D3.1 CCAM services architecture description

Project title: **Enhancing Integration and Interoperability of CCAM eco-system**

Project acronym: **IN2CCAM**

Horizon Research and Innovation Actions
Project No. 101069573
Call HORIZON-CL5-2021-D6-01

in2ccam.eu

Dissemination level	Public (PU) - fully open
Work package	WP3 - Architectural and technical specifications for CCAM Services development, integration, intermodal interfaces and interoperability
Deliverable number	D3.1 – CCAM services architecture description
Status - Version	Final – V1.0
Submission date	25/07/2024
Keywords	Architecture Design, V2X Infrastructure, IN2CCAM Services, Uniform design and interfaces

Quality Control

	Name	Organisation	Date
Peer review 1	Markku Niemi	TAMP	12/07/2024
Peer review 2	Coen Bresser	ERTICO	10/07/2024
Quality review	Marie-Laure Watrinet	LIST	04/07/2024

Version history

Version	Date	Author	Summary of changes
0.1	30/11/2023	Simon Krepper (Akkodis)	First draft of the document for the Milestone
0.2	27/03/2024	Simon Krepper	Updated document structure
0.3	24/04/2024	Simon Krepper	High level functional and general architecture view for all Living Labs (Chapter 3)
0.4	02/05/2024	Shawan Mohammed (Akkodis)	Detailed functional and architecture view of all LLs (Chapter 4)
0.5	29/05/2024	Shawan Mohammed	Review and Updated version of all architecture views

0.6	03/06/2024	Dimitrios Simopoulos (Akkodis)	Update overall structure and text for the chapters
0.7	08/06/2024	Living Labs	Adding and updating Graphics and contents from LLs
0.8	28/06/2024	Simon Krepper	Addressing of comments and ready for peer review
0.9	22/07/2024	Simon Krepper, Shawan Mohammed (Akkodis)	Version after peer-review integrating comments from peer-reviewers
1.0	25/07/2024	Simon Krepper, Shawan Mohammed (Akkodis)	Final version after final check

LEGAL DISCLAIMER

The IN2CCAM project is funded by the European Union. However, the 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, CINEA, can be held responsible for them.

The IN2CCAM project Consortium members shall have no liability for damages of any kind, including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
AD	Autonomous Driving
ADS	Automated Driving System
API	Application Programming Interface
AV	Autonomous Vehicles
CAM	Cooperative Awareness Message
CAV	Connected Autonomous Vehicles
CCAM	Cooperative Connected and Automated Mobilit
CPM	Collective Perception Messages
CV	Connected Vehicles, Connected Vehicles
DT	Digital Twin
FMI	Finnish Meteorological Institute
GLOSA	Green Light Optimal Speed Advice
HMI	Human Machine Interface
IoT	Internet of Things
ITS	Intelligent Transportation Systems
LL	Living Lab
OBU	On Board Unit
ROC	Remote Operation Centre
RSU	Road Side Unit
SUMO	Simulation of Urban Mobility
TCC	Traffic Control Center

TIC	Traffic Information Center
TLC	Traffic Light Controller
TraCI	Traffic Control Interface
UVAR	Urban Vehicle Access Restriction
V2X	Vehicle to Everything
VRU	Vulnerable Road User

1 EXECUTIVE SUMMARY

This deliverable outlines a detailed architectural framework essential for integrating services within the IN2CCAM project's Living Labs. These labs are at the forefront of deploying innovative mobility solutions, aiming to enhance urban transportation systems across Europe.

The IN2CCAM project aligns with Horizon Europe's goals to advance urban mobility through innovative Connected and Automated Vehicle (CAV) technologies. Presented as an Innovation Action, this initiative involves a consortium focused on developing, testing, and validating CCAM services that integrate seamlessly into European urban transport systems. It encompasses six Living Labs—four 'Lead' and two 'Follower'—tasked with the full integration of physical, digital, and operational solutions aimed at enhancing safety, reducing environmental impact, and ensuring inclusive mobility. While the lead living labs are implementing the services in the respective cities, the follower living labs are presenting the services in a simulation environment.

The detailed content of the document is structured around several key aspects of the architectural framework. The functional architecture section details the essential modules such as ITS Platform APIs, which facilitate data exchange and interoperability among urban mobility systems, and Database Systems that manage data critical for operations. Decision Making APIs are highlighted for their role in supporting traffic management and safety enhancements, along with the Digital Twin Framework, which simulates urban environments to predict and plan services effectively.
The modular view elaborates on how each module interconnects, demonstrating robust data flows and integral service delivery across different urban contexts. Meanwhile, the scenario flow section illustrates how the general architecture applies to specific real-world scenarios, showcasing the system's adaptability and practical utility.

In describing the local architectures within the Living Labs, the document notes how Tampere showcases enhancements to last-mile mobility through automated shuttles and a sophisticated Remote Operation Centre, adapting the architecture to local dynamics. Trikala's focus is on integrating autonomous services to augment urban mobility and safety, customized to fit local infrastructure and societal needs. Turin utilizes dynamic re-routing based on real-time data to optimize traffic flows and reduce congestion. In Vigo, improvements in system-wide communication and strategic V2I interactions are made to enhance mutual awareness within the CCAM ecosystem.

2 INTRODUCTION

2.1 Project intro

The IN2CCAM (Enhancing Integration and Interoperability of CCAM eco-system) project is an Innovation Action that refers to the Horizon Europe call HORIZON-CL5-2022-D6-01-04: Integrate CCAM services in fleet and traffic management systems (CCAM Partnership).

The IN2CCAM consortium aims at developing, implementing, and demonstrating innovative services for connected and automated vehicles, infrastructure, and users. This is all in accordance with the vision of the Horizon Europe framework programme (2021-2027), which seeks to accelerate the implementation of innovative CCAM technologies and systems for passengers and goods.

At the core of IN2CCAM, there are six Living Labs (LL), which will implement a full integration of CCAM services in their transport systems. A set of physical, digital, and operational solutions will be proposed and implemented in four Lead LLs: Tampere (Finland), Trikala (Greece), Turin (Italy) and Vigo (Spain). Furthermore, two Follower LLs will take part in the design phase by providing ideas and data assessed by simulation test and validation: Bari (Italy) and Quadrilátero (Portugal).

As a whole, the IN2CCAM consortium expects that the full integration of CCAM services in future mobility will primarily benefit: (a) safety (i.e. reducing the number of road accidents due to human error); (b) the environment (i.e. reducing transport emissions and congestion by smoothening traffic flow and avoiding unnecessary trips); and (c) inclusiveness (i.e. ensuring inclusive mobility and good access for all). IN2CCAM project (in extenso: Enhancing Integration and Interoperability of CCAM ecosystem) is an Innovation Action referring to the Horizon Europe call HORIZON-CL5-2022-D6-01-04: Integrate CCAM services in fleet and traffic management systems (CCAM Partnership).

IN2CCAM aims to address the three following main challenges: update new physical infrastructures, use and update novel digital infrastructures, and propose suitable operational infrastructures. To reach such general objectives, the overall methodology of IN2CCAM is based on the definition, organization, implementation and evaluation of a set of six Living Labs (LLs) that will be the basis for implementing a full integration of CCAM services in the transport system.

The project runs for 36 months (3 years), with a start date of 1 November 2022 and an end date of 31 October 2025, with a consortium of 21 partners. IN2CCAM develops and tests innovative services to accelerate the implementation of innovative CCAM technologies and systems for passengers and goods, integrating them into the transport system.

2.2 Purpose of the deliverable

WP3 focuses on developing detailed architectural and technical specifications to support the integration and interoperability of Cooperative, Connected, and Automated Mobility (CCAM) services. It includes deliverables such as the CCAM services architecture description (D3.1), optimization of multimodal mobility services (D3.2), mobility network load balancing solutions (D3.3), and advanced simulation models and digital twin designs (D3.4). This work package aims to provide a robust foundation for enhancing urban mobility through innovative CCAM technologies.

This deliverable presents the output of Task 3.1 'Architecture and technical specifications of CCAM services and integration'. Through detailed descriptions and technical specifications, this deliverable highlights the necessary modules and their relationships for the essential data flows to provide a solid architecture basis that can be adapted within the IN2CCAM Living Labs but also to new city environments.

The deliverable shows the general architecture in Chapter 3 with a functional view, a modular view and the scenario flow. It will focus on the four lead LLs (Tampere, Trikala, Turin and Vigo) which are implementing services on set destination routes in the cities. The follower LLs will be simulating their use cases and have different needs for their Digital Twin models.

The deliverable should provide an easy understanding of the layout of each Living Lab as well as what modules are necessary for each service.

2.3 Intended audience and deviations from the proposal

The intended audience for this deliverable is all the participants of the Living Labs and anyone interested in the setup of the local architectures who wishes a better understanding of the service implementation in the Living Labs.

In IN2CCAM, there are 4 main Living Labs in 4 different cities where the services are being implemented and tested. Each of these Living Labs has a unique set up and is on different levels of integration of hardware and preexisting services. On the one hand there are hardware setups and services already running that are being extended during the project and on the other hand there is new integration of hardware and services on another location.

This presents a challenge to architectural design, which must accommodate both requirements. It must be compatible with existing platforms while also allowing for implementation and adaptation in other locations.

The final architecture results in various modules that can be adapted to suit the specific requirements of each Living Lab. Furthermore, a comprehensive description of all data streams is provided for each scenario, including details of the formats and purposes of each stream.

2.4 Structure of the deliverable and its relation with other work packages and deliverables

The first part of the deliverable describes the general architecture which has to include all the modules for the local architectures as well as a modular view which describes the physical modules and data connections. The scenario flow shows only the relevant modules for a specific service.

In the following chapter each Living Lab describes its implementations derived from the general architecture as well as an overview of the relevant parts for each service. Additionally all the data relevant for each service is shown, described in different tables.

The architecture takes into account all the results from Task 2.2 and the deliverable 2.3 where the requirements of communication, road infrastructure, safety and interoperability of the CCAM ecosystem have been laid out.

All the services that will be implemented have were laid out in a first description and can be shown in the scenario flow.

3 GENERAL ARCHITECTURE

3.1 Functional View

Chapter 3 describes the general architecture and covers all common elements of the Living Labs (LLs). The different data streams used in the LLs are detailed in chapter 4 in the respective section of each city.

The architecture will be described on distinct levels from top to bottom in a functional, modular, and scenario view.

A functional view of the architecture describes functions and modules necessary to process data to provide specific services. It includes gathering and pushing data, its preprocessing and saving in a database.

For certain services, additional processing is required on a separate server using the provided data and algorithms.

Figure 1: Functional View illustrates the functional view of the general architecture.

The functional view is divided into 3 different blocks. The first part comprises all functionalities related to the infrastructure, e.g. traffic lights, RSU and processing units.

In the middle is the backend block for processing the data and hosting the functions and modules that provide the services targeted in IN2CCAM. The required resources can be provided in the cloud or on edge hardware.

The block on the right includes all services that are provided in IN2CCAM.

This general functional view takes into account all the functionalities and modules that the Living Lab architectures have in common.

Because the services provided in the cities are vastly different, it is not practical to include them all in one functional view. Instead, they are grouped according to their specific characteristics in the relevant service modules to enhance differentiation among services within their local contexts, each city will feature a distinct functional view in Chapter 4. This view will represent the services and consider the local architecture.

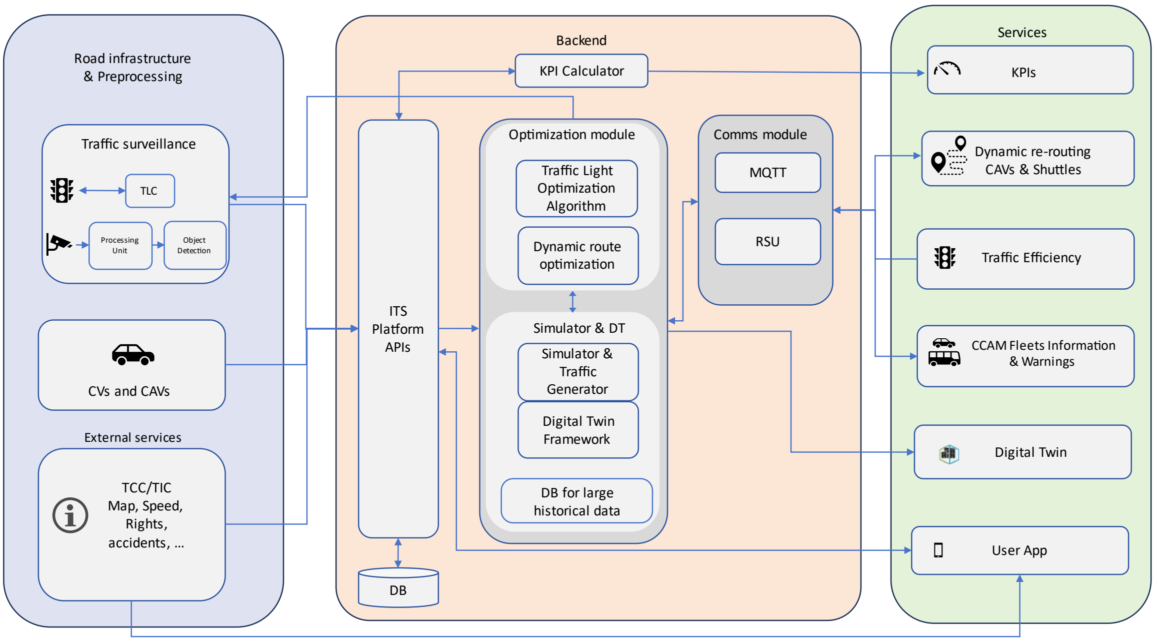

Figure 1: Functional View

3.1.1 Backend description

ITS Platform APIs play a vital role in enabling seamless communication among various components within the V2X (Vehicle-to-Everything) ecosystem. By adhering to standardized protocols and formats, these APIs ensure interoperability and compatibility across diverse devices and systems. Specifically, ITS Platform APIs provide traffic management applications with access to data collected from sensors and devices deployed on roads and vehicles. This valuable data can be utilized for tasks such as traffic monitoring, congestion management, and optimizing the overall traffic flow. Most entities within the ecosystem engage in information exchange via ITS Platform APIs, making them a fundamental building block for efficient transportation systems. Numerous examples include Vehicle-to-Everything (V2X) APIs and Public Transit APIs.

Furthermore, databases are a reliable infrastructure for storing, managing, and analyzing the data generated by connected vehicles, infrastructure, and other components of the transportation system. Their critical role extends beyond mere data storage; they facilitate data-driven decision-making, enhance transportation efficiency, and contribute to overall road safety. In the data flow, databases interact bidirectionally with ITS Platform APIs, forming a crucial link in the exchange of information within the V2X ecosystem. Notable examples include relational databases such as MySQL and PostgreSQL, as well as non-relational databases such as DynamoDB and MongoDB.

Decision Making and General APIs are essential for Intelligent Transportation Systems (ITS). These APIs facilitate decision-making across multiple levels: vehicle, roadside unit, and centralized controller. Decision makers utilize incoming data to evaluate the current traffic situation and determine the most suitable actions. Their objectives include enhancing traffic flow, managing traffic lights, enhancing safety, and improving overall efficiency. The data flow involves interactions between Decision Making & General APIs, ITS Platform APIs, MQTT, RSU, and diverse services. For instance, edge devices such as on-board computers and RSUs are pivotal, while control centers such as Traffic Management Centers, Fleet Management Systems, and cloud-based services contribute to effective decision-making processes.

In addition, the digital twin, or at least some form of access to collected traffic data, is an essential component of ITS. Its purpose is to seamlessly integrate real-time data, simulation, and environmental data to create a virtual representation of physical objects and systems within the transportation ecosystem. By utilizing advanced capabilities in modeling, simulation, and analytics, this framework generates a digital representation of the entire transportation system. This digital representation facilitates greater understanding, prediction, and optimization of transportation processes, resulting in improved safety, efficiency, and sustainability in urban mobility. The data flow involves interactions between the Digital Twin Framework, ITS Platform APIs, RSU, and the Digital Twin Service. A few examples of Digital Twin systems include Siemens Digital Twin, Bosch Digital Twin Tools, and NVIDIA Omniverse.

Closing, API-Interfaces are crucial for enabling efficient and reliable communication across various components of transportation systems. These interfaces enable seamless data exchange between vehicles, infrastructure, and central servers. When a vehicle or infrastructure system requires to communicate with other services, API-Interfaces come into play. They act as a communication bridge between decision-making/general APIs, Digital Twin frameworks and services. Two notable examples of API-Interfaces are MQTT (a lightweight messaging protocol for real-time data exchange) and RSUs (Roadside Units), which serve as dedicated devices in vehicles. By integrating these interfaces, cooperative and connected transportation environments can be incorporated, resulting in improved safety, efficiency, and mobility for road users.

3.1.2 Service descriptions

Traffic light control in a Digital Twin environment plays a vital role in optimizing traffic flow, enhancing safety, reducing environmental impact, and enabling a seamless integration with connected vehicles for efficient and intelligent transportation systems. The system can be employed to enhance traffic efficiency, for instance, by implementing green waves. Furthermore, sophisticated algorithms can be utilized to dynamically adjust traffic light timings in accordance with real-time conditions, thereby optimizing traffic flow in response to fluctuating demands.

Dynamic re-routing plays a pivotal role in optimizing traffic flow, enhancing efficiency, and improving safety. Furthermore, it provides drivers with real-time navigation assistance, enabling them to navigate the road network more effectively.
The system can be employed for the purpose of routing, with the objective of optimizing traffic flow by diverting vehicles away from congested areas or traffic incidents. By analyzing real-time traffic data, including congestion levels, road conditions, and incidents, the system is capable of dynamically updating routes to distribute traffic more evenly across the road network, with the consequence of reducing congestion and travel times.

CCAM Fleets Information and Warnings facilitate the enhancement of safety, improvement of traffic efficiency, enhancement of the passenger experience, facilitation of efficient fleet management, and promotion of integration with infrastructure components for the development of intelligent and sustainable transportation systems.
As a central element, the GLOSA App acquires data from a number of sources, including the TLCs, the AVs and additional data from the local server.

The Digital Twin employs advanced technologies such as the Internet of Things (IoT), data analytics, and simulation to create virtual representations of reality that enhance understanding, enable optimization, and drive innovation across various domains. This results in improved performance, efficiency, and sustainability, for example, by providing users with information about parking and UVAR status.

The User App provides users with access to real-time information, personalised services, navigation, and routing assistance, as well as safety and security features. For instance, the Vigo Driving App offers users traffic light and vehicle information.

3.2 Modular View

The incorporation of a modular perspective within the proposed architecture is crucial for numerous reasons. Firstly, modularity significantly improves scalability by facilitating the seamless integration of new features as the technological requirements evolve over time. This adaptability ensures that the system remains viable, accommodating advancements in autonomous vehicle technology and urban infrastructure without requiring a complete overhaul.

Furthermore, a modular architecture fosters interoperability, allowing various components to communicate effectively and share resources. Additionally, for the successful implementation of the architecture and functionalities outlined in Section 3.1, it is essential to have the appropriate hardware and communication infrastructure. Therefore, we have summarized these requirements within a modular framework.

The Modular View shown in Figure 2 describes the connections of the architecture where data is being generated, processed and used. It gives an overview of the communications between the modules necessary to provide a service.

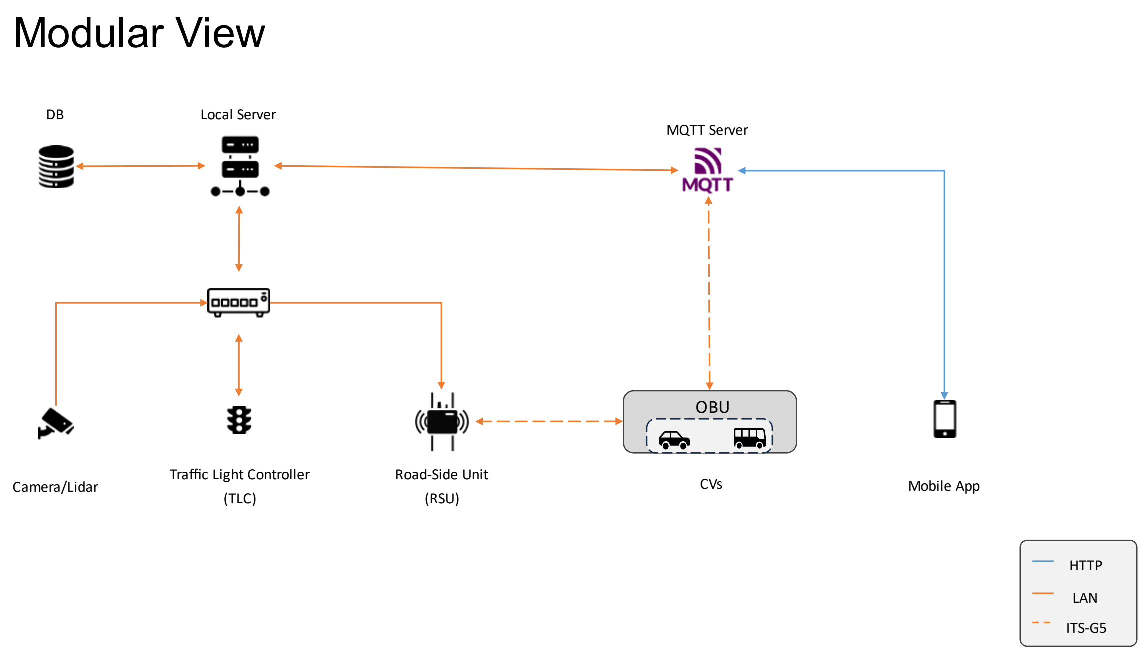

Figure 2: Modular view: Showing High Level view of different components used

3.2.1 Description of the Modules

The modular view depicts the respective modules, each providing a specific function and playing a significant role in the proposed general architecture. First, the Database module is a repository of critical data, such as vehicle movement, traffic pattern, historical trend, and user preference data. Additionally, this data storage and management system provides real-time decision-making and optimization of the necessary operations.

The Local Server module serves as a communication hub, hosting applications for data processing, algorithmic decision-making, and coordination among various components of this ecosystem. It facilitates bidirectional communication with physical infrastructure elements and vehicles, ensuring seamless operation and responsiveness to dynamic urban conditions.

The MQTT Server module utilizes a lightweight messaging protocol to enable efficient data exchange between connected vehicles, roadside units, and centralized systems. This ensures efficient information dissemination and system-wide synchronization.

The Mobile App module provides a user-friendly interface for accessing smart city services and information. It provides features such as real-time traffic updates, navigation assistance, and personalized recommendations, enabling users to navigate the city efficiently while contributing to the collective intelligence of the smart city ecosystem.

Embedded within autonomous vehicles, cameras and LiDAR sensors capture real-time data on surrounding objects, obstacles, and traffic conditions, enabling autonomous navigation and hazard-free navigation. These sensors enhance the safety and operational efficiency of autonomous vehicles within urban environments.

Regulating traffic flow at intersections is the Traffic Light Controller (TLC) module, which automatically adjusts signal timings based on actual-time traffic conditions and system priorities. Integrated with the smart city network, the TLC optimizes traffic flow, minimizes congestion, and enhances intersection safety.

Road-Side Units (RSUs) provide communication nodes along roadways, facilitating vehicle-to-infrastructure communication and transmitting critical information regarding road conditions, traffic advisories, and safety alerts. This enhances situational awareness and promotes safer mobility within the smart city environment.

Connected Vehicles (CVs) establish bidirectional communication with the smart city infrastructure, utilizing real-time data on vehicle dynamics, traffic conditions, and environmental factors. This connectivity enables cooperative driving, adaptive navigation, and enhanced safety features, resulting in the resilience and efficiency of the smart city ecosystem.

The Modular View describes the architecture in which data is generated, processed and used. It describes the different hardware and communication modules which are part of the specific services of the LLs.
Furthermore, it defines the hardware, the communication standards (MQTT, OBU, Switcher, …) and the different protocols (HTTP, 5G, ITS-G5, Local Network…) as well as the servers (local, cloud…) and Sensors (Camera, TLC…).

3.3 Scenario Flow

The scenario flow section outlines numerous scenarios described in D2.3 and demonstrates how they utilize the General Architecture for their data flows. These scenarios serve as practical examples, illustrating how the architecture components interact and exchange data in various contexts. By examining these scenarios, stakeholders can gain insights into the system's behaviour, identify potential bottlenecks, and refine their understanding of the architecture's effectiveness.

3.3.1 Scenario: GLOSA Application

The GLOSA (Green Light Optimal Speed Advisory) application serves a dual purpose for human drivers and autonomous vehicles (AVs). The GLOSA app provides real-time information on traffic light status and optimal speed for approaching intersections. This data empowers the human driver to make informed decisions, adjusting the vehicles speed to synchronize with green lights, thereby optimizing traffic flow and enhancing safety. Simultaneously, within the AV's internal system, GLOSA information guides the fully autonomous driving function. By adjusting its speed based on real-time traffic signal data, the AV aims to pass through intersections without stopping, contributing to efficient urban mobility, and minimizing delays.

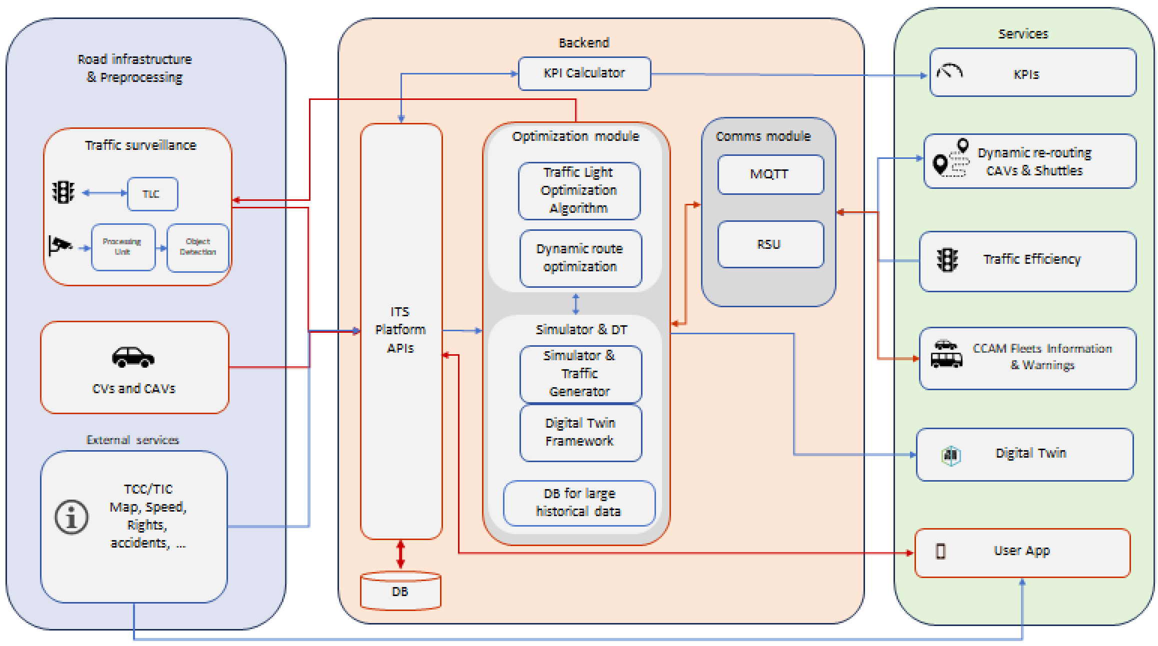

Figure 3: GLOSA scenario flow

As depicted with the red arrows on the diagram above, the 'CCAM Fleet Information & Warnings' module informs the 'Optimization' as well as the 'Simulator & DT' blocks via the 'Comms' module, providing critical information to the infrastructure. The 'Optimization' module processes the received data, which can calculate the optimal speed for approaching intersections based on traffic light status, distance, and other relevant factors. Before deployment, the proposed speed advisory is validated and tested using a Simulator or Digital Twin.

Moreover, the Traffic Surveillance module continuously monitors real-time traffic conditions, collecting data from various sources such as traffic cameras and sensors. This data is then relayed to the Connected Vehicles (CVs) and Connected Autonomous Vehicles (CAVs) equipped with communication capabilities. The respective information is finally sent to the GLOSA Application via Transportation System Platform Application Programming Interfaces (ITS APIs). This allows the User Application to interact with the AV. For AVs with external safety drivers, it provides real-time speed advisories. For fully autonomous AVs, it guides decision-making to optimize speed and enable efficient passage through intersections.

3.3.2 Scenario: AV detected event

When an autonomous vehicle (AV) detects an event, such as a slippery road due to rain or ice, an AV blocking a lane or tram line, or an accident involving another vehicle, it immediately sends a notification to both the Remote Operations Center (ROC) and the Cooperative, Connected, and Automated Mobility (CCAM) services. The ROC acts as a central hub for managing AV operations and safety. Upon receiving the alert, the ROC shares this warning with other AVs operating in the same location. These AVs adjust their behaviour accordingly, slowing down, rerouting, or taking other necessary steps to avoid the danger. In parallel, the CCAM services provide a crucial role in ensuring the safety and coordination of individuals. They extend the warning beyond AVs to potential external entities. Public transport management centers receive real-time alerts about road conditions or incidents. Using this information, they can adjust public transportation schedules, reroute buses or trams, and take preventive measures to minimize disruptions. By reducing these warnings across AVs and external entities, the system enhances safety, reduces accidents, and promotes efficient traffic management. The diagram below illustrates this scenario by illustrating the data flow between the discrete modules with red arrows.

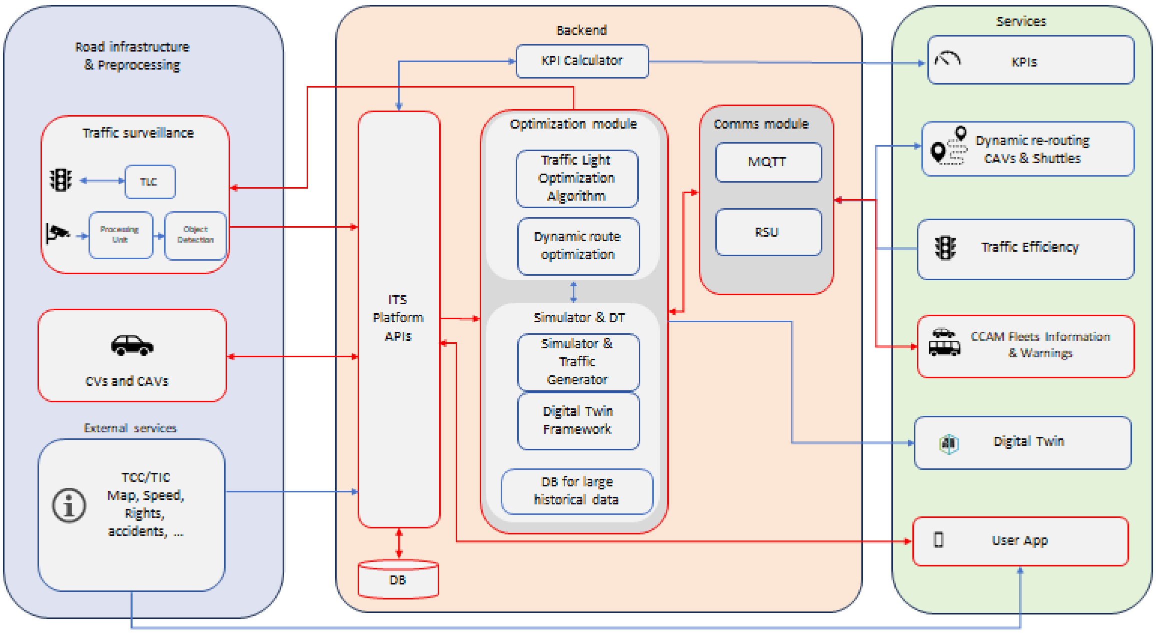

Figure 4: AV detected event Scenario Flow

3.3.3 Scenario: Dynamic re-routing of AV based on traffic-load

The Connected and Cooperative Automated Mobility (CCAM) vehicle operates within the city, collecting data on its position and destination. The C-ITS (Cooperative Intelligent Transport Systems) platform receives this information through the data-sharing architecture. The platform utilizes computed traffic-load data and the CCAM vehicle's position, enabling it to accurately calculate the optimal route to the destination. Load-balancing algorithms are crucial in determining the most efficient path. The C-ITS platform transmits the updated routing information to the CCAM vehicle using the same data-sharing architecture. The CCAM vehicle utilizes this routing information into its automated driving decision algorithm. Additionally, due to driving restrictions in certain corridors, this part will be simulated to assess the impact of load-balancing.

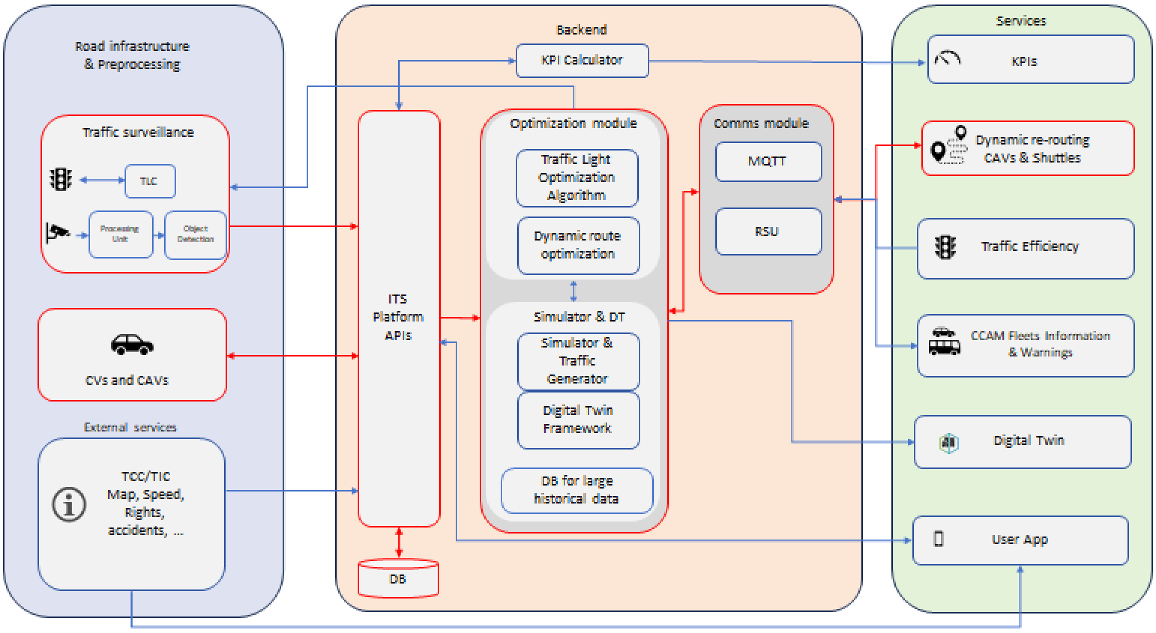

Figure 5: Dynamic Re-routing Scenario Flow

4 LOCAL ARCHITECTURES

4.1 Tampere LL

The objective of the Tampere Living Lab (LL) in IN2CCAM project is to develop and introduce improvements in the local last-mile mobility for people in Hervanta. In Tampere LL, the CCAM services are demonstrated with automated shuttles (and a passenger car) together with a Remote Operation Centre (ROC) in a real traffic environment. CCAM services in AV vehicles are implemented as a mobile phone app, which will be provided to safety operators of external AV shuttles or integrated to the Automated Driving System (ADS) of VTT automated vehicles. In Tampere LL, three CCAM services and scenarios are implemented which were initially defined in deliverable D2.3.

4.1.1 Generalized vs LL-Specific Architecture

Tampere LL architecture differs only slightly from the generalized architecture. Figure 6 shows the modified version of the functional view of the generalized architecture where some of the modules are removed, and one additional module has been added. The following modules have been removed from the generalized architecture as related services are not implemented in Tampere:

- Road infrastructure and Preprocessing: Traffic surveillance
- Local server backend:
  - Traffic Light Optimization Algorithm
  - Dynamic route optimization
  - Digital Twin: Framework (connected with simulations)
  - RSU
- Applications/services:
  - Traffic efficiency
  - Dynamic re-routing

The following modules are added or moved in Tampere LL compared to the generalized architecture functional view:

- Remote Operation Centre for AV fleet
  - Digital Twin application (used to support the ROC operator)

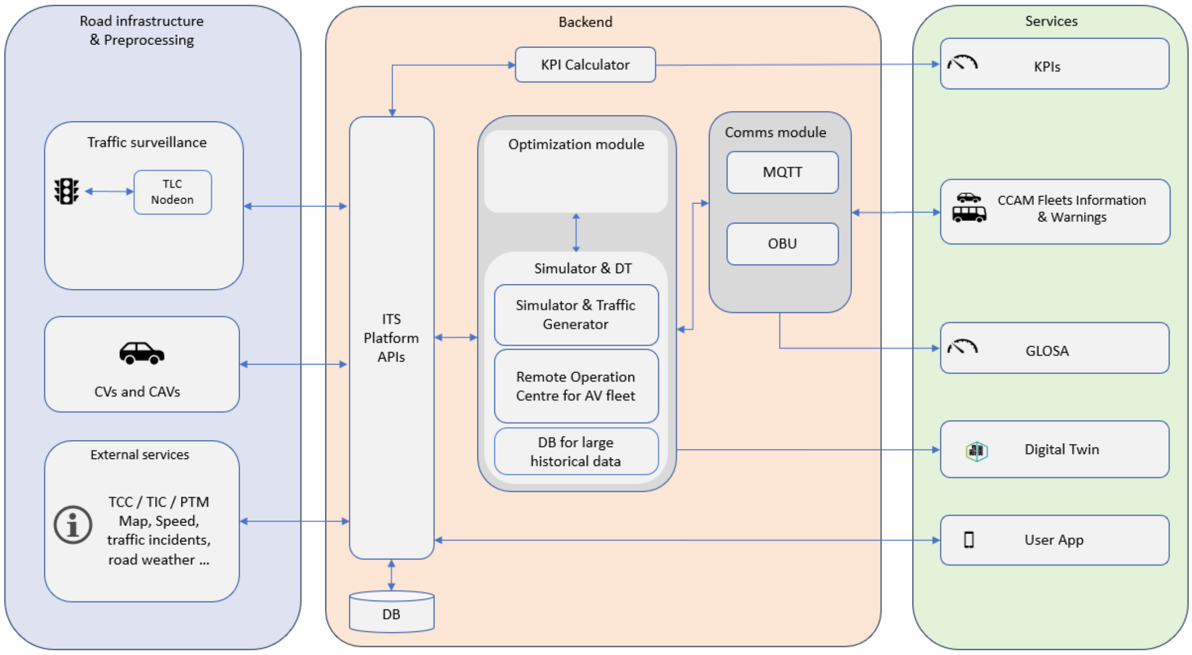

Figure 6: Tampere LL functional architecture, modified from the generalized architecture

4.1.2 Overview of the Various Modules

4.1.2.1 Remote Operation Center modules

Remote Operation Center (ROC) contains the following modules running on the backend server at VTT: ROC HMI (Human Machine Interface), AV event application, Tram warning application, Road condition warning and Digital twin. ROC HMI is the main monitoring tool for the ROC operator. It shows all the needed information on a single map in Tampere area including real time location of the Connected and Autonomous Vehicle (CCAV) fleet, events detected and reported by CCAVs, real-time status of the traffic lights, and traffic warnings from Digitraffic, a service operated by the Finnish company Fintraffic for real time traffic information. AV event application is reading C-ITS MQTT topics and receives information to be presented on map. Road condition warning is reading current local road condition warnings from the Finnish Meteorological Institute (FMI) open data API, Tram warning application is reading data from Tampere city open public transport sources and calculates and sends warnings to the Tram warning android application. Digital Twin provides up-to-date overall 3D presentation for the ROC operator in Tampere, Hervanta area. Tampere LL is using a MQTT broker for queue/message service. It has separated topic trees for Remote Operation Center and for C-ITS messages.

4.1.2.2 CCAV modules

CCAV applications running in VTT automated vehicle OBUs are the following: SPAT & GLOSA, event detection, and HMI. In addition, there is a Tram warning mobile application running on an Android mobile. SPAT & GLOSA receives data from TLC and provides speed suggestions to the low-level control system of the Automated Driving System (ADS) of the vehicle. Event detection application is receiving data from two main sources of the AV; from the vehicle traction control system and from the environmental perception system. The former calculates the slippery level of the road surface, and the latter detects obstacles in front of the ego vehicle (e. g., on-the-route parked vehicles). The HMI is used to present the needed information to the safety driver of the CCAVs for testing and demonstration purposes. The Tram warning application is a self-contained android application used by the safety driver of the automated shuttle bus.

4.1.3 Insights into Modules and Data Flow

4.1.3.1 Remote Operation Center modules

Figure 7: Tampere LL ROC applications illustrates details of the Tampere LL Remote Operation Center applications. The Road condition warning gets warnings from FMI open data (using WSF protocol), presents all current local road weather warnings to the ROC operator, and sends current road condition warning level to CCAVs. The AV event app gets local current road incidents from Fintraffic and presents them on the ROC map. The tram warning User App gets real-time public transport vehicle location data from Tampere open data, and it broadcasts tram warning at relevant spots on the road network. AV event app shows the events on the ROC map and sends the (HLN-TSR / HLN-OR / HLN - SV) warnings to the VTT vehicles over MQTT.

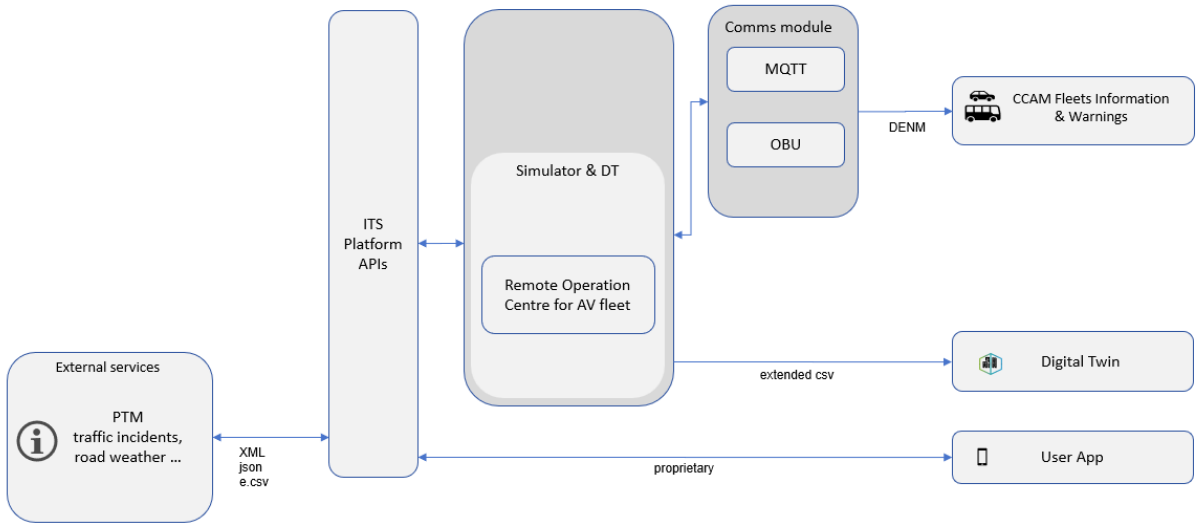

Figure 7: Tampere LL ROC applications

Table 1: Table of insights into Module and Data flow of Module 1 (Tampere LL)

Data Format	Data Link	Protocol	From	To	Data purpose
XML	LAN	WSF	External services / road weather	Road condition warning	Current local road weather is presented to the ROC operator.
json	LAN	DATEX II Light	External services / traffic incidents	AV event app	Is used to present local current traffic incidents on map.
extended csv	LAN	GTFS, Industry standard by Google	External services / PTM	Tram warning User App	Is used to get real time PT location data to tram warning server app
DENM	LAN	MQTT	ROC for AV fleet	CCAM Fleets	Tampere ITS Platform shares the (HLN-TSR / HLN-OR / HLN - SV) to the ROC operator.
extended csv	LAN	GTFS, Industry standard by Google	External services / PTM	Digital Twin	Is used to present real-time locations of trams and busses in Digital Twin

4.1.3.2 CCAV modules

Figure 8: Tampere LL CCAV applications presents the detailed message flow from and to CCAV OBU applications. SPAT & GLOSA are subscribing SPATEM and MAPEM json messages over MQTT protocol. Event detection is broadcasting DENM json messages over MQTT protocol. All the relevant road incidents on the road network are subscribed from MQTT using DENM json format.

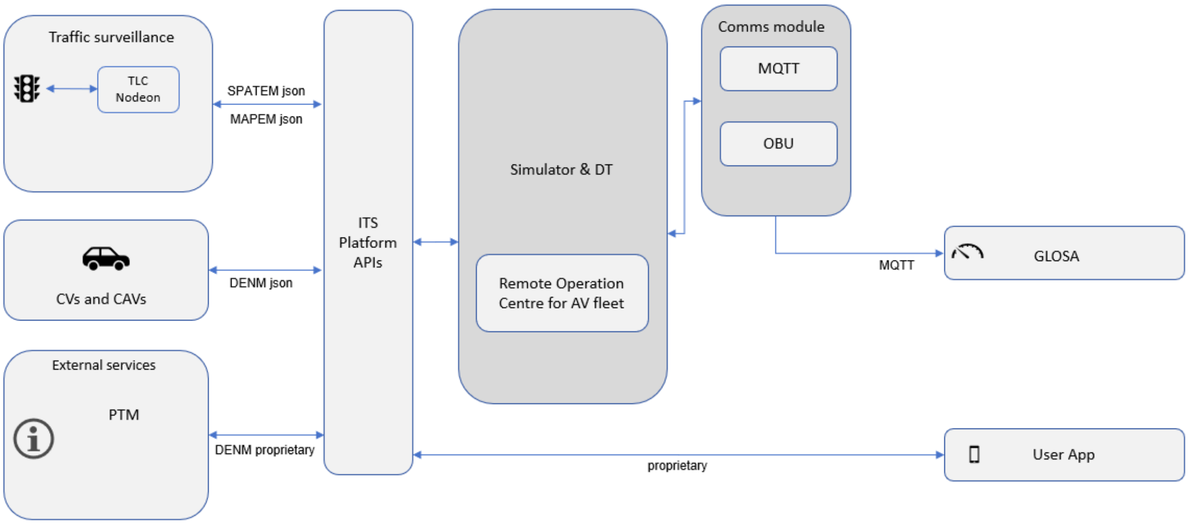

Figure 8: Tampere LL CCAV applications

Table 2: Table of insights into Module and Data flow of Module 2 (Tampere LL)

Data Format	Data Link	Protocol	From	To	Data purpose
SPATEM json	5G / 4GLTE	MQTT	TLC Nodeon	Vehicle OBU	Signal Phase and Timing Information (SI-SPTI), backend system transmits periodically and in real time the current phase state and predicted timing of the traffic lights and road topology for the intersection ahead.
MAPEM json	5G / 4GLTE	MQTT	TLC Nodeon	Vehicle OBU	Green Light Optimal Speed Advisory (SI-GLOSA), backend system transmits periodically and in real time the current phase state and predicted timing of the traffic lights and road topology for the intersection ahead.
DENM (proprietary)	5G / 4GLTE	MQTT	External services / PTM	User App	DENM message for Hazardous Location Notification (HLN)-PTVC and (HLN)-PTVS are used.
DENM json	5G / 4GLTE	MQTT	CAV	AV event app in ROC	Temporarily slippery road (HLN-TSR), DENM message for HLN-TSR is used.
DENM json	5G / 4GLTE	MQTT	CAV	AV event app in ROC	Obstacle on the road (HLN-OR)
DENM json	5G / 4GLTE	MQTT	CAV	AV event app in ROC	Stationary vehicle (HLN - SV)

DENM	5G / 4GLTE	MQTT	ROC	CAV	ROC shares the (HLN-TSR / HLN-OR / HLN - SV) warning to relevant vehicles on fleet.

4.2 Trikala LL

4.2.1 Generalized vs LL-Specific Architecture

The architecture supporting the on-site demonstrations of the autonomous service in Trikala, Greece, is designed to seamlessly integrate various components, all aimed at enhancing urban mobility and safety. An outline of the architecture is presented below.

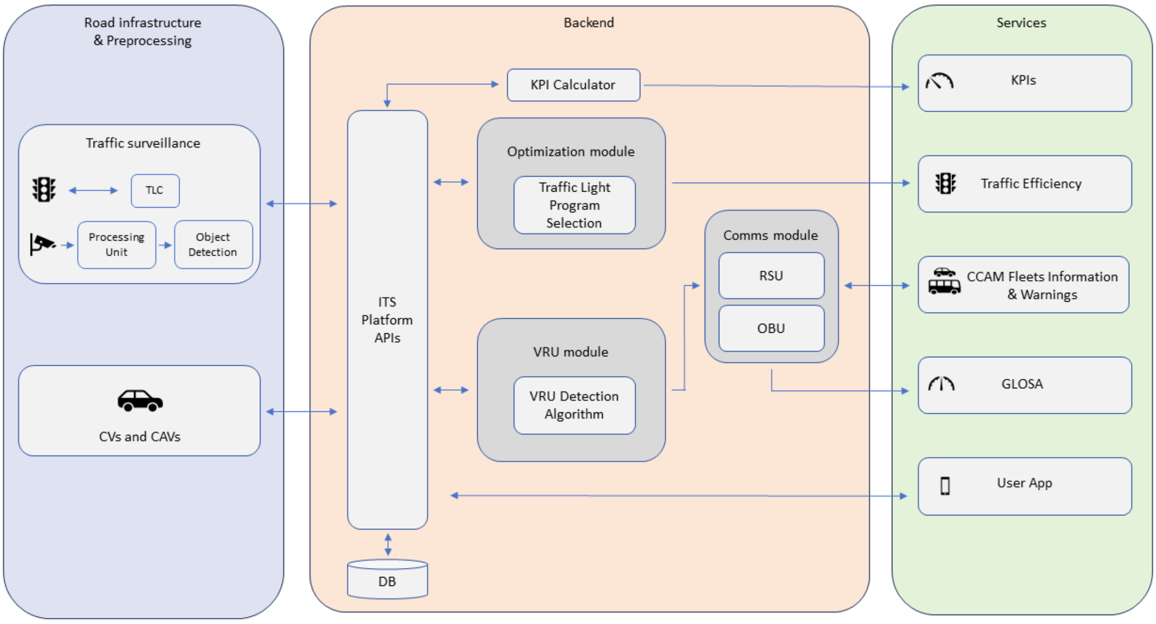

Figure 9: An overall visual representation of the Trikala LL architecture

Drawing from the general IN2CCAM architecture as a reference, adjustments, additions, and exclusions have been implemented to adapt the framework to the specific requirements of Trikala. This is to ensure that the Trikala-specific framework reflects the innovations being developed and demonstrated in this pilot project. The main differences are:

Route infrastructure & Processing

- External services TCC/TIC (removed).

Backend

- The dynamic route optimisation algorithm/component in the optimization module (removed).
- A database for large historical data in the Simulator and Digital Twin (DT) component (removed).
- An MQTT broker in the comms module (removed).
- An On-Board Unit (OBU) in the comms module (added).
- The simulator & Traffic Generator module (removed)
- A VRU module (added).
- The Traffic Light Optimization Algorithm (removed).
- The Traffic Light Program Selection (added).

Services

- The DT service (removed).
- The GLOSA service (added).
- Dynamic re-routing CAVs & Shuttles service (removed).

Note: CV and CAV not directly connected to traffic surveillance. Modules connect only with the infrastructure through the ITS platform APIs.

4.2.2 Overview of the Various Modules

Module 1: Traffic Light Program Selection
This module describes the interaction between the backend processes and the physical traffic lights. This bi-directional connection also involves the traffic light controllers, which are physically installed at the signals, and the ITS Platform APIs developed to this aim.

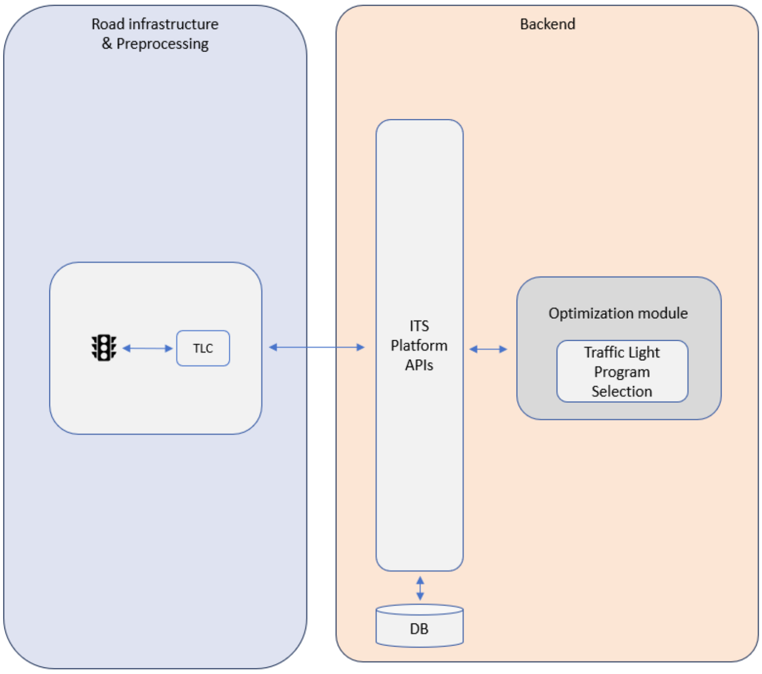

Figure 10: Module 1 'Traffic Light Program Selection' of the Trikala LL architecture

Module 2: Object detection

Stationary AI traffic cameras, mounted on traffic poles, have been configured to detect Vulnerable Road Users (VRUs) that cross the road in an uncontrolled way (I. e., without using designated crosswalks or traffic signals). When such an event is detected, a backend process is initiated to prompt the transmission of C-ITS messages from the RSU installed at the crossing.

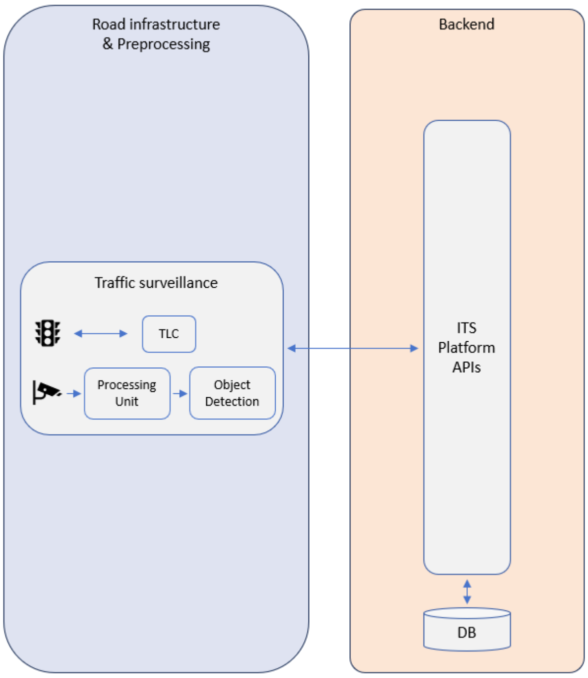

Figure 11: Module 2 'Object detection' of the Trikala LL architecture

Module 3: Infrastructure to Vehicle (I2V)

This module refers to the established communication between the RSUs installed at the signalised junctions of the AV route in Trikala with the OBUs mounted on top of the automated minivans. The ITS platform (backend) communicates directly with the RSU to exchange data required for the C-ITS solutions of the pilot.

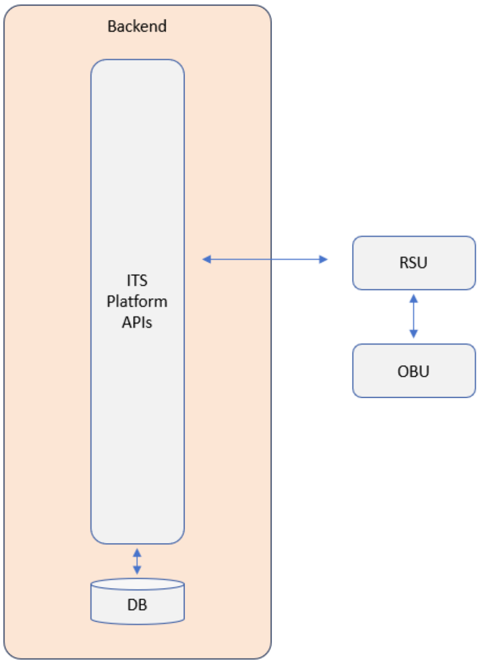

Figure 12: Module 3 'Infrastructure to Vehicle' of the Trikala LL architecture

Module 4: OBU to C-ITS services

This module includes the interaction and data flows between the OBUs mounted on top of the AVs and the final outputs of the C-ITS services developed around them (I. e. VRU warning, GLOSA).

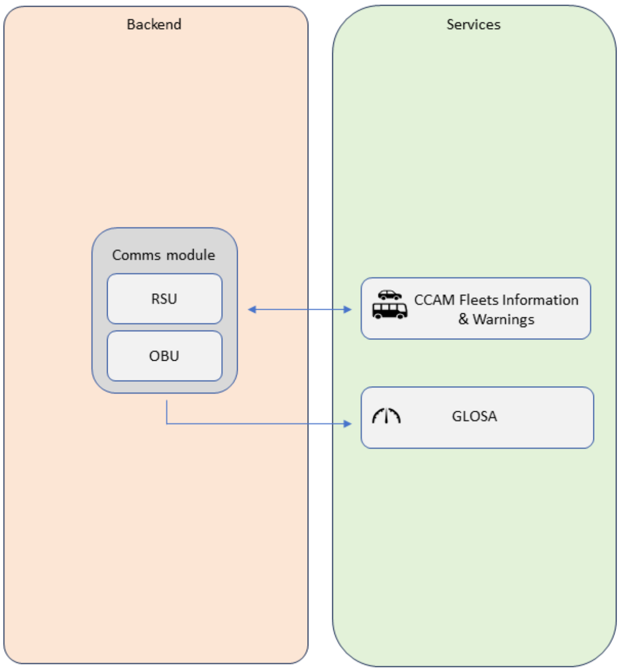

Figure 13: Module 4 'OBU to C-ITS' of the Trikala LL architecture

Module 5: Journey planning mobile application
This module includes the journey planning functionalities offered by the mobile application developed in the Trikala LL. The application, among others, allows its users to book on-demand trips with the automated services offered by the retrofitted minivans, view the available public transport options for the journey they wish to make, and plan trips using any combination of the available modes.

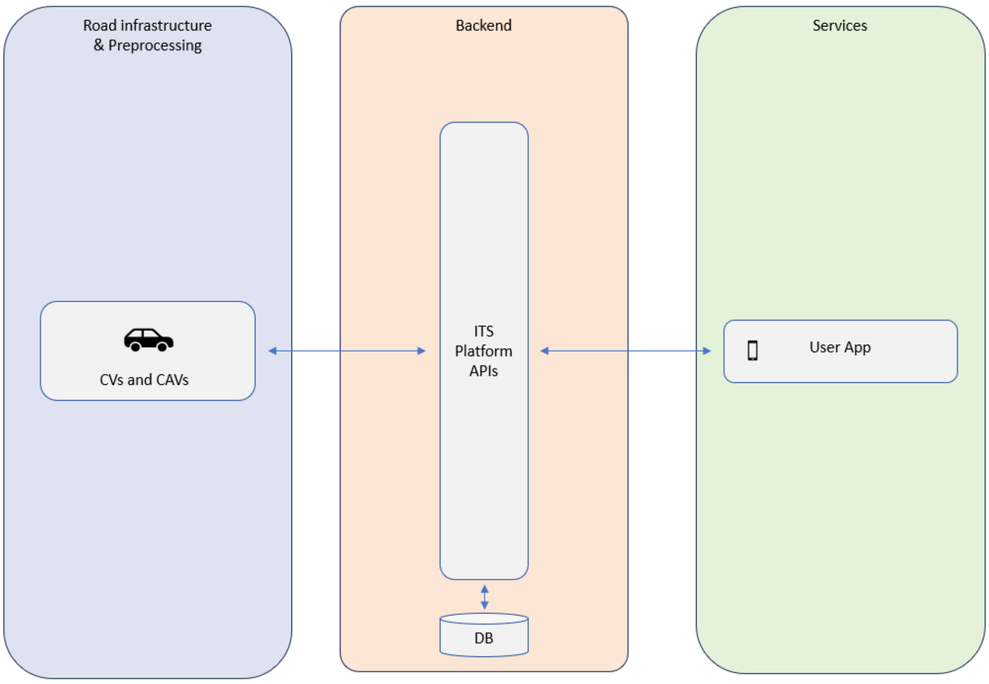

Figure 14: Module 5 'Journey planning mobile application' of the Trikala LL architecture

Module 6: KPI calculator

The ‘KPI calculator’ module involves the required processes to calculate the KPIs which are defined in IN2CCAM for the applications developed for the Trikala test pilot. The necessary datasets for those calculations are all stored in local databases in Trikala and include several types of data, such as vehicular data (e. g. coordinates, speeds) and signal traffic counts from the AI cameras.

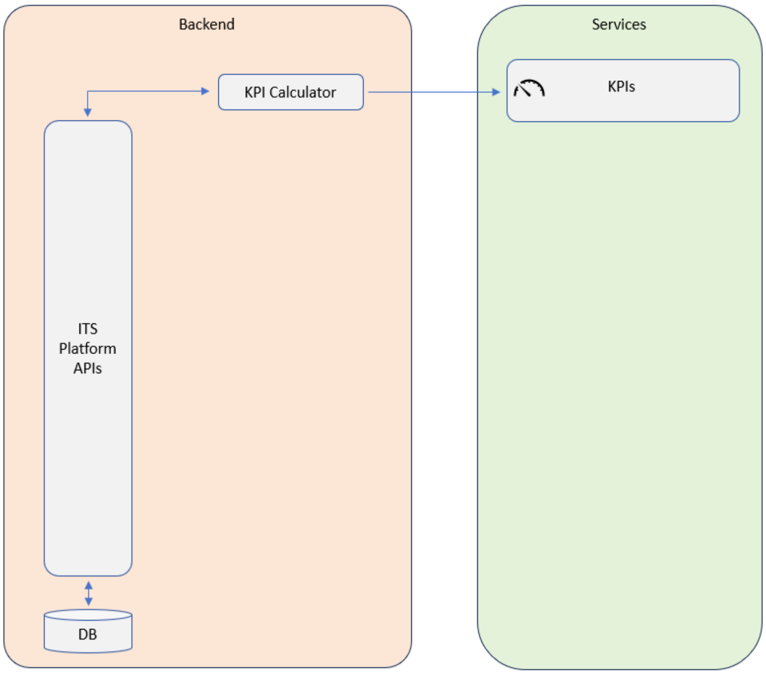

Figure 15: Module 6 ‘KPI calculator’ of the Trikala LL architecture

4.2.3 Insights into Modules and Data Flow

Module 1: Traffic Light Program Selection
The data flow between the traffic lights and the Traffic Light Controllers (TLC) takes place in a proprietary system, whereas APIs are utilised to achieve the exchange of data between the controllers and the ITS platform. The interaction within the backend, in other words the connections with the database and the optimisation module is achieved via internal connections.

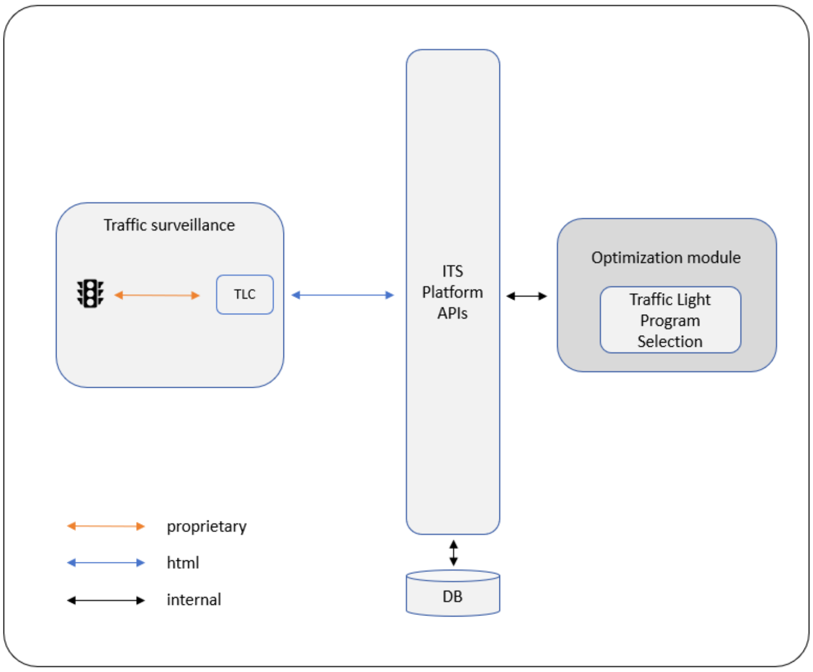

Figure 16: Diagram of insights into Module and Data flow of Module 1 (Trikala LL)

Table 3: Table of insights into Module and Data flow of Module 1 (Trikala LL)

Data Format	Data Link	Protocol	From	To	Data purpose
JSON	Cellular	HTML	TLC	ITS Platform	Provide signal timing data.
JSON	Cellular	HTML	ITS Platform	TLC	Provide signal timing data

Module 2: Object detection

The AI traffic cameras continuously record traffic data and classify the inputs into VRU groups, types of vehicles, etc. This is done internally within a proprietary system that involves the cameras themselves and a processing unit physically installed at the relevant intersections. The data flow from the processing unit to the ITS platform is done via APIs, and then the storage to a local database is achieved via an internal process.

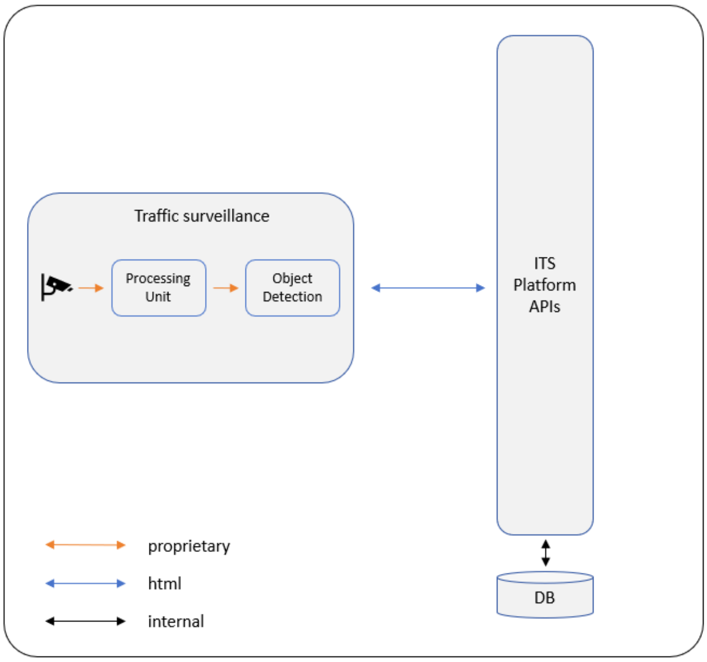

Figure 17: Diagram of insights into Module and Data flow of Module 2 (Trikala LL)

Table 4: Table of insights into Module and Data flow of Module 2 (Trikala LL)

Data Format	Data Link	Protocol	From	To	Data purpose
JSON	Cellular	HTML	Camera's processing unit	ITS Platform	Provide classified traffic data.

Modules 3 & 4: Infrastructure to Vehicle (I2V) & OBU to C-ITS services

Modules 3 & 4 have been combined here to showcase the whole route between the infrastructure and specific C-ITS services developed for the demonstrations in Trikala, namely the GLOSA and VRU warning applications. The ITS platform exchanges information with the RSU via API calls which in turn uses C-ITS messages to communicate with the OBU. A Human Machine Interface (HMI) then displays the relevant information coming from the OBU.

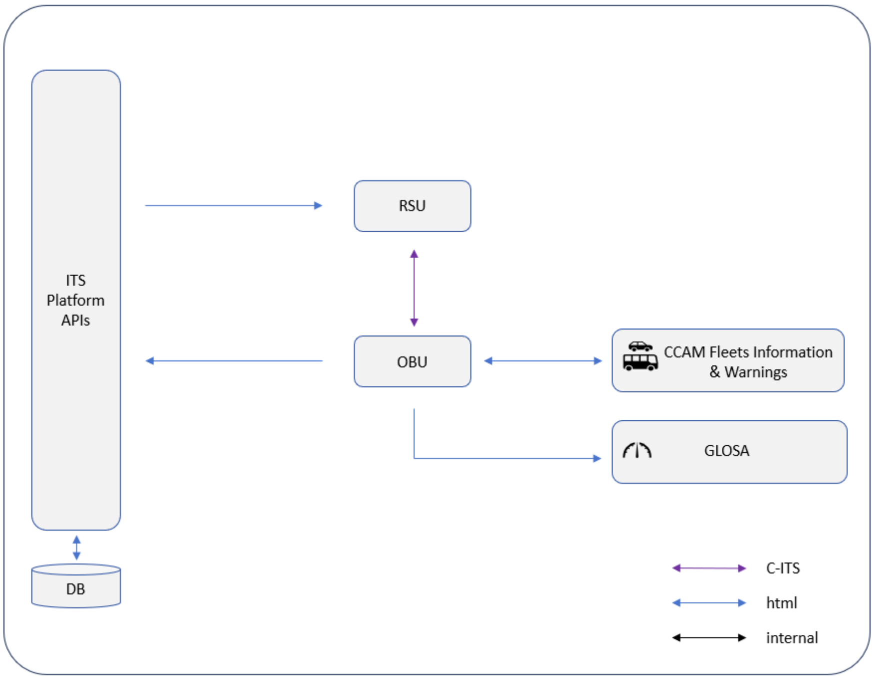

Figure 18: Diagram of insights into Module and Data flow of Modules 3 & 4 (Trikala LL)

Table 5: Table of insights into Module and Data flow of Modules 3 & 4 (Trikala LL)

Data Format	Data Link	Protocol	From	To	Data purpose
ETSI DENM	ITS-G5	ETSI GN	RSU	OBU	Provides information about hazardous locations on the road
ETSI MAPEM	ITS-G5	ETSI GN	RSU	OBU	Provides the geometry of signalized intersections
ETSI SPATEM	ITS-G5	ETSI GN	RSU	OBU	Provides the traffic light status of signalized intersections
XML	Cellular	HTML	ITS Platform	RSU	Triggers DENM broadcasting
XML	Cellular	HTML	OBU	ITS Platform	Provides vehicular data.
XML	LAN	HTML	OBU	HMI	Calculates GLOSA and displays outputs (i.e. VRU warnings, GLOSA output).

Module 5: Journey planning mobile application

In this module, the data flows between the automated minivans, the C-ITS platform, and the user mobile application which are facilitated via HTTP.

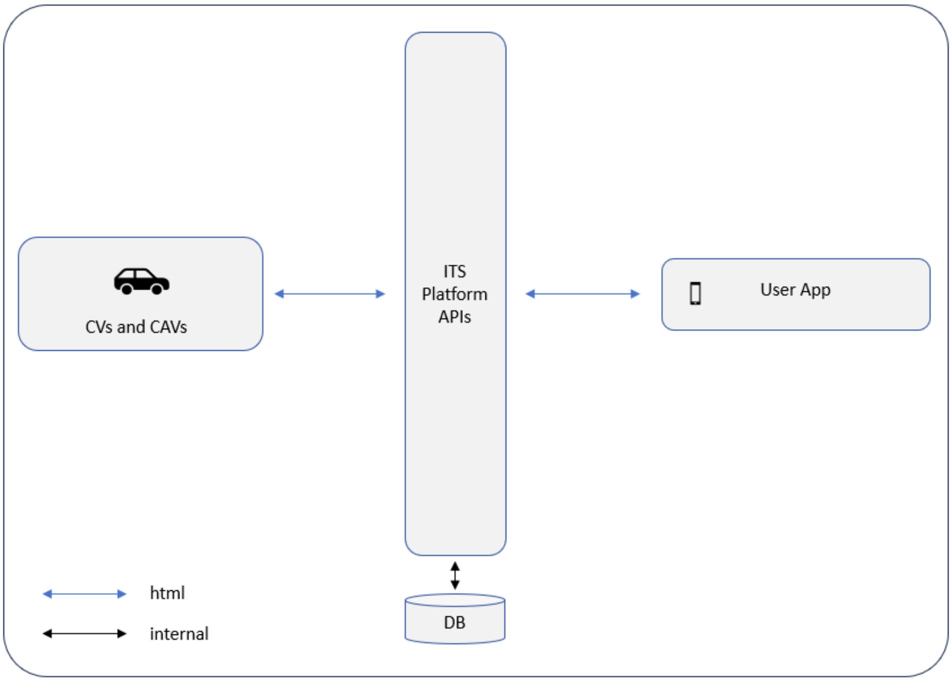

Figure 19: Diagram of insights into Module and Data flow of Module 5 (Trikala LL)

Table 6: Table of insights into Module and Data flow of Module 5 (Trikala LL)

Data Format	Data Link	Protocol	From	To	Data purpose
JSON	Cellular	HTML	CAVs	ITS Platform	Provides vehicular data and confirms/rejects trips requested.
JSON	Cellular	HTML	ITS Platform	CAVs	Requests trips.
JSON	Cellular	HTML	User app	ITS Platform	Requests trips.
JSON	Cellular	HTML	ITS Platform	User app	Confirms/rejects trips and provides vehicular data.

Module 6: KPI calculator

There are no data flows to be discussed in this module, since all the processes are internal towards the computation of the project's KPIs. The main components of the module can be seen in .

4.3 Turin LL

Turin LL is responsible for the implementation and demonstration of Use Case #3, i.e., Dynamic traffic management, for improving road safety and reducing the likelihood of traffic congestion during roadworks and other hazardous events on the road. In particular, the goal of the LL is to illustrate two scenarios of "dynamic re-routing" of CVs and CAVs: the first scenario involves the exploitation of data about the vehicles in the monitored area, whereas the second scenario includes additional information about parking lot availability and the status of UVAR.

4.3.1 Generalized vs LL-Specific Architecture

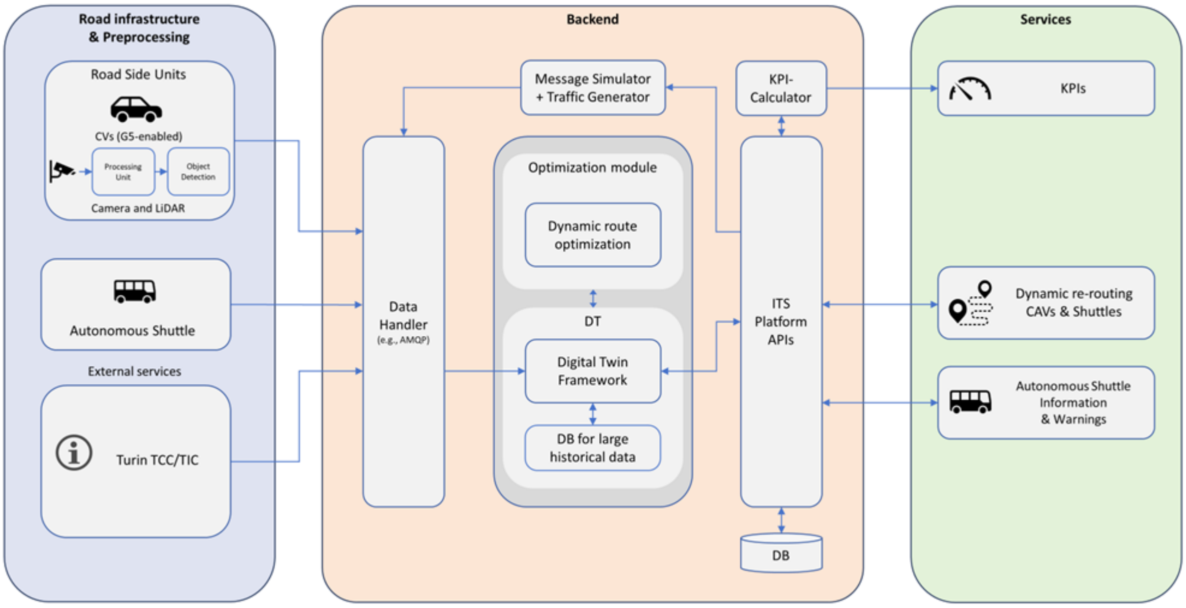

Figure 20: An overall visual representation of Turin LL-Specific Architecture

Figure 20: An overall visual representation of Turin LL-Specific Architecture shows the LL-specific functional architecture. With respect to the generalized architecture of the project, the infrastructure of Turin LL comprises two Road Side Units – equipped with camera and LiDAR sensors – and the city's TCC/TIC.

The services demonstration focuses on hybrid scenarios, which consider both synthetic and real-world data to assess varying rates of connected vehicles penetration. Real-world data is collected using C-V2X communication devices from an autonomous shuttle that is operating in the designated area. In addition, by taking advantage of the installed RSUs' short-range communication capabilities, messages from G5-capable commercial vehicles – like Volkswagen – are gathered. As recommended by C-Roads, a pub-sub methodology based on AMQP 1.0 is used to manage all incoming data flows, i.e., V2I and I2I communication channels. Synthetic data is received via the "Data Handler" module, just like actual data, and forwarded to the DT instead of the ITS Platform APIs as illustrated in the common architecture.

The "dynamic re-routing" service implemented in the Turin LL is directly connected to the Digital Twin, in contrast to the generalized architecture. This enables automatic "on-demand" service management, particularly in the case that a potentially dangerous event is reported to the DT. The output of the service can then be used by the DT to suggest alternative routes to the connected vehicles (real and simulated ones).

The LL architecture does not include any functionality relating to traffic light optimization. Nonetheless, the city's TCC shares via ETSI C-ITS SPATEM the status of traffic lights located along the automated shuttle's route.

The adoption of standardized technical and service specifications (ETSI C-ITS messages, DATEXII protocol, C-Road's service recommendations, etc.) and data exchange via the IN2CCAM data platform enable interoperability between the Turin LL and other current and future platforms (e.g., C-ITS-S, TCC, TIC).

4.3.2 Overview of the Various Modules

4.3.2.1 Infrastructure

ROAD SIDE UNITS

Each Road Side Unit provides an additional source of data, by enabling the sharing of information about its surroundings with the Digital Twin deployed on the local server.
The percentage of commercial CVs is low nowadays, and most of them are only short-range communication capabilities. As a result, the installation of infrastructure alongside the roads guarantees the collection – and subsequent exploitation - of the messages produced by such vehicles. Moreover, the RSU is equipped with camera and LiDAR sensors to gather information from the surroundings, build an Environmental Perception Model and generate ETSI C-ITS Cooperative Perception Messages.

TURIN TCC/TIC

The City's TCC and TIC platforms – operated by 5T – are designed to monitor, forecast, and control traffic and mobility, providing users with reliable information about road network and transport services. The key components of the system comprise the following: the Regional Traffic Supervisor system (SVR); extra and urban traffic sensors; a floating car data aggregator module; systems for traffic light and urban traffic control; off-street parking data; enforcement devices for the Turin Limited Traffic Zone; DATEXII Node; traffic cameras; infomobility platform; and ITS stations. These elements collect and transmit data in real-time about road network conditions and traffic data, supporting the microsimulation framework of dynamic re-routing.

4.3.2.2 Local Server

DATA HANDLER MODULE

This module acts as a protocol and data interoperability layer. It receives and forwards data from external sources to the Digital Twin, e.g., ETSI C-ITS messages sent by the RSUs, the autonomous shuttle, and the TCC/TIC. To preserve adaptability of the overall architecture, this module will also accept messages generated by simulated vehicles. Depending on the information type, the data may be validated and/or processed before forwarding. Communication is performed via ActiveMQ, adopting AMQP 1.0 protocol.

MESSAGE SIMULATOR + TRAFFIC GENERATOR MODULE

The message simulator module is a web-based application that let users create geographic traces or upload pre-recorded ones (GPX or NMEA) that can be played back for generating realistic CAM messages, enabling the simulation of connected vehicles. It allows testing and validation of the scalability of CCAM systems in actual large-scale settings, without the requirement for connected vehicles on the streets. To improve simulation quality and provide an abstraction layer to the definition of vehicle paths, the message simulator has been integrated with a traffic generator module, i.e., SUMO (Simulation of Urban Mobility). This is helpful to obtain scenarios which are close to real-world traffic scenarios. A SUMO TraCI (Traffic Control Interface) client has been implemented to collect vehicle-related data from the simulator.

DYNAMIC ROUTE OPTIMIZATION MODULE

This module provides a dynamic rerouting on-demand service. Upon a hazardous event on the road, the DT submits a rerouting request. The findRoute script in TraCI is used to determine the best route between a given start and end edge on the network based on predetermined criteria (such the shortest journey time) and real-time traffic circumstances.

4.3.3 Insights into Modules and Data Flow

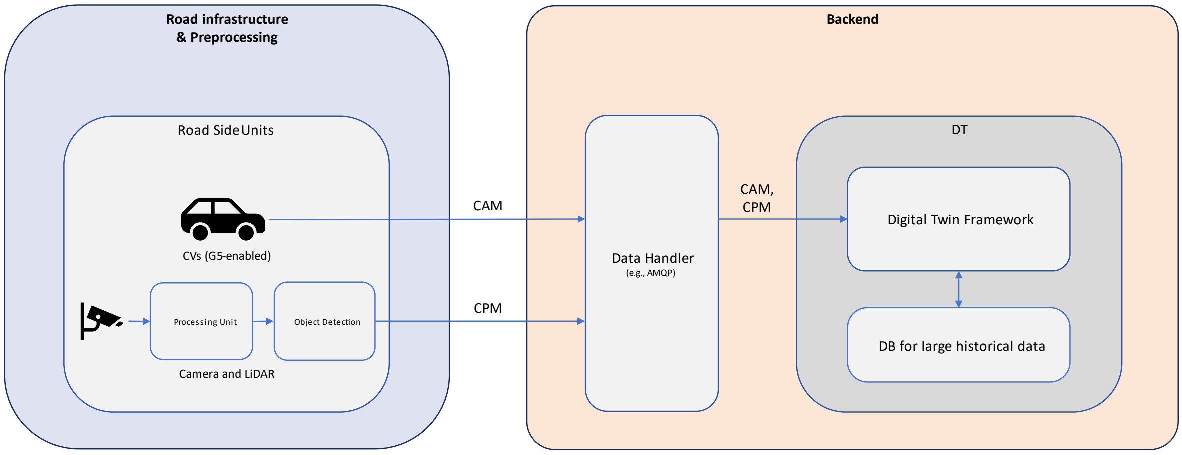

Figure 21: Data Flow from Road Side Units to the Digital Twin

Road Side Units collect data from the real world. All the Cooperative Awareness Messages (CAMs) generated by G5-enabled vehicles are gathered and forwarded to the Digital Twin via a Data Handler, i.e., an ActiveMQ broker, using the AMQP 1.0 protocol. Similarly, each RSU's camera and LiDAR sensors will feed the DT with information about detected objects in its surroundings, by generating and transmitting Collective Perception Messages (CPMs).

Table 7: Table of insights into Module and Data flow of Module 1 for Turin LL

Data format	Data link	Protocol	From	To	Data purpose
ETSI CAM	ITS-G5	ETSI GN	ALL	RSU	Provide the position of ITS station
ETSI CAM	Cellular	AMQP 1.0	RSU	DT	Provide the position of ITS station (forwarded from G5-vehicles to DT)
ETSI CPM	Cellular	AMQP 1.0	RSU	DT	Provide information about detected obstacles on the road

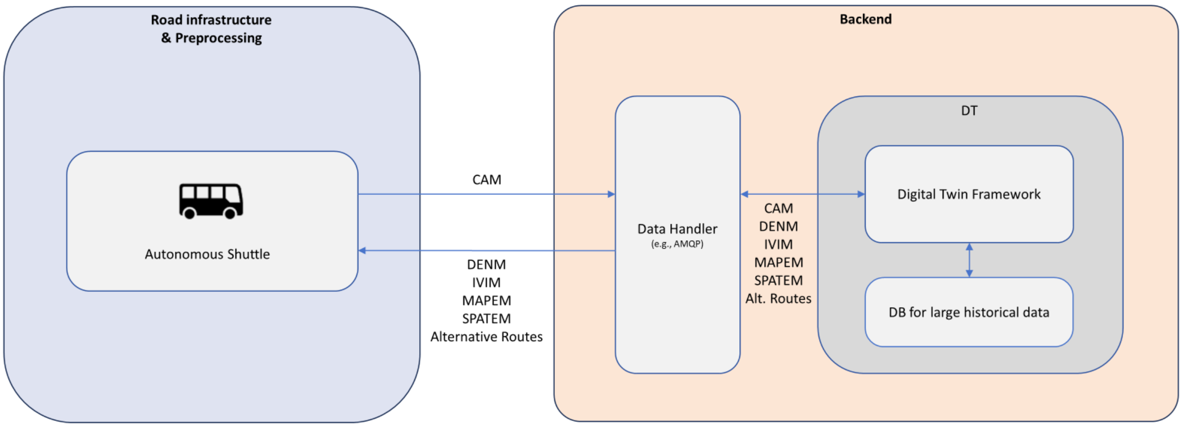

Figure 22: Data Flow between the Autonomous Shuttle and the Digital Twin

In the real world, a CCAV will be generating CAM messages. Such messages will be collected in the Digital Twin, via the Data Handler module. Communication will take place also in a backward direction, enabling the sharing of information (hazardous road events, traffic conditions, road signage and topology, traffic lights statuses, etc.) from the DT to the autonomous shuttle.

Table 8: Table of insights into Module and Data flow of Module 2 for Turin LL

Data format	Data link	Protocol	From	To	Data purpose
ETSI CAM	Cellular	AMQP 1.0	OBU	DT	Provide the position of ITS station
ETSI DENM	Cellular	AMQP 1.0	DT	OBU	Provide information about hazardous locations on the road
ETSI IVIM	Cellular	AMQP 1.0	DT	OBU	Provide various information about road

					signaling and text messages
ETSI SPATEM	Cellular	AMQP 1.0	DT	OBU	Provide the traffic light status of signalised intersections
JSON	Cellular	AMQP 1.0	DT	OBU	Provide information about alternative routes to avoid hazardous events on the road

4.3.3.3 TCC/TIC Communication

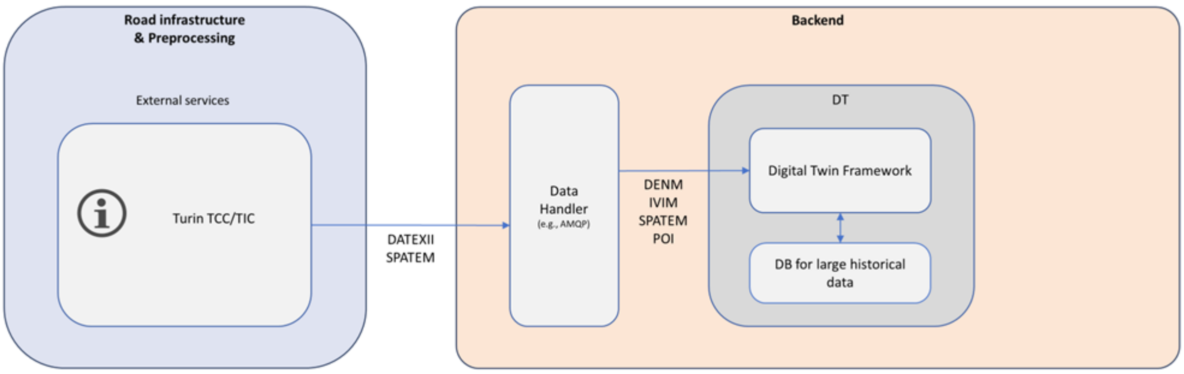

Figure 23: Data Flow from the City's TCC/TIC to the Digital Twin

Data about traffic status in the operating area is collected from Turin's Traffic Information and Control Center and forwarded to the DT. In this case, the Data Handler is also responsible for converting DATEXII messages to ETSI messages, e.g., DATEXII to DENM, before sending them to the DT, for easing the exploitation of information by other C-ITS actors, e.g., vehicles.

Table 9: Table of insights into Module and Data flow of Module 3 for Turin LL

Data format	Data link	Protocol	From	To	Data purpose
DATEXII XML	LAN	AMQP 1.0	TCC/TIC	Data Handler	Provide information about traffic-related events (e.g. incidents, roadworks, …), road signalling (e.g., VMS text), parking lot occupancy, and other data.
ETSI DENM	LAN	AMQP 1.0	Data Handler	DT	Provide information about hazardous locations on the road (converted from DATEXII)
ETSI IVIM	LAN	AMQP 1.0	Data Handler	DT	Provide various information about text messages (UVAR VMS, converted from DATEXII)
ETSI POI	LAN	AMQP 1.0	Data Handler	DT	Provide the parking lot occupancy (converted from DATEXII)
ETSI SPATEM	LAN	AMQP 1.0	TCC/TIC	DT (via Data Handler)	Provide the traffic light status of signalized intersections

4.3.3.4 Simulator + Traffic Generator Communication

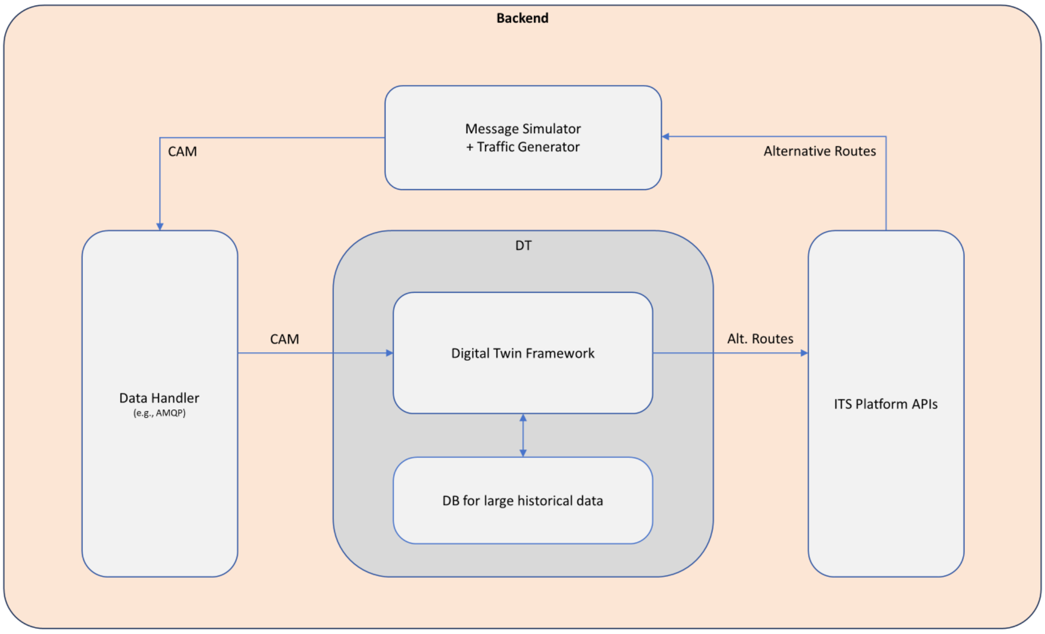

Figure 24: Data Flow between the Message Simulator + Traffic Generator module and the Digital Twin

The Message Simulator module is responsible for simulating connected vehicles, by generating CAM messages. A hazardous event in the monitored area will trigger the DT to ask the Optimization module for alternative routes to avoid traffic congestion near the event position. The results are shared by the DT via its APIs, back to the Traffic Generator module.

Table 10: Table of insights into Module and Data flow of Module 4 Turin LL

Data format	Data link	Protocol	From	To	Data purpose
ETSI CAM	LAN	AMQP 1.0	Message Simulator	DT	Provide the position of simulated ITS stations
JSON	LAN	HTTP	DT	Traffic Generator	Provide vehicles with information about alternative routes to avoid hazardous events on the road

4.3.3.5 Dynamic rerouting

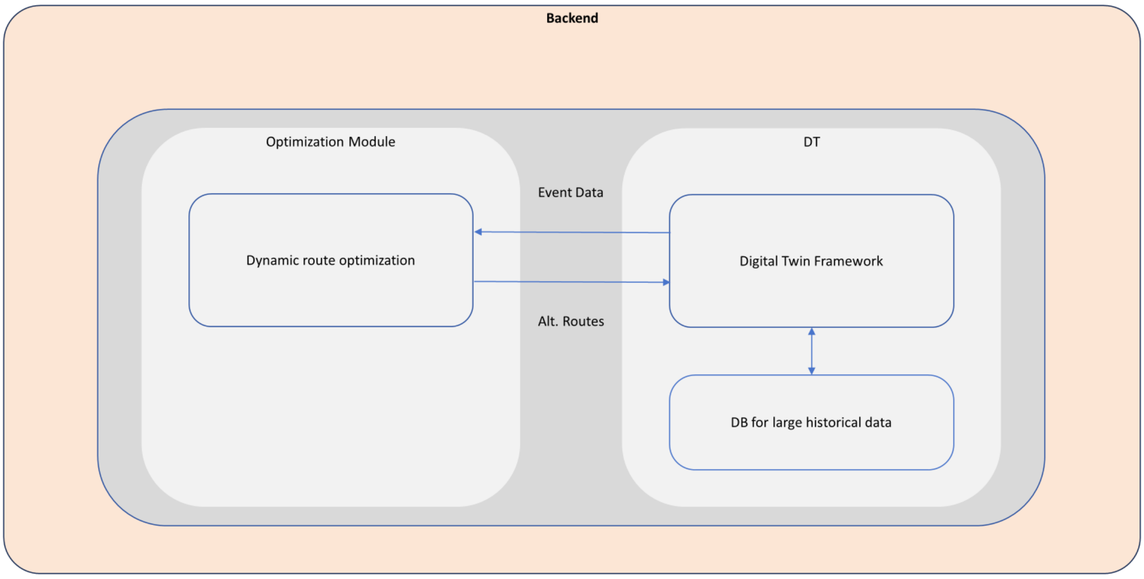

Figure 25: Data Flow between the Optimization Module and the Digital Twin

Following a hazardous incident on the road, the DT will ask the optimization module to compute alternative routes to share with the vehicles in the monitored area.

Table 11: Table of insights into Module and Data flow of Module 5 for Turin LL

Data format	Data link	Protocol	From	To	Data purpose
JSON	LAN	HTTP	DT	Dynamic Route Optimization module	Requests for alternative routes based on hazardous event information and vehicles' location
JSON	LAN	HTTP	Dynamic Route Optimization module	DT	Provides alternative routes to avoid the hazardous event, for each vehicle

4.4 Vigo LL

The Vigo LL objective is to enhance mutual awareness between the CCAM fleet, infrastructure and other users by ensuring a smooth and secure data exchange among them.

This connectivity enables the use of existing C-ITS services and their adaptation to the CCAV fleet (e. g. GLOSA, Road-works warning, Road hazard warning, In-Vehicle information). Furthermore, it allows the deployment of management strategies adapted to CCAM fleets based on V2I interaction. This includes traffic light priority, green phase for platooning and dynamic re-routing.

4.4.1 *Generalized vs LL-Specific Architecture*

The following figure shows the specific architecture put in place in Vigo LL.

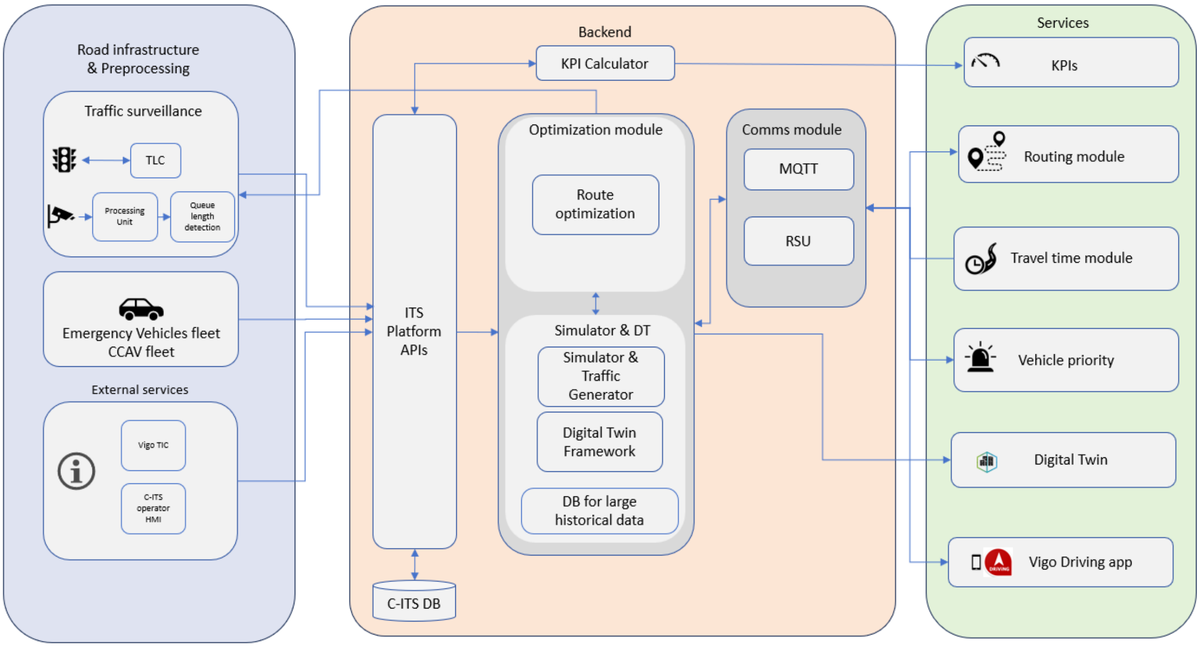

Figure 26: Architecture Vigo LL

4.4.2 Overview of the Various Modules

4.4.2.1 Data sources

Vigo TLC

Vigo Traffic Light Controllers are operating the signalized intersections. The TLC is connected to the local network and to the service provider platform. It exposes an API allowing an external application to fetch data about the real time traffic light phases and to request priority (e.g.: virtual button push).

Vigo TIC

Vigo TIC is the traffic management system in Vigo. It operates parking lots, variable message signs, Bluetooth sensors, cameras, etc. This module exposes several APIs allowing an external application to get data about field equipment (e.g.: variable message signs display information, Bluetooth sensors detection and trip time, parking lot occupancy).

4.4.2.2 Service providers

C-ITS SPATEM

C-ITS SPATEM is a software module that gets real time phase information from the traffic light controllers and creates ETSI-Standardized SPATEM messages from it. It makes SPATEM messages available in APIs to be used by other components of the architecture (C-ITS MQTT publication, RSU).

Additionally, it connects to cameras installed on the field to get queue length information. Whenever a queue is detected at a signalized intersection equipped with a camera, the C-ITS SPATEM module adds this information to the SPATEM.

C-ITS PRIORITY

C-ITS Priority is a software module that provides traffic signal priority for designated vehicles. It is connecting to the RSU API to listen for incoming priority requests (i.e.: SREM messages). When relevant, it triggers a priority demand on the TLC by pushing a virtual button (hurry call). It also uses the RSU API to answer priority requests (i.e.: SSEM messages).

C-ITS DB

C-ITS DB is a module that transforms data sources input into C-ITS compliant messages (i.e.: DENM, IVIM, POIM). Then it stores those messages and manages the lifecycle of C-

ITS messages (i.e.: checks expiration, repetition). It exposes an API for other components to use C-ITS messages (C-ITS MQTT publication, C-ITS HMI).
C-ITS information is also stored in an historical database.

C-ITS HMI

C-ITS HMI is a web platform used by traffic management operators. It has multiple functions:

- Configuration: It allows the configuration of RSU, OBU, MAPEM intersections, Priority routes.
- Maintenance: It allows remote access to RSU, OBU if a problem occurs. It also collects logs from deployed devices.
- Supervision: It shows the status of field equipment.
- Operation: It exposes several HMIs that are used by road operators to provide C-ITS services: creation of manual DENM, IVIM, POIM messages, traffic status visualization, priority requests historic log.

Trip-Time module

The trip-time module is gathering information from the Vigo TIC, from RSUs and from the communication provider and computes trip time on traffic sections in the city. It has multiple sources:

- Bluetooth sensors (Vigo TIC)
- Cellular application fleet (through MQTT broker CAM messages)
- V2X fleet (through RSU API)
- Trip time data is made available through APIs to several modules:
  - Routing module, to compute the best route based on the traffic.
  - C-ITS DB for traffic jam detection.

Routing module

The routing module is connected to the trip-time module to get an overview of traffic sections and their status in real time. Based on this and upon request coming in from the MQTT broker, it will compute an optimal route on the known traffic sections and return it to the requesting vehicle.

Log DB

The log DB is storing logs from all software modules for KPI computation. Log gathering can be automated based on the collecting system. Logs are gathered from:

- The RSU / OBU
- The routing module
- The priority module

4.4.2.3 Field equipment

Cameras

Cameras are installed at signalized intersections and are watching the approach lanes to measure queue length. Cameras have an embedded software module that calculates the queue length on each lane and exposes an API for external applications to get this information.
This API is used by C-ITS SPATEM.

RSU

The RSU acts as a V2X router between the infrastructure and the vehicles. It broadcasts C-ITS messages from the C-ITS DB for equipped vehicles. It also receives messages from the V2X fleet and makes them available for all service provider modules.

OBU

The OBU acts as a V2X router for the vehicles. It connects vehicles to the infrastructure by exchanging messages with RSUs. It receives downwards information about hazardous locations, traffic lights, virtual message signs, parking availability. This information is made available to vehicular applications (on-board HMI, AD stack) through an API.
On the other hand, it sends information gathered from vehicle sensors to the infrastructure (i.e.: CAM, CPM).

Vigo Driving APP

The Vigo Driving APP is an end-user application available on Android and iOS platforms for the public in Vigo. It provides access to C-ITS services in cellular. It also collects user position through CAM messages used to compute travel time in the city.

4.4.2.4 Communication providers

C-ITS MQTT publication

The C-ITS MQTT publication module is publishing C-ITS messages on the MQTT broker for end-user applications to access the C-ITS information in cellular mode based on their location. The location filter is made using a map tile system where messages are published only where relevant. This module oversees choosing the tiles and managing the publication and removal of deprecated messages.

MQTT broker

The MQTT broker is a standard message exchange utility. It accepts connections from the outside and routing messages based on the client location. The MQTT broker contains an access management plugin that allows the service provider to restrict access to some functionalities under certain conditions.

4.4.3 Insights into Modules and Data Flow

4.4.3.1 V2X communication

V2X communication is a pre-condition for all scenarios and the base of UC#4 Scenario#1. The following figure and table show the nature of V2X exchanges.

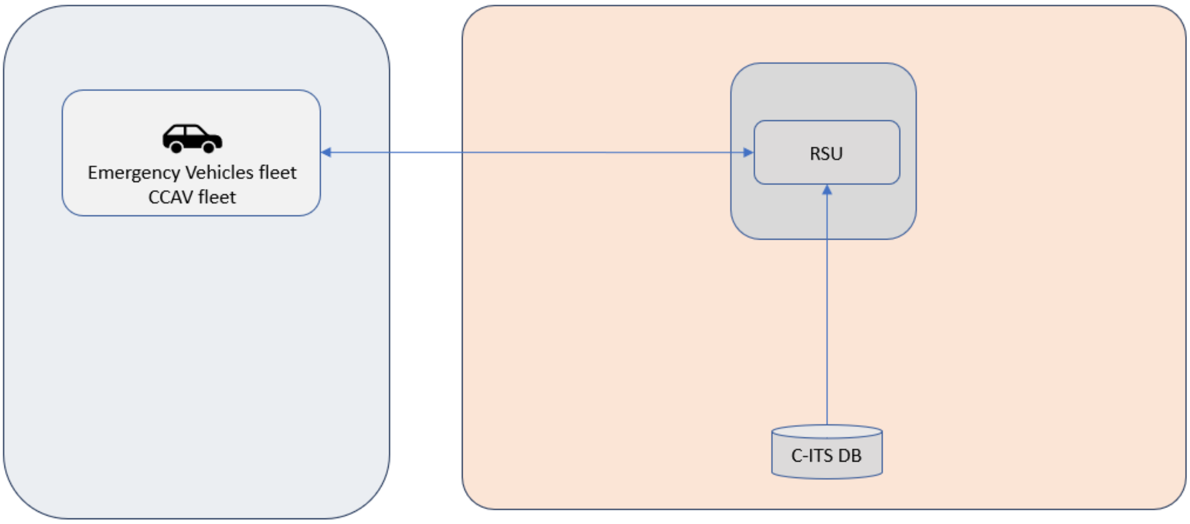

Figure 27: Data flow for V2X Communication

Table 12: Table of insights into Module and Data flow of Module 1 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
ETSI CAM	ITS-G5	ETSI GN	ALL	ALL	Provide the position of ITS station
ETSI DENM	ITS-G5	ETSI GN	RSU	OBU EV	Provide information about hazardous locations on the road

				OBU CCAV	
ETSI IVIM	ITS-G5	ETSI GN	RSU	OBU EV OBU CCAV	Provide various information about road signaling and text messages
ETSI MAPEM	ITS-G5	ETSI GN	RSU	OBU EV OBU CCAV	Provide the geometry of signalised intersections
ETSI SPATEM	ITS-G5	ETSI GN	RSU	OBU EV OBU CCAV	Provide the traffic light status of signalised intersections
ETSI SREM	ITS-G5	ETSI GN	OBU EV OBU CCAV	RSU	Ask for traffic signal priority on a specific signalised intersection approach
ETSI SSEM	ITS-G5	ETSI GN	RSU	OBU EV OBU CCAV	Answer traffic signal priority requests
ETSI CPM	ITS-G5	ETSI GN	OBU CCAV	RSU	Provide information about detected obstacles on the road
Binary	LAN	TCP	C-ITS DB	RSU	Provides ETSI messages to be broadcasted (DENM, IVIM, POIM, MAPEM)
Binary	LAN	TCP	C-ITS SPATEM	RSU	Provides ETSI messages to be broadcasted (SPATEM)

JSON	LAN	TCP	RSU	C-ITS DB	Forwards received ETSI messages to the infrastructure (CAM, CPM)

4.4.3.2 Vigo Driving cellular application

Vigo Driving application is a cellular application for C-ITS use case. It is part of the data interchange UC#4 Scenario#1. The following picture and tables show how the application receives information from the infrastructure and provides information to it.

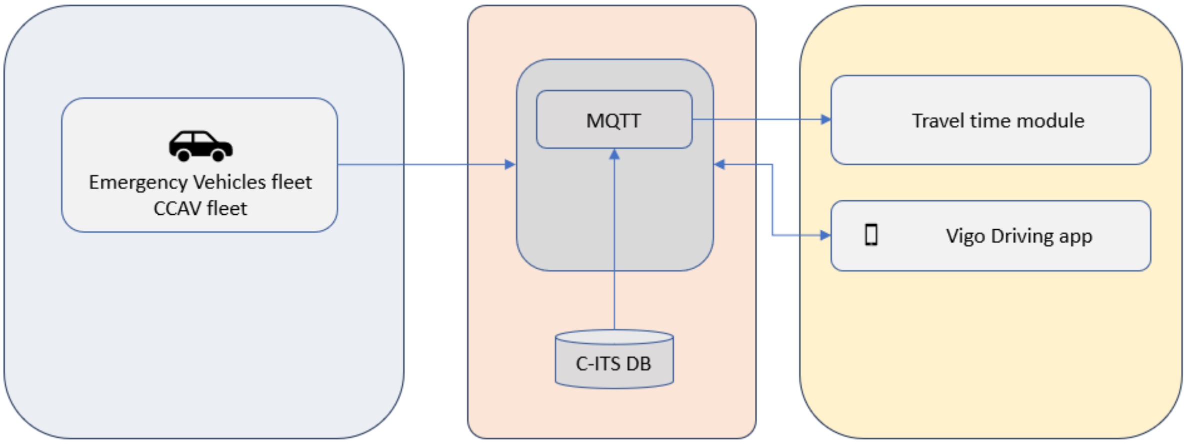

Figure 28: Data flow for Vigo Driving cellular Application

Table 13: Table of insights into Module and Data flow of Module 2 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
ETSI CAM	Cellular	MQTT	Vigo Driving APP	Trip-Time module	Provide the position of ITS station to compute travel time
ETSI DENM	Cellular	MQTT	C-ITS DB	Vigo Driving APP	Provide information about hazardous locations on the road

ETSI IVIM	Cellular	MQTT	C-ITS DB	Vigo Driving APP	Provide various information about road signaling and text messages
ETSI MAPEM	Cellular	MQTT	C-ITS DB	Vigo Driving APP	Provide the geometry of signalised intersections
ETSI SPATEM	Cellular	MQTT	C-ITS SPATEM	Vigo Driving APP	Provide the traffic light status of signalised intersections
ETSI POI	Cellular	MQTT	C-ITS DB	Vigo Driving APP	Provide the parking lot occupancy
ETSI CAM	Cellular	MQTT	OBU EV	Vigo Driving APP	Provide the position of emergency vehicles to the Vigo Driving APP to trigger a warning when an emergency vehicle is close

4.4.3.3 Traffic-based routing

Traffic-based routing is implemented to route CCAVs and enable road network load-balancing in UC#5 Scenario#4. The following picture and table show how vehicles can receive routing information from the infrastructure.

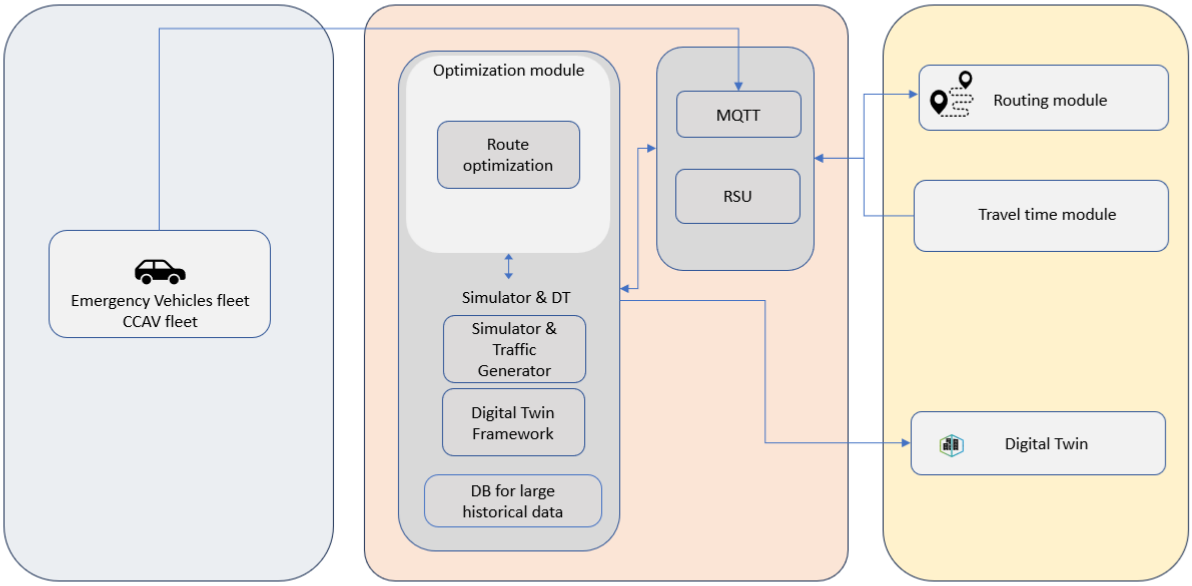

Figure 29: Data flow for Traffic-based Routing

Table 14: Table of insights into Module and Data flow of Module 3 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
JSON	LAN	HTTP	Trip-Time module	Routing module	Provides the current traffic status on the road network
JSON	Cellular	MQTT	OBU CCAV	Routing module	Requests for an itinerary based on an origin location and a destination location
JSON	Cellular	MQTT	Routing module	OBU CCAV	Provides an itinerary based on the current traffic and requested parameters
JSON	LAN	FTP	DT simulation	Routing module	Provides an estimation of the traffic based on the simulation model using historical traffic data. The

					routing module uses this information

4.4.3.4 Traffic signal priority

Traffic signal priority is implemented within UC#5 Scenario#1 and Scenario#2. The following picture and table show the data flow put in place in this context.

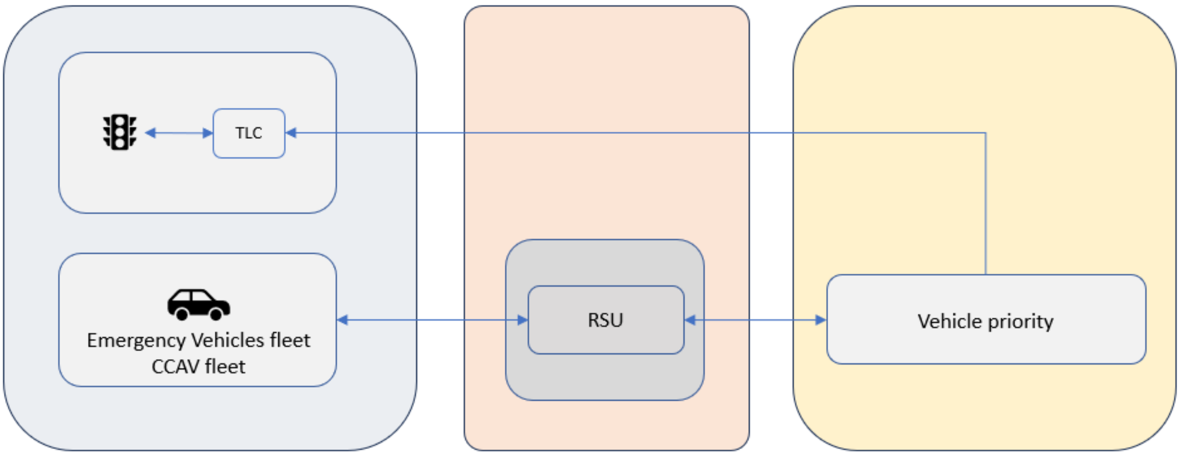

Figure 30: Data flow for Traffic signal priority

Table 15: Table of insights into Module and Data flow of Module 4 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
JSON	LAN	TCP	RSU	C-ITS Priority	Forwards received V2X messages for priority (CAM, SREM)
JSON	LAN	TCP	C-ITS Priority	RSU	Triggers V2X broadcast of V2X messages for priority (SSEM)
Binary	LAN	TCP	C-ITS Priority	Vigo TLC	Requests priority on an intersection using a virtual button push

4.4.3.5 Traffic light status

Traffic light status is used within UC#4 Scenario#1 as part of the data interchange between infrastructure and vehicles. The following picture and table show the information data flow.

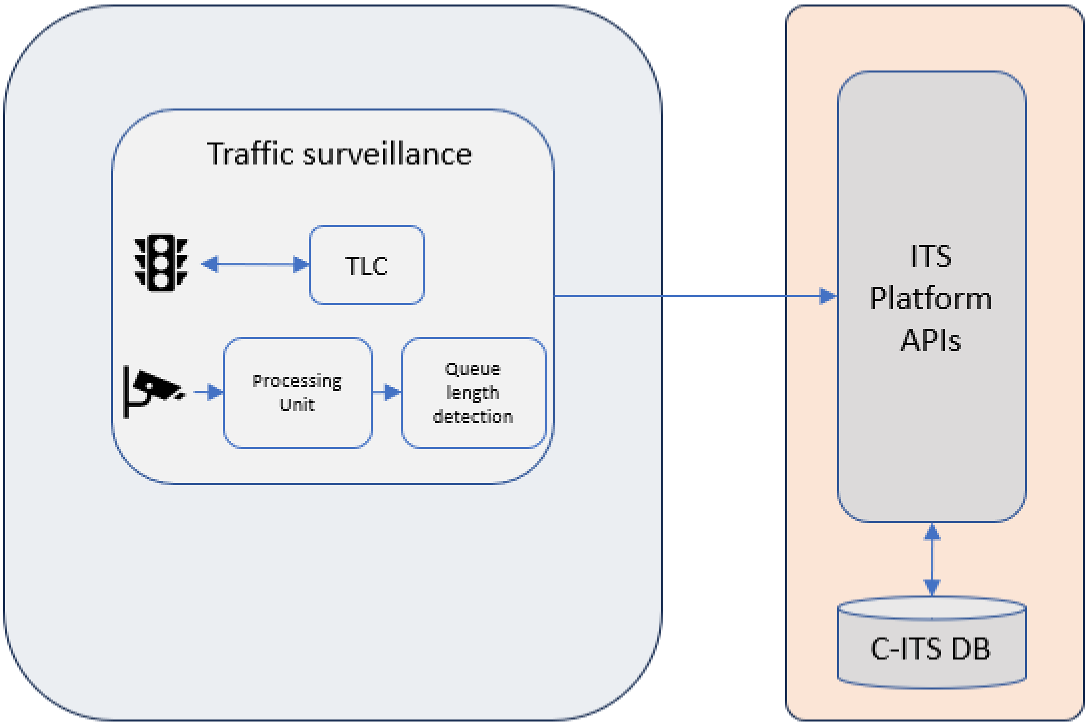

Figure 31: Data flow for Traffic light status

Table 16: Table of insights into Module and Data flow of Module 5 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
XML	LAN	HTTP	Camera	C-ITS SPATEM	Provides queue length information
JSON	LAN	TCP	C-ITS Priority	RSU	Triggers V2X broadcast of V2X messages for priority (SSEM)
Binary	LAN	TCP	C-ITS Priority	Vigo TLC	Requests priority on an intersection using a virtual button push

4.4.3.6 Travel time computation

Travel time computation is achieved thanks to data interchange and an additional data flow coming from Vigo OpenData. Travel time computation is useful in the context of UC#5 Scenario#4 to compute routing information based on traffic. The following picture and table show the data flow for this functionality.

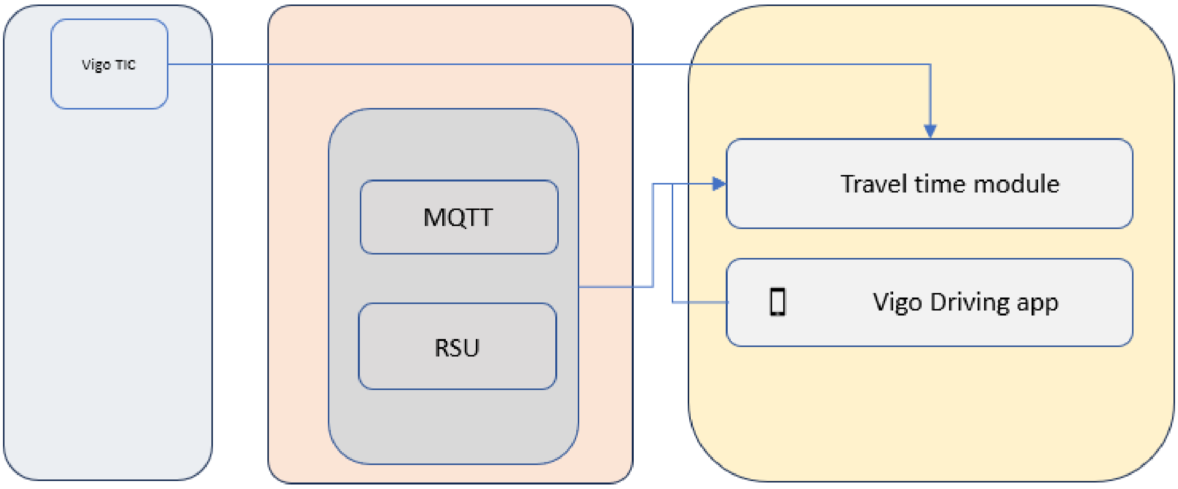

Figure 32: Data flow for Travel Time Computation

Table 17: Table of insights into Module and Data flow of Module 6 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
JSON	LAN	HTTP	Vigo TIC	Trip Time Module	Provides travel time information gathered from bluetooth sensors.
JSON	LAN	TCP	RSU	Trip Time Module	Provides location of V2X vehicles to compute V2X travel time.
ETSI CAM	LAN	TCP	Vigo Driving App	Trip Time Module	Provides location of cellular APP users to compute cellular APP travel time.

4.4.3.7 C-ITS database providers

Before transmitting C-ITS messages into the architecture, C-ITS information has to be created within the platform. To do so, several data flows are necessary and described below.

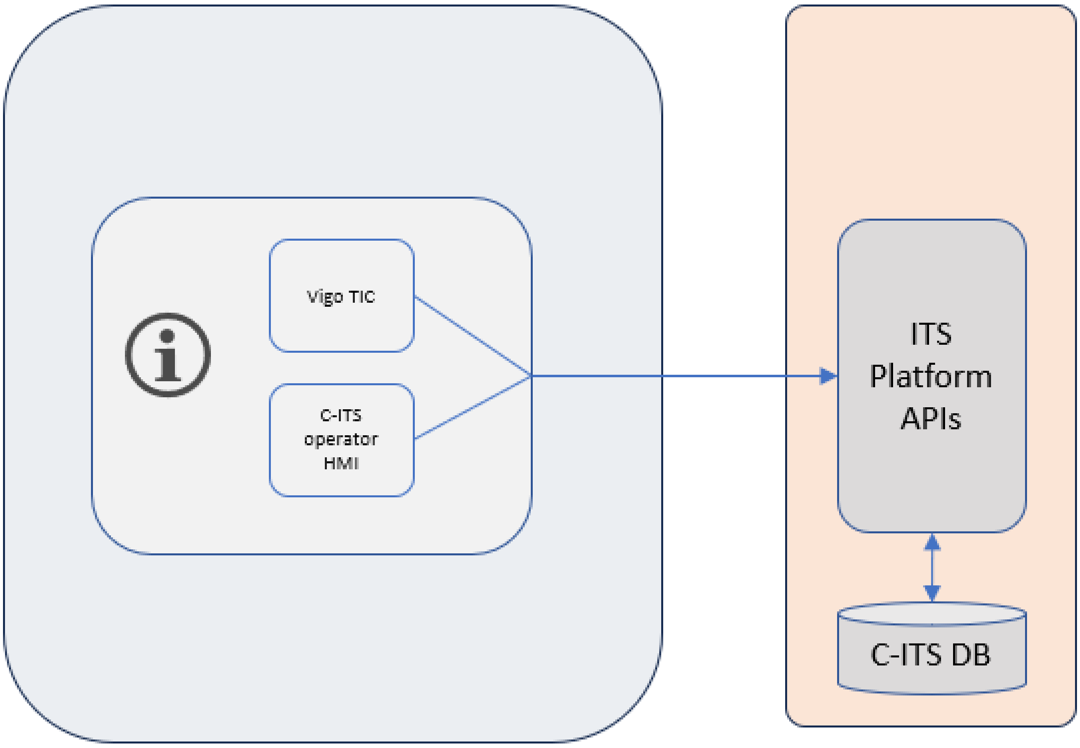

Figure 33: Data flow for C-ITS Database providers

Table 18: Table of insights into Module and Data flow of Module 7 Vigo LL

Data format	Data link	Protocol	From	To	Data purpose
JSON	LAN	HTTP	Vigo TIC	C-ITS DB	Provides parking occupancy
JSON	LAN	HTTP	Vigo TIC	C-ITS DB	Provides display information on the variable message signs in the city
JSON	LAN	HTTP	C-ITS HMI	C-ITS DB	The C-ITS HMI is used to create MAPEM intersections manually.

					MAPEM are transmitted to the C-ITS DB via HTTP.
JSON	LAN	TCP	C-ITS HMI	C-ITS DB	The C-ITS HMI can be used to create manual events (DENM or IVIM) transmitted in the architecture.

5 CONCLUSION

The discrepancies between the Living Labs and their varying stages of implementation presented a challenge in the creation of a general architecture.

The general architecture has been created and each of the modules described. The general functional view is the basis for the implementation in each of the Living Labs. The Living Labs take the modules that are relevant for their implementations in order to derive an architecture that can be applied on a local level. This approach facilitates the understanding of all the implementations for a potential transfer of services or other parts of local architectures.

The Tampere Living Lab successfully completed the concept phase for enhancing local last mile mobility for Hervanta residents and has finalized the architectural design to support these improvements.

In Trikala, the architectural design for an autonomous service has been fully developed. This architecture is designed to integrate various components seamlessly, enhancing urban mobility and safety effectively.

For Turin, the project's architectural phase has outlined two "dynamic re-routing" scenarios for CVs and CAVs, with both designs now complete.

In Vigo, the conceptual and architectural design phases have established a system to enhance mutual awareness among the CCAM fleet, infrastructure, and other road users. All of which have been designed and are ready for implementation.

It is currently not planned to transfer services during the project. However, all services follow set standards, and this deliverable can be an aid to understand the other implementations to facilitate interoperability or a transfer of services in the future. Furthermore, facilitating the gathering of KPIs or possible optimizations of services can also be beneficial.

The delivery was successful, and the new circumstances were successfully adapted. All Living Labs employ the same language from the general architecture when communicating their services and describing their architecture and the relevant data flows for the scenarios. This creates consistency and transparency among the cities. Furthermore, all Living Labs have undergone further refinement to update and finalise their architectures in many rounds.