

Preparing the ground for **AUTO**nomous Multimodal **SUP**ply Chains

Grant Agreement Number: 101147468



D2.3 DSS FOR OPERATIONAL AUTOMATION IN MULTIMODAL HUBS (v1)

Work Package	WP2
Deliverable Submission Date	29.11.2025
Dissemination Level	PU
Lead Beneficiary	KNT

Document Control Sheet

Acronym	AUTOSUP
Grant Agreement No.	101147468
Work Package No & Name	WP2 - Digital Twin of the Autonomous Multimodal Supply Chain
Deliverable No., Name	D2.3 DSS for operational automation in multimodal hubs (v1)
Version	1.0
Contractual due date	29.11.2025
Actual submission Date	29.11.2025
Dissemination Level	PU
Deliverable Type	OTHER
Lead Beneficiary	KNT
Contributor(s)	INLE, FRONTIER, VLTN, TUM, ARS, PoAB
EU Agency	European Climate, Infrastructure and Environment Executive Agency
Program	Horizon Europe Framework Programme (HORIZON)
Topic	HORIZON-CL5-2023-D6-01-07
Rights	AUTOSUP Consortium



Document Revision History

Version	Date	Author / Contributor	Changes
v0.1	2025/04/25	Stephane Kundig	ToC
v0.2	2025/10/30	Aristea Zafeiropoulou	Report ready for peer-review
v1.0	2025/11/28	Aristea Zafeiropoulou	Final Version for submission

Authoring

Author / Contributor Name	Organisation (Acronym)
Aristea Zafeiropoulou	KNT
Stephane Kundig	KNT
Arkadiusz Drabicki	TUM
Kostas Zavitsas	VLTN
Thanos Karydis	INLE
Ilias Gkotsis	INLE
Konstantinos Christidis	FRONTIER

Quality Control

Version	Date	Name (Organisation)	Role & Scope
v0.1	2025/06/18	WP2 consortium	ToC Approval
v0.2	2025/11/17	Florian Krietsch	Peer Review

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

Copyright: © AUTOSUP Consortium, 2024-2027. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.



Executive Summary

This deliverable presents the work undertaken in the context of Task 2.3 related to the design and development of the AUTOSUP Decision Support System (DSS). The report outlines its requirements, conceptual architecture, and the components that underpin the DSS's operation.

The report begins by defining the functional and non-functional requirements derived from the analysis of the Living Lab (L-Hub) use cases, ensuring alignment with stakeholder needs and AUTOSUP's overarching objectives. It then introduces the conceptual architecture, which provides a system-level overview of the DSS and its integration within the broader AUTOSUP Digital Twin ecosystem.

The document further details the modelling and simulation components that form the analytical backbone of the DSS. These include the Interdependency Graph for representing physical and informational linkages across logistics infrastructures, the Digital Twin models for simulating transport and operational dynamics, the Information Flow models for automation of data exchange processes, the Network Optimisation algorithms, and the Cost-Benefit Analysis (CBA) model for assessing the economic viability and impact of the automation scenarios. Together, these components enable the DSS to execute and compare "what-if" scenarios, which, in turn, shall drive feasibility and investment analysis.

The report also presents the technical architecture and platform development steps, including the implementation of the orchestrator, authentication mechanisms, the Knowledge Graph, and the User Interface. These elements collectively support model execution, data management, and visualisation of simulation results through interactive dashboards and map-based visualisation interfaces.

The report concludes by highlighting the upcoming steps in the development of the DSS and its underlying components, focusing on continued model integration, refinement of data exchange mechanisms, and validation through the L-Hub use cases to ensure the system's operational relevance and readiness for deployment in real-world logistics environments.



Table of Contents

1	Introduction	8
1.1	Addressing the AUTOSUP Description of Action	8
1.2	Overview and Structure of the Document	10
2	DSS Requirements and Architecture	11
2.1	Objectives and Role in AUTOSUP	11
2.2	Requirements Definition	11
2.2.1	Analysis of the L-Hub Use Cases	12
2.2.2	Functional and Non-Functional Requirements.....	12
2.3	Conceptual Architecture	15
3	DSS Models and Simulation Components	18
3.1	Interdependency Graph	18
3.2	Digital Twin Models.....	19
3.3	Information Flows	21
3.4	Network Optimisation Algorithms	22
3.5	Cost-Benefit Analysis.....	23
4	DSS Platform Development.....	25
4.1	Technical Architecture and Components	25
4.1.1	Authorisation and Authentication	26
4.1.2	Knowledge Graph	27
4.1.3	Orchestrator	29
4.2	Model Integration and Execution Pipeline.....	32
4.2.1	Model Execution Pipelines	32
4.2.2	Model integration.....	34
4.3	User Interface.....	34
4.3.1	Interdependency Graph	35
4.3.2	Scenario Builder.....	36
4.3.3	Visualisation of Results.....	38
5	Conclusion and Future Work.....	40



Table of Figures

Figure 1. AUTOSUP Digital Twin Context Diagram	16
Figure 2. AUTOSUP Digital Twin Container Diagram	17
Figure 3. Interdependency Graph Pipeline	18
Figure 4. Example of PI Hub representation	23
Figure 5. CBA and DT Models interaction	24
Figure 6. AUTOSUP DSS Architecture	25
Figure 7. Keycloak Main Dashboard	27
Figure 8. The AUTOSUP Knowledge Graph	28
Figure 9. Graph initialisation process	30
Figure 10. Simulation creation process	31
Figure 11. Model Execution Pipeline (PoAB, Intervention 3.1)	33
Figure 12. Model Execution Pipeline (Trieste, Use Case 1)	33
Figure 13. Antwerp-Bruges Interdependency Graph exposed in the UI	35
Figure 14. Trieste Interdependency Graph exposed in the UI	36
Figure 15. Trieste - Use Case selection.....	36
Figure 16. Trieste - Scenario Definition of inputs.....	37
Figure 17. Antwerp-Bruges - Use Case selection.....	37
Figure 18. Antwerp-Bruges - Scenario Definition of inputs.....	38
Figure 19. Antwerp-Bruges - Example Dashboard of outputs.....	38
Figure 20. Trieste - Example Dashboard of outputs.....	39
Figure 21. Example Dashboard - Model Outputs	39

Table of Tables

Table 1-1: Glossary	7
Table 1-2: List of Acronyms	7
Table 2-1: L-Hub Use Cases	11
Table 2-2. Functional Requirements	13
Table 2-3. Non-Functional Requirements	14
Table 2-4. Mapping table between L-Hub needs and functional and non-functional requirements.....	14



Glossary & Acronyms

Table 1-1: Glossary

TERM	DEFINITION
Decision Support System	Information system that supports business or organisational decision-making activities.
Knowledge Graph	Structured representation of information that organises data as a network of entities (nodes) and their relationships (edges), enabling both humans and machines to interpret, query, and reason over complex domains.
Interdependency Graph¹	Representation of operational or causal dependencies (tasks/infrastructure/requirements) for execution, scheduling, or impact analysis.

Table 1-2: List of Acronyms

ACRONYM	DEFINITION
API	Application Programming Interface
BO	Backend Orchestrator
BPMN	Business Process Modelling Notation
CBA	Cost-Benefit Analysis
DT	Digital Twin
DSS	Decision Support System
KG	Knowledge Graph
KPI	Key Performance Indicator
LoA	Level of Automation
L-Hub	Living Hub
NPV	Net Present Value
PoAB	Port of Antwerp Bruges
PoT	Port of Trieste
Ro-Ro	Roll-on/Roll-off
SC	Supply Chain
T&L	Transport & Logistics
UC	Use Case
UI	User Interface
WP	Work Package

¹ In D2.3 the Interdependency Graph is depicted by the Knowledge Graph, so they are used within the text as a one common graph representation.



1 Introduction

This deliverable presents the outcomes of Task 2.3 *Decision Support through Simulation*, which focuses on the design and implementation of the Decision Support System (DSS) forming the core of the AUTOSUP Digital Twin (DT) framework. Task 2.3 directly contributes to Objective O2 of the project, with the ultimate purpose of empowering Transport & Logistics (T&L) stakeholders with an open, ready-to-use DSS and a customisable DT model of autonomous supply chains (SCs). The DSS enables stakeholders to implement and evaluate automated processes and solutions for logistics and multimodal freight transport by conducting simulation-based feasibility studies (Task 3.3) and assessing the benefits and drawbacks of different interventions, while accounting for sustainability, financial, and social impacts.

The DSS builds upon the DT models developed in Tasks 2.1 and 2.2 and integrates a Cost-Benefit Analysis (CBA) model (Task 3.1) to simulate and compare alternative automation scenarios. Through this integration, the system supports users in exploring “what-if” situations involving different levels of automation, operational configurations, and investment strategies. By orchestrating relationships between physical, digital, and organisational components, the DSS enables the comprehensive analysis of how automation and digitalisation influence the performance and resilience of multimodal freight transport operations.

Overall, the DSS serves as the main interface layer of the AUTOSUP Digital Twin ecosystem, providing stakeholders with simulation-based foresight on the benefits of operational automation in T&L. It facilitates performance benchmarking, and impact evaluation to support strategic decision-making regarding automation investments, thereby contributing to a more efficient, resilient, and sustainable European logistics landscape.

1.1 Addressing the AUTOSUP Description of Action

AUTOSUP GA Item/Requirement	Description of the item/requirement	Deliverable chapter(s)	Brief description
D2.3 DSS for operational automation in multimodal hubs (v1)	DSS tool integrating data, scenarios, and optimisation results, enabling the assessment of operational automation processes in the multimodal logistics hubs. Includes generalised DT model.		
Task 2.3: Decision Support through simulation	Develop a DSS to support feasibility analysis, compare and demonstrate through simulation the benefits of operational automation to current standard flows and operations. This task will utilise the DT models from T2.1 and T2.2, model and orchestrate the relationships between physical, digital, and operational elements captured in the DTs. The DSS will use the DT models and historical operational data to compare and demonstrate the benefits of operational automation in the studied areas. It will support a layered view of a. the physical network infrastructure, automation systems in place and freight flows, b. the Information Flows based on	Sections 3.2, 3.5, 4.2 and 4.3	The DT models are presented in section 3.2, while 3.5 presents the CBA. Their integration into the DSS is described in section 4.2, while section 4.3 presents the User Interface that exposes them to the end users.





	established collaborations and, c. the costs associated with operations given the selected LoA.		
ST2.3.1 Interdependency graph modelling	For each Use Case create the high-level representation of a network of interconnected logistics infrastructures and the transport network assets around the L-Hubs ports, including the connected automated systems. This will allow monitoring the performance of the network under different conditions (demand, bottlenecks, new infrastructures, control terminals, new automation, incidents affecting automated processes, critical cybersecurity issues at the hub and network level. Nodes in the graph will represent critical elements (digital, such as telematics control towers, servers, management systems or physical infrastructure) that might be impacted by the change of specific parameters (automation levels, downtime, physical disasters), and edges will represent dependencies between them. The graph will use the results of ST1.1.5 (SCDT ontology) to develop models representing the interconnectivity and interdependencies between assets and infrastructure. Develop risk matrices assessing disruption probability and potential impact severity and provide interactive visualizations of interdependency relationships. The interdependency graph will be the backbone of the DTs and will raise awareness about potentially indirect dependencies, enabling the simulation of freight flows and the calculation of performance and associated cost indicators.	Sections 3.1, 3.3	Section 3.1 presents the Interdependency Graph, while section 3.3 presents the Information Flows.
ST2.3.2 Develop network optimisation algorithms	Develop and integrate graph algorithms to optimise network- wide flows and logistics processes e.g., load balancing across infrastructure, assets, and terminals and to minimise transfer times during the implementation of multimodal scenarios. Integrate algorithms to streamline traffic flows for avoiding congestion and to improve scheduling across supply chain nodes,	Section 3.4	Section 3.4 presents the work undertaken in modelling.



	with the aim of increasing reliability. Use and integrate open-source AI/ML algorithms and solutions, adapt, and refine these based on the L-Hubs setup. Perform optimisation runs using DT simulations for validation.		
ST2.3.3 DSS Development	Develop the monitoring system, providing situational awareness and a real-time operational picture/map about the status of the transport network in the vicinity of AUTOSUP’s ports (including hinterland road corridors and inland waterways) in the form of an interactive map. It will also serve as a tool for simulating alternative automation scenarios and enable the simulation of cascading effects and mitigation strategies for planning SC resilience-strengthening measures. Data sources of the L-Hubs (ST2.1.1, ST2.2.1) and the CBA model (T3.1) will be integrated to allow the execution of simulations and feasibility analysis. The system will consist of a web application; visualisations will employ powerful geospatial analysis tools (e.g., kepler.gl, Mapbox).	Chapter 4	This chapter is dedicated to the DSS: the technical architecture, the integration of the models in the DSS, and the User Interface, where alternative automation scenarios shall be explored by the users.

1.2 Overview and Structure of the Document

This report is structured as follows:

- Chapter 2 is dedicated to the requirements behind the design and development of the DSS. It begins by laying out the requirements elicited through the analysis of the L-Hub Use Cases and continues by identifying the functional and non-functional requirements. Following that, it presents the conceptual architecture of the AUTOSUP Digital Twin Platform and the DSS.
- Chapter 3 introduces all the models developed in AUTOSUP, such as the DT models, the Information Flows and the Interdependency Graph, and the Cost-Benefit Analysis model.
- Chapter 4 focuses on the technical architecture of the DSS and the integration of the previously mentioned models.
- Finally, Chapter 5 concludes the report by offering the key takeaways and upcoming steps in the DSS development.



2 DSS Requirements and Architecture

This chapter is devoted to the analysis of the DSS requirements and role within the project. It also presents an overview of the DSS conceptual architecture.

2.1 Objectives and Role in AUTOSUP

In the AUTOSUP project, the Decision Support System (DSS) plays a central role as the main enabler for data-driven decision-making and strategic planning toward the automation and digitalisation of supply chains. Built around a customisable Digital Twin (DT) model of automated supply chains, the DSS empowers transport and logistics stakeholders (including shippers, terminal operators, policymakers, and transport mode operators) to design, simulate, and evaluate future autonomous logistics scenarios without disrupting ongoing operations. By integrating DT models, and Cost-Benefit Analysis models, the DSS allows users to assess the feasibility, efficiency, and sustainability of various automation strategies and Levels of Automation across hubs and transport routes. Through its simulation and visualisation capabilities, it demonstrates the benefits of operational automation compared to current standard flows, supports experimentation with multimodal freight networks, and visualises information and freight flows in dynamic, interactive dashboards. Ultimately, the DSS acts as the core analytical and decision-making engine of AUTOSUP, facilitating stakeholder collaboration, guiding investment decisions, and providing a scalable framework to support future autonomous supply chain developments beyond the project's scope.

2.2 Requirements Definition

Each L-Hub has a set of different Use Cases defined and presented previously in D1.3 *AUTOSUP Use Case Scenarios and KPIs*. These are summarised in *Table 2-1*, as follows:

Table 2-1: L-Hub Use Cases

L-Hub	Use Case number	Use Case name	Intervention	Intervention Description
Trieste	UC1	Automated maintenance and management of intermodal rail wagons	1.1	Predictive maintenance for rail wagons in the overall freight flow efficiency
			1.2	Assessment of impact from shifting to a closer maintenance workshop (Gorizia)
	UC2	Automated management of multimodal slots and last-mile routes	2.1	Automation technologies for multimodal slot management and route optimization
	UC3	Automated cross-border roll-on/roll-off (Ro-Ro) transshipment	3.1	Automated Ro-Ro transshipment with Digital Automated Coupler (DAC) and automated terminals
Antwerp-Bruges	UC1	Port automations coordination and environmental performance	1.1	Evaluating Cargo Hubs and shifting freight to night operations for congestion relief.





L-Hub	Use Case number	Use Case name	Intervention	Intervention Description
	UC2	Port-Shipper collaboration towards increased modal shift	2.1	Real-time ETA-based dynamic slot adjustment across terminals and distribution centers
	UC3	Port – Transport Mode Automation-systems integration	3.1	Automated trucking between MPET Term. and Medrepair Depot using autonomous vehicles
			3.2	Assessing MagRail feasibility as a high-speed intermodal solution

2.2.1 Analysis of the L-Hub Use Cases

In the context of Task 1.3 *Identification of Strategic Innovations and Solutions - AUTOSUP Use Case Scenarios Definition*, the two L-Hubs (PoAB, and Trieste) developed detailed Use Case scenarios for their respective Use Cases. These are aimed at providing a guide tool for the definition and overall description of the envisaged Use Cases of the AUTOSUP project, encompassing overall strategy, key objectives, gaps and requirements, available dataset, operational business models, criteria for automations selection, and KPIs. As part of the scenario definition, the processes that shall undergo automation were defined (AS-IS, TO-BE Process Models in BPMN diagrams), including the conditions that shall enable seamless multimodal freight transport. These are also included in the relevant deliverable *D1.3 AUTOSUP Use Case Scenarios and KPIs* and form the basis for the development of the Functional and Non-Functional requirements presented in the following section. Furthermore, several bilateral meetings with the L-Hubs were held to clarify any pending questions regarding the design of the DSS and the DT models.

Overall, the analysis of the L-Hub Use Cases led to the identification of several needs that the DSS shall address:

- Need 1. Each L-Hub Use Case shall translate into a “what-if” scenario in the DSS, allowing the users to experiment with different parameters of a scenario.
- Need 2. The DSS aims to support strategic decision-making for future investments; therefore, the end users are decision makers, not modellers. This implies that the User Interface shall be an easy-to-use and intuitive application.
- Need 3. The scenario definition leads, in essence, to the parameterisation of the necessary inputs for the DT models prior to their execution.
- Need 4. The DSS shall integrate the DT models and the CBA model, allowing them to run in a sequence (i.e., one after the other), before returning all their outputs to a scenario-dedicated dashboard.
- Need 5. Users shall be able to compare different “what-if” scenarios through the User Interface by comparing the model outputs to the different scenarios.
- Need 6. Finally, the entire infrastructure network of each L-Hub shall be depicted in a Knowledge Graph, highlighting the physical, but also the Information Flows between them.

These needs are revisited in the following section, indicating a mapping between them and the functional and non-functional requirements (Table 2-4).

2.2.2 Functional and Non-Functional Requirements





The functional requirements define the expected capabilities and behaviours of the AUTOSUP Decision Support System (Table 2-2). They have derived from the analysis of the L-Hub requirements and reflect the operational and decision-support needs identified across the L-Hub Use Cases. Specifically, the requirements specify how the DSS should enable decision makers to define and configure “what-if” scenarios, execute simulations through the Digital Twin and Cost Benefit Analysis models, and visualise and compare the resulting outputs through an intuitive User Interface. They also describe the representation of infrastructure and information interdependencies within the Knowledge Graph, ensuring that the structure and operation of each L-Hub are accurately modelled and presented. Collectively, these requirements ensure that the DSS meets the functional expectations of the L-Hubs and supports evidence-based assessment of strategic and operational scenarios.

Table 2-2. Functional Requirements

ID	Requirement Category	Requirement Description
FR1	User Interface	The system shall provide an intuitive, user-friendly interface designed for decision makers, allowing easy scenario creation, configuration, and comparison without requiring modelling expertise.
FR2	Scenario Definition	The system shall enable users to define simulation scenarios, which are derived from the analysis of the UC templates.
FR3		The system shall allow the comparison between different scenarios.
FR4	Model Library	The system shall integrate all AUTOSUP WP2 models as part of the system’s Model Library.
FR5		The system shall allow users to configure simulation parameters as part of the scenario definition.
FR6		The system shall enable sequential execution of the Digital Twin and Cost–Benefit Analysis models and automatically return consolidated outputs to the scenario dashboard.
FR7	Model Outputs & Visualisation	The system shall display simulation outputs through dashboards with charts, maps, and tables to enable decision-making and performance evaluation.
FR8		The system shall allow users to compare results across different scenarios using KPIs (e.g., delay, CO ₂ emissions).
FR9	Knowledge Graph	The system shall represent port components as nodes in the interdependency graph.
FR10		The system shall represent physical and informational interdependencies as directed edges between the nodes.
FR11		The system shall import the Knowledge Graph and expose its entities and relationships through the User Interface.

The non-functional requirements define the quality attributes and operational constraints of the DSS (Table 2-3). Derived in alignment with the L-Hub requirements, they ensure that the DSS performs reliably



and efficiently under realistic operational conditions. These requirements specify performance thresholds and security measures, as well as expectations for scalability to support multiple users, simulation scenarios, and large data volumes. Particular attention is given to usability and user experience, ensuring that decision makers can access and interpret model outputs effectively through clear and responsive visualisations. Together, these requirements complement the functional specifications presented previously.

Table 2-3. Non-Functional Requirements

ID	Requirement Category	Requirement Description
NFR1	Performance	All User Interface pages should load and render in under five seconds, for fifty concurrent users.
NFR2		The system shall support concurrent simulation executions by at least 5 users without performance degradation.
NFR3	Security	Access to the system shall be restricted to authenticated and authorised users only.
NFR4		The system shall enforce HTTPS encryption for in transit data.
NFR5	Scalability	The system shall support scaling to handle large volumes of graph entities, multiple simulation scenarios and concurrent users.
NFR6	Availability	The system shall maintain an uptime of at least 99% since its launch.
NFR7	Usability & User Experience	The system shall offer intuitive navigation and clear data visualisations, enabling users to interpret simulation results and metrics effectively.
NFR8	Extensibility	The system shall be extensible by project and third-party partners, allowing them to introduce new analyses.

The identified functional and non-functional requirements have been mapped against the initial L-Hub requirements to ensure full traceability and alignment with user needs. This mapping, presented in Table 2-4, demonstrates how each system capability directly addresses the operational and decision-support needs defined by the L-Hubs.

Table 2-4. Mapping table between L-Hub needs and functional and non-functional requirements

L-Hub Requirement	Description (Summary)	Mapped FR & NFR
Need 1	The DSS shall enable the creation and execution of configurable “what-if” scenarios.	FR2 – Scenario Definition, FR5 – Parameter Configuration, FR6 – Sequential Execution scenarios
Need 2	DSS shall be usable by non-technical users and support strategic decision-making.	FR1 – User Interface, FR7 – Model Outputs & Visualisation, NFR1 – Performance, NFR7 – Usability & UX





Need 3	Scenarios configure model inputs before simulation.	FR5 – Parameter Configuration, FR6 – Sequential Execution
Need 4	The system shall integrate models and automate their sequential execution, displaying results.	FR4 – Model Library, FR6 – Sequential Execution, FR7 – Model Outputs & Visualisation, NFR2 – Concurrent Executions, NFR5 – Scalability, NFR6 – Availability
Need 5	The system shall allow users to compare results across scenarios.	FR3 – Scenario Comparison, FR8 – Results Comparison, FR7 – Visualisation, NFR7 – Usability & UX
Need 6	DSS must include a Knowledge Graph representing physical and informational interdependencies.	FR9 – KG Nodes, FR10 – KG Edges, FR11 – KG Import & UI Exposure, NFR5 – Scalability

2.3 Conceptual Architecture

The conceptual architecture provides a high-level overview of the entire WP2 offering, representing the AUTOSUP Digital Twin Platform in its entirety, including the DSS and all integrated models. It highlights the main components, their responsibilities, and interactions with external actors. It focuses on what the system is and how it supports L-Hub operations and decision-making, without delving into implementation-specific details.

The C4 (Context, Containers, Components, and Classes) model has been adopted as the diagrammatic technique to structure this representation. It provides a clear and concise way to depict the different levels of abstraction in a system, from high-level context to low-level implementation details. The C4 diagram technique consists of four diagram types:

- Context Diagram: Illustrates the high-level system boundaries and its interactions with external entities, such as users, systems, and external services.
- Container Diagram: Focuses on the internal components or containers that make up the system and shows their relationships, interactions, and dependencies.
- Component Diagram: Breaks down individual containers into their respective components or modules, showing the internal structure and dependencies within each container.
- Class Diagram: Represents the detailed structure and relationships between classes within a component, showcasing the implementation details and associations.

For the purposes of this report, the top two levels—Context and Container diagrams—are presented. The Context diagram illustrates the Digital Twin and DSS in their environment and the interactions with users, data sources, and external services, while the Container diagram details the main system building blocks, including the DT models, Cost–Benefit Analysis, Knowledge Graph, and their interactions. The focus on the first two levels of C4 ensures that the conceptual architecture remains clear, accessible, and aligned with the needs of project partners and modellers, providing sufficient detail to understand system structure and communications while avoiding unnecessary technical complexity.



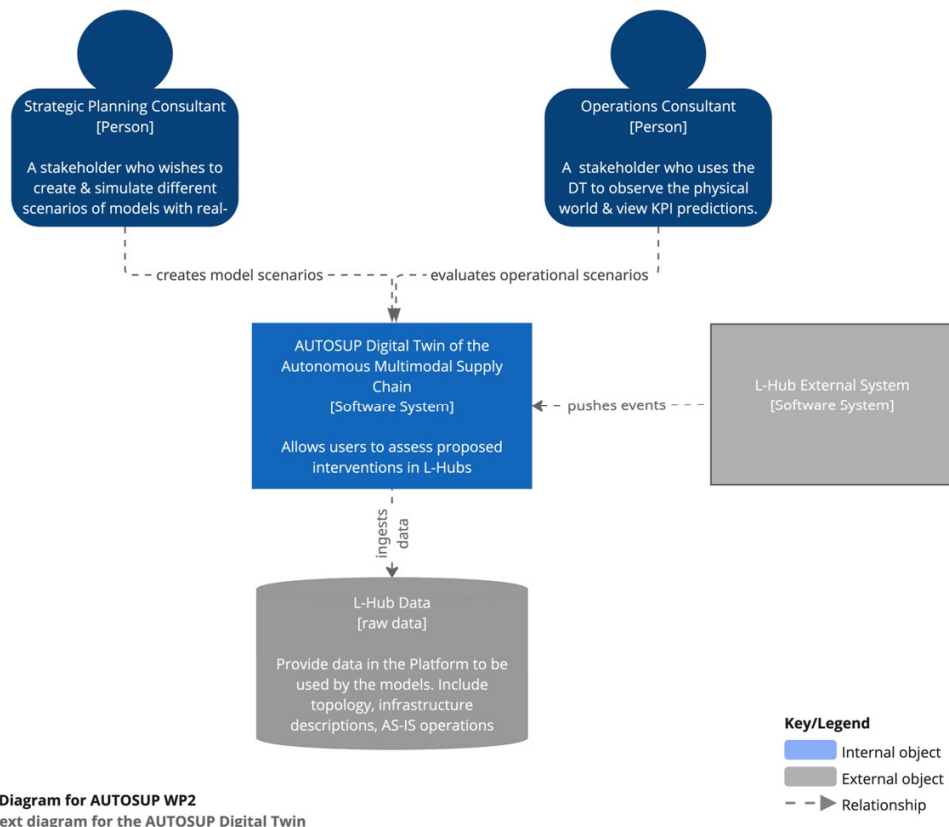


Figure 1. AUTOSUP Digital Twin Context Diagram

The first diagram developed is the Context Diagram, as displayed in Figure 1. The entire offering of WP2, that is the AUTOSUP Digital Twin Platform, is located at the centre of the diagram, linked to any other L-Hub system it may interact with, as well as L-Hub data. At the same time, it identifies the main user roles identified so far in the project:

- **Strategic Planning Consultant:** A stakeholder wishing to create and simulate different scenarios of models using real-life data for better decision-making. They explore different "what-if" questions through the relevant models. They do not need to have a technical or engineering background; they are rather domain experts in decision-making positions that use the DT to drive autonomy in operations and make strategic decisions about future investments.
- **Operations Consultant:** A stakeholder who wishes to observe the current state of things in the physical world and view KPI predictions. They are not expected to have a technical background; they make use of the DT to explore the logistics network at an operational level.



Container Diagram for AUTOSUP WP2
The container diagram for the AUTOSUP Digital Twin

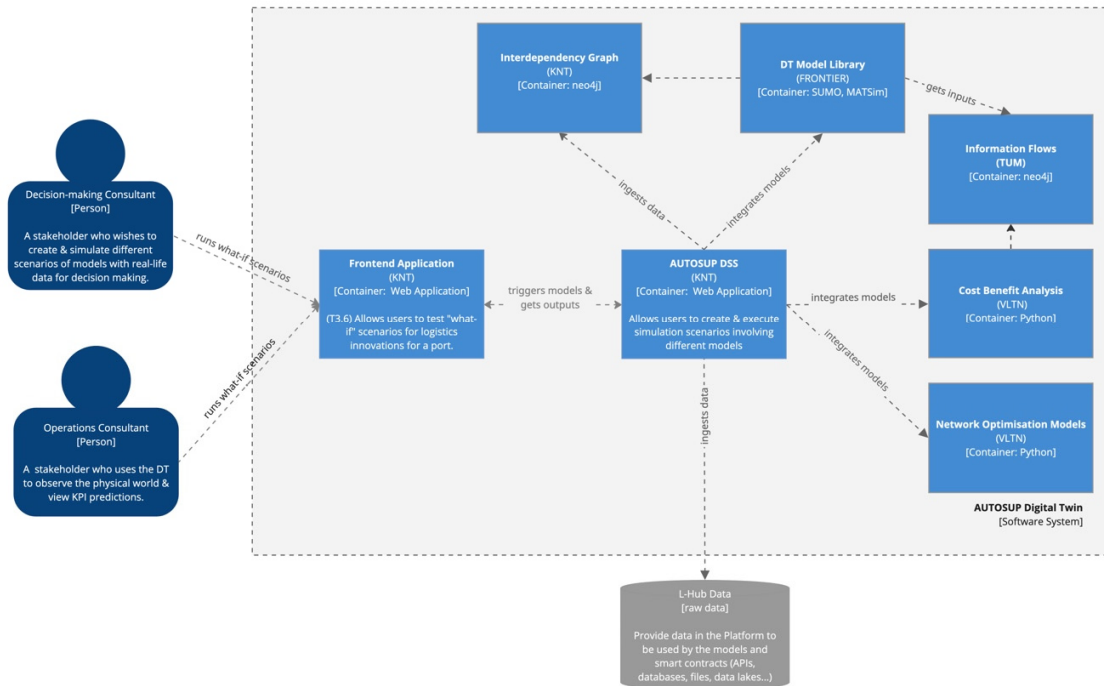


Figure 2. AUTOSUP Digital Twin Container Diagram

The second diagram – presented in Figure 2 - takes a step inside the WP2 offering, identifying all the main components and services developed in this Work Package and how they are expected to interact with each other. At the heart of the offering lies the DSS, which shall integrate all the technical components and services of AUTOSUP, which are: (a) the Interdependency Graph, (b) the Library of DT models, (c) the Information Flows, (d) the Network Optimisation Models, and (e) the Cost-Benefit Analysis (CBA) model. Following the integration of all the underlying components into the DSS, the solution shall be available to the defined user roles through a dedicated frontend application. The following chapter presents each of these components in more detail.



3 DSS Models and Simulation Components

This chapter presents the models and simulations developed within the Decision Support System (DSS) of the AUTOSUP project. The DSS combines structured data and analytical models to support decision-making at the L-Hub level, enabling the assessment of the efficiency and impact of innovative logistics solutions.

The chapter introduces the Knowledge / Interdependency Graph, which organises and links information on infrastructure, operations, and data exchanges. It then presents the Digital Twin models used to simulate logistics processes, followed by a description of the Information Flows that define data interactions within the system. The chapter concludes with the Cost-Benefit Analysis, which evaluates the impacts and effectiveness of alternative scenarios.

Together, these components provide the analytical framework of the DSS, supporting integrated modelling and evidence-based evaluation within AUTOSUP.

3.1 Interdependency Graph

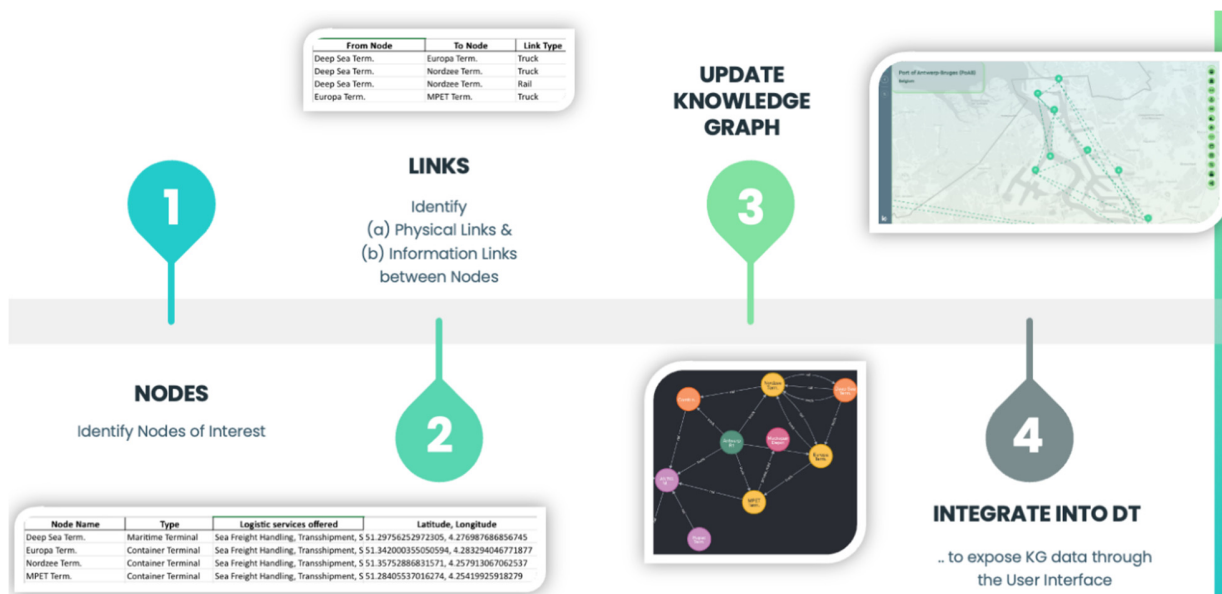


Figure 3. Interdependency Graph Pipeline

The development of the Knowledge Graph (KG) followed a structured workflow, with a particular emphasis on the generation and structuring of the L-Hub data in support of the project’s decision-support objectives. The entire process is reflected in Figure 3 and it can be summarised as follows:

Step 1. Node generation – Infrastructure elements

The initial dataset comprises of a network of nodes representing key logistics infrastructure assets of the L-Hubs (e.g., rail terminals and port terminals) and the physical connections between them (i.e., transport network). These were identified through (a) a thorough analysis of the use-cases documented in Deliverable *D1.3 AUTOSUP Use Case Scenarios and KPIs*; and (b) bilateral consultations with the L-Hubs to validate and refine the extracted information.



All identified infrastructure entities and linkages were recorded in a shared Excel workbook, which served as a central data repository and was made accessible to the project partners.

Step 2. Information-link generation – Flows and logical links

In parallel, the Technische Universität München (TUM) team had already drafted the Information Flows (IFs) in diagrammatic form (presented later in section 3.3). These flow diagrams were then captured in the same Excel workbook, though in a dedicated worksheet, thereby segregating the logical/information-link layer from the physical-infrastructure layer.

Step 3. Automation of Knowledge Graph instantiation

A script was developed to automate the reading of the Excel workbook and instantiate a new Knowledge Graph in Neo4j each time the script is executed. This automated workflow ensures that the KG is reproducible, up-to-date and traceable, avoiding manual error and ensuring that partner-provided updates in the Excel file are captured. This process is described in section 4.1.3.2.

Step 4. Integration into the Decision Support System (DSS)

Finally, the KG was integrated into the DSS, enabling the underlying data to be accessed via the user interface and visualised on a map. The details of this integration, including system architecture and UI visualisation, are presented in section 4.1.2.

3.2 Digital Twin Models

The Digital Twin (DT) modelling scope is closely linked with the effort to quantify the operational characteristics and environmental outcomes of different automation options relative to the current, i.e., “as-is” situation. To that end, the DT is expected to offer a number of quantitative outputs and key metrics to support the investment and application of the innovative solutions defined within the L-Hub Use Cases – listed in Table 2-1. This needs to be achieved with a high degree of reliability, in a manner that is comparable between the different Use Cases. Furthermore, the outputs of the DT models shall be used as inputs to the Cost-Benefit Analysis (CBA) model, to enable meaningful analytical outcomes regarding the economic feasibility of the innovations of the Use Cases.

With all that in mind, the key outputs of the DT models are expected to be of the following nature:

- Trip characteristics, such as duration, delays, and fuel/energy consumption.
- Environmental emissions, such as Greenhouse Gases (GHG) and other air pollutants [particulate matter (PMP). Nitrous Oxides, and the like].
- The ability to test performance under different automation and operational scenarios (e.g., other traffic, the volume of cargo / number of containers to be transported, and so on).

Digital Twins for transport are usually described as a set of nodes and links (edges), the level of detail of which categorises them as macroscopic, mesoscopic, or microscopic models, as well as descriptions of agents that move over the network (persons, containers, some other type of unit of freight, vehicles), which have their own characteristics with different levels of abstraction/detail. A summary of these is provided below:

- Macroscopic Models:
 - Nodes regard to origin and destination centroids, as well as transport network junctions (such as interchanges and intersections, rail shunting yards, etc.).
 - Links are described in general terms (length, allowed speed under free flow conditions, traffic density and speed curves) rather than their geometry.





- Agents are treated on an aggregate level, e.g., how many containers will be transported by rail or road based on a modal split function for the population, all rail vehicles are considered to have the same operational characteristics, and so on.
- Mesoscopic Models:
 - Links and nodes are treated in a similar manner, as explained in the macroscopic case.
 - Agents are treated on a more individual basis, i.e., choices are based on a per-trip basis, and agent characteristics can have variability within the same set, i.e., have vehicles that serve the same function with different kinematic profiles, energy consumptions, and pollutant emissions.
- Microscopic Models:
 - The geometric characteristics of the network are explicitly considered to evaluate the behaviour of agents over them rather than relying on generic functions.
 - Transport nodes are explicitly modelled according to their geometry and operational characteristics (lane widths, traffic signalization programs, signalling systems).
 - Individual agents have their own characteristics.

Considering the nature of the expected outputs in the L-Hub Use Cases, microscopic modelling was chosen as the most appropriate approach to calculate energy consumption and pollutant emissions for different types of vehicles, and to compare these outcomes for modal and route shifts that are expected to take place as a result of the proposed automations. Given that comparisons between different Use Cases presume the same methodological approach to modelling between them, this was followed even for cases where a more abstract approach may have been more expedient, as the trade-off between comparability and effort was seen to be favourable. As such, the DT component of the DSS takes into account the following:

- The transport network, which offers a representation of the physical links (roads, railway lines) linking different nodes.
- The demand, which is split in two different features:
 - Vehicle descriptions, including their behavioural characteristics (kinematics, rules that they employ to move over a route), energy consumption (be it fossil fuels or electricity), and emissions (air pollutants, noise).
 - Traffic flows between different nodes. These refer to the actual numbers of vehicles, and the distribution of their travel flows from one node of the network to the other over time.

In the context of the AUTOSUP Digital Twin, SUMO² shall be used to develop the transport model and simulate multimodal freight transport flows within the L-Hubs. Simulation of Urban MObility (SUMO) is an open-source, microscopic traffic simulation software that models the movement and interaction of vehicles, pedestrians, and public transport within a defined transport network. The tool allows the integration of detailed network and traffic data, enabling the analysis of transport performance indicators such as travel time, throughput, congestion, and emissions. When coupled with the DSS, SUMO

² <https://eclipse.dev/sumo/>



simulations can support “what-if” scenario evaluations, enabling the assessment of the operational impact of automation levels, infrastructure changes, or policy interventions.

In AUTOSUP, the DT simulation is expected to offer the following metrics regarding a transport system’s performance using SUMO:

- Trip statistics:
 - Total and average trip duration (seconds).
 - Total and average delays (seconds).
 - Delayed departures by number and duration.
 - Emissions and energy consumption:
- Fuel consumption. This is recorded by vehicle and time-step of the simulation and is aggregated for the entire network. SUMO reports this in milligrams per second (mg/s).
- Energy consumption. This refers to the electricity consumption of electric vehicles, either battery fed or through a power transfer network, such as a catenary or third rail for electric railway locomotives. SUMO reports this in watt-hours per second.
- Air pollutant emissions, recorded in milligrams per second (mg/s):
 - CO₂.
 - CO
 - HC (hydrocarbons).
 - Nox.
 - PM_x (particulate matter).
 - Noise, reported in dB.

These outputs - generated by the SUMO simulations - serve as key inputs to the Cost Benefit Analysis (CBA) model, enabling the quantitative assessment of the “what-if” scenarios defined in the L-Hub Use Cases and supporting evidence-based evaluation of their economic and environmental impacts. More details, on the application of the DT models in the L-Hubs can be found in the L-Hub deliverables, i.e., D2.1 *Digital Twin Models of the Trieste L-Hub* and D2.2 *Digital Twin Models of the Antwerp-Bruges L-Hub*.

3.3 Information Flows

The modelling of Information Flows (IFs) within the AUTOSUP framework focuses on the automation and optimisation of data exchange processes in multimodal freight transport. This work examines how information is generated, transferred, and utilised among key logistics systems operating at different Levels of Automation. The approach involves identifying the main systems involved in freight information exchange, documenting their processes, and defining both functional and technical requirements to promote real-time data exchange across modes and stakeholders. The resulting Information Flow models are closely linked to the SUMO transport simulations and CBA models, ensuring that data-driven insights on operational and economic performance can be integrated within the AUTOSUP Digital Twin and assessed through the DSS.

The systematic description of automated freight operations requires a structured taxonomy for mapping system entities, their interdependencies, physical and data flows, and fields for real-time data exchange. This can be addressed by Information Flow mapping of freight port operations, which consists of 4 steps:

1. **Identifying the system components.** Firstly, there is a need to define boundaries and key entities (nodes) of the physical and Information Flow system. The scope (boundaries) of this system can be outlined either on a macroscopic level - i.e. interactions between the freight port and external nodes such as rail or road terminals, facilities, networks; or on a microscopic level – i.e.





interactions within the freight port that concern specific entities and operations (e.g. maintenance facilities). A hybrid model might be necessary in certain instances that involve operations and flows between internal (port) and external entities.

2. **Mapping the Physical and Information Flows.** Physical flows and dependencies between system components are identified - e.g., train or vehicle movements, shuttle or shunting operations, (un)loading operations. The physical object flows should be then complemented with digital Information Flows, i.e. exchange of data that is directly associated with - and significant to - system operations: e.g., travel time estimates, vehicle data, freight load information, slot allocation, diagnostic protocols, clearance or authentication decisions, etc.
3. **Identifying the Information Flows for real-time exchange facilitation.** The third step involves assessing the key Information Flows to be supported with real-time data exchange solutions. Especially, Information Flows central to the efficiency and reliability of proposed freight automation solutions should be prioritised here. Relevant and feasible technologies for real-time data exchange shall be preferably indicated as well, e.g., automated vehicle location or recognition, vehicle-to-infrastructure or vehicle-to-vehicle communications, dynamic transport management dashboards, etc.
4. **Integrating the Information Flow maps into the DSS.** In this final step, the Information Flows are incorporated as a DSS development input. The DSS is a virtual tool that contains data from multiple sources, such as (a priori) assumptions, historical and/or present-day data, and processed simulation outputs. It aims to aid the strategic decision-making for prioritising investments in port automation. Using different model output metrics, the DSS will present the potential advantages of different port automations to relevant decision-makers. In that context, the Information Flows map the data movements across nodes and processes, gaps and bottlenecks during the DSS development. Furthermore, they help reveal potential vulnerabilities in data flows (and processes), thus simplifying the communication between internal (port) and external (logistics) stakeholders and improving the decision-making transparency.

Information flow modelling helps clarify the components and interdependencies necessary in freight operations to enable their automation and improve their interoperability. Moreover, this approach emphasises the information (data) exchange as the key driver of automation. Dynamic information exchange and utilisation are instrumental for real-time decision support, predictive analytics and robust operational management that form the core of future freight automation. The application of the Information Flows in the AUTOSUP Use Cases is presented in the relevant L-Hub deliverables, i.e., D2.1 *Digital Twin Models of the Trieste L-Hub* and D2.2 *Digital Twin Models of the Antwerp-Bruges L-Hub*.

3.4 Network Optimisation Algorithms

The EU network flow model is a macro-level model that captures aggregate cargo movements within Europe's TEN-T network, treating it as a system of Physical Internet (PI)-enabled nodes and corridors. The model sets-up a single commodity network with predefined source and sink nodes and their associated supply and demand capacities, respectively. Sources are then linked optimally to sink nodes, considering the capacitated links available between various nodes in the network. Links are associated with costs that arise from a generalised cost function and travel times.

The model considers links connecting European cities for road, rail, sea and river modes. The PI-enabled nodes are represented as transshipment locations where multimodal terminals are available. The PI nodes are also associated with normalised trade inflow or outflow volumes, which represent the export and



import flows between at least two network nodes. It then calculates the optimal routes based on distance, travel time, or other user-defined parameters, while considering the throughput capacity for each node and link. It allows for the representation of Living-Hubs at a different aggregation level that accounts for multiple port and transshipment terminals and other PI Hub functionalities, as illustrated in Figure 4.

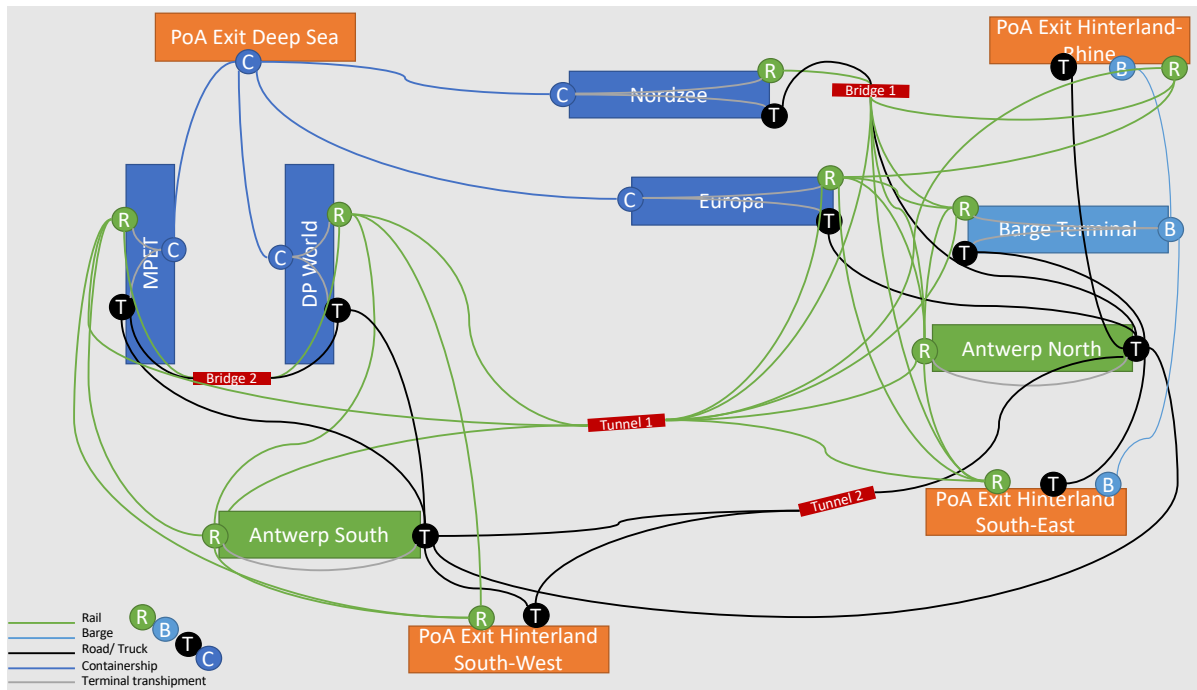


Figure 4. Example of PI Hub representation

Disaggregate Living-Hub representation enables the accurate modelling of the port cargo movement and transshipment costs, which have a significant impact on transshipment potential. The model can therefore be configured to quantify aggregate flows and how they are impacted by infrastructural and operational improvements in the network.

Each node is associated with a positive or negative trade-balance, classifying them into source or sink nodes. A flow assignment algorithm is used to quantify the total cost to satisfy demand, which is used as a proxy for network performance. The algorithm considers KPIs representing the generalised cost of travel for multiple modes and transshipments and can be extended to accommodate user-defined indicators.

In the context of AUTOSUP, the EU network flow model is expected to offer macro-level insights for the effectiveness of specific technologies and produce supply-demand fluctuation estimates and potential profits from demand changes in the cost-benefit analysis. More details on this model shall be presented in the upcoming version of this deliverable, i.e. D2.4 *DSS for operational automation in multimodal hubs (v2)*.

3.5 Cost-Benefit Analysis

The Cost-Benefit Analysis (CBA) model enables the direct financial comparison of technology investment potential against a baseline scenario. The tool analyses investment potential by considering:



- capital and operational expenses,
- future operational benefits,
- environmental benefits,
- demand, productivity and competitiveness benefits.

For each category, a thorough methodological approach was defined to capture and consolidate all the required information. Existing literature and generic sources were utilised to populate generic values, while the integration with the project's DSS and DT Models enables the accurate modelling of cargo movements and transshipment costs within each L-Hub.

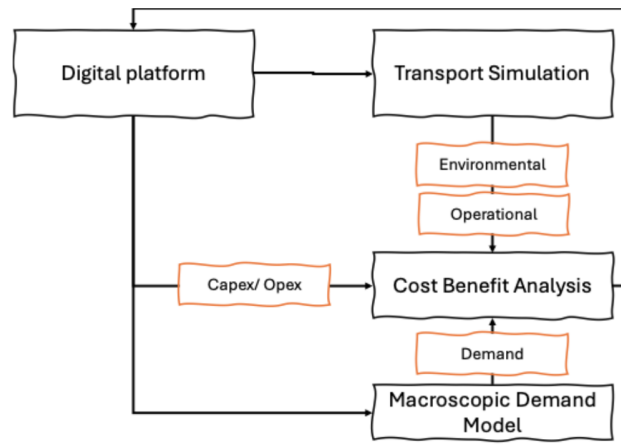


Figure 5. CBA and DT Models interaction

As Figure 5 illustrates, the DSS (i.e., digital platform) provides the initial investment costs associated with the operational set-up that is examined in each scenario. The DT model - presented in section 3.2 - i.e., Transport Simulation model, provides information on operational efficiency and environmental performance. The user-defined planning horizon and an annual discount rate enable the calculation of investment NPV (Net Present Value) to discount cost and profit components. The macroscopic demand model quantifies how aggregate flows are impacted by infrastructural and operational improvements in the network and cause demand changes for the planning horizon of the analysis.

The CBA model itself is developed as an active component of the project's WP2 Digital Twin offering. Following its integration within the DSS, it shall be possible to perform automated, generic and robust analysis of technological innovations in port systems. Its requirements and components, therefore, align with the outputs of predecessor models, and the calculations are designed to fit in the context of AUTOSUP. A detailed presentation of the CBA model and the underlying methodology can be found in the report *D3.2 Cost-Benefit Analysis*.



4 DSS Platform Development

This chapter presents the development of the Decision Support System (DSS) platform, detailing its overall technical architecture, the integration of DT models and Knowledge Graph, and the design of the user interface (UI) that enables interaction with the system. It describes how the platform's components are structured to ensure seamless data flow between different data sources, models, and visualisation layers, supporting efficient analysis and decision-making. The chapter also explains how the integrated models and the Knowledge Graph are exposed through an interactive dashboard, allowing users to explore system relationships, evaluate scenarios, and interpret model outputs in real time. Together, these elements form a cohesive digital environment that transforms complex data and models into practical decision-support insights.

4.1 Technical Architecture and Components

The system architecture shown in Figure 6 highlights the main components of the application and how they interact: the User Interface (UI), Backend Orchestrator (BO), Knowledge Graph (KG) modules, Keycloak authentication service, and the DT Model Library. Each element contributes a specific function to enable secure access, effective data processing, and smooth model execution, ultimately delivering a seamless user experience. This layered design promotes modularity, scalability, and flexibility, ensuring the platform can evolve and expand over time.

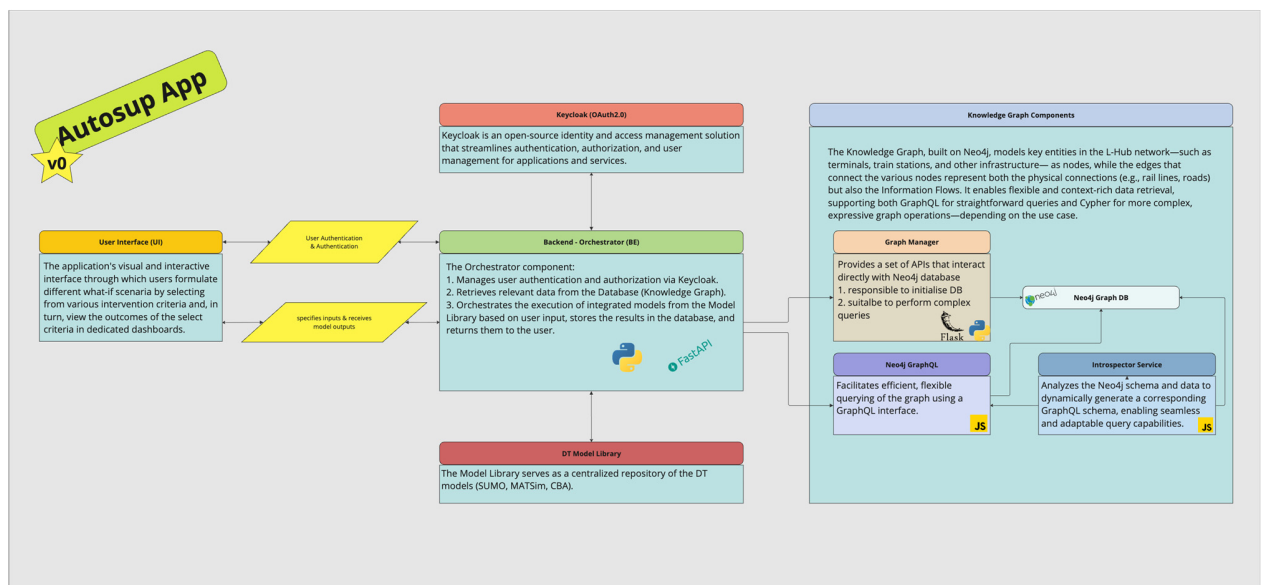


Figure 6. AUTOSUP DSS Architecture

At a high level, the architecture comprises of the following components:

- **User Interface:** The front-end layer that allows users to interact with the system, define input parameters, and review results through an intuitive graphical interface.
- **Authorisation/Authentication - Keycloak Service:** Provides secure identity and access management using industry-standard protocols such as OAuth 2.0 and OpenID Connect.





- **Backend Orchestrator:** Acts as the central logic layer, handling user authentication, managing data flow, orchestrating model execution, and interfacing between the UI, the Knowledge Graph, and the models.
- **Knowledge Graph:** A graph-based data structure that organises entities as nodes and their relationships as edges, enabling intuitive representation of both connections and interactions within the system. It allows for flexible, context-aware data retrieval and supports a variety of query approaches depending on the use case.
- **DT Model Library:** A set of models that address different problem dimensions within the system. Some models are designed to simulate operational and environmental aspects, such as routing efficiency and emissions, while others focus on evaluating economic or financial impacts. The library remains flexible and expandable, supporting the integration of additional models as new needs emerge.

The following sections present these components in more detail.

4.1.1 Authorisation and Authentication

Keycloak³ is an open-source Identity and Access Management solution that provides robust support for modern authentication and authorization standards, including OAuth 2.0⁴ and OpenID Connect⁵. In the AUTOSUP application, Keycloak serves as the central service responsible for securing access to backend resources.

The key benefits of using Keycloak for identity and access:

1. Open-source and enterprise-ready: No licensing costs
2. Supports Industry-Standard Protocols: Native support for OAuth 2.0 and OpenID Connect (OIDC) ensures wide compatibility.
3. Comprehensive User and Access Management: Built-in admin console for managing users, roles, groups, and policies with fine-grained access control.
4. Easy Integration and Extensibility: Works well with microservices and frameworks such as FastAPI⁶.
5. Strong Security Features: Offers brute force detection, multi-factor authentication, session management, and credential protection.

³ <https://www.keycloak.org/>

⁴ <https://oauth.net/2/>

⁵ <https://openid.net/developers/how-connect-works/>

⁶ <https://fastapi.tiangolo.com/>



6. Active and Supportive Community: Large open-source community with frequent updates, plugins, and broad ecosystem support.

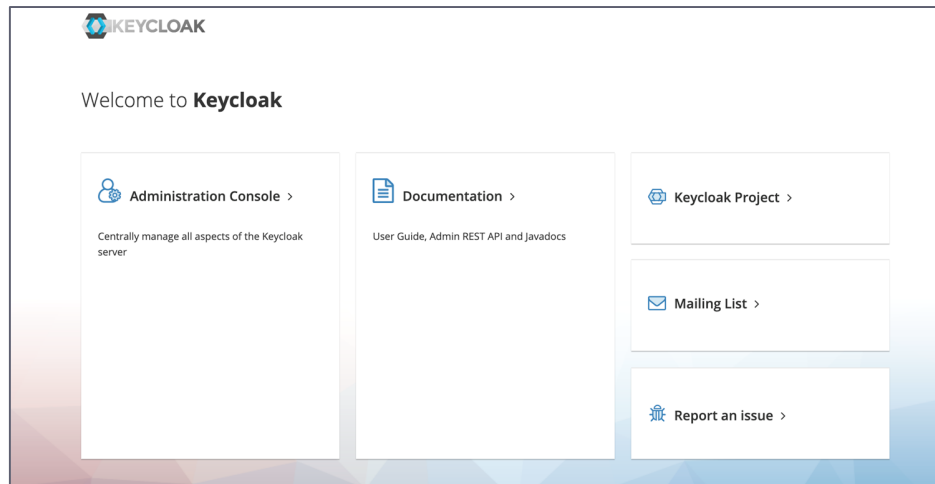


Figure 7. Keycloak Main Dashboard

In this system, presented in Figure 7, each user is assigned unique credentials, initially generated by the administrator through the Keycloak admin console. When a user logs in via the User Interface (UI), these credentials are forwarded to the backend, which then interacts with Keycloak for authentication. If the login is successful, Keycloak issues a JSON Web Token⁷ (JWT) with a configurable expiration period. The token is sent back to the UI and appended to all subsequent backend requests. For every request, the backend validates the token with Keycloak to confirm its authenticity and active status. If the token is invalid or expired, the backend rejects the request and returns an unauthorized response. In such cases, the UI redirects the user to the login page to re-enter their credentials and retrieve a new token.

4.1.2 Knowledge Graph

The Knowledge Graph serves as the central data backbone of the AUTOSUP platform (presented in Figure 8). It represents the Interdependency Graph of AUTOSUP, i.e., the high-level representation of the network of interconnected logistics infrastructures and the transport network assets in the L-Hubs. Implemented using a Neo4j⁸ graph database, it models key entities in the logistics network—such as terminals, train stations, and other infrastructure—as nodes, while the edges capture both physical connections (e.g., rail lines, roads) and Information Flows between them. This dual representation allows the system to integrate structural and informational aspects of the network within a single data model.

⁷ <https://www.jwt.io/introduction>

⁸ <https://neo4j.com/>



Such an approach supports context-rich and flexible data retrieval, making it possible to extract meaningful insights for both operational and strategic Use Cases.

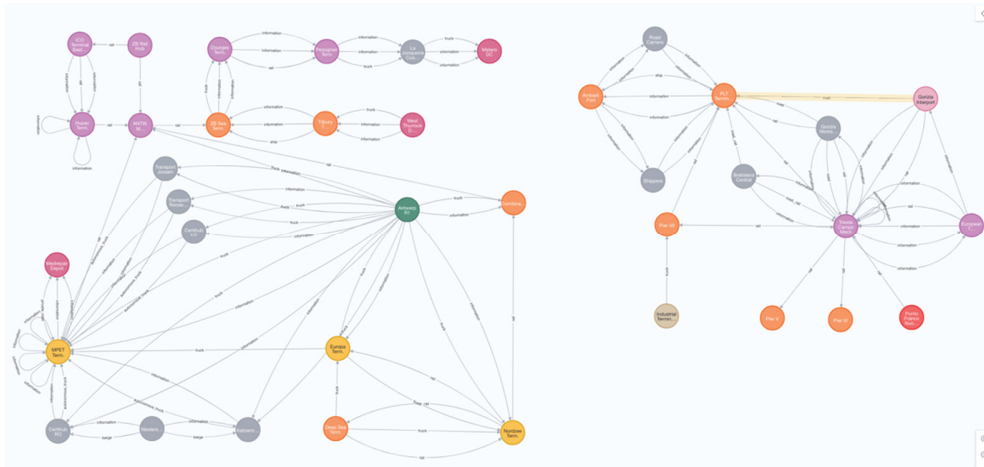


Figure 8. The AUTOSUP Knowledge Graph

To accommodate different levels of complexity in querying, the Knowledge Graph offers two complementary access methods:

- GraphQL⁹ for efficient and straightforward queries, enabling user-friendly data access.
- Cypher¹⁰ for advanced, expressive graph operations suited to complex analysis.

In the overall architecture, the Knowledge Graph is complemented by supporting services:

- Graph Manager, which provides an API layer to interact with the database, initialize data, and manage complex queries.
- Introspector Service, which analyses the Neo4j schema and dynamically generates GraphQL schemas, ensuring seamless and adaptable query capabilities.

Together, these elements ensure that the Knowledge Graph is not only a repository of network data, but also a flexible query engine that empowers the Backend Orchestrator, DT Model Library, and Simulation Engine to perform integrated operations. Its modular design ensures scalability and extensibility, aligning with the platform's long-term evolution.

⁹ <https://graphql.org/>

¹⁰ <https://neo4j.com/docs/getting-started/cypher/>



4.1.3 Orchestrator

The Backend Orchestrator serves as the core component of the AUTOSUP architecture, facilitating seamless interactions and orchestrating key operations across different system layers. Built using FastAPI, a modern and high-performance Python web framework, the orchestrator performs as the central coordination hub, ensuring smooth communication between the User Interface (UI), the Neo4j Knowledge Graph (KG), and the Model Execution Layer. It also interacts with a Queue that ensures decoupling of simulation execution tasks and the overall Model Execution process. It abstracts the complexities of graph management, simulation initiation, and model orchestration, providing a seamless user experience that supports informed decision-making and scenario analysis.

Below, there is a detailed explanation of the orchestrator's functions, responsibilities, and how it integrates with other system components in the architecture.

4.1.3.1 User Authentication and Authorisation

The orchestrator is responsible for managing user authentication and authorisation, ensuring that only authorised users can trigger simulations or access sensitive system components. This is handled through integration with **Keycloak**, a robust identity and access management system.

- **Authentication:** When users log into the UI, the orchestrator interacts with Keycloak to authenticate their credentials. If valid, Keycloak returns an **access token** that the orchestrator will use for subsequent interactions.
- **Token Validation:** Before each request is processed, the orchestrator verifies the **JWT (JSON Web Token)** associated with the access token by introspecting it to ensure:
 - The token is valid.
 - The token is active and unexpired.
 - The token corresponds to an authorized user.

This process ensures secure and controlled access to the backend, preventing unauthorized or malicious activities.

4.1.3.2 Graph Initialisation

In the AUTOSUP project the Knowledge Graph (KG) serves as the virtual representation of the L-Hub network. The orchestrator is responsible for managing graph initialisation, thus ensuring the system is set up properly. Specifically, it supports the import of structured data via an Excel file.



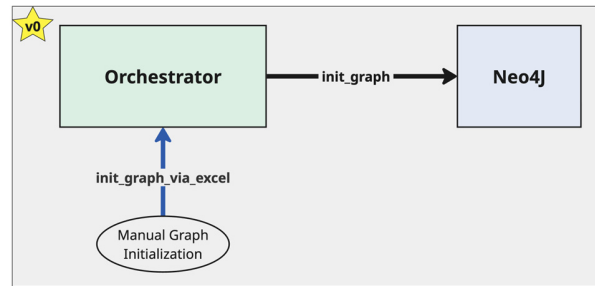


Figure 9. Graph initialisation process

In this file, nodes must be clearly defined with the correct format, including their names, types, coordinates, and hub designations. Additionally, both physical and/or information links between nodes must be accurately defined, specifying which nodes are interconnected and the types of relationships between them. This functionality allows administrators to preload datasets into the graph, ensuring that the system can quickly accommodate custom datasets and scenarios. The entire process is illustrated in Figure 9.

4.1.3.3 Interaction with Neo4j Knowledge Graph (KG)

The orchestrator is responsible for the interaction with the **Neo4j Knowledge Graph**, which stores the foundational data of the system. More specifically, the orchestrator oversees the following:

- **Query Construction and Execution:** Based on user inputs and simulation parameters, the orchestrator constructs dynamic **Cypher queries** to extract relevant entities and relationships from the graph. For example, it might retrieve information about roads, traffic intersections, or public transport routes.
- **Data Transformation:** The retrieved data is transformed into structured outputs that are used in subsequent processing stages, such as model execution or visualization.
- **Update Simulation Data:** As the simulation progresses and results are generated, the orchestrator updates the corresponding simulation node in the Neo4j graph with the latest status and the results as new properties.

4.1.3.4 Simulation Creation & Coordination

Regarding the simulation part of the Orchestrator, its role begins when a user initiates a simulation through the UI. The simulation process includes the following key steps and is presented in Figure 10:

- **Choose Port:** The user selects an L-Hub port for which the simulation will be run. The UI allows users to load predefined city graphs, which are then displayed in the system for further analysis and modification.
- **Load Graph:** After selecting the port, the corresponding graph, which represents the urban infrastructure and other relevant data, is loaded from the **Neo4j Knowledge Graph**.



- **Modify Graph:** Users can make temporary modifications to the graph. Importantly, these changes only apply to the current simulation session and do not affect the base graph.
- **Define Input for Models:** The user defines the input parameters for the simulation models.
- **Run Simulation:** After the inputs are defined, the orchestrator triggers the creation of a new simulation instance.

Upon receiving the simulation initiation request, the orchestrator performs the following tasks:

- **Create a Unique Simulation UUID:** A unique identifier is generated for the simulation to ensure traceability and proper management.
- **Create Simulation Node in Neo4j:** The orchestrator creates a new Node of Simulation Type in the Neo4j database. This node represents the simulation, storing initial details such as the simulation status.
- **Push to Queue:** The orchestrator then pushes the simulation entry to the **Queue**, signalling the beginning of the simulation workflow. This decouples the orchestration of the simulation from the execution itself.

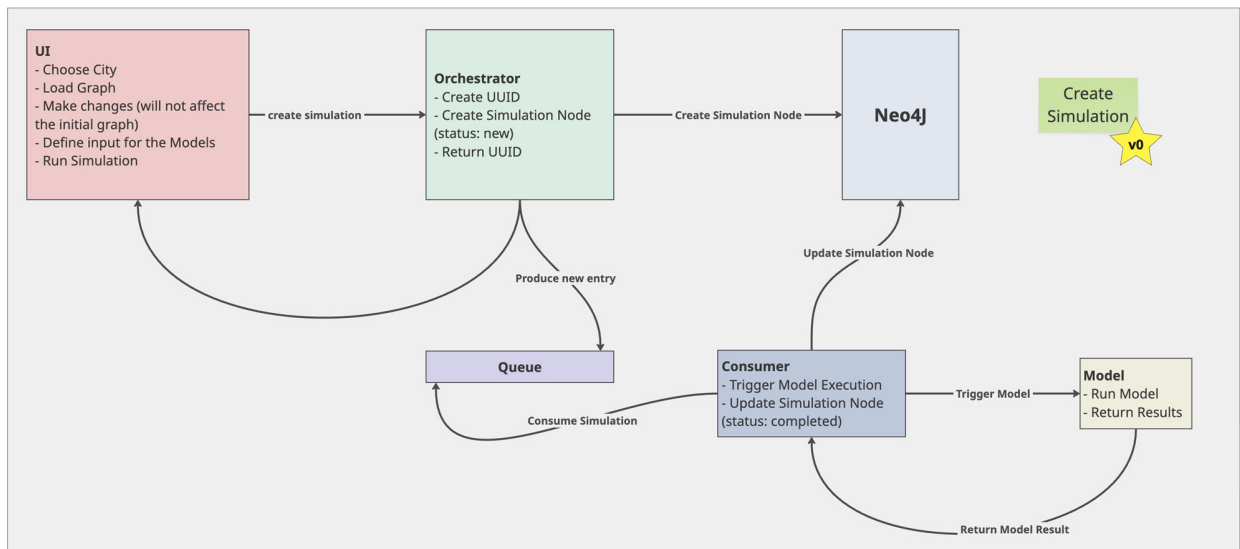


Figure 10. Simulation creation process

4.1.3.5 Model Execution via Queue-based Architecture

The orchestrator decouples the model execution process from other components using a **Queue**, which ensures scalability and robustness in handling multiple simulations and model executions:

- **Produce Simulation Entry:** Once the orchestrator creates the simulation node in Neo4j, it produces a new entry in the queue that triggers the simulation workflow.
- **Queue Consumption:** A **Consumer** component listens for new simulation requests in the queue. Upon receiving a request, the consumer picks up the simulation task and triggers the appropriate





model(s) from the **Digital Twin Model Library**, which could include simulation tools such as **SUMO** and **CBA** (Cost-Benefit Analysis).

- **Model Execution:** The chosen model is executed with the input data provided by the orchestrator. This step involves running the model, analysing the system's response to the given inputs, and generating results.
- **Return Results:** After the model execution is complete, the **Model** service sends back the simulation results to the orchestrator. These results are then processed and stored as properties in the corresponding simulation node in Neo4j.

4.2 Model Integration and Execution Pipeline

This chapter describes the mechanisms through which the models presented in Chapter 3 are integrated and executed within the AUTOSUP DSS. It outlines the overall model execution pipeline and details the model integration specifications, which establish the technical principles and standards for incorporating models into the DSS. Together, these processes ensure a consistent, scalable, and interoperable modelling environment that supports automated scenario analysis and the seamless orchestration of the AUTOSUP Digital Twin components.

4.2.1 Model Execution Pipelines

The model execution pipeline within the AUTOSUP DSS defines the sequence and interaction of the integrated models, such as the Digital Twin (DT) transport models and the Cost-Benefit Analysis (CBA), from input configuration to output visualisation, ensuring a seamless data flow, controlled execution environment, and consistent integration of model results within the DSS User Interface.

In line with the AUTOSUP requirements, which emphasise the need to explore a range of “what-if” scenarios assessing potential investments in logistics innovations across the L-Hubs, the models are designed to be executed in an integrated manner to deliver a unified output to the user. Accordingly, the models are executed sequentially, forming a coherent model chain or scenario pipeline, where the outputs generated by the DT transport models serve as inputs to the CBA model. This sequential execution ensures that operational performance indicators derived from transport simulations are directly translated into economic and sustainability assessments within the DSS. Additionally, all the models allow users to define input parameters directly through the DSS User Interface, enabling the exploration of different operational and strategic scenarios and the evaluation of their impacts in terms of both performance and economic outcomes.



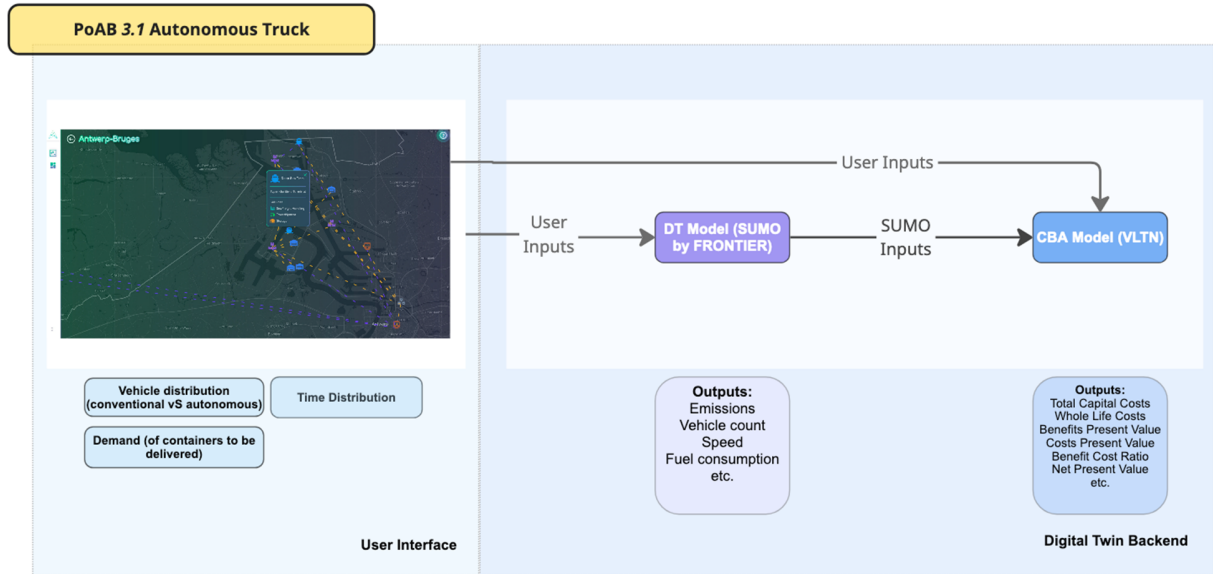


Figure 11. Model Execution Pipeline (PoAB, Intervention 3.1)

This pipeline is depicted in Figure 11 and Figure 12. The first figure presents the model execution pipeline for Use Case 3 (Intervention 3.1) in the Port of Antwerp Bruges. In this case, the users are called to define different parameters, such as the demand of containers to be delivered and the vehicle and time distribution directly on the User Interface. Their chosen options will be used as inputs to the models, which in turn will be executed as a sequence, and will deliver the final outputs to the user in an integrated manner through the User Interface. Similarly, Figure 12 presents the scenario configuration parameters for the Use Case of the Port of Trieste.

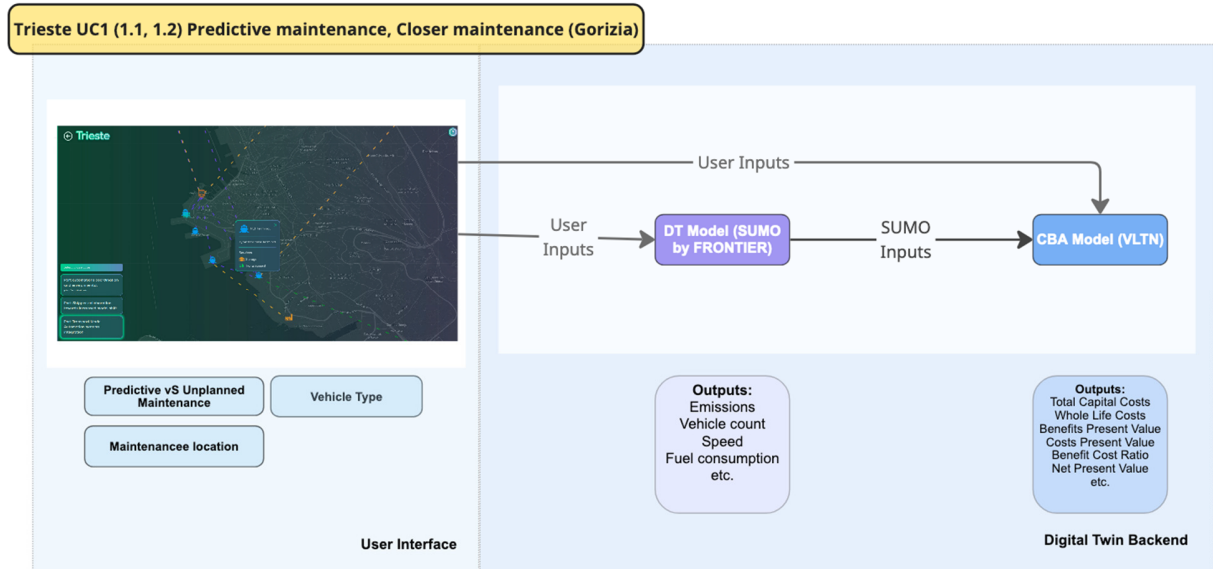


Figure 12. Model Execution Pipeline (Trieste, Use Case 1)



4.2.2 Model integration

The integration of the DT models and the CBA model into the DSS is a key step in ensuring seamless execution and interoperability across all system components. The integration approach depends on the accessibility of the model code.

If the source code of the models is available, the preferred method is to containerise each model and integrate it directly within the DSS. Containerisation, typically achieved through Docker, allows each model to be packaged with its dependencies and configuration, ensuring reproducibility and platform independence. This approach enables the DSS to execute the models in a controlled environment and ensures that all simulations can be deployed consistently across different systems.

If, however, the models are provided as closed-source executables or external applications, integration will instead be achieved through Application Programming Interfaces (APIs). In this case, the DSS will communicate with the models via API calls, exchanging input and output data dynamically. The integration methods shall include the use of RESTful APIs, allowing HTTP-based communication and JSON or XML data exchange between the DSS and the models.

For successful integration, each model must adhere to a common configuration standard, which includes the following key requirements:

- Clearly defined input and output variables.
- No hardcoded variables within the model code.
- Capability to run independently on any machine, not limited to the modeller's local environment.
- Support for containerisation, ensuring consistency, scalability, and ease of deployment within the DSS.

The integration of the AUTOSUP models into the AUTOSUP DSS establishes a unified framework in which all modelling components can be executed, managed, and visualised through a common User Interface, enabling coherent scenario analysis and decision support across the L-Hub use cases. By following these specifications, the DSS can execute and orchestrate all models efficiently, enabling consistent “what-if” scenario simulations and analysis across different L-Hub use cases.

4.3 User Interface

The User Interface (UI) serves as the interactive layer between the user and the DSS's underlying Knowledge Graph, DT models, and their outputs. Through an intuitive dashboard, it visualises the complex relationships between the nodes, dependencies, and simulation results in a clear and accessible manner. Users can explore entities and their semantic connections within the Knowledge Graph, explore different what-if scenarios through scenario execution, and immediately view the corresponding outputs in dedicated dashboards. By combining visualisation, interactivity, and contextual information, the UI transforms the underlying data and models into actionable intelligence that supports informed decision-making.

The UI is developed using React and TypeScript, which offer a type-safe and modular way for developing dynamic and maintainable components. React enables the creation of an interactive interface, while TypeScript adds strong typing and improved code reliability. For styling, Tailwind CSS is used to efficiently design a clean and modern layout, simplifying the process of creating a consistent and visually appealing interface. The map visualisation is implemented using Leaflet, an open-source JavaScript library that provides powerful and flexible tools for rendering interactive maps. Together, these technologies create



a smooth, visually appealing, and interactive experience that allows users to input data, run simulations, and view map-based and dashboard visualisations.

4.3.1 Interdependency Graph

The Interdependency Graph offers a high-level representation of the network of interconnected logistics infrastructures and the transport network assets around the L-Hubs ports. It is depicted in the Knowledge Graph developed in neo4j and presented previously in section 4.1.2. As shown earlier in Figure 6, the Knowledge Graph is exposed to the end-user through the DSS User Interface.

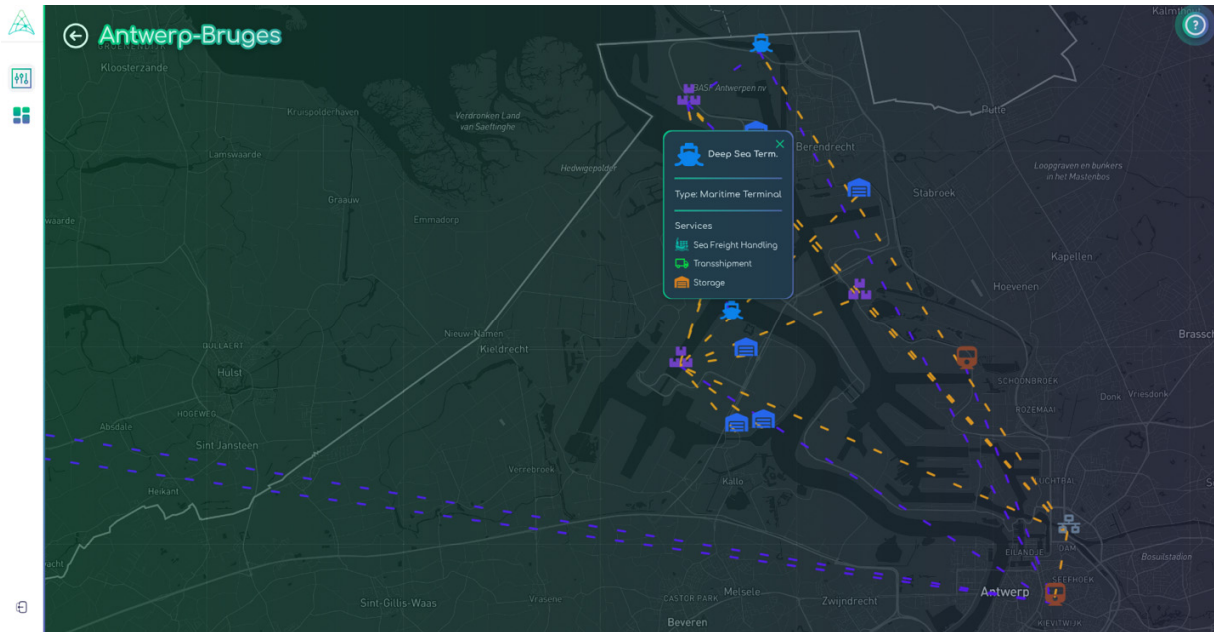


Figure 13. Antwerp-Bruges Interdependency Graph exposed in the UI

As Figure 13 illustrates, the different nodes of the Knowledge Graph of Antwerp-Bruges (presented earlier in Figure 8) are represented with various icons, depending on what they represent (e.g., a port, rail terminal etc.), in a map visualisation of the L-Hub. Once the user hovers over any of these icons, they view further node attributes (e.g., services and type).



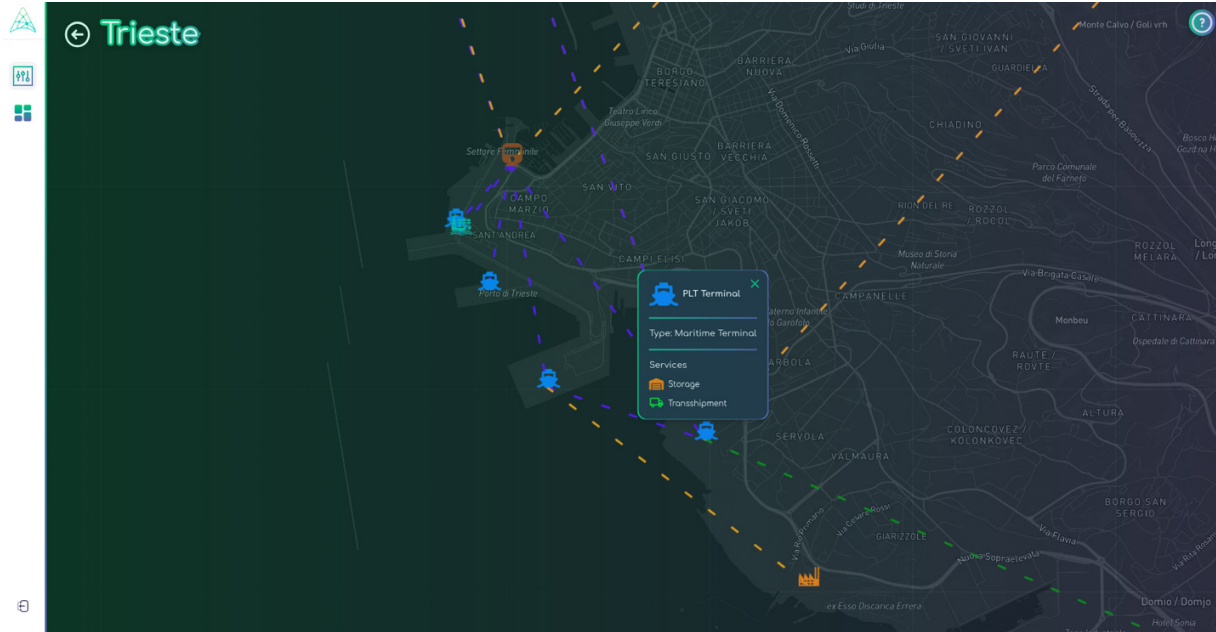


Figure 14. Trieste Interdependency Graph exposed in the UI

Similarly, Figure 14 shows how the User Interface presents the different nodes of the Trieste Knowledge Graph. Once the integration of the Information Flows in the Knowledge Graph is completed, these will also be visible in the User Interface.

4.3.2 Scenario Builder

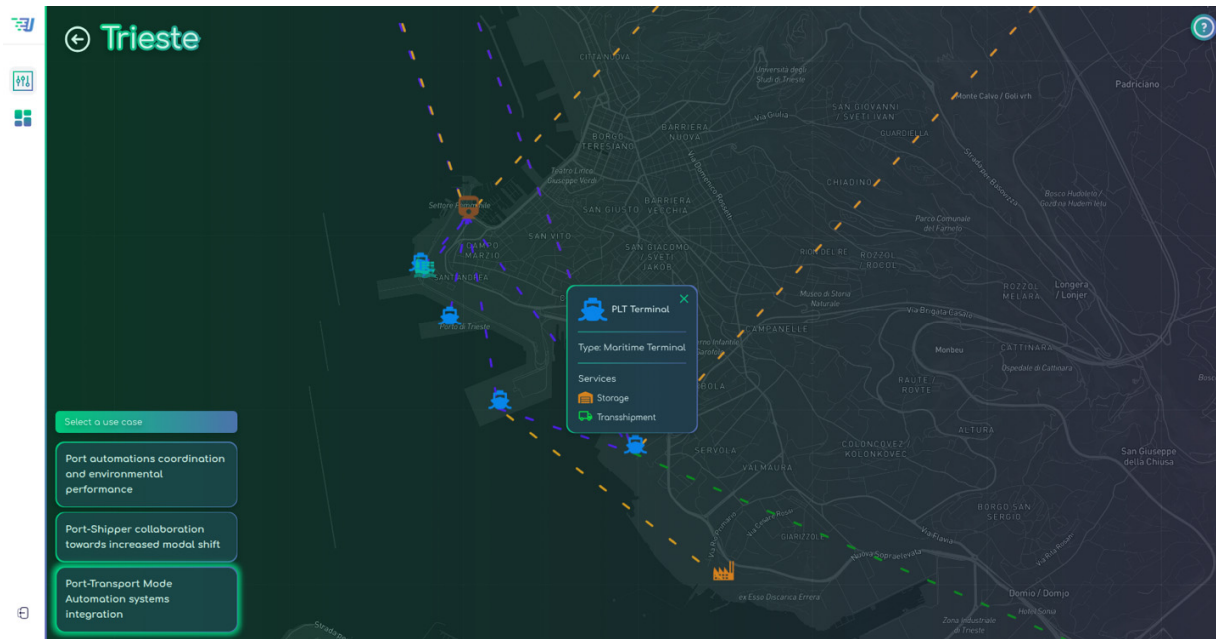


Figure 15. Trieste - Use Case selection

As presented earlier in Figure 11 and Figure 12, as the user interacts with the UI and selects different criteria, different scenarios for the various Use Cases shall be developed and executed. Figure 15 presents



the initial screen for the Trieste L-Hub. By interacting with the map, the user may view the different nodes of the Knowledge Hub, including the physical and information links between them. They may also select one of the available Use Cases. After selecting a Use Case, they are directed to a new screen representing the scenario that will be examined, as shown in Figure 16. In the example presented in this Figure, the relevant inputs for a Use Case in Trieste are expected to be adjusted by the user. Upon pressing the “Calculate” button, the sequence of models (defined earlier in Figure 12) shall be executed.

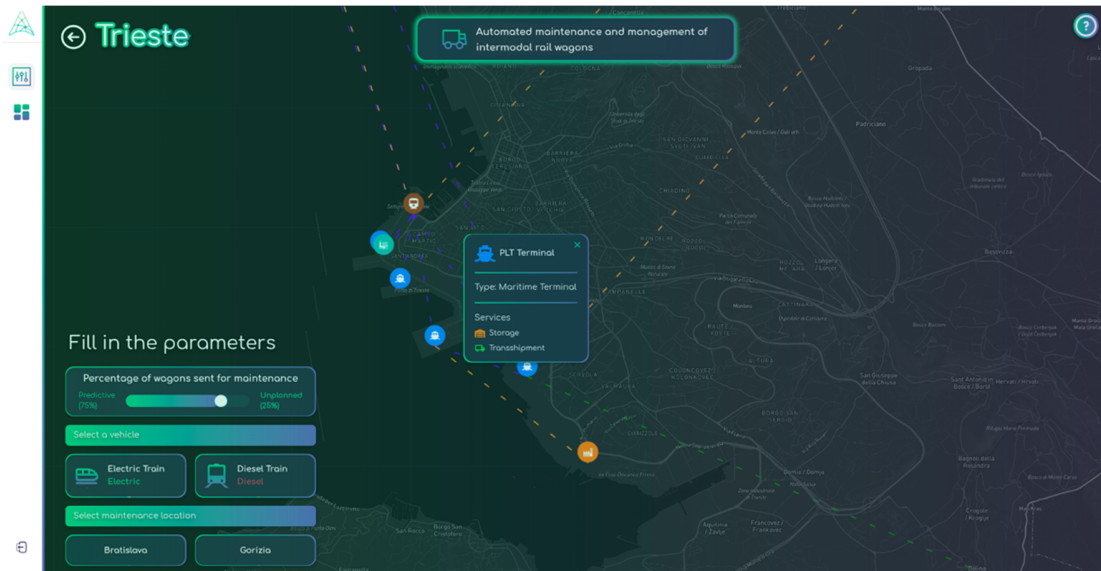


Figure 16. Trieste - Scenario Definition of inputs

Similarly to Trieste L-Hub, the UI offers a dedicated screen and map visualisation for the Antwerp-Bruges L-Hub (Figure 17). Here, the user may view the nodes and edges of the L-Hub’s KG, but also select one of the available Use Cases.

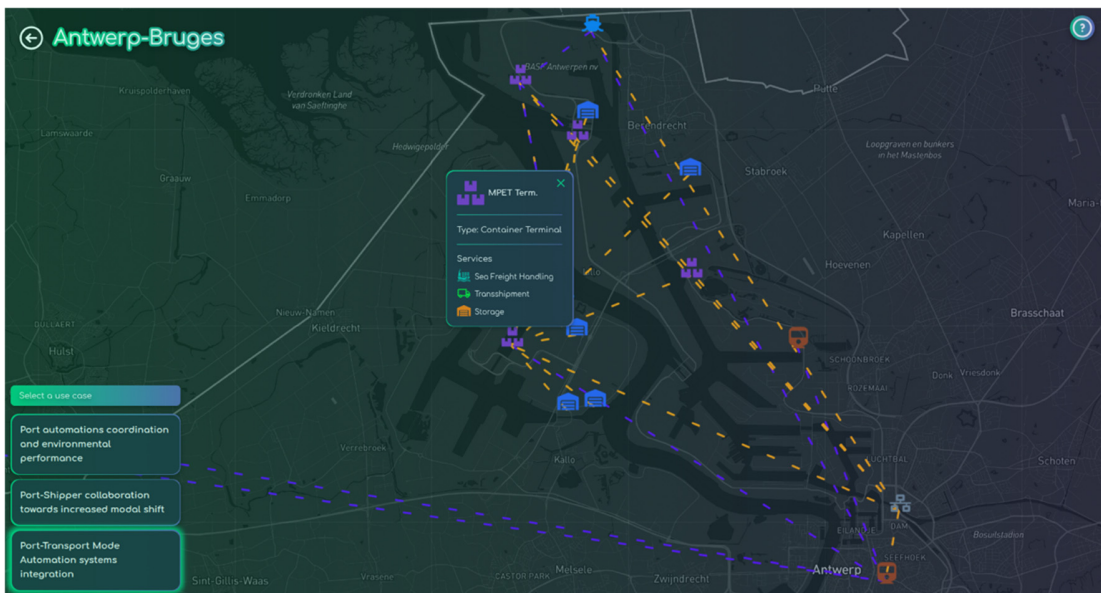


Figure 17. Antwerp-Bruges - Use Case selection



Upon selecting one of the Use Cases, the user is directed to a new screen representing the details of the selected scenario (Figure 18). In a similar manner to the previous L-Hub, the user shall define a scenario to explore by selecting the relevant inputs appearing on the screen. The next step is the execution of the underlying models associated with this scenario. Once the models finish, their results will become available in a dedicated dashboard. This is presented in the following section.

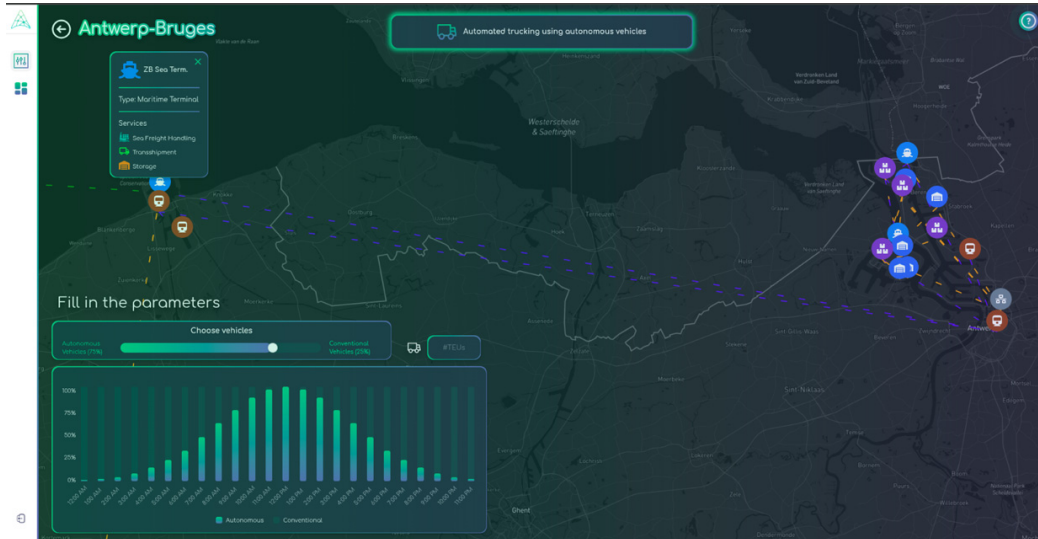


Figure 18. Antwerp-Bruges - Scenario Definition of inputs

4.3.3 Visualisation of Results

In each L-Hub a dedicated dashboard is generated, once the model execution is finished, and presented in Figure 19 and Figure 20 respectively. A map visualisation is available at the top of the dashboard, showcasing the nodes of the KG that are involved in the specific scenario. The dashboard also presents the inputs defined by the user in the scenario, as well as KPIs calculated by the DT models, such as Noise, Total Travel Time, Speed Metrics, and Energy Consumption.

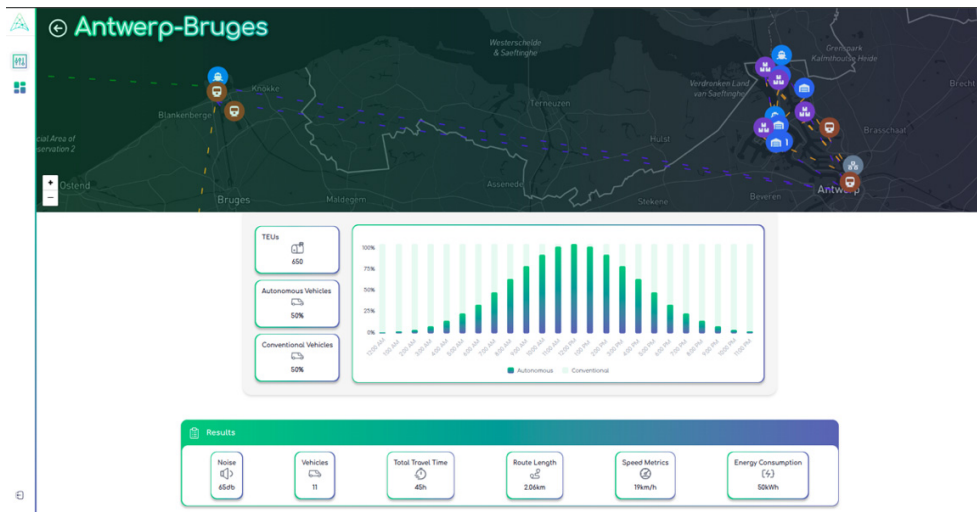


Figure 19. Antwerp-Bruges - Example Dashboard of outputs



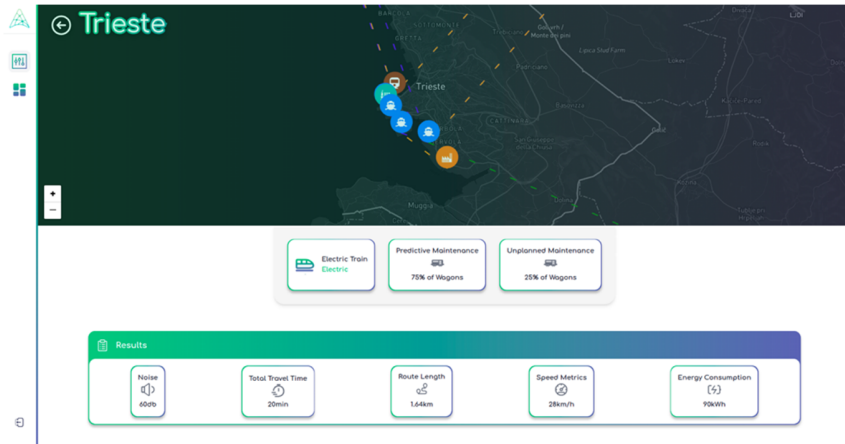


Figure 20. Trieste - Example Dashboard of outputs

Finally, the Dashboard presents different charts for the various outputs calculated by the DT models and the CBA model (Figure 21).

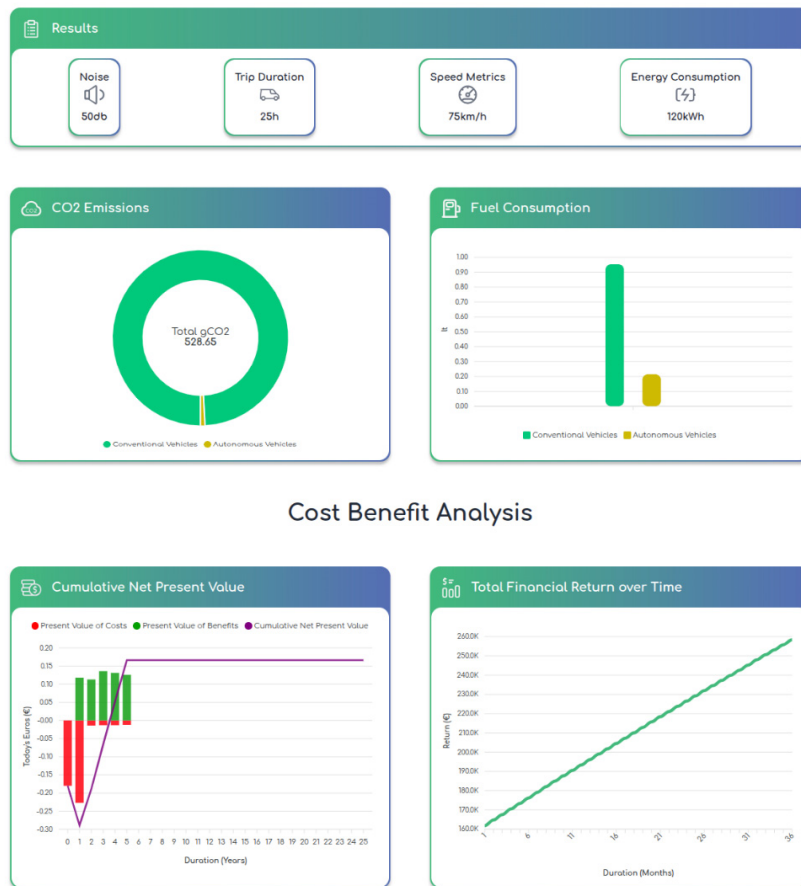


Figure 21. Example Dashboard - Model Outputs



5 Conclusion and Future Work

This report has presented the work leading up to the design and development of the AUTOSUP Decision Support System (DSS), which constitutes a key element of the project's Digital Twin of the Autonomous Multimodal Supply Chain (WP2). The report outlined the system's requirements, conceptual and technical architecture, and the integration of simulation-based components, including the Interdependency Graph, Digital Twin models, Information Flow analysis, Network Optimisation algorithms, and the Cost-Benefit Analysis model. Together, these elements provide the foundations for a unified environment that enables the simulation and evaluation of automation scenarios across multimodal freight transport operations.

By defining how models, data, and user interactions are orchestrated within the DSS, this work contributes directly to Objective O2 of the AUTOSUP project, empowering transport and logistics stakeholders with a data-driven tool that supports feasibility studies, scenario exploration, and informed decision-making for automation and digitalisation in supply chains. The deliverable also establishes the methodological and technical groundwork necessary for the integration of the various models, paving the way for a cohesive and operational Digital Twin ecosystem.

The next steps shall focus on the technical integration of all models developed for the L-Hub use cases within the DSS, enabling their execution and visualisation through a dedicated User Interface. This integration will allow L-Hub stakeholders to test and validate the DSS in real operational contexts, assessing its functionality and usefulness for strategic planning and investment evaluation. In parallel, efforts will be directed towards ensuring that the DSS and its underlying models are generalised beyond the AUTOSUP use cases, supporting future extensions and applications. This generalisation will be essential for establishing the DSS as a replicable and transferable solution, capable of serving diverse logistics environments and promoting wider adoption of automation across the European supply chain landscape.





**Funded by
the European Union**

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement N. 101147468. Funded by the European Union. Views and opinions expressed are however those of the autor(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

