www.orchestra2020.eu

ORCHESTRA Project Deliverable: D5.1

Simulation architecture

Authors:

Alessandro Bombelli, Ilias Parmaksizoglou (TUDELFT)

Lucio Truaisch, Jan Huber, Vincent Robatel (HES-SO)

Elena Branchini, Francesca Romano (SEA)

Nicola Cavagnetto (DBL)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Deliverable ID:	D5.1	Deliverable title:	Simulation architecture
Release Number:	1.0		
Release Date:	2022-10-31		
Deliverable Description	Description of the simulation design for a multimodal traffic management ecosystem, covering the different levels of abstraction.		
Dissemination Level	PU = Public		
Deliverable Type	R = Report		
Due date	2022-10-31		

Release History

Version	Date	Internal Review Milestone Reached (if relevant) *	Summary of main changes introduced in this version
0.1	2022-05-31		Draft
0.2	2022-08-22	Intermediate	First structure and description of simulation architecture is provided. Information specific to each Living Lab (LL) are also provided to highlight those features of the architecture that are LL-specific
0.5	2022-10-23	External proposed	
1.0	2022-10-31	Released	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

About ORCHESTRA

The problem addressed by ORCHESTRA is that traffic caused by transport has many negative effects. There are congestions, delays, emissions, and negative impacts on urban environments, and in case of disruptions, there may be huge consequences on the efficiency and timeliness. These challenges are hard to handle due to lack of coordination between the different transport modes.

The long-term vision of ORCHESTRA is a future where it is easy to coordinate and synchronise the traffic management of all modes to cope with diverse demands and situations. The overall objective of ORCHESTRA is to provide European policy makers, public authorities, transport providers and citizens with new knowledge and technical and organisational solutions to enhance collaboration and synchronising of operations within and across transport modes.

The project will:

- Establish a common understanding of multimodal traffic management concepts and solutions, within and across different modes, for various stakeholders and multiple contexts
- Define a Multimodal Traffic Management Ecosystem (MTME) where traffic managements in different modes and areas (rural and urban) are coordinated to contribute to a more balanced and resilient transport system, bridging current barriers and silos
- Support MTME realisation and deployments, through the provision of tools, models, and guidelines – including the integration of connected and automated vehicles and vessels (CAVs)
- Validate and adjust MTME for organisational issues, functionality, capability, and usability
- Maximise outreach and uptake of project results through strong stakeholder involvement

ORCHESTRA's main advancements beyond state-of-the-art are related to four focus areas:

- MTME facilitated by: 1) a Polycentric Multimodal Architecture (PMA) specifying how systems collaborate. 2) Flexible organizational and business models. 3) Simulation and training tools. 4) Policy and regulatory recommendations. 5) data governance and sharing framework
- Traffic orchestration supporting optimal traffic flows, adapted to current and foreseen situations and societal aspects. Data on ongoing and planned transports as well as other issues that may affect the traffic will be monitored and used in decision support and to facilitate resilience
- Coordination across modes and networks bridging current silos, ensuring best possible utilisation of transport system as a whole
- Traffic management supporting more optimal multimodal transport services and fleet operations, those carried out by CAVs included. Transport operations will be guided and controlled according to pre-defined rules and trade-offs between different optimisation targets.

The project will validate and evaluate the multimodal traffic management concept and related tools in its two Living Labs, both in Norway and Italy, covering freight and person transports across road, rail, water, and air.

Legal disclaimer

For more information

Project Coordinator: Runar Søråsen, runar.sorasen@its-norway.no
Dissemination Manager (WP7 leader): Trond Hovland, trond.hovland@its-norway.no

Executive Summary

The primary goal of this document is to provide a description of the simulation architecture in terms of modelling techniques, assumptions, features, and goals of the simulator employed in the context of ORCHESTRA's two Living Labs (LLs).

The first major decision that was taken regards the level of granularity of the simulation. The ORCHESTRA project aims at establishing a common understanding of multimodal traffic management concepts and solutions, within and across different modes, for various stakeholders and in multiple contexts. As such, a macrosimulation capable of monitoring and influencing traffic flow conditions seems an appropriate modelling approach. Notwithstanding, in both LLs a crucial role is played by a (air)port. i.e., Herøya Industrial Park (HIP) in Norway and Malpensa airport (MXP) in Italy, where the traffic flows merge. One of the goals of ORCHESTRA is to enhance operational efficiency inside such ports. Such goal is best achieved if a microsimulation approach is used, where each user is modelled singularly to best capture dynamics, capacity levels, and potential operational bottlenecks in the LLs. For the aforementioned reasons, a hybrid macroscopic/microscopic simulation approach is chosen. A first simulation block, based on a graph representation of the different transport modes, is designed using the Python programming language and the OSMnx and networkX packages. This approach allows for a faithful representation of the transport networks, as data is exported from Google Open Street Maps, together with optimization capabilities such as shortest path computation within the network. Use of a real-time Application Programming Interface (API) is also facilitated that can compute such paths and patterns across the multimodal network. Goal of this simulation block is to provide a macroscopic demand (expressed as traffic flow) profile per time-step that reaches the LL-specific port. This demand profile is then used as input by a second simulation block based on the commercial software Aimsun, which models the two ports of interest (HIP and MXP) using a microscopic approach. The demand profiles are disaggregated so that single users (trucks for HIP, passengers for MXP) dynamics are simulated and internal operations can be monitored.

When simulating operations inside the two ports (HIP and MXP), a set of KPIs defined in workpackage (WP) 6 will be monitored. If operations result in KPI conditions that fall outside the defined bounds (e.g., excessive waiting time at security checkpoints in MXP), enabling tools devised in WP4 will be activated which will re-distribute traffic flows (first simulation block) and will result in a new microsimulation (second simulation block) in an iterative fashion, until all KPIs are within bounds. Given the size of the networks and the desired level of accuracy, this architecture is considered to perform best in an off-line setting. For (quasi) real-time operations, such as disruption management, two options are proposed that can speed up the computational effort so that good quality solutions are achieved with a limited computational effort. While the simulation architecture is the same for both LLs, LL-dependent specifications in terms of transport network representation, assumptions, or needed inputs for the simulation are provided in the document. For HIP, the simulation will focus on truck arrivals at the entrance of the port, as they need to be regulated in a way that avoids queues and congestions. In addition, internal operations inside HIP will entail a combination of trucks, connected and automated vehicles (CAVs), and barges. In future operations, a railway track operating as an additional source of inbound/outbound freight is envisioned. As such, synchronization of all these actors in conjunction to a seamless transition of freight is key to ensure operational efficiency. Hence, this LL is mostly multimodal within its own premises. On the contrary,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

for MXP multimodal operations are envisioned to be carried out enroute to the airport, to facilitate and enhance smooth trips for passengers from their origin to the airport. The key here is to exploit multimodality from the municipality of Milan so that passengers on their way to the airport make a balanced use of the different modes available (with the traffic orchestrator influencing such flows), in a way that does not overload the traffic network and enables all users to be guaranteed a satisfactory travelling experience. Hence in this LL multimodality is mostly exploited outside the premises of the final focal point (i.e., MXP).

In addition, in this deliverable a focus is put on how the simulation architecture supports and complies with the paradigms of the Polycentric Multimodal Architecture (PMA) and Multimodal Traffic Management Ecosystem (MTME). In particular, it is shown how the architecture naturally captures the concept of multimodality and polycentricity and how the concept of centres can be adjusted dynamically according to the specific scenario under scrutiny. Finally, considerations on potential computational limitations the current simulation architecture might face once it is scaled-up are analysed and potential remedies are provided.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

Table of Figures

List of Tables

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

List of Abbreviations

Table 1: List of abbreviations

Abbreviation	Explanation
API	Application Programming Interface
CAV	Connected and Automated Vehicles/vessels
GA	Governance Area
HIP	Herøya Industrial Park
KPI	Key Performance Indicator
LL	Living Lab
MTME	Multimodal Traffic Management Ecosystem
MXP	Malpensa Airport
NIL	Nuclei di Identita’ Nazionale
OD	Origin-Destination
PMA	Polycentric Multimodal Architecture
WP	Work Package

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

1 About this Deliverable

1.1 Why would I want to read this deliverable?

The deliverable focuses on the description of the simulation architecture that serves as the main engine to study and assess the polycentric architecture monitoring and controlling multimodal traffic flows. It takes input from work package (WP) 2 (Target vision and scenarios) and WP3 (Polycentric traffic management design) at the higher level, and from WP4 (Enabling toolkits) and WP6 (Definition of KPIs) at the more operational and implementation level.

1.2 Intended readership/users

This report is relevant for project members of ORCHESTRA involved in the design of the simulation environment and Living Lab (LL) operations. From a modelling standpoint, interested readers can get insights into how the simulation architecture is designed and provide valuable feedback on the modelling choices, assumptions, and simplifications made. On the other hand, this is also interesting for practitioners and involved stakeholders, in a two-fold manner. First, the simulation architecture and the subsequent simulation will be either directly used by them in real-life operations as provided, or will support their decision-making process. Hence, this deliverable describes in a high-level manner what the added value of the simulation is with respect to current operations. Second, as this project is still in development, they can provide feedback on such modelling assumptions and simplifications, and on how well they might represent real operations.

1.3 Other project deliverables of interest

In addition to this report, the reader is invited to read the other deliverables dealing with technical development for multimodal traffic[1]:

- D4.1: Initial version of technical tools
- D4.2: Final version of technical tools
- D5.2: Simulator

D4.1 and D4.2 relate to, respectively, the initial and final version of the technical tool that support and integrate the simulation architecture and the simulator environment (D5.2). As such these four deliverables are closely intertwined. We will rely on our simulator to simulate and propagate vehicle/passenger dynamics around and inside the two LLs (HIP and MXP), to identify if some of the defined KPIs are not satisfied and, if so, the enabling toolkits will compute recovery actions that the traffic orchestrator can implement.

1.4 Involvement in work

Partners involved in this deliverable are TUDELFT, HES-SO, SEA and DBL.

TUDELFT has led the deliverable designing the simulation architecture and gathering relevant information concerning input data. HES-SO has provided expertise and support with respect to the

[1] Note that these deliverables are marked as confidential, i.e. only accessible for members of the consortium (including the Commission Services)

microsimulation, as it will performed by them due to license availability and background knowledge and expertise. SEA and DBL have been providing input and feedback with respect to the Italian LL.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

2 Types of Traffic Simulation Models

In the current research on traffic modelling, 3 major modelling perspectives are generally applied. This is also translated into a set of three main traffic simulation modelling techniques, namely *microscopic*, *mesoscopic*, and *macroscopic* simulation, which range from a more disaggregated to a more aggregated paradigm. A brief description of microscopic and macroscopic simulation will be provided first, followed by a description of the mesoscopic approach (which is a middle ground approach compared with the other two).

2.1 Microscopic simulation

Microscopic modelling of traffic flows is based on the description of the motion of each individual agent composing the traffic stream. Note that we use the term "agent" to keep the setting general. According to the specific application, agents could be vehicles, aircraft, pedestrians, etc. As the motion of each agent needs to be mapped, implies modelling the actions of each agent with the level of accuracy the simulation requires. Such actions might feature such as acceleration, deceleration, speed, position, heading etc. Microscopic traffic simulation has been applied mainly, yet not solely, to road vehicles (cars). In such scenario, a rule of thumb was established as early as the fifties by the California Motor Vehicle Code, namely "*A good rule for following another vehicle at a safe distance is to allow yourself at least the length of a car between your vehicle and the vehicle ahead for every ten miles per hour of speed at which you are traveling.*". This concept was further extended and became the starting point of the so-called *car-following* models as a form of stimulus response equations, where the behaviour of a following vehicle is directly affected by the behaviour of the vehicle(s) directly downstream.

To summarize, microscopic simulation models and propagates in time the dynamics of each agent playing an active role in the simulation environment. The way agents affect each other is simulation-specific, but a common denominator is that this approach is generally computationally heavy, as it entails storing and updating information for all agents with the same level of accuracy. An exemplificative example of microscopic simulation is shown in Figure 1.

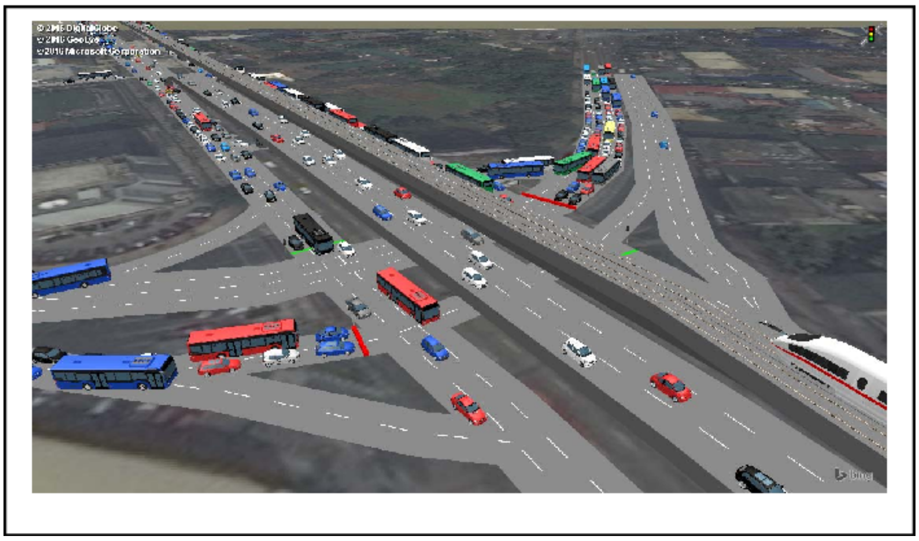


Figure 1: Example of microscopic simulation (Ejercito, Gayle, Nebrija, Feria, & Figueroa, 2017)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

2.2 Macroscopic simulation

Macroscopic modelling of traffic flows is based on the assumption that the status of the system is mapped, monitoring the so-called *control volumes*, i.e., some regions of the modelling space where aggregate variables are computed. Control volumes can be intersections, neighbourhoods, cities etc for vehicle and/or urban simulation, while they are generally sectors or centres for air traffic. General macroscopic variables that are used to monitor the state of the system are traffic density, volume, and speed to cite a few examples. In essence, macroscopic modelling gives up the ambition to map every agent in the system, and focuses instead on aggregate variables computed in each control volume. Conservation laws such as conservation of flow between control volumes are used to propagate the dynamics in the system.

To summarize, a macroscopic simulation focuses on control volumes where aggregate variables are computed. As such, agent-specific features are lost in favour of aggregate variables (e.g., vehicle flow rather than individual vehicles' trajectories). While the level of accuracy is decreased, the lower computational effort makes it possible the analysis of larger-scale scenarios. An exemplificative example of microscopic simulation is shown in Figure 2

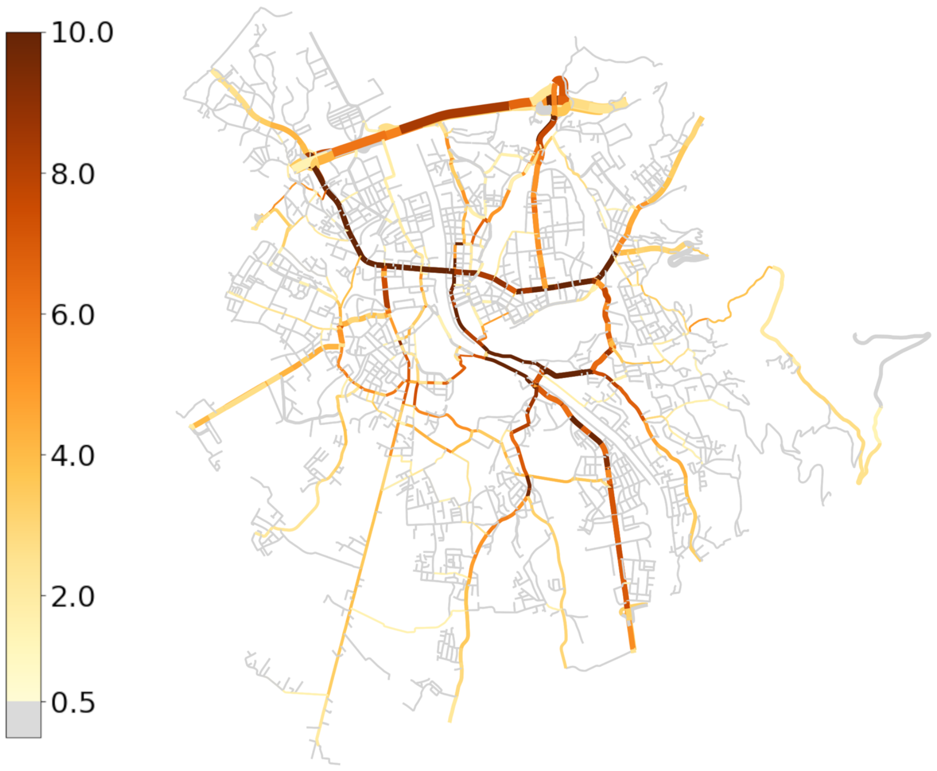


Figure 2: Example of macroscopic simulation (Plakolb, Jager, Hofer, & Fullsack, 2019)

2.3 Mesoscopic simulation

Mesoscopic modelling of traffic flows is a middle ground solution between the previous 2 approaches. The idea behind mesoscopic modelling is to reduce the complexity of the dynamics of individual agents (hence reducing the computational complexity of the model as a macroscopic model would do), while retaining some capabilities to map behaviour at the agent-level (as a microscopic model would do). Two distinct options to achieve this is: (i) allow platooning between agents (hence

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

reducing the number of agents, as now the number of platoons is the number of agents), or (ii) keep to original number of agents, but to heavily simplify their dynamics.

As it concerns microscopic and macroscopic simulations, the evolution of time is generally discretized using a fixed time-internal Δt. Only timestamps that are multiple of Δt, i.e., $\{0, \Delta t, 2\,\Delta t, \dots, T\}$ (where T is the length of the simulation horizon) are defined in the simulation. This approach is also known as *synchronous timing*. For mesoscopic simulation, on top of the synchronous timing, there exists another approach called *asynchronous timing* (or event-based), where the state of the system changes anytime a new event occurs (a new event can be a car crossing an intersection, a truck arriving at a port, a pedestrian starting to cross a road, etc). As events might, and generally will not be equally spaced in time, this approach is asynchronous (de Souza, Verbas, & Auld, 2019).

2.4 Choice of simulation modelling technique

In the context of the ORCHESTRA project, both *traffic flow management* and *transport management* are considered. The former pertains the necessity to guarantee smooth traffic conditions at the system level, e.g., that the traffic networks around HIP or MXP are not congested. This is best achieved with a macroscopic approach, where variables such as traffic density and volume can be monitored. Note that this is also in line with the concept of a *polycentric* architecture, where control centres can be regions of the road network (or stations for the rail network) that can coordinate with each other using information such as traffic flow between centres. This paradigm is aligned with the concept of control centres that the Polycentric Multimodal Architecture (PMA) advocates for (we refer interested readers to D3.1 for more details). At the same time, ORCHESTRA end-users (trucks headed to HIP, passengers relying on ORCHESTRA headed to MXP) share full or partial information with the traffic orchestrator and the transport service providers, which entails transport management to be carried out. For this set of agents, actual routing can be computed that accounts for personal preferences, while considering traveling time and other constraints from the system level (e.g., level of congestion which has a direct effect on traveling time). Given the aforementioned requirements, our approach is split into two blocks. The first one focuses on arrival/departure flows per unit of time (to be determined) to/from the focal point of the LL under scrutiny. To this avail, we will employ a microsimulation approach that allows for quick computation of trajectories to the LL (focusing on arrivals). Our goal is to define arrival patterns of trucks (HIP) or passengers (MXP) at the (air)port that are computed using microsimulation considering origin point, preferred arrival times, and current traffic conditions via access to a real-time API. Such patterns, which are time-continuous in nature, are then aggregated per time-unit (e.g. 30 minutes) to provide an aggregated demand for the second simulation block. The reason behind the aggregation process is that we believe the (air)port can control arrivals (via traffic orchestration) at such aggregated level, but uncertainty regarding precise arrivals times shall always be present. As such, working on arrival rates per unit-time implies a loss in accuracy, but a more realistic representation in our view. With a slight abuse of notation, we will consider this first step a macrosimulation step as the final outcome is arrival rates per unit-time, although the way they are computed relies on a microsimulation paradigm. The second step uses as input the arrival rates computed as just described. It then performs a microsimulation that allows every agent (truck or passenger) to move inside the premises of the LL in an individual and decentralized fashion. In conclusion, our simulation architecture entails a first step that, although relying on some microsimulation concepts, we consider a macrosimulation. The output of this first step becomes input to the second step that operates within the premises of each LL and is based on a microsimulation paradigm.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

3 Choice of simulation software

Two implementation approaches have been considered and used in the first version of the simulation environment. They are the programming language Python and the software Aimsun, and the reasoning behind the choice is explained and motivated in the following.

3.1 Python

Python is an object-oriented open-source programming language that has been extensively used by researchers in the last decades. Its open-source nature makes it ideal for research purposes. In addition, most packages are well-documented, and the Python community is very extensive, making code development and prototyping easier. In addition, the two partners mostly involved in task 5.1 (TUDELFT and HES-SO) have extensive experience with this programming language.

The other advantage of Python is that it enables for more code control and verification with respect to a commercial software, as all source code can be checked and analysed, and code developers can follow the development process "from cradle to grave". This also allows for more flexibility and customization, which is paramount in such a project.

Three Python packages that are used extensively in the simulation are **OSMnx** (https://osmnx.readthedocs.io/en/stable/), **networkX** (https://networkx.org/), and **geopandas** (https://geopandas.org/en/stable/). OSMnx is a package that enables users to download and process geospatial data from OpenStreetMap and model, project, visualize, and analyse real-world street networks using a network theory perspective interfacing with the networkX package. OSMnx can even retrieve, given a geographic region of interest, transport networks of different modes (e.g., rail, railway, and subway networks), which perfectly aligns with the multimodality paradigm of the LLs defined within the ORCHESTRA project. NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Note that, although OSMnx already offers many capabilities that are directly inherited from networkX, we are listing this latter package explicitly as a stand-alone package as manipulating data retrieved by OSMnx with networkX might be easier and offer more post-processing capabilities in specific circumstances.

As both LLs are topologically defined as multimodal traffic networks, Python can exploit the capabilities of OSMnx in conjunction to geopandas to define the networks of interest in terms of nodes and edges, which will be LL-specific. For example, in both LLs the road network of interest around HIP and MXP can be easily defined. Nodes are intersections between roads, and edges are directed connections between nodes, i.e., portions of roads. Note that the capabilities of OSMnx can be exploited to filter out unnecessary roads, such as secondary ones, and only include major arteries and highways, for example. For the Italian LL, the railway network can be modelled in a similar fashion, with nodes being train stations and edges being directed railway connections between stations. Development of technical tools to support the simulation, as defined by D4.1 (Initial version of Technical Enabling tools), has been at a significant part open-source and reliant on the aforementioned Python packages.

The use of a specialized travel information API will be employed to update the network structures with both real-time and historical link specific characteristics. Current implementations make use of the HERE Routing API, an HTTP JSON REST API that provides route calculation between two or more locations for several regions in the world. The HERE Routing API allows users to specify options that influence route calculation, as well as request additional information related to the route

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

to suit specific needs. Calculation of routes can include user-imposed preferences, such as specific modes, avoidance of areas and inclusion of a stopover. Information extracted from the API included trajectories (polylines of coordinates), time of traversal (historical and currently estimated), lengths, directions etc. For the Milan LL the use of the HERE Public Transit API will also be deployed for information gathering, mostly regarding the first/last mile of travellers to/from the airport. Both APIs can be directly implemented within a Python environment. Finally, geopandas a package that focuses on geospatial data. It extends the datatypes used by pandas (https://pandas.pydata.org/) to allow spatial operations on geometric types. Geometric operations are performed by shapely (https://shapely.readthedocs.io/en/stable/manual.html).

3.2 Aimsun

Aimsun (Aimsun, 2022) is a commercial software for transportation planning and traffic management that offers simulation capabilities ranging from micro- to macroscopic simulation. It provides excellent visualization tools and an ample spectrum of post-processing tools for data analysis. In addition, one partner heavily involved in task 5.1 (HES-SO) has previous experience with Aimsun.

Moreover, Aimsun enables to model geometry and topology of transport infrastructure and buildings with an unrivalled level of precision, which is paramount for some microsimulation applications that might be relevant within the ORCHESTRA project. A potential application of this kind is the modelling of passenger dynamics inside the MXP terminal, at baggage drop-off, security checkpoints, and gates.

However, the real strength of Aimsun is the possibility to interact programmatically with it using the Advanced Programming Interface (API). This API exposes all the variables, which are accessible using the graphical simulation interface, and makes them accessible outside of Aimsun, and also allows for the addition of new ones. Knowing this, it is possible to generate agents, and modify their attributes and behaviour. Furthermore, it is also possible to link agents with other agents, which is especially useful for the simulations of connected and automated vehicles and vessels (CAVs).

All of this allows for an easy interface between the Python simulation and the Aimsun simulation. The Python simulation will output all the required information in a pre-defined format, and through a simplified abstraction layer, this information will be translated and injected into the Aimsun simulation using the API.

The information used as input for the Aimsun simulation, which is the output of the Python simulation, is the following:

- Origin-Destination (OD) matrices for a pre-defined time period
    - Number of vehicles
    - Vehicle type and attributes
    - Origin and destination of the vehicles

The vehicle types and their attributes are the following:

- Car
    - Capacity
    - Emission model
    - Vehicle dynamics
- CAV
    - Vehicle dynamics

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

- Truck
    - Capacity
    - Emission model
    - Vehicle dynamics
    - Type of transport
- Boat
    - Capacity
    - Emission model
    - Vehicle dynamics
    - Type of transport
- Train
    - Capacity
    - Emission model
    - Vehicle dynamics
- Plane
    - Capacity
    - Emission model
    - Vehicle dynamics
- Pedestrian
    - Vehicle dynamics

Note that in the aforementioned definitions, we apply the term “vehicle dynamics” also for pedestrian dynamics (with a slight abuse of notation), as there is no explicit modelling difference, apart from input parameters, in the way the different dynamics are computed and propagated in the Aimsun environment.

It is important to note that “Destination” in a Python or Aimsun OD-matrix has a different meaning, and therefore a conversion needs to take place. In our microscopic simulations there are four data points that define the path of a vehicle (including pedestrians) within the overall context of the simulation. These data points are:

- Provenance
    - Where the journey of the vehicle started
    - E.g., Milan
- Entrance
    - Where the vehicle enters the microscopic simulation (simulated by Aimsun)
- Destination Infrastructure
    - The infrastructure the vehicle visits after entering and before exiting the microscopic simulation
    - E.g., port facility
- Exit
    - Where the vehicle exits the microscopic simulation (simulated by Aimsun)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

Table 2 below lists the above-mentioned data points in the column "Micro simulation", illustrates which data point corresponds to the origin and destination of each simulation, and shows the necessary conversions to be usable in the Aimsun simulation:

Table 2: Conversion between Python and Aimsun data types

Micro simulation	**Python OD-Matrix**	**Aimsun OD-Matrix**	**Conversion**
Provenance	Origin	-	-
Entrance (into the simulation)	-	Origin	Provenance -> Entrance
Destination Infrastructure (path to take)	Destination	(Route)	-
Exit (out of the simulation)	-	Destination	Definition Infrastructure -> Exit

3.3 Advantages, disadvantages, and synergies of the two simulations

The adoption of Python and Aimsun to perform simulation is intended to leverage the advantages of both software tools, and hence define a framework where the two simulation approaches can be used in synergy.

Python, being the more customizable among the two, will be exploited for fast simulation and integration with the enabling tools from WP4. In fact, the status of the (network) system can be monitored and assessed using the KPIs defined by WP6 relying on the capabilities of the networkX package. For example, at the macroscopic level, traffic density, speed, and flows can be defined as properties of each edge and easily updated at every time-step of the simulation as traffic flow dynamics are propagated. In case of congestion or disruption, the enabling tools from WP4 can be ***directly*** used to re-compute shortest paths across the networks to both (i) influence traffic via smart infrastructure and (ii) provide routing alternatives to end-users directly affected by ORCHESTRA (e.g., trucks headed to HIP or passengers headed to MXP). Note that this smooth integration of the enabling tools (WP4) and the simulation environment (WP5) is not directly possible, to the best of the authors' knowledge, with Aimsun. While the properties of the (network) system can still be monitored and benchmarked against the thresholds defined in WP6, the developed enabling tools can only ***indirectly*** be employed as follows:

The simulation is run in Aimsun and, once a warning is triggered that the system is not operating within capacity limits (as established by WP6), the output referring to the time-step that triggered the warning is saved, exported, and converted into a Python-friendly format. Then, the enabling tools (WP4) are activated and steer traffic flow conditions within capacity limits. This output is saved and re-converted into an Aimsun-friendly format that will become the initial condition of a new simulation. As such, this second approach is clearly more cumbersome and convoluted. This highlights the main disadvantage of Aimsun with respect to Python, i.e., ***a less straight-forward integration with the enabling tools (WP4)***.

Conversely, Aimsun gets the edge when ***requirements on the level of detail needed for the environment of the simulation are more stringent***, due to its superior graphical interface and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

microscopic-related features when compared to Python. We envision Aimsun to be employed when modelling internal operations of CAVs chaperoning trucks at HIP, or passenger dynamics inside the MXP terminal between the entrance, the check-in desks, and the security checkpoints.

We have hence identified a synergy, rather than an intrinsic incompatibility, between the two simulations. We envision Python to be used to simulate (and control via enabling tools) traffic flow conditions around the two LLs (mostly at the macroscopic level), with Aimsun then simulating internal operation in the LLs (mostly at the microscopic level). Note that this is intended to be performed mostly in an offline setting. For example, assume traffic flow conditions around MXP are simulated in Python for a full day of operations. This simulation is translated into an expected flow of passenger per hour (or selected time-step) at MXP. This time-series is received as input by Aimsun, which then models internal operations at MXP according to the simulated passenger arrival profile. If the microscopic simulation performed by Aimsun highlights congestion is likely to occur as, maybe, a spike of passenger is arriving within a specific time-window that cannot be properly handled by the airport, two options are possible. (i) The Aimsun simulation is run again increasing (if possible) the handling capabilities of the airport (e.g., opening an extra security lane), or (ii) an additional set of requirements is defined that is used as additional input for the enabling tools that are integrated with the Python simulation so that arrivals at the airport are smoothed and the aforementioned peak disappear. Note that this second approach is preferred as it better embodies the ORCHESTRA paradigm. We envision this feedback between the Python and Aimsun simulation to work well offline, but not to be employed in online setting as computational issues might arise similar to the intrinsic Python-Aimsun disadvantage we previously mentioned.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

4 Simulation Architecture Structure

Given the intrinsic differences between the two LLs, the final simulation tools will be LL-specific with features that might even be scenario-specific. Notwithstanding, the simulation architecture is LL-independent and is described in the following.

Consistently with what stated in the previous sections of the report, the simulation architecture is based on two main blocks that focus on; (i) traffic to/from the focal point of interest of the LL (HIP for the Norwegian LL, MXP for the Italian LL), and (ii) internal operations in the focal point of interest of the LL.

The first simulation block is coded in Python and is based on a graph-based structure that leverages the availability of open-source data for road/railway network structures to build such graph and the availability of complex network theory-based packages to update the information of the graph when simulating traffic flow. Given a specific transport mode, an associated graph can be built by defining **nodes** and **edges**. Nodes represent physical (infra)structures of the network, such as road intersections or train stations. Edges represent directed connections between nodes, such as portions of road or railway tracks. Goal of this simulation block is to map traffic flow along each edge of the graph as a function of a chosen time-step. Among all the edges comprising the graph, a special focus is put on those stemming from/headed to the focal point of interest, i.e., the node representing HIP or MXP, as their cumulative values represent, respectively, the outflow/inflow to the node of interest per unit-time. Hence, the output of the first simulation block is a time-series representing demand (traffic) in or out from HIP or MXP. In the rest of the report, we will focus on incoming traffic to the focal node, as this is the traffic defining the input for the second simulation block that simulates internal operations in HIP or MXP.

The second simulation block leverages the intrinsic versatility offered by Aimsun when a microsimulation is needed. As such, both HIP and MXP are modelled in great details in terms of layout and infrastructure within the Aimsun environment. The demand time-series stemming from the first simulation block are disaggregated into single individuals (also referred to as agents in the rest of the report). The dynamics of each agent is then handled within the Aimsun environment via microsimulation. To properly simulate such dynamics, each agent must be assigned a goal. For HIP, the goal of an agent (truck) is a destination yard, as the truck will have to offload cargo there to be later loaded on a barge that is stationed there. For MXP, the goal of an agent (passenger) is the gate associated to the departing flight. In both LLs, to reach the final goal each agent will interact with other agents and infrastructure. For example, in MXP passengers will need to pass through a security checkpoint before being able to move through the terminal and reach the intended gate. Within Aimsun, relevant KPIs will be monitored in accordance to the evaluation criteria defined in WP6. For example, the queuing level at the entrance of HIP or at the security checkpoints in MXP. In the latter case, the number of passengers who missed their flights due to excessive congestion in the terminal could also be monitored.

Every KPI of interest is compared with a (possibly time-varying) threshold as defined by WP6. If the comparison highlights operational issues due to capacity violations, such as excessive queues, then the time intervals when the violation occurred, and severity of such capacity violations are saved and sent to the enabling tools of WP4 in a feedback loop fashion. The appropriate enabling tool will use such information to modify traffic flow conditions in the network (first simulation block), so that the new arrival rates translate into no capacity violations once inputted into the microsimulation environment (second simulation block).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

It is expected, given the size of the network around the LLs and the complexity of the microsimulation performed within Aimsun, that a full simulation run (encompassing, for example, a few hours of operations) will be computationally heavy. As such, the simulation architecture as presented in this report is to be intended to be effective in an offline setting. This approach is clearly in contrast with scenarios that rely on online capabilities, such as disruption management. For such scenarios (assuming the disruption occurs in the network and not in the focal point of the LL), two preliminary options are proposed:

1. focus solely on the traffic flow around the LL (first simulation block) and use the enabling tools of WP4 to re-route flows. Assuming the disruption takes the form of a partial of complete blockage of a subset of nodes and edges in the network (e.g., accident on the road or electrical failure along the railway line), this disruption can be easily modelled in our graph representation as nodes can be removed and edges can be assigned a null capacity. Hence, the enabling tools can re-route traffic flows along the disrupted network. The downside of such approach is that, while it provides contingency solutions for traffic flow around the LL, it is myopic with respect to what happens in the LL itself as the second simulation block is not considered;
2. keep both simulation blocks but use simplified surrogate models in order to speed up the computation. In essence, this second approach relies on the original simulation architecture, but the simulation environment is simplified to overcome computational issues. The downside, in this case, is that the level of accuracy of the simulation, while capturing the inter-dependencies between the two blocks, is lower due to the presence of the surrogate models. In this context, an option to be explored is pattern recognition or machine learning here based on data from a series of full off-line simulations, and to use the derived pattern/ML-model as part of the simplified surrogate model.

A schematic of the simulation architecture, with the two simulation blocks and the relationship with WP4 and WP6 is shown in Figure 3.

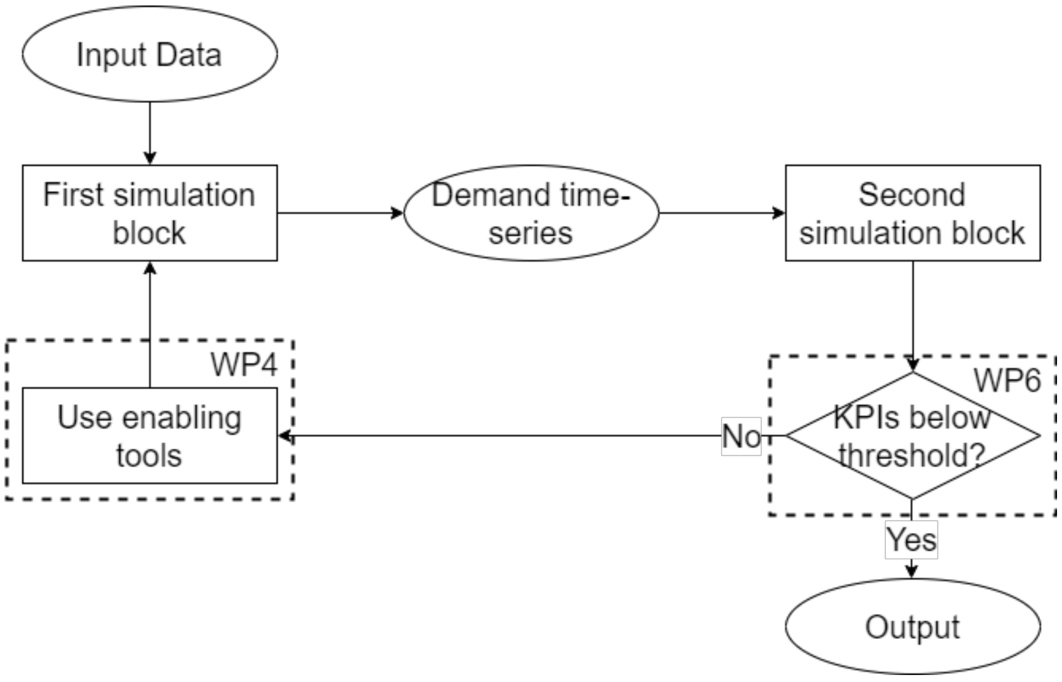


Figure 3: Flowchart depicting the devised simulation architecture

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Note that, in the figure, we assume that all KPIs should be below a pre-defined threshold. This is generally true when it comes to KPIs related to capacity, but some other KPIs might need to be above a pre-determined threshold. A more general definition would entail to check if KPIs are ***within the intended bounds***.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

5 Connection to PMA and MTME

The simulation architecture should support and cover four levels of abstraction as it was originally specified in the project proposal. This section provides an overview of such levels of abstraction and how the simulation architecture supports them.

1. **Verification of Polycentric Multimodal Architecture (PMA) on all levels, from top to bottom**. This is accomplished by the own nature of the simulation environment (especially the first simulation block), that allows the modelling of a *polycentric* and *multimodal* environment. In fact, the intended graph representation can be sub-divided into centres in Governance Areas that can interact and communicate with each other, while different graphs can be defined for every mode of interest. In this context, a special focus is put on those nodes and edges that allow transition from one mode to the other. In accordance to Tasks 3.1, 3.2, and 3.3, the context and requirement view is accounted for to align the simulation with the relevant use cases and needed requirements. The component view is also crucial, as assumptions on information to be collected and required interactions between services to enhance multimodality and collaboration must be modelled iteratively as defined in WP3.
2. **Assessing requirements defined in WP3**. This abstraction level stems directly from the previous one, as the simulation should support the polycentric aspects of the ecosystem, i.e., how the trade-off between the needs of different stakeholders and different performance targets should be solved. This is achieved by keeping track of the KPIs of interest as defined by WP6 to ensure that performance targets are achieved. Safety and security mechanisms needed in the ecosystem are also automatically accounted for as they are directly mapped through some of the aforementioned KPIs.
3. **Verification of enabling tools from WP4 by providing a real-time simulation environment**. The enabling tools defined in WP4 are inter-twined with the simulation environment as shown in Figure 3. The verification of such tools is carried out by analysing if their activation brings the system back to a "safe" condition as defined by the KPIs of interest. As it concerns a real-time environment, preliminary work highlighted the computational complexity associated to the geographical extension and infrastructure complexity of the two LLs. As such, the simulation architecture as defined in this report is considered to perform best in an offline setting. Two modifications to allow a (quasi) real-time simulation environment have been proposed
4. **Simulation for the use case verification**. The proposed simulation architecture is designed to allow verification of the proposed use cases. In particular, for disruptions or non-nominal conditions in the network, the underlying graph representation of first simulation block can be modified accordingly to mimic the specific use case. The same applies to the second simulation block if not-nominal conditions are to be modelled in HIP or MXP.

The aforementioned four levels of abstraction are strictly related to the Multimodal Traffic Management Ecosystem (MTME) architecture and stakeholders involved as described in D3.1, and, in addition, support and enhance the MTME as a management system. In the rest of this section, we elaborate how the simulation architecture (and simulation as a tool) supports the MTME and related stakeholders. We use as baseline Figure 4 from D3.1.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

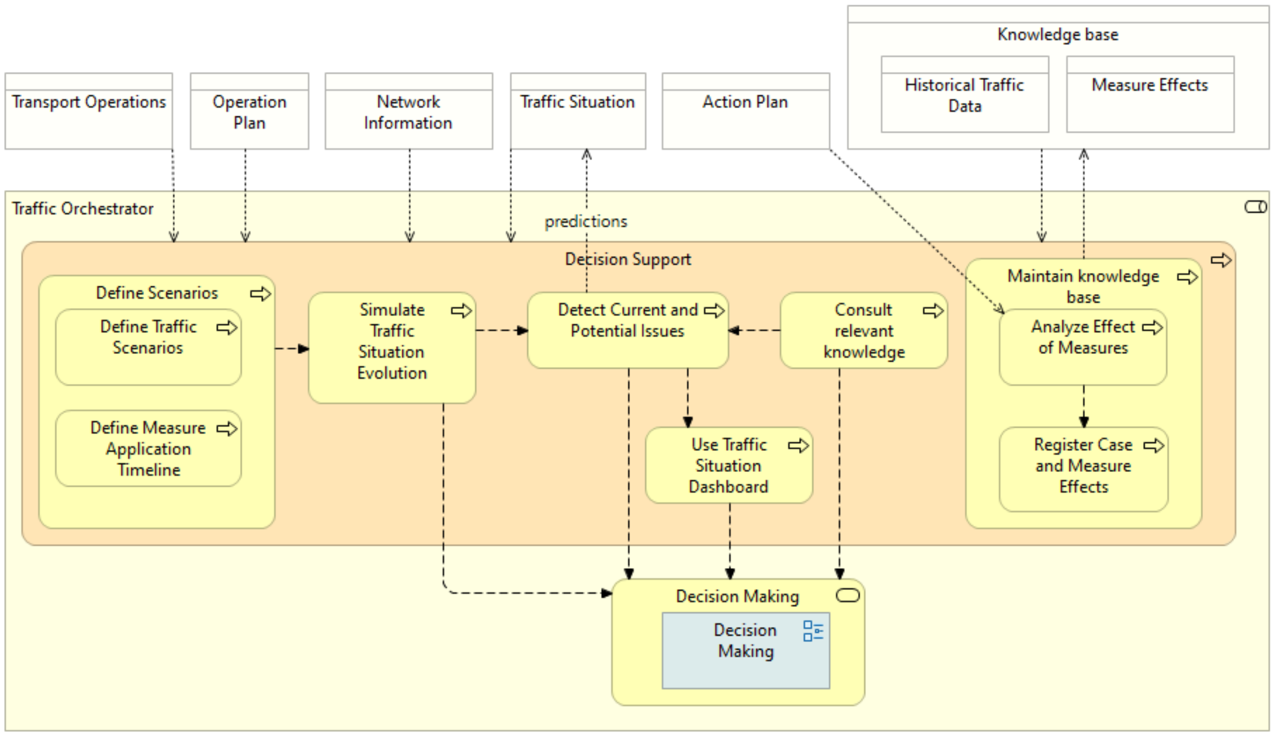


Figure 4: Overview of the Decision Support process as defined in D3.1

Our simulation architecture supports the MTME and the Traffic Orchestrator, in particular in the definition of the scenarios (left-most block in the figure), as simulation can be directly adapted to the needed scenario. For example, a disruption hindering a Transport Service Provider in the delivery of services (e.g., interruption of train connections due to system failure) is easily introduced in the simulation as our network representation allows to "disconnect" (remove) those edges defining the disrupted part of the transport system. In addition, the detection of current and potential issues (another block in the figure) can be exploited via stand-alone simulation (detection of current issues) or combining simulation with a prediction mechanism (detection of potential future issues). In an MTME context, the second approach is more powerful as it features a proactive nature and might call for traffic orchestration interventions capable of preventing issues from happening. This is opposed to the first approach that is reactive in nature and takes countermeasure actions once an issue has already occurred. Note that our simulation architecture can be directly exploited to export real-time KPIs in a traffic situation dashboard fashion. This step is of paramount importance for the Traffic Orchestrator, so that the effect of the orchestration can be easily visualized and interpreted. Additionally, some orchestration decision can be made on the basis of historical data and a knowledge-based set of rules, without necessarily involving the enabling tools defined in WP4 ("Maintain knowledge base" block in the figure). Being able to visualize the status of the traffic in a dashboard-like setting will facilitate such decisions. Additionally, while ownership and control of the dashboard should still be attributed to the Traffic Orchestrator, the dashboard itself can be shared across stakeholders, e.g., with a Transport Service Provider or a Fleet Operator, to enhance communication and traffic management.

Another strong connection between the simulation architecture and the PMA, is that our simulation framework can easily be adapted geographically to encompass different centres (or zones/Governance

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Areas) that can be changed dynamically according to the scenario under scrutiny. In our current setting, we are already considering different centres once we transition from the first simulation to the second simulation block of our architecture. Inheriting the terminology of Governance Area (GA) from other deliverables, i.e., different geographical regions regulated by different stakeholders and/or authorities, we show the division into GA for HIP (Figure 5) and MXP (Figure 6).

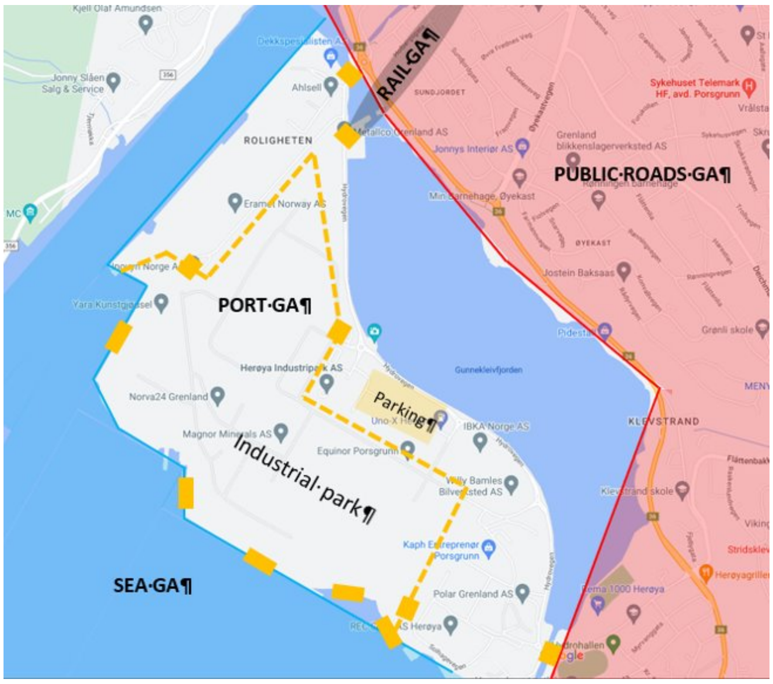


Figure 5: Division into GAs for HIP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

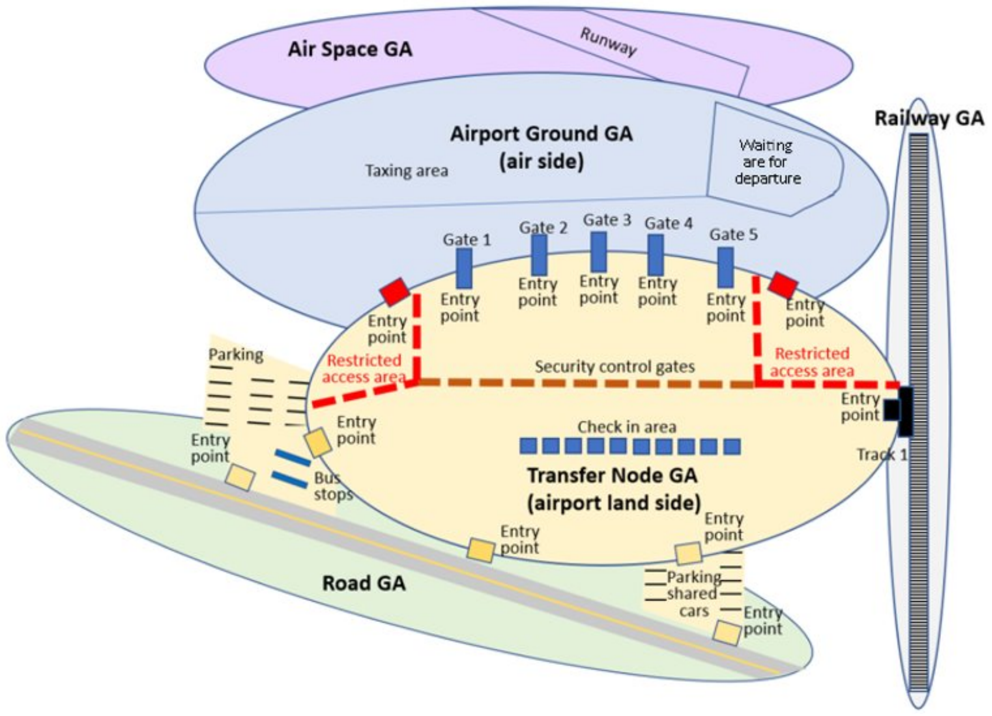


Figure 6: Division into GAs for MXP

Our macrosimulations, in both LLs, supports access to the (air)port by focusing on traffic flow management related to the public road GA, rail GA, and sea GA for HIP (note that rail and sea GAs have not been defined and modelled yet with accuracy in this version of the simulation architecture). Conversely, simulation supports traffic flow management for MXP that relates to road and rail GAs. The second simulation step focuses instead on the port GA (HIP) and airport landside GA (MXP).

Given that this division already embraces a PMA, the PMA paradigm can be further expanded in case of necessity. For example, in both LLs the road network (which is considered as a single GA for now) can be split into several centres to allow for more decentralized and local control of traffic and monitoring of KPIs. Our simulation framework and our graph representation of the multimodal network naturally steers towards a polycentric representation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

6 Simulation Architecture Specifications

In the rest of this section, details of the simulation architecture that are more specific for each LL are provided. In this deliverable, more detailed specifications are provided with respect to the Italian LL, compared to the Norwegian LL, as the involved partners have decided to partially work in sequence. More tailored specifications will be devised as part of D5.2 (Simulation) for both LLs.

6.1 HIP

In the context of the Norwegian LL, the first simulation block will focus on the road network around HIP. A catchment radius of around 80 km will be employed to provide a good balance between accuracy and computational complexity. There are plans to have a railway station to serve, in future times, the industrial port as well. While a final choice has not been made, the introduction of a (simple) railway network with a station inside HIP is possible and easy to implement. The interface between the first and second layer is characterized by the entrance to HIP, where traffic flow demand from the first block is disaggregated into single trucks that need to enter the premises of the industrial park to deliver freight. To simulate truck arrivals, arrivals patterns following historical demand will be generated. The observed demand inside HIP is relatively uniform and consistent over the year, so the assumption of a standard daily average traffic flow can be taken. However, different accumulation of arrivals inside the day will be tested to validate the robustness of the simulation architecture. The points of origin of trucks accessing HIP, is currently not recorded. To remedy this lack of data, demand generation points of interest (POIs) will be selected within the specified catchment area. Currently, the main selected POIs are the terminals of Grenland Harbour that service freight (Skien and Brevik), as well as the adjacent port of Larvik. Further POIs may be included as development of D5.2 (Simulator) progresses. The considered geographical layout around HIP is depicted in Figure 7.

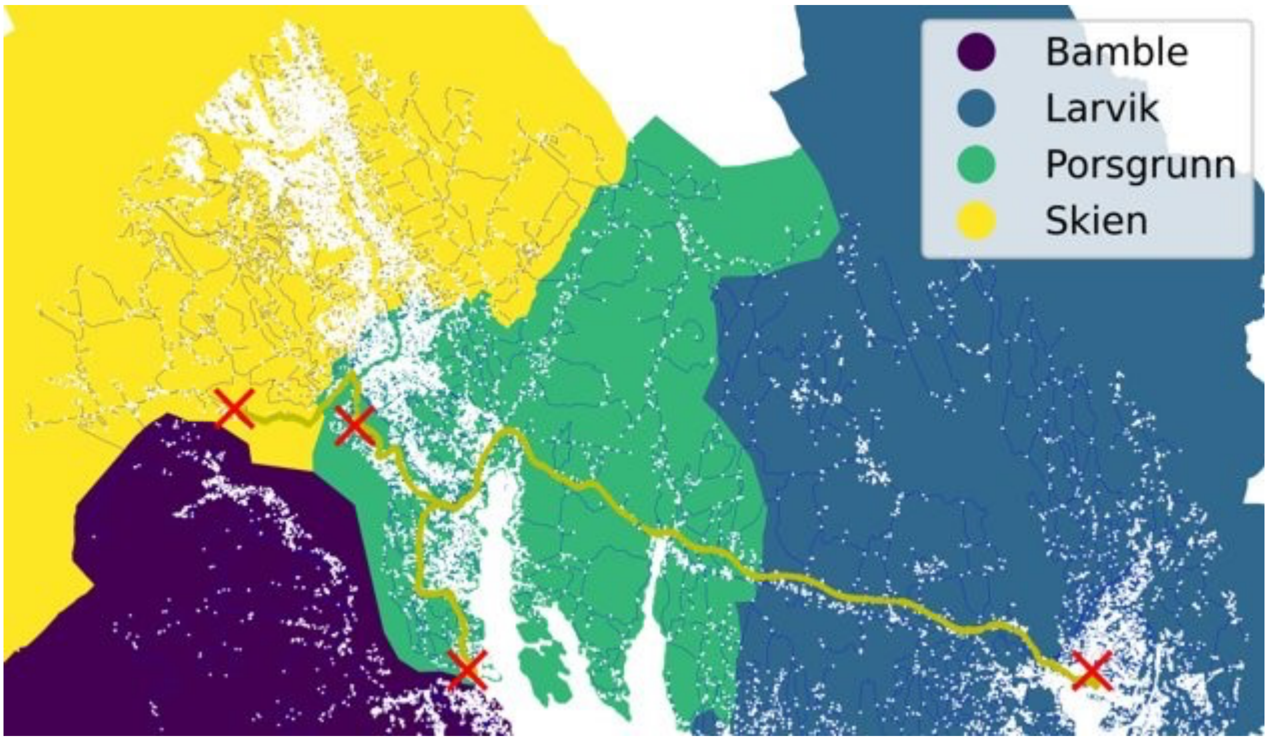
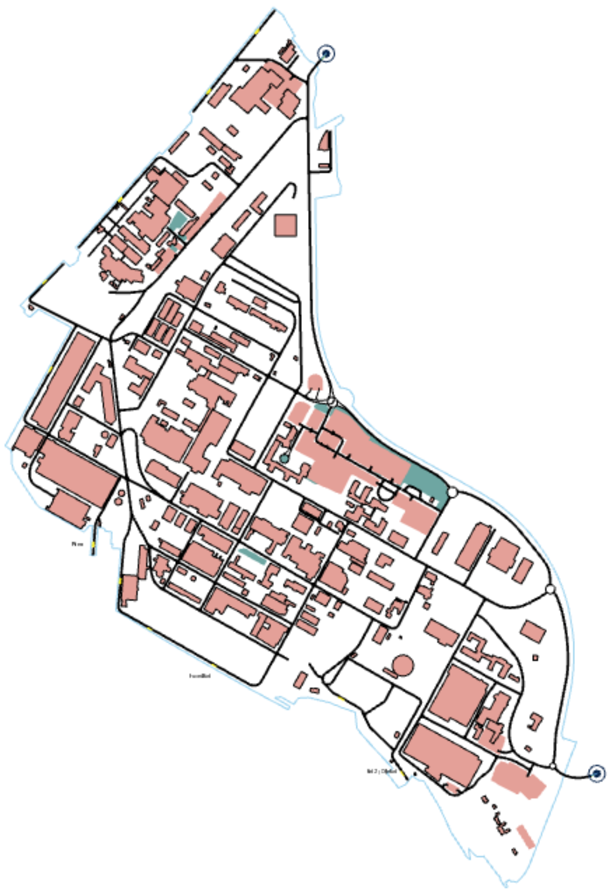


Figure 7 Area around HIP, divided in zones

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

To regulate, service, and dynamically adjust just-in-time access inside HIP, the use of a specialized API will be deployed, containing real-time data regarding traffic conditions of the examined network. Specialized requests concerning truck-specific information can be requested though the API. Such information relates to arrival and departure time from POI, and path selected, including service roads only accessible by trucks. In conjunction with tools determining access to the port, as developed in D4.1, exact scheduling and arrival rates inside HIP will be determined.

The second simulation step will focus on internal operations in HIP, where traffic demand as defined by the first simulation block is disaggregated into single truck arrivals. Dynamics of each truck are modelled according to the Aimsun microscopic specifications inside HIP. Each truck has a destination target (e.g., a yard) where cargo needs to be offloaded and, depending on the specific trucker, might or might not require chaperoning services by a CAV inside the port. Dynamics of CAVs are also modelled within the Aimsun environment. Finally, dynamics of barges docking at the industrial park to load cargo will also be modelled microscopically in the second simulation block. While barges define an "external" traffic flow with respect to the premises of HIP and, as such, might belong to the first simulation block according to our previous definition, the reason why they are assigned to the second simulation block is two-fold. First, barge dynamics do not resort on a fixed and well-defined infrastructure (as the road or railway network). While barge dynamics are not completely unregulated, especially in proximity to ports where merging points are defined, applying the same graph structure



to model them in a macroscopic fashion is not possible. Second, after discussions with the Norwegian

Figure 8: Internal layout of HIP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

port authority, it was assessed that HIP can handle simultaneously no more than 6/7 barges, hence making a microsimulation (barge-specific) perspective more meaningful than a macrosimulation one.

The internal layout of HIP that lays the foundations for the microsimulation is depicted in Figure 8, with building and warehouses in light red and internal roads in black. The entry point to HIP from the road network is located in the top-right part of the figure (black dot), while docks for barges are located on the left side of the perimeter. Readers are also referred to Figure 5 for a better visual description of the different GAs.

In terms of input data, the following inputs are needed to properly simulate the Norwegian LL:

- average traffic flow conditions per day and time of the day. A subset of significant days could be considered, for example analysing a summer and a winter day, together with a “good weather” and a “bad weather” day;
- expected arrival times of trucks needing to deliver cargo for the day of interest;
- quantity and type of cargo of each truck, to assess priorities in case of perishable or time-sensitive cargo;
- expected arrival and departure times of barges, to ensure synchronization with arrival of trucks
- number and characteristics (e.g., average operational speed) of CAVs

6.2 MXP

In the context of the Italian LL, the first simulation block will focus on the road and railway network serving MXP. As the catchment area is quite vast, and since MXP is located north-west with respect to the city of Milan (its main feeder), it was decided to consider a latitude-longitude box where Milan is located in the bottom-right corner, while MXP is in the top-left corner. Using such box, the road and railway networks were generated by means of networkX. They are shown in Figure 9.

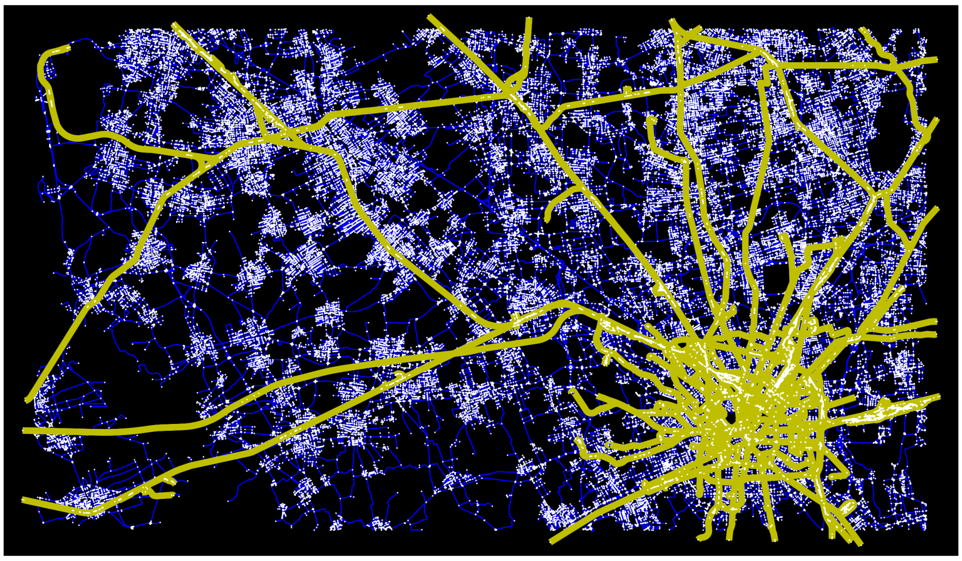


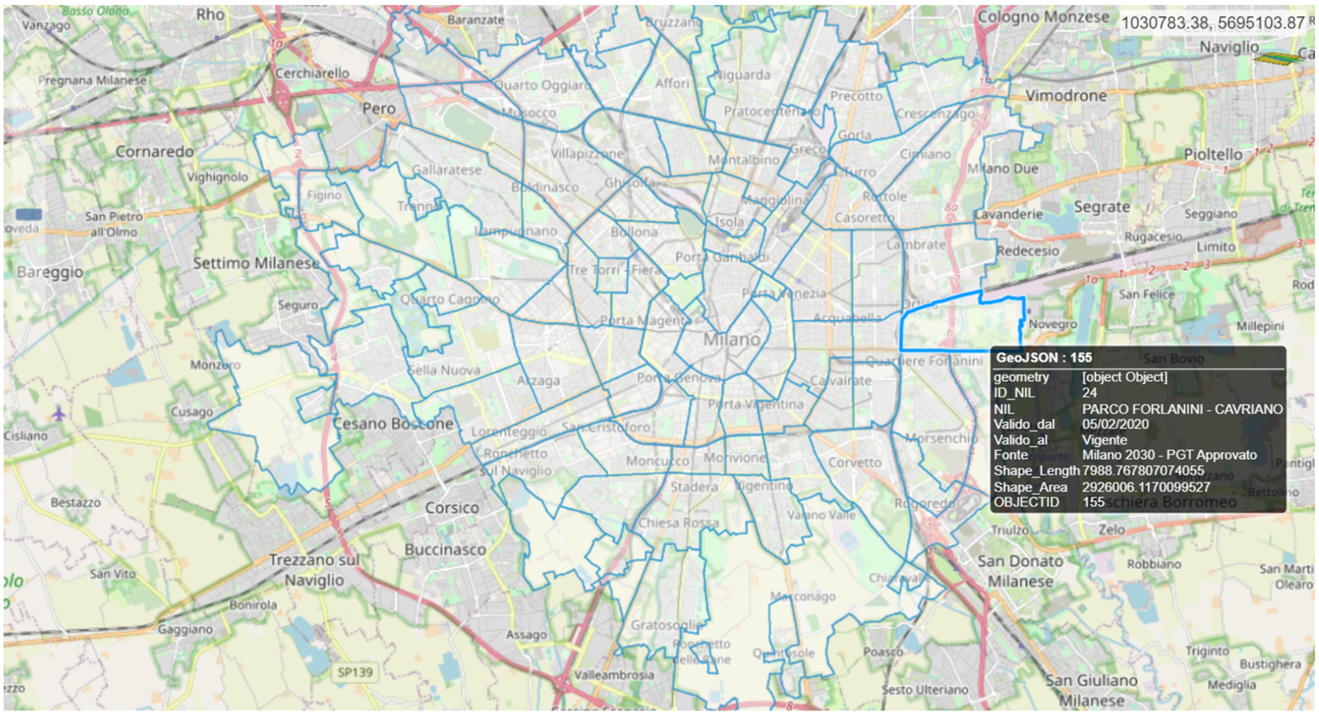
Figure 9: Road and railway network in the region of interest for the Italian LL as obtained from networkX (unfiltered)

The road network is depicted in blue, while the railway network is depicted in yellow. Colours refer to edges of the underlying graph. In both networks, nodes are represented with white dots. Milan is

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

easily recognizable in the bottom-right corner due to the density of the railway network there. Conversely, MXP is in the top-left corner, served by a single train line (Malpensa Express) and two main road arteries. Preliminary analyses highlighted that the current road network might be too dense for the needs of the first simulation block. This is explicit if we consider the density of the white dots (nodes), most of which represent urban areas that could be clustered to obtain a smaller (in terms of number of nodes, not geographically), yet still representative network. Even after experimenting with node aggregation techniques to reduce the size of the network, the resulting underlying graph was deemed too dense and containing unnecessary pieces of information given the expected outcome of the first simulation block. As such, a different approach was chosen that is explained as follows: Passenger demand headed to MXP is divided into two sets: (i) demand originating from the municipality of Milan, and (ii) demand originating somewhere else. Historical data can be leveraged to determine the percentages of passengers belonging to either category. For demand not originating from Milan, an aggregated value will be provided per simulation time-step, irrespective of the actual point of origin or mode.

The reasoning behind this assumption is that multi-modal options are scarcer for passengers originating from smaller towns. For demand originating from Milan, historical data on modal split will be used to infer percentages of passengers arriving at the airport using the road network or the railway network. For the railway network, only the Malpensa Express train network is considered, as it is the dedicated line connecting the city of Milan with MXP. As schedules are publicly available, defining departure times from each station and arrival times at the airport terminal is trivial. For the road network, we decided to simplify the problem at hand by discretising Milan into 88 neighbourhoods or Nuclei d'Identita' Locale (NILs, Local Identity Nuclei in English) as defined by the municipality of Milan. A representation of the 88 NILs defining Milan is provided in Figure 10 below.



GeoJSON : 155	
geometry	[object Object]
ID_NIL	24
NIL	PARCO FORLANINI - CAVRIANO
Valido_dal	05/02/2020
Valido_al	Vigente
Fonte	Milano 2030 - PGT Approvato
Shape_Length	7988.767807074055
Shape_Area	2926006.1170099527
OBJECTID	155

Figure 10: 88 NILs dividing the municipality of Milan

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

As our goal is to simulate arrivals at MXP, the idea is to move backwards as follows. Given a certain flight schedule for the planning horizon of interest (e.g., a full day of operations) and the expected number of passengers per flight, using historical data we can assess how many passengers will stem from outside Milan or from Milan. For the latter case, we can further assess how many of them will be using the Malpensa Express or use a car to get to MXP. For each passenger, we can assign a preferred arrival time at MXP that is based on, again, historical data or stated preferences. For example, passengers tend to arrive at an airport, on average, 1.5 hours before departure for an intra-continental flight and 2.5 hours before departure for an inter-continental one. Given a preferred arrival time, the problem is now to back trace each passenger to their origin and assign a proper departure time that is then consistent with the expected arrival time. In our simulation, lacking higher-quality data, we assume the probability that a passenger originates from a specific NIL is proportional to the population count of such NIL. Using publicly available data from the municipality of Milan, we were able to assign a spawn probability to each NIL following this approach. The population count per NIL referring to year 2018 (the most recent year available, and hence the year we used to compute the spawn probabilities) is shown in Figure 11.

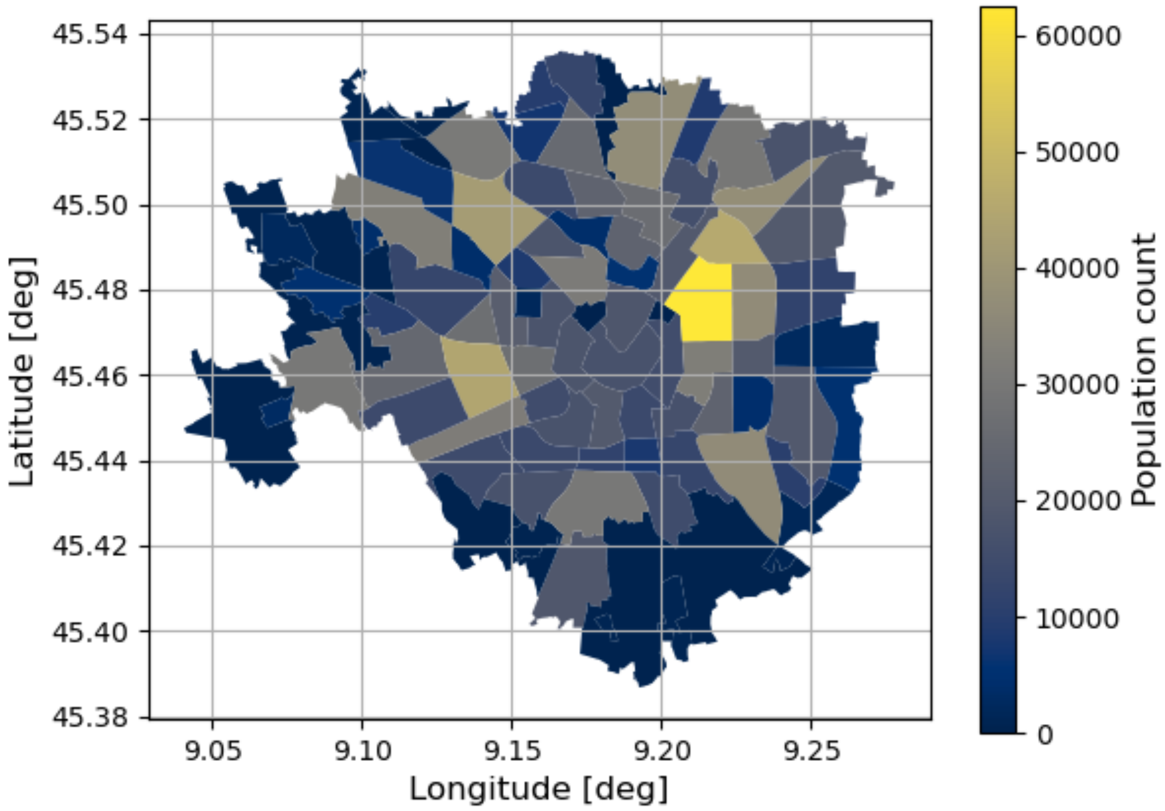


Figure 11: Population count in the 88 NILs.

For example, the probability that a passenger spawns from the "Buenos Aires – Porta Venezia" NIL (yellow-most, and hence most populated NIL in the figure according to the chosen colormap) is 4.4%. When generating a new passenger and assigning it to a specific NIL according to the aforementioned probabilities, we randomly generate an actual origin, defined as a (latitude, longitude) tuple inside

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

the NIL for that passenger. Given the current architecture, we consider all points inside the NIL as equally likely to generate demand. This is an approximation, as in each NIL population density is not constant and infrastructure constraints prevent regions inside each NIL to generate any demand. Notwithstanding, this approximation can be relaxed in a future version of the simulation provided more accurate geospatial information.

Once a passenger is generated, a set of additional features are associated to such passenger randomly. Recalling that the original feature is the associated flight and preferred arrival time at MXP, additional features can be:

- Number of passengers. With a slight abuse of notation, the term "passenger" defines a set of single passengers traveling as a whole. For example, in case of a family travelling towards MXP, the associated passenger count could be 4;
- Number of checked-in luggage. This feature defines a different behaviour in the second simulation block, as this passenger will need to visit the baggage drop-off kiosk before security checkpoints in case of checked-in luggage;
- Chosen mode: car or public transport. If public transport is selected, an additional choice is made between Malpensa Express (train) or bus, based again on historical modal splits

Finally, given a passenger, a preferred route is computed using the **HERE RoutingAPI** (HereAPI, 2022) and HERE Public Transit API via Python script. The routing API provides routing information better suited for private vehicles, while the Public Transit API, is updated with schedules and frequencies of the Milan Metropolitan Area transit. Schedules of specific modes that directly interact with MXP, are also embedded within tools developed as part of D4.1. As the preferred mode can be stated in the query, each travel request can reflect the travelling preference of the passenger. Each travel request is defined by a sequence of (latitude, longitude) tuples and overall travelling time. The travel request can be either computed using real-time traffic conditions or historical ones, and hence can capture the congestion level of the traffic network. Because of this, given an initial estimated departure time, the computed arrival time at MXP might fall outside the defined interval around the preferred arrival time (e.g., preferred arrival time at 9.00 AM with a flexibility interval of ± 20 minutes). If so, the departure time is updated in an iterative fashion until the final computed arrival time falls inside the specified interval. An example of such routine is depicted in Figure 12.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

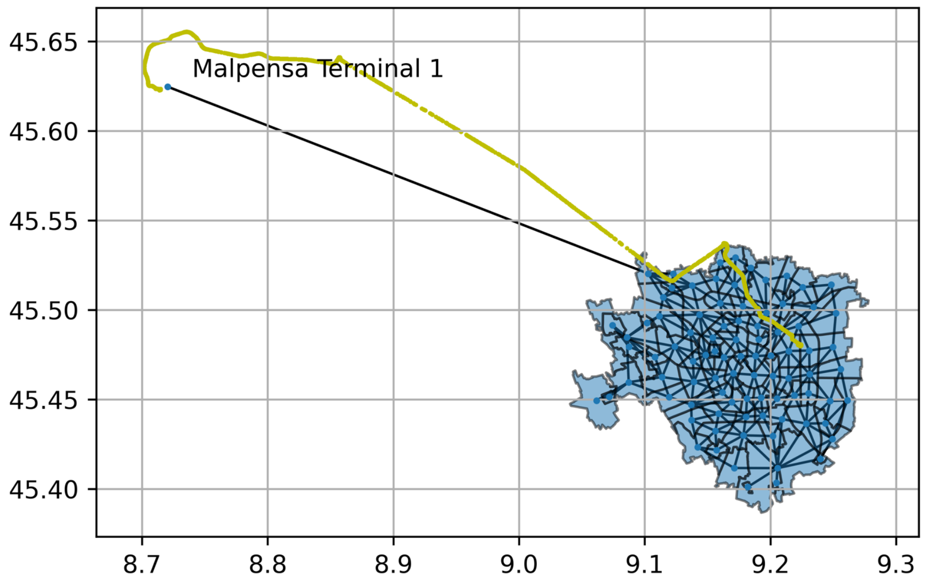


Figure 12: Example of car trajectory from a randomly generated point in Milan to MXP

In Figure 12, blue dots represent the centroids of the 88 NILs, with an additional dot in the upper-right corner representing MXP (Terminal 1). Black edges represent connections between nodes in the graph representation of the network. The yellow sequence of dots, on the other hand, represents the output of the API query for a passenger request travelling by car from the eastern side of Milan. The trip encompasses a first leg inside the city pointing towards the A8 highway (Milan-Varese), that is the main road artery connecting Milan with MXP. The equivalent trip highlighted using Google Maps is shown in Figure 13.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

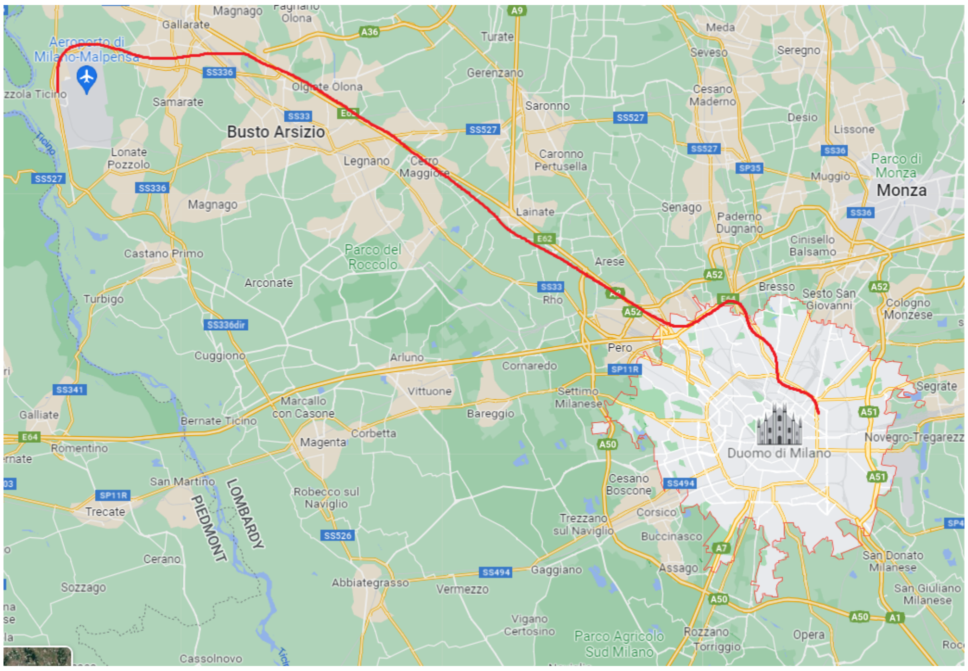


Figure 13: Visualisation of an actual car trip (with a focus on the highway portion of the trip) using the road network as background

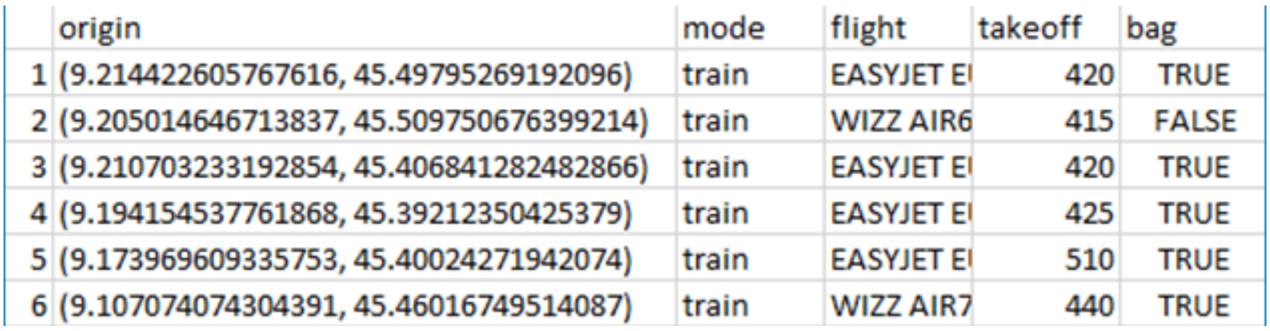
For the aforementioned reason, we expect trip requests using the road network to spatially differ, according to their specific origin, within the premises of Milan, but to all converge at the merging point of the A8 highway.

To extend this specific example to the whole simulation architecture layer, given a specific flight schedule at MXP, the full passenger population is translated into arrivals at MXP that depend on the intended flight and hence preferred arrival time. Such arrivals, for passengers stemming from Milan, are in turn propagated back into specific origins and departure times that are computed accessing real-time or historical traffic flow conditions to get an accurate estimate of the travelling time. Given the multimodal nature of the transportation network under scrutiny, the chosen mode of transport per passenger is also based on historical data, and could be made part of the decision-making process of the final version of the simulator (i.e., mode preferences could be influenced for traffic smoothing or for disrupted events).

As it concerns the interaction between the first and second simulation blocks, the output of the first simulation block for the Italian LL is a set of arriving passengers per time-window, which we defined as a 30-minute time interval in our current implementation. This set is divided into subsets depending whether passengers originate from Milan or not, and depending on the chosen mode of transport. Passengers arriving at a specific time-window are associated to flights roughly departing 2 hours later, as the average arrival time we impose is 2 hours prior to departure. As it concerns the actual arrival time within the time-window, a choice can be made. It can be the actual arrival time as computed by the API, or an ad-hoc distribution could be used to sample arrivals. To provide the reader with a more explicit description of the output of the first simulation block, a sample of such output is reported in tabular form in Figure 14 (arrivals by train) and Figure 15 (arrivals by car). The first table provides an excerpt of expected passengers arriving to MXP from Milan using the Malpensa

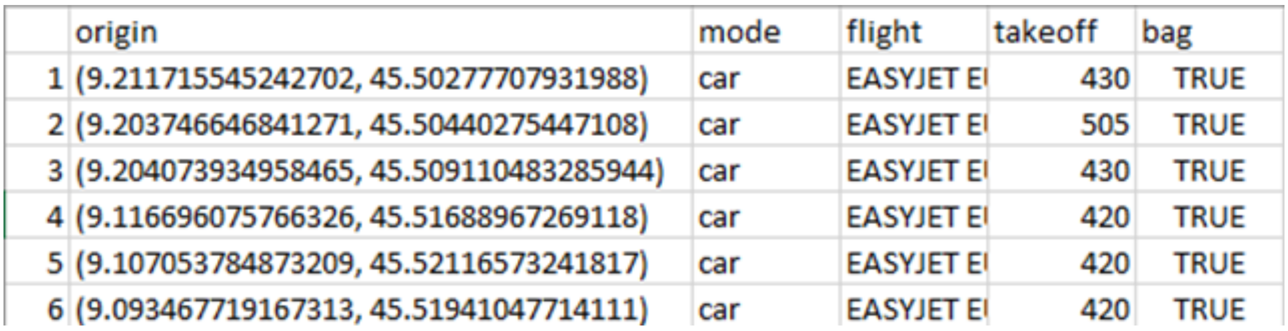
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Express between 5.00 AM and 5.30 AM, while the second one provides an excerpt of expected passengers arriving to MXP from Milan using a car in the same time-interval.



	origin	mode	flight	takeoff	bag
1	(9.214422605767616, 45.49795269192096)	train	EASYJET E	420	TRUE
2	(9.205014646713837, 45.509750676399214)	train	WIZZ AIR6	415	FALSE
3	(9.210703233192854, 45.406841282482866)	train	EASYJET E	420	TRUE
4	(9.194154537761868, 45.39212350425379)	train	EASYJET E	425	TRUE
5	(9.173969609335753, 45.40024271942074)	train	EASYJET E	510	TRUE
6	(9.107074074304391, 45.46016749514087)	train	WIZZ AIR7	440	TRUE

Figure 14: Example of output of the macrosimulation step for the Italian LL (passenger flow arriving at MXP by train in a specific time-window)



	origin	mode	flight	takeoff	bag
1	(9.211715545242702, 45.50277707931988)	car	EASYJET E	430	TRUE
2	(9.203746646841271, 45.50440275447108)	car	EASYJET E	505	TRUE
3	(9.204073934958465, 45.509110483285944)	car	EASYJET E	430	TRUE
4	(9.116696075766326, 45.51688967269118)	car	EASYJET E	420	TRUE
5	(9.107053784873209, 45.52116573241817)	car	EASYJET E	420	TRUE
6	(9.093467719167313, 45.51941047714111)	car	EASYJET E	420	TRUE

Figure 15 Example of output of the macrosimulation step for the Italian LL (passenger flow arriving at MXP by car in a specific time-window)

In both tables, each passenger is uniquely defined by their origin position, the mode, the intended flight, the take-off time (in minutes), and whether checked-in luggage is carried or not. Note that, in its current and preliminary form, not all features defining a passenger are included in the algorithm. In addition, as previously introduced, all flights depart between 6.30 AM and 9.00 AM, consistently with arrivals at MXP between 5.00 AM and 5.30 AM (so that we can also model "early" and "late" arrivals).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

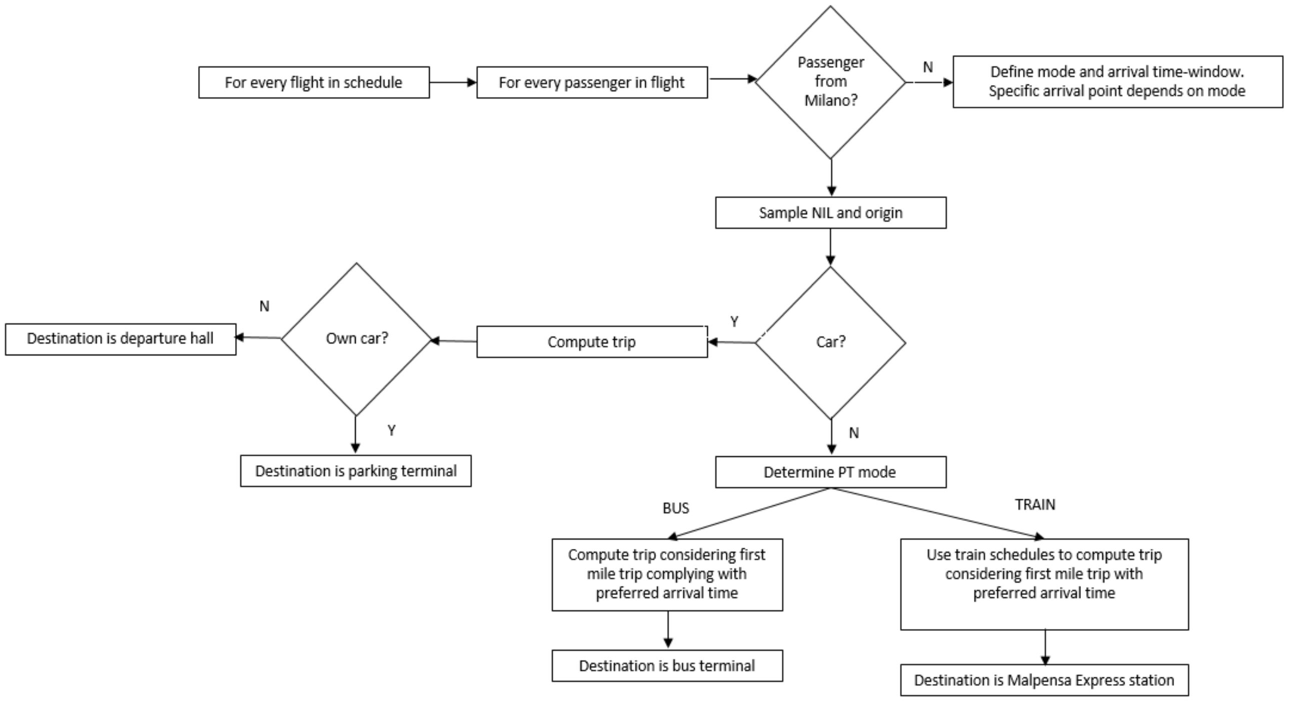

For every flight in schedule
For every passenger in flight
Passenger from Milano?
N
Define mode and arrival time-window. Specific arrival point depends on mode
Sample NIL and origin
Car?
Y
Compute trip
Own car?
N
Destination is departure hall
Y
Destination is parking terminal
N
Determine PT mode
BUS
TRAIN
Compute trip considering first mile trip complying with preferred arrival time
Destination is bus terminal
Use train schedules to compute trip considering first mile trip with preferred arrival time
Destination is Malpensa Express station

Figure 16: Flowchart depicting how passengers who booked a ticket for a specific flight are back-propagated to a randomly generated origin and are assigned a (multi)modal trip to MXP accounting for the preferred arrival time at MXP

The flowchart summarizing the algorithm that defines passenger traffic arrivals per time-window is defined in Figure 16. Additionally, in Figure 17, a visualisation of different trips generated from the north-eastern region of Milan is provided. The “long-haul” portion of trip for all users is the Malpensa Express (red lines). It can be noted a bifurcation that represents the two origin stations (Cadorna on the left and Centrale on the right), as the API allocates the departure station according to the origin of each user. The first mile trip (from the origin to the departure station), is computed by the API according to a set of pre-defined preferences (e.g., minimisation of travelling time), and might entail multiple modes, such as a combination of walking and bus, walking and subway, etc. The overall time duration of the trip should comply with the preferred arrival time-window of the user, and all users arriving (per mode) at MXP in a specific time-window, as previously introduced, define an arrival rate as shown in Figure 14 or Figure 15. This leads us, within our simulation architecture framework, to the second simulation block.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author’s view and the Agency is not responsible for any use that may be made of the information it contains.

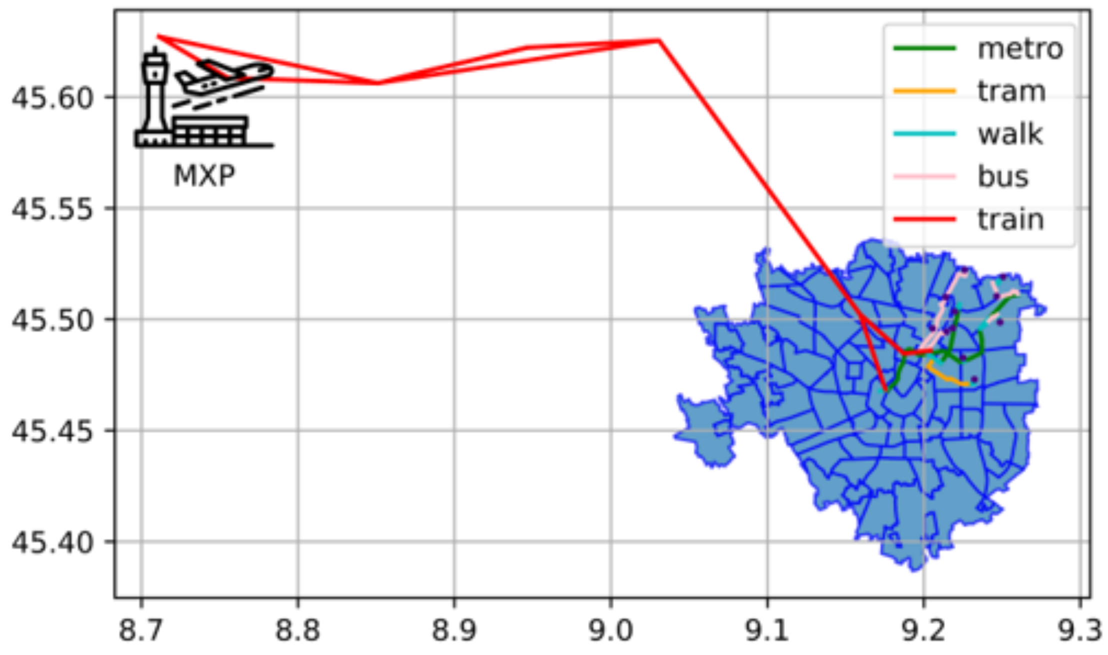


Figure 17: Example of computation of multimodal trip using a combination of public transport (first mile) and Malpensa Express to reach MXP

The second simulation block will focus on internal operations in MXP, simulating at the microscale level passengers from the moment they enter the terminal to the moment they board the intended aircraft. As such, each single entity (which can be a single passenger, or multiple passengers traveling as a group as defined in the first simulation block) is assigned a final goal, i.e., the specified gate where the aircraft will be docked. Depending on the nature of the passenger, the route leading to the gate will be different according to whether baggage drop is necessary or not (while security checkpoints are mandatory for all passengers). While passenger operations until the gate are deemed integral part of the project, it is still to be assessed whether boarding operations and taxiing operation of aircraft along runways will be part of the microsimulation. The layout used as the foundation for the second simulation block of the Italian LL is depicted in Figure 18. Note that, in the current setting, the airside (green area) is considered. The focus on the microsimulation is on internal operations at the terminal (left part of the figure), which will be further expanded in terms of simulation complexity in deliverable D5.2.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

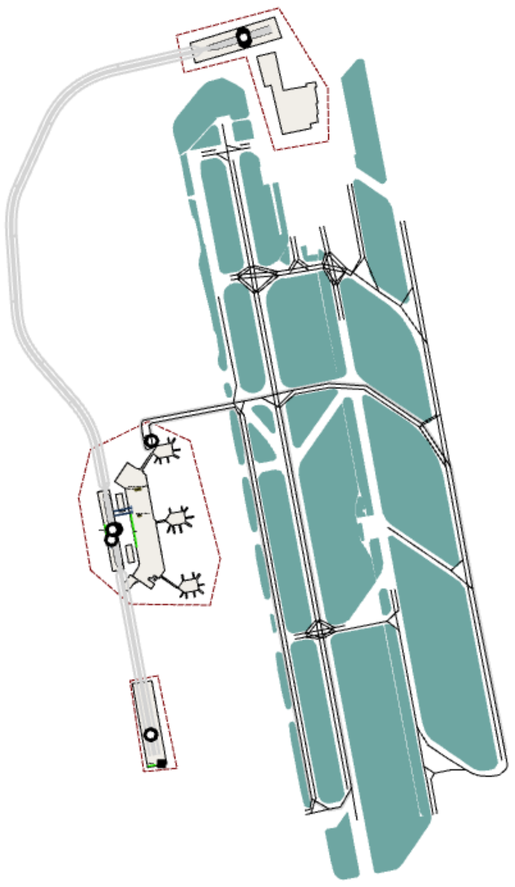


Figure 18: Internal layout of MXP

In terms of input data, the following inputs are needed to properly simulate the Italian LL:

- average traffic flow conditions per day and time of the day. A subset of significant days could be considered, for example analysing a summer and a winter day, together with a "good weather" and a "bad weather" day;
- average modal split between car, train, and bus for passengers headed to MXP;
- average origin geographical split for passengers headed to MXP;
- flight departure time for each passenger headed to MXP. This info defines, on average, when to expect such passengers at the airport;
- number and capacity (per hour, for example) of active security checkpoints

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

7 Aimsun Simulation Data Types

As it concerns the second simulation block, the Aimsun simulation environment requires the definition of several data types for a proper microsimulation. Given the preliminary work carried out until now, the data types identified in Table 3 have been defined and implemented. Note that they are divided into four main categories: agent, edge, node, and access. Agent data types are specific for each agent in the simulation. Edge and node data define the environment where agents move. Note that the modelling approach (graph-based representation) is the same as in the first simulation block, although the granularity level is different as single agents move in the graph with additional features and capabilities. Finally, access data define restrictions related to some specific access points.

Table 3: Microsimulation data types

Data	Format	Description	LL	
			Norway	Italy
Agent's data				
Agent Type	Vehicle/Pedestrian/Train/Truck/Plane/Vessel	Define the nature of the agent. CAV can be considered as vehicles.	x	x
Agent ID	INTEGER	Unique identifier that is attributed to the agent.	x	x
Agent position	FLOAT	Agent's current position. X-Y values in meters.	x	x
Agent speed	FLOAT	Agent's current speed. Value in m/s.	x	x
Agent travelled distance	FLOAT	Current distance travelled by agent. Value in meters.	x	x
Agent travelled time	FLOAT	Time that the agent already spent travelling.	x	x
Agent arrival time	FLOAT	The time when the agent arrived at the destination.	x	x
Agent destination	FLOAT	The destination that the agent wants to reach.	x	x
Agent planned arrival time	FLOAT	The planned time to reach the destination.	x	x
Agent priority	INTEGER	Priority assigned to a specific agent. This is not mandatory data, used only in certain situations.	x	x

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Edge data				
Traffic flow	INTEGER	Current traffic flow that is travelling through a specific edge. Value in agents/hour.	x	x
Traffic speed limit	FLOAT	Maximum allowed speed in a traffic edge.	x	x
Flow Capacity	INTEGER	Maximum admitted traffic flow in a specific edge. Value in agents/hour.	x	x
Allowed agent	INTEGER	Specification of the allowed agent type.	x	x
Node data				
Node Current Value	INTEGER	Current number of agents inside a node.	x	x
Node Capacity	INTEGER	Maximum allowed number of agents within a node.	x	x
Allowed agent	INTEGER	Specification of the allowed agent type.	x	x
Access data				
Access restriction	-	Restriction rules to the access points.	x	x
Access Waiting Time	FLOAT	Waiting time required to pass through a specific access point.	x	x

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

8 Conclusions

The goal of this deliverable was to describe the simulation architecture supporting the Polycentric Multimodal Architecture (PMA), a key pillar of ORCHESTRA. A two-block approach has been identified for both Living Labs (LLs) as a promising architecture, where each block employs microsimulation as modelling tool, yet with different nuances. The goal of the first block is to simulate arrival patterns to the (air)port of interest, i.e., the core of each LL. For the Norwegian case, that is Herøya Industrial Park (HIP), while for the Italian case, that is Malpensa airport (MXP). The current approach relies on a graph representation of the multimodal network, that allows for easy integration of different modes and definition of multiple centres in each Governance Area (GA) of interest, hence allowing application of the PMA paradigm as defined in WP3. In the current setting, arrival patterns at the (air)port are computed starting with preferred arrival times related to the delivery date of cargo (HIP) or the intended departure time of a flight (MXP), and backpropagated to origin points, departure times, and trajectories across the network computed via an Automated Programming Interface (API). We would like to point out that this approach does not play the role of a travel planner, but rather a simulation of how arrival patterns affect the network upstream in terms of usage, preferred modes, etc. Current potential issues that might arise once the first simulation step is deployed for full-scale scenarios have also been identified. For example, in the current approach the behaviour of each network user is computed calling the API. While a single request is not time-consuming, dealing with a large number of network users with become computationally challenging. A possible remedy is to use a large set of off-line simulated trajectories, and then use Machine Learning (ML) techniques to infer trajectories and arrival patterns without the need of explicitly computing them. We believe this approach embodies even better the main role of *traffic flow monitoring and management* the first simulation step needs to play.

The second simulation step of our architecture uses the output of the first step to perform an accurate microsimulation focused on dynamics of single network users (trucks and CAVs for HIP, passengers for MXP) inside each LL. The need for such accuracy is paramount, unlike the first simulation step, to ensure the defined Key Performance Indicators (KPIs) defined in WP6 are properly satisfied. As a matter of fact, the inter-connectivity between WPs 4, 5, and 6 lies in whether the simulation capabilities of WP5 can identify operational issues as suggested by WP6, so that the enabling tools developed in WP4 can provide the traffic orchestrator with countermeasure actions. This inter-connectivity, ideally, calls for pre-emptive actions rather than reactive ones, and hence the prediction capabilities of the simulation still need to be fully assessed from a computational perspective. As such, the adoption of surrogate models that give up some precision but are better (and in a quicker fashion) capable of predicting future conflicts or inefficiencies can be a suitable option to improve the traffic orchestration capabilities of ORCHESTRA. Additionally, the role of disruptions is crucial. By definition, as a disruption is generally hard to predict, the combination of simulation (WP5) and enabling tools (WP4) should be able to provide the traffic orchestrator with a revised solutions to bring the system back to a quasi-normal setting (i.e., the system should be resilient) within a limited time. Similarly, to what previously stated, the real-time application of the simulation architecture in combination with the enabling tools is still to be tested with a computational campaign to fully understand the enablers and inhibitors of the approach and apply proper simplifications or revisions.

In terms of future work, we already covered that a computational campaign tailored to the needs and inputs of the two LLs is paramount to improve the simulation architecture. In addition, while polycentricity is already exploited with the division between the first and second simulation blocks, that address different GAs, future work is needed to further define multiple centres upstream of the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

(air)ports for a more decentralized approach that better fits with the ORCHESTRA paradigm and should provide computational benefits that could limit the aforementioned expected computational issues.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

9 References

HereAPI. (2022, October 14). *Here API.* Retrieved from Here API website: https://www.here.com/platform/map-data

Aimsun. (2022, October 14). *Aimsun*. Retrieved from Aimsun: https://www.aimsun.com/about-aimsun/

Ejercito, P. M., Gayle, K., Nebrija, E., Feria, R., & Figueroa, L. L. (2017). Traffic simulation software review. *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA).*

de Souza, F., Verbas, O., & Auld, J. (2019). Mesoscopic Traffic Flow Model for Agent-Based Simulation. *Procedia Computer Science, 151*, 858-863.

Plakolb, S., Jager, G., Hofer, C., & Fullsack, M. (2019). Mesoscopic Urban-Traffic Simulation Based on Mobility Behavior to Calculate NOx Emissions Caused by Private Motorized Transport. *Atmosphere*.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

Members of the ORCHESTRA consortium

ITS Norway	**ITS Norway** c/o Tekna – Teknisk-naturvitenskapelig forening Postboks 2312 Solli NO-0201 Oslo Norway its-norway.no	**Project Coordinator:** Runar Søråsen runar.sorasen@its-norway.no **Dissemination Manager:** Jenny Simonsen jenny.simonsen@its-norway.no
SINTEF	**SINTEF AS** NO-7465 Trondheim Norway www.sintef.com	**Technical Manager:** Marit Natvig Marit.K.Natvig@sintef.no
TU Delft	**Technische Universiteit Delft** Stevinweg 1 2628 CN Delft The Netherlands	**Evaluation Manager:** Alexei Sharpanskykh O.A.Sharpanskykh@tudelft.nl
ROSAS	**ROSAS Center Fribourg** Passage de Cardinal 13B Halle bleue CH-1700 Fribourg Switzerland info@rosas.center	**Contact:** Lucio Truaisch lucio.truaisch@rosas.center
CERTX	CERTX AG Route de l'Ancienne Papeterie 106 CH-1723 Marly Switzerland	**Contact:** Samuel Rieder samuel.rieder@certx.com
IKEM	**Institut Fur Klimaschutz Energie Und Mobilitat-Recht, Okonomie Und Politik Ev (IKEM)** Magazinstraße 15-16 10179 Berlin Germany	**Data Manager / Legal, Privacy and Policy Issues Officer (LEPPI) officer:** Anne Freiberger anne.freiberger@ikem.de
IOTA FOUNDATION	**IOTA Foundation** c/o Nextland Straßburger Straße 55 10405 Berlin Germany	**Contact:** Michele Nati michele@iota.org Siddhant Ghongadi siddhant.ghongadi@iota.org
SEA Milan Airports	**Societa Per Azioni Esercizi Aeroportuali Sea (SEA)** Presso Aeroporto Linate 20090 Segrate MI Italy	**Contact:** Massimo Corradi massimo.corradi@seamilano.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

deepblue consulting & research	**Deep Blue Srl** Via Ennio Quirino Visconti, 8 00193 Roma Italy	**Innovation Manager:** Alessandra Tedeschi alessandra.tedeschi@dblue.it
RÉPUBLIQUE FRANÇAISE Cerema	**Cerema** 25 Avenue François Mitterrand 69500 Bron France	**Contact:** Sylvain Belloche Sylvain.Belloche@cerema.fr
FS TECHNOLOGY GRUPPO FERROVIE DELLO STATO ITALIANE	**FSTechnology SpA** Piazza della Croce Rossa, 1 00161 Roma RM Italy	**Contact:** Jessica Bonanno jessica.bonanno@it.ey.com
Information Sharing Company	**Information Sharing Company Srl (ISC)** Via di Tor Pagnotta, 94/95 00143 Roma Italy	**Contact:** Antonio Martino a.martino@gruppoisc.com
APPLIED AUTONOMY	**Applied Autonomy AS** Kirkegardsveien 45 NO-3601 Kongsberg Norway	**Contact:** Olav Madland olav.madland@appliedautonomy.no
HERØYA INDUSTRIPARK www.heroya-industripark.no	**Herøya Industripark AS** Hydrovegen 55 NO-3936 Porsgrunn Norway	**Contact:** Tone Rabe tone.rabe@hipark.no
enav	**ENAV SpA** Via Salaria, 716 00138 Roma Italy	**Contact:** Patrizia Criscuolo Patrizia.Criscuolo@technosky.it
Statens vegvesen Norwegian Public Roads Administration	**Statens vegvesen** Rynsengfaret 6A NO-0667 Oslo Norway	**Contact:** Elisabeth Skuggevik elisabeth.skuggevik@vegvesen.no

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953618. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.