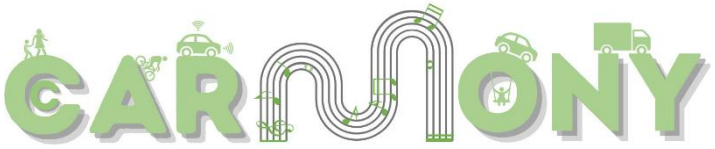
Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.



Traffic management services - Initial Version

Deliverable Number	D3.2
Deliverable Type	Other
Dissemination Level	PU - Public
Deliverable Responsible	José Triano (HYP)
Document Version & Status	V2.0 | Final

Project Acronym	CARMONY
Project Title	Coordinated Action for Responsive Mixed Orchestration and Network Yield
Grant Agreement Number	101202858
Project Coordinator	Virtual Vehicle Research GmbH
Project Website	www.carmony-project.eu

The document is submitted to the European Commission for review but has not been accepted so far. As soon as the final version of this document is available, this document should be deleted.

DRAFT



CARMONY has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101202858.

Author(s)

Name	Organisation
Kostas Cheliotis	FRON
Nasia Kakouri	FRON
José Triano	HYP
David Fleury	HYP
Dina Oulad Chair	HYP
Gerhard Benedikt Weiß	ViF
Georg Nestlinger	ViF
Luis Manuel Viso Dominguez	TEC

Reviewers

Name	Organisation	Date
Jordi Carbonell Silva	EUT	2026-07-13
Sara Janhäll	RISE	2026-07-08

Change History

Version	Date	Name/Organisation	Description
0.1	2026-05-08	Dina Oulad Chair /HYP	Created the table of contents
0.2	2026-05-27	Dina Oulad Chair /HYP	Completed the initial draft of the document
0.3	2026-06-09	Kostas Cheliotis /FRON	Added the FRON contribution
0.4	2026-06-09	José Triano/HYP	Updated the introduction
0.5	2026-06-15	Gerhard Benedikt Weiß /ViF	Added the VIF contribution
0.6	2026-06-15	V Luis Manuel Viso Dominguez /TEC	Added the TECNALIA contribution
0.7	2026-06-29	Dina Oulad Chair /HYP	Added the abbreviation List and acceptation of list of modifications and contributions

DRAFT

0.8	2026-07-02	Dina Oulad Chair /HYP	Added the list of figures and list of tables
1.0	2026-07-03	All authors	Final version of D3.2 to be reviewed internally by Carmony reviewers
1.1	2026-07-09	José Triano /HYP	Reviewer 1 comments integration
1.2	2026-07-15	José Triano /HYP	Reviewer 2 comments integration
2.0	2026-07-15	José Triano /HYP	Final version

DRAFT

Table of Contents

Executive Summary 8
1. Introduction 9
2. Objectives of tasks T3.2 and T3.3 10
3. Positioning of the Software within the Carmony architecture and uses cases 11
3.1 Interface definition and integration status 11
3.2 List of modules in alignment with the architecture in D3.1 11
3.3 Main functionalities of each module 12
3.3.1 Decision logic 12
3.3.2 Traffic situation identification 12
3.3.3 Database with known decisions 12
3.3.4 Traffic Forecasting 12
3.3.5 Route planner 13
3.4 Interfaces with other orchestrator modules 13
3.4.1 Real Time/Low Fidelity Simulation 13
3.4.2 Monthly/High Fidelity Simulation 13
3.4.3 Environment Simulation 13
3.4.4 Traffic Management Center (TMC) 14
4. Software functional scope 15
4.1 Decomposition of functional modules into SW components 15
4.2 Internal architecture of each module and interaction between SW components 17
4.2.1 Decision Logic & Traffic Situation Identification internal architecture 17
4.2.2 Traffic Forecasting architecture 17
4.2.3 Route Planner architecture 18
4.2.4 Simulation Situation Database architecture 18
4.3 Description of internal data flows and SW components outputs 19
4.3.1 TrafficJam flow 19
4.3.2 RegulationRecommendation flow 19
4.3.3 DynamicSlotRecommendation flow 20
4.3.4 JamDetectionLookup flow 20
4.3.5 JamDetectionLookupAnswer flow 20
4.3.6 RankedMitigationStrategies flow 21
4.3.7 RT-RankedMitigationStrategies flow 21
4.3.8 Traffic forecasting internal data flows 22
4.3.9 Route planner internal data flows 22
4.3.10 Simulation situation database internal data flows 22

DRAFT

5. SW technical scope 23
5.1 Technology stack of each SW component 23
5.2 Preliminary API configuration and characteristics 25
5.2.1 Current Endpoints of Decision Logic and Traffic Situation Identification 25
5.2.1.1 GET api/v1/predicted-traffic 26
5.2.1.2 GET api/v1/real-time 26
5.2.2 Traffic Forecasting endpoints 26
5.2.2.1 /forecast/flow/at_time/ 27
5.2.2.1 /forecast/speed/at_time/ 27
5.2.2.2 /forecast/density/at_time/ 28
5.2.3 Data Structure (SCHEMAS) 29
5.2.3.1 Snapshot 29
5.2.3.2 SegmentHistory 29
5.2.3.3 TrafficJam 29
5.2.3.4 RegulationRecommendation 29
6. Progress status of SW component development 30
7. Interfaces and integration with Carmony (HYP) 32
7.1 Context overview 32
7.2 User guide for the web-based frontend 32
7.2.1 Frontend URL 32
7.2.2 Create an account 32
7.2.3 Global navigation 33
7.2.4 Platform features 34
7.2.4.1 Feature 1: API Management & Documentation 34
7.2.4.1.1 API Documentation 35
7.2.4.1.2 API keys Management 35
7.2.4.1.3 Endpoint Documentation Pages 36
7.2.4.2 Feature 2: Data Visualization & Demonstration 37
7.2.4.2.1 Dashboard 37
7.2.4.2.2 Input data 38
7.2.4.2.3 Real time 40
7.2.4.2.4 Exploitation projects 41
7.3 User guide for the Traffic Forecasting service 44
8. Preliminary setup 46
8.1 A3 Network 46
8.1.1 A3 network configuration 46

DRAFT

8.2 Traffic data 46
8.2.1 Traffic forecasting calibration 46
8.2.1.1 Data Sources 47
8.2.1.2 Data Harmonization 47
8.2.1.3 Data Quality and Cleaning 47
8.2.1.4 Seasonal Traffic Patterns for selected motorway sensors 48
8.2.1.5 Anomaly Detection for selected motorway sensors 48
8.2.1.6 Models 49
8.2.1.7 Predictions 50
8.2.1.8 Model-based prediction profiles for selected motorway sensors 50
8.2.1.9 Sensor-level model performance across the A3 Highway 51
8.2.1.10 Summary 52
9. Current limitation and Conclusion 54
10. Abbreviations 55
11. References 56

List of Figures

Figure 1: Internal architecture of the Decision Logic and Traffic Situation Identification components. The preventive layer uses historical data to anticipate traffic degradations and propose preventive actions, while the reactive layer processes real-time data to detect emerging situations and trigger appropriate responses. 17
Figure 2: Conceptual architecture of the Traffic Forecasting module 18
Figure 3: Conceptual architecture of the Route Planner module 18
Figure 4: Conceptual architecture of the Simulation Situation Database module 19
Figure 5: Account registration page 33
Figure 6: Global navigation menu 34
Figure 7: API documentation page 35
Figure 8: API Key Management Page 35
Figure 9: API Endpoints Menu 36
Figure 10: API Endpoint Documentation Example 37
Figure 11: Interactive API Testing Interface 37
Figure 12: Dashboard page 37
Figure 13: Input RAW data page 38
Figure 14: Snapshot visualization 39
Figure 15: Real time snapshots 40
Figure 16: List of created exploitation projects 41
Figure 17: Project edition page 42
Figure 18: Filtered historical input data for an exploitation project 42
Figure 19: Predictions and regulations page 1 43
Figure 20: Predictions and regulations page 2 44
Figure 21: Traffic Forecasting service interactive documentation 45
Figure 22: Sample response from density forecast endpoint 45

DRAFT

Figure 23: Median typical daily traffic profiles for Sensor PK 12526, showing traffic flow, density and speed for both traffic directions. 48
Figure 24: Z-score-based anomaly detection results for sensor PK 12526, showing hourly traffic flow, density proxy and speed in both traffic directions. 49
Figure 25: Comparison of model-based predictions for the representative sensor PK 12526 during the test period, for both directions. 51
Figure 26: Sensor-level comparison of optimized model prediction performance across the A3 network, expressed as RMSE, for density, hourly traffic flow and speed in both traffic directions. 52

List of Tables

Table 1: Decomposition of functional modules into SW components 15
Table 2: Technology stack of Decision Logic SW components 23
Table 3: Technology stack of Traffic Situation Identification SW components 23
Table 4: Technology stack of Database with Know Decisions 24
Table 5: Technology stack of Traffic Forecasting SW component 24
Table 6: List of current endpoints of Decision logic and Traffic Situation Identification 25
Table 7: Description of the Traffic Forecasting endpoints 26
Table 8: Progress status of SW component development 30
Table 9: Prediction performance of the evaluated forecasting models for density, hourly traffic flow and speed. 50

DRAFT

Executive Summary

This deliverable presents the initial version of the CARMONY traffic management services developed within WP3. It supports the transition from conventional traffic management towards traffic orchestration for connected and automated mobility, by enabling the early identification, anticipation and mitigation of traffic degradation situations.

The work covers Task T3.2, focused on traffic pattern recognition and proactive incident detection, and Task T3.3, focused on dynamic mitigation measures generation. The services are designed to support the CARMONY Interlinked Orchestrator by identifying evolving traffic situations and proposing mitigation actions, such as speed regulation recommendations and dynamic slot creation strategies for carpool lane management.

The software includes several functional modules: decision logic, traffic situation identification, database of known decisions, traffic forecasting, route planner, simulation situation database and fleet operator optimisation. These modules use historical traffic data, traffic observations, predefined scenarios, simulation outputs and KPI-based evaluation to support road operators in selecting appropriate traffic management strategies.

The deliverable also describes the first software architecture, internal data flows, technology stack, preliminary APIs and demonstration frontend. The initial setup is applied to the A3 motorway in Luxembourg, providing a first basis for testing forecasting, situation identification and mitigation recommendation capabilities.

This initial version establishes the functional and technical foundations for future CARMONY traffic orchestration services, with further work dedicated to testing, refinement, integration and extension towards connected and automated vehicle interactions.

Key words: Traffic orchestration; CCAM; traffic degradation; incident detection; mitigation measures; speed regulation; traffic forecasting; decision logic; simulation; Interlinked Orchestrator.

DRAFT

1. Introduction

Within the CARMONY project, WP3 is developing software components to support the transition from conventional traffic management towards traffic orchestration for connected and automated vehicles. This work addresses the increasing complexity of road mobility systems, where infrastructure, vehicles, operators and digital services need to interact in a coordinated and dynamic manner.

The software presented in this intermediate deliverable D3.2 Traffic management services - Initial Version, focuses on early identification and anticipation of traffic degradation. By analyzing historical traffic data, the solution aims to identify situations that may lead to network degradation and support preventive traffic management actions. In the current development stage, this includes the generation of speed-based recommendations that can be broadcast to human drivers within a defined regulation area, ensuring that the right vehicle receives the right recommendation at the right place and at the right time.

This represents a first step towards more advanced traffic orchestration capabilities. Future development stages may support interaction with ADAS-equipped vehicles and, subsequently, direct communication with the navigation systems of connected and automated vehicles. In this way, the solution contributes to a progressive deployment path from human-readable instructions to increasingly automated traffic management interactions.

The work carried out by CARMONY stakeholders builds operational experience in complex mobility systems, including autonomous vehicle fleet management and rail control centre operations. This background has informed the design of the software so that it reflects the needs, constraints, and working processes of road operators.

The current deliverable presents the functional role, operational scope and integration perspective of the software components within the CARMONY architecture. A frontend environment has been developed to demonstrate the software outputs and support interaction with the developed functions during the project. However, the final contribution will consist of backend solutions integrated into the CARMONY Interlinked Orchestrator.

Detailed algorithms, internal decision rules and source code are not disclosed in this deliverable in order to preserve HYP’s proprietary know-how.

DRAFT

2. Objectives of tasks T3.2 and T3.3

Task T3.2: Patterns recognition and proactive incident detection services

The objective of task T3.2 is to develop software services capable of identifying traffic flow degradations using real-time traffic data combined with historical traffic data analysis.

The expected outcome of this task is a set of software services enabling traffic situation recognition and proactive incident detection functionalities within the Carmony interlinked orchestrator.

T3.3: Dynamic mitigation measures generation services

The objective of task T3.3 is to develop software services capable of generating dynamic traffic mitigation measures in response to detected or predicted traffic degradation situations.

The software services are designed to support traffic operators and connected mobility systems by providing relevant mitigation actions based on predefined scenarios, simulation results and traffic context analysis.

The expected outcome of this task is a set of software components able to generate and evaluate traffic mitigation measures for reducing traffic degradation impacts within the Carmony interlinked orchestrator.

DRAFT

3. Positioning of the Software within the Carmony architecture and uses cases

3.1 Interface definition and integration status

Incident Detection Services and the Dynamic Mitigation Measures Generation Services (Tasks 3.2 and 3.3) are implemented as backend components integrated within the CARMONY Interlinked Orchestrator. In the context of this initial version deliverable, the focus is placed on the solutions developed for UC6*, and more specifically on the preventive layer and real time layer. These elements provide the analytical and decision-support capabilities required to identify evolving traffic situations, detect early signs of potential disruptions, and support the generation of proactive mitigation measures.

To support the demonstration of these backend functions, a dedicated frontend environment has also been developed. This environment illustrates the outputs produced by the software components and enables user interaction with the developed functionalities. It provides a practical interface for visualising identified traffic situations, detected patterns, potential incidents, and proposed mitigation measures, thereby supporting both technical testing and stakeholder-oriented demonstrations.

Although the integration activities will continue to be tested, refined, and fine-tuned over the coming months, an initial draft of the API definitions has already been prepared. These preliminary interface specifications establish the basis for communication between the CARMONY Interlinked Orchestrator, the backend services, and the frontend environment. They define the expected structure of the exchanged information, the main service interactions, and the format of the data inputs and outputs.

The following subsection presents the available information on these interfaces, including the current draft API structures and their intended role in enabling the exchange of information between the different software modules developed for the UC6* preventive layer and reactive layer.

** Use Case 6 uses simulation-based evaluations to assess how the Interlinked Orchestrator can prevent or dissipate traffic congestion on Luxembourg's A3 motorway by harmonising vehicle speeds before traffic flow breaks down.*

3.2 List of modules in alignment with the architecture in D3.1

In consistency with the architecture defined in deliverable D3.1, the software developments performed with tasks T3.2 and T3.3 mainly concern six functional modules of Carmony orchestrator architecture:

- Decision logic
- Traffic situation identification
- Database of known decisions
- Route planner
- Simulations situation database
- Fleet Operator: Car Sharing Optimization

DRAFT

We also present an additional module not highlighted in the initial architecture diagram of D3.1, that of the Traffic Forecasting module.

3.3 Main functionalities of each module

3.3.1 Decision logic

The decision logic module mainly supports the preventive layer of the orchestrator.

It integrates software components dedicated to:

- Traffic pattern analysis and recognition
- Prediction of traffic degradation situations
- Generation of preventive mitigation measures
- Evaluation of possible regulations strategies through simulation-based approaches

This module uses historical traffic data, predefined baseline scenarios and simulations results to support preparation of proactive traffic management actions

3.3.2 Traffic situation identification

The traffic situation identification module mainly supports the reactive layer of the orchestrator.

It integrates software components dedicated to:

- Real-time traffic degradation detection
- Identification of current traffic situation
- Triggering of reactive mitigation measures
- Interaction with real-time simulations module for evaluation of alternative actions

This module relies on real-time traffic data to identify traffic situations requiring operational mitigation actions

3.3.3 Database with known decisions

The database with known decisions modules is used to store and manage previously evaluated traffic management strategies generated within the preventive layer.

This module stores:

- Analysed traffic situations
- Evaluated mitigation measures
- Associated simulation results and KPIs
- Ranked decision strategies

3.3.4 Traffic Forecasting

The Traffic Forecasting module is used to produce estimates of traffic conditions in known locations. It employs models that have been calibrated on historical data, and allows users to generate traffic estimates using minimal input variables. The Traffic Forecasting module does

DRAFT

not store any data apart from the pre-trained models used for the predictions, but instead it is deployed and exposes a web API so that users can make requests for traffic estimates.

3.3.5 Route planner

The Route Planner module is used to generate trip suggestions within a given transport network. It is part of the preventive layer of the orchestrator. It is available via a web API exposing an interface that receives trip requirements (origin/destination, modes of transport, etc) and calculates valid trip suggestions fulfilling all trip requirements. The route planner may return multiple trip suggestions, each optimized for a particular parameter (e.g. faster, least distance travelled, etc.) Simulation situation database.

The Simulation Situation Database is a module located within the preventive layer of the orchestrator, with the purpose of storing traffic scenarios that need to be simulated at a later time. The traffic scenarios stored in the Database are codified and homogenized for consistency with other systems of the CARMONY orchestrator, and to allow for comparisons between scenarios. Fleet Operator: Car Sharing Optimizer (CSO).

Fleet operators should also benefit from the harmonization of traffic in systems like CARMONY and, at the same time, contribute to optimizing traffic in cities. In this context, the Car Sharing Optimizer is designed to resolve service demand through an assignment mechanism that minimizes fleet operational expenditure (OPEX) while simultaneously maximizing the Quality of Service (QoS) utilizing suggested routes retrieved from the TMC´s CARMONY component. The precise definition of the cost function remains under evaluation, while the aggregate route distance will serve as the initial baseline, the integration of multi-objective parameters, such as passenger waiting times and emissions might be considered. Thus, this component focuses on a holistic fleet optimization approach rather than treating vehicles as independent agents.

3.4 Interfaces with other orchestrator modules

3.4.1 Real Time/Low Fidelity Simulation

The real-time, low-fidelity simulation is part of the reactive layer of the orchestrator. It is triggered when no suitable decision is available in the database of known decisions for the current traffic situation. The purpose of this simulation is to rapidly evaluate the potential effects of proposed mitigation measures and regulatory actions generated by the decision logic before their application to real traffic or, in the case of pure simulation, to the environment simulation.

3.4.2 Monthly/High Fidelity Simulation

The monthly, high-fidelity simulation is part of the preventive layer of the orchestrator. It is used to re-simulate previously observed traffic situations stored in the Simulation Situation Database. The objective is to evaluate and validate new regulatory strategies and, where appropriate, update or extend the database of known decisions with improved solution.

3.4.3 Environment Simulation

The environment simulation represents the traffic environment in scenarios where no real-world traffic data is used. In such cases, it provides the traffic state on which the orchestrator operates. Regulatory measures, such as speed limits or carpool lane availability, are applied within this simulated environment. The environment simulation interacts with the orchestrator

DRAFT

in a closed-loop manner, enabling continuous feedback between traffic conditions and control actions.

3.4.4 Traffic Management Center (TMC)

Regarding traffic network utilization, two primary elements interact with the demand and supply sides of the CSO system. Firstly, the individual routing for each vehicle may be integrated within the TMC, generating multiple trajectory alternatives for each request. These alternatives serve as primary inputs for the CSO, enabling the optimization of trip concatenation for each vehicle within the fleet.

Secondly, the TMC is designed to interact with both supply and demand sides. In terms of supply, the TMC reports real-time traffic conditions and the occurrence of traffic incidents. This data should be utilized by the CSO system to incorporate deviations from the planned scenario.

Furthermore, the TMC might provide information on potential demand-generating nodes, identified either through active incidents or through historical analysis of travel requests via integrated applications. Should the connection between the TMC and the demand simulator be unavailable, this functionality might be managed through a demand predictor.

DRAFT

4. Software functional scope

4.1 Decomposition of functional modules into SW components

The functional modules identified in alignment with D3.1 architecture are decomposed into software components supporting both preventive and reactive layers of the Carmony orchestrator

Table 1: Decomposition of functional modules into SW components

Functional module	Software component	Sub Software component	Main role	leader
Decision logic	Predicted Jam Identification (DL-PJI)	-	Identifies potential future traffic degradation situations based on historical data and baseline scenarios	HYP
	Predicted Speed Regulation (DL-PSR)	DL-PSR-Actions	Generates preventive speed regulation recommendations for predicted traffic degradation situations	HYP
		DL-PSR-KPI analysis	Analyses and ranks simulation results associated with preventive speed regulation strategies	HYP
	Predicted Dynamic Slot creation for carpool lane (DL-PDS)	DL-PDS-Actions	Generates preventive dynamic slot creation strategies for carpool lane management	HYP
		DL-PDS-KPI-Analysis	Analyses and ranks simulation results associated with preventive dynamic slot creation strategies	HYP
Traffic situation identification	Real-time Jam Detection (TSI-RTJD)	-	Identifies and characterises traffic degradation situations using real-time traffic data	HYP
	Real-time Speed Regulation (TSI-RTSR)	TSI-RTSR-Actions	Generates reactive speed regulation recommendations based on the current traffic situation	HYP
		TSI-RTSR-KPI Analysis	Analyses and evaluates the performance of reactive speed	HYP

DRAFT

Functional module	Software component	Sub Software component	Main role	leader
			regulation strategies using real-time simulation results	
	Real-time Dynamic Slot creation (TSI-RTDS)	TSI-RTDS-Actions	Generates reactive dynamic slot creation strategies based on the current traffic situation	HYP
		TSI-RTDS-KPI Analysis	Analyses and evaluates the performance of reactive dynamic slot creation strategies using real-time simulation results	HYP
Database with known decisions	Decision Storage Service (DKD-DSS)	-	Stores evaluated traffic situation, proposed actions, simulations results and KPIs	HYP
	Decision Retrieval Service (DKD-DRS)	-	Retrieve evaluated decisions when a similar traffic situation is identified	HYP
	Decision Ranking and Metadata service (DKD-DRM)	-	Organises stored decisions according to KPIs, scenario metadata and applicability conditions	HYP
Traffic Forecasting	Traffic forecasting component	-	Provides estimates for traffic conditions using pre-trained forecasting models	FRON
Route planner	Route planning engine	-	Calculates optimal route suggestions within a given transportation network	FRON
Simulation situation database	Traffic scenarios database	-	Stores traffic scenarios in encoded format to be tested in simulation environments at a later time.	FRON
Fleet Operator (CSO)	Demand Predictor	-	Optimizes vehicle fleet based on recommended routes from the Orchestrator, through a multi-objective optimization framework	TEC
	CSO Optimizer Fleet Simulator	-		

DRAFT

Functional module	Software component	Sub Software component	Main role	leader
	Demand Simulation			

4.2 Internal architecture of each module and interaction between SW components

4.2.1 Decision Logic & Traffic Situation Identification internal architecture

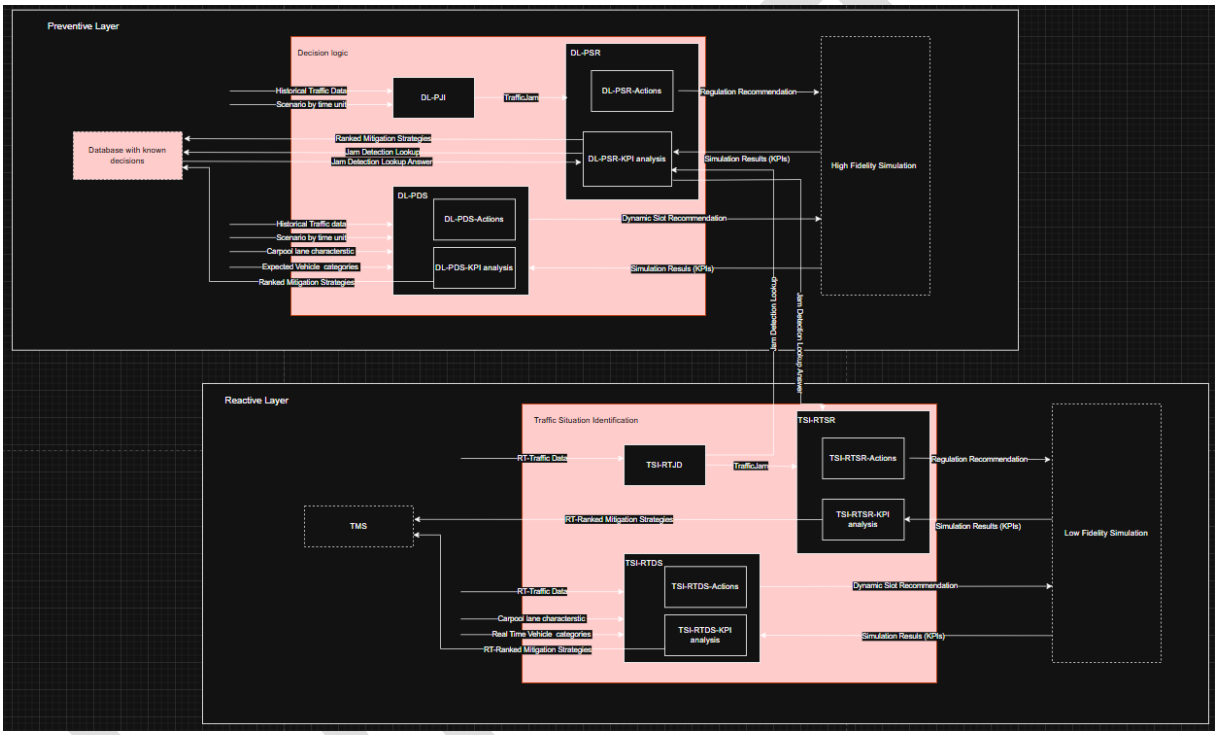


Figure 1: Internal architecture of the Decision Logic and Traffic Situation Identification components. The preventive layer uses historical data to anticipate traffic degradations and propose preventive actions, while the reactive layer processes real-time data to detect emerging situations and trigger appropriate responses.

4.2.2 Traffic Forecasting architecture

The Traffic Forecasting module implements a simple architecture, consisting of three functional components (user interface, data handling, and predictors) all communicating with one another. The three components are split across two layers, the public layer that is exposed to users, and the private layer that contains all the operational software components. The traffic Forecasting module architecture is shown in Figure 2.

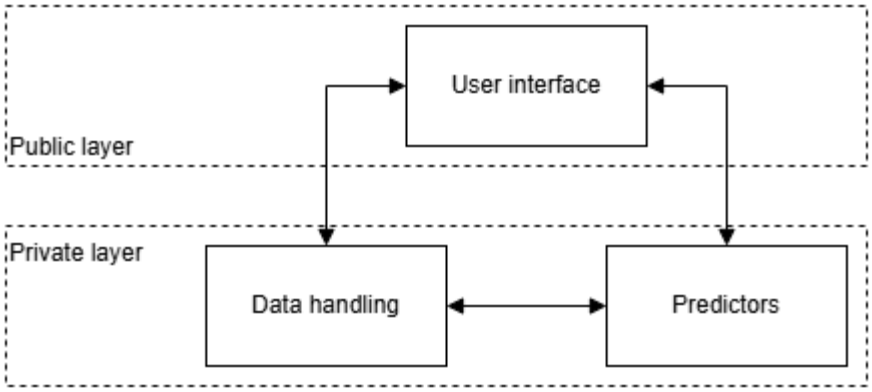


Figure 2: Conceptual architecture of the Traffic Forecasting module

4.2.3 Route Planner architecture

The Route Planner module follows a simple architecture with three functional components: The Network component, the Pathfinding module component, and the User interface. The overall architecture is presented in Figure 3. The Network component acts as a database that holds the configuration of the transportation network and its current conditions and receives input in the form of network conditions. The pathfinding module contains all the logic for calculating routes and retrieves the current state of the network from the Network component. The User interface allows external systems to access the other two components.

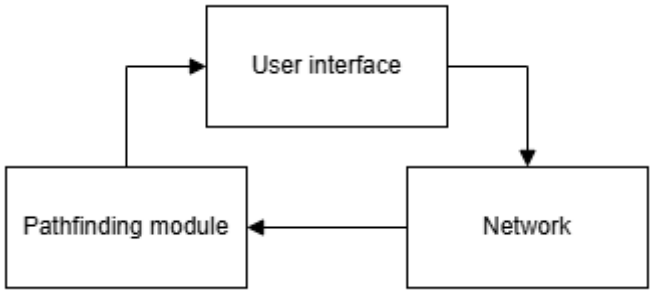


Figure 3: Conceptual architecture of the Route Planner module

4.2.4 Simulation Situation Database architecture

The Simulation Situation Database module follows a simple architecture of three functional components. The three functional components are: a Public API that acts as the user interface with users, a Database with required controllers, and a scenario comparer and ranker that filters elements from the database. The overall architecture is shown in Figure 4.

DRAFT

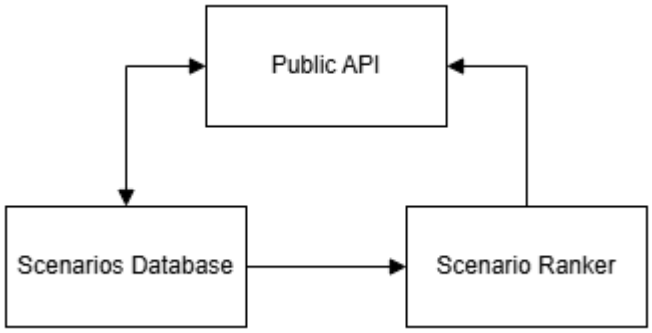


Figure 4: Conceptual architecture of the Simulation Situation Database module

4.3 Description of internal data flows and SW components outputs

This section provides a description of the main internal data flows and SW outputs exchanged between HYP SW components and orchestrator modules.

4.3.1 TrafficJam flow

The TrafficJam flow represents traffic degradation information exchanged between SW components.

In the preventive layer, the TrafficJam flow is exchanged between DL-PJI and DL-PSR-Actions, it represents a predicted traffic degradation situation based on historical data and baseline scenarios.

In the reactive layer, the TrafficJam flow is exchanged between TSI-RTJD and TSI-RTSR-Actions. It represents a detected traffic degradation situation based on real-time data.

The main information included in TrafficJam flow:

- Affected direction
- Congestion score representing the estimated level of traffic degradation
- Jam length
- Jam duration

The detailed structure and proprieties of TrafficJam flow are presented on 5.2.3.3.

4.3.2 RegulationRecommendation flow

The RegulationRecommendation flow represents speed regulation measures.

In the preventive layer, the flow is generated by DL-PSR-Actions and transmitted to the Monthly Simulations (external module)

In the reactive layer, the flow is generated by TSI-RTSR-Actions and transmitted to the RT-Simulations external module.

The main information included in SpeedRegulationRecommendation flow:

- Affected direction
- Regulation speed value
- Regulation zone length

DRAFT

- Regulation duration

The detailed structure and properties of RegulationRecommendation flow are presented on §6.2

4.3.3 DynamicSlotRecommendation flow

The DynamicSlotRecommendation flow represents a proposed dynamic slot configuration generated for carpool lane management

In the preventive layer, the flow is generated by DL-PDS-Actions and transmitted to the Monthly Simulations (external module)

In the reactive layer, the flow is generated by TSI-RTDS-Actions and transmitted to the RT-Simulations (external module)

The main information included in DynamicSlotRecommendation flow:

- Affected road direction
- Proposed slot activation period
- Slot creation configuration
- Slots numbers
- Slots sizes
- Spacing
- Headway

The detailed structure and proprieties of DynamicSlotRecommendation flow are not presented on the initial version of D3.2 and will be further matured in the final version of D3.5.

4.3.4 JamDetectionLookup flow

The JamDetectionLookup flow represents a request sent from the reactive layer to the Database with known decisions in order to identify whether a similar traffic degradation situation has already been analysed in the preventive layer.

This flow is generated when a traffic jam is detected in real time by TSI-RTJD and when the system needs to check whether a previously evaluated mitigation strategy is available.

The mean information included is the JamDetectionLookup flow:

- Affected road direction
- Detected traffic degradation context
- Jam characteristics
- Time context of the detected situation

The detailed structure and proprieties of JamDetectionLookup flow are not presented on the initial version of D3.2 and will be further matured in the final version of D3.5.

4.3.5 JamDetectionLookupAnswer flow

The JamDetectionLookupAnswer flow represents the response provided by the Database with known decisions after lookup request from reactive layer.

DRAFT

This flow indicates whether one or more relevant previously evaluated mitigation strategies are available for the detected traffic situation.

The main information included in the JamDetectionLookupAnswer flow:

- Lookup results status
- Reference to similar known traffic situations
- Availability of ranked mitigation strategies

The detailed structure and proprieties of JamDetectionLookupAnswer flow are not presented on the initial version of D3.2 and will be further matured in the final version of D3.5.

4.3.6 RankedMitigationStrategies flow

The RankedMitigationStrategies flow represents the output of the preventive KPI analysis components after evaluation of proposed mitigation strategies through monthly simulations.

This flow generated by:

- DL-PSR-KPI Analysis for preventive regulation speed strategies.
- DL-PDS-KPI Analysis for preventive dynamic slot strategies.

The flow is transmitted to the Database with known decisions in order to store evaluated strategies and support future reuse.

The main information included in the RankedMitigationStrategies flow:

- Reference to the analysed traffic situation
- Mitigation strategy type
- KPI summary
- Ranking information

The detailed structure and proprieties of RankedMitigationStrategies flow are not presented on the initial version of D3.2 and will be further matured in the final version of D3.5.

4.3.7 RT-RankedMitigationStrategies flow

The RT-RankedMitigationStrategies flow represents the output of the reactive KPI analysis components after evaluation of candidate mitigations strategies through real-time simulation.

This flow is generated by:

- TSI-RTSR-KPI Analysis for reactive speed regulation strategies
- TSI-RTDS-KPI Analysis for reactive dynamic slot strategies

The flow is transmitted to the TMS in order to support operational traffic management decision (Human in the loop)

The main information included in the Rt-RankedMitigationStrategies flow:

- Reference to the current traffic situation
- Mitigation strategy type
- KPI summary
- Ranking information

DRAFT

The detailed structure and proprieties of RT-RankedMitigationStrategies flow are not presented on the initial version of D3.2 and will be further matured in the final version of D3.5

4.3.8 Traffic forecasting internal data flows

A flowchart that shows the Traffic Forecasting operations pipeline and interactions between components is shown in Figure 2.

The request parameters include a timestamp in UNIX time corresponding to the time for which the prediction is needed, and a location for which the prediction is required. These are first validated and then converted to model input data formats (i.e. the timestamp broken down to year/month/day/hour/minute in local time). Once the data is prepared, the appropriate model is loaded that corresponds to the requested traffic variable, and a prediction is generated based on the input data. The estimate is then returned to the user in JSON format.

For an instance of the Traffic Forecasting module to be deployed for a new location, a set of models needs to be trained beforehand on traffic data for that location. Once calibrated, these models can provide estimates of traffic conditions for that location. The process of calibrating the models is presented in detail in 8.2.1, using the Luxemburg A3 highway as a testbed.

4.3.9 Route planner internal data flows

The Route Planner implements a single core flow, supporting the request for route suggestions. The data flow for this request is presented in Figure 3.

4.3.10 Simulation situation database internal data flows

There are two core flows in the Simulation Situation Database module, one for adding a scenario in the database and one for retrieving scenarios from the database. Both flows are implemented using straightforward data pipeline.

Figure 3 demonstrates the process for adding a new scenario to the database, which is performed using an HTTP POST request. The module first verifies that the data payload containing the scenario information is valid, proceeds with storing the scenario in the database, and returns a message with the operation result (success or fail).

Figure 4 provides an overview for the process of retrieving scenarios from the database. The retrieval of scenarios is performed using an HTTP GET request, and additional parameters can be provided in the request to filter the returned scenarios. The list of filter parameters has not been finalized yet at this stage, but indicatively may include: datetime from/to, type of day (weekday/weekend/only specific day), weather conditions, specific traffic conditions (e.g. increased traffic). Once the valid scenarios are retrieved from the database, they are additionally ranked according to any provided metric (ranking metrics have not been finalized yet at this stage), before returned to the caller.

DRAFT

5. SW technical scope

5.1 Technology stack of each SW component

This section provides an overview of the technology stack and execution characteristics of the SW components developed by HYP within the preventive and reactive layers of the Carmony architecture.

Decision logic SW components

The decision logic SW components include:

- DL-PJI
- DL-PSR (DL-PSR-Actions and DL-PSR-KPI Analysis)
- DL-PDS (DL-PDS-Actions and DL-PDS-KPI Analysis)

Table 2: Technology stack of Decision Logic SW components

Characteristic	Description
Programming Language & Runtime	C# .Net 8
Execution Model	externally triggered
Base step size/execution frequency	on demand (externally triggered)
Time control	internal timer
Real-time capable	not real-time
Interface/deployment type	REST API

Traffic situation identification SW components

The Traffic situation identification SW components include:

- TSI-RTJD
- TSI-RTSR (TSI-RTSR-Actions and TSI-RTSR-KPI Analysis)
- TSI-RTDS (TSI-RTDS-Actions and TSI-RTDS-KPI Analysis)

Table 3: Technology stack of Traffic Situation Identification SW components

Characteristic	Description
Programming Language & Runtime	C# .Net 8
Execution Model	fixed-step
Base step size/execution frequency	>1 minute
Time control	internal timer
Real-time capable	Soft real-time

DRAFT

Characteristic	Description
Interface/deployment type	REST API

Database with know decisions :

The Database with known decisions services mainly include:

- DKD-DSS
- DKD-DRS
- DKD-DRM

Table 4: Technology stack of Database with Know Decisions

Characteristic	Description
Programming Language & Runtime	C# .NET 8 (backend service)
Execution Model	request-driven (REST API)
Base step size/execution frequency	On demand
Time control	Controlled by incoming requests
Real-time capable	Not real-time
Interface/deployment type	Relational database: PostgreSQL REST API Hosting: Linux VPS with Nginx

Traffic Forecasting SW component:

Table 5: Technology stack of Traffic Forecasting SW component

Characteristic	Description
Programming Language & Runtime	python 3.13.9
Execution Model	request-driven (REST API)
Base step size/execution frequency	On demand
Time control	Controlled by incoming requests
Real-time capable	real-time
Interface/deployment type	Interface: RestAPI (using fastAPI) Deployment: containerized module (using docker containers)

DRAFT

5.2 Preliminary API configuration and characteristics

This section presents the preliminary API configurations currently implemented for the HYP SW components.

At the current development stage, only a limited subset of API endpoints is exposed in this initial version of D3.2.

5.2.1 Current Endpoints of Decision Logic and Traffic Situation Identification

The following endpoints are currently structured:

Table 6: List of current endpoints of Decision logic and Traffic Situation Identification

Method	Endpoint	Main Purpose
GET	api/v1/predicted traffic	Returns predicted traffic situations and preventive speed regulation recommendations based on historical traffic data and scenarios
GET	api/v1/doc	Returns the list of supported parameters for the different endpoints, along with usage instructions and additional guidance on how to use the API and its endpoints
GET	api/v1/real-time	Returns the current traffic situation and associated reactive speed regulation recommendations based on real-time traffic data

The *api/v1/predicted-traffic* endpoint exposes information generated by

- DL-PJI through TrafficJam structures
- DL-PSR-Actions through RegulationRecommendation structures

The *api/v1/real-time* edpoint exposes information generated by

- TSI-RTJD through TrafficJam structures
- TSI-RTSR-Actions through RegulationRecommendation structures

DRAFT

5.2.1.1 GET api/v1/predicted-traffic

Description:

Returns a representative traffic state for the requested day and time period, based on historical traffic data, together with the currently detected TrafficJam objects and their associated RegulationRecommendation object.

Query params:



Json Response

```
{
   "trafficJams": TrafficJam[]
   "regulationRecommendations": RegulationRecommendation[]
}
```

5.2.1.2 GET api/v1/real-time

Description:

Return the most recent snapshot, current TrafficJams and associated RegulationRecommendations.

Query params: NONE

Json Response:

```
{
   "snapshot": Snapshot
   "trafficJams": TrafficJam[]
   "regulationRecommendations": RegulationRecommendation[]
}
```

5.2.2 Traffic Forecasting endpoints

The endpoints currently implemented for the Traffic Forecasting module are shown in Table 7.

Table 7: Description of the Traffic Forecasting endpoints

Method	Endpoint	Main Purpose
GET	/health	Auxiliary endpoint, used to verify that the service is operational
GET	/help	Auxiliary endpoint, returns a list of all endpoints available via the Traffic Forecasting module and additional help
GET	/forecast/flow/at_time/	Returns a prediction of traffic flow for the specified time

DRAFT

GET	/forecast/speed/at_time/	Returns a prediction of traffic speed for the specified time
GET	/forecast/density/at_time/	Returns a prediction of traffic density for the specified time

5.2.2.1 /forecast/flow/at_time/

Description:

Returns the predicted traffic flow for a given time of day at a specified location.

Query params:

timestamp: required The timestamp in UNIX time for which to get predicted traffic flow

station_id: required The id of the station for which to generate a prediction, as an integer

direction: required The street direction for which to generate a prediction, as a string. Expected values: “mv” or “vm”

Json Response:


```
{
  "properties": {
    "station": 110,
    "direction": "mv",
    "prediction_time": "2026-06-23T12:45:15+02:00"
  },
  "flow": {
    "value": 62.91619833551873,
    "unit": "vehicles per 5 minutes"
  }
}
```


5.2.2.1 /forecast/speed/at_time/

Description:

Returns the predicted traffic speed for a given time of day at a specified location.

Query params:

timestamp: required The timestamp in UNIX time for which to get predicted traffic speed.

station_id: required The id of the station for which to generate a prediction, as an integer

direction: required The street direction for which to generate a prediction, as a string. Expected values: “mv” or “vm”

DRAFT

Json Response:


```
{
  "properties": {
    "station": 110,
    "direction": "mv",
    "prediction_time": "2026-06-23T12:45:15+02:00"
  },
  "average_speed": {
    "value": 69.91741363420154,
    "unit": "km per hour"
  }
}
```


5.2.2.2 /forecast/density/at_time/

Description:

Returns the predicted traffic density for a given time of day at a specified location.

Query params:

timestamp: required The timestamp in UNIX time for which to get predicted traffic density.

station_id: required The id of the station for which to generate a prediction, as an integer

direction: required The street direction for which to generate a prediction, as a string. Expected values: “mv” or “vm”

Json Response:


```
{
  "properties": {
    "station": 110,
    "direction": "mv",
    "prediction_time": "2026-06-23T16:12:26+02:00"
  },
  "density": {
    "value": 8.845545435334136,
    "unit": "vehicles per km"
  }
}
```


DRAFT

5.2.3 Data Structure (SCHEMAS)

5.2.3.1 Snapshot

The Snapshoot structure represents collection of segment histories for a given date and time

Field	Type	Description	Example
id	integer	Unique identifier.	4242
sourceUpdatedAt	string	Timestamp indicating when the snapshot was taken.	"2026-06-02T14:08:00"
SegmentHistories	Schema.SegmentHistory[]	List of SegmentHistory for the given date and time.	[{"id": 8558445, congestionScore: 0.43, "pkStart": 12542, ...}, ...]

5.2.3.2 SegmentHistory

The SegmentHistory structure represents traffic data for a specific road segment (from one PK) at a given time.

Field	Type	Description	Example
id	integer	Unique identifier.	4242
sourceUpdatedAt	string	Timestamp indicating when the SegmentHistory was taken.	"2026-06-02T14:08:00"
direction	string	"mv" | "vm"	"mv"
pkStart	integer	Starting PK of the segment (in meters).	4200
pkEnd	integer	Ending PK of the segment (in meters).	8246
length	integer	Length of the segment (in meters).	4046
lanesCount	integer	Number of lanes.	3
speed	number	Current speed in km/h	92.8
speedLimit	number	Speed limit in km/h	130
flowRate	number ?	Vehicles per hour	76.3
occupacity	number ?	Percetage of occupation	23
congestionScore	number	From 0 (no congestion) to 1 (maximum congestion).	0.43

5.2.3.3 TrafficJam

The TrafficJam represents one or more consecutive congested road segments form a TrafficJam.

Field	Type	Description	Example
id	integer	Unique identifier.	4242
direction	string	"mv" | "vm"	"mv"
pkStart	integer	Start PK of the traffic jam area (in meters).	4200
pkEnd	integer	End PK of the traffic jam area (in meters).	8246
length	integer	Length of TrafficJam (in meters).	4046
startTime	string	Indicating when the traffic jam starts.	"08:00:00"
speed	number	Current speed in the jammed area in km/h	92.8
congestionScore	number	From 0 (no congestion) to 1 (maximum congestion).	0.43

5.2.3.4 RegulationRecommendation

The RegulationRecommendation structure represents area (from pkStart to pkEnd) to regulate.

Field	Type	Description	Example
id	integer	Unique identifier.	4242
trafficJamId	number	Identifier of the parent traffic jam.	1
direction	string	"mv" | "vm"	"mv"
pkStart	integer	Start PK of the regulation area (in meters).	4200
pkEnd	integer	End PK of the regulation area (in meters).	8246
length	integer	Length of the regulation area (in meters).	4046
startsAt	string	Indicating when the regulation starts.	"08:00"
endsAt	string	Indicating when the regulation ends.	"08:35"
recommendedSpeed	number	Regulation Speed recommendation in km/h	72.8

6. Progress status of SW component development

The following table presents the current maturity and implementation status of the main SW components of T3.2 and T3.1 at the time of initial version of D3.2.

The presented status reflects the current progress of the functional definition, SW development and internal unit testing activities.

Integration activities with Carmony interlinked orchestrator remain dependent on the availability and stabilization of interfaces and interconnections.

Table 8: Progress status of SW component development

SW component	Partner	Functional Status	Development status	Integration and Test status
DL-PJI	HYP	Matured	Advanced Development	Preliminary API and internal tests
DL-PSR-Actions	HYP	Matured	Advanced Development	Preliminary API and internal tests
DL-PSR-KPI analysis	HYP	Defined	On going	Not integrated
DL-PDS-Actions	HYP	Partially defined	Not Started	Not Started
DL-PDS-KPI-Analysis	HYP	Partially defined	Not Started	Not Started
TSI-RTJD	HYP	Matured	Advanced Development	Preliminary API and internal tests
TSI-RTSR-Actions	HYP	Matured	Advanced Development	Preliminary API and internal tests
TSI-RTSR-KPI Analysis	HYP	Defined	On going	Not integrated
TSI-RTDS-Actions	HYP	Partially defined	Not Started	Not Started
TSI-RTDS-KPI Analysis	HYP	Partially defined	Not Started	Not Started

DRAFT

SW component	Partner	Functional Status	Development status	Integration and Test status
Traffic forecasting	FRON	Defined	Advanced Development	Preliminary API and internal tests
Route planner	FRON	Defined	Not started	Not started
Simulation situation database	FRON	Defined	Not started	Not started
Car Sharing Optimization	TEC	Architecture and functional aspects defined	Defining data modeling, interface with TMC component defined.	Not Started

DRAFT

7. Interfaces and integration with Carmony (HYP)

7.1 Context overview

To facilitate the demonstration of the software components developed within Tasks T3.2 and T3.3, Hypervisoul has developed a web-based frontend.

This frontend is intended exclusively for demonstration, and visualization purposes. It provides access to the outputs generated by the Traffic Flow Regulation backend services and allows stakeholders to explore the different functionalities implemented during the project.

The web-based frontend provides:

- Visualization of the road network topology used by the Traffic Flow Regulation services.
- Visualization of historical and real-time traffic data.
- Visualization of detected and predicted traffic degradations.
- Visualization of speed regulation recommendations.
- API documentation and endpoints descriptions.
- API testing capabilities.
- API key management functionalities.

It should be noted that this frontend does not represent the target operational deployment of the Carmony solution. In the final Carmony architecture, the software modules developed in Tasks T3.2 and T3.3 will be integrated with the other components of the Carmony Interlinked Orchestrator and will interact through standardized APIs. Consequently, the final delivery focuses on the backend services and their APIs, while the frontend presented in this document is provided solely as a temporary demonstration and validation tool.

7.2 User guide for the web-based frontend

7.2.1 Frontend URL

https://traffic-flow-regulation.hypervisoul.com/

7.2.2 Create an account

For your first connection to the Traffic Flow Regulation platform, you must create an account.

1. Open the Login/Register menu on the left side of the screen.
2. Click Register.
3. Enter your email address.
4. Choose and confirm your password.
5. Accept the Terms and Conditions regarding data protection.
6. Click Create Account.

After submitting code :registration form, a confirmation email will be sent to the provided email address.

DRAFT

Once the confirmation email has been received and your account has been activated, you can log in using your credentials.

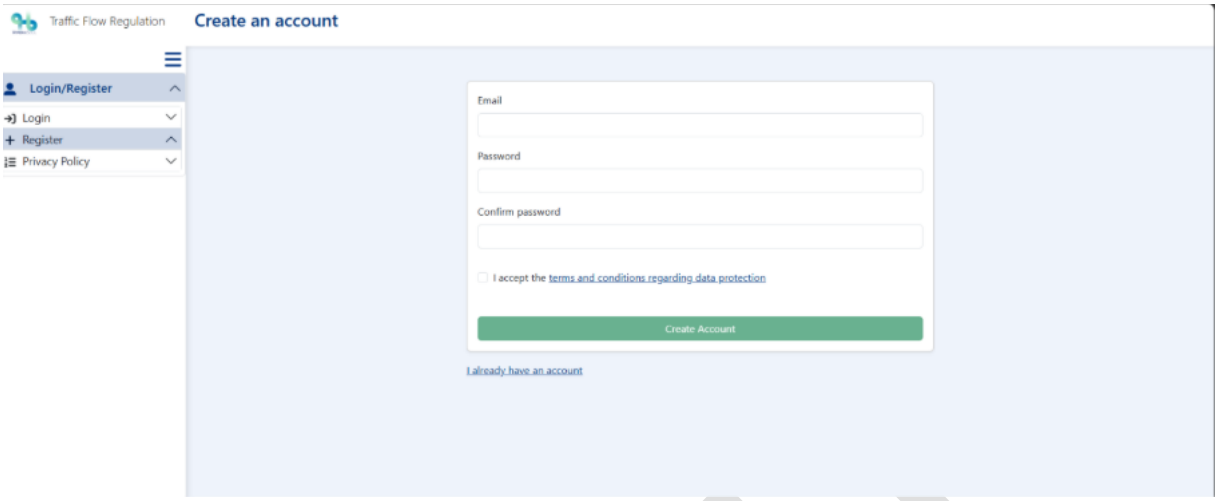


Figure 5: Account registration page

7.2.3 Global navigation

The web-based frontend is designed to support road operators in the preparation and operation of traffic orchestration strategies on the A3 motorway corridor in Luxembourg.

Before real-time operation of the orchestration system, two main preparatory activities are required:

- Configuration of the road network model used by the orchestration system.
- Integration of traffic data collected from the road operator infrastructure.

After logging in, the user is redirected to the Home page of the Traffic Flow Regulation platform.

The main navigation menu is located on the left-hand side of the screen and provides access to all available functionalities.

The navigation structure is organized as follows:

- User Account: access to user profiles and account-related functions.
- API: access to the API documentation and related services.
- Demo: demonstration environment containing:
    - Dashboard: An overview of the A3 motorway corridor configuration, including the road network structure and its main characteristics.
    - Input Data: access to the raw traffic data collected from the road operator infrastructure and used by the orchestration system.
    - Real Time: monitoring of the current traffic conditions and associated orchestration outputs
    - Exploitation Projects: preparation and management of preventive traffic regulation strategies based on day types.

The navigation menu can be expanded or collapsed using the menu icon located at the top of the left panel.

DRAFT

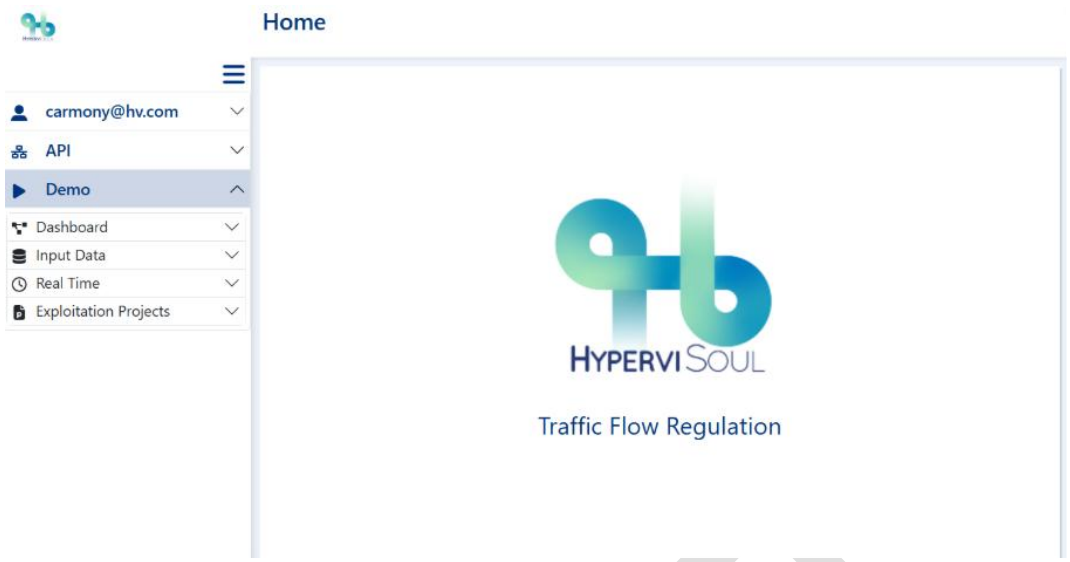


Figure 6: Global navigation menu

7.2.4 Platform features

The platform is structured around two features:

API Management and Documentation

Provides complete API documentation, endpoint descriptions, schema definitions, and authentication management.

Data Visualization & Demo

Provides interactive visualization tools to explore the A3 motorway network configuration, traffic data, real-time traffic conditions, congestion detection results, and traffic regulation recommendations.

7.2.4.1 Feature 1: API Management & Documentation

This section describes the REST API endpoints exposed by the Carmony software to support data exchange with other systems within the Carmony orchestrator. This API is intended to provide structured access to the network topology, historical and real-time traffic data, detected congestion situations, predicted traffic conditions, and speed regulation recommendations.

The API responses are based on a set of shared data schemas. The main real-time schemas are Snapshot, SegmentHistory, TrafficJam, and RegulationRecommendation. A Snapshot represents the traffic state at a given date and time, composed of several SegmentHistory records. Each SegmentHistory describes the traffic conditions on a specific road segment, including direction, PK position, speed, speed limit, congestion score, lane configuration, and other traffic indicators. Consecutive congested segments may be grouped into a TrafficJam, while the system may generate one or more RegulationRecommendation objects defining the road section and speed value proposed to mitigate the detected or predicted congestion.

All distances and positions are expressed using the Carmony PK reference system. Traffic speeds are expressed in km/h. The CongestionScore is expressed on a scale from 0 to 1, where 0 represents no congestion and 1 represents maximum congestion. Directions are represented using the values mv and vm.

DRAFT

This API is read-oriented and is intended to provide consistent traffic state information and recommendations to the rest of the Carmony ecosystem. The API is designed to support data exchange between the different components of the Carmony orchestration framework and external consumer systems.

The Base API URL is: https://traffic-flow-regulation-back.hypervisoul.com/api/v1

7.2.4.1.1 API Documentation

The API Documentation page serves as the central entry point for understanding and using our Traffic Flow Regulation API.

Key Information:

- Base URL: The base address for all API endpoints (https://traffic-flow-regulation-back.hypervisoul.com/api/v1)
- API Key Explanation: Learn how to authenticate your requests using API keys and how they secure your API calls
- Quick reference for all available endpoints.

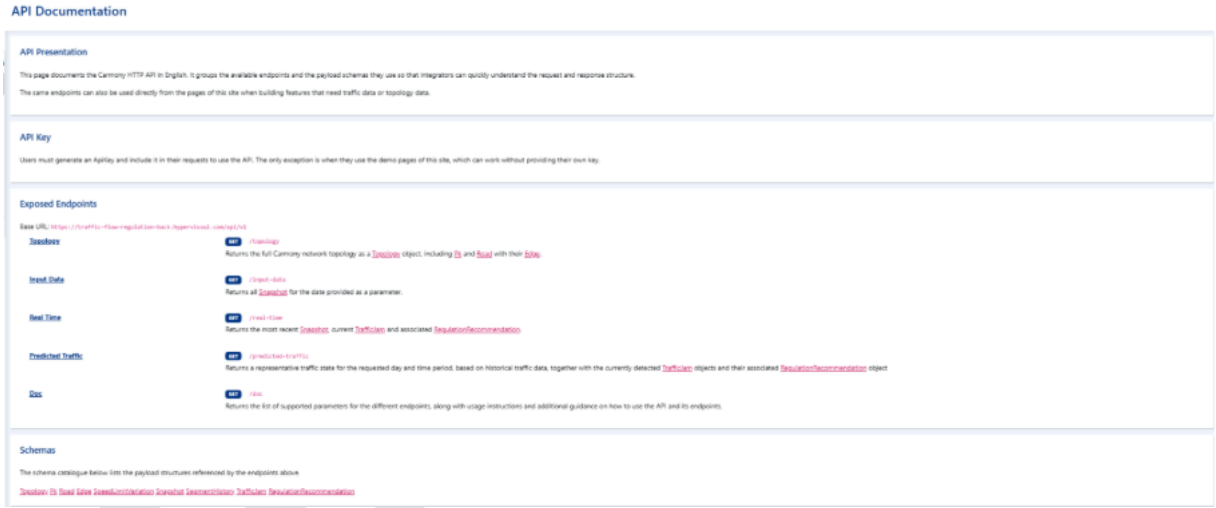


Figure 7: API documentation page

7.2.4.1.2 API keys Management

The API Keys page allows you to manage your authentication credentials.

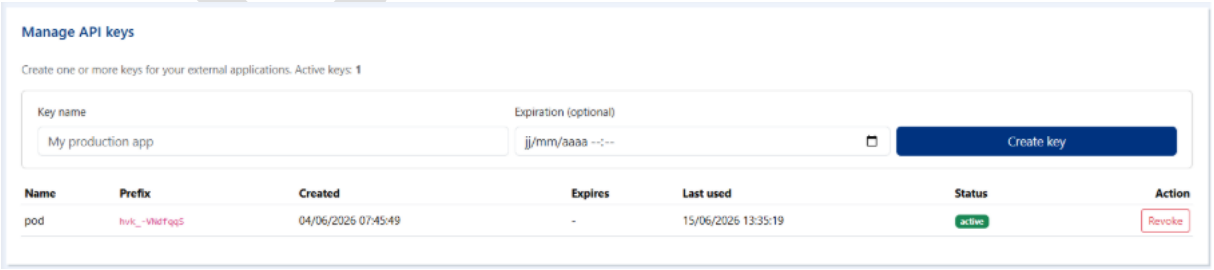


Figure 8: API Key Management Page

Available Actions:

- Create new API keys for different applications or use cases
- Revoke or delete API keys when they are no longer needed

7.2.4.1.3 Endpoint Documentation Pages

The platform provides dedicated documentation pages for each of our four main endpoints.

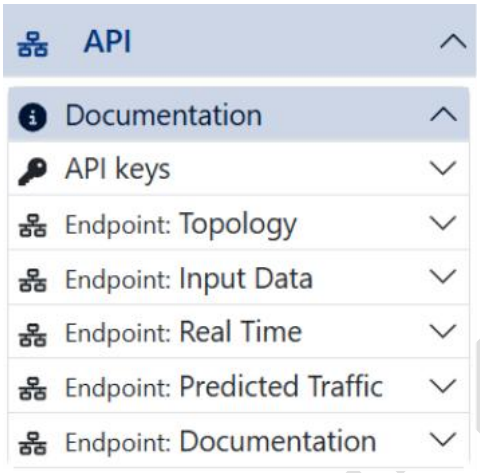


Figure 9: API Endpoints Menu

Endpoints of the API:

- The GET /topology endpoint returns the complete Carmony network topology, including PK reference points and roads with their associated edges, as a structured Topology object.
- The GET /input-data endpoint returns all available Snapshot objects for the date provided as a request parameter.
- The GET v1/real-time endpoint returns the latest Snapshot, the currently detected TrafficJam objects, and their associated RegulationRecommendation objects.
- The GET /predicted-traffic endpoint returns a representative traffic state for the requested day and time, based on historical traffic data, together with the currently detected TrafficJam objects and their associated RegulationRecommendation objects
- The GET /doc endpoint returns the list of supported parameters for the different endpoints, along with usage instructions and additional guidance on how to use the API and its endpoints.

For each endpoint, it is possible to:

- View the Description
- Review Parameters
- Explore Response Schemas
- Copy ready-to-use Curl commands for testing
- Test Directly in Your Browser (without needing an API key)

DRAFT



Figure 10: API Endpoint Documentation Example



Figure 11: Interactive API Testing Interface

7.2.4.2 Feature 2: Data Visualization & Demonstration

7.2.4.2.1 Dashboard

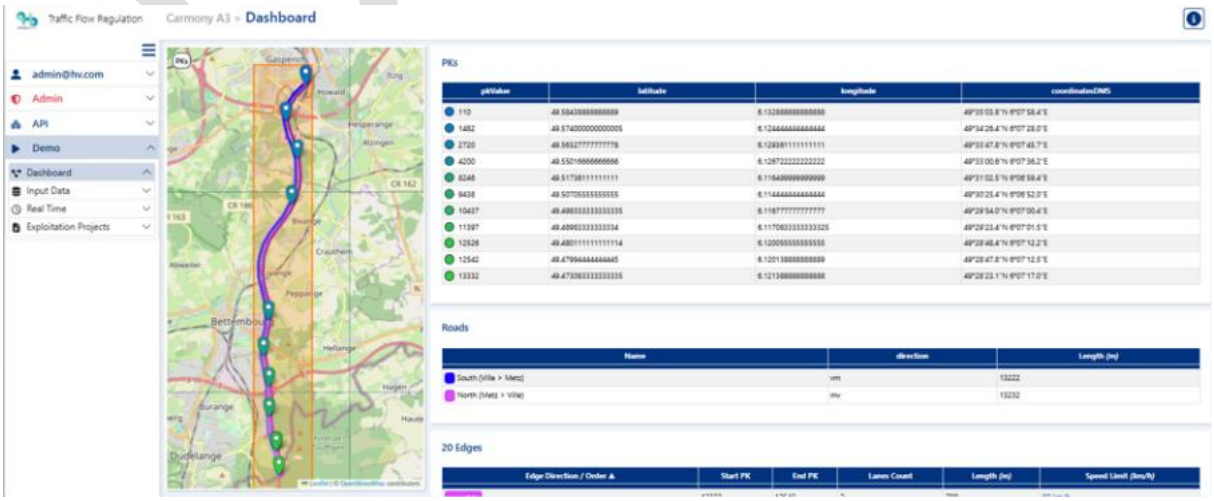


Figure 12: Dashboard page

The Dashboard provides an overview of the road network configuration used by the Traffic Flow Regulation system. The road network constitutes the topological reference on which all traffic monitoring, congestion detection, prediction, and regulation algorithms operate.

This page allows operators to verify that the road corridor has been correctly configured before traffic data is processed by the orchestration services.

The Dashboard is organized into four main sections:

- Map: displays the configured road corridor, including the road geometry, travel directions, and PK reference points used by the system.
- PKs: lists the PK reference points available on the corridor, together with their geographical coordinates.
- Roads: summarizes the configured roads and their main characteristics, including travel direction and total length.
- Edges: provides a detailed view of the road segmentation used by the orchestration algorithms, including start and end PKs, section length, number of lanes, and speed limits.

Together, these elements define the topological model of the corridor and provide the spatial framework required by the Traffic Flow Regulation services.

7.2.4.2.2 Input data

Traffic data is provided as snapshots by the Luxembourg Administration des Ponts et Chaussées. A traffic snapshot is a time-stamped representation of traffic conditions on the road network, covering both travel directions and including variables such as traffic speed and flow.

Historical traffic snapshots are consumed, processed, and organized by the advanced-time module of the Traffic Flow Regulation system. This preprocessing step is essential for filtering and classifying traffic situations before they are used by the orchestration algorithms.

The Input Data page provides access to the available historical traffic data and offers a first level of traffic analysis and visualization.

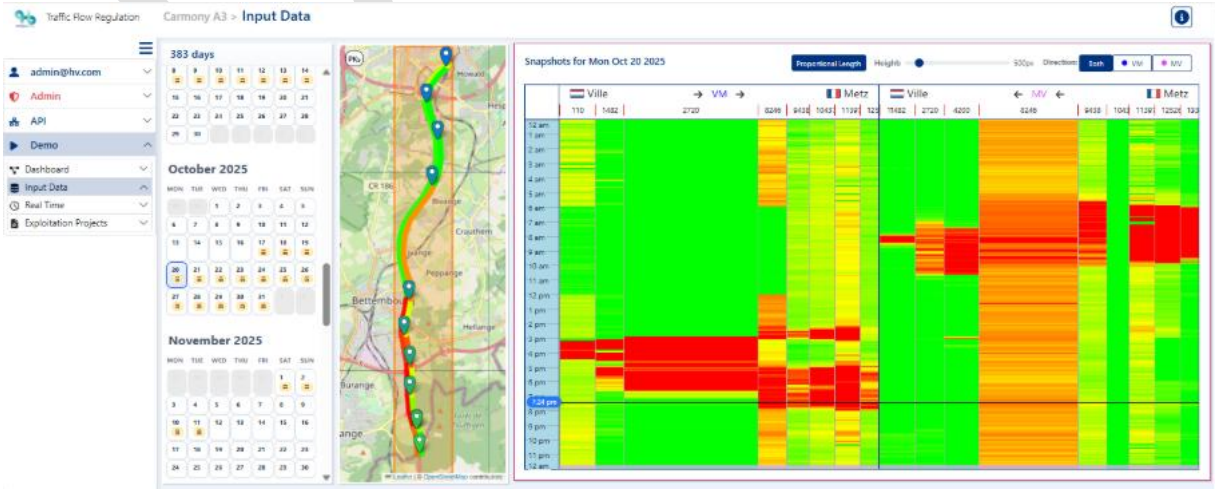


Figure 13: Input RAW data page

DRAFT

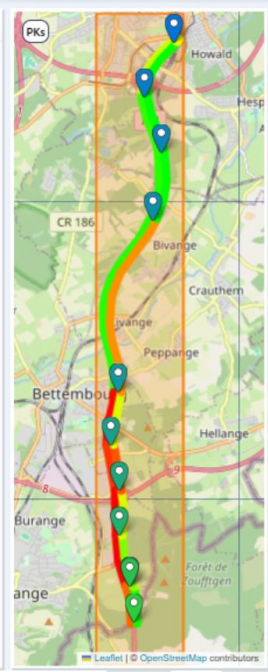
The page is organized into three main areas:

Data Availability Calendar



The information button located in the upper-right corner indicates the date from which historical traffic data is available. The calendar displays all days for which traffic data is available in the project data source. Users can select a day to visualize the corresponding traffic snapshots.

Corridor Map



The map displays the A3 corridor and the state of the selected traffic snapshots. The road segments are colored according to the traffic conditions observed during the selected period.

Snapshot visualization

The snapshot view provides a graphical representation of traffic conditions along the corridor throughout the day.

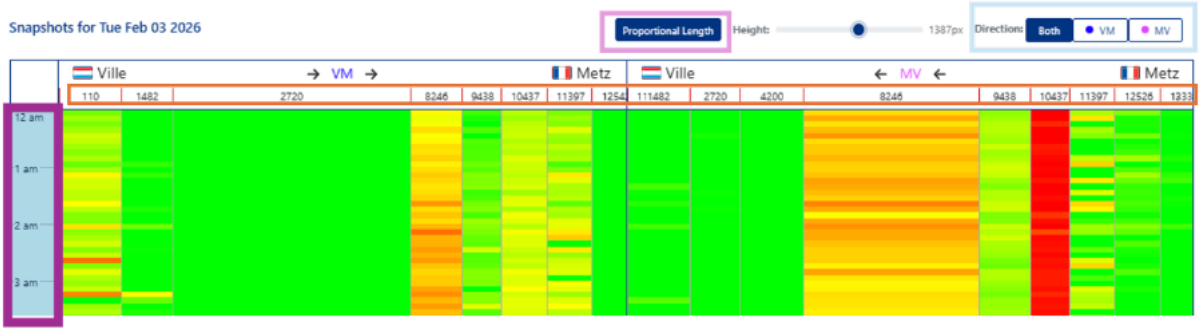


Figure 14: Snapshot visualization

Users can:

DRAFT

- Switch between a schematic representation and a proportional-length representation of the corridor (pink box)
- Select the traffic direction to analyse (Blue box):
- VM: Luxembourg City (Ville) -> Metz.
- MV: Metz -> Luxembourg City.
- Both: display both directions simultaneously.
- Explore traffic conditions over time.

Traffic snapshots are delivered with a time granularity of 6 minutes and are organized chronologically throughout the day.

The horizontal axis represents the corridor segmented by PK reference points (orange box), while the vertical axis represents time (purple box).

The Traffic Flow Regulation algorithms analyse these historical snapshots to identify traffic degradation patterns and congestion events that may affect traffic flow.

The colour scale used in the visualization is interpreted as follows:

- Green: free-flow traffic conditions.
- Yellow: early signs of traffic degradation.
- Light Orange: moderate traffic degradation.
- Dark Orange: significant traffic degradation.
- Red: severe congestion or traffic jam.

7.2.4.2.3 Real time

The Real Time page provides a live view of the traffic conditions currently observed on the A3 corridor.

Within the Carmony architecture, this page represents the Traffic Situation Identification bloc of the reactive layer, where current traffic conditions are continuously monitored and analysed to detect traffic degradations, congestion events, and potential regulation opportunities.



Figure 15: Real time snapshots

This page displays:

- The most recent traffic snapshot received from the data source.
- The current traffic state along the corridor.
- Traffic degradation areas detected by the incident detection service.
- Congested segments highlighted in red according to the traffic degradation level.

DRAFT

- Regulation segments highlighted in blue, representing the road sections where a speed regulation is recommended.

Users can apply the following visualization filters:

- Proportional Length: display road segments according to their actual length.
- Direction Selection

By hovering over a traffic jam or regulation segment, users can access detailed information, including:

- PK range.
- Direction.
- Segment length.
- Number of lanes.
- Current speed.
- Speed limit.
- Congestion score.

7.2.4.2.4 Exploitation projects

The Exploitation projects section supports the preventive layer of the traffic flow regulation system. It allows road operators to classify historical traffic data according to operational day types, such as working days, weekends, or holidays, so that traffic degradation prediction and regulation algorithms can operate on traffic patterns representative of each type of day.

Created projects can be accessed at any time from the Exploitation Projects list.

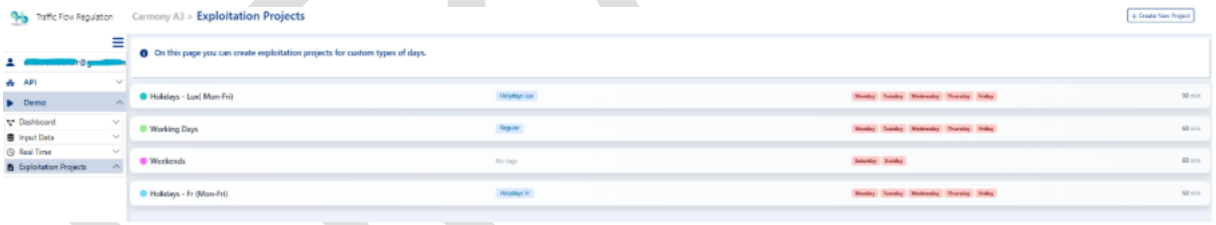


Figure 16: List of created exploitation projects

New projects can be created using the Create New Project button located in the upper-right corner of the page.

The exploitation workflow is organized into three successive stages:

1. Project Edition
2. Input Data
3. Predictions and Regulations

Project edition:

The Project Edition page allows users to define the characteristics of a new exploitation project.

DRAFT

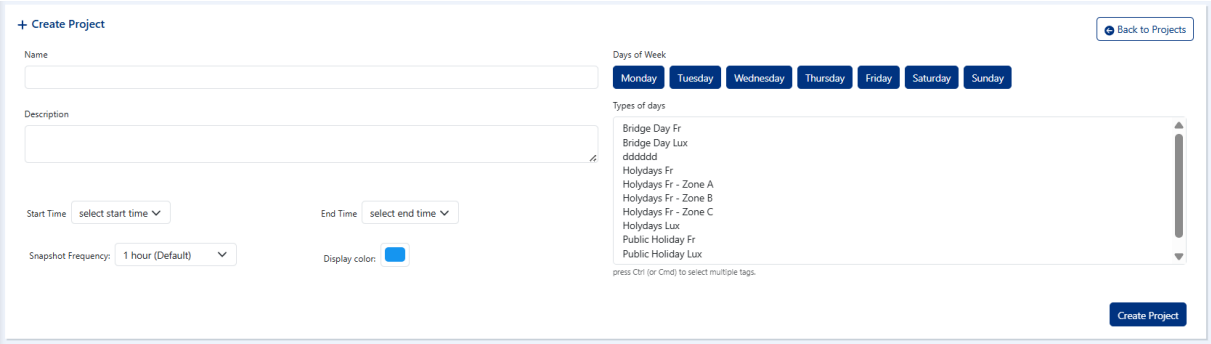


Figure 17: Project edition page

Users can configure:

- The project name and description.
- The days of the week to be considered.
- The associated types of days (working days, holidays, bridge days, etc.).
- The time range of interest.
- The snapshot aggregation frequency.
- The project display color.

These parameters define the subset of historical traffic data that will be used for traffic pattern analysis and prediction.

Input data:

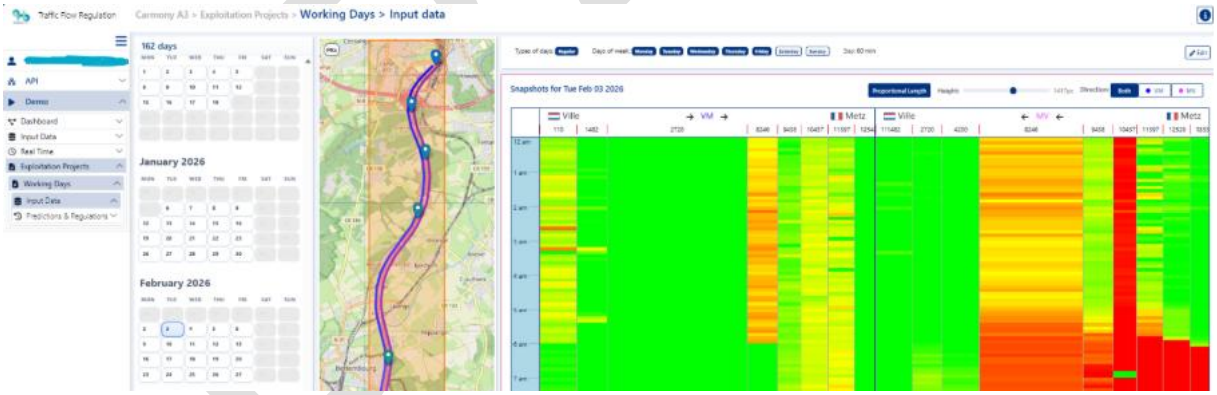


Figure 18: Filtered historical input data for an exploitation project

This page displays all traffic snapshots associated with the selected project configuration and day type.

Unlike the input data module described previously, which provides access to the complete ran historical traffic data, this page displays only the filtered traffic data corresponding to the selected exploitation project criteria.

The visualization capabilities are similar to those available in the Input Data module and include:

- The number of historical days matching the selected criteria.

DRAFT

- The calendar of available snapshots.
- The corridor map.
- Traffic snapshot visualization.
- Direction filters (VM, MV, Both).
- Timescale and display options.

This filtered dataset constitutes the input used by the prediction and regulation algorithms.

Predictions and regulations

The Predictions and Regulations page provides an overview of the traffic degradations predicted by the system and the regulation measures associated with them.



Figure 19: Predictions and regulations page 1

Predicted degradations correspond to traffic slowdowns and congestion situations identified by the backend algorithms using the filtered historical traffic data associated with the selected day type.

The screen allows user to visualize these predicted events and refine the display using several filters, including:

- Road direction.
- Proportional or non-proportional corridor representation.
- Edge visibility.
- Predicted degradations.
- Regulation recommendations.

Regulations correspond to speed instruction calculated by the backend to manage traffic conditions upstream of a predicted degradation.

For each regulation recommendation, the system determines:

- The recommended regulation speed.
- The length of the road section where the regulation should be applied.

DRAFT

The objective of these recommendations is to prevent or mitigate the downstream traffic degradation by harmonizing traffic flow before congestion develops.

By selecting a degradation or regulation area, users can access detailed information such as:

- Traffic jam location.
- Start time.
- Direction.
- Traffic speed.
- Flow rate.
- Congestion score.
- Regulation speed.
- Regulation length.



Figure 20: Predictions and regulations page 2

The Actions menu, located in the upper-right corner of the page, provides additional project management functions:

- Reset All: removes all previously calculated degradations and regulation recommendations for the selected project and restores the project to its initial state.

These actions allow operators to refresh prediction results after modifying project parameters or to restart the analysis process from scratch.

7.3 User guide for the Traffic Forecasting service

The Traffic Forecasting service has been developed as a standalone service, and can be deployed at any machine, both locally and remote. The module has been configured to be compiled as a docker container and deployed via the docker engine.

The source code is currently placed in the following github repository, and access is available upon request: https://github.com/frontier-innovations/carmony-traffic-forecasting. The source code contains a Dockerfile that can be used to build a docker image and deployed locally in a docker container. Once deployed, users can access the Traffic Forecasting service via a browser at the following URL: http://localhost/

Documentation about the Traffic Forecasting API is available at the URL: http://localhost/docs, which produces an interactive documentation page, containing all available endpoints. Figure 7 shows a screenshot of the documentation page.

DRAFT

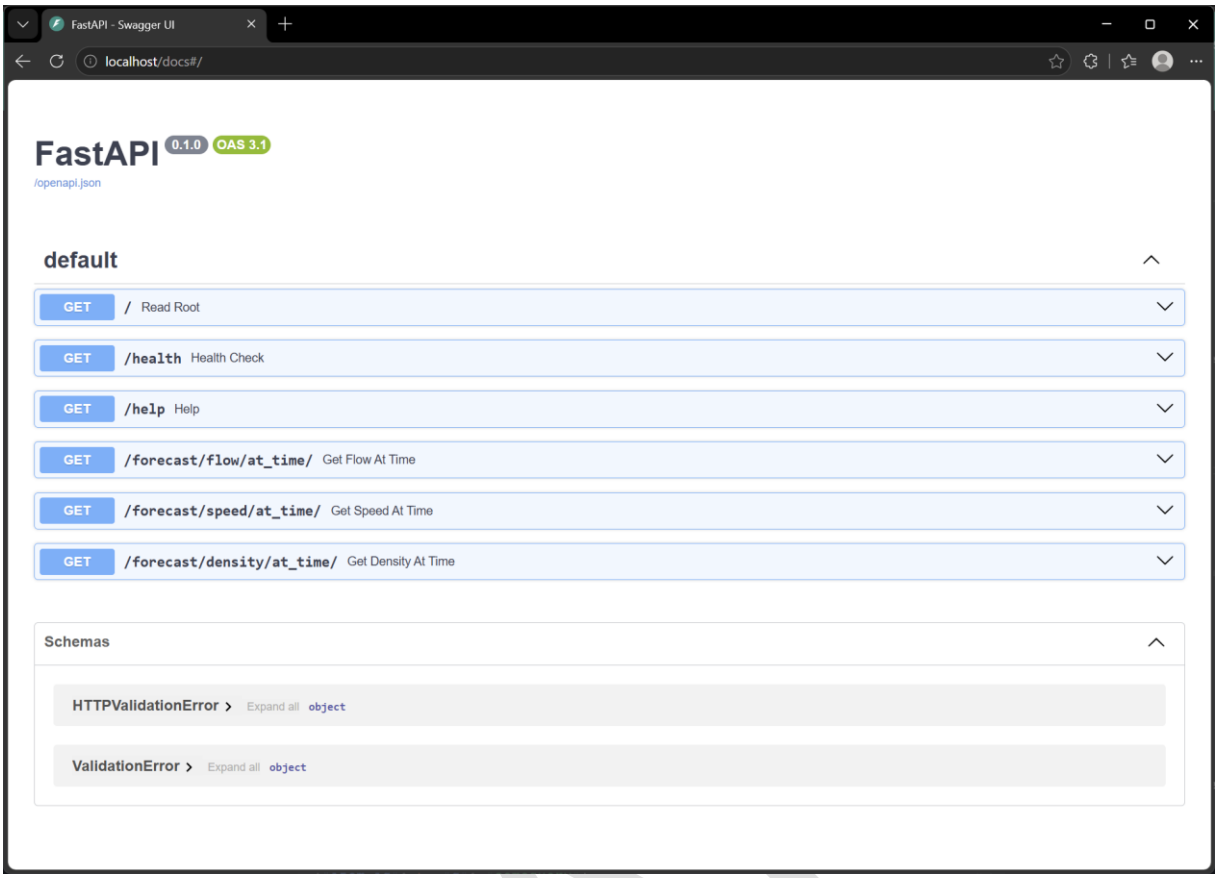


Figure 21: Traffic Forecasting service interactive documentation

In addition to exploring the Traffic Forecasting service via the interactive documentation, users can request predictions directly using the dedicated API endpoints. An example for traffic density prediction is shown in Figure 22, which shows the response of the following URL call:

http://localhost/forecast/density/at_time/?timestamp=1782896579&station_id=12526&direction=mv

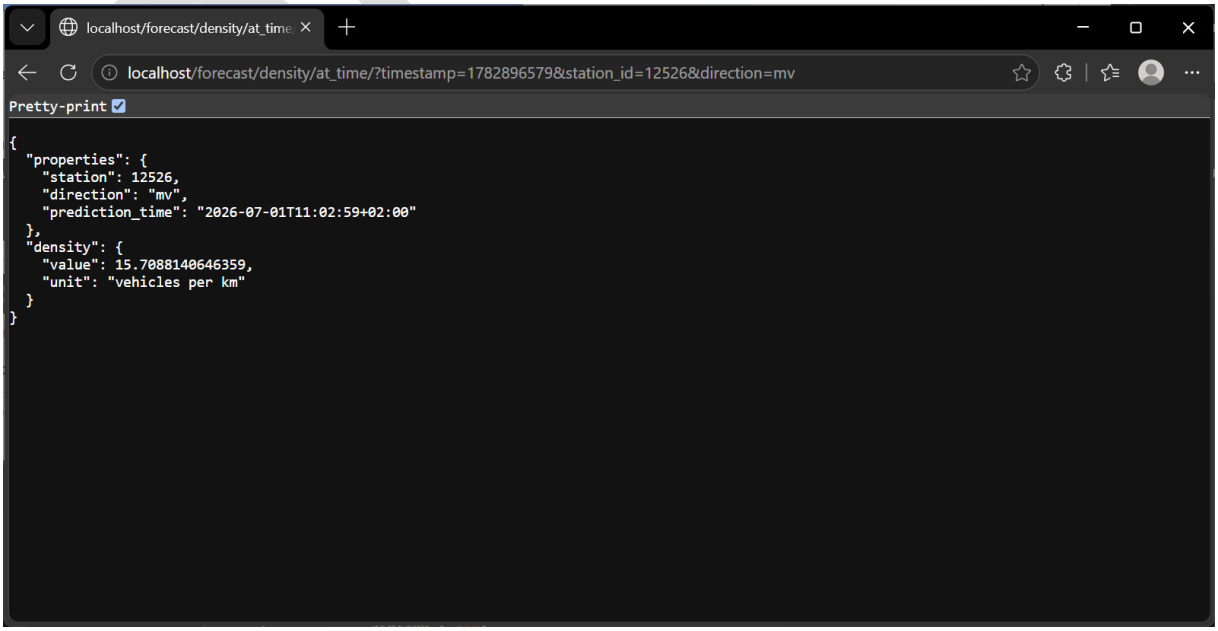


Figure 22: Sample response from density forecast endpoint

8. Preliminary setup

8.1 A3 Network

8.1.1 A3 network configuration

The A3 network configuration was developed using SUMO (Simulation of Urban Mobility) to create a realistic microscopic traffic simulation environment.

For the network generation, we extracted the road infrastructure from OpenStreetMap (OSM) using the osmWebWizard, a graphical tool provided by SUMO that simplifies the creation of simulation scenarios. The resulting SUMO network represents a directed graph consisting of nodes (junctions) and edges (road segments), where each lane is modeled explicitly, enabling detailed vehicle interactions and traffic flow analysis.

Traffic demand was derived from traffic detector measurements to reproduce the observed traffic volumes during the simulation period. The following procedure outlines the steps to generate traffic demand for SUMO:

- Collect Traffic Counts: Traffic volumes were obtained from roadside detectors, providing the number of vehicles passing each detector during a defined time interval.
- Prepare Count Data: Traffic counts were converted into the format required by SUMO, specifying the measured vehicle volumes at individual detector locations and time intervals.
- Define Network Entry Points: Detector locations were mapped to the corresponding edges of the SUMO network.
- Create Vehicle Flows: Measured traffic counts were converted into SUMO vehicle flows by using traffic counts and detector locations.
- Generate Candidate Routes: Based on the network topology, a set of possible routes is created.
- Estimate Routes with routeSampler.py: The SUMO tool routeSampler.py was used to generate vehicle demand that matches the observed detector counts. The script iteratively samples from the candidate route set and adjusts the number of vehicles assigned to each route until the simulated detector flows closely reproduce the measured traffic counts. This allows the creation of realistic route files without requiring explicit origin–destination matrices.
- Create SUMO Demand Files: The routes selected by routeSampler.py were converted into SUMO route files (*.rou.xml), defining vehicle departures, vehicle types, and route assignments for the simulation period.

8.2 Traffic data

8.2.1 Traffic forecasting calibration

This section presents the traffic forecasting framework developed for the A3 motorway within the CARMONY project. The analysis is based on harmonized historical and real-time traffic observations collected from loop detectors installed along the motorway corridor. Traffic forecasting was performed for three traffic variables, flow, density, and speed, using a set of statistical and machine learning algorithms. The forecasting framework focused on the predictive capability of temporal descriptors only, while four models were evaluated.

DRAFT

This chapter is structured as follows. First, a description of data used for the experiments is provided. Second, the models applied are presented. Finally, the preliminary results are discussed.

8.2.1.1 Data Sources

In this study, traffic observations collected from loop detectors installed along the A3 highway in Luxembourg were analysed. A total of 12 sensor locations (23 sensor-direction pairs) were available for this analysis. Two complementary datasets were considered. The historical dataset has a temporal resolution of 6 minutes and covers the period from 1st March 2025 to 1st March 2026. In addition, a real-time dataset with a temporal resolution of 5 minutes was collected and covers the period from 26th March 2026 to 24th May 2026. A short gap of approximately 25 days is observed between the historical archive and the operational real-time services. Only a limited number of stations exhibited incomplete temporal coverage or missing traffic directions in the real-time feed. The two datasets were subsequently harmonised into a common format and quality-controlled prior to their use in the forecasting experiments.

The following sections present traffic flow, density and speed profiles for selected sensors. The objective is to identify observed traffic patterns and anomalies that can inform the development and interpretation of the forecasting models.

8.2.1.2 Data Harmonization

Traffic measurements were obtained from two complementary data sources characterised by different formats and temporal resolutions. Historical observations were provided as comma-separated files (CSV) with a temporal resolution of 6 minutes, whereas real-time measurements were retrieved through a DATEX II service in XML format with a temporal resolution of 5 minutes. To ensure consistency between the two datasets, a harmonization procedure was implemented prior to any analysis. The process included the standardisation of variable names, timestamps, sensor IDs, traffic directions, and spatial references. Sensor locations were extracted from the DATEX II metadata and associated with the corresponding road section IDs (PK values), while traffic direction information was reconciled between the historical and real-time datasets through a common direction classification scheme.

Particular attention was given to traffic flow measurements. In the DATEX II service, traffic flow is reported as an average hourly flow rate (veh/h), whereas the historical dataset stores vehicle counts over the corresponding observation interval. Therefore, real-time hourly flow rates were converted to equivalent vehicle counts per 5-minute interval (veh/5min) to obtain a common representation of traffic flow across datasets. The original DATEX II flow-rate values together with the reporting period were retained as supplementary variables to ensure full traceability of the harmonization procedure. The resulting dataset provides a unified structure for all traffic variables and enables the seamless integration of historical and real-time observations within a common analytical framework.

8.2.1.3 Data Quality and Cleaning

A systematic data quality assessment was subsequently applied to all sensor-direction pairs to identify observations that could adversely affect subsequent analyses and model development. Quality indicators were computed separately for traffic flow, speed, and density time series and included the proportion of missing observations, the proportion of zero values, the occurrence of plateau behaviour (i.e., long sequences of identical consecutive values), and the expected temporal resolution of each dataset (6 minutes for historical measurements and

DRAFT

5 minutes for real-time measurements). Based on predefined quality thresholds, each sensor-direction-variable combination was classified as either usable or unusable. Sensors exhibiting excessive data gaps, anomalously high proportions of zero values, persistent plateau behaviour, or extremely low information content were excluded from the cleaned dataset. The quality assessment indicated that most traffic sensors satisfied the established criteria, with only a limited number of historical sensor streams being identified as problematic. The resulting cleaned and harmonised dataset formed the basis for the subsequent seasonal analyses and the development of the traffic forecasting models.

8.2.1.4 Seasonal Traffic Patterns for selected motorway sensors

Seasonal traffic patterns were analysed to quantify the temporal regularity and predictability of traffic conditions along the A3 highway. Median typical daily profiles were computed from the harmonised dataset by computing the median traffic conditions for each time-of-the-day across all available days, separately for all days, weekdays and weekends, considering traffic flow, density and speed in both directions. The resulting profiles reveal recurring directional differences, while also highlighting temporal variability and sensor-specific traffic behaviour that may influence forecasting performance.

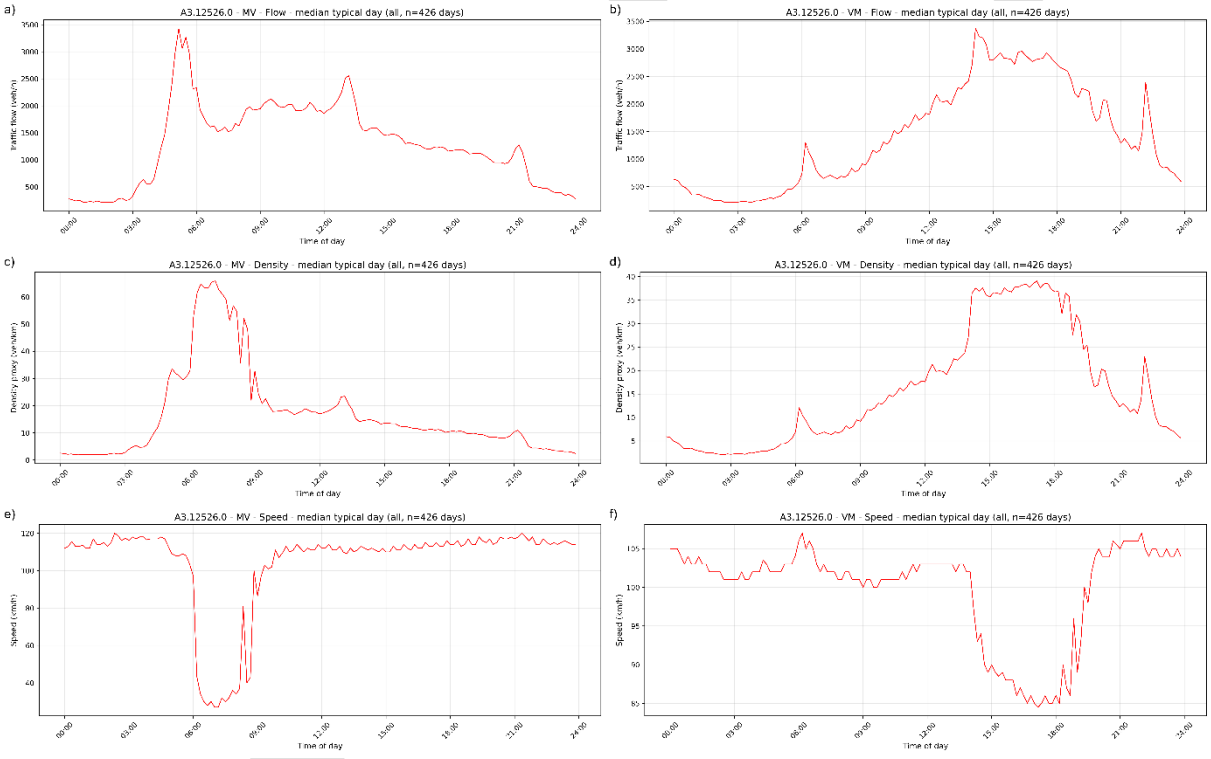


Figure 23: Median typical daily traffic profiles for Sensor PK 12526, showing traffic flow, density and speed for both traffic directions.

Sensor PK 12526 exhibits a pronounced daily traffic cycle in both directions, with distinct peak periods, direction-dependent congestion patterns and consistent speed profiles, indicating highly regular traffic behaviour (Figure 22).

8.2.1.5 Anomaly Detection for selected motorway sensors

Anomaly detection was applied to the cleaned and harmonised traffic dataset to identify observations deviating substantially from the expected temporal traffic behaviour. Time-specific baseline values were estimated from the harmonized dataset for each

sensor,direction, variable, weekday/weekend status, hour and minute. Observations with an absolute z-score greater than 3 were classified as anomalies.

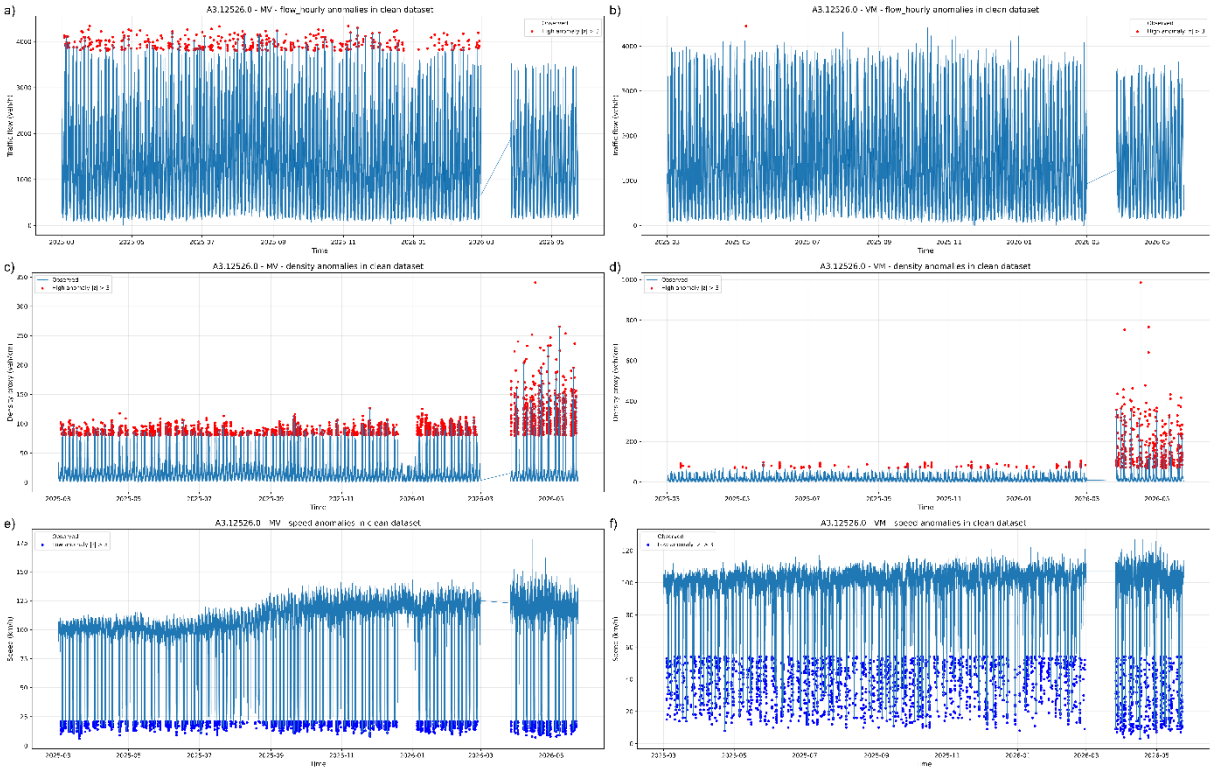


Figure 24: Z-score-based anomaly detection results for sensor PK 12526, showing hourly traffic flow, density proxy and speed in both traffic directions.

Sensor PK 12526 exhibits a relatively regular anomaly structure. Most detected anomalies coincide with traffic peak periods, where elevated traffic flow and density are accompanied by temporary speed reductions (Figure 23). Apart from these periods, only a limited number of observations are identified as anomalous, indicating a generally stable and predictable traffic regime.

8.2.1.6 Models

Four regression models were evaluated for predicting traffic flow, density and speed from temporal, directional and positional predictors. Linear Regression was initially evaluated as a transparent baseline, that estimates the target variable as a linear combination of the input predictors using least-squares fitting (James et al., 2023; Hastie et al., 2009). Polynomial Regression was subsequently evaluated as an extension of Linear Regression by introducing 4th degree transformations of the predictor variables, allowing the model to capture non-linear relationships while preserving the least-squares estimation framework. Random Forest was applied as a non-linear ensemble method that builds multiple decision trees on randomized subsets of the data and aggregates their predictions to reduce variance and improved robustness (Breiman, 2001). XGBoost was implemented as a regularized gradient-boosted tree algorithm that adds trees sequentially, with each new tree correcting the residual errors of the previous ensemble while controlling model complexity (Friedman, 2001; Chen and Guestrin, 2016). LightGBM was used as a computationally efficient gradient-boosting framework that grows trees using leaf-wise strategy and accelerates training through techniques such as Gradient-based One-Side Sampling and Exclusive Feature Bundling (Ke et al., 2017). Hyperparameter optimisation for the Random Forest, XGBoost and LightGBM

DRAFT

models was performed using GridSearchCV with five-fold cross-validation. The optimal hyperparameter combinations identified during the search were subsequently used to train the final forecasting models on the complete training dataset.

8.2.1.7 Predictions

Table 9: Prediction performance of the evaluated forecasting models for density, hourly traffic flow and speed.

Model	Density			Flow			Speed		
	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE
Polynomial Regression	0.23	18.87	10.33	0.43	778.56	598.96	0.15	23.44	17.56
Random Forest	0.33	17.45	6.08	0.68	518.46	333.03	0.43	18.65	11.19
XGBoost	0.32	17.37	6.16	0.69	503.34	335.28	0.45	18.22	11.09
LightGBM	0.33	17.24	5.99	0.69	499.51	329.22	0.44	18.17	11.16

The results indicate that traffic patterns along the A3 motorway exhibit strong non-linear temporal behaviour. Among the evaluated models, Polynomial Regression (4th degree) improved upon simple linear modelling, while the ensemble machine-learning algorithms (Random Forest, XGBoost and LightGBM) consistently achieved the highest predictive performance across most variables. The best-performing models achieved R2 values of up to 0.69 for traffic flow, while density and speed remained more challenging prediction targets.

Although Table 9 summarizes the overall model performance, noticeable variability was observed among individual sensor-direction combinations. Sensors exhibiting stable and recurrent traffic regimes generally achieved higher predictive accuracy, whereas locations characterised by stronger traffic variability or more complex traffic dynamics produced larger prediction errors.

Overall, the findings demonstrate that temporal descriptors alone can explain a significant fraction of the observed traffic variability, particularly for traffic flow. Nevertheless, incorporating additional explanatory variables, such as weather conditions, incidents or upstream traffic conditions, is expected to further improve forecasting performance.

8.2.1.8 Model-based prediction profiles for selected motorway sensors

To complement the aggregate performance metrics, model-based prediction profiles were examined for three representative sensors. The plots compare the temporal behaviour of Polynomial Regression, Random Forest, XGBoost and LightGBM for flow, density and speed in both traffic directions during the test period. They provide a visual assessment of how consistently each model reproduces the main traffic dynamics at locations with different levels of temporal regularity.

DRAFT

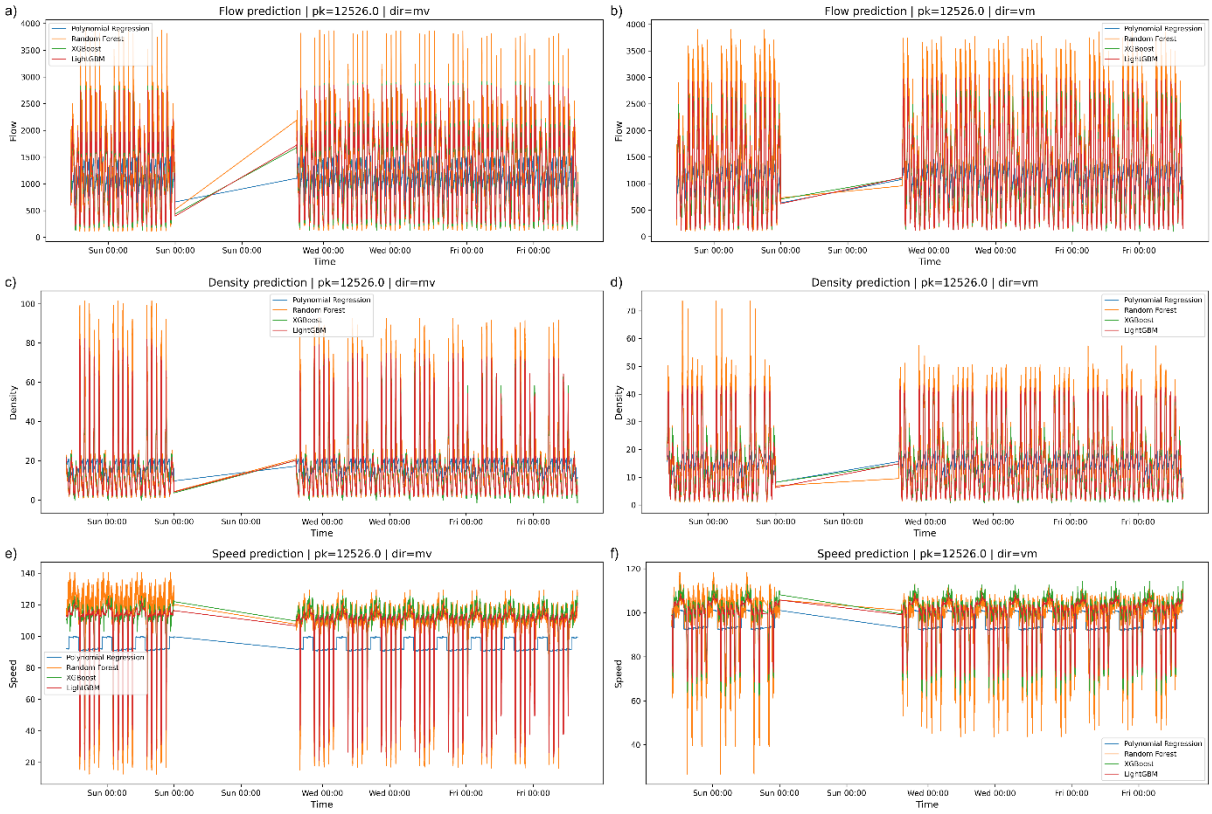


Figure 25: Comparison of model-based predictions for the representative sensor PK 12526 during the test period, for both directions.

Sensor PK 12526 exhibits stable and recurrent traffic behaviour in both directions (Figure 25). The four models produce very similar predictions throughout most of the test period, with only minor differences during rapid traffic transitions and local peaks. Overall, the optimized tree-based models and Polynomial Regression provide comparable representation of the dominant traffic dynamics.

8.2.1.9 Sensor-level model performance across the A3 Highway

To further assess the spatial robustness of the forecasting framework, model performance was evaluated separately for each sensor-direction combination. Figure 26 reports the RMSE of the four models for all three variables in both directions. The comparison should be interpreted with each traffic variable, since RMSE is scale-dependent and therefore should not be compared directly across different traffic variables.

DRAFT

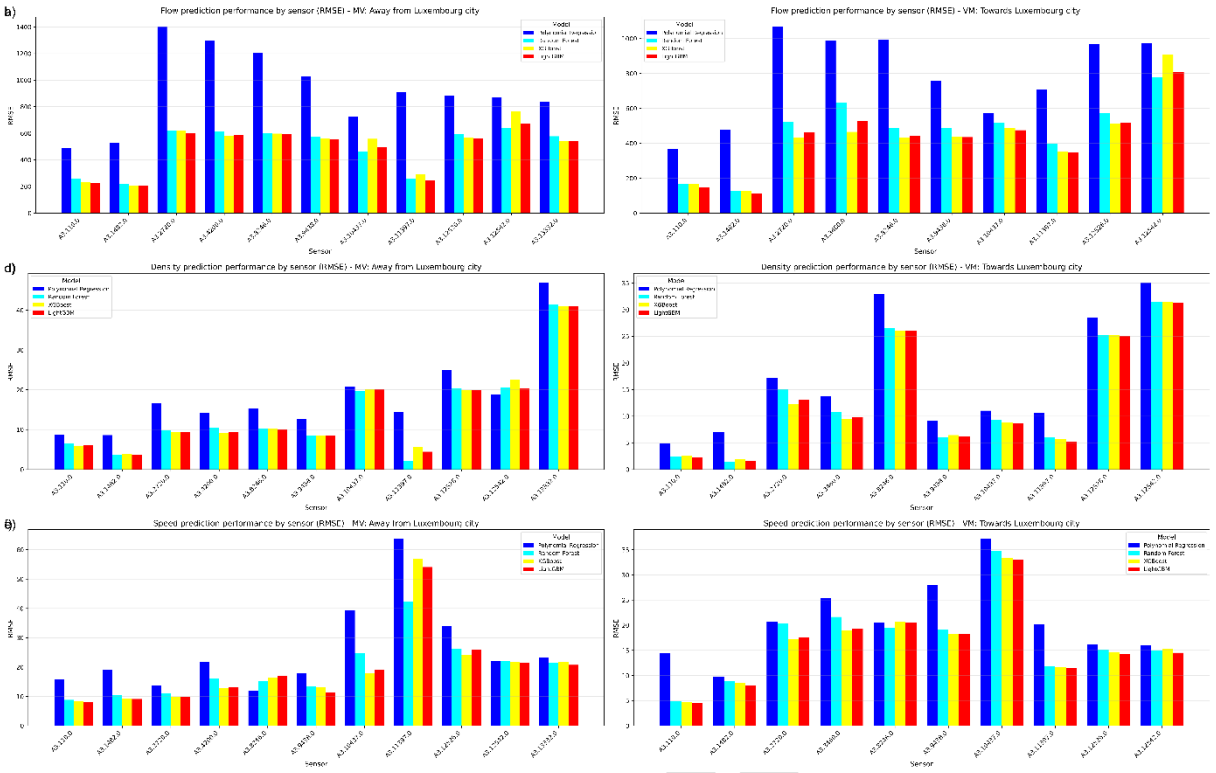


Figure 26: Sensor-level comparison of optimized model prediction performance across the A3 network, expressed as RMSE, for density, hourly traffic flow and speed in both traffic directions.

Figure 25 confirms that predictive performance depends strongly on location. Across most sensor-direction-variable combinations, Polynomial Regression produces higher prediction errors, whereas the optimized tree-based models systematically reduce RMSE. XGBoost and LightGBM exhibit the most consistent performance across sensors, while Random Forest achieves comparable accuracy for many sensor-direction combinations. These findings indicate that non-linear machine learning models better capture the temporal structure of the traffic data of A3 highway.

The largest spatial variability is observed for flow, which remains the most challenging target variable due to its sensitivity to short-term demand fluctuations and local disturbances. Density and speed show more stable behaviour overall although selected sensors still exhibit elevated errors, particularly where previous exploratory analysis indicated irregular or non-stationary traffic patterns. These results demonstrate that temporal, directional and positional predictors provide a useful baseline, but they are not sufficient to fully explain the site-specific disruptions.

At sensor level, the lowest RMSE values are mainly observed at A3.110 and A3.1482, indicating more regular and predictable traffic conditions. In contrast, A3.12542, A3.13332 and A3.8246 show higher errors for flow, reflecting stronger sensitivity to local demand fluctuations. For speed, the most challenging locations are A3.10437 and A3.11397 in the MV direction, consistent with the irregular patterns identified in the exploratory analysis. Overall, model performance varies across the corridor, with prediction accuracy generally higher at stable recurrent locations and lower at sensors affected by direction asymmetry, abrupt changes or more complex local dynamics.

8.2.1.10 Summary

The analysis demonstrates that the harmonized historical and real-time dataset provides a suitable basis for preliminary traffic forecasting along the A3 motorway. The exploratory,

DRAFT

seasonal and anomaly-detection analyses show that traffic behaviour differs substantially across sensor locations and directions.

The forecasting results confirm that temporal, directional and positional predictors provide a useful baseline for short-term traffic prediction. Polynomial Regression improved the linear baseline but remained less accurate than the optimized tree-based machine-learning models. Random Forest, XGBoost and LightGBM consistently achieved the best predictive performance across the evaluated traffic variables, highlighting the importance of modelling non-linear traffic behaviour. The remaining sensor-level variability indicates that future improvements should incorporate lagged traffic-state variables and upstream/downstream dependencies to better represent local traffic propagation.

DRAFT

9. Current limitation and Conclusion

This section presents the current limitations of the software and of the proposed approach at this stage of development.

The final solution is expected to consist of three complementary modules: a preventive layer operating in advanced time, a reactive layer operating in real time, and a module dedicated to KPI analysis. The current version of the software is mainly focused on the advanced-time preventive layer. As a result, real-time reaction capabilities and KPI analysis functions are not yet fully implemented in the present release.

In its target configuration, the solution may interact with different types of road users and vehicle systems, including human drivers, ADAS-equipped vehicles, and connected and automated vehicles, through ITS standard communication protocols. However, this level of interaction depends on the maturity of the integration environment, the availability of interfaces, and the implementation of standardised communication channels within the overall Carmony architecture.

The main limitation of the current approach concerns data infusion, as the quality, coverage, and resolution of available traffic data directly affect the performance of the software. At this stage, the following limitations have been identified. These limitations also highlight a broader challenge that interconnected orchestrator systems may face soon: ensuring sufficient data quality in terms of spatial and temporal granularity.

- Dependence on publicly available traffic data, which may limit the completeness, reliability, and continuity of the input information.
- Limited spatial granularity of traffic information, which can restrict the ability to detect and characterise traffic degradation at a very local level.
- Limited temporal resolution of some data sources, which may reduce the accuracy of advanced-time detection and forecasting, especially for rapidly evolving traffic conditions.

These limitations should be progressively addressed by developments in the ITS and intelligent road sectors, particularly through improved data integration, higher-resolution traffic data, and more mature communication interfaces. This section is aligned with the proposed structure for D3.2 and D3.3, where current limitations include dependence on public traffic data and constraints related to spatial and temporal resolution.

DRAFT

10. Abbreviations

Term	Definition
A3	Luxembourg A3 motorway
ADAS	Advanced Driver Assistance systems
API	Application Programming interface
CSO	Car Sharing Optimizer
Dashboard	Main platform page providing an overview of the current traffic situation and software outputs.
FRON	FRONTIER (Project Partner)
GET	HTTP GET Method
HYP	HYPERVISOUL (project partner)
ITS	Intelligent Transport Systems
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
OPEXO	Operational Expenditure
OSM	OpenStreetMap
PK	Point Kilometre (road reference position)
QoS	Quality of Service
REST	Representational State Transfer
SUMO	Simulation of Urban MObility
SW	Software
TEC	TECNALIA (project partner)
TMC	Traffic Management Center
UC	Use case
URL	Uniform Resurce Locator
VIF	Virtual Vehicle Research (project partner)
VPS	Virtual Private Server
WP	Work Package

DRAFT

11. References

[1] Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[2] Chen, T., and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

[3] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[4] Hastie, T., Tibshirani, R., and Friedman, J. (2009). Linear Methods for Regression. In The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. https://doi.org/10.1007/978-0-387-84858-7_3

[5] James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). Linear Regression. In An Introduction to Statistical Learning. Springer Texts in Statistics. https://doi.org/10.1007/978-3-031-38747-0_3

[6] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems, 30

DRAFT