s_{t+1}^i | s_t^i, a_t^i; \phi\right)\right) \quad (5)$$

where d is a distance metric that measures the discrepancy between the transition dynamics $P_{T_{env}}$ of the target simulator and the transition dynamics $P_{S_{env}}$ of the source simulator. The central idea of GAT is to augment the source simulator with a differentiable action-transformation function g. Rather than executing the agent's action directly, the simulator executes g's transformed action, chosen so that the resulting transition in the source matches the one the same original action would have produced in the target. In effect, g grounds the source dynamics in those of the target. The function is realized as a parameterized function approximator (i.e., neural network), and its parameters constitute the grounding parameters ϕ of the augmented simulator, so GAT needs no access to simulator internals. More precisely, the grounded action transformation function is defined as,

$$\hat{a}_t = g(s_t, a_t) = H_{S_{env}}\left(s_t, s_t + F(s_t, a_t)\right) \quad (6)$$

where $\hat{a}_t$ is the grounded action returned by g given a state s_t and action a_t $(= \pi_\theta(s_t))$. At this point we should emphasize that PPO is updated with a_t, not $\hat{a}_t$. The transformer belongs to the environment, not the policy, and together with the source simulator form the grounded environment.

The grounded action transformation function includes two distinct functions, i.e., the forward model F and the inverse model $H_{S_{env}}$ as it is also shown in Figure 7. Each model is approximated with a deep neural network with parameters ϕ_F and ϕ_H, respectively $(\phi = \phi_F \cup \phi_H)$. The forward model captures how the target environment responds to control actions. It is a mapping $F: S \times A \to X$,

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

where $X \subset \mathbb{R}^5$ denotes the space of state changes, and it is fitted exclusively on transitions recorded in the target simulator. It predicts the state change $\widehat{\Delta s}_{t+1}$ that would result from applying an action a_t being in state s_t, and is defined as

$$F(s_t, a_t) = \widehat{\Delta s}_{t+1} \quad (7)$$

From Eq. (7) the predicted next state $\hat{s}_{t+1}$ follows as $s_t + F(s_t, a_t)$. This approach follows the effect-space formulation of (Hanna et al., 2021), predicting the change rather than the absolute next state avoids requiring the network to reproduce the current state at its output and retains resolution, since consecutive states differ by a small fraction of the normalised range.

Given a set $\mathcal{B}_T$ of transitions (s_t, a_t, s_{t+1}) that are collected in the target environment by applying policy π_θ, the parameters ϕ_F of the forward model are obtained by minimizing

$$\mathcal{L}_F(\phi_F) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{B}_T} \left[\| F(s_t, a_t) - (s_{t+1} - s_t) \|_2^2 \right] \quad (8)$$

The model is fitted on target states but, during policy training, is evaluated at source states, as the target simulator is not part of that stage. This is admissible because both environments share the same state representation, and because the data collection procedure exposes the model to the state region the policy visits. As predicted state changes may carry $s_t + F(s_t, a_t)$ outside the unit box, the predicted next state is clipped to $[0,1]^5$ before being passed to the inverse model.

The inverse model predicts the action that reproduces a given transition in the source environment. It is a mapping $H_{S_{env}}: S \times S \rightarrow A$, and it is fitted on transitions that are recorded in the source simulator. Given the reconstructed next state $\hat{s}_{t+1}$ and the current state s_t the inverse model predicts the grounded action $\hat{a}_t$ which eventually is applied to the source simulator. It is defined as,

$$\hat{a}_t = H_{S_{env}}(s_t, \hat{s}_{t+1}) \quad (9)$$

Given a set $\mathcal{B}_S$ of transitions (s_t, a_t, s_{t+1}) that are collected in the source environment by applying policy π_θ, the parameters ϕ_H of the inverse model are obtained by minimizing

$$\mathcal{L}_H(\phi_H) = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{B}_S} \left[\left\| H_{S_{env}}(s_t, s_{t+1}) - a_t \right\|_2^2 \right] \quad (10)$$

During policy training the model is queried with an observed source state together with the reconstructed next state by utilizing F, and the resulting action is clipped to $[-1,1]$. Together, the two models define the grounding function of Eq. (6). Applying $g(s_t, a_t)$ in the source simulator instead of a_t aligns one-step transition behaviour between the two environments,

$$P_{S_{env}}(\cdot \,|s_t, g(s_t, a_t)) \approx P_{T_{env}}(\cdot \,|s_t, a_t) \quad (11)$$

so, the policy is optimised against source dynamics aligned with those of the target and is expected to transfer with a smaller performance gap.

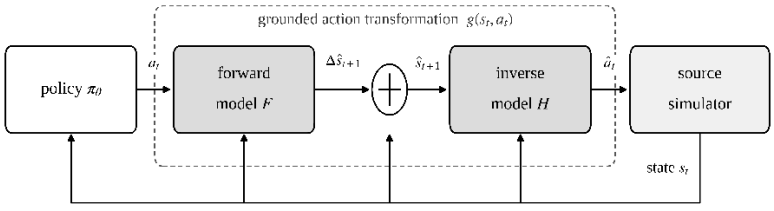

Figure 7. Overview of the grounded environment.

3.1.4 TRAINING PIPELINE

The training process follows an iterative procedure, and each iteration can be divided into two distinct steps as shown in Algorithm 1. The grounding step where the forward and inverse models are fitted from data that are generated by the target and source

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

simulators, respectively, by the current policy, and the policy refinement step. A single grounding step is not sufficient, because the forward and inverse models are fitted from data generated by the policy while the policy itself changes during optimization. Models fitted at one iteration therefore become progressively less representative of the states the policy subsequently visits. This creates the need for alternating between grounding and policy improvement over K iterations. The policy π_θ is first pre-trained for M steps in the source simulator without transformation, resulting in an initial policy π_{θ_0} that is competent in the source environment and can be used to generate informative transitions. Each subsequent iteration $k = 1, \dots, K$ then collects transitions from both simulators under $\pi_{\theta_{k-1}}$, refits the two models according to Eqs. (8) and (10) and optimises the policy for a fixed number of PPO steps in the grounded source environment, continuing from θ_{k-1}. Collecting under the current policy concentrates the data on the states the policy visits, so that both models are fitted where they are subsequently queried. The procedure returns π_{θ_K}, which is evaluated in the target simulator with the transformation removed.

The quantity that controls the cost of the framework is the number of transitions collected from the target simulator, since in a deployment setting this is the only data that must be gathered from the environment of interest. This is expressed through a data ratio $a \in (0,1]$, defined relative to the number of interactions required to train a policy directly in the target simulator, N_{max}. The total target budget is aN_{max}, divided equally across iterations, so each iteration contributes aN_{max}/K transitions. Thus, a value of $a = 1$ is a framework that consumes exactly as much target data as direct training in the target simulator, while $a < 1$ quantifies the reduction in target-data requirement obtained through grounding.

Algorithm 1: Iterative grounding for perimeter control

Input: source simulator S_{env}, target simulator T_{env}, iterations K, data ratio a, target budget N_{max}, pre-training steps M

Output: π_{θ_k}

$\theta_0 \leftarrow \text{PPO}(S_{env})$ for M steps — *pre-train in source, no transformation*

for $k = 1$ **to** K **do**

  $\mathcal{B}_T^k \leftarrow$ collect aN_{max}/K transitions from T_{env} under $\pi_{\theta_{k-1}}$ ▶ *Grounding*

  $\mathcal{B}_S^k \leftarrow$ collect aN_{max}/K transitions from S_{env} under $\pi_{\theta_{k-1}}$

  $\phi_F \leftarrow \text{argmin}\, \mathcal{L}_F(\phi_F)$ on B_T^k - Eq. (8)

  $\phi_H \leftarrow \text{argmin}\, \mathcal{L}_H(\phi_H)$ on B_S^k - Eq. (10)

  $g(s,\text{a}) \leftarrow H_{S_{env}}\left(s, clip\left(s + F(s,a)\right)\right)$ - Eq. (6)

  $\theta_k \leftarrow \text{PPO}(\theta_{k-1}, S_{env} \circ g)$ for N_{max}/K steps ▶ *Policy refinement*

end for

return π_{θ_k}

3.2 EXPERIMENTAL SETUP

3.2.1 SIMULATION ENVIRONMENTS

In this study, CityFlow serves as the source simulator and SUMO as the target. CityFlow is an open-source traffic simulator designed for large-scale and computationally efficient traffic experiments (Zhang et al., 2019). SUMO is also an open-source traffic simulator widely used for modeling urban traffic flow and vehicle interactions (Krajzewicz et al., 2012). Both are microscopic and both use car-following models of the Krauß family, but they were built for different purposes. CityFlow focuses on computational efficiency and can achieve simulation speeds more than twenty times faster than SUMO, making it well suited for the repeated interactions required during RL training. In contrast, SUMO offers a richer set of behavioral models and more detailed junction dynamics, providing a more realistic representation of traffic operations. These characteristics defines the assignment of roles. The source must

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

be fast enough to support policy optimisation, and the target is the more realistic representation of the system of interest. This resembles the asymmetry between a training simulator and a real network. Using a second simulator makes it possible to quantify the transfer gap under controlled conditions, since its dynamics can be sampled repeatedly to establish a reliable performance benchmark, something that would be impractical in a real traffic network.

First experiments showed that CityFlow could not reproduce network recovery under oversaturation and queues spilled back across successive intersections into circular blocking configurations, in which each vehicle is obstructed by another along a closed path. When these states occurred , they could not be self-resolved, so no further trips complete and the decongestion branch of the MFD is never observed. Thus, three extensions were implemented. A teleportation mechanism removes a vehicle stopped for longer than a predefined time limit and reinserts it on the next available edge of its route, releasing the blocking configuration without changing its origin or destination. Adaptive rerouting reassigns active vehicles with a given frequency a route to their destination using Dijkstra's algorithm on edge travel times, with a tolerance of preventing unnecessary rerouting between almost equivalent alternatives. An output module was also added to produce metrics comparable to those reported by SUMO and both behavioural extensions replicate mechanisms that are already provided in SUMO.

3.2.2 DESCRIPTION OF THE TESTBED NETWORK

The testbed is a 7×7 grid network, it includes 49 signalized intersections and 112 bidirectional edges with 200m length and two lanes per direction (Figure 8(a)). The network also includes 28 external boundary nodes, which are used as entry and exit points for vehicles. All intersections are controlled by fixed-time traffic signals with a 90 second cycle which is also the control step duration. Each signal has four green phases, separated by 3 second yellow intervals. For perimeter-control experiments, the 24 intersections on the boundary of the grid are treated as perimeter intersections. Note that for the perimeter intersections, one green phase per approach is assigned to better differentiate entrance from no-entrance movements. The same network topology is used in both SUMO and CityFlow.

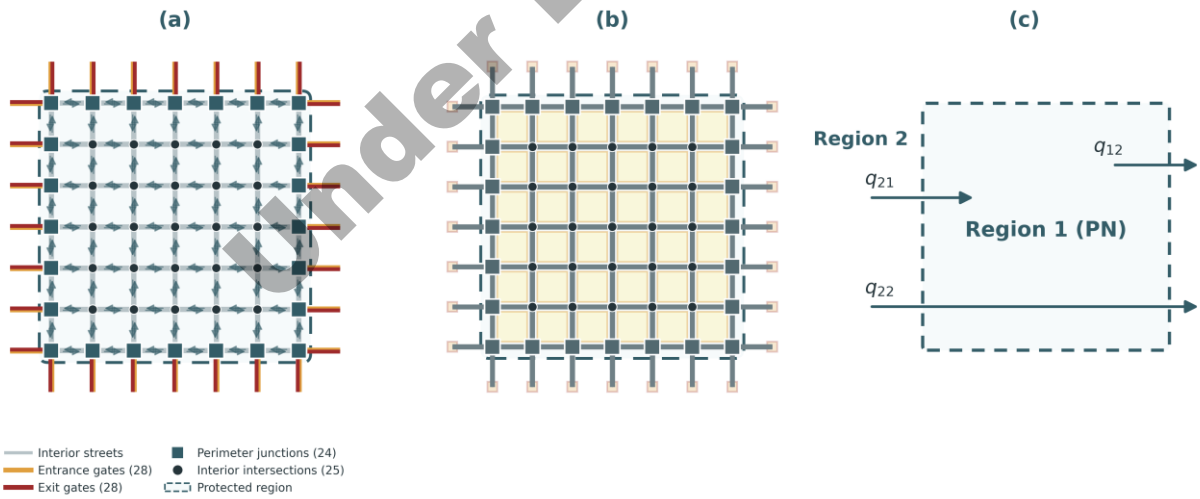

Figure 8. The test-bed network (a); The origin-destination zones (b); Trip-type synthesis (c)

Origin-destination zones are placed on the inside and the outside the PN. The inside area is divided into 36 internal zones, corresponding to the blocks of the 7x7 grid network, while 28 external zones are placed at the boundary entry and exit edges (Figure 8(b)). Three types of trips are considered in this study (Figure 8(c)): trips entering the PN from outside (q21), trips exiting the PN (q12), and through trips that start outside the PN and end outside it after crossing the network (q22). Internal trips, where both origin and destination are inside the PN, are not included in this study. The demand is mainly incoming to the PN, resembling morning

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

rush-hour conditions. In the main demand period, about 68% of the trips are outside-to-inside, 15% are inside-to-outside, and 17% are outside-to-outside through trips (Figure 9).

The demand is consisted of a 15-minute warm-up period followed by a 1-hour peak period and after that the demand drops to zero. A total of 24561 trips are generated including the warm-up period. External origins and destinations are distributed almost uniformly across the four sides of the network, with the assumption that a vehicle that enters from one side does not exit the network from the same side. Initial routes are generated in SUMO using macroscopic route assignment with stochastic user equilibrium, allowing up to five alternative routes per trip. During simulation, dynamic rerouting is also enabled every 180 s so vehicles can adapt their routes according to traffic conditions. The same demand pattern is applied to both simulations.

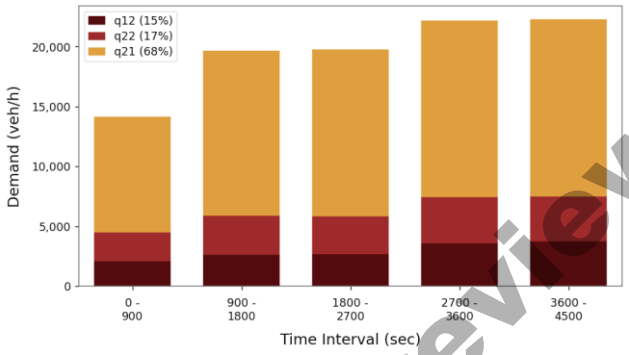

Figure 9. Demand profile

Figure 10 presents the MFDs obtained from 10 runs in SUMO and CityFlow without perimeter control. Both simulations run until all trips have completed. It is evident that both simulators exhibit a fully developed MFD that has a unimodal shape, but differences can still be observed between the two MFDs. SUMO produces a sharply peaked MFD. The maximum observed trip completion rate is ~4.5veh/s and the critical accumulation is around 3000 veh with a maximum value of 8000 veh. CityFlow, in contrast, yields a flatter and broader profile. The peak trip completion rate is lower, at approximately 3.5 veh/s and the critical accumulation is closer to 3800 veh. Its congested branch is also shorter and noisier. Accumulation reaches around 6500 veh and empties more quickly. This shows that even if the complete information regarding the network topology and demand is given from the target to the source environment differences in traffic evolution and network performance can still emerge due to the difference in the underlying dynamics. The differences between the two simulators are evident and confirm that SUMO and CityFlow induce distinct macroscopic dynamics, making the pair a suitable testbed for evaluating sim-to-real transfer in RL-based perimeter control.

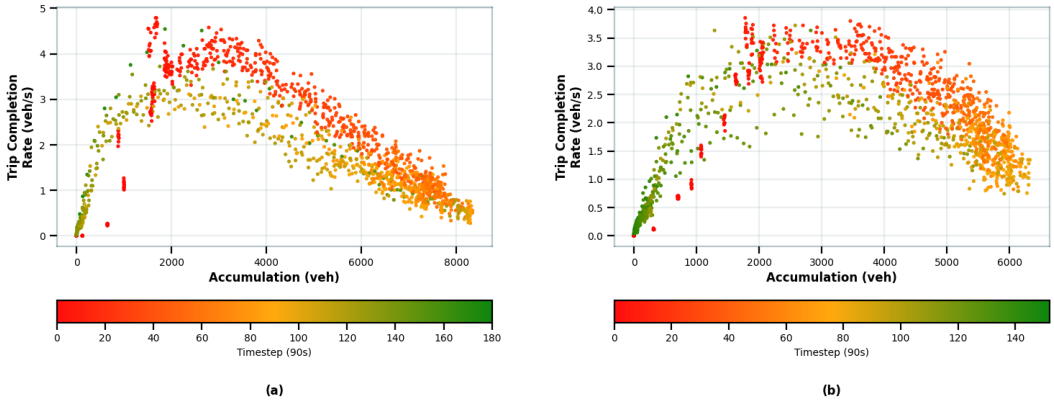

Figure 10. MFD (10 replications) without perimeter control for SUMO (a) and CityFlow (b)

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

3.2.3 TRAINING SETUP

Three training configurations were developed: (i) PPO trained directly in SUMO, (ii) PPO trained in CityFlow and directly transferred to SUMO, and (iii) PPO trained in CityFlow using the proposed GAT framework. To make a fair comparison between the different trainings, all RL policies were trained for 24000 ($N_{max} \equiv M$) environment steps. GAT was evaluated using both three- and five-iteration schedules ($k = 3 \,\&\, 5$). Moreover, 5 data ratios ($\alpha = 0.2 - 1.0$) were considered. The combination of k and a hyperparemeters led to 10 different GAT configurations. Both the forward and inverse model are approximated by deep neural networks. The architecture and the hyperparameters for the GAT model where kept the same across the different GAT configurations and are presented in Table 3. Finally, the same PPO hyperparameters were used for all training configurations regarding the policy training and are presented in Table 3.

Table 3. Neural network architectures and training parameters.

PARAMETER	PPO (ACTOR/CRITIC)	FORWARD MODEL	INVERSE MODEL
Input / output dimensions	5 / 1	6 / 5	10 / 1
Hidden layers	[64, 64]	[64, 64]	[64, 64]
Activation	tanh	ReLU	ReLU
Optimizer	Adam	Adam	Adam
Learning rate	0.0003	0.001	0.001
Loss	clipped surrogate	MSE	MSE
Mini-batch / batch size	20	64	64
Epochs per update / per fit	10	100	100
PPO algorithm			
Rollout length	160 steps		
Discount factor, γ	0.99		
GAE coefficient, λ	0.95		
Clipping range	0.20		
Value-function coefficient	0.50		
Entropy coefficient	0.00		
Maximum gradient norm	0.50		

To verify the efficiency of the proposed method 10 replications with different random seeds were run in the target environment (SUMO) to ensure that any potential improvement is not coincidental, as different initial seeds may lead to different traffic conditions due to the intrinsic stochasticity of the simulator which is also observed in real networks.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft / Confédération suisse / Confederazione Svizzera / Confederaziun svizra — Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER — State Secretariat for Education, Research and Innovation SERI

3.3 RESULTS

Table 4 and Table 5 summarize the performance of all trained configurations deployed in the target environment, i.e., SUMO, for $k = 5$ and $k = 3$ iterations, respectively. The comparison is based on four network-level performance indicators: i. Vehicle Hours Travelled (VHT), ii. Total Travel Time (TTT), iii. Total Depart Delay (TDD), and iv. average network speed (Speed). In both tables, the policy trained directly in SUMO (Target-trained) is used as the reference baseline. By comparing the reference baseline with the direct transfer, we can observe that performance is deteriorated considerably showing that the performance gap is real. More precisely, VHT, TTT, and TDD increased by 15.89%, 15.91%, and 15.87%, respectively, while the average network speed decreased by 6.68%.

Table 4. Performance of the 5-iteration GAT control policies

Control Scenario	VHT	% Diff.	TTT (h)	% Diff.	TDD (h)	% Diff.	Speed (km/h)	% Diff.
SUMO trained model	7784	-	4501	-	3283	-	7.93	-
GAT, α=1.0	**7687**	**-1.25**	**4843**	**7.60**	**2844**	**-13.37**	**7.67**	**-3.28**
GAT, α=0.8	8269	6.23	4457	-0.98	3812	16.11	7.98	0.63
GAT, α=0.6	8168	4.93	4440	-1.36	3728	13.55	7.99	0.76
GAT, α=0.4	8385	7.72	4971	10.44	3414	3.99	7.57	-4.54
GAT, α=0.2	9083	16.69	4576	1.67	4506	37.25	7.88	-0.63
CityFlow direct transfer	9021	15.89	5217	15.91	3804	15.87	7.40	-6.68

Table 5. Performance of the 3-iteration GAT control policies

Control Scenario	VHT	% Diff.	TTT (h)	% Diff.	TDD (h)	% Diff.	Speed (km/h)	% Diff.
SUMO trained model	7784	-	4501	-	3283	-	7.93	-
GAT, α=1.0	7859	0.96	4516	0.33	3343	1.83	7.91	-0.25
GAT, α=0.8	8182	5.11	5398	19.93	2785	-15.17	7.39	-6.81
GAT, α=0.6	7921	1.76	4487	-0.31	3434	4.60	7.94	0.13
GAT, α=0.4	8422	8.20	4502	0.02	3921	19.43	7.93	0.00
GAT, α=0.2	8760	12.54	6123	36.04	2637	-19.68	7.08	-10.72
CityFlow direct transfer	9021	15.89	5217	15.91	3804	15.87	7.40	-6.68

Introducing GAT largely closes this performance gap. With the exception of the lowest data ratio ($\alpha = 0.2$) for the case of 5 iterations, all GAT configurations outperformed the direct-transfer policy. The best results were obtained with the 5-iteration framework using α = 1.0, which reduced VHT by 1.3% relative to the policy trained in directly in the target environment. Although GAT is intended to bridge the dynamics gap, the results suggest that iterative grounded action transformation can improve policy robustness when

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

sufficient target-domain data are available. This observation is in-line with previous sim-to-real studies that showed that exposure to more diverse dynamics enhances policy generalization (Andrychowicz et al., 2020; Peng et al., 2018; Tobin et al., 2017).

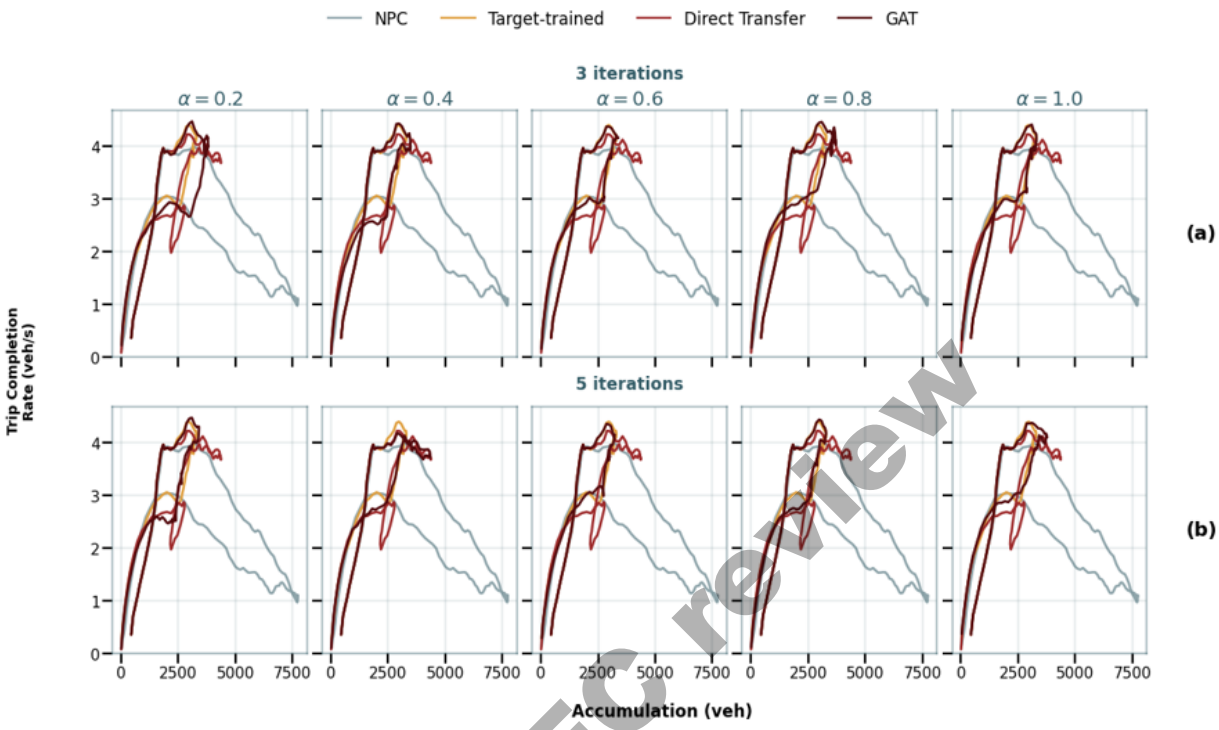

Figure 11. MFD trajectory for the 5- (a) and 3-iteration (b) GAT under different target-data budgets (α)

In general, transfer performance improved as more target-domain data became available. Ranked by their VHT gap to the target-trained policy, the configurations order identically under both schedules: α = 1.0, α = 0.6, α = 0.8, α = 0.4 and α = 0.2. The ordering is therefore monotonic in the target-data budget apart from a single transposed pair, which recurs in both settings: α = 0.6 achieved lower VHT than α = 0.8 by 1.3 and 3.4 percentage points under the 5- and 3-iteration frameworks respectively. For the 3-iteration framework, α = 0.6 offered the best trade-off, increasing VHT by just 1.8% compared with the target-trained policy. In contrast, the α = 0.2 configurations remained well below the performance of the baseline, increasing VHT by 16.7% and 12.5% for the 5- and 3-iteration frameworks respectively; the former is exceeded even by direct transfer at 15.9%. This indicates that very small target datasets are not sufficient to learn dynamics models that can support effective policy transfer, and that below a certain sample size the transformation ceases to assist the policy at all. These results indicate that transfer performance depends both on how many target transitions are collected and also on which transitions they are, and they motivate two directions: selecting the transitions that are most informative for fitting the dynamics models rather than sampling them uniformly and identifying when a policy should be evaluated in the target environment, since the target-data budget (data ratio) alone does not reliably indicate when grounding has converged.

Figure 11 shows the evolution of the network operating state (MFD trajectory) for all trained configurations against the case where no perimeter control is applied to the network (NPC). As expected, the network in the NPC case quickly exceeds the optimal operating region, causing trip completion rate to decline and enters the congested regime. On the other hand, the controller trained directly in the target environment maintains the network close to the high-production region for most of the simulation. The direct transfer policy at first follows a similar trajectory but begins to diverge as congestion develops, showing the differences in traffic dynamics between the two simulators. The GAT framework narrows this gap. The trajectories increasingly resemble those of the target-trained

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

controller, as more target-domain data become available, indicating that the transferred policy can better captures the target traffic dynamics. This trend is most pronounced for $\alpha = 1.0$, where both the loading and recovery phases closely match the reference trajectory. In contrast, the $\alpha = 0.2$ configurations deviate more noticeably, suggesting that the limited target data are not sufficient to accurately learn the target dynamics.

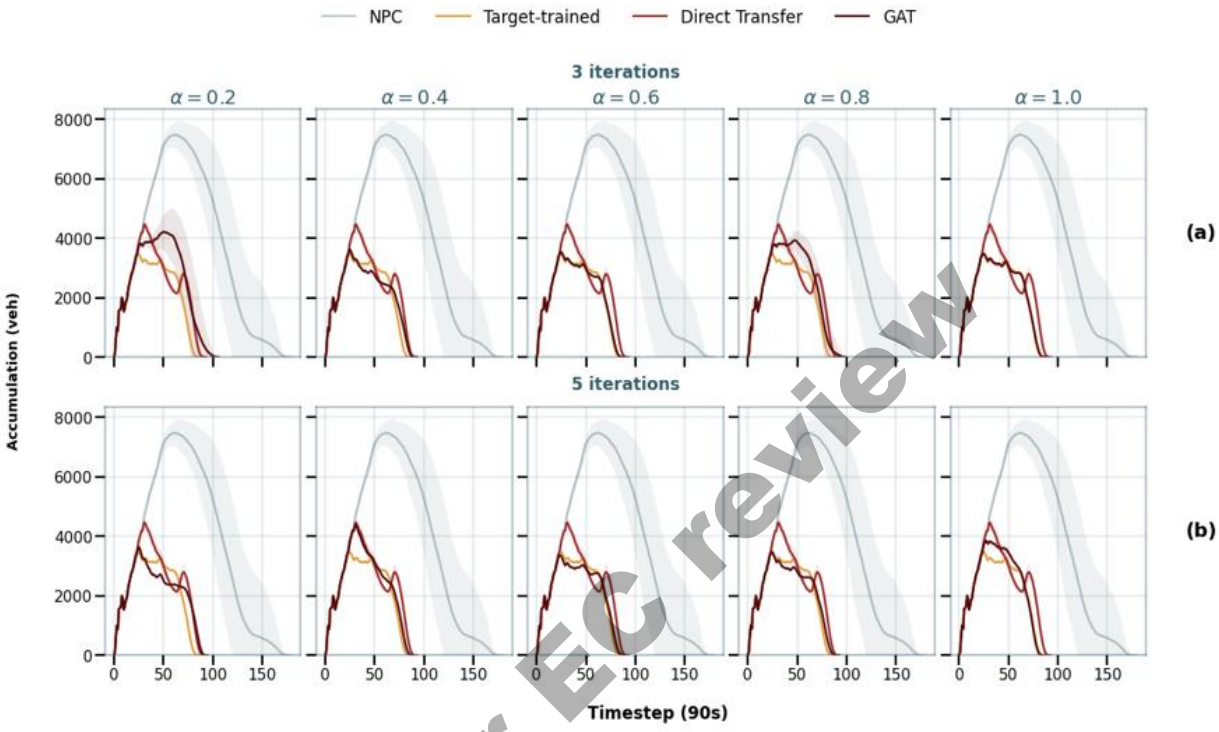

Figure 12. Accumulation over time for the 5- (a) and 3-iteration (b) GAT under different target-data budgets (α)

Figure 12 shows the evolution of network accumulation for all trained configurations against NPC. Without perimeter control, accumulation increases continuously during the demand period, reaching approximately 7500 vehicles before gradually declining. The target-trained controller maintains accumulation close to the critical operating level, whereas the direct transfer controller exhibits larger fluctuations during the start and recovery of congestion, reflecting the dynamics mismatch between the two simulators. The proposed GAT framework progressively reduces these differences as the amount of target-domain data increases. The $\alpha = 1.0$ configuration provides the closest match to the target-trained controller, while the $\alpha = 0.6$ and $\alpha = 0.8$ configurations also maintain the network close to the desired accumulation level.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

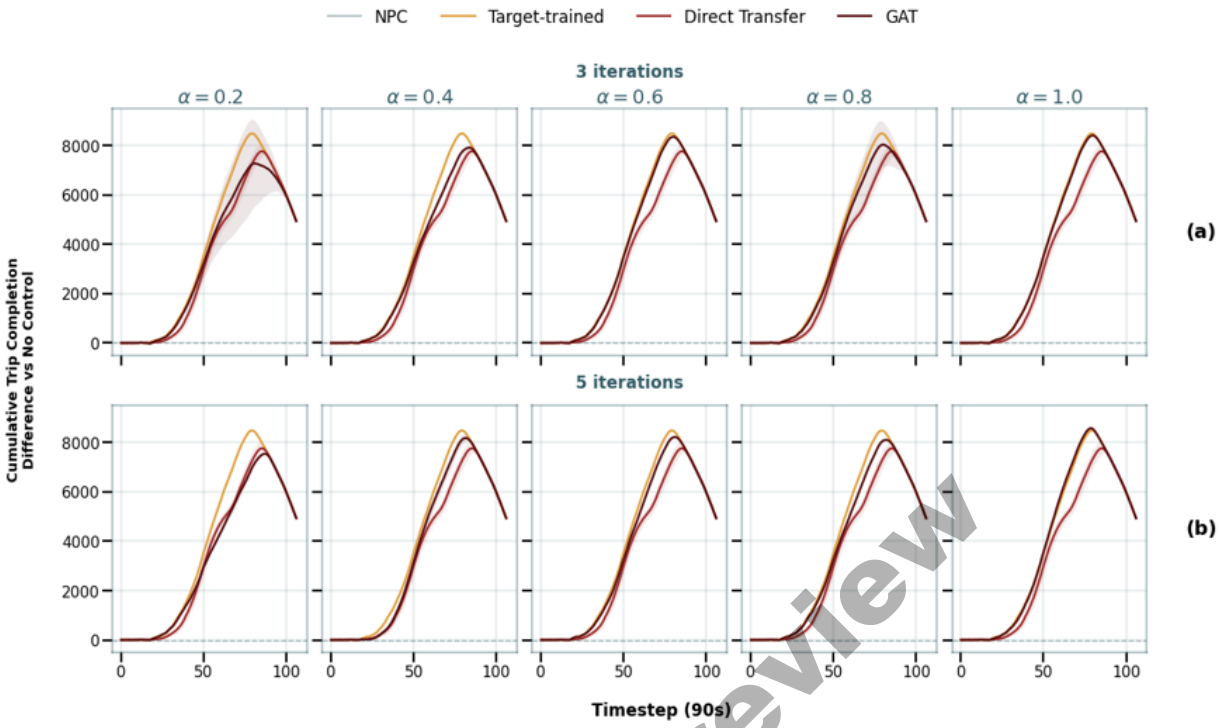

Figure 13. Difference in cumulative trip completion for the 5- (a) and 3-iteration (b) GAT under different target-data budgets (α)

Figure 13 illustrates the difference in cumulative trip completions relative to the NPC case. The GAT policies almost progressively recover the production profile of the target-trained controller. Across both the three- and five-iteration frameworks, they maintain higher trip-completion rates than the direct-transfer policy, with the α = 1.0 configuration providing the closest match to the target-trained controller as it almost overlaps with the target-trained policy and briefly exceeds it during the peak congestion period, indicating a slightly higher vehicle completion rate over that interval.

3.3.1 EFFECT OF THE GAT METHOD ON ACTIONS

Figure 14 shows how the proposed GAT framework improves perimeter-control performance by comparing the evolution of network accumulation with the corresponding ordered inflow actions. For the sake of brevity, only the results of the 5-iteration GAT framework are presented, as the 3-iteration framework exhibits the same overall behaviour. Figure 14(a) shows the evolution of network accumulation, and Figure 14(b) presents the corresponding inflow actions generated by each trained PPO agent.

The direct transfer policy initially applies inflow actions that are appropriate for the source simulator but overly permissive for the target environment. Consequently, network accumulation exceeds the critical accumulation of the target simulator, allowing congestion to develop beyond the level that maximizes network production. The policies trained under GAT gate more conservatively as accumulation builds, holding it closer to the target's critical point, and relax that restriction as the network discharges without producing a second period of excessive accumulation. This behaviour is a property of the learned policies rather than of the transformation itself, which is applied only during training.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

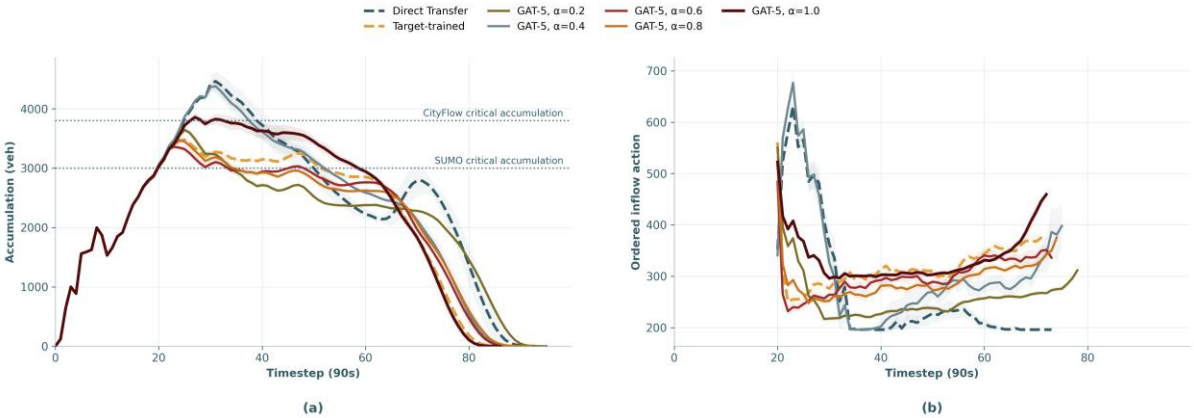

Figure 14. (a) Evolution of network accumulation under the different trained PPO policies; (b) The trained-PPO agent's actions.

The magnitude of this adjustment is not monotone in the target-data budget. The $\alpha = 0.2$ configuration is the most conservative, limiting accumulation effectively but delaying recovery, and α = 0.4 remains close to the direct transfer profile, whereas $\alpha = 0.6$, $\alpha = 0.8$ and $\alpha = 1.0$ balance restriction during congestion against the restoration of throughput as conditions improve. Reproducing the actions of the target-trained policy is not an objective of the framework, and these profiles are therefore best read as a diagnostic of the grounding rather than a measure of transfer quality. An important finding though is that the $\alpha = 1.0$ attained the lowest VHT while departing more from the target-trained action profile compared to $\alpha = 0.6$ and $\alpha = 0.8$.

3.3.2 GROUNDED ACTION TRANSFORMATION FUNCTION SENSITIVITY ANALYSIS

A sensitivity analysis was conducted to understand how the proposed GAT framework changes the actions selected by the original policy. Since GAT adjusts the control action before it is applied in the source environment, this analysis provides insight into whether these modifications are negligible, random, or follow a systematic pattern. The analysis focuses on the best-performing configuration (5-iteration GAT with α = 1.0). Figure 15 was generated by sampling traffic states and their corresponding raw actions (PPO) from the evaluation trajectories and passing them through the fitted grounded action transformation function.

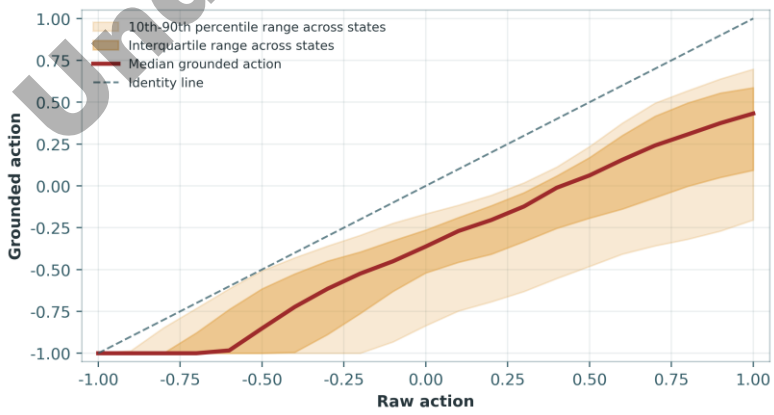

Figure 15. Sensitivity analysis of the fitted grounded action transformation function for the best-performing GAT configuration.

For each raw action, the resulting grounded actions were collected across a range of traffic states. The solid red line shows the median grounded action. The dark shaded area is the interquartile range, while the lighter area shows the 10th to 90th percentiles. The dashed diagonal is the identity line, i.e., points on this line are left unchanged by GAT.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The grounded actions mostly remain below the identity line, meaning that GAT does not pass the original actions through unchanged. This behaviour becomes clear especially for positive raw actions, where the grounded action increases more gradually and remains lower than the original value. As a result, actions that would allow high inflows are moderated before being executed in the source environment. In contrast, strongly negative raw actions rapidly approach the lower bound of -1, suggesting that highly restrictive control decisions are preserved whenever they are needed. The shaded ranges also show that the adjustment changes with the traffic state. The same raw action does not always produce the same grounded action under different traffic conditions. The fitted transformation is thus state-dependent, with GAT adjusting the policy action based on both the original action and the current state of the network.

3.4 INTEGRATION INTO THE FEDORA MODULAR SIMULATION PLATFORM

Within the FEDORA modular simulation platform described in Task 5.4, the proposed Sim2Real perimeter-control framework can be naturally integrated as a pluggable logic module, while each traffic simulator or deployment platform is represented as an execution environment. The logic module encapsulates the trained reinforcement learning controller and, when required, the Grounded Action Transformation (GAT) component, exposing the standard input-output interface expected by the orchestration framework, rendering them compatible for a possible later integration.

During each control interval, the Orchestrator requests the current traffic state from the connected environment. For perimeter control, this state includes the information required by the RL controller, such as entrance queue lengths, protected-network inflow and outflow, network accumulation, trip-completion rate, and, when available, the status of perimeter traffic signals or detectors. The Orchestrator forwards this information to the logic module, which generates a corresponding logic command, for example the desired perimeter inflow or the associated signal settings after the queue-balancing procedure. When the GAT component is enabled, the raw action produced by the PPO policy is first transformed into a grounded action before being translated into executable control commands. The resulting command is then returned to the environment through the apply and advance interface, where it is implemented for the next control interval.

An advantage of this architecture of this architecture is that the same control logic can operate across different execution backends without requiring modifications to the controller itself. In a computationally efficient simulator such as CityFlow, the environment provides abstract network-level traffic states and accepts perimeter inflow or green-time commands, making it well suited for policy training. In a more detailed microscopic simulator such as SUMO, the environment supplies richer information on vehicle movements, queues, traffic signals, and network performance, allowing more realistic evaluation and serving as the target domain for Sim2Real experiments. In a real-world deployment, the simulator can be replaced by a field-interface layer that converts detector measurements, signal-controller data, and traffic management system information into the same traffic state representation, while translating the controller's logic command into valid traffic signal or inflow-control instructions.

Throughout this process, the Orchestrator is responsible for coordinating communication between modules, managing execution timing, handling lifecycle operations, logging interactions, and, when necessary, polling the environment for updated state information. At the same time, the Recorder stores communication traces together with traffic and performance data for subsequent analysis. This modular architecture demonstrates how the FEDORA platform facilitates the transfer of intelligent traffic management applications across different environments. Once implemented as a logic module, the same perimeter-control strategy can be trained in a source simulator, evaluated in a higher-fidelity target simulator, or ultimately deployed in a real traffic management system through the same standardized communication interface.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

3.5 CONCLUSION

Task 5.5 investigates the use of Sim2Real reinforcement learning to improve the transferability of network-level traffic management strategies across environments with different traffic dynamics. As a proof of concept, the approach was evaluated on the perimeter control problem, where a reinforcement learning controller trained in one microscopic traffic simulator was transferred to another using a GAT framework.

The experimental results showed that differences in traffic dynamics between simulation environments can lead to a considerable loss in performance when a trained policy is transferred directly. The proposed Sim2Real framework effectively reduced this performance gap by adapting the controller's actions to the characteristics of the target environment. Across all evaluated scenarios, the framework consistently outperformed direct policy transfer and, under the best-performing configuration, achieved performance comparable to, and in some cases better than a controller trained directly in the target simulator. The action-level analysis further revealed how the learned transformation modifies perimeter inflow according to the level of congestion, allowing the controller to keep network accumulation closer to its optimal operating region.

Beyond the proof-of-concept study, this work demonstrates how Sim2Real reinforcement learning can contribute to the broader objectives of FEDORA by enabling traffic management applications to be transferred across different simulation environments without requiring complete retraining. The proposed controller was also conceptually integrated into the modular simulation platform developed in Task 5.4, illustrating how the same control logic can be connected to different simulators and ultimately, to real traffic management systems, through a common orchestration interface.

Although the proposed framework was evaluated using a synthetic grid network and two microscopic traffic simulators, the results show the potential of Sim2Real techniques to improve the robustness and practical deployment of RL-based traffic management systems. These results can be used as the basis for several future research directions. Since the smallest budgets failed while intermediate ones occasionally outperformed larger ones, showed that transfer depends on which target transitions are collected and not only on the number of the collected data. Consequently, a next step could be to select transition data based on how informative they are for the training of the dynamics models. A convergence test could also be beneficial to stop grounding when further iterations are no longer yielding better results. Additionally, accounting for model uncertainty by transforming actions selectively where it is low can be a next step for environments whose dynamics differ more than those examined in this study, because the transformation is deterministic and fitted under a squared-error loss. Deployment on a real network could show how to obtain target transitions without relying on costly or potentially disruptive exploratory control actions, showing the importance of logged operational data and safety-aware data collection. Finally, extension to multi-region perimeter control with many interacting agents, and to settings where demand and topology also differ, could show whether grounding is also effective when the differences extend beyond the traffic dynamics.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

4 FORESIGHT ANALYSIS SCENARIOS GENERATION AND EVALUATION

Task 5.6 develops the foresight scenario generation and evaluation capability required to assess FEDORA solutions under plausible future mobility conditions. The work connects scenario design, microscopic multimodal simulation, UAV service modelling and automated KPI extraction in a repeatable in-silico environment. The present Chapter documents the rationale, state of the art, methodology, current implementation, integration pathway and next development steps of the platform.

4.1 TASK DESCRIPTION

Task 5.6 realises Expected Innovation EI4.6 (A foresight analysis scenarios generator designed to explore use cases of emerging mobility concepts in diverse traffic conditions, enabling a deeper understanding of how these concepts adapt to varying needs and strategies) by providing a scenario-driven evaluation environment for network-wide foresight analysis. It is designed to test and evaluate FEDORA solutions, including the optimisation and control functions developed across WP4, and the modular orchestration concepts of Task 5.4. Its immediate application is the Nicosia pilot (Pilot #3) in Task 6.5, where UAV passenger services, drone-based traffic sensing, vertiport placement and innovative solution of multimodal transfers can be investigated. The task is therefore not limited to producing a single simulation case. It establishes a scenario engine through which demand growth, incidents, network restrictions, UAV fleet design, service capacity, behavioural adoption and control settings can be combined into reproducible what-if experiments. The resulting evidence is intended to identify the conditions under which a FEDORA solution provides measurable benefits, the operational limits that may remove those benefits, and the configurations that should be prioritised for pilot deployment. Hence, the capabilities of the foresight generation engine include, but are not limited to the following:

- Generation of plausible combinations of future mobility demand, disruptions, service configurations, and traveller behaviour.
- Execution of controlled, uncontrolled, and UAV-assisted scenarios within a common microscopic simulation environment.
- Comparison of road-only and multimodal person journeys on a door-to-door travel-time basis.
- Quantification of network, traveller, multimodal-service, and UAV-service KPIs using a consistent output pipeline.
- Provision of a calibrated evidence base for WP6 pilot design and large-scale impact estimation beyond the field trial.

Figure 16. Foresight scenario engine, from horizon scanning to stress-tested simulation and pilot preparation.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 16 depicts the architecture of the foresight scenario engine. Scenario levers cover traffic demand, network stress, UAV service design, behavioural adoption and control strategies. The workflow translates horizon-scanning outputs into parameterised scenarios, executes them through the SUMO-UAV environment, evaluates performance and robustness, and returns evidence for Nicosia pilot design and cross-pilot assessment.

4.2 STATE OF THE ART AND RESEARCH GAP

4.2.1 MICROSCOPIC AND INTERMODAL URBAN MOBILITY SIMULATION

Microscopic traffic simulation is widely used to evaluate traffic-management strategies before real-world deployment because it represents the movement and interaction of individual vehicles and travellers. SUMO is an open-source, microscopic and continuous simulation suite that supports road vehicles, public transport, pedestrians and bicycles, while TraCI enables external applications to observe and control the simulation online [19], [20]. Its person model represents trips as sequences of walking, riding, waiting and stopping stages, and its intermodal router can search combinations of walking, private vehicles and scheduled public transport [20], [23]. These capabilities provide an appropriate ground-transport foundation for FEDORA.

A number of research scenarios demonstrate the value of constructing reusable SUMO environments from open and observed data. The Monaco SUMO Traffic scenario incorporated public transport, pedestrians and three-dimensional urban information for cooperative-ITS studies [21]. The InTAS scenario represented Ingolstadt Road infrastructure, traffic lights and public transport and validated simulated flows against measurements from multiple counting locations [22]. Such projects show that credible simulation requires much more than a digital road map: demand, control plans, schedules, behavioural parameters and observed traffic conditions must be calibrated and validated.

4.2.2 UAV FLIGHT, SENSING AND URBAN-AIR-MOBILITY SIMULATION

UAV simulators address a different part of the problem. AirSim combines visual rendering, sensor emulation and a high-frequency physics engine for autonomous vehicle research [24]. RotorS provides modular multirotor dynamics, controllers and state estimation within Gazebo [25], while newer frameworks such as Pegasus provide multi-UAV simulation and integration with PX4 and ROS 2 [26]. These tools are valuable for flight control, perception, autonomy and hardware- or software-in-the-loop testing, but they are not designed to model city-scale road congestion, public-transport schedules, pedestrian trips or passenger-level mode chains.

At the transport-system level, urban-air-mobility studies analyse demand, vertiports, fleet operations, network effects and integration with existing modes. Reviews identify a fragmented modelling landscape in which many studies simplify either the ground network, passenger transfers or aircraft operations [27], [28]. Conversely, UAV traffic-monitoring research demonstrates the value of aerial cameras for vehicle detection, trajectory extraction and network observation [29], but this sensing role is rarely integrated within the same platform used to evaluate UAV transport services and ground traffic management.

4.2.3 RESEARCH GAP ADDRESSED BY FEDORA

The research gap is therefore not the absence of capable simulation tools; it is the absence of a single, calibrated and extensible environment that links their complementary domains. Ground-traffic simulators provide detailed multimodal network operations but do not natively represent a scheduled UAV passenger service. UAV physics simulators emulate aircraft dynamics and sensors but generally omit city-scale traveller demand and public-transport interaction. System-level Urban Air Mobility (UAM) models often represent access, waiting, capacity and continuation modes in an abstract manner. In addition, simulation accuracy remains conditional on the availability of real data for network, demand and behavioural calibration.

The FEDORA contribution addresses this gap at the service and transport-system level by integrating SUMO, TraCI-controlled UAV operations, passenger-level door-to-door evaluation, hub queues and capacity, configurable scenario generation, UAV sensing

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

coverage and automated KPI reporting. The current platform deliberately does not claim high-fidelity aerodynamics. Instead, it provides a modular operational layer that can later be coupled to a dedicated flight-physics simulator, while preserving the city-scale multimodal and foresight-analysis capabilities.

Table 6. Comparison of relevant simulation approaches and the gap addressed by the FEDORA platform.

Approach	Primary strength	Limitation for the FEDORA use case	FEDORA response
SUMO and validated SUMO scenarios	Microscopic road traffic, persons, buses, walking, cycling, intermodal routing and online TraCI control.	No native UAV passenger mode; accuracy depends on local network, demand and behaviour calibration.	Adds UAV service, hubs, queues, sensing and passenger-level UAV alternatives while retaining SUMO ground modes.
Agent-based demand platforms	Large-scale activity and demand assignment over complete populations.	Typically, less detailed for UAV operational events and microscopic hub interactions.	Uses scenario demand as input to a microscopic operational evaluation.
AirSim, RotorS, Pegasus and related UAV simulators	Flight dynamics, autonomy, control, sensor and software/hardware-in-the-loop testing.	Do not natively reproduce city-scale multimodal passenger traffic and public transport.	Provides the transport-system layer and defines a future coupling point for physical flight and sensor models.
System-level UAM models	Vertiport, fleet, demand and market or network analysis.	Ground access, congestion, transfers and sensing are often abstracted.	Executes access and continuation stages within the same SUMO network and records operational interactions.

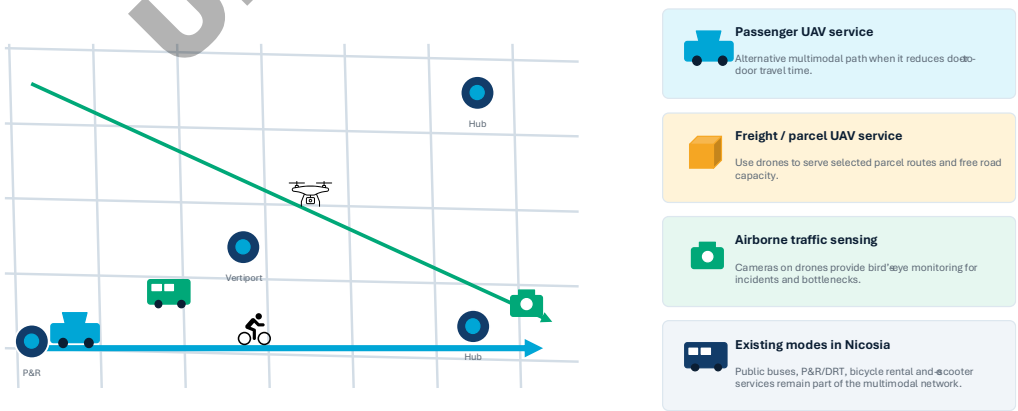

Figure 17. Nicosia pilot concept- UAV passenger and freight services and airborne sensing integrated with existing modes.

Table 6 positions the FEDORA foresight platform with respect to the principal simulation approaches relevant to Fedora framework. The comparison covers microscopic and intermodal traffic simulation, agent-based demand modelling, high-fidelity UAV simulation

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

and system-level urban air mobility analysis, highlighting the main capability contributed by each family of tools and the limitation that remains for the FEDORA use case. In this regard, Figure 17 summarizes the application scope that motivates the integrated platform. The UAV is treated as an additional service layer over buses, park-and-ride, demand-responsive transport, cycling and micromobility rather than as an isolated mode.

4.3 METHODOLOGY

The methodology starts from a parameterised scenario description. Each scenario will use the Nicosia's pilot calibrated network and will combine the demand state with a set of service and control assumptions. This separation is essential for repeatability as the same network can be evaluated under different demand levels, incidents, UAV fleet sizes, capacities, speeds, station layouts and traveller-acceptance assumptions without modifying the simulation code. This configuration stores these settings in a validated JSON configuration and creates a scenario-specific output directory.

Table 7. Principal foresight scenario levers and representative parameters.

Scenario dimension	Representative parameters	Purpose
Demand	Peak/off-peak volume, OD distribution, departure-time profile, special events and growth factors.	Stress the network and test the sensitivity of benefits to future mobility demand.
Network state	Incidents, road closures, bottlenecks, reduced capacity and control failures.	Assess robustness and identify conditions in which alternative modes become valuable.
UAV service	Fleet size, speed, capacity, path, schedule, hub/vertiport locations and service availability.	Evaluate operational design, waiting time, utilisation and spatial coverage.
Traveller behaviour	Eligibility, acceptance, transfer penalty and mode availability.	Represent adoption and the perceived cost of multimodal transfers.
Control policy	Road-only baseline, uncontrolled assignment, multimodal reservation and FEDORA optimisation settings.	Compare alternative traffic-management and service-allocation policies.

Table 7 defines the main dimensions used to generate foresight scenarios. Demand, network conditions, UAV service design, traveller behaviour and control policy can be modified independently or combined to construct consistent what-if experiments for the Nicosia pilot. Each dimension serves a different analytical purpose as follows:

- Demand and network-state variations stress the transport system,
- UAV-service parameters test operational design and capacity,
- Behavioural parameters represent adoption and transfer sensitivity;
- Control-policy settings enable comparison between road-only, uncontrolled and FEDORA-supported strategies.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

4.3.1 PLATFORM ARCHITECTURE AND EXECUTION WORKFLOW

The platform follows a five-stage workflow as follows:

1. Scenario inputs, define the network, demand, hubs, UAV fleet, UAV paths, disruptions and policy assumptions.
2. The configuration studio validates these inputs and constructs a reproducible run.
3. SUMO executes microscopic ground traffic and person events, while TraCI exposes the state needed by the UAV and multimodal manager.
4. The multimodal manager evaluates eligible passenger journeys, reserves capacity and controls pickup and drop-off events.
5. The dashboard presents the SUMO and UAV outputs are consolidated into KPI tables, figures and scenario comparisons.

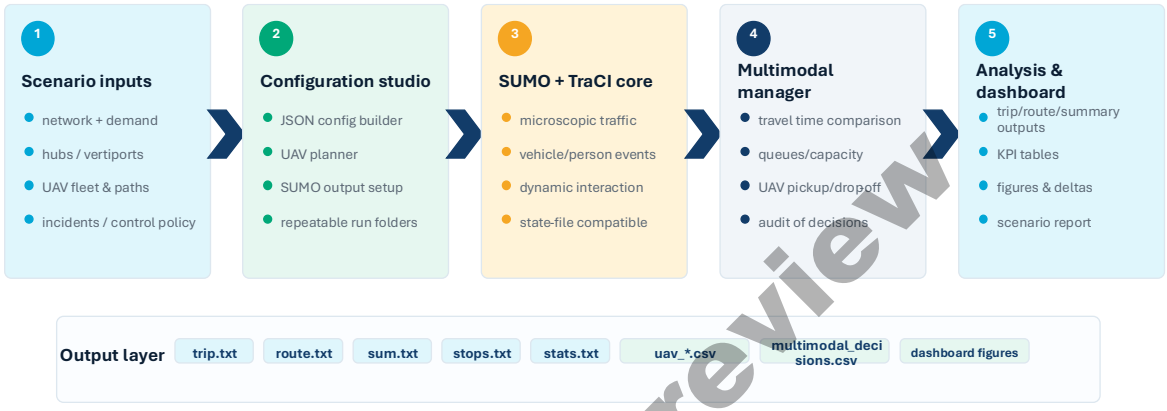

Figure 18. In-silico simulation architecture connecting scenario inputs, configuration, SUMO-TraCI, multimodal logic and analytics.

Figure 18 presents the implemented architecture from left to right. The modular separation between configuration, execution, decision logic and analytics allows the same platform to support repeated foresight studies and provides a clear future interface to the FEDORA orchestrator described below.

4.3.2 UAV INTEGRATION IN SIMULATION PLATFORM

The implemented coupling integrates SUMO's microscopic ground-mobility environment with a high-level UAV service layer. SUMO represents road vehicles, passenger movements, public transport, walking stages, network conditions and ground travel times, while the UAV layer provides hub locations, service schedules, flight times, capacity constraints and pickup/drop-off events. A TraCI-based coordination module exchanges the information required to evaluate and execute UAV-assisted journeys without modifying SUMO's native ground-traffic logic.

For each eligible passenger journey, the current high-level logic compares the estimated door-to-door travel time of the best available non-UAV journey with a candidate UAV-assisted chain. The UAV-assisted chain includes access to the departure hub, waiting for the next feasible UAV service, the scheduled aerial leg, and the best feasible continuation from the arrival hub by vehicle, public transport or walking. The UAV option is selected only when it is operationally feasible and reduces the passenger's total estimated travel time. In particular, the current decision rule is therefore to select the UAV-assisted journey when the sum of access time, waiting time, scheduled UAV flight time and final-continuation time is lower than the best non-UAV door-to-door alternative, subject to service availability and capacity. When the UAV option is selected, the passenger's access vehicle is rerouted to the departure hub. After the aerial transfer, a road-vehicle is used or when walking or public transport provides the shortest remaining journey, the passenger continues using that mode instead.

The decision module is intentionally replaceable. More advanced methods, including reservation-based assignment, predictive optimisation, network-level coordination, multi-objective optimisation or learning-based policies, can be connected through the same integration interface without changing the SUMO execution core. The UAV schedule, trajectory representation and operational

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

modelling used by the platform follow the specifications of WP4, analytically described in D4.1; the contribution reported here concerns their high-level integration with SUMO and their use in foresight scenario evaluation.

Table 8. Travel-time components used by the UAV-SUMO journey-selection approach.

Component	Estimation basis	Operational interpretation
Baseline journey	Best available non-UAV route from the passenger origin to destination.	Reference against which the UAV chain is assessed.
Access leg	SUMO road route and estimated travel time to the UAV departure hub.	The passenger arrives using the original vehicle, which is coloured red after acceptance.
Waiting time	Difference between predicted hub arrival and the next feasible scheduled UAV departure, including capacity.	Captures the main operational penalty of low frequency or insufficient fleet capacity.
UAV flight	Configured UAV path, timestamps, speed and service schedule.	Represents the aerial transfer between the two hubs.
Continuation leg	Minimum feasible vehicle, walking or public-transport continuation from the arrival hub.	A vehicle continuation is represented by a blue reinserted vehicle; walking or public transport does not require a blue vehicle.

Table 8 the passenger journey into the components used by the current UAV-SUMO decision logic. The non-UAV baseline is compared with a UAV-assisted chain consisting of the ground access leg, waiting time at the departure hub, the scheduled UAV flight and the best feasible continuation from the arrival hub.

Likewise, Figure 19 depicts the high-level coupling between the ground-mobility simulation and the UAV service layer. SUMO supplies the ground access and continuation alternatives, while the UAV module supplies the scheduled aerial service, capacity and hub-transfer information. The current implementation selects the UAV-assisted chain only when it improves the complete passenger journey; the same interface can subsequently host more advanced optimisation and coordination algorithms.

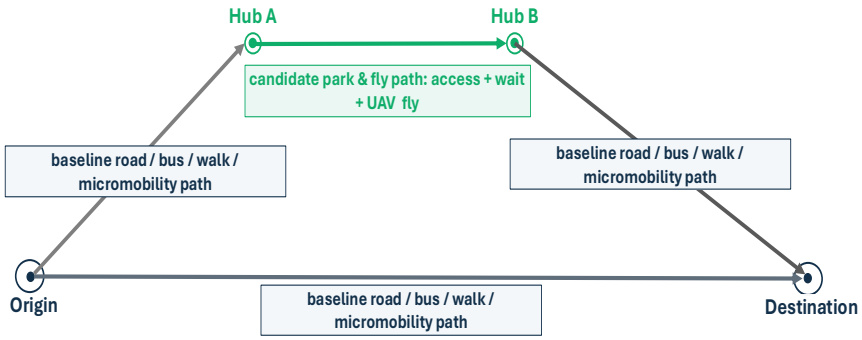

Figure 19. Passenger-level door-to-door decision rule for selecting a UAV-assisted journey.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

4.3.3 UAV SERVICE, HUB AND STATE-TRANSITION MODEL

The UAV path, service schedule and operational model implemented in the platform follow the UAV modelling and planning specifications developed in the relevant FEDORA WP4 technical work, analytically described in D4.1 Each service is represented by a time-stamped three-dimensional trajectory and the associated speed, altitude, passenger capacity, turnaround assumptions and repeat schedule. Hubs are mapped to both geographic coordinates and SUMO road edges, allowing the ground and aerial layers to exchange passengers through access routing, queues, reservations and pickup/drop-off events. The service lifecycle is therefore represented through the states planned, assigned, travelling to the departure hub, waiting and boarding, transported by the UAV, arrived at the destination hub, and completed through the selected continuation mode. The same UAV object may also include a field-of-view model that records observations over selected road segments and supports the traffic-sensing use case.

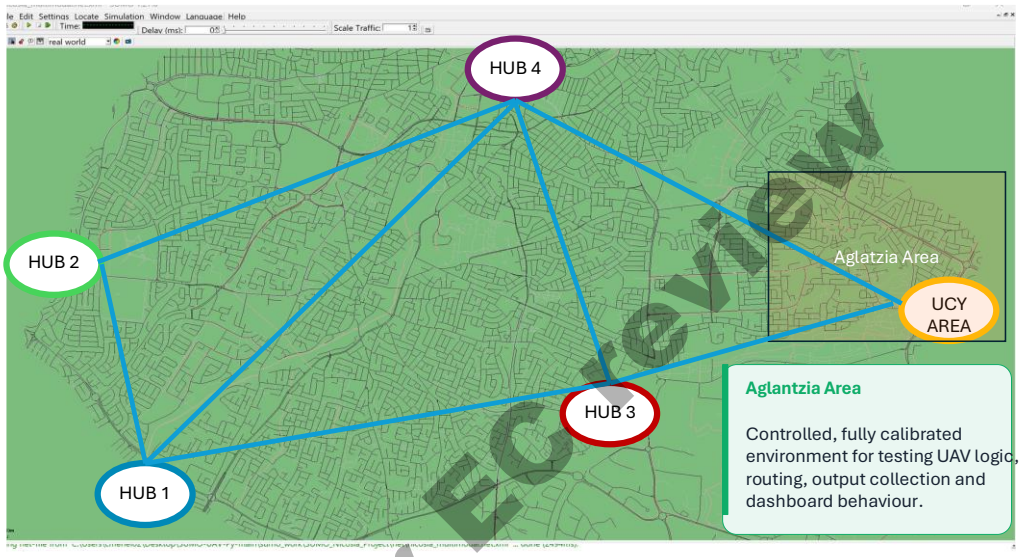

Figure 20. Greater Nicosia pilot area and candidate hub connections used for scenario development.

Figure 20 represents the target city-scale configuration for the final stage of the work. It depicts the Greater Nicosia area and the candidate hub and UAV connections that will be used to examine spatial coverage, ground-access time, vertiport placement and the potential demand served by UAV-assisted journeys. Before the final deliverable, this wider network will be calibrated using available real traffic, public-transport and mobility data, so that the foresight scenarios are evaluated on a representative city-scale baseline rather than on an uncalibrated network.

Figure 21 represents the current validation environment in Aglantzia, the smaller part of the Greater Nicosia area in which UCY/KIOS is located. At this stage, the purpose is not to claim city-wide results, but to test the complete FEDORA solution pipeline in a controlled network: scenario configuration, UAV scheduling, hub operations, passenger transfer, continuation by vehicle, walking or public transport, state and colour transitions, output generation and dashboard analysis. Once these functions are verified, the same workflow will be transferred, without changing its core interfaces, to the calibrated Greater Nicosia model for the final deliverable.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

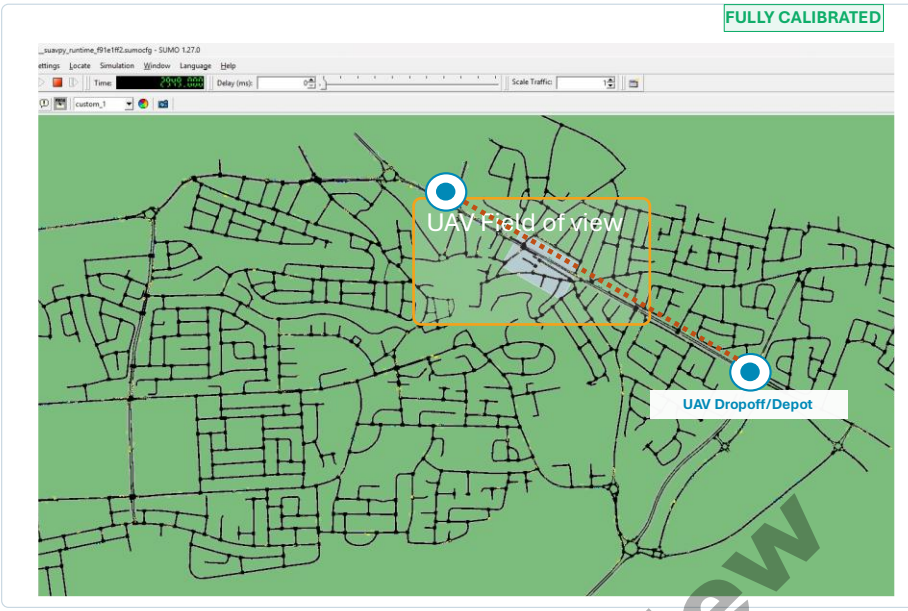

Figure 21. Aglantzia test-module validation environment used before Greater Nicosia scale-up.

4.3.4 CONFIGURATION, REPEATABILITY AND DATA MANAGEMENT

The configuration studio removes the need for manual editing of simulation code. User-entered values are written to the scenario configuration and are treated as the source of truth for the simulation. This is particularly important for UAV speed, capacity, path timing, simulation horizon, update intervals and output locations: values entered during configuration must not be silently replaced by hard-coded model defaults. The platform validates mandatory inputs, completes derived station mappings from the selected UAV path and network, and stores the resulting configuration with the run outputs for auditability.

State files are supported to initialise complex multimodal demand and network conditions. The state is loaded by SUMO, but the multimodal decision is not applied to every object at loading time. Instead, the manager listens only for true departure events, which ensures that each passenger vehicle is evaluated once at the beginning of its journey and avoids the substantial computational overhead of continuously inspecting all vehicles. Figure 22 consolidates the capability stack: scenario inputs, configuration builder, SUMO-TraCI execution, multimodal logic and analytics. Standard SUMO outputs are preserved alongside UAV dispatch, travel, sensing and multimodal-decision logs, allowing the same run to be analysed from both network and service perspectives.

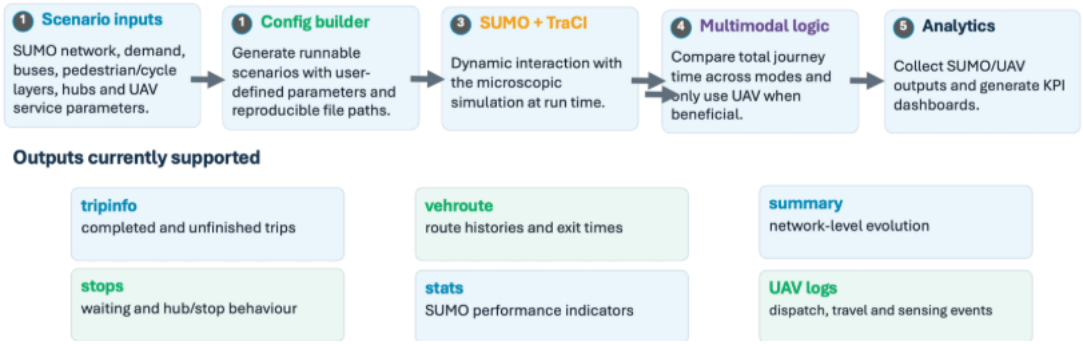

Figure 22. Implemented SUMO-UAV capability stack and supported output families.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

4.4 IMPLEMENTATION AND EVALUATION

4.4.1 MICROSCOPIC MULTIMODAL ENVIRONMENT

The current implementation is evaluated using a microscopic representation of the Aglantzia area in Nicosia, which serves as the controlled validation environment for the FEDORA foresight simulation workflow before its transfer to the Greater Nicosia pilot network. The represented network covers a bounding box of approximately 4.26 km × 2.30 km (approximately 9.8 km²) and contains 2,167 road edges, 6,711 road lanes, 929 non-internal junctions and 11 signalised junctions. In addition to private road traffic, the environment represents public transport, pedestrians, bicycles and shared mobility, enabling multimodal person journeys to be reproduced within the microscopic simulation. The current validation demand comprises 6,000 persons and 6,933 vehicle objects across private cars, buses, bicycles and shared-mobility services. Four hub locations are incorporated to support the testing of FEDORA multimodal and UAV-assisted scenarios. UAV-related operational parameters, including fleet size, capacity, trajectory, speed, altitude and service schedule, remain scenario-configurable rather than being fixed properties of the network, allowing different foresight assumptions to be evaluated using the same underlying mobility environment. A simulation step of 0.1 s is used to support fine-grained coordination between SUMO traffic evolution and the externally controlled FEDORA components.

Table 9 provides the principal scale and configuration characteristics of the environment used for the current implementation. The Aglantzia model is not intended to constitute the final spatial extent of the Nicosia foresight analysis. Instead, it provides a sufficiently detailed and multimodal microscopic test environment in which the complete FEDORA pipeline, including scenario configuration, multimodal interaction, UAV integration, output generation and KPI processing, can be verified under controlled conditions. Following this validation stage, the same methodological and software pipeline will be transferred to the Greater Nicosia network, where traffic demand and network conditions will be calibrated against real data for the final network-wide foresight experiments and pilot preparation. The current implementation represents road vehicles, buses, cyclists, pedestrians, person-in-vehicle stages, bus loading and signalised crossings within SUMO. These native modes remain under SUMO control; they are not reimplemented in the UAV manager. The manager is invoked only for eligible passenger-vehicle departure events and only when the UAV service is enabled. This preserves the validated ground-traffic logic and limits additional computation to vehicles for which an alternative is relevant.

Table 9. Main characteristics s of the Aglatzai microscopic validation enviroment

Characteristic	Current configuration
Study area	Aglantzia, Nicosia, Cyprus
Approximate area	9.8 km²
Road edges	2,167
Road lanes	6,711
Non-internal junctions	929
Signalised junctions	11
Ground-mobility representation	Private vehicles, buses/public transport, pedestrians, bicycles and shared mobility
Demand in current validation dataset	6,000 persons; 6,933 vehicle objects across the represented modes

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

UAV configuration	Scenario-dependent fleet size, capacity, speed, altitude, trajectory and repeat/service schedule
Multimodal functionality	Road, public transport, walking, cycling/shared mobility and UAV-assisted journey evaluation
Simulation step	0.1 s

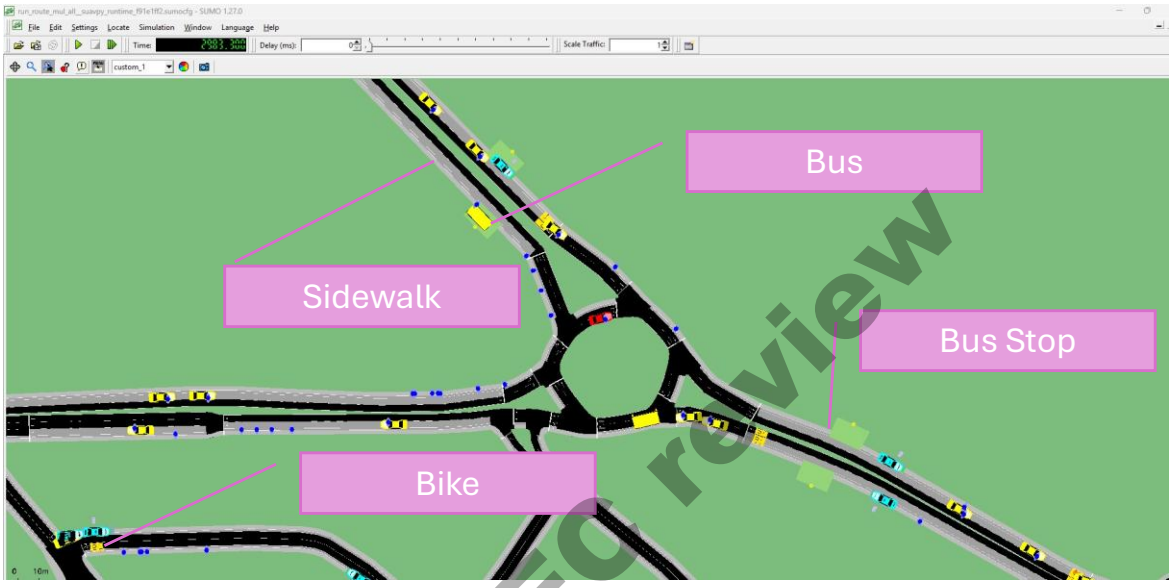

Figure 23. Aglantzia SUMO environment with bus, sidewalk, bicycle and bus-stop infrastructure.

Figure 23 indicates that the validation environment includes the principal ground-mode infrastructure required for multimodal analysis. Road vehicles, buses, bicycles and pedestrians interact on the same network, with bus stops and sidewalks providing the physical transfer points used by person stages.

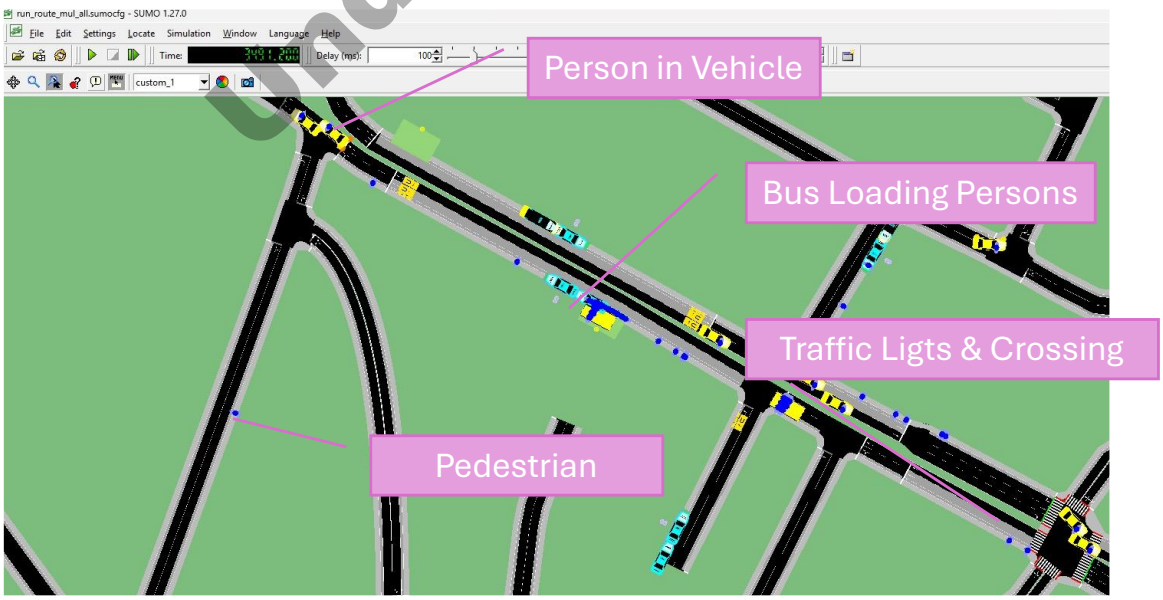

Figure 24. SUMO representation of persons in vehicles, bus boarding, pedestrians and signalised crossings.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 24 illustrates the person-level elements of the simulation. Passenger association with vehicles and public transport is necessary because the objective is total person travel time rather than vehicle travel time alone. Signalised crossings and walking stages affect access and continuation times and therefore influence whether a UAV-assisted chain is advantageous.

4.4.2 UAV MOTION AND AIRBORNE SENSING

UAV motion is displayed using the configured trajectory, altitude, timing and heading. The operational service model uses the same schedule for passenger pickup and drop-off calculations. In parallel, the UAV field of view can be projected over the road network to identify observation periods for specific road segments. This provides a second contribution of the platform: the same UAV resource can be evaluated as a transport service and as a mobile traffic sensor.

At the present stage, sensing is represented geometrically through coverage and observation windows. Camera optics, image quality, weather, occlusion, vibration and perception errors are not yet physically emulated. The platform therefore reports where and when sensing is possible, while future coupling to a robotics simulator will be required for sensor-level fidelity.

Figure 25. UAV trajectory and field-of-view representation over the Aglantzia road network.

Figure 25 shows the UAV and its field-of-view footprint. The geometric representation enables the platform to calculate coverage intervals and associate aerial observations with road segments, supporting foresight scenarios in which traffic monitoring capacity is varied together with transport-service demand.

Figure 26. Passenger redirection to the UAV service and passenger drop-off at the arrival hub.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 26 visualises the passenger transition between ground and air services. The redirection and drop-off markers support verification that accepted passengers follow the intended sequence and that the final continuation is activated only after the scheduled UAV arrival.

4.5 SCENARIO EXECUTION AND KPI FRAMEWORK

The scenario suite executes baseline, controlled, uncontrolled and UAV-assisted cases with consistent file structures. The output pipeline combines standard SUMO files with UAV and multimodal logs. Expected values computed at the decision event are kept separate from realised values obtained from simulation, which allows model error, queue effects and downstream congestion to be identified.

Table 10. Output files and principal KPI families supported by the platform.

Output/KPI family	Examples	Evaluation purpose
SUMO trip and route outputs	Tripinfo, completed/unfinished trips, route history, exit time and travelled distance.	Measure traveller and vehicle performance and validate route execution.
Network evolution	Summary, speed, time loss, congestion, bottlenecks and edge-level delay.	Assess network-wide impacts and identify stress locations.
Stops and person stages	Waiting, boarding, walking, ride stages and transfer durations.	Evaluate complete door-to-door multimodal journeys.
Multimodal decisions	Accepted/rejected alternatives, reason, expected component times, chosen continuation mode and expected savings.	Audit the passenger-level decision and identify why UAV service is or is not selected.
UAV service	Dispatch, flight, pickup/drop-off, queue, capacity, utilisation and sensing windows.	Assess operational feasibility and service design.
Scenario comparison	Baseline versus controlled deltas, expected versus realised performance and KPI dashboards.	Support foresight analysis and pilot decision-making.

Table 10 summarises the output files and KPI families used to evaluate a simulation run. The platform combines standard SUMO trip, route, network, stop and person-stage outputs with multimodal decision logs, UAV operational records and scenario-comparison dashboards.

The output structure supports analysis at three complementary levels:

1. Traveller-level outputs verify complete door-to-door journeys and transfer stages,
2. network-level indicators quantify congestion, speed, time loss and bottlenecks,
3. service-level logs explain UAV availability, queueing, capacity utilisation and the reasons why alternatives are accepted or rejected.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

Keeping expected decision values separate from realised simulation outcomes also enables validation of the estimation logic and identification of downstream congestion or queue effects.

Figure 27 summarises the T5.6 analysis package. Trends are converted into parameterised experiments, multiple runs are executed to expose operational limits, and the resulting KPI package supports demonstration design and interpretation of FEDORA value.

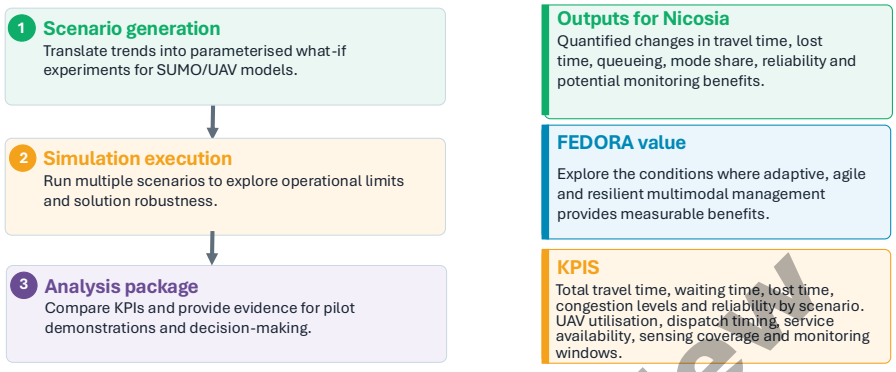

Figure 27. Scenario-generation, execution and analysis package for T5.6 foresight evaluation.

4.5.1 GRAPHICAL INTERFACES AND REPEATABLE STUDY WORKFLOW

A graphical configuration builder, UAV planner, scenario suite and dashboard have been developed to make the platform usable without modifying Python source code. Each interface writes the parameters required by the corresponding logic, creates scenario-specific folders and exposes the resulting outputs. The interfaces support rapid testing during development and will also provide a repeatable workflow for WP5/WP6 evaluation activities.

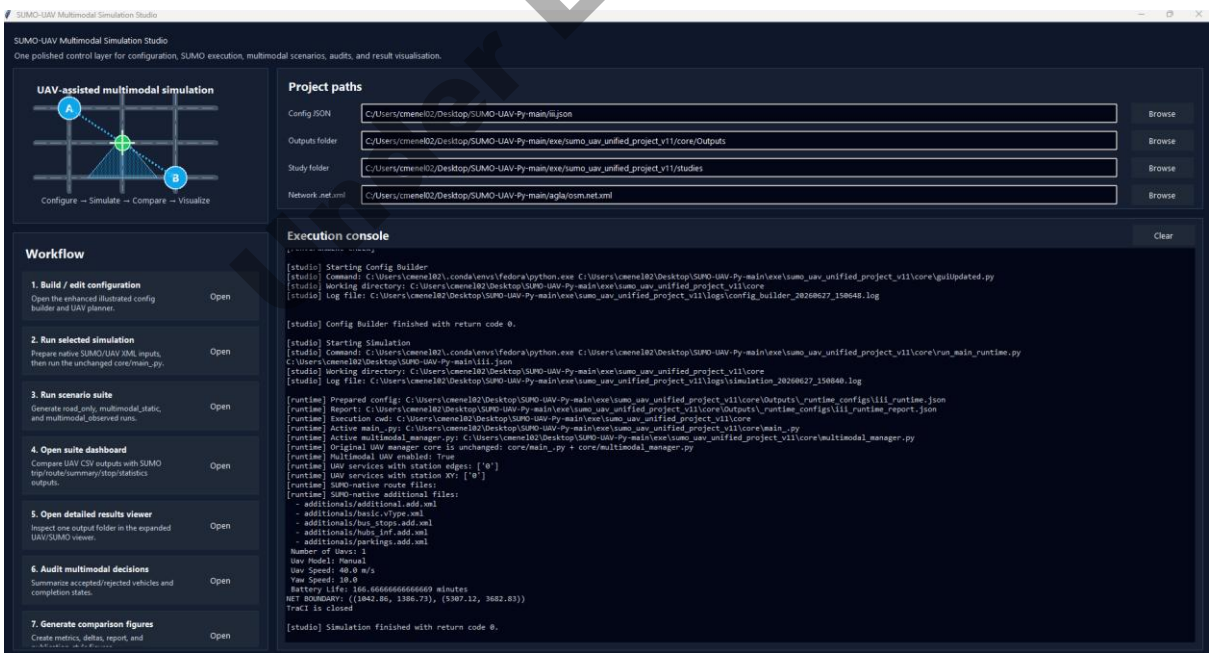

Figure 28. Configuration builder for reproducible SUMO-UAV scenario definition.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 28 shows the configuration builder used to define the network, simulation, UAV and output settings. The interface validates user input and stores the exact values used by the algorithms, preventing inconsistencies between the graphical configuration and hard-coded defaults.

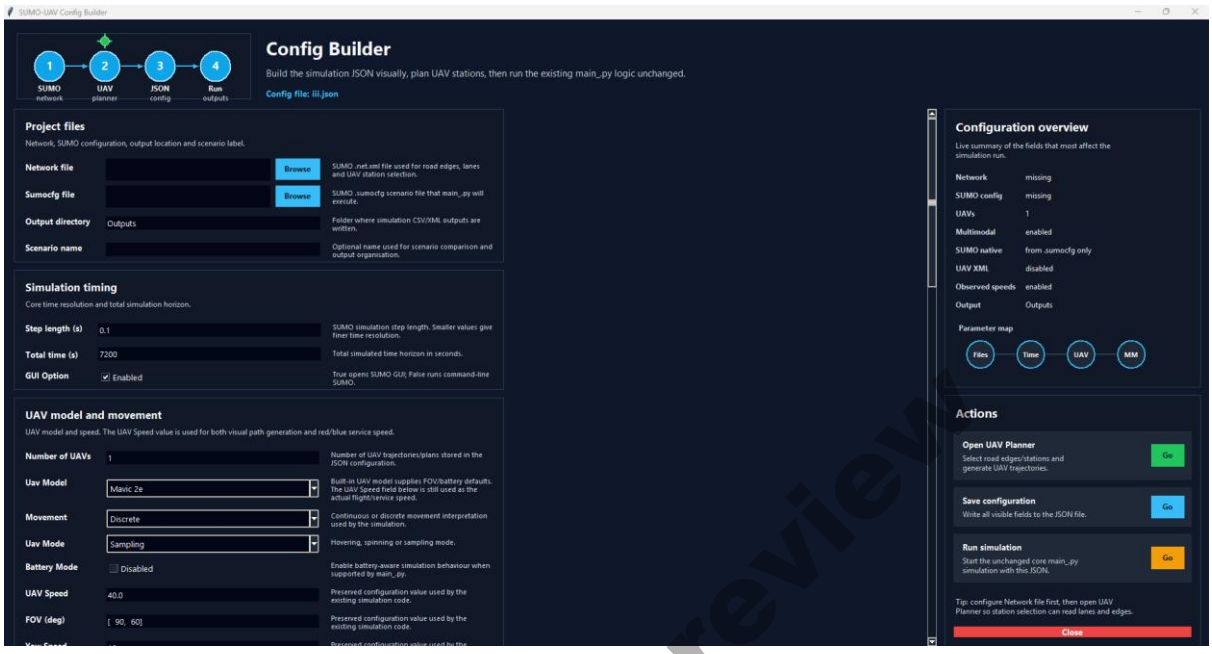

Figure 29. UAV planner interface for path, speed, timing, altitude and operational settings.

Figure 29 presents the UAV planning interface. User-defined path and service parameters are converted into the time-stamped trajectory and station information required by both the visual UAV movement and the passenger-service calculations.

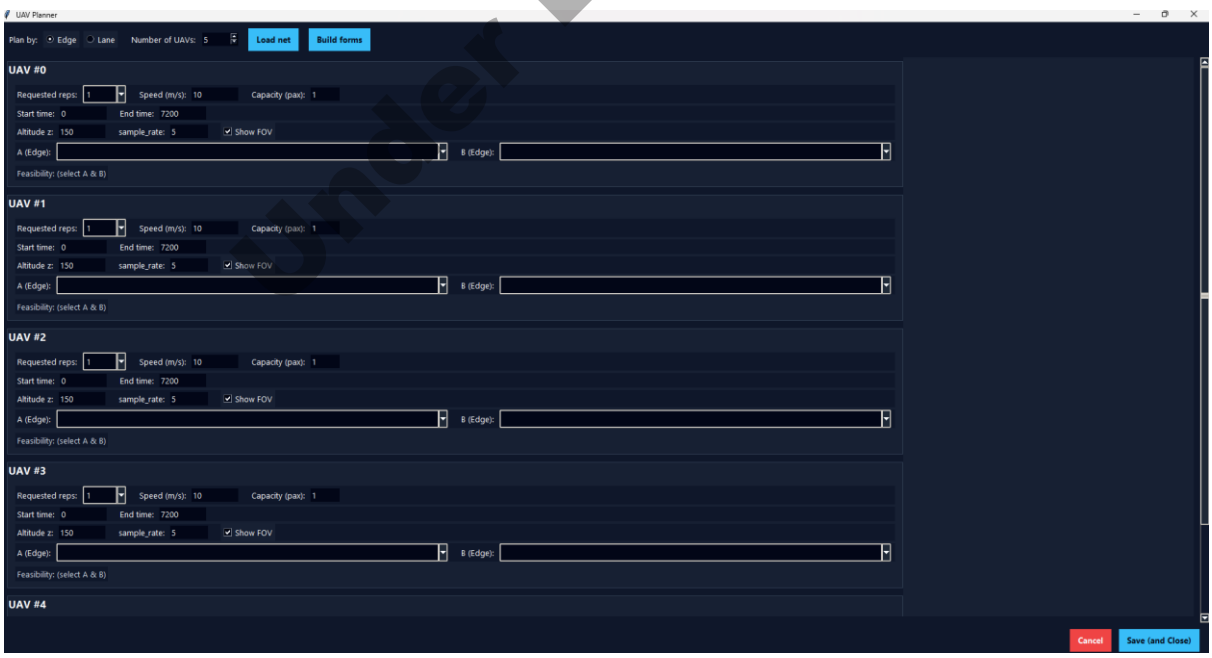

Figure 30. Scenario study-suite interface for repeated execution and consistent storage of results.

Figure 30 shows the study-suite interface used to organise repeated cases. Consistent run folders and metadata are essential for comparing scenario variants and for tracing each KPI result back to the corresponding demand, network and service settings.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 31. Dashboard view of UAV observation periods over a selected road segment.

Figure 31 illustrates the sensing-analysis output. The dashboard records the temporal windows during which the UAV field of view covers a selected road segment. This output can be combined with congestion and incident scenarios to evaluate whether the UAV is available at the location and time at which monitoring is most valuable.

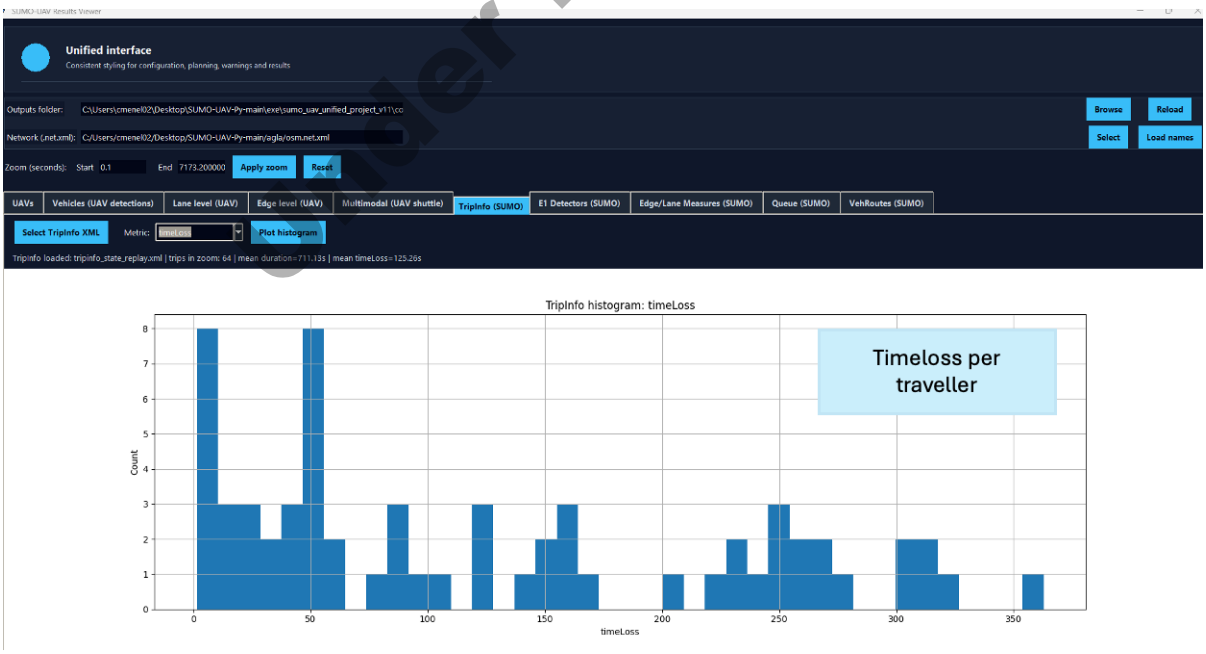

Figure 32. Traveler time-loss visualization generated from simulation outputs.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

Figure 32 presents traveller-level time-loss results. Distributional indicators are important because an improvement in network average performance may conceal substantial losses for specific passenger groups, access locations or transfer chains.

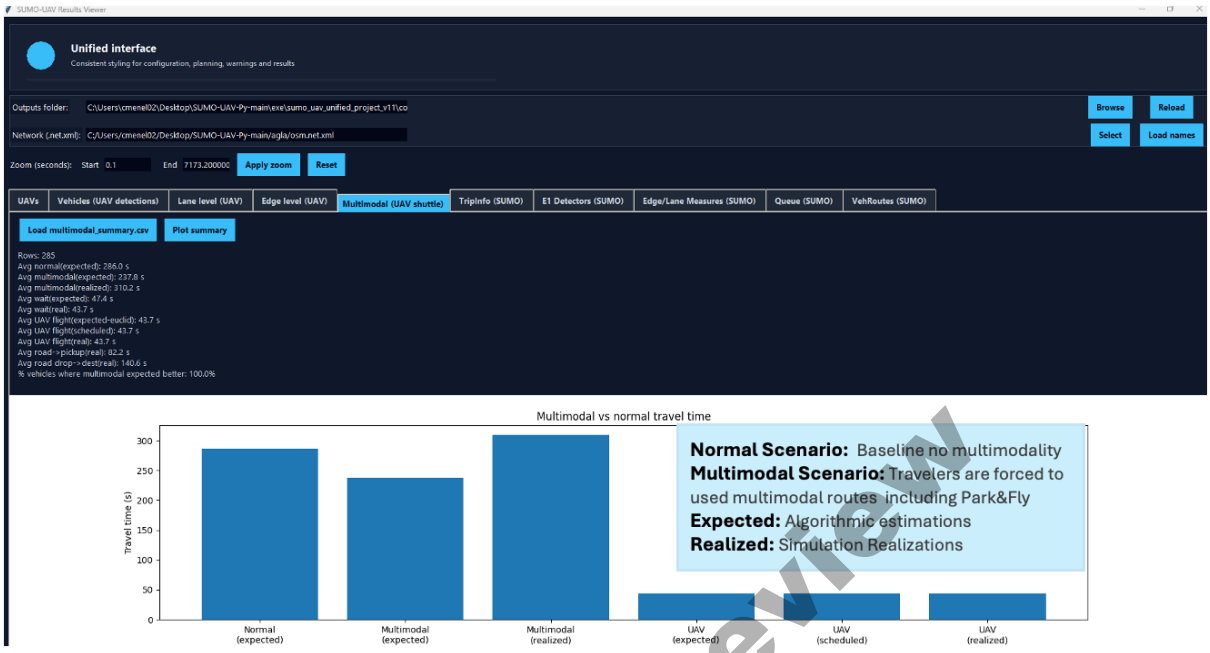

Figure 33. Baseline and multimodal scenario comparison, including expected and realised performance.

Figure 33 distinguishes the normal road-only baseline from a multimodal scenario and separates algorithmic estimates from realised simulation outcomes. This distinction is used to verify whether predicted passenger savings remain after actual waiting, congestion and route interactions are simulated.

4.5.2 VALIDATION RESULTS

This section presents the initial validation results obtained from the Aglantzia simulation environment, focusing on the comparison of the evaluated scenarios under consistent simulation conditions. These results are used primarily to verify the correct operation of the implemented FEDORA simulation and evaluation pipeline and to demonstrate its ability to quantify the effects of alternative multimodal strategies. At this stage, the results should therefore be interpreted as validation of the developed methodology and software workflow rather than as the final assessment of the Nicosia pilot, which will be conducted after calibration and deployment of the framework on the Greater Nicosia network.

In Figures below for validation purposes, we compared the UAV multimodal paths with the normal uncontrolled scenario over the test area of the Aglatzia. For this scenario a control mechanism of route reservation is performed aiming to maximize the utilization of multimodal efficiency and minimized the passenger travel time. The solution will be presented in the next IEEE ITSC 2026 conference (as accepted paper). Note that the numerical values shown in each figure correspond to the current illustrative Aglantzia runs and are therefore interpreted as prototype evidence rather than as final calibrated Nicosia results.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

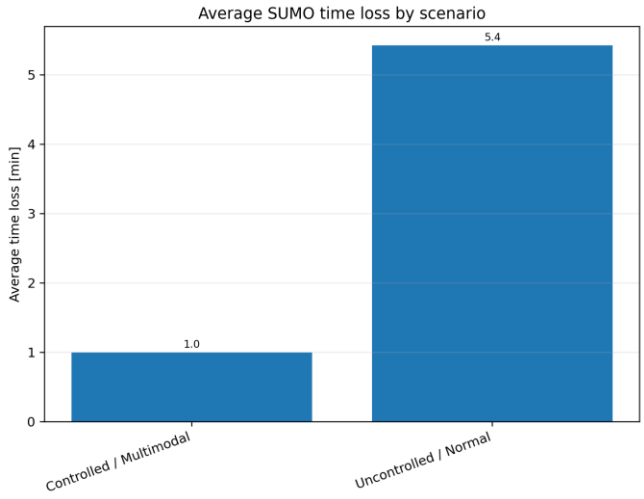

Figure 34. Average SUMO time loss in the controlled/multimodal and uncontrolled/normal scenarios.

Figure 34 compares the average SUMO time loss across the two scenario configurations. In the illustrated run, the controlled/multimodal case records an average time loss of 1.0 min, compared with 5.4 min in the uncontrolled/normal case. Time loss measures the additional delay experienced relative to unconstrained movement and therefore provides a compact indication of congestion and control effectiveness. The lower value in the controlled case indicates that the multimodal reservation logic limited the accumulation of network delay in this experiment. This result will be reassessed through repeated and calibrated runs before it is used for pilot-level impact claims.

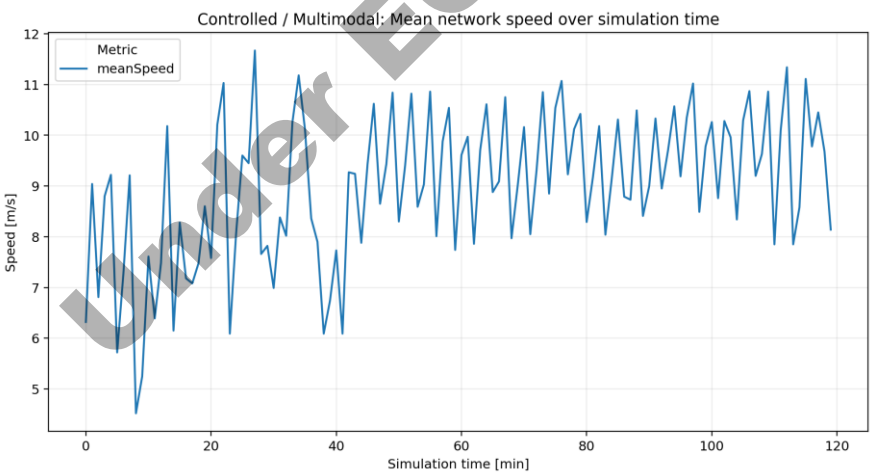

Figure 35. Mean network speed over time in the controlled/multimodal scenario.

Figure 35 presents the temporal evolution of mean network speed under the controlled/multimodal strategy. After the initial loading period, the network speed generally remains within a stable operating range, with short-term oscillations reflecting variations in demand, traffic-signal conditions and route interactions. Most importantly, the time series does not exhibit a persistent downward collapse. This behaviour indicates that, for the illustrated scenario, the controlled strategy preserves network flow despite recurring local disturbances.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

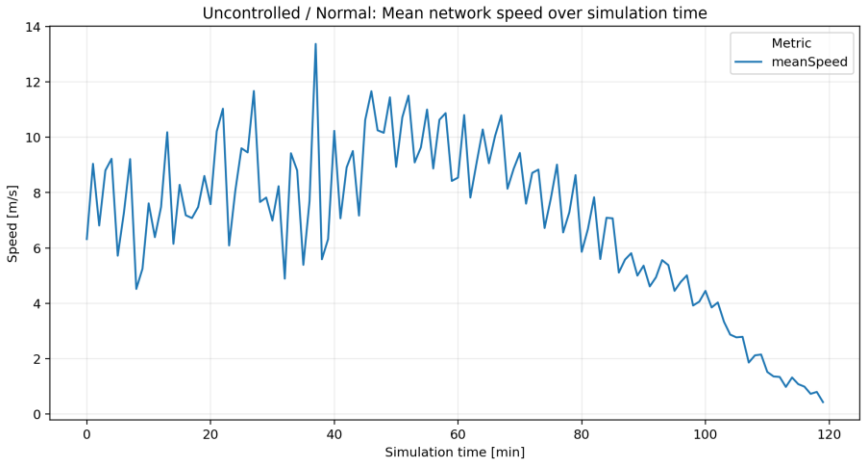

Figure 36. Mean network speed over time in the uncontrolled/normal scenario.

Figure 36 shows the corresponding mean-speed profile for the uncontrolled/normal scenario. The network initially operates at speeds comparable to the controlled case; however, after approximately the middle of the simulation, the mean speed progressively deteriorates and approaches very low values near the end of the run. The profile is consistent with the accumulation of severe congestion and eventual network breakdown.

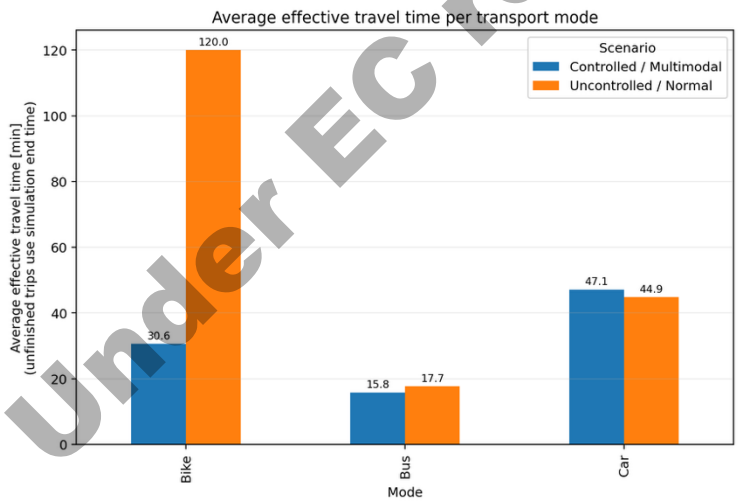

Figure 37. Average effective travel time by transport mode on each scenario.

Figure 37 disaggregates effective travel time by bicycle, bus and car. In the illustrated controlled/multimodal run, the bicycle and bus values are lower than in the uncontrolled/normal run, while the car value is slightly higher. The comparison therefore shows that the effects of a network-management intervention are not necessarily uniform across modes. The effective-travel-time indicator also accounts for unfinished trips by using the simulation end time, and it must consequently be interpreted together with completion rates. This mode-level view is essential for identifying whether an intervention improves overall performance by transferring delay from one user group to another.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

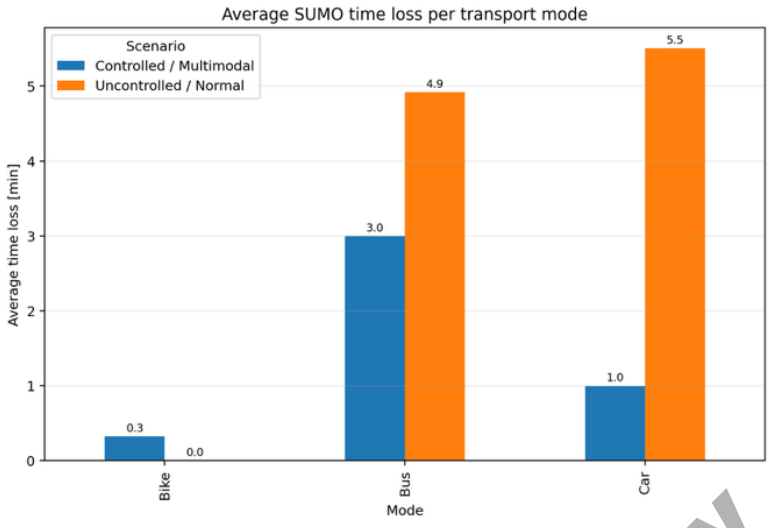

Figure 38. Average time loss by transport mode on each scenario.

Figure 38 reports the average time loss separately for bicycle, bus and car trips. The controlled/multimodal scenario substantially reduces bus and car time loss in the illustrated run, whereas the bicycle category shows a small increase from 0.0 to 0.3 min. Note that prior the enforcement of the control strategy none of the travellers was preferring to utilize bikes while with application of the controlled strategy some travellers are instructed to use bike in their journey as benefit their overall travel time. A network-wide improvement should not be interpreted as universally beneficial unless the impacts on each mode and traveller group have also been examined.

4.5.3 CURRENT EVALUATION STATUS AND INTERPRETATION

The current Aglantzia implementation verifies the complete technical workflow: configuration, state-file and route-file initialisation, entrance-only passenger evaluation, UAV reservation, red access vehicle, UAV transfer, output generation and dashboard analysis. It can therefore be used to test algorithms and scenario mechanisms. However, numerical outputs should currently be interpreted as prototype scenario estimates rather than validated predictions for Greater Nicosia, because the full pilot area and demand model are still under calibration.

A result in which few or no passengers select the UAV is not automatically a software failure. The passenger-level decision may correctly reject the service when road access to the hub, waiting time, capacity or continuation time exceeds the benefit of the aerial leg. The decision logs are designed to expose each component and rejection reason. Conversely, accepted alternatives must be checked against realised simulation travel time to ensure that the expected saving was achieved.

4.6 INTEGRATION INTO THE FEDORA MODULAR SIMULATION PLATFORM

Within the three-layer architecture introduced in Chapter 2, SUMO and the UAV execution environment belong to the application layer, while the passenger multimodal manager, scenario generator and FEDORA optimisation functions are logic components. The interoperability layer is responsible for configuration, lifecycle coordination, state exchange and output collection. At a simulation departure event, the environment publishes the eligible passenger state; the multimodal logic returns the selected path and reservation; the environment applies the vehicle and person transitions; and the recorder stores decisions and KPIs.

Funded by the European Union
Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The platform can absorb network and demand scenarios from T5.1, candidate hub and service designs, and optimisation parameters from WP4. It can provide network state, passenger alternatives, queue and capacity information, UAV availability, sensing coverage and realised KPIs. This modular framing permits the present SUMO-based environment to be replaced or complemented by another simulator without rewriting the decision logic, and it establishes a future coupling point to real pilot data and physical UAV emulation.

4.7 NEXT STEPS

4.7.1 CALIBRATION AND VALIDATION WITH REAL DATA

The highest-priority next step is the calibration and validation of the Aglantzia and Greater Nicosia models using real data. Network geometry, lane configuration, speed limits, signal plans and public-transport stops will be checked against authoritative sources. Traffic demand will be calibrated using available vehicle counts, OD information, travel-time observations and departure profiles. Bus routes, timetables and passenger demand will be aligned with observed or operational data, while pedestrian and cycling demand will be refined where measurements are available. Calibration will be followed by validation against independent datasets and explicit reporting of error measures rather than visual agreement alone.

UAV service parameters will likewise be grounded in real or pilot-specific evidence. Planned flight logs will be used to verify speed, travel time, hover and turnaround time. Hub processing and transfer assumptions will be measured or specified with the pilot team. Passenger acceptance and transfer penalties will be informed by surveys and live-pilot feedback. This calibration is necessary before scenario results are interpreted as expected pilot impacts.

4.7.2 GREATER NICOSIA SCALE-UP AND SCENARIO CATALOGUE

After validation in Aglantzia, the same workflow will be transferred to the Greater Nicosia final pilot network. The scenario catalogue will include demand growth, peak and off-peak conditions, incidents, road closures, special events, alternative hub layouts, UAV fleet sizes, capacity and frequency settings, passenger-adoption levels and FEDORA control policies. Repeated experiments will quantify not only average benefits but also robustness, queue formation, reliability and the spatial distribution of impacts.

4.7.3 FEDORA INTEGRATION AND PILOT EVIDENCE PACKAGE

The platform interfaces will be aligned with the FEDORA orchestration contract so that scenario inputs, network state, optimisation decisions and KPIs can be exchanged with other components. The final WP5/WP6 package will include versioned scenarios, calibrated inputs, reproducible run configurations, decision audits, KPI definitions, expected-versus-realised comparisons and a concise evidence set for live demonstration planning. Computational profiling and parallel scenario execution will also be addressed to support the Greater Nicosia scale.

4.7.4 EXTENSION TO THE REMAINING FEDORA PILOTS

Beyond the Nicosia Pilot (Pilot #3), the simulation framework will be further refined, where applicable, to support the evaluation needs of the remaining FEDORA pilots associated with EI4.6, subject to their final scope and implementation. This refinement will extend the simulator to capture pilot-specific transport networks, mobility services, operational constraints, and evaluation scenarios while preserving a common modelling and KPI assessment methodology. The objective is to provide a harmonised simulation environment that enables consistent assessment of pilot-specific interventions, facilitates cross-pilot comparisons, and supports the evaluation of the overall FEDORA framework under diverse operational conditions.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

4.8 CONCLUSION

Task 5.6 has established an operational in-silico platform for generating and evaluating foresight scenarios that combine microscopic urban traffic, native SUMO intermodality, UAV passenger services, hub operations, UAV sensing and automated KPI analysis. The principal methodological contribution is a passenger-level door-to-door comparison in which an UAV alternative is selected only when access, waiting, flight and continuation jointly improve the journey and capacity is feasible. The entrance-only evaluation event preserves computational efficiency and supports both loaded state files and conventional route files without changing the core red/blue vehicle logic.

The platform fills an important integration gap between ground-mobility simulation and UAV operational modelling, while remaining transparent about its present limits. Its current role is to validate the complete scenario and evaluation workflow in Aglantzia. The next phase will calibrate the model using real data, scale it to Greater Nicosia, improve physical UAV and sensor fidelity, calibrate passenger behaviour, align the interfaces with the FEDORA orchestrator and prepare a validated KPI package for the live pilot. The simulation framework will also be further refined, where applicable and subject to the final scope of the respective activities, to support the evaluation needs of the remaining FEDORA pilots associated with EI4.6.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

5 RESEARCH AND SCIENTIFIC INNOVATION

Chapters 2 to 4 presented the three tasks reported in this deliverable: the model-agnostic and modular simulation platform of Task 5.4, the model-to-application transferability (Sim2Real) techniques of Task 5.5, and the foresight analysis of Task 5.6. This chapter reflects on the research and scientific innovation that this work represents. It first draws the three tasks together, setting out how they relate and can be integrated within the modular platform (Section 5.1), before relating the individual contributions to the innovations and results the project set out to achieve (Section 5.2), identifying where they advance the state of the art (Section 5.3), and discussing their readiness, exploitation and dissemination (Section 5.4).

5.1 CROSS-TASK SYNTHESIS AND INTEGRATION OUTLOOK

The three tasks are distinct contributions that are nonetheless designed to operate within the common structure developed in Task 5.4 and presented in Chapter 2. The platform's role reaches well beyond the tasks reported here, as it is intended as an interoperability framework for the technical components developed across WP4 and WP5, bringing the traffic and network optimisation and control services on one side together with the simulation environments and real-world interfaces on the other, so that they can be combined and assessed within a single setting. The three-layer architecture of Figure 1, with the orchestrator, interchangeable logic modules and application layer environments, is the consolidated view within which these components are placed. In this deliverable the framework is realised in the form of a concrete demonstrator for traffic signal control (Chapter 2), which connects a microscopic simulation environment and several controllers through standardized interfaces and message-passing with TCP. The demonstrator establishes that the mechanism can be applied in an end-to-end setup, and that as every module can only be contacted and controlled through the defined interfaces, the remaining components integrate in the same manner rather than requiring the platform to be re-engineered for each. The WP4 network optimisation services and WP5 simulators and controllers would therefore simply be plugged in with minimal integration effort as additional logic modules and environments, respectively.

The work reported in this deliverable represents an intermediate stage, and its continuation aims at extending the integrated platform developed in Task 5.4 and finalizing the developed methodologies in Tasks 5.5. and 5.6, the results of which are to be reported in deliverable D5.3. The demonstrator is planned to be extended beyond signal control to accommodate further optimisation and control services developed across WP4 and WP5, which may then be evaluated within the platform. Depending on the needs of the component deployments in the real-world pilot context of WP6, the application layer environment interface, which already anticipates connections beyond simulation, may be extended towards interfaces to pilot sites in the second round of demonstrations. In parallel, the technical integration efforts between the components of WP4 and WP5 discussed here will continue to be aligned with the project's wider platform through the integration activities in WP2, aiming at an operation alongside the data and service spaces rather than as a standalone tool. Together these steps cover the individual task contributions towards the integrated and interoperable framework that the final deliverable will report.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

5.2 CONTRIBUTION TO THE FEDORA EXPECTED INNOVATIONS AND RESULTS

The conducted work is organised around a set of expected innovations and the results they feed. The three tasks reported in this deliverable each correspond to one such innovation: Task 5.4 to the agent-based interoperability framework (EI4.4), Task 5.5 to the model-to-application transferability techniques (EI4.5), and Task 5.6 to the foresight analysis scenarios generator (EI4.6). Together with the multimodal simulation engine and the demand and controller models developed in the earlier tasks of the work package and reported in Deliverable D5.1 (EI4.1 to EI4.3), they contribute to two expected results, namely the integrated and interoperable simulation framework (ER4.1) and the foresight analysis with model-to-application transferability frameworks (ER4.3). The contributions of the three tasks are set out separately in the following subsections.

5.2.1 TASK 5.4: AGENT-BASED INTEROPERABILITY FRAMEWORK (EI4.4)

Task 5.4 delivers Expected Innovation EI4.4, the agent-based interoperability framework for integrating decision-making modules with simulation environments and for assessing them across a range of configurations. This innovation is the model-agnostic and modular platform presented in Chapter 2, whose orchestrator mediates between interchangeable logic modules and application layer environments through a standardized communication scheme, demonstrated end-to-end at the example of traffic signal control with different methodologies. Its main contribution lies in the interoperability layer itself, which allows to connect heterogeneous methodologies and environments through common interfaces, while also guaranteeing exchangeability without additional integration work. The framework thereby provides the interoperable element of the integrated simulation framework (ER4.1), complementing the framework of simulation engines of EI4.1 so that the compatible technical components developed across the project can be connected on a common basis.

Since the demonstrator is already realised as a reusable open-source platform rather than a study tied to one application, the framework forms a natural basis for potential future exploitation and extensions towards the inclusion of methods developed in the other tasks and work packages as they mature. Its readiness, licensing and exploitation are discussed in Section 5.4.

5.2.2 TASK 5.5: MODEL-TO-APPLICATION TRANSFERABILITY (EI4.5)

Task 5.5 contributes Expected Innovation EI4.5, the model-to-application transferability (Sim2Real) techniques for transferring control methods from simulation towards real operation, contributing to the Expected Result on foresight analysis with model-to-application transferability frameworks (ER4.3). Its contribution is described in Chapter 3. The work presented demonstrates how RL-based traffic management policies can be adapted to operate beyond the simulation environment in which they were originally trained, addressing one of the main barriers to the practical deployment of learning-based traffic management systems. As a proof of concept, the proposed methodology was evaluated on the network-level perimeter control problem, where a RL controller trained in one microscopic traffic simulator was successfully transferred to another simulator with different traffic dynamics using a GAT framework. The experimental results showed that Sim2Real techniques can substantially reduce the performance degradation associated with direct policy transfer while requiring only a limited amount of information from the target environment. Beyond the proof-of-concept case study, future work within Task 5.5 will extend the proposed framework to larger and more realistic traffic networks and investigate the transfer of traffic management policies across different simulation fidelities. These approaches will strengthen the contribution of Task 5.5 to Expected Innovation EI4.5 and Expected Result ER4.3 by advancing the model-to-application transferability of intelligent traffic management solutions.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

5.2.3 TASK 5.6: FORESIGHT ANALYSIS SCENARIOS GENERATION (EI4.6)

Task 5.6 contributes to Expected Innovation EI4.6 by establishing a scenario-generation and evaluation capability for network-wide foresight analysis. The work combines parameterised scenario design, multimodal traffic simulation, UAV–SUMO integration and systematic KPI assessment to examine how changes in demand, network conditions, service configurations and control strategies affect multimodal transport performance. The resulting framework supports the comparison of road-only, multimodal and UAV-assisted alternatives and provides evidence on the conditions under which emerging mobility services can improve passenger journeys, network efficiency and operational resilience. By enabling the testing of future mobility configurations before their deployment in the pilot environment, Task 5.6 directly supports Expected Result ER4.3 on foresight analysis with model-to-application transferability frameworks. The technical methodology, simulation environments and evaluation workflow are described in Chapter 4

5.3 ADVANCES BEYOND THE STATE OF THE ART

The work carried out in Task 5.5 advances the state of the art in the application of Sim2Real reinforcement learning to intelligent traffic management. While existing reinforcement learning approaches for perimeter control generally assume that training and deployment take place in the same environment, the proposed framework explicitly addresses the transfer of learned policies across environments with different traffic dynamics. To the best of our knowledge, this is the first study to apply GAT to network-level perimeter control. In addition, the study provides one of the first systematic evaluations of Sim2Real reinforcement learning for perimeter control by analysing the impact of target-domain data availability, iterative grounding, and action transformation on transfer performance.

The work carried out in Task 5.6 advances the state of the art in foresight-oriented multimodal transport simulation by integrating microscopic ground-mobility modelling, UAV-enabled transport, airborne traffic sensing and scenario-based evaluation within a common simulation workflow. Existing urban traffic simulators provide mature representations of road traffic, public transport, pedestrians and cyclists, while UAV simulation environments typically focus on flight dynamics, trajectory execution or sensor emulation as separate functions. The proposed FEDORA approach bridges this gap by enabling network-wide scenarios in which UAV-assisted journeys are assessed together with existing transport modes, considering complete passenger travel times, hub access, waiting, capacity constraints and onward travel. In addition, the framework supports the representation of UAVs as mobile traffic-sensing platforms and provides a repeatable pipeline for varying demand, disruptions, infrastructure configurations and control strategies and evaluating their effects through consistent KPIs. To the best of our knowledge, few existing platforms combine these capabilities within a unified environment designed both for large-scale foresight analysis and for the progressive transfer of simulation findings towards real pilot preparation.

5.4 READINESS, EXPLOITATION AND DISSEMINATION

This section considers how far the work reported here has matured and how it is being made available beyond this deliverable report and the project itself. The three tasks are treated in turn, since their outputs differ in form.

The modular simulation platform of Task 5.4 is provided as a descriptive concept through this report in Chapter 2 and additionally presented in an end-to-end demonstrator for traffic signal control. The latter is released as open-source software alongside a comprehensive documentation (Section 2.4). Future possible extensions to facilitate a more thorough exploitation of the introduced scheme include the connection of further WP4 optimisation and control services as well as the inclusion of interfaces to additional environments, potentially including real-world pilot sites, both directed at deliverable D5.3. Due to its open availability and the structure built around defined interfaces, the platform is the main exploitable result of this first deliverable, into which the methods of the other tasks and work packages can be integrated as they mature, addressing the interoperability aspect of the integrated

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

simulation framework identified as Expected Result ER4.1. The open code, its documentation and the records it produces (Section 2.4) allow the demonstrations to be reproduced and extended by other technical partners, with exploitation and further dissemination pursued through the activities of WP7.

The Sim2Real framework developed so far in Task 5.5 represents a proof-of-concept demonstration of Sim2Real RL where the methodology has been validated in a controlled simulation environment. The proposed framework was evaluated through extensive experiments involving the transfer of reinforcement learning policies between two independent microscopic traffic simulators with different traffic dynamics, demonstrating both its feasibility and its effectiveness. Within FEDORA, this methodology is expected to evolve beyond the current proof of concept. Future work will focus on scaling the framework to larger and more realistic urban networks and will investigate budget-aware and uncertainty-aware Sim2Real approaches that minimise the amount of target-domain information required while explicitly accounting for uncertainty during the action transformation process. Furthermore, future developments will explore how RL policies trained in lower-fidelity simulation environments, particularly the macroscopic simulation framework developed in Task 5.1, can be effectively transferred to higher-fidelity microscopic simulators, enabling seamless knowledge transfer across different levels of traffic modelling. The scientific outcomes of Task 5.5 are going to be disseminated through presentations in scientific conferences and academic publications.

The UCY components developed under Task 5.6 are currently available as an operational in-silico prototype comprising scenario-configuration tools, a UAV trajectory planner, SUMO–TraCI integration, multimodal decision logic, repeatable study execution, output processing and KPI visualisation. The complete workflow is being validated in the Aglantzia test network, while its application to the Greater Nicosia pilot area will follow the completion of calibration using real traffic, public-transport and mobility data. The components are intended to be exploited as a reusable experimentation and pilot-preparation environment for testing alternative service designs, operational assumptions and FEDORA logic modules before real-world deployment. Their planned alignment with the FEDORA orchestration framework will further facilitate their reuse with alternative optimisation, control and forecasting components. Dissemination will be pursued through public FEDORA deliverables, project demonstrations, technical workshops, and the presentation of pilot-relevant results to public authorities, transport operators and research stakeholders.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

6 CONCLUSION AND NEXT STEPS

This deliverable reports the outputs of Tasks 5.4, 5.5 and 5.6. Task 5.4 has developed the model-agnostic and modular simulation platform, an agent-based interoperability framework in which interchangeable logic modules and application layer environments are coordinated by a single orchestrator. It has been demonstrated end to end for traffic signal control and released as open-source software. Task 5.5 has developed the Sim2Real techniques for model-to-application transferability and evaluated them on the network-level perimeter control problem as a proof of concept. Task 5.6 has established the foresight analysis scenario generation and evaluation capability, which combines parameterised scenario design, microscopic multimodal simulation, UAV service modelling and automated KPI extraction. The three tasks correspond to Expected Innovations EI4.4, EI4.5 and EI4.6. Together with the engines and models reported in D5.1, they contribute to the integrated and interoperable simulation framework (ER4.1) and to the foresight analysis with model-to-application transferability frameworks (ER4.3).

The results support the approach taken. In the demonstrator, logic modules ranging from a fixed schedule to a state-responsive strategy are exchanged without altering the environment, the orchestrator or the recorder. The compatible technical components developed across the project can therefore be connected on a common basis, where compatible. A policy transferred directly to a simulator with different dynamics loses a considerable part of its performance, and Grounded Action Transformation recovers most of that loss where sufficient data from the target environment are available. The foresight platform has been used to verify the complete scenario configuration, execution and evaluation workflow, as well as the passenger-level door-to-door decision on which the assessment of UAV-assisted journeys rests.

The work reported here represents an intermediate stage, and the components documented in this deliverable lay the groundwork for its continuation as outlined in Section 5.1. The demonstrator shows that further optimisation and control services from WP4 and WP5 can be accommodated as additional logic modules without the platform being re-engineered for each of them. The application layer environment interface already anticipates connections beyond simulation, including to pilot sites in the context of WP6. The integration efforts between the components discussed here remain aligned with the project's wider platform through the integration activities in WP2. On the methodological side, the transferability work provides the basis for its extension to larger and more realistic networks, to budget-aware and uncertainty-aware approaches and to transfer across simulation fidelities. The foresight platform provides a workflow that can be aligned with the FEDORA orchestration contract and that is being prepared for the Greater Nicosia pilot network. The platform of Task 5.4 is the main exploitable result of this first deliverable. It is openly available together with its documentation, so that the demonstrations can be reproduced and extended by other technical partners. Further exploitation and dissemination are pursued through the activities of WP7 and through the publications and demonstrations foreseen by the individual tasks. The results of this continuation are to be reported in deliverable D5.3.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

7 REFERENCES

Chapter 2

1. Alvarez Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938
2. Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., & Hubaux, J.-P. (2008). TraCI: An Interface for Coupling Road Traffic and Network Simulators. Proceedings of the 11th Communications and Networking Simulation Symposium (CNS '08), 155–163. https://doi.org/10.1145/1400713.1400740
3. Rondinone, M., Maneros, J., Krajzewicz, D., Bauza, R., Cataldi, P., Hrizi, F., Gozalvez, J., Kumar, V., Röckl, M., Lin, L., Lazaro, O., Leguay, J., Härri, J., Vaz, S., Lopez, Y., Sepulcre, M., Wetterwald, M., Blokpoel, R., & Cartolano, F. (2013). iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications. Simulation Modelling Practice and Theory, 34, 99–125. https://doi.org/10.1016/j.simpat.2013.01.007
4. Schrab, K., Neubauer, M., Protzmann, R., Radusch, I., Manganiaris, S., Lytrivis, P., & Amditis, A. J. (2023). Modeling an ITS Management Solution for Mixed Highway Traffic With Eclipse MOSAIC. IEEE Transactions on Intelligent Transportation Systems, 24(6), 6575–6585. https://doi.org/10.1109/TITS.2022.3204174
5. Sommer, C., German, R., & Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing, 10(1), 3–15. https://doi.org/10.1109/TMC.2010.133

Chapter 3

6. Daganzo, C. F. (2005). Improving city mobility through gridlock control: An approach and some ideas. https://escholarship.org/uc/item/7w6232wq
7. Gartner, N. H., & Wagner, P. (2004). Analysis of Traffic Flow Characteristics on Signalized Arterials. Transportation Research Record: Journal of the Transportation Research Board, 1883(1), 94–100. https://doi.org/10.3141/1883-11
8. Geroliminis, N., & Daganzo, C. F. (2007). Macroscopic modeling of traffic in cities. https://trid.trb.org/View/801089
9. Aboudolas, K., & Geroliminis, N. (2013). Perimeter and boundary flow control in multi-reservoir heterogeneous networks. Transportation Research Part B: Methodological, 55, 265–281. https://doi.org/10.1016/j.trb.2013.07.003
10. Geroliminis, N., Haddad, J., & Ramezani, M. (2013). Optimal Perimeter Control for Two Urban Regions With Macroscopic Fundamental Diagrams: A Model Predictive Approach. IEEE Transactions on Intelligent Transportation Systems, 14(1), 348–359. https://doi.org/10.1109/TITS.2012.2216877
11. Hajiahmadi, M., Haddad, J., De Schutter, B., & Geroliminis, N. (2015). Optimal Hybrid Perimeter and Switching Plans Control for Urban Traffic Networks. IEEE Transactions on Control Systems Technology, 23(2), 464–478. https://doi.org/10.1109/TCST.2014.2330997
12. Kampitakis, E., & Vlahogianni, E. I. (2025). Decentralized multi-region perimeter control in complex urban environments using reinforcement and imitation learning. Transportation Research Part C: Emerging Technologies, 178, 105253.
13. Ni, W., & Cassidy, M. J. (2019). Cordon control with spatially-varying metering rates: A Reinforcement Learning approach. Transportation Research Part C: Emerging Technologies, 98, 358–369. https://doi.org/10.1016/j.trc.2018.12.007
14. Yoon, J., Kim, S., Byon, Y.-J., & Yeo, H. (2020). Design of reinforcement learning for perimeter control using network transmission model based macroscopic traffic simulation. PLOS ONE, 15(7), e0236655. https://doi.org/10.1371/journal.pone.0236655
15. Zhou, D., & Gayah, V. V. (2021). Model-free perimeter metering control for two-region urban networks using deep reinforcement learning. Transportation Research Part C: Emerging Technologies, 124, 102949. https://doi.org/10.1016/j.trc.2020.102949

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

16. Zhou, D., & Gayah, V. V. (2023). Scalable multi-region perimeter metering control for urban networks: A multi-agent deep reinforcement learning approach. Transportation Research Part C: Emerging Technologies, 148, 104033. https://doi.org/10.1016/j.trc.2023.104033
17. Jakobi, N., Husbands, P., & Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Morán, A. Moreno, J. J. Merelo, & P. Chacón (Eds.), Advances in Artificial Life (pp. 704–720). Springer. https://doi.org/10.1007/3-540-59496-5_337
18. Chebotar, Y., Handa, A., Makoviychuk, V., Macklin, M., Issac, J., Ratliff, N., & Fox, D. (2019). Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience. 2019 International Conference on Robotics and Automation (ICRA), 8973–8979. https://doi.org/10.1109/ICRA.2019.8793789
19. Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. 2018 IEEE International Conference on Robotics and Automation (ICRA), 3803–3810. https://doi.org/10.1109/ICRA.2018.8460528
20. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 23–30. https://doi.org/10.1109/IROS.2017.8202133
21. Hanna, J. P., Desai, S., Karnan, H., Warnell, G., & Stone, P. (2021). Grounded action transformation for sim-to-real reinforcement learning. Machine Learning, 110(9), 2469–2499. https://doi.org/10.1007/s10994-021-05982-z
22. Da, L., Mei, H., Sharma, R., & Wei, H. (2023a). Sim2Real Transfer for Traffic Signal Control. 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), 1–2. https://doi.org/10.1109/CASE56687.2023.10260398
23. Da, L., Mei, H., Sharma, R., & Wei, H. (2023b). Uncertainty-Aware Grounded Action Transformation Towards Sim-to-Real Transfer for Traffic Signal Control. 2023 62nd IEEE Conference on Decision and Control (CDC), 1124–1129. https://doi.org/10.1109/CDC49753.2023.10383645
24. Da, L., Gao, M., Mei, H., & Wei, H. (2024). Prompt to Transfer: Sim-to-Real Transfer for Traffic Signal Control with Prompt Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 38(1), 82–90. https://doi.org/10.1609/aaai.v38i1.27758
25. Turnau, J., Da, L., Vo, K., Rafi, F. A., Bachiraju, S., Chen, T., & Wei, H. (2025). Joint-Local Grounded Action Transformation for Sim-to-Real Transfer in Multi-Agent Traffic Control (arXiv:2507.15174). arXiv. https://doi.org/10.48550/arXiv.2507.15174
26. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms (arXiv:1707.06347). arXiv. https://doi.org/10.48550/arXiv.1707.06347
27. Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., & Li, Z. (2019). CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. The World Wide Web Conference, 3620–3624. https://doi.org/10.1145/3308558.3314139
28. Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of SUMO-Simulation of Urban MObility. International Journal on Advances in Systems and Measurements, 5(3 & 4), 128–138.
29. Andrychowicz, O. M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., & Zaremba, W. (2020). Learning dexterous in-hand manipulation. The International Journal of Robotics Research, 39(1), 3–20. https://doi.org/10.1177/0278364919887447

Chapter 4

30. Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). SUMO - Simulation of Urban MObility: An Overview. Proceedings of SIMUL 2011, Barcelona, Spain, pp. 63-68.
31. Alvarez Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. 21st IEEE International Conference on Intelligent Transportation Systems, pp. 2575-2582. https://doi.org/10.1109/ITSC.2018.8569938.
32. Codeca, L., and Härri, J. (2017). Towards Multimodal Mobility Simulation of C-ITS: The Monaco SUMO Traffic Scenario. 2017 IEEE Vehicular Networking Conference, pp. 97-100. https://doi.org/10.1109/VNC.2017.8275627.

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI

33. Lobo, S. C., Neumeier, S., Fernandez, E. M. G., and Facchi, C. (2020). InTAS - The Ingolstadt Traffic Scenario for SUMO. arXiv:2011.11995.
34. Eclipse SUMO Project (2026). SUMO Documentation: Intermodal Routing, Person Trips and Traffic Control Interface. Retrieved 25 July 2026 from https://eclipse.dev/sumo/docs/.
35. Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In M. Hutter and R. Siegwart (Eds.), Field and Service Robotics, Springer Proceedings in Advanced Robotics, Vol. 5, pp. 621-635. Springer.
36. Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). RotorS - A Modular Gazebo MAV Simulator Framework. In A. Koubaa (Ed.), Robot Operating System (ROS): The Complete Reference, Vol. 1, pp. 595-625. Springer.
37. Jacinto, M., Pinto, J., Patrikar, J., Keller, J., Cunha, R., Scherer, S., and Pascoal, A. (2023). Pegasus Simulator: An Isaac Sim Framework for Multiple Aerial Vehicles Simulation. arXiv:2307.05263.
38. Jiang, X., Tang, Y., Cao, J., Bulusu, V., Yang, H., Peng, X., Zheng, Y., Zhao, J., and Sengupta, R. (2023). Simulating the Integration of Urban Air Mobility into Existing Transportation Systems: A Survey. arXiv:2301.12901.
39. Garrow, L. A., German, B. J., and Leonard, C. E. (2021). Urban Air Mobility: A Comprehensive Review and Comparative Analysis with Autonomous and Electric Ground Transportation for Informing Future Research. Transportation Research Part C: Emerging Technologies, 132, 103377. https://doi.org/10.1016/j.trc.2021.103377.
40. Butilă, E. V., and Boboc, R. G. (2022). Urban Traffic Monitoring and Analysis Using Unmanned Aerial Vehicles (UAVs): A Systematic Literature Review. Remote Sensing, 14(3), 620. https://doi.org/10.3390/rs14030620.
41. Schaffland, A., Nelson, J., and Schöning, J. (2024). Simulating Traffic Networks: Driving SUMO Towards Digital Twins. SUMO Conference Proceedings, 5.

Funded by the European Union

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

ANNEX – ONLINE SOFTWARE AND MANUALS

This annex provides references to the software developed in the context of this deliverable, focussing on the orchestration platform described in Chapter 2, including source code repositories and user documentation. The listed references include the location of the source code repositories, associated documentation (README files, user manuals, installation guides, and administrator guides), and, where applicable, the corresponding software release, version tag, or commit identifier. The referenced artefacts represent the versions available at the time of submission of this deliverable.

ANNEX A – SOFTWARE REPOSITORY REFERENCES

IDENTIFIER	DESCRIPTION	REPOSITORY	VERSION / COMMIT
SW-01	Project source code	https://github.com/fedora-horizon	Release v1.0
SW-02	FEDORA Platform Demonstrator	https://github.com/fedora-horizon/fedora_platform	Main branch (accessed version)

ANNEX B – USER DOCUMENTATION

DOCUMENT ID	TITLE	LOCATION
UM-01	User Manual	https://ivt-baug-ethz.github.io/fedora_platform/
UM-02	Platform Architecture Documentation	https://ivt-baug-ethz.github.io/fedora_platform/architecture/
UM-03	Component and Demonstrator Documentation	https://ivt-baug-ethz.github.io/fedora_platform/components/

Funded by the European Union

Project funded by Schweizerische Eidgenossenschaft Confédération suisse Confederazione Svizzera Confederaziun svizra Swiss Confederation — Federal Department of Economic Affairs, Education and Research EAER State Secretariat for Education, Research and Innovation SERI