

**Jialiang Chen**

**Development of an ITS Platform for C-ITS services that use V2X  
communications**

**TREBALL DE FI DE GRAU**

**dirigit pel Dr. Marc Sánchez Artigas**

**Grau d'Enginyeria Informàtica**



**UNIVERSITAT ROVIRA I VIRGILI**

**Tarragona**

**2024**



## **Abstract**

This bachelor's thesis describes the research and development of Idiada's Cooperative Intelligent Transportation Systems (C-ITS<sup>i</sup>) platform, consisting of a set of applications and services that enables advanced inter-vehicular communications.

The C-ITS platform uses Vehicle-to-Everything (V2X<sup>ii</sup>) technology, a wireless communication that enables vehicles to interact with other entities that affect or are affected by them, including other vehicles, road infrastructure and emergency services. Specifically, the platform focuses on Cellular V2X (C-V2X<sup>iii</sup>), an implementation of V2X that uses 4G and 5G cellular technology.

The thesis includes, alongside the research, a practical application, where I follow a professional methodology and help Idiada develop their C-ITS platform that employs C-V2X communications. The platform is composed of multiple applications structured as microservices. These microservices interact with each other through REST<sup>iv</sup> API<sup>v</sup> calls and MQTT<sup>vi</sup> messages, a messaging protocol designed for use cases where the bandwidth is limited, like IoT<sup>vii</sup> or moving vehicles.

All these microservices are designed using hexagonal architecture, which effectively decouples core logic from specific implementations for the platform, enhancing the platform's maintainability and scalability.

## Resum

Aquest treball de fi de grau descriu la recerca i el desenvolupament de la Plataforma Cooperative Intelligent Transportation Systems (C-ITS) d'Idiada, dissenyada per facilitar comunicacions avançades de vehicles. L'objectiu principal d'aquesta plataforma és millorar la coordinació dels vehicles, creant un flux de trànsit més segur i eficient.

La plataforma C-ITS utilitza la tecnologia Vehicle-to-Everything (V2X), una comunicació sense fils que permet als vehicles interactuar amb altres entitats que els afecten o són afectades per ells, incloent altres vehicles, infraestructura vial i serveis d'emergència. Específicament, la plataforma es centra en "Cellular V2X" (C-V2X), una implementació de V2X que utilitza la tecnologia 4G i 5G.

La tesi combina la recerca teòrica amb una aplicació pràctica, on segueixo una metodologia de treball professional i ajudo a Idiada a desenvolupar la seva plataforma C-ITS que utilitza comunicacions C-V2X i 5G. La plataforma està composta per múltiples aplicacions estructurades com a micro-serveis. Aquestes interactuen a través de peticions API REST i missatges MQTT, un protocol de missatgeria dissenyat per aplicacions d'IoT on l'ample de banda de la xarxa és limitada, situació similar a la dels vehicles en moviment.

Tots aquests micro-serveis estan dissenyats utilitzant l'arquitectura hexagonal, la qual desacobla la lògica central de les implementacions específiques per a la plataforma, millorant la facilitat de manteniment i l'escalabilitat de la plataforma.

## Resumen

Este trabajo de fin de grado describe la investigación y el desarrollo de la Plataforma Cooperative Intelligent Transportation Systems (C-ITS) de Idiada, diseñada para facilitar comunicaciones avanzadas de vehículos. El objetivo principal de esta plataforma es mejorar la coordinación de los vehículos, creando un flujo de tráfico más seguro y eficiente.

La plataforma C-ITS utiliza la tecnología Vehicle-to-Everything (V2X), una comunicación sin cables que permite a los vehículos interactuar con otras entidades que les afectan o son afectadas por ellos, incluyendo otros vehículos, infraestructura vial y servicios de emergencia. Específicamente, la plataforma se centra en "Cellular V2X" (C-V2X), una implementación de V2X que utiliza la tecnología 4G y 5G.

La tesis combina la investigación teórica con una aplicación práctica, donde sigo una metodología de trabajo profesional y ayudo a Idiada a desarrollar su plataforma C-ITS que utiliza comunicaciones C-V2X y 5G. La plataforma está compuesta por múltiples aplicaciones estructuradas como microservicios. Estas interactúan a través de peticiones API REST y mensajes MQTT, un protocolo de mensajería diseñado para aplicaciones de IoT donde el ancho de banda de la red es limitado, similar a la situación de los coches en movimiento.

Todos estos microservicios están diseñados utilizando la arquitectura hexagonal, la cual desacopla la lógica central de las implementaciones específicas para la plataforma, mejorando la mantenibilidad y la escalabilidad de la plataforma.

## Index

1. Figures.....	3
2. Declaration of Confidentiality.....	4
3. Introduction.....	5
3.1 Motivation.....	5
3.2 Objectives.....	6
4. Analysis of ITS Standards.....	7
4.1 Intelligent Transportation Systems.....	7
4.2 ITS and Smart Cities.....	8
4.2 Different uses of ITS.....	8
4.2 Cooperative Intelligent Transportation Systems.....	12
4.2.1 Definition.....	12
4.2.2 Automated Vehicles and why we need C-ITS.....	12
4.2.3 Deployment Phases.....	14
4.2.4 C-ITS Components.....	15
4.2.5 Communication Technologies.....	17
4.2.6 Communication Safety.....	23
4.2.7 ETSI, SAE and ASN.1.....	23
4.2.8 ITS C-V2X Messages.....	25
4.2.9 GeoNetworking protocol.....	27
5. Specification of Idiada's ITS Platform.....	28
5.1 ITS Platform.....	28
5.2 Idiada's C-ITS Platform.....	29
5.3 Technologies Used.....	29
5.3.1 Tiles and Map Matching.....	29
5.3.2 MQTT.....	30
5.4 Requirements.....	30
5.4.1 Functional Requirements.....	30
5.4.2 Non-Functional Requirements.....	31
5.5 Use Cases.....	33
5.5.1 Provider.....	33
5.5.2 External User.....	33
5.5.3 Authorized Clients.....	33
6. Design.....	35
6. Development of the C-ITS platform.....	36

6.1 Modify Extractor to be able to subscribe to multiple Message Types in a single instance. ....	38
6.2 Perform a load test on the Extractor service to investigate whether more threads should be used. ....	38
6.3 Fix JWT Bug .....	38
6.4 Update All services that depend on a library. ....	39
6.5 Create a PostMan Collection for Message Manager API .....	42
6.6 Remove JSON implementation in the Message Manager.....	42
6.7 Update All Applications to Spring 3.....	42
6.8 Investigate the ETSI standard to check if Idiada’s MAPEM comply with the standard.....	44
6.9 Define a KPI logging required for upstream and downstream in Extractor. ...	45
6.10 Create a Sequence diagram about the C-ITS gets a JWT from the identity manager. ....	45
6.11 Create a new Service (push adapter) for a client.....	45
6.12 Document how C-ITS generates CAM messages and sends the location to GeoMessaging. ....	46
6.13 Modify Logging in all projects. ....	46
7. Conclusions .....	47
8. References .....	48
9. Annex .....	50
10. Keywords .....	51

## 1. Figures

Figure 1: Tesla Auto Pilot (Tesla Motors, 2016) .....	7
Figure 2: Sweden's Swarco Hub Connected Traffic (Swarco, 2024) .....	8
Figure 3: Smart Semaphores (RadioYTVMexiquense, 2020) .....	9
Figure 4: Vehicle ADAS (Synopsys, 2024) .....	10
Figure 5: Audi's GLOSA system (Auto Motor Sport, 2024) .....	10
Figure 6: e-Call system (TodoMecanica, 2018) .....	11
Figure 7: Bike Rental Service that uses ITS (Times, 2016) .....	11
Figure 8: Automated driving levels (Synopsys, 2024) .....	13
Figure 9: Volkswagen's traffic hazard alert function (Volkswagen, 2020) .....	14
Figure 10: Q-free OBU Unit, an extra antenna is also needed (Q-free, 2021) .....	15
Figure 11: Q-free's adaptive RSU system (Q-free, 2021) .....	16
Figure 12: C-ITS Components (Obaid, 2019) .....	16
Figure 13: TMC Use in action (ResearchGate, 2018) .....	17
Figure 14: DSRC's architecture .....	18
Figure 15: The two interfaces of C-V2X: C-V2X and 4G/5G (Castells, 2023) .....	20
Figure 16: C-V2X Data Link Modes (CNSLab, 2020) .....	21
Figure 17: DSRC VS. C-V2X (Castells, 2023) .....	22
Figure 18: C-ITS Messages Summary (Castells, 2023) .....	26
Figure 19: Other C-ITS Messages Summary (Castells, 2023) .....	26
Figure 20: Idiada's Logo (Applus, 2024) .....	28
Figure 21: GeoBroadcast protocol (Castells, 2023) .....	29
Figure 22: Tile Mapping System (Castells, 2023) .....	30
Figure 28: Hexagonal Architecture (Baeldung, 2024) .....	37
Figure 35: KPI Platform .....	45

## 2. Declaration of Confidentiality

This document contains confidential data that is the property of Idiada Automotive Technology SA. As per the terms of the previously signed confidentiality agreement, the information included in this thesis is confidential and is intended solely for the authorized parties. The confidentiality applies to all data and information contained within the sections described below:

**[-- CONFIDENTIAL SECTION --]**

Some confidential data ...

...

**[-- END OF CONFIDENTIAL SECTION --]**

Readers are required to maintain the strictest confidentiality of this information and must not disclose or utilize it except as permitted by the agreement. Unauthorized use or disclosure of this information may lead to consequences specified in the agreement.

## **3. Introduction**

### **3.1 Motivation**

Traffic jams are a common cause of frustration for drivers. Imagine spending 90 hours a year stuck in traffic, the engine idling and your schedule slipping away. This isn't just a nuisance, it's a recurring challenge that affects millions daily. Not only do traffic jams cause a massive loss of time, but the stress and urgency induced by the congestion also lead to other serious problems such as aggressive driving and traffic accidents.

This problem has substantial social and economic implications. Inrix, a world leader in transportation analytics, reported that traffic hotspots across Europe are projected to cost its countries £183.2 billion from 2016 to 2026 (Cookson, 2017); Moreover, the same company indicated that Americans have lost 3.4 billion hours due to congestion in 2021 (Inrix, 2021).

These statistics led to the need for developers and engineers to create innovative solutions to the problem, in this context, the development of ITS has emerged as a promising approach to improve road safety and efficiency. ITS is a broad term that refers to any technology that integrates vehicles, road infrastructure and users, including, but not limited to, automated vehicles, optimal route planners, traffic management systems, connected vehicle technology, electronic toll systems, and emergency vehicle notification systems.

V2X is one of the technologies used to achieve such intelligent platforms, it is mainly used in vehicle communications, allowing vehicles to communicate with other vehicles, deliver real time traffic information, react pre-emptively to road conditions, read road signs, and more.

This document focuses on Idiada's Cooperative ITS implementation, which utilizes C-V2X and 5G technologies for communication. This platform allows vehicles to send and receive information about their surroundings and other entities, effectively increasing traffic efficiency and enabling better vehicle coordination. This not only reduces traffic and pollution but also decreases accidents and enhances overall traffic flow.

## 3.2 Objectives

There are two main objectives in this thesis:

First, the thesis aims to learn about the technologies currently available that improves the traffic environment for all users. This includes an in-depth analysis of various Intelligent Transportation Systems (ITS), with a particular focus on those that use V2V<sup>8</sup> and V2X technology. The goal is to not only understand these technologies but also to develop an innovative platform that uses said technology to facilitates smart vehicle communications.

Second, the objective extends to the practical application of the knowledge and skills acquired during my bachelor's degree. This involves applying theoretical knowledge in a professional setting, learning new development skills, tools, and programming languages while sticking to a professional workflow. To achieve this, I have joined Idiada's Connected Vehicles team with an internship, where I will carry out the research about C-ITS, V2X and other aspects of this sector, then form part of the process of developing Idiada's C-ITS platform.

By meeting these objectives, the thesis will bridge the gap between academic study and practical, professional application, contributing to the field of intelligent transportation with a focus on improving traffic management and vehicle communication technologies.

## 4. Analysis of ITS Standards

### 4.1 Intelligent Transportation Systems

Intelligent Transportation Systems, or ITS in short, are advanced platforms that provide useful services that relate to different transportation modes and traffic. Their main purpose is to help users make a smarter, more efficient and coordinated use of transportation.

These advanced systems are being adopted worldwide to create more efficient traffic flows, save on fuel consumption, gather information about road users and reduce pollutants, journey times, and number of accidents.

Although it may sound advanced, ITS isn't a new concept, its formal establishment started in the 1990's, when the United States' Department of Transportation created the Intelligent Transportation Systems Joint Program Office (ITS JPO<sup>9</sup>), followed by a significant push towards the research and deployment of ITS technologies in larger scales.

In the early 2000's, advancements in GPS<sup>10</sup> and GIS<sup>11</sup> allowed more advanced ITS applications, which allowed real time location sharing. In more recent years, the introduction of connected and automated vehicles (CAVs) led to a shift in the focus of ITS developments, leaning more towards the automation and connectivity aspect of vehicles.

Tesla is one of the first companies that implemented such innovations, in 2014, it introduced Tesla Autopilot, a semi-autonomous driving system that was part of the company's Model S series. This system uses an array of cameras, radar, and ultrasonic sensors to provide driver-assistance features such as lane keeping, self-parking and traffic cruise control.

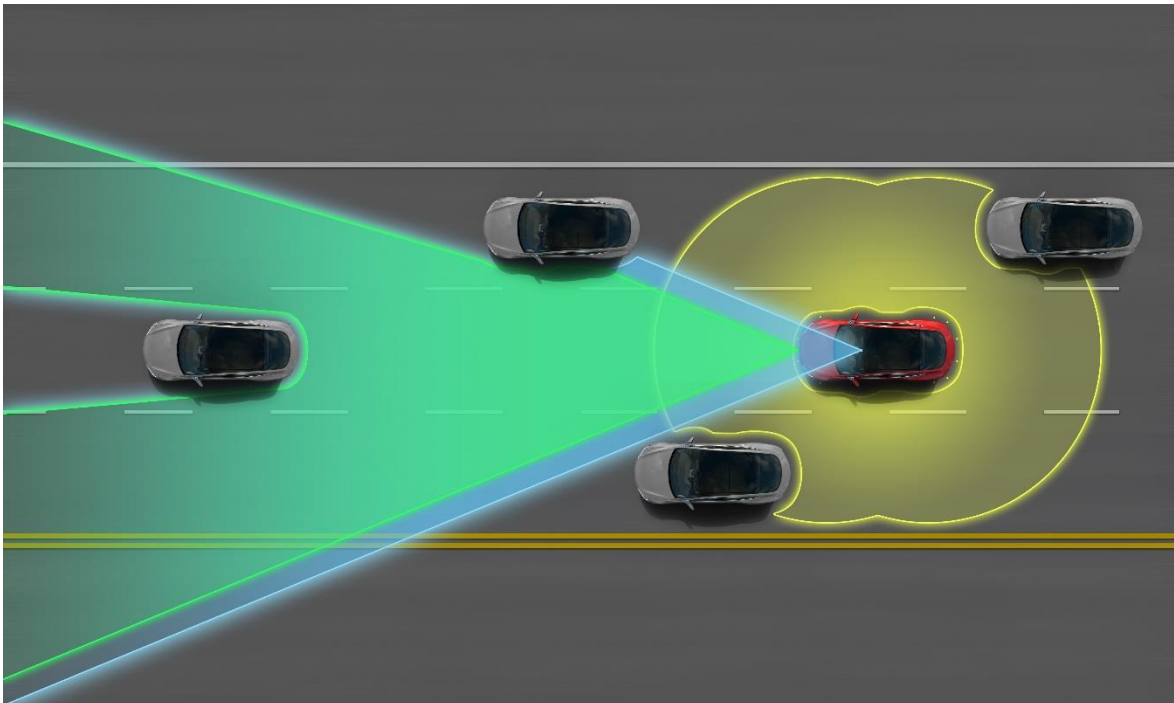


Figure 1: Tesla Auto Pilot (Tesla Motors, 2016)

Ever since the introduction of Tesla's Autopilot, alongside the rising popularity of electric vehicles (EVs), and the need of a solution for crowded cities, the sector of Intelligent Transportation Systems has witnessed an unprecedented pace of innovation and investment by various companies.

## 4.2 ITS and Smart Cities

Although the use of ITS is not limited to road transportation, it is its focus in recent years, some big cities across the world have already implemented some form of ITS in their Smart Cities projects, examples include Stockholm, Singapore, Barcelona, and New York City.

By integrating different ITS technologies in the transportation infrastructure, travel times, fuel consumption and vehicle operating costs are reduced drastically, resulting in cost savings for both businesses and governments.

For example, in Stockholm, a smart traffic platform helped cut gridlock<sup>12</sup> by 20 percent, reduce emissions by 12 percent and increase use of public transportation drastically. In London, a congestion management system lowered traffic volume to the mid-1980's levels. (Smartcitiesdive, 2017). These examples show how much of an impact these new platforms can have in our society.



Figure 2: Sweden's Swarco Hub Connected Traffic (Swarco, 2024)

## 4.2 Different uses of ITS

If you live in a city, chances are you've encountered some form of ITS countless times during your commute or trips. One of the most common forms of ITS are speed radars, which monitor the speed of vehicles and impose fines for individuals that are driving

recklessly. Another form of common ITS are smart road signs, these are reversible lane signs, digital signs that alert drivers of heavy rain, ice, accidents ahead, and more. Smart parking sensors that can read your car's plate is also a form of ITS that keeps track of the parking's capacity and number of users. Lastly, if you have used some kind of trip guide software or device, it is also a kind of ITS that provide route planning, such services include Google Maps (in your smart phone) and on-board GPS systems.

More advanced forms of ITS use a combination of smart sensors, radars, and remote cloud services to provide even better solutions for vehicular safety and efficiency, some of these newest technologies include:

- **Traffic signal control systems:** Smart traffic lights that change their state according to the traffic needs and smart signs that can send messages to the vehicles indicating their presence. They use sensors and radars to capture data and process the information offline or reports the status to a central hub which then processes the information and sends the status update.

For example, in a city's ingress lane, a semaphore can be installed so if it detects a speeding vehicle, it changes its state to a red light, forcing the driver to reduce its speed. Once the vehicle is within the speed limit, the light shall return to green.



*Figure 3: Smart Semaphores (RadioYTMexiquense, 2020)*

- **ADAS<sup>13</sup>:** These systems are installed on the vehicle itself and are some of the newest innovations in the latest years, it includes various technologies that help prevent accidents, including:
  - Blind spot detection system.
  - Parking assistant.
  - Rear collision warning system.
  - Emergency braking, pedestrian detection, collision avoidance systems.
  - Lane departure detector.

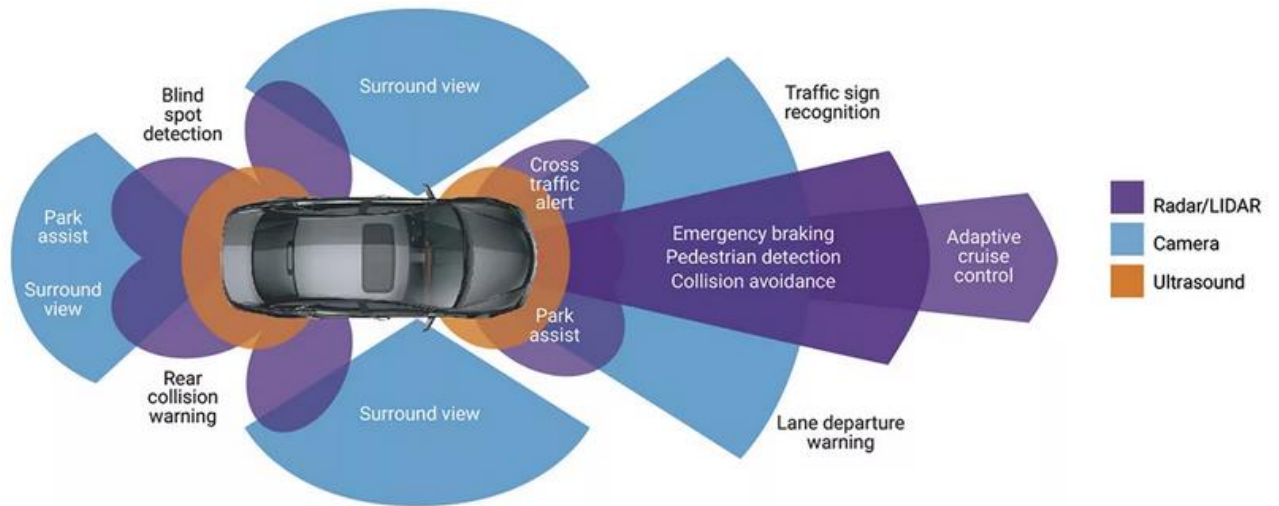


Figure 4: Vehicle ADAS (Synopsis, 2024)

- Connected Vehicles:** Smart vehicles that communicate with each other, providing advanced coordination in difficult traffic conditions. This technology is generally still in production, but some form of it already exists, for example, Audi released several vehicles that has Glosa<sup>14</sup>, a form of communication between semaphores and cars that reduces unnecessary stops and reduces fuel consumption (GLOSA will be explained further in the next chapter).



Figure 5: Audi's GLOSA system (Auto Motor Sport, 2024)

- Emergency Systems:** Technologies like **e-Call**, capable of calling the emergency services and sending the GPS location automatically upon accident detection. In Europe, all European vehicles that homologated after March 31<sup>st</sup>, 2018, **must** include this system.



Figure 6: e-Call system (TodoMecanica, 2018)

- **Commercial platforms:** Some companies that provide bike / scooter rental services use ITS to keep track of their vehicles and their power levels.



Figure 7: Bike Rental Service that uses ITS (Times, 2016)

- **Public transport:** Some cities have smart bus stops that tell the passengers when the public transportation is arriving, this is possible thanks to a tracking ITS platform that calculates the ETA<sup>15</sup> of busses or trains.

## 4.2 Cooperative Intelligent Transportation Systems

### 4.2.1 Definition

While ITS focus on the technology in vehicles and the infrastructure, Cooperative Intelligent Transportation System's (C-ITS) main focus is on the communication aspect **between** those systems, may it be vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P<sup>16</sup>), vehicle-to-infrastructure (V2I<sup>17</sup>) or V2X communications.

C-ITS enables road infrastructure and vehicles equipped with the technology to coordinate manoeuvres, send, or receive important messages, warnings about their surroundings, or a nearby hazard in real time. To accomplish this, different wireless technologies can be used to enable these communications.

The primary benefits of C-ITS are:

- **Reduced number of accidents:** Most accidents happen due to human errors, by enabling coordination between different connected vehicles, we can reduce the amount of human error during the choice making in complex scenarios. Additionally, central hubs can send alerts or advice to vehicles heading towards an accident or hazard, increasing driver precaution.
- **Less traffic jams:** C-ITS enables real time traffic monitoring; The monitoring system can adjust the traffic lights as needed to achieve an efficient traffic flow. Alternatively, messages can be sent from the platform to vehicles approaching to a traffic concentration, suggesting a different route to its destination.
- **Faster incident management:** If accidents do happen, emergency services will be able to act instantaneously, reducing the amount of time emergency services arrive at the scene.
- **Minimized pollutants:** By implementing systems like GLOSA and vehicle coordination, less stops and fast accelerations will be necessary, reducing the amount of pollution vehicles produce.

The use of C-ITS in autonomous vehicles further increases the effectiveness of both systems, by enabling both autonomous driving and cooperation between systems.

### 4.2.2 Automated Vehicles and why we need C-ITS

To better understand the need of C-ITS for automated vehicles, we need to learn what automated vehicles are. An autonomous vehicle (or self-driving vehicle) is one that can sense its environments and operating the vehicle without any driver assistance, that means, no human interaction or presence is needed at all. These vehicles operate by integrating technologies such as sensors, cameras, lidar (light detection and ranging), radar, artificial intelligence, and machine learning to perceive their surroundings and make decisions.

Automated vehicles have 6 levels of automation, each providing different levels of safety:

- 0- **No automation:** Driver is completely responsible for controlling the vehicle, such as steering, braking, accelerating, or parking. Level 0 vehicles can have features like cameras and blind spot warnings, but it is classified as level 0 because it doesn't act consistently over a sustained period.
- 1- **Driver Assistance:** In this level, automated systems can start taking control at specific scenarios, but doesn't take full control of the vehicle, some examples of

this are adaptive cruise control (keep a safe distance with the preceding vehicle) and lane keeping assistance.

- 2- **Partial Automation:** In this stage, the automated vehicle can take over both steering and acceleration / deceleration in certain scenarios (such as highways with good weather conditions). Driver must remain engaged and always pay attention at the environment. Advanced cruise control is an example of partial automation.
- 2- **Level 2+:** Not officially recognized by the SAE<sup>18</sup>, this level groups vehicles that are more advanced than level 2 but doesn't meet the criteria for level 3. These include vehicles that include more sophisticated environmental detection capabilities and better complex scenario handling. In this level, systems are basically driving autonomously, but the driver should always be ready to step in if needed. The main benefit of this level is the more affordable price compared to level 3 vehicles.
- 3- **Conditional Automation:** Drivers can disengage from the act of driving in this level of automation, but only in certain situations where weather and road conditions are optimal. This is considered the entry point to autonomous driving.
- 4- **High Automation:** At this level, vehicles can take full control of all driving tasks in specific conditions in urban areas or dedicated lanes, no driver intervention is required in these scenarios, but outside of these areas, driver assistance may be needed.
- 5- **Full Automation:** The vehicle is fully autonomous and does not require a human driver, it can operate in all kinds of conditions and environments, vehicles at this level might not even have human interface (steering wheel, brakes, etc.).

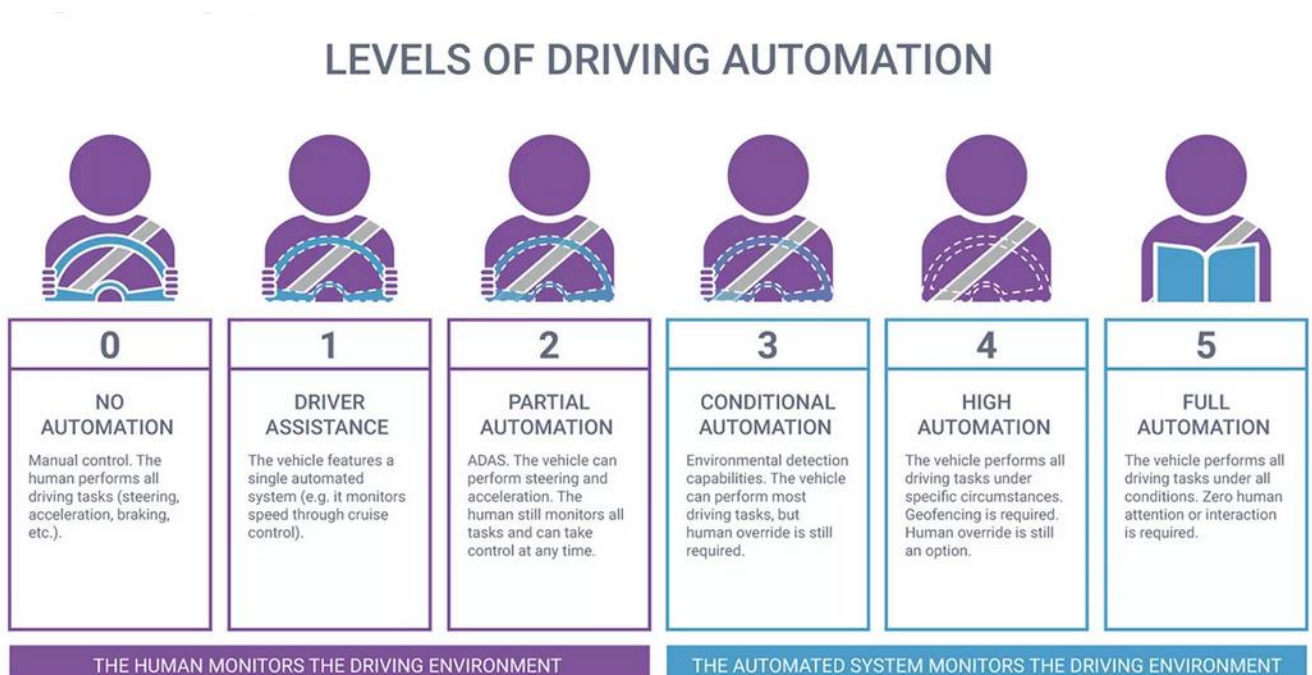


Figure 8: Automated driving levels (Synopsys, 2024)

Given these details about different levels of autonomous vehicles, and the fact that most of the commercial vehicles are equipped with either level 1 or level 2 automation, drivers still

cannot fully disengage with the act of driving, and in complex scenarios, drivers are still fully responsible of the driving.

In this context, automated vehicles are still limited to the information that the array of sensors and cameras that are installed on each vehicle can provide, so it cannot sense further than the immediate proximity and the surroundings of the physical vehicle.

This is where C-ITS comes in, by enabling communication between different automated vehicles and other entities, we are essentially enabling “far sight” for vehicles, so they can “see” beyond the visual line of sight (BVLOS, a term used in aviation systems), this gives us the capability of seeing behind corners, seeing in front of vehicles, anticipate harsh weather conditions, communicate our intentions to manoeuvre to distant vehicles, coordinate the priority of each vehicle, etc. The combination of this automation and the connectivity gives us the definition of CAVs (Connected and Automated Vehicle).

### 4.2.3 Deployment Phases

C-ITS has defined three deployment phases:

1. **Phase one** will be systems that deliver messages to vehicles. For example, if an object or construction is obstructing the road ahead, it can send this warning to all vehicles that are approaching its location. Or smart signs that can send speed limits or other information to vehicles nearby.
2. **Phase two** is mainly characterized by information gathering from vehicles in the field, using sensors and antennae. Using this surrounding information, the system determines if any action should be taken, if so, the system will notify the driver using an alert.
3. **Phase three** is about collaboration between different vehicles, it involves continuous exchange of communication between vehicles and the infrastructure, this allows a better coordination between vehicles and avoid possible accidents.

Currently, different OEMs are investing in C-ITS services, with some of them already implementing it in vehicles (e.g., Volkswagen’s 2020 Golf with Car2X technology, capable of traffic hazard alert function that collects and sends data to other vehicles that have the same technology via WIFI) (Volkswagen, 2020).



Figure 9: Volkswagen's traffic hazard alert function (Volkswagen, 2020)

Another example is the **GLOSA** technology already present in many Audi vehicles; this technology enables traffic signals to send speed advice to nearby vehicles so that they don't need to completely halt and wait for a green light. Instead, they can continue moving at a constant pace and, by the time they reach the stop line, the light will be green.

#### 4.2.4 C-ITS Components

To enable C-ITS in vehicles and infrastructure, we need different components that can connect to each other and communicate through a defined protocol.

There are three main components in the C-ITS system:

- **OBU:** On-Board Unit, a system that is installed in the vehicle that handles all the signal processing and reception / emission of messages between vehicles (V2V, RSU and the central hubs). It collects important vehicle data (speed, direction, and road condition) either through the vehicle's sensors or inside the same OBU unit. Since C-ITS is mostly still in the development phase, the commercial OBU usually come with all the sensors necessary to capture the data mentioned before, but once C-ITS is incorporated in newer vehicle models, it won't require internal sensors. Commercial versions of OBU units are sold by different manufacturers, including Q-free, Cohda, Commsignia, Harman International, etc.



Figure 10: Q-free OBU Unit, an extra antenna is also needed (Q-free, 2021)

- **RSU:** Road-Side Unit, these are different infrastructure units that are C-ITS enabled to send messages to vehicles and statistics to a central hub. Semaphores, signals, cameras, radars are examples of possible RSU infrastructure. Most implementations of RSU systems have an adapter for existing infrastructure, only requiring little modification to the infrastructure already deployed.



Figure 11: Q-free's adaptive RSU system (Q-free, 2021)

- **Central Hub / TMS:** Traffic Management System, this is the central hub where processing of data collected from OBUs and RSUs are carried over. It can analyse the traffic and manage traffic signs, respond to incidents, and send alerts to drivers. It doesn't necessarily need to be a traffic management application, as ITS supports various applications that benefit road users.

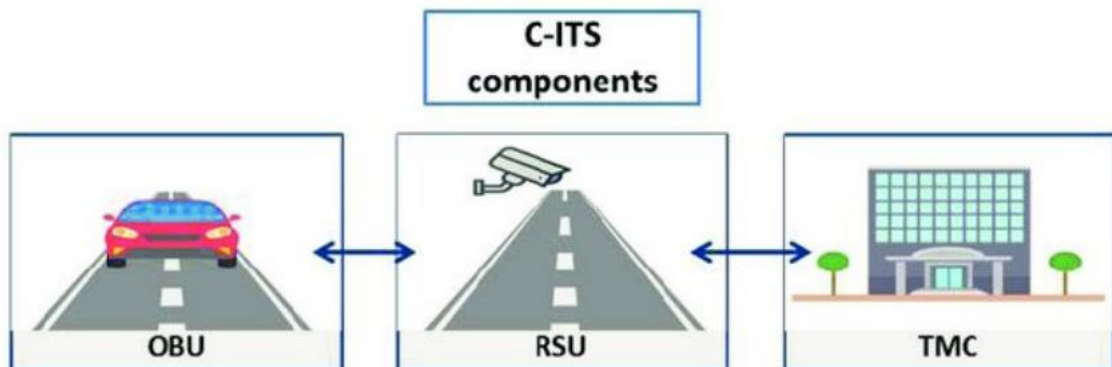


Figure 12: C-ITS Components (Obaid, 2019)

In the figure 13 we can see the TMC coordinating different RSU while individual vehicles send updates / positions to the TMC and other vehicles.

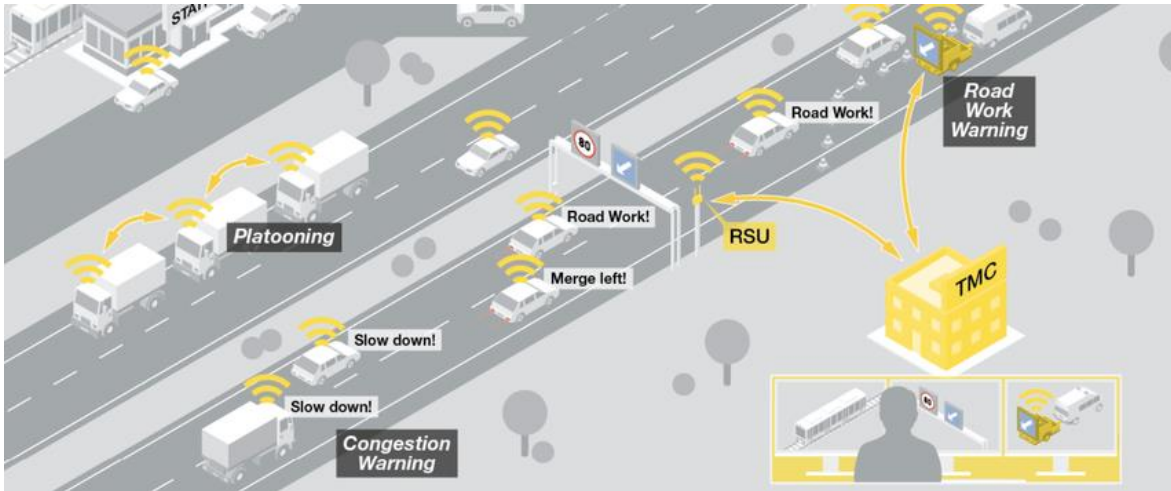


Figure 13: TMC Use in action (ResearchGate, 2018)

#### 4.2.5 Communication Technologies

V2X allows the use of different wireless technologies to send the messages or alerts, each manufacturer has its preferences, the main battle is deciding whether to use C-V2X (WLAN + Cellular) or DSRC<sup>19</sup> (WLAN technology only) for C-ITS deployment, being DSRC the oldest in the ITS sector.

##### 4.2.5.1 DSRC

Dedicated Short-Range Communication, or DSRC, is a communication technology based on 802.11p (WLAN technology) designed for vehicle use in ITS platforms. It was developed by the *Comité Européen de Normalisation* (CEN) in the 1990's to enable high-speed, authenticated vehicle identification for applications such as charging stations (automatic fee collection for using highways, for example) and enforcement of driving hours. DSRC is included in the WAVE<sup>20</sup> suite of standards and is one of the two main technologies to be used in V2X communications.

The equipment required to use DSRC are twofold:

- Road-Side Equipment (**RSE**): This equipment is the “master” of the protocol that is usually installed on the roadway. It controls the activation of the OBE; it writes and reads to the OBE and ensures message validity and correct delivery of messages.
- On-Board Equipment (**OBE**): This equipment is installed in vehicles, and it communicates with the RSE.

The standard operates on the 5.8GHz, 5.9GHz RF<sup>21</sup> and the infrared channel (depending on the region). The range of this communication protocol is about 300-500m, depending on the environment (on open environments it can reach 1000m). The architecture of the standard is based on a reduced communication protocol stack, formed by only 3 OSI<sup>22</sup> layers (due to a constraint of real-time communication requirement):

- **Physical Layer:** Standard EN 12834. This layer is responsible for the transmission and reception of raw bit streams over a physical medium. There are two standard proposals for the DSRC physical media: 5.8GHz (and 5.9GHz) and infrared at 850nm.

- **Data Link Layer:** Standard EN 12795. This layer is divided in two sub-layers, LLC<sup>23</sup> and MAC<sup>24</sup>. Its main function is to ensure transaction completion, data organization, sequence control, flow control, error detection and error recovery.
- **Application Layer:** This layer defines functions and message formats to support ITS services. Any data encryption also happens in this layer.

The architecture of DSRC standard can be seen below:

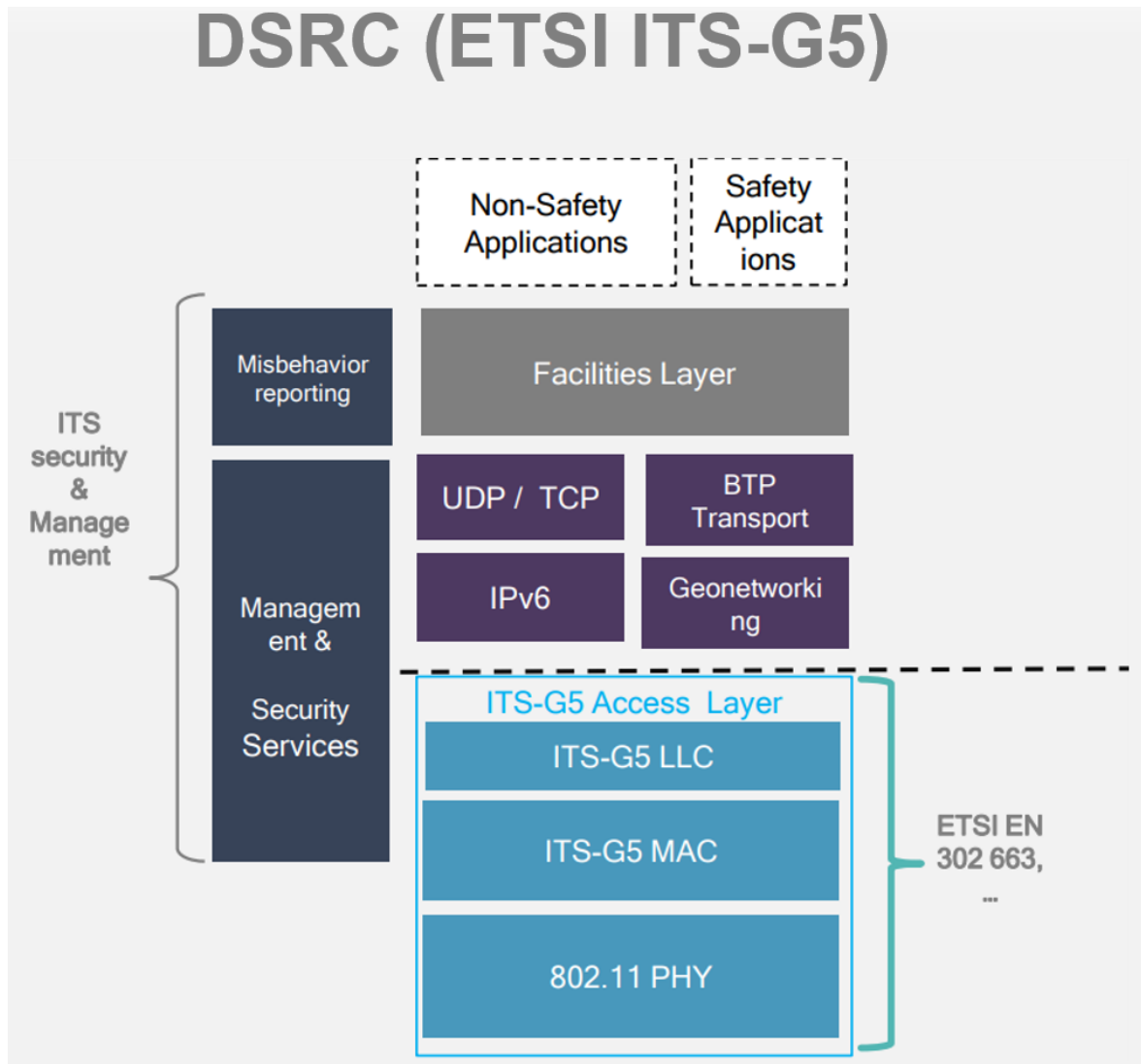


Figure 14: DSRC's architecture.

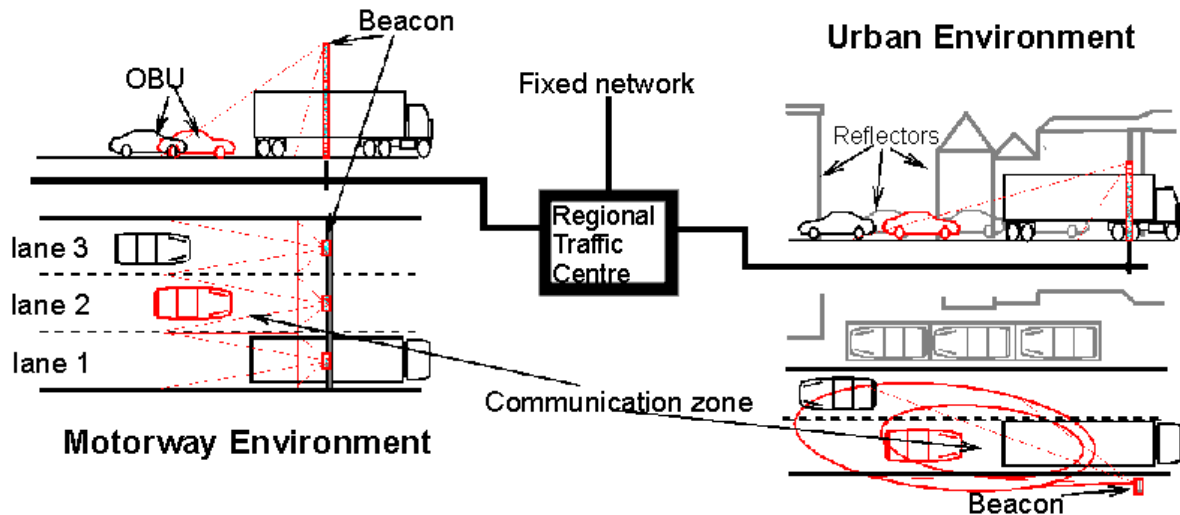
The adoption of DSRC was very poor and slow, as only few countries implemented it due to the high costs of the devices (at the time), fragmented ecosystem and the slow pace of innovation and regulations within the ecosystem.

In 2003, Europe and Japan used DSRC for electronic toll collection, and in 2017, Utah (USA) used the technology to allow certain busses to request for the traffic signal to change if they were behind schedule. Initially (2008), ETSI<sup>25</sup> allocated 30MHz of the 5.8GHz spectrum for DSRC's ITS use (in Europe), but due to the lack of adoption, it was reduced to only 10Mhz in the 5.8Ghz RF in 2020 by the FCC<sup>26</sup>, the 20MHz taken from DSRC was reallocated to its successor, Cellular V2X (C-V2X).

On top of the lack of adoption, DSRC wasn't cross-region compatible, that is, each region had its own implementation, creating significant differences in channel and spectrum usage

(5.8GHz vs 5.9GHz vs Infrared) as well as differences in transmission rate and protocol. Europe's implementation of DSRC is called ITS-G5 (standardized by ETSI) while the North American variant is WAVE (by IEEE<sup>27</sup>).

An example of the usage of DSRC in public roads can be seen below (note: OBU refers to OBE in this picture and Beacon refers to RSE):



#### 4.2.5.2 C-V2X

Cellular Vehicle-to-Everything, or C-V2X, just like its predecessor DSRC, is a messaging technology. It is an umbrella term that encapsulates all 3GPP<sup>28</sup> V2X technologies, including both V2V and V2N<sup>29</sup> communications. It enables advanced communication between vehicles, pedestrians, infrastructure, and other entities. It was developed by 3GPP to replace DSRC in the USA and ITS-G5 in Europe (Europe then decided to adopt a technology-neutral approach to C-ITS, leaving all technologies to be part of V2X applications and services) and is an alternative to WAVE (802.11p), IEEE's standard for V2X that includes DSRC.

The origins of C-V2X can be traced back to the origins of DSRC (1990's), because the objectives of both technologies are the same and C-V2X is the **successor** of the original proposed technology. Then, from 2014 to 2017, 3GPP's release 13-14 introduced C-V2X and then added extra functionalities like 5G for V2N (Vehicle-to-Network) and 5G NR<sup>30</sup> direct communications for V2V and V2I in release 15-16.

Unlike DSRC and ITS-G5 (which use WLAN technology), C-V2X uses both cellular 4G LTE<sup>31</sup> or 5G and WLAN technology to exchange ITS messages. This allows C-V2X to take advantage of the existing cellular infrastructure globally and also use WLAN technology for close range V2V communications.

The C-V2X architecture is like DSRC's OSI based model, but adding a few layers, this is so its V2N mode is seamlessly interoperable with traditional 4G/5G technology (and be able to evolve alongside 5G), it enhances security and reliability with the use of the Transport layer.:

- **Physical Layer:** Responsible for the transmission, encoding/decoding, and modulation/demodulation of raw data streams (either V2V or V2N).

- **Data Link Layer:** Manages the Medium Access Control (MAC) protocol, which is designed to manage how multiple devices share the communication medium effectively. It also ensures that the data can be transferred over the air without interference from other devices.
- **Network Layer:** Responsible for routing and forwarding data packets between different nodes in the C-V2X system. C-V2X uses a GeoNetworking protocol for this layer (explained in the following chapters).
- **Transport Layer:** Ensures reliable and efficient transmission of data between the application layer and the lower layers. It manages segmentation and reassembly of data packets, flow control and error detection / correction.
- **Application Layer:** This is where C-V2X applications reside, it includes all the services C-ITS can provide.

C-V2X's data link layer includes 2 communication modes (**Mode 3** for V2V & V2I, and **Mode 4** for V2N, figure 16), in both modes, the upper layers of the protocol are the same, while physical and data link layers differ. In the figure 15 we can see C-V2X's two modes of operation, C-V2X (V2V & V2I) and 4G/5G (V2N) mode, C-V2X's V2N interface shares the same architecture as cellular 4G and 5G protocol (same Physical and data link layers), making them compatible.

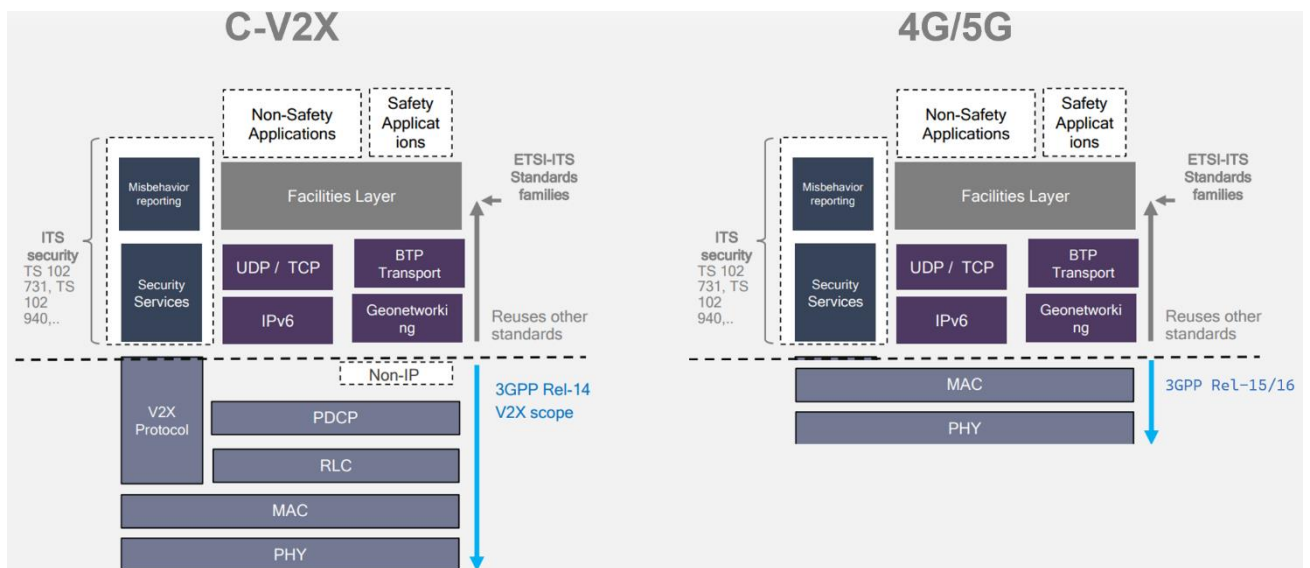


Figure 15: The two interfaces of C-V2X: C-V2X and 4G/5G (Castells, 2023)

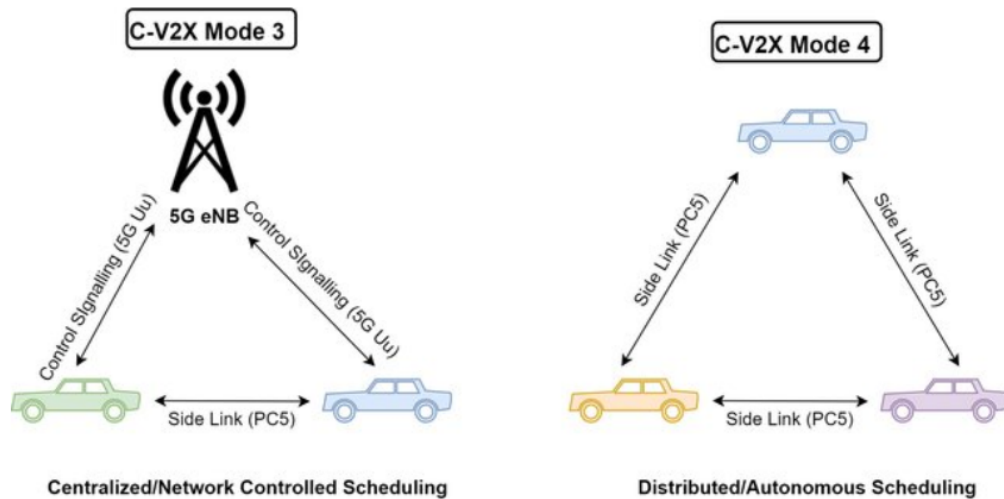


Figure 16: C-V2X Data Link Modes (CNSLab, 2020)

In the Physical layer, C-V2X supports two interfaces:

- 1- **Direct Communication (PC5) or V2X:** This mode allows vehicles to communicate with other road users (other vehicles, infrastructure, pedestrians, etc.) directly. PC5 operates in the 5.9GHz band (the official designated ITS frequency in most countries).
- 2- **Network Communications (Uu) or V2N:** This mode uses conventional mobile networks (4G LTE or 5G) to allow vehicles to connect to cloud services to send and receive real-time information about road conditions and traffic updates.

Like DSRC, C-V2X also requires a set of equipment to function properly:

- **On-Board Unit (OBU):** Physical hardware devices that are installed in vehicles, it enables C-V2X communication between vehicles and infrastructure, it collects vehicle data (direction, speed, road conditions, etc.) and sends / receives information from the communication modules (V2N and V2V).
- **Road-Side Unit (RSU):** Like DSRC’s RSE, OBUs are hardware devices installed at roadsides and other road infrastructure such as semaphores or signals. It enables V2I applications, coordinating different vehicles, sending relevant traffic information to vehicles, etc.
- **C-V2X Applications:** These are the services that local C-ITS systems implement, such as Traffic Management Systems, Vehicle Coordination System, Emergency Services Alert Systems or Hazard Alert Systems. Services can be cloud based or deployed locally, it is also responsible for managing different C-V2X users, prevent hostile attacks to the system, etc. Usually these applications have a “close to real-time” requirement to minimize delays.

### 4.2.5.3 DSRC vs C-V2X

DSRC and C-V2X are two different technologies developed for the same purpose (vehicle communications), they both have advantages and disadvantages, which I will summarize in a table:

	DSRC (ITS-G5)	C-V2X
Technology	IEEE's 802.11p (WLAN)	3GPP standards (4G LTE and 5G)
Range	300m-500m	~25% more than DSRC and up to 200% more (eeNews, 2017)
Latency	Low latency (1-2ms)	Equal or better than DSRC with 5G advancements
Spectrum	5.8GHz, 5.9GHz (802.11p) or Infrared, depending on region	Cellular (4G and 5G), it also operates in the 5.9GHz band ()
Deployment	Slow adoption	Many governments and brands have invested in C-V2X, it is gaining traction rapidly.
Ecosystem	Established ecosystem, but with differences across regions	Benefits from cellular 4G / 5G ecosystems
Interoperability	Independent from other ecosystems, may face problems in the future	Compatible with future cellular networks

The main downside of DSRC (ITS-G5) is its difference in the protocol's structure, making it incompatible with other technologies, while C-V2X (Cellular V2N mode) can seamlessly work with 4G and 5G networks, this reduces deployments costs immensely.

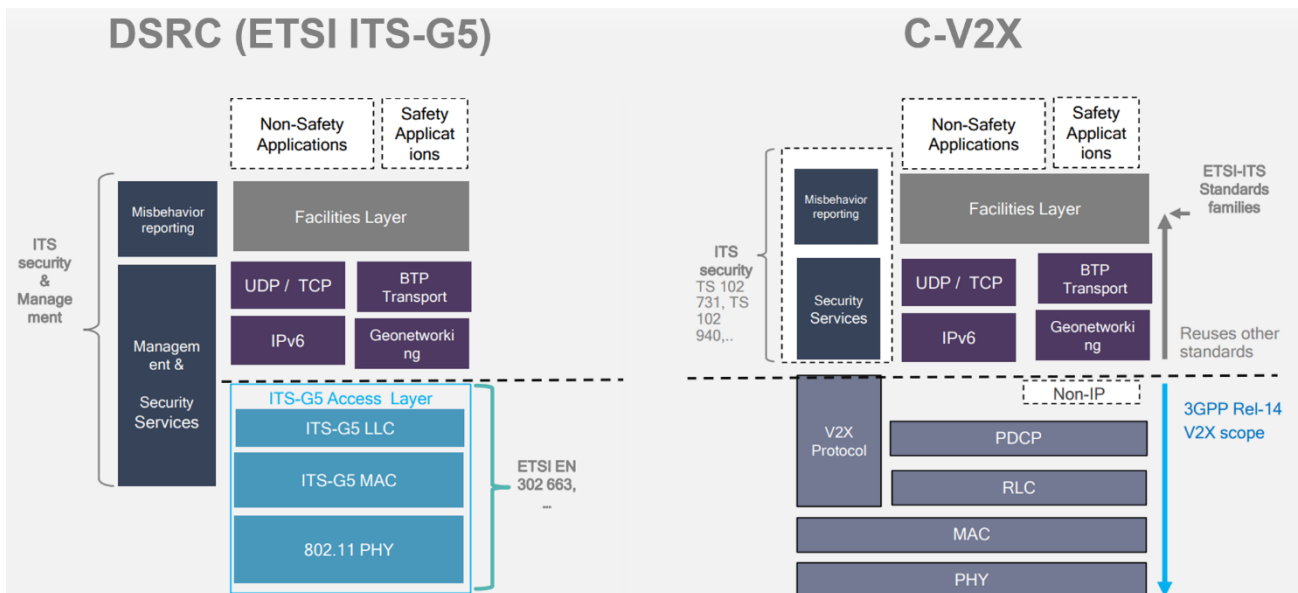


Figure 17: DSRC VS. C-V2X (Castells, 2023)

In general, C-V2X is the more popular technology, countries like China and the USA have invested heavily into the technology. The fact that not many nations have widely deployed DSRC in the infrastructure gives C-V2X a good starting point, as it can take advantage of the existing infrastructure for V2N communications, and eventually roll out C-V2X V2I equipment in the infrastructure.

#### **4.2.6 Communication Safety**

Given the amount of data that C-ITS will be able to collect and send to other vehicles, it is critical to employ cybersecurity measures to preserve the user's privacy and the security of communication between the infrastructure and users.

There isn't a set standard for C-ITS data encryption due to its global deployment; therefore, the implementation of cybersecurity measures highly depends on each region's data protection laws and the regulatory bodies governing C-ITS's deployment in that area.

For example, in Canada, the cybersecurity for C-ITS is mainly composed of three parts: front-end, back-end, and a certificate authority (CA). The front-end is responsible for encrypting and decrypting messages sent to and received from other C-ITS systems using secure encryption standards. The back end of the C-ITS system operates the PKI (public key infrastructure). When a new vehicle tries to connect to a C-ITS platform, it sends a request for a digital certificate to the CA (V2N service). If data provided is correct, the CA issues a new certificate (which doesn't necessarily contain the user's private information) for the vehicle to use when signing and encrypting its messages to other C-ITS users. Usually, the CA is the Traffic Management System (TMS) or government bodies. It is in charge of issuing, revoking, and monitoring users. If any vehicle is misbehaving, it can revoke its certificate (adding it to the Certificate Revocation List). When that misbehaving vehicle sends messages to other C-ITS systems, the OBU in those systems will ignore the messages because the certificate used to sign and encrypt the message is revoked and no longer valid. This way, all three parts of the system are separated and work seamlessly, guaranteeing data privacy and security.

Similar systems can be used in all regions, although this is not the only possible system. For example, a system where users need to register to use C-ITS is also feasible, but this system is slower and requires a manual signup process. If the central hub fails to respond to queries, the entire C-ITS system will become unusable, therefore, different clones of the service should be deployed to prevent this.

#### **4.2.7 ETSI, SAE and ASN.1**

To enable interoperability, different implementations of **C-ITS must follow standards** that define how the messaging format is. SAE, ISO and ETSI creates these standards for ITS messages.

The initial Standard was developed by Society of Automotive Engineers (SAE) for DSRC related uses, but its technology independent approach allows other technologies such as C-V2X to adopt it and apply it in the C-ITS ecosystem, that is exactly what ETSI has done with some of SAE's J2735 and ISO standard messages, adapting it and improving it for C-V2X use in **Europe**.

ETSI provides a structured message set and data frames (DF<sup>32</sup>) using the Abstract Syntax Notation One (ASN.1<sup>33</sup>), a standard (from ITU-T<sup>34</sup> and ISO) and formal notation used for describing data structures **independent of the programming language used**, the data structures defined with ASN.1 can be serialized and deserialized, this is important as complex ITS messages sent with C-V2X need to be serialized and deserialized to optimize efficiency and latency.

This is an example of an ASN.1 format for an Employee's data structure:

```
EmployeeInfo MODULE-IDENTITY
  LAST-UPDATED "202401010000Z"
  ORGANIZATION "Company XYZ"
  CONTACT-INFO "contact@example.com"
  DESCRIPTION "ASN.1 definition for employee information"

EmployeeInfoData ::= SEQUENCE {
  employeeID    INTEGER,
  name          UTF8String,
  department    UTF8String,
  projects      SEQUENCE OF ProjectInfo
}

ProjectInfo ::= SEQUENCE {
  projectID     INTEGER,
  projectName   UTF8String,
  projectStatus ENUMERATED { ongoing(0), completed(1), onHold(2) }
}
```

**Encoding** is the process in which a complex data structure is converted into a format that can be transmitted or stored (binary or hex strings), while **Decoding** is the reverse operation. The encoding and decoding process has a set of rules that tell the encoders how to treat each data type.

ETSI uses ASN.1 to define the messages, ASN.1 helps in creating a precise source code for implementing the standard, which is then compiled into an ASN library, this library can then be used to encode and decode the standard's messages using UPER<sup>35</sup>, one of the encoding rules defined for ASN.1 that is designed to be efficient and add minimum overhead (important for V2X applications).

To better understand how ASN.1 encoding works, let's define a simple example:

Suppose we have an ASN.1 structure representing a person's information including their name, age, and whether they are employed or not:

```
PersonInfo ::= SEQUENCE {  
    name      UTF8String,  
    age       INTEGER (0..100),  
    employed  BOOLEAN  
}
```

Now, let's say we want to encode the following data:

```
name: "John Doe"  
age: 35  
employed: true
```

UPER does not require alignment (we don't need to add byte padding), so we can encode "John Doe" into its ASCII number representation:

```
"John Doe" = 4A 6F 68 6E 20 44 6F 65 // 8 Bytes
```

Since age is in the range of 0-100, UPER can use 1 single byte to represent it:

```
35 = 00100011 = 23 // 1 Byte
```

To encode a Boolean, Uper represents it with 1 bit:

```
True = 1 // 1 Bit
```

With those conversions, we get the following encoded data:

```
4A 6F 68 6E 20 44 6F 65 23 1
```

And the data to be transmitted via C-V2X would be:

```
0100101001101111011010000110111000100000010001000110111101100101001000111
```

Depending on each encoder's rules, padding and other rules can be added to the encoding / decoding process.

For a recipient to decode the data stream back into the ASN.1 format, both parties must agree on a max String size, if the string does not reach the max size, padding could be added, another solution is to send the size of that field along with the data stream or in the message's header.

For each type of C-V2X message, there is an ASN1 library, these libraries use a series of ASN.1 files (source code file for the ETSI / J2735 / ISO standard messages) to perform the encoding and decoding of data streams into the ASN.1 format and back to data stream.

The repository `asn1tools` by `eerimoq` in Github is an example of an ASN.1 parsing, encoding, and decoding library.

#### 4.2.8 ITS C-V2X Messages

After standardizing the communication, data format and medium aspect of C-V2X, a set of standard messages can be defined to accommodate each use case. Different message standard exists for each use case, each is used for a special purpose and together they create the ITS platform.

To not repeat the same ASN.1 specifications, ETSI divided common data specifications in a set of CDD<sup>36</sup>, where a common specification is defined, it is then used in different other specifications (by importing the definition). One example of an CDD can be found in the following link (last visited 04/2024):

[https://www.etsi.org/deliver/etsi\\_ts/102800\\_102899/10289402/01.03.01\\_60/ts\\_10289402v010301p.pdf](https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.03.01_60/ts_10289402v010301p.pdf)

Such common specification includes the ITS PDU header (“Header” in figure 24), a common section of the message sent that indicates the protocol version (Version of the ITS message), messageID (indicates the type of message, for example, CAM’s messageID is 2, while DENM’s is 1.) and the origin stationID of the originating ITS-S<sup>37</sup> (ITS Station).

There are many messages in the ETSI standard, like VAM, MCM or RTCMEM, but most of them are still under development, and more messages may be created for future needs.

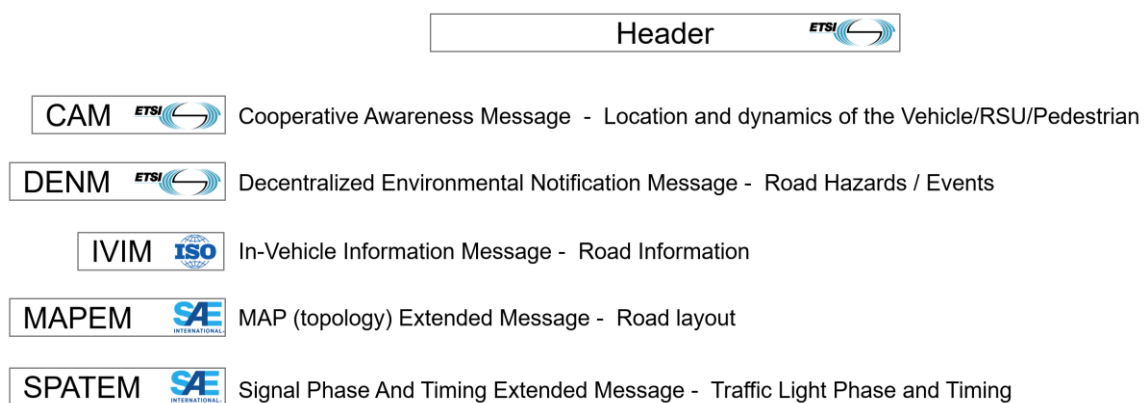


Figure 18: C-ITS Messages Summary (Castells, 2023)

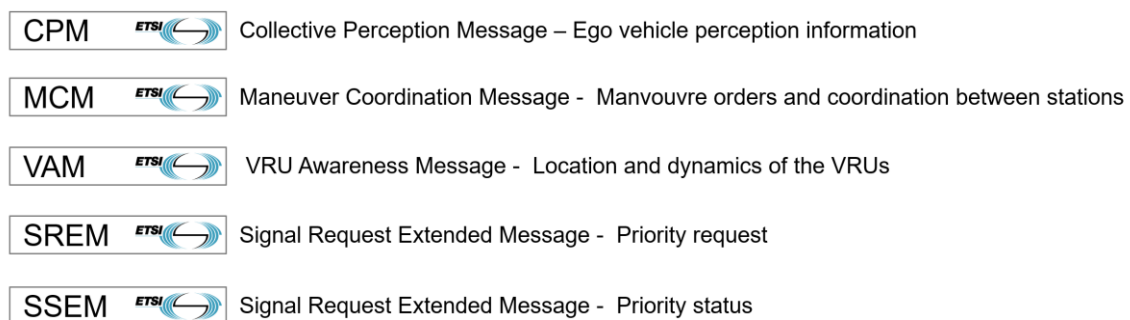


Figure 19: Other C-ITS Messages Summary (Castells, 2023)

The two figures above show the origins (ETSI, SAE or ISO) of each type of messages and the summary of their purpose.

#### ***4.2.9 GeoNetworking protocol***

The ITS platform requires a special routing protocol to quickly redistribute safety messages in a fast manner. For this purpose, GeoNetworking was specifically developed for vehicle communications and operates at the network layer.

The GeoNetworking protocol in ITS helps route packets within an ad-hoc network, that is, a decentralized type of wireless network where, all nodes of the network communicate directly with each other, and all nodes participate in packet routing by forwarding data to other nodes. It uses geographical positions to move packets around, which includes adding geographical data to the IPV6 IP address. GeoNetworking enables communication between different ITS stations and manages the distribution of packets across specific geographical areas ([a] ETSI, 2019).

While different ITS setups might have their own versions of this network, they all rely on some kind of geographical data to help route packets, for example GeoBroadcast sends messages to all clients in a specific region and Unicast sends it to specific vehicles.

## 5. Specification of Idiada's ITS Platform

Idiada Automotive Technology SA (Idiada) is a Spanish company that provides design, engineering, testing and homologation services for the automotive sector. The company has more than 3150 employees across 22 countries and is a subdivision of Applus+, a world leader in inspection, testing and certificating services for vehicles.



*Figure 20: Idiada's Logo (Applus, 2024)*

Idiada, like many other organizations, is implementing its own ITS and C-ITS platform to provide clients with a standardized testing environment for ITS platforms. These two systems serve different purposes and are deployed in different ways, but they work seamlessly together, bringing clients the ultimate C-ITS testing services through C-V2X and 5G. The main difference is that the ITS platform focuses on providing 5G V2N communications for C-ITS systems and is deployed as a cloud service where clients can connect to it, while the C-ITS platform is capable of both 5G V2N and Cellular C-V2X communications and is mainly deployed in vehicles and infrastructure RSUs.

### 5.1 ITS Platform

Idiada's ITS platform offers collaborators and B2B<sup>38</sup> clients a V2N platform that sends and receives ITS messages through 5G connectivity with which they can test their vehicles and any ITS systems. To do this, Idiada has several RSUs installed in their testing grounds (for V2I communications) and 5G antennas around the facility to test the V2N communications.

The platform uses a message delivery system that is based on the location of the system. The advantage of Idiada's platform is that it provides different collaborators an "easy" way to test their C-ITS systems through an API interface before deploying it.

The ITS platform offers Idiada's clients and local governments the possibility of publishing their own ITS messages (generated by their vehicles or RSU's) through Idiada's infrastructure and receive relevant ITS messages back through an API interface using V2N communication (using 5G, for example). This platform's main purpose is to provide a fast, secure, and reliable ITS V2N infrastructure for all participating parties.

Participating bodies will need to authenticate themselves with the platform authentication service, then, they can send their own ITS messages to Idiada's ITS network and receive relevant messages back.

## 5.2 Idiada's C-ITS Platform

The C-ITS platform is an application that is mainly deployed in vehicles, it contains all the application logic to perform C-ITS coordination and cooperation with other systems. The platform can send vehicle information to nearby vehicles (V2V using C-V2X) or to Idiada's ITS platform (V2N using 5G). It can also process ITS messages received either from Idiada's ITS platform or direct V2V C-V2X communication from RSU or vehicles.

With the information that it sends and receives, it is capable of:

- Performing vehicle manoeuvre coordination with other C-ITS entities.
- Sending alerts to driver.
- Providing the GLOSA function.
- Alerting the driver of emergency vehicles approaching.
- Notifying the driver of possible collision ahead.
- Informing the driver about road conditions and weather hazards.

On top of these two services, Idiada also has its own RSU management system, where different RSU placed in the road infrastructure can send ITS messages to the ITS cloud service, then replicate the messages to relevant locations.

## 5.3 Technologies Used

Both Idiada's ITS platform and C-ITS platform are mainly developed in Java using Spring Boot, they are still in development and are currently deployed in Docker Swarms and Kubernetes in Idiada's servers. The reason Java is used is because the libraries of the ITS message encoders for ASN.1 that Idiada uses is compiled in Java.

The routing mechanism used by Idiada is based on publish / subscribe pattern using MQTT and its topics system (explained in next chapters), although other systems can also be used in the V2N scenario in the future.

This does not mean that any other third party's ITS systems that use other languages or applications cannot interact with Idiada's services, as both ITS and C-ITS follow the standard ETSI message's structure and adhere to their ASN1 formats. So, if the third party's ITS platform comply with the standard and use the correct ASN1 format, interoperability will be possible using the same MQTT messaging protocol for V2N or using C-V2X for V2X communications (ASN.1 formatted messages are sent in nV2X).

### 5.3.1 Tiles and Map Matching

The GeoNetworking protocol Idiada's ITS platform implements is the GeoBroadcast model, which delivers ITS messages based on the location of the recipients:

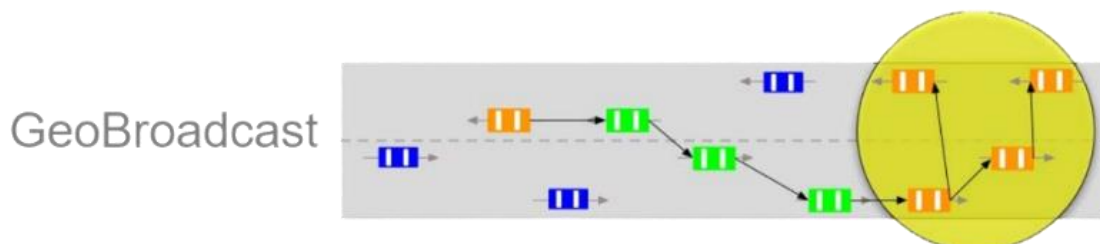


Figure 21: GeoBroadcast protocol (Castells, 2023)

This routing system is based on the geographical position of nodes, delivering to those that are within the region specified in the message. To do this, Idiada’s ITS platform uses a Tile based system, which divides the earth’s surface in 4 quadrants, assigning a value from 0 to 3 to each quadrant. Within each of the 4 quadrants, it is divided into 4 quadrants again, this process repeats for as many times as needed to achieve the desired precision.

A graphical representation of the system looks like this:

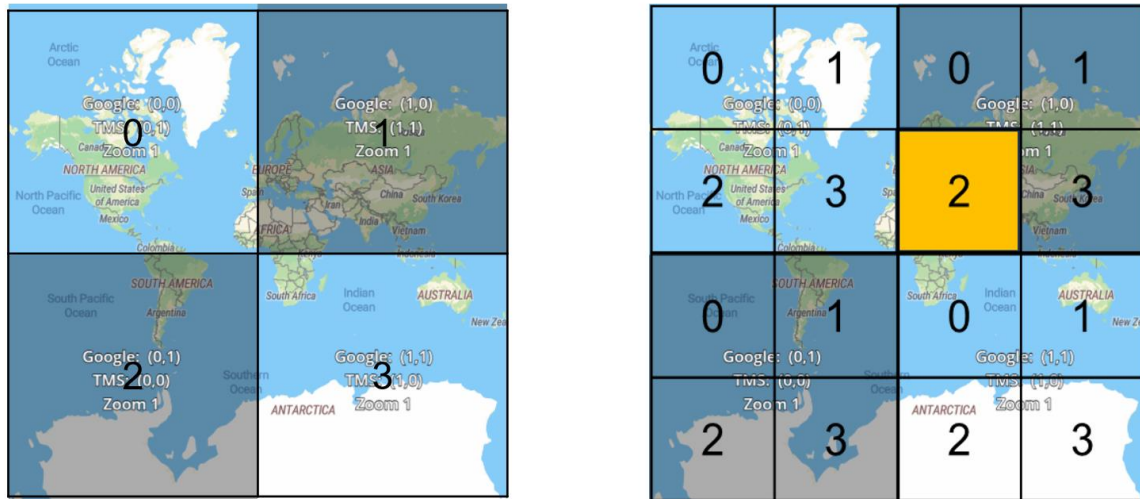


Figure 22: Tile Mapping System (Castells, 2023)

To represent a specific area, for example, we could define 1,2,3,2,1,0,0,1,2, which is choosing the corresponding quadrant for each zoom level. The first quadrant (the whole world quadrant) is the zoom level 0, for each quadrant we “zoom” in, it adds a level to the zoom.

The process which checks if a message affects a vehicle or entity is called “Map Matching”, where a process uses the altitude and longitude values to calculate the “tile” which the vehicle is at, and checks if it matches the affected zone by the message received.

### 5.3.2 MQTT

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

## 5.4 Requirements

For different projects, different functional and non-functional requirements are needed.

### 5.4.1 Functional Requirements

There are requirements for both platforms and specific requirements for each platform, the general requirements are the following:

0. Clients (manufacturers, OEM<sup>39</sup>s, partners) must be able to test their ITS platforms in Idiada's facilities using the ITS and C-ITS platforms.
1. Idiada's ITS systems must be compatible / interoperable with all other ITS platforms that follow ETSI's standards.
2. Idiada must offer both V2N and V2V communications.

The list of functional requirements (FR) is the following for the two platforms:

- **ITS Platform:**

- 0- Authorized & authenticated users can subscribe to the GeoMessaging MQTT broker.
- 1- Authorized & authenticated users can publish to the GeoMessaging MQTT broker.
- 2- Any user can subscribe to message extractor's broker (External Broker).
- 3- External users cannot publish to the external broker.
- 4- Message Extractor must extract all messages from GeoMessaging Broker to External Broker except RSU and internal messages.
- 5- Only Idiada's testing ground RSU units can register in the RSU Manager.
- 6- Idiada's Push / Pull adapters must comply with the external project's format requirements.
- 7- Providers can provide ITS Messages to the Message Manager (GeoMessaging).
- 8- The Message Manager must have an API interface for external entities.
- 9- The GeoMessaging platform must disseminate messages using a GeoNetworking protocol.
- 10- Only authorized users can authenticate via the Identity Manager.

- **C-ITS Platform:**

0. Only authorized and authenticated users can connect / transmit messages with Idiada's ITS platform.
1. C-ITS can use V2N, V2I, and V2V.
2. C-ITS must provide Cooperative Intelligent Transport Systems functions.
3. The application must provide a human interface.
4. V2V must use C-V2X protocol.
5. JWT security is used for GeoMessaging related authentication.
6. JWT should expire in 1h.

#### ***5.4.2 Non-Functional Requirements***

General non-functional requirements:

0. Idiada's V2N should work via GeoBroadcast using Tiles.
1. MQTT should be used for publisher / subscriber message dissemination.
2. The platforms should be scalable and offer close to real-time latency.
3. V2N should use 5G connection.
4. V2V should use C-V2X
5. Spring Boot is used for services applications.
6. Services should be deployed in Docker Swarms or Kubernetes.
7. Data should be encrypted using a PKI system.
8. A KPI<sup>40</sup> system should be implemented to monitor statistics.
9. Applications should be configurable using environment variables.

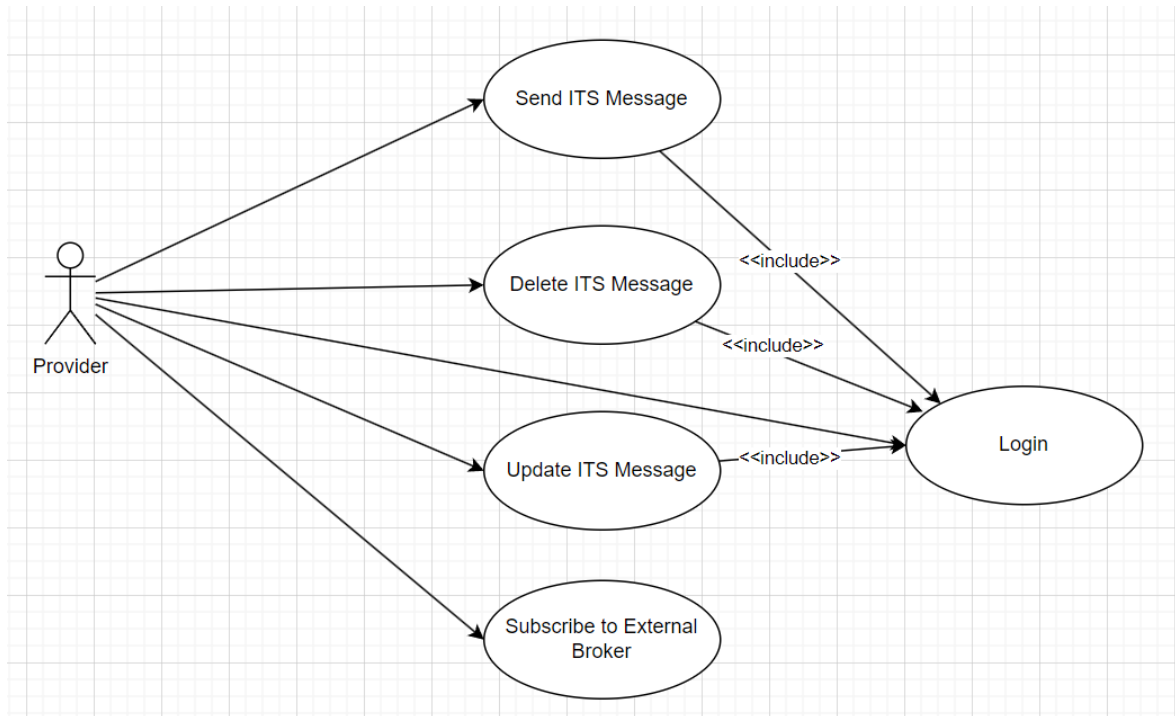
The list of non-functional requirements (NFR) is the following for the two platforms:

- **ITS Platform:**
  - 0- A SECRET shared between clients and Idiada must be provided when authenticating in the Identity Manager.
  - 1- Java is used for all libraries / applications.
  - 2- A special Java Object should be used when processing the ITS message in the ITS systems.
  - 3- The Message Manager implements REST API for external interaction.
  - 4- Message Extractor should be able to handle a high throughput.
  - 5- A valid Project\_Key should be provided to interact with GeoMessaging.
  
- **C-ITS Platform:**
  - 0. The latency between message processing and user notification in the HMI should be minimal.
  - 1. Angular should be used for the HMI.
  - 2. C-ITS should be able to handle a high throughput.
  - 3. Minimal resources should be used when not actively receiving messages.
  - 4. V2N or V2V should be able to be turned off via environment variables.

## 5.5 Use Cases

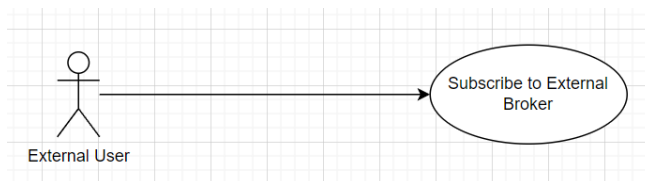
### 5.5.1 Provider

A provider has the following use cases:



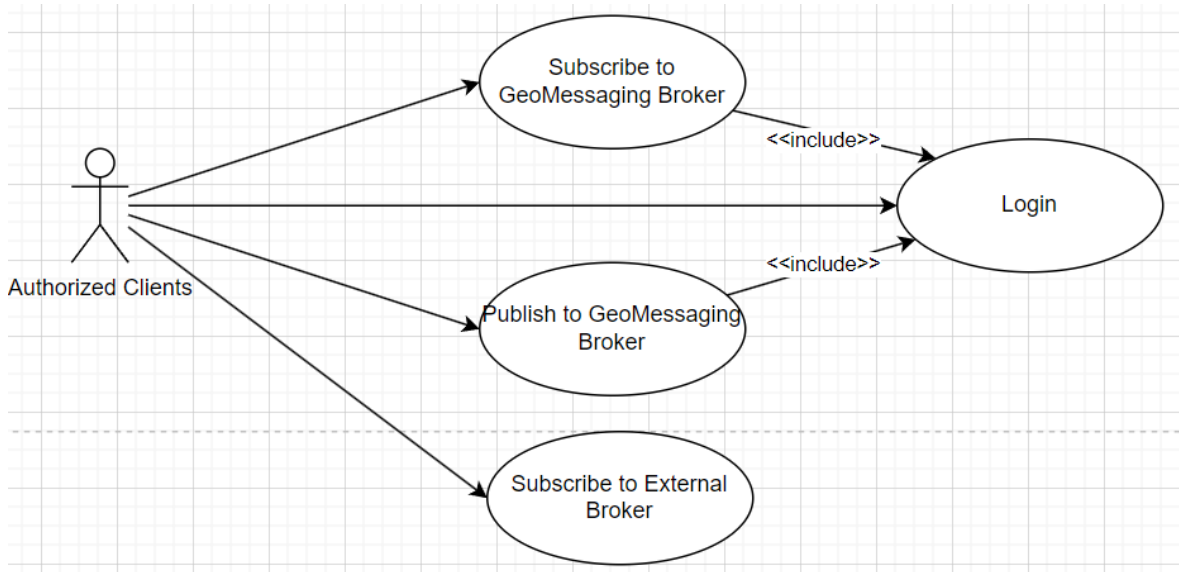
### 5.5.2 External User

An external user has only 1 use case:



### 5.5.3 Authorized Clients

Authorized clients have the privilege to subscribe and publish directly from / to the GeoMessaging broker.



**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

## **6. Design**

Idiada's ITS can be split into two platforms, GeoMessaging and ITS Platform.

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

## 6. Development of the C-ITS platform

During the development and learning process of Idiada's ITS platforms, a series of tools and development skills were acquired, including:

- **Java:** Java is a programming language that I've learned during my education in the degree, and it was used during the development of the platform.
- **Spring Boot:** A framework based on Spring Framework that facilitates the creation of Spring projects. It is designed to streamline the setup and development, allowing developers to focus more on business features rather than the intricate setup processes. The most important part of Spring is its auto-injection functionality that works with "beans", it automatically detects classes with the same name as a variable noted with @autowire and injects that instance of the object into that bean.
- **Maven:** Maven is a comprehensive project management and build automation tool primarily used for Java projects, during the development of Idiada's platform Maven was used to manage dependencies and deploy / run tests.
- **WSL<sup>41</sup>:** WSL provides a Linux kernel (in WSL 2) to deploy and run Linux binaries in windows, this platform was used during the development to deploy instances of all services in the local machine using Docker Images.
- **Docker Swarm:** To deploy different services in the same virtual WSL environment, Docker Swarms were used to deploy clusters of applications that can be managed seamlessly, in this context, Portainer is used to manage the clusters (accessible through port docker\_ip:9000).
- **MQTT:** The MQTT messaging technology used in the ITS platform requires special libraries and entities known as "brokers" that send messages according to the topic's subscriber. Different applications exist to manage these brokers (such as MQTTx).
- **Jira:** To have a more flexible, productive, and efficient software production cycle, an **Agile** project management is adopted in the software team to adopt a combination of incremental / iterative development approach. In the team, Kanban was used initially, then we adopted a Scrum methodology.
- **Testing:** To test the applications and services developed, an exhaustive set of unit tests are developed for all components of an application, as well as for new functionalities added to existing applications. A set of exhaustive integration tests are also developed to verify the correct functionalities when integrating different components. To implement the tests, Junit and Mockito was used for both types of tests.  
To verify the set of tests, SonarQube is used to check for code coverage, code smells, and code weaknesses.
- **Jenkins:** A **CI/CD** (Continuous Integration / Continuous Delivery) workflow is used in all projects, where projects are continuously uploaded (committed), which triggers a pipeline in the integration server (Jenkins). The pipeline tests the uploaded the project with tests, and if successful, it produces a binary that is automatically uploaded to Idiada's repository. To enable this, projects contain a JenkinsFile and a deployment script to automate the steps.
- **IDEs:** A series of IDE applications are used; the most common ones include IntelliJ Idea (for java-based applications) and Visual Studio Code (for all other applications).
- **Hexagonal Architecture:** The applications and services are developed following the hexagonal architecture (also known as adapters / ports pattern), it decouples functionalities of the application into the core (business logic), adapters (bridge

between the external world and the ports) and ports (interfaces that define points of interaction between the application and the outside world).

These three layers are labelled as **domain**, **application**, and **infrastructure**, respectively.

- **Domain Layer:**  
This is the core of the application, where business logic and domain model reside. The domain layer is strictly focused on entities, value objects, and domain events. It should define interfaces that the other layers need to implement to interact with the domain.
- **Application Layer:**  
The application layer sits outside the domain layer and acts as a middleman between the domain and the external world. This layer implements entry points to the application (a REST API for instance). It communicates with the domain layer through interfaces (ports) and implements functionality using services.
- **Infrastructure Layer:**  
Specific infrastructure implementations are handled in the infrastructure layer, which includes everything outside the application's core logic that is needed to run the application, for example Spring framework and specific database implementations.

For more information on Hexagonal Architecture in Spring Applications, visit this website: <https://www.baeldung.com/hexagonal-architecture-ddd-spring>

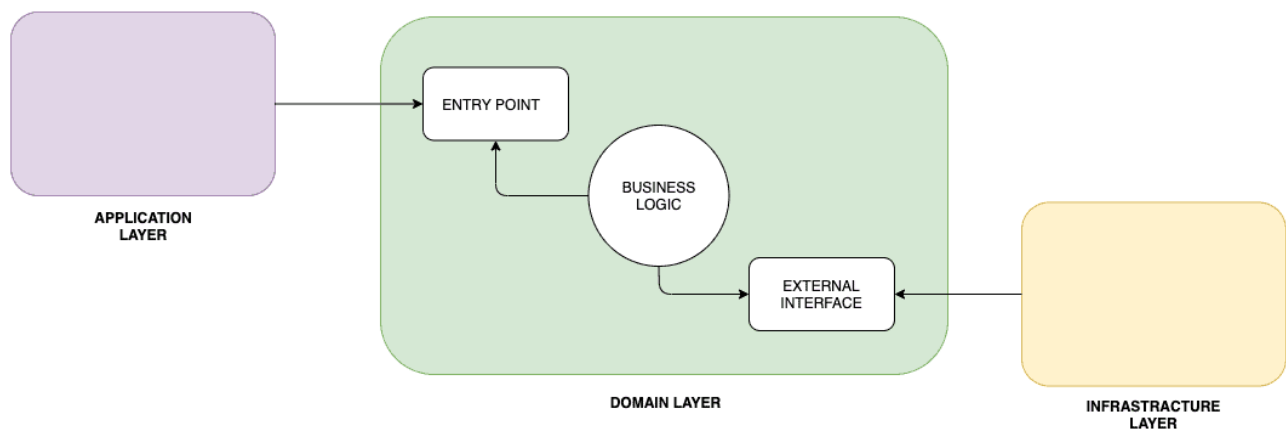


Figure 23: Hexagonal Architecture (Baeldung, 2024)

The initial weeks of my internship at Idiada I completed a series of mandatory courses at the company (Laboral Risk Prevention, Cybersecurity, Anti-Corruption, etc.), followed by a series of training videos about Spring Boot, WSL & Docker, and Hexagonal Architecture.

After the initial introduction to the technologies and environments, I started working on the ITS platform's projects. Jira's Agile workflow assigns each team member "tickets" to complete, each ticket is an "issue" or a new feature to implement in the ITS platform, different projects with different clients require specific features to be implemented, creating new services or enhance existing ones.



```
}
```

Payload:

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```

Signature:

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret (256bits)  
)
```

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

To verify the functionality of the new bug fix, PostMan was used to verify the different cases where different fields were null or missing, after success in that test, unit tests were implemented to expand the existing component test.

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

#### 6.4 Update All services that depend on a library.

This maintenance task was assigned to me to learn about maven and POM.xml. To do this task, I had to check the different dependencies across all services we had and check the ones that used the SDK library, once detected the projects that depend on it, I started from the bottom up, changing the pom.xml file to update the dependency. A normal pom.xml file would look like this:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>com.idiada</groupId>  
  <artifactId>exampleProject</artifactId>  
  <version>1.0 </version>  
  
  <!-- Properties section for managing dependency versions -->  
  <properties>  
    <java.version>11</java.version>  
    <spring.version>5.3.15</spring.version>  
    <spring.boot.version>2.5.6</spring.boot.version>
```

```

</properties>

<dependencies>
  <!-- Spring Boot Starter Web -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <version>${spring.boot.version}</version>
  </dependency>

  <!-- Spring Context -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${spring.version}</version>
  </dependency>

  ... More dependencies
</dependencies>

<build>
  <plugins>
    <!-- Spring Boot Maven Plugin -->
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <version>${spring.boot.version}</version>
    </plugin>
  </plugins>
</build>
</project>

```

The highlighted area is where one needs to change to update the version of dependencies, we could write the version directly into the `<version>` tag area, but having a properties section simplifies big projects that have thousands of dependencies.

To check for the available updates in maven, we can use the command:

```
mvn versions:display-dependency-updates
```

This command would show us the list of available updates:

```

[INFO] The following dependencies in Dependencies have newer versions:
[INFO]   com.fasterxml.jackson.core:jackson-databind ..... 2.9.10.1 ->
2.12.3
[INFO]   junit:junit ..... 4.12 ->
4.13.2
[INFO]   org.apache.commons:commons-lang3 ..... 3.9 ->
3.12.0

```

With this information, we can manually change the version in the properties section, then manually running all tests to see if any errors occur, since my task was to change a major version (breaks retro-compatibility), it led to a few errors that I needed to fix in all services to implement the new features.

To run tests with maven, we can use the commands:

```
Mvn clean verify  
Mvn clean test
```

Once all tests pass and no warnings / errors are found, I committed the changes and created a merge request to merge with the main branch and update the repository with Jenkins. This whole process had to be repeated for all projects that depended on this library.

## 6.5 Create a PostMan Collection for Message Manager API

Start Date: 06/12/2023 → Finish Date: 11/12/2023.

This task consisted of creating a set of PostMan API calls to test all Message Manager functionalities, including all types of messages (both V1 and V2), different formats (ASN.1 and Idiada's DTO) as well as invalid requests (all types of errors produced by incorrect formats, fields, messages, etc.).

[-- CONFIDENTIAL SECTION --]

...

[-- END OF CONFIDENTIAL SECTION --]

## 6.6 Remove JSON implementation in the Message Manager

It was decided that the Message Manager would only support ASN.1 format and Idiada's own DTO format for the API interface, so my task was to remove all implementation of the JSON format in the message manager and refactor if necessary. To complete it I investigated the application's behaviour and removed all classes / methods that had a Json implementation or did not provide any function to the service.

For example, coders and decoders for the JSON format were no longer needed, so I removed those from the asn.1 adaptation lib instance, as well as other controllers that responded to a JSON format, finally, unit tests and integration tests that tested the Json format were also removed.

Once refactored, tests were run to check functionality with the remaining two interfaces (DTO and ASN.1), which were still functioning, then I committed the changes and created a merge request for a senior developer to review and approve.

## 6.7 Update All Applications to Spring 3

Start Date: 02/01/2024 → Finish Date: 24/01/2024.

This was one of the harder tasks I was assigned, because all projects were using Spring 2.X, and Spring 3 had a few changes that would require a refactor in all projects, mainly, the changes were that javax was no longer used and Jakarta is now used, since all projects were using javax (persistence and servlet), the infrastructure layer had to be changed to adapt to the new "infrastructure" (Jakarta persistence and servlet).

Details can be found in the following page for a simple overview of the changes:

<https://www.baeldung.com/spring-boot-3-migration>

A snippet of an example of the change could be the following:

### Spring 2.X:

```
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.servlet.http.HttpServletRequest;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
```

```

@Controller
@RequestMapping("/api/data")
public class DataController {

    @PersistenceContext
    private EntityManager entityManager;

    @GetMapping("/get")
    public String getData(HttpServletRequest request) {
        // Example query using EntityManager
        String query = "SELECT d FROM Data d WHERE d.id = :id";
        Data data = entityManager.createQuery(query, Data.class)
            .setParameter("id",
request.getParameter("id"))
            .getSingleResult();
        return "dataView"; //
    }
}

```

### Spring 3.X:

```

import jakarta.persistence.EntityManager;
import jakarta.persistence.PersistenceContext;
import jakarta.servlet.http.HttpServletRequest;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/api/data")
public class DataController {

    @PersistenceContext
    private EntityManager entityManager;

    @GetMapping("/get")
    public String getData(HttpServletRequest request) {
        // Example query using EntityManager
        String query = "SELECT d FROM Data d WHERE d.id = :id";
        Data data = entityManager.createQuery(query, Data.class)
            .setParameter("id",
request.getParameter("id"))
            .getSingleResult();
        return "dataView";
    }
}

```

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

## **6.8 Investigate the ETSI standard to check if Idiada's MAPEM comply with the standard**

Start Date: 22/02/2024 → 27/02/2024.

This ticket required diving into the standards of ETSI and SAE (because MAP is adapted from SAE's J2735) and look for the field that the ticket is talking about, specifically, the ConnectionTrajectory-addGrpC under intersection->laneSet->regional (Reg-GenericLane).

After investigating the necessary documents and Idiada's implementation, I created a report (**Annex [B]**).

## 6.9 Define a KPI logging required for upstream and downstream in Extractor.

Start Date: 27/02/2024 → Finish Date: 04/03/2024

Implementation of the KPI in extractor started: 10/03/2024 → Finished: 22/03/2024

A new requirement (NF requirement 8) was created for certain projects, which consists of sending relevant statistics to a remote KPI platform that will be used to monitor and improve existing services if any bottle neck or lack of performance is detected:

[-- CONFIDENTIAL SECTION --]

...

[-- END OF CONFIDENTIAL SECTION --]

My task was to define the KPIs for the throughput of the extractor (uplink and downlink), as well as implementing it into the existing application. To complete this task, I had to modify some functions of the extractor to include a calculation of the amount of data that the extractor processes (both incoming and outgoing messages) each second. This is done by using a counter that is increased each time the extractor sends or receives a message, and a thread in the application that is in charge of calculating the difference between the counter 1s ago and now and send that data to a broker.

[-- CONFIDENTIAL SECTION --]

...

[-- END OF CONFIDENTIAL SECTION --]

## 6.10 Create a Sequence diagram about the C-ITS gets a JWT from the identity manager.

Start Date: 04/03/2024 → Finish Date: 08/03/2024.

I was also tasked to make a sequence diagram about the interaction between C-ITS and identity Manager, which resulted in a complex diagram (Annex [C]). This task was carried out between the finish of the KPI definition phase and the implementation of the KPI in the extractor phase.

## 6.11 Create a new Service (push adapter) for a client.

*Figure 24: KPI Platform*

Start Date: 08/03/2024 → Finish Date: 25/04/2024.

This is the most important task that I was assigned to complete, it consists of creating a new project that serves as a push adapter in the ITS platform for a specific Client (i2cat, in this case).

[-- CONFIDENTIAL SECTION --]

...

[-- END OF CONFIDENTIAL SECTION --]

## 6.12 Document how C-ITS generates CAM messages and sends the location to GeoMessaging.

Start Date: 25/04/2024 → Finish Date: 30/04/2024.

This ticket consists of documenting the flow of the geographical location of vehicles to the OBU (where C-ITS is deployed), then to the GeoMessaging platform. To do this, I created an activity diagram (Annex [D]).

## 6.13 Modify Logging in all projects.

Start Date: 02/05/2024 → Finish Date: 27/05/2024.

The last ticket I was assigned was the longest one, the task is to modify all existing projects and unify the logger used to avoid warning messages across different services, this is because each team member implemented different loggers for different projects (Log4J, Logback, etc.).

The solution I proposed is to use the SLF4J (Simple Logger Façade for Java), which allows the use of other loggers underneath it, since it is just a façade. As for the specific logger, we decided to use Logback as it is more popular.

For this, I've created a logger settings xml file to have 1 design for all logs across the platform:

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

This logging config file enables color for different logging levels (TRACE, DEBUG, WARN, INFO, ...).

Modifying all classes and all lines of code where a logger is used is time consuming (because the older logger had different method names and parameters), so I took almost a month to carry out this feature fix.

## **7. Conclusions**

The development of this thesis has been enriching in my personal and professional growth in several ways.

On one hand, the research and development of ITS platforms and C-ITS platforms have given me a unique experience where I learned the newest, most advanced technology in the communication / automotive sector. This knowledge not only prepares me to enter the professional world but also highlights the extensive effort behind some of the "common" technologies we use daily, which are much more complex than they appear.

On the other hand, the development of this thesis has given me a lot of insight into the professional programming world, where a wide set of applications and techniques are applied to achieve an efficient workflow. On top of that, progressing, learning, collaborating, and sharing skills with my colleagues have taught me that group work is essential in the daily life of a professional, which differs by a lot from my previous opinion on teamwork, because I used to love working alone and without other people's help. My struggles during the internship showed me how important a simple advice can be, instead of wasting hours of my time "googling" and trying to find a solution, a simple trick from a colleague can fix my issue instantly.

Regarding the objectives of this thesis, I am pleased to confirm that they have been successfully achieved. I have both acquired a deep knowledge in the Intelligent Transportation Systems and have developed (although partly) an ITS platform that adheres to a global standard and is being used to test production applications that aren't even on the streets yet, but someday they will be out there, and I will know that I had a part (really small one, that is) in the process of achieving it.

## 8. References

- [a] ETSI. (2019). Retrieved from [https://www.etsi.org/deliver/etsi\\_en/302600\\_302699/3026360401/01.04.00\\_20/en\\_3026360401v010400a.pdf](https://www.etsi.org/deliver/etsi_en/302600_302699/3026360401/01.04.00_20/en_3026360401v010400a.pdf)
- [b] ETSI. (2019, April). Retrieved from [https://www.etsi.org/deliver/etsi\\_en/302600\\_302699/30263702/01.04.01\\_60/en\\_30263702v010401p.pdf](https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf)
- 5GAA. (2024). Retrieved from <https://5gaa.org/c-v2x-explained/#:~:text=C%2DV2X%20explained%20%2D%205GAA%20Go%20to%20menu%20Go%20to%20content%20Membership%20Menu%205GAA%20About%20us%20Governance%20COMMUNITY%20Our%20Members%20Our%20Partners%20Projects%20Advisory%20Join%20the%205GAA%20F>
- Applus. (2024). Retrieved from [www.applus.com](http://www.applus.com)
- Auto Motor Sport. (2024). *Auto Motor Sport*. Retrieved from <https://www.auto-motor-und-sport.de/tech-zukunft/mobilitaetsservices/glosa-audi-reitet-die-gruene-welle/>
- Baeldung. (2024). Retrieved from <https://www.baeldung.com/hexagonal-architecture-ddd-spring>
- Castells, J. (2023).
- CNSLab. (2020). Retrieved from [https://www.researchgate.net/publication/343758202\\_Evolution\\_of\\_V2X\\_Communication\\_and\\_Integration\\_of\\_Blockchain\\_for\\_Security\\_Enhancements](https://www.researchgate.net/publication/343758202_Evolution_of_V2X_Communication_and_Integration_of_Blockchain_for_Security_Enhancements)
- Cookson, G. (2017). Retrieved from Inrix: <https://inrix.com/wp-content/uploads/2017/02/INRIX-Europes-Traffic-Hotspots-Research-FINAL-hires-1.pdf>
- eeNews. (2017). Retrieved from <https://www.eenewseurope.com/en/dsrc-vs-c-v2x-looking-to-impress-the-regulators/>
- ETSI. (2014, September). Retrieved from [https://www.etsi.org/deliver/etsi\\_en/302600\\_302699/30263703/01.02.01\\_30/en\\_30263703v010201v.pdf](https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.01_30/en_30263703v010201v.pdf)
- ETSI. (2018). Retrieved from [https://www.etsi.org/deliver/etsi\\_ts/102800\\_102899/10289402/01.03.01\\_60/ts\\_10289402v010301p.pdf](https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.03.01_60/ts_10289402v010301p.pdf)
- ETSI. (2020). Retrieved from [https://www.etsi.org/deliver/etsi\\_ts/103300\\_103399/103301/01.03.01\\_60/ts\\_103301v010301p.pdf](https://www.etsi.org/deliver/etsi_ts/103300_103399/103301/01.03.01_60/ts_103301v010301p.pdf)
- ETSI. (2023). Retrieved from [https://www.etsi.org/deliver/etsi\\_ts/103300\\_103399/103324/02.01.01\\_60/ts\\_103324v020101p.pdf](https://www.etsi.org/deliver/etsi_ts/103300_103399/103324/02.01.01_60/ts_103324v020101p.pdf)
- Inrix. (2021). *Inrix*. Retrieved from <https://inrix.com/press-releases/2021-traffic-scorecard/>
- ISO 19321. (2015).

- Obaid, M. (2019). *ResearchGate*. Retrieved from [https://www.researchgate.net/publication/333254227\\_A\\_Novel\\_Model\\_Representation\\_Framework\\_for\\_Cooperative\\_Intelligent\\_Transport\\_Systems/figures?lo=1](https://www.researchgate.net/publication/333254227_A_Novel_Model_Representation_Framework_for_Cooperative_Intelligent_Transport_Systems/figures?lo=1)
- Q-free. (2021). Retrieved from <https://www.q-free.com/wp-content/uploads/2021/03/C-ITS-On-Board-Unit-Product-Sheet.pdf>
- RadioYTMexiquense. (2020). Retrieved from <https://radioytmexiquense.mx/index.php/2020/07/06/comienzan-a-operar-semaforos-inteligentes-en-vialidades-de-la-cdmx/>
- ResearchGate. (2015). Retrieved from [https://www.researchgate.net/publication/312934832\\_The\\_Effect\\_of\\_Radio\\_Channel\\_Modelling\\_on\\_the\\_Network\\_Performance\\_in\\_VANET/figures?lo=1](https://www.researchgate.net/publication/312934832_The_Effect_of_Radio_Channel_Modelling_on_the_Network_Performance_in_VANET/figures?lo=1)
- ResearchGate. (2018). Retrieved from [https://www.researchgate.net/publication/325681154\\_In-vehicle\\_data\\_recording\\_storage\\_and\\_access\\_management\\_in\\_autonomous\\_vehicles/figures?lo=1](https://www.researchgate.net/publication/325681154_In-vehicle_data_recording_storage_and_access_management_in_autonomous_vehicles/figures?lo=1)
- SAE J2735. (2022).
- slideserve. (2014). Retrieved from <https://www.slideserve.com/gavivi/abstract-syntax-notation-one-asn-1>
- Smartcitiesdive. (2017). *Smartcitiesdive*. Retrieved from <https://www.smartcitiesdive.com/ex/sustainablecitiescollective/intelligent-transport-systems-smarter-cities/18319/#:~:text=In%20London%2C%20a%20congestion%20management%20system%20lowered%20traffic%20volume%20to%20the%20mid%2D1980s%20levels>
- Swarco. (2024). *Swarco*. Retrieved from <https://www.swarco.com/stories/swarco-hub-connected-traffic>
- Synopsys. (2024). Retrieved from <https://www.synopsys.com/automotive/autonomous-driving-levels.html>
- Synopsys. (2024). *Synopsys*. Retrieved from <https://www.synopsys.com/automotive/what-is-adas.html>
- Tesla Motors. (2016). *Tesla*. Retrieved from <https://arstechnica.com/cars/2016/05/four-hundred-miles-with-teslas-autopilot-forced-me-to-trust-the-machine/>
- Times, T. A. (2016). Retrieved from <https://astanatimes.com/2016/04/rental-bike/>
- TodoMecanica. (2018). *TodoMecanica*. Retrieved from <https://www.todomecanica.com/noticias/20511-el-sistema-e-call-obligatorio-para-todos-los-vehiculos-nuevos-de-la-ue.html>
- Volkswagen. (2020). Retrieved from <https://www.volkswagen-newsroom.com/en/stories/car2x-in-the-new-golf-a-technological-milestone-5919>
- WHO. (2023). *WHO*. Retrieved from <https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023>
- Wirelesscommunication. (1998). Retrieved from <http://www.wirelesscommunication.nl/reference/chaptr01/dtmmsyst/dsrc/dsrc4.htm#:~:text=The%20DSRC%20Data%20Link%20Layer%20consists%20of%20the>

## 9. Annex

**[-- CONFIDENTIAL SECTION --]**

...

**[-- END OF CONFIDENTIAL SECTION --]**

## 10. Keywords

- 
- <sup>i</sup> C-ITS: Cooperative Intelligent Transportation Systems
  - <sup>ii</sup> V2X: Vehicle-to-Everything
  - <sup>iii</sup> C-V2X: Cellular Vehicle-to-Everything
  - <sup>iv</sup> REST: Representational State Transfer
  - <sup>v</sup> API: Application Programming Interface
  - <sup>vi</sup> MQTT: Message Queuing Telemetry Transport
  - <sup>vii</sup> IoT: Internet of Things
  - <sup>8</sup> V2V: Vehicle-to-Vehicle
  - <sup>9</sup> ITS JPO: Intelligent Transportation Systems Joint Program Office
  - <sup>10</sup> GPS: Global Positioning System
  - <sup>11</sup> GIS: Geographic Information Systems
  - <sup>12</sup> Gridlock: A specific type of traffic congestion where the entire network of intersecting streets is so congested that no vehicular movement is possible in any direction.
  - <sup>13</sup> ADAS: Advanced Driver Assistance Systems
  - <sup>14</sup> GLOSA: Green Light Optimized Speed Advisory
  - <sup>15</sup> ETA: Estimated Time of Arrival
  - <sup>16</sup> V2P: Vehicle-to-Pedestrian
  - <sup>17</sup> V2I: Vehicle-to-Infrastructure
  - <sup>18</sup> SAE: Society of Automotive Engineers
  - <sup>19</sup> DSRC: Dynamic Short-Range Communications
  - <sup>20</sup> WAVE: Wireless Access in Vehicular Environments
  - <sup>21</sup> RF: Radio Frequency
  - <sup>22</sup> OSI: Open Systems Interconnection
  - <sup>23</sup> LLC: Logical Link Control
  - <sup>24</sup> MAC: Medium Access Control
  - <sup>25</sup> ETSI: European Telecommunications Standards Institute
  - <sup>26</sup> FCC: Federal Communications Commission
  - <sup>27</sup> IEEE: Institute of Electrical and Electronics Engineers
  - <sup>28</sup> 3GPP: 3rd Generation Partnership Project: A worldwide standardization development organisation developing standards for different generations of mobile networks.
  - <sup>29</sup> V2N: Vehicle-to-Network
  - <sup>30</sup> 5G NR: (5G New Radio) A more capable version of 5G that has enhanced data rates and improved capacity.
  - <sup>31</sup> LTE: Long Term Evolution
  - <sup>32</sup> DF: Data Frame: It refers to the structured format used for encoding the data within a message.
  - <sup>33</sup> ASN.1: Abstract Syntax Notation One
  - <sup>34</sup> Technical Committee of the International Telecommunication Union
  - <sup>35</sup> UPER: Unaligned Packed Encoding Rules
  - <sup>36</sup> CDD: Common Data Dictionary
  - <sup>37</sup> ITS-S: Intelligent Transportation Systems Station
  - <sup>38</sup> B2B: Business to Business
  - <sup>39</sup> OEM: Original Equipment Manufacturer
  - <sup>40</sup> KPI: Key Performance Indicator
  - <sup>41</sup> WSL: Windows Subsystem for Linux