Efficient and affordable Zero Emission logistics

through **NEXT** generation **E**lectric **TRUCK**s

HORIZON Innovation Actions ¦ Project Number: 101056740

D3.3
Design of interoperable IoT concept and Interfaces for connected electric truck fleet

Co-funded by the European Union

NextETRUCK has received funding from the European Union Horizon Innovation Actions programme under grant agreement No. 101056740.

Dissemination level	Sensitive (SEN) - limited under the conditions of the Grant Agreement
Work package	WP3: System Architecture and TCO optimization: e-Powertrain and charging infrastructure
Deliverable number	D3.3. Report on modular and interoperable IoT design concept for connected electric truck fleet allowing predictive maintenance services
Version	V2.2
Submission date	28/12/2023

Quality Control

	Name	Organisation	Date
Peer review 1	Sophie Naylor	CENEX	13/12/2023
Peer review 2	Dai-Duong Tran	VUB	14/12/2023

Version history

Version	Date	Author	Summary of changes
2.0	13/12/2023	Leo Xenakis	First draft sent to reviewers
2.1	19/12/2023	Leo Xenakis	Draft amended as per feedback from reviewers
2.2	21/12/2023	Leo Xenakis	Final

Legal disclaimer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

Abbreviation	Meaning
TCO	Total Cost of Ownership
GSM	Global System for Mobile communication
IoT	Internet of Things
CAN	Controller Area Network
IVI	In-Vehicle Infotainment
WP	Work Package
DT	Digital Twin
API	Application Programming Interface
VM	Virtual Machine
xCU	(Generic) Control Unit, derived from ECU, Electronic Control Unit, but the E replaced with x to signify it could be any control unit
MQTT	Message Queuing Telemetry Transport is an ISO standard publish-subscribe based messaging protocol for use on top of the TCP/IP protocol
OEM	Original Equipment Manufacturer
FMU	Functional Mock-up Unit
GPU	Graphics Processing Unit

EXECUTIVE SUMMARY

The NextETRUCK project aims to develop a suite of technologies for the development, deployment, and operation of the next generation of medium duty all electric truck fleets. The project will demonstrate this on 3 separate trucks running in 3 locations around Europe where significant improvements in overall efficiency and TCO are expected versus the existing state of the art.

A major part of the NextETRUCK project aims to enable modular and interoperable IoT design concept, for cloud connected functionality at the fleet level. This allows the latest up-to-date information from the vehicles, and also from the cloud, to be utilized to the fullest to obtain the best efficiency, not just at a vehicle level, but also at the fleet level.

D3.3 will focus on the specification of this cloud connected functionality over Cellular/GSM (mobile phone) networks. This data can be used on short time horizons to aid in daily tasks such as routing, scheduling and energy optimizations during driving. Or it can be used over longer time horizons for predictive maintenance, vehicle lifetime prediction and overall cost calculations.

The task successfully identified appropriate software and hardware that is compatible with the planned vehicles. Architecture and functionality was also demonstrated in a prototype system that is capable of sending and receiving messages between a dummy vehicle and the cloud. A concept for the bidirectional communication of 50 CAN signals from a dummy vehicle (in this case a PC with a CAN card) and some code running in Microsoft's Azure cloud, was demonstrated at 1Hz for the whole round trip.

1 INTRODUCTION

1.1 About NextETRUCK

NextETRUCK is a 3-year Horizon Europe project that develops ZEV concepts tailored for regional medium freight haulage, running from 1 July 2022 until 31 December 2025.

The project aims at playing a pioneering role in the decarbonisation of vehicle fleets, demonstrating next-generation e-mobility concepts. It also contributes to the development of zero-emission vehicles and ecosystems that are holistic, innovative, affordable, competitive, and synergetic.

NextETRUCK is expected to build concepts tailored for regional medium freight haulage with at least a 10% increase in energy efficiency compared to existing highest-end benchmark electric vehicles. In addition, it shall prepare concept and infrastructure demonstrators for fast charging and offer new business models to increase end-user acceptance and foster the market uptake of the project solutions.

The project's consortium consists of 19 partners from 8 countries: The Netherlands, Belgium, Germany, Spain, Greece, Austria, Turkey, United Kingdom[1]. The project's coordinator is TNO (Netherlands Organization for Applied Scientific Research).

NextETRUCK shall conduct demonstrations in Istanbul, Barcelona, and UK (changed from Utrecht).

1.2 Purpose of the deliverable

In today's world, everything is connected. Nowadays, OEMs manage a huge variety of different devices from various tool suppliers and vendors, distributed at many different locations around the globe. All the while, high security and consistent data exchange across all devices, software, vehicles, etc. are required but difficult to ensure. This task will design and develop an interoperable and multiuse IoT system for different vehicle applications (i.e., delivery electric truck, refuse electric truck, …). This task designs the Hardware and Software to interlink the many, globally distributed devices of a company whilst keeping isolated networks unaffected.

WP3 has listed the following objectives, with the item relevant for Task 3.4, Deliverable 3.3 **highlighted in bold**:

[1] The UK participants in this project are co-funded by the UK.

The main objective of WP3 is to define the system architecture, design and overall TCO optimization including the functional distribution and the main parameters per vehicle component/powertrain and charging infrastructure.
• Development of generic co-design optimization tool/framework to optimize the vehicles systems including the E/E architectures in the use-cases and their interaction with charging infrastructure and to provide final design of the developed technologies with a TCO reduction up to 20%.
• Innovative HVAC system and cabin conceptual for the demo electric trucks with improvement in energy saving and increased driving range up to 15%.
• New vehicle thermal control and management system resulting in up to 15% efficiency improvement in the vehicle thermal management.
• Modular and interoperable IoT design concept and interfaces for connected electric truck fleet allowing predictive maintenance services for the fleet.
• Optimized and new functional safety concept of the electric trucks ensuring safe driving.

Table 1: Task split for Task 3.4, Deliverable 3.3

Role:	Who:
AVL (Austria) – Task Leader	Connectivity device and technology to enables highly secure, hardware protected connectivity, enhances intelligence at your devices with the Smart Hub and having full control over data exchange
Tecnalia	Support on V2x concept and development of telematic devices (HW) to enhance the connected fleet management by sending CAN data
OEMs (TEVVA, IRIZAR)	Will verify the design of developed IoT concept for their vehicles
DATIK	Design of the smart and interoperable IoT system and connectivity for modular vehicle architecture of IRIZAR use-case for several applications

1.3 Structure of the deliverable

The report is structured as follows:

- Section 1 : Introduction
- Section 2 : Overview
- Section 3 : Summary of all Cellular communications devices
- Section 4 : Hardware and Software options
- Section 5 : Vehicle and communications device CAN system assumptions
- Section 6 : Cellular/GSM communications system assumptions

- Section 7 : Cellular/GSM system management and Cloud assumptions
- Section 8 : Conclusion
- Section 9: References
- Section 10: Annex

1.4 Relationship to other work packages

Where there is a strong link to another work package, particularly those that are mentioned in this report, they are briefly listed below:
WP 2.4 : Details which DTs and other cloud components need to communicate with the physical vehicle. Was useful in determining the transfer rates and how many signals would need to use the link in this work package.
WP 7 : Details the offline logging capabilities for the on-board data logger. This work package may well have used the same hardware as for this purpose, so it was important to clarify if this was possible.

1.5 Main changes from previous version

The main changes of the present version of the deliverable compared with the previously delivered version are:

- v2.1 – All changes received from reviewers, were reviewed and incorporated. Annex added with compressed figures showing all the signals. Includes link to Excel also.

2 OVERVIEW

2.1 Overall System Architecture

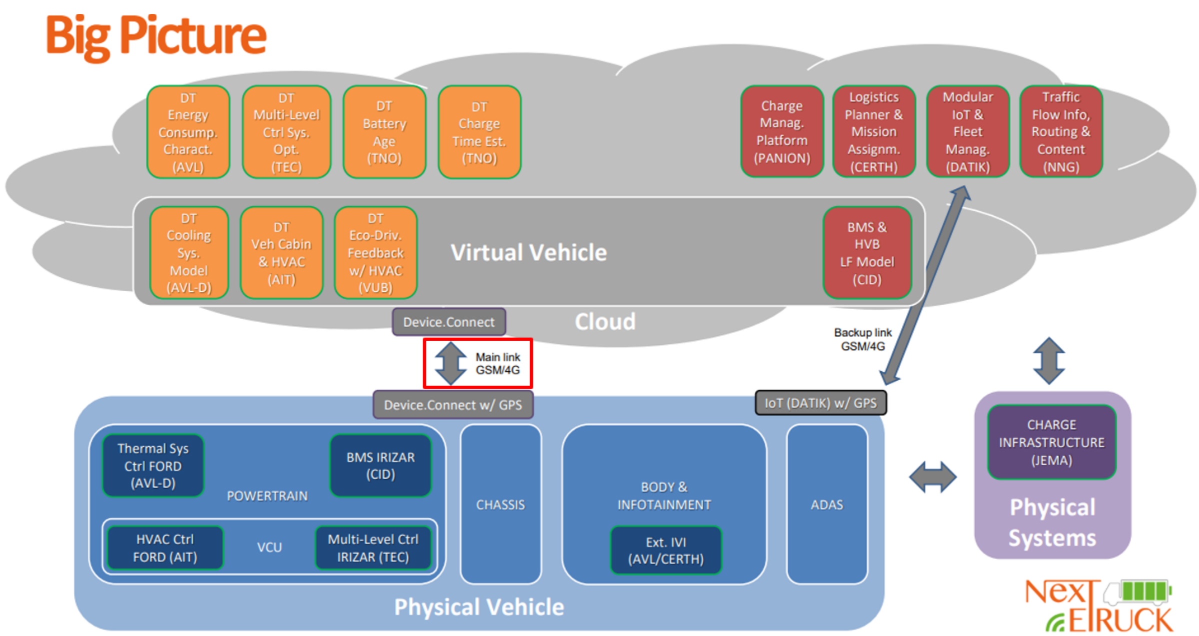

Figure 1: Overall System Architecture as defined in Task 2.4, highlighting the communications link that is the topic of this work package (red square marked Main Link)

The purpose of this work package is to ensure a secure wireless link for fast live communication between the vehicle and the cloud. There are of course other cellular links planned, but the focus of this work package will be the one used for bidirectional live CAN message transmission between the vehicle and the cloud.

2.2 Live CAN-Cloud Communication System Architecture

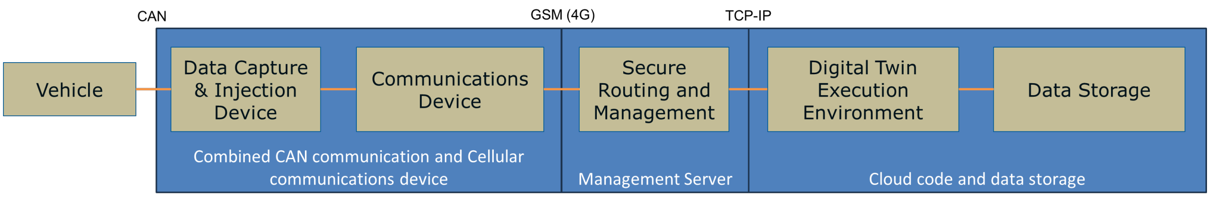

Figure 2: Live CAN-Cloud communication system architecture

The work package will define the requirements of the 6 parts of this system, suggest possible software and hardware solutions, assess their suitability, and present the decision on which hardware and software solutions were chosen for each OEM solution. This is required, as it is not necessarily the case that all the OEMs will use the same solution.

2.3 Overall System Targets

There are 4 main targets of the system. These were partially derived in earlier work packages, but did not automatically give the targets that were subsequently agreed upon.

Table 2: Overall system requirements for WP3.4

#	Requirement	Target	Units
1	Required CAN signals (up / down)	50 / 50	channels
2	Maximum round trip time	1	s
3	Security and safety assured	Encryption/Authentication	-
4	Vehicle side power supply voltage	12 or 24	V

An overview of the target assumptions and derivations are given below:

1. **50 up / 50 down CAN signals to be transmitted / received by the vehicle:**
   Task 2.4 specified that 36 signals needed to leave the vehicle and be transmitted to the cloud. 30 signals also needed to be received back from the cloud. Please see Annex for a list of these signals.
   a. It was agreed that if the system could transmit and receive 50 channels both ways (so 50 up and 50 down) that this would be a safe target. Including this margin was agreed with the other stakeholders, to ensure that the system could still function even if there was a 33% increase in the number of signals in both directions from our current assumptions.

2. **Round trip time under 1s under optimal conditions**
   It was also agreed with the partners that data transmission for a round trip would under no foreseeable circumstances need to be faster than 1Hz.
   a. Round trip here refers to this full sequence:
      i. Reading from the vehicle CAN-bus
      ii. Transmitting this data to the cloud
      iii. Performing calculations in the cloud to generate response signals
      iv. Receiving the calculated signals at the vehicle
      v. Writing those signals back to the vehicle CAN-bus
   b. It should also be made clear that if the complexity of the calculations that need to be performed is high, then this will of course delay this loop.
      i. It is assumed for the purposes of this work package that the calculations are very simple and will take negligible time.
      ii. The target also assumes that all parts of the system are in an optimal condition and do not assume a worst-case scenario. i.e., the cellular signal strength is good and no retransmissions are necessary, cloud resources are not congested, xCUs and CAN-buses are not overloaded etc.

iii. Overall latency of the transmission system was believed to be more critical than the calculation speed in the cloud, which if required, could always be made faster with more compute resource. Therefore an very simple dummy calculation was subsequently agreed as the baseline calculation for the upcoming demonstration.

3. **Secure, safe and robust transmission and calculation**
   It is assumed that the system must be secure and the safety of all users, including those of other road users, must be maintained.
   a. The data and calculations are assumed to be sensitive information and will be protected either via encryption, or authentication to ensure no unauthorized access.
      i. Access must of course be granted to those requiring it, and there may be many parties (e.g. operators of the fleet, charge system, logistics, OEMs etc.). Therefore, a management layer must exist to give granular access to only authorized persons, to access the required information, including read/write access rights.
      ii. Where the communication channel itself could relatively easily be compromised, then data encryption would be used to ensure the transmitted data can not be interpreted, even if intercepted.
   b. Safety is also critical, especially with the ability to write directly to the CAN-bus (which inherently has no security built-in, unless CAN-FD is supported). Therefore, the system must be designed to ensure that safety is not compromised.
      i. Firstly, direct ECU write access from the cloud was specifically agreed as not being in scope of this system, as this was deemed too risky.
      ii. Secondly, it was agreed that CAN-bus commands would go only to xCUs, for example, to change control modes within the xCU. No direct actuation commands to “dumb” actuators (that bypass on-board xCUs and bypass their safety algorithms) can be sent via the CAN-bus directly from the cloud. To illustrate this more clearly:
         1. Allowed: Thermal system control mode switch command sent from the cloud to the heating controller, which executes a heating mode switch. All individual actuator control stays with the controller.
         2. Not allowed: Direct actuation of a pump in the thermal system from a CAN command received from the cloud directly, bypassing the thermal controller.
      iii. Overall, this ensures that dangerous combinations of commands can not be injected into the system, as the xCUs are in control. As long as the correct safety analysis and validation has been performed on the controllers, it should not be possible to get any of the systems to enter a safety critical condition.

c. The final requirement is robustness. Transmission errors will occur in such a system, mostly stemming from signal dropouts in the cellular system. The system must ensure robustness to signal dropout, especially incomplete transmission of signals. Ultimately it must be able to function safely in the absence of any cloud communications.

4. **Vehicle side power supply voltage**
   a. Most of the automotive related auxiliary systems under consideration in this work package will operate at the standard 12 or 24V, common in vehicle electrical systems with combustion engines. There will undoubtedly be other sensors and equipment on board that will also require such voltages. Therefore, it is assumed that there will be 12 or 24V power available in these vehicles to power such equipment.

3 SUMMARY OF ALL CELLULAR COMMUNICATIONS DEVICES

3.1 Overview

As shown in Figure 1 in the previous section, we are concentrating on a single communications link in this work package. However, not all Cellular communications will use this link. There are in fact many components and sub-systems that need to communicate with the internet, and currently, the only practical way to do this is via Cellular, or GSM communications.

It was necessary to also identify other cellular links, mostly to ensure that they do not interfere with each other, but also to see if there may be scope for reducing the number of SIM cards, mobile data contracts and the associated hardware. Management of the data, software workflows and hardware, would also likely be facilitated by fewer devices and data streams.

3.2 Identifying all other cellular communications devices in the system

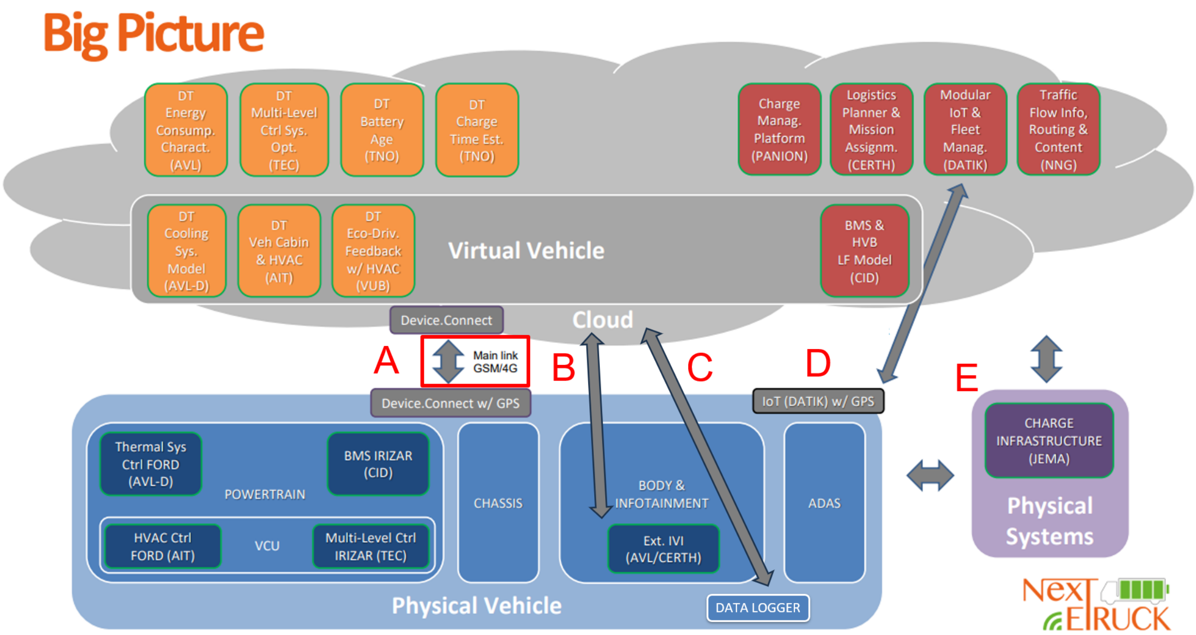

Figure 3: Overall system architecture showing all identified cellular communications links.

Below are listed the purpose and WP owner of each cellular link:

Table 3: All cellular communications devices in the system

	Link Description	Direction	Owner
A	Vehicle CAN-bus “Live” continuous CAN interface for DTs, and Data upload for Preventative Maintenance	Bi-directional Upload from vehicle	AVL-AT
B	Driver External Tablet (IVI) Interface for routing, driver aid etc	Bi-directional	CERTH
C	“Offline” logging of all channels (WP7) of interest for the project to on-board storage (CAN logging and possibly other protocols also supported) – Emptied once a day via cellular or WiFi	Upload from vehicle	CENEX
D	Fleet management interface	Upload from vehicle	DATIK
E	Charger management interface	Bi-directional	JEMA

3.3 Analysis of communication channels and identification of opportunities for optimization

The 5 cellular communications channels identified above were analyzed to determine if there were any commonalities that could be exploited.

3.3.1 A: Vehicle CAN-bus “Live” continuous CAN interface for DTs, and Data upload for Preventative Maintenance

This link is the main focus of this work package and the baseline to assess commonality. In order to possibly share the cellular device hardware with other communications links, we would need to meet certain criteria.

1. Communication via CAN - The communication with this device would have to happen over the main vehicle CAN-bus. This is because the only other communication channel that is an absolute requirement of this system (other than the cellular link) is via CAN and no other is guaranteed.
    a. To put it more simply, all data that needs to be transmitted via cellular, if routed through this device, must either already be on the main vehicle CAN-bus, or must be converted and placed onto the CAN-bus, as this is the only input to this device.
2. Compatibility with CAN - The data to be transmitted must be compatible with the CAN protocol.
3. Overloading the CAN - The data transmitted is not going to overwhelm the main vehicle CAN-bus.

The 3 criteria above will be referenced by number in the following discussion.

3.3.2 B: Driver External Tablet IVI

The Driver External Tablet IVI is a custom tablet that will take over duties from the in-built infotainment system. It is expected to perform bi-directional communications to send and receive data that is relevant to the driver from the cloud. The tablet interface will allow both display and interaction, to allow feedback input from the driver. It may also have a CAN interface for local communications (e.g. to pick up current gear, vehicle speed, pedal positions etc.), however the main interface will be between the driver and the cloud.

It is expected to take over from the in-built infotainment system with regards to the following.

- Routing/Navigation including updates and stop times for breaks and charging.
- Driver's aid – Instruction to the driver about recommended gear, accelerator and brake pedal actuation and possibly recommended vehicle speed. May also include eco-metrics such as how economically the driver is driving.
- Cabin thermals – Will display current cabin temperature setpoint and possibly allow override adjustments (adjustment may need to be performed on the physical vehicle controls).
- Other NextETRUCK project specific custom information that is deemed to be useful to the driver, or for development purposes.

In terms of viability in sharing the cellular link, there are however several problems that make this problematic for this device.

1. Communication via CAN – the signals to be sent in this case, such as route data are not already on the CAN, so needs conversion and injection into the CAN system
2. Compatibility with CAN – route data is unlikely to be compatible with the CAN, which is generally for realtime communications, not the bulk sending of data like a full route profile
3. Overloading the CAN – the data amounts may also overwhelm the CAN-bus, if transmission were even possible (see 2. above)

3.3.3 C: "Offline" data logger for all channels

The "Offline" data logger is designed to log everything that may be of interest on this project for later post-processing. It is read-only and similar to device A, with the main differences being the number of signals logged, that not all of them will originate from the CAN-bus, and that none of this data is required to be live. The emptying of this device will take place usually once a day preferably via WiFi, with the backup option of cellular.

There are several properties of this device that could make it a good fit for sharing the cellular link.

- The vehicle is not going to be driving when the upload of the data is happening, so the live link and this offline link will likely not need to use the same cellular link at the same time.
    - However it may be charging, so data associated with charging may be transmitted live, possibly leading to congestion on the cellular link.
- Both the live link and offline logger will likely be logging at least some of the same data. Transmitting this data twice seems like a waste of resources.
- Configuring the live and offline signals at the same time, though the same interface would make a lot of sense if possible, meaning potentially one less device and software workflow to configure and maintain.

In terms of viability in sharing the cellular link, there are however several problems that make this problematic for this device.

1. Communication via CAN – This data, the bulk of which likely originated on one of the many CAN-buses will need reinjecting into the CAN-bus. However, dumping all of the logged data into the CAN-bus before transmission will act as a huge bottleneck, massively increasing the time to transmit all the data. In fact, it will likely slow things down so much that there may well not be enough time to transmit all the data and the on-board storage would eventually fill up, as it can not be fully emptied in the limited vehicle downtime.
2. Compatibility with CAN – this data is unlikely to be compatible with the CAN, which is generally for real-time communications, not the bulk sending of data like a full day's worth of log data.
3. Overloading the CAN – the data volumes will almost certainly overwhelm the CAN-bus.

3.3.4 D: Fleet management interface

This device is designed to perform read-only data transmission to the Modular IoT and Fleet Management DT, specifically for fleet management. It also requires live data transmission.

In terms of viability in sharing the cellular link, there are however several problems that make this problematic for this device.

1. Communication via CAN – Not all this data will already be on the CAN, so needs injecting.
2. Compatibility with CAN – There may well be route data also in this uplink, which as discussed under device B, and this is likely to be incompatible with the CAN-bus
3. Overloading the CAN – The data amounts are likely to be quite modest, so may work, here if it wasn't for the data incompatibility (see 2. above)

3.3.5 E: Charger management interface (JEMA)

This device is designed to perform bidirectional communication with the charger and the charger management platform. This link exists on the physical charger, disconnected from the vehicle at all times other than when charging, so sharing a link with this device is not under consideration.

3.4 Summary of possible sharing of the cellular link with the main live CAN communication device

The analysis of the possibility of sharing the cellular link of device A with the other 4 devices has concluded that this is not viable. In production, it is likely that everything could be communized into just 2 cellular links (one for the vehicle and one for the charger), but in this project, there are just too many issues that are likely to arise if commonization is pursued.

This does not mean that the other links may not be able to share a cellular link (B and D look like good candidates).

The biggest issue that has not yet been discussed, is that each device has its own communications device built-in. If it was possible to extricate the communications device from each of these combined CAN and cellular communications devices, then it may have made the commonization task easier. If there was an external communications device, separate from the CAN device, the main issue we have encountered with device A would disappear, as there is more than enough bandwidth on the GSM side, and it is the CAN-bus side which is limiting access to the cellular link.

Having said this, device A is almost permanently active and must ensure all signals get through in real-time. Increasing data-throughput and interface capabilities is not just risking the integrity of the link and data, but could lead to extra safety concerns.

Therefore, for the purposes of this project it will be deemed out-of-scope for this cellular link to be shared with any other on-board device.

4 HARDWARE AND SOFTWARE OPTIONS

4.1 Overview

As previously shown in Figure 2, we will now list the possible options for software and hardware that constitute the 6 basic building blocks of the live CAN communication system.

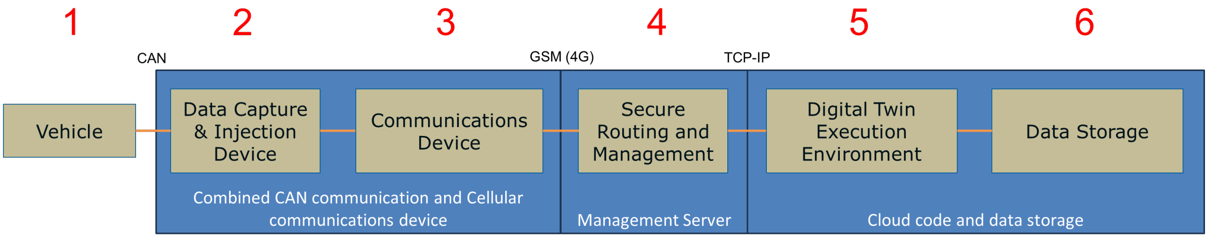

Figure 4: Numbered Live CAN-Cloud communication system architecture

The components are:

1. **Vehicle:** Includes the 3 OEM vehicles and also the dummy vehicle used to test the system.
2. **Data Capture & Injection Device:** Used to capture CAN messages and also inject them onto the vehicle CAN-bus.
3. **Communications Device:** Used to transmit and receive messages over GSM/Cellular.
4. **Secure Routing and Management:** A management server, ensuring secure message pass-through, routing and management of the configuration of the data capture and communication device.
5. **Digital Twin Execution Environment:** Cloud provider where the DTs are executed.
6. **Data Storage:** Cloud or local storage environment where the data gathered from the vehicles, as well as the data required for execution of the DTs is stored.

4.2 Overview of Hardware & Software options

In order to ascertain whether the hardware and software meet the requirements of the live CAN communications system, all possible hardware and software options were first gathered from the partners. This was in fact an iterative process of listing the possibilities, investigating whether they were compatible with the requirements or each OEM, and the overall toolchain, then adding further options and repeating the process. The final options list, regardless of whether they fully met the requirements, is listed below, so as to show the full extent of the software and hardware considered.

Each item is formatted as follows:
Hardware or Software name (Manufacturer) – Partner who entered this solution*

*Where the Manufacturer matches the Partner who entered the solution, only the Manufacturer will be listed

1. Vehicle options

1. Irizar Truck (Irizar)
2. Tevva Truck (Tevva)
3. Ford Truck (Ford)
4. Dummy vehicle: Workstation PC with CAN interface card (AVL)

2, 3 & 4. Data Capture & Injection Device, Communications Device & Secure Routing and Management options

Since only one of the solutions under consideration separated the hardware for 2 & 3, they are listed together, with the name of the associated 4. Secure Routing and Management system, listed alongside

a. Device.CONNECT & Device.CONNECT Framework (AVL)
b. Rebel LT CAN data logger & Streamlog Service (Influx Technology) - CENEX
c. Tecnalia Solution (Tecnalia)
d. DATIK DCB (DATIK)
e. VN1640A & Unknown cellular communications device (Vector) - Ford

5. Digital Twin Execution Environment options

a. Azure (Microsoft) - AVL
b. AWS (Amazon) - AVL
c. GCP (Google) - AVL

6. Data Storage Environment options

1. Azure (Microsoft) - AVL
2. AWS (Amazon) - AVL
3. GCP (Google) - AVL
4. Irizar On-premise - AVL
5. Tevva On-premise - AVL
6. Ford On-premise - AVL

5 VEHICLE AND COMMUNICATIONS DEVICE CAN SYSTEM ASSUMPTIONS

5.1 Overview

The vehicle CAN-bus specifications and those of the data capture and injection CAN device (hereafter referred to as the CAN device) must be compatible with each other, and also meet the requirements of the project. Therefore, a document was prepared that allowed easy gathering and comparison of the specifications of the vehicles, the CAN device options, and the required project specifications. Only those that met all the minimum criteria would be eligible for consideration in the project.

5.2 Overall CAN Requirements

The CAN protocol has evolved greatly since its inception (in 1986) and has also had many automotive-specific updates and enhancements. Therefore, it is imperative that we identify the protocols in use, and ensure compatibility. It is also common for modern vehicles to have multiple CAN-buses, so identifying the correct CAN-bus(es) that carry the required signals is also crucial.

It was agreed that all required channels for this project would reside on the main vehicle CAN-bus, so connection to only 1 CAN-bus is required. This reduces complexity and cost, as well as improving safety. Where this was potentially not directly possible (e.g. because a direct CAN connection was not possible from the requisite xCU), then passthrough of the required signals to and from the main vehicle CAN-bus was to be implemented.

5.3 Vehicle CAN-bus Specifications

The table below identifies all the main CAN specifications and protocols supported by each of the OEMs. The protocols supported refer to the main vehicle CAN only. The items in **bold** highlight the main vehicle CAN-bus and the protocols requiring support on this project.

Table 4: Main vehicle side CAN specifications

	Ford	TEVVA	IRIZAR
CAN1 Speed (Main vehicle CAN)	**500 Kbps**	**250 Kbps**	**250 Kpbs**
CAN2 Speed (CAN subnet)		500 Kbps	500 Kbps
CAN3 Speed (CAN subnet)			1 Mbps
CAN4 Speed (CAN subnet)			
# CAN channels	1	2	4

CAN Protocol support on main vehicle CAN			
ISO 11898 / 15765 (Classical CAN)	**Yes**	**Yes**	**Yes**
SAE J1939	**Yes**	**Yes**	**?**
OBD2	Yes	Yes	Yes
CANopen	?	Yes	?
CAN FD	No	No	?
CCP	?	?	?
XCP	?	?	?
LIN	Yes	Yes	?
UDS	Yes	Yes	?
Miscellaneous CAN requirements			
Remote xCU flash[1]	No	No	No

[1]Remote xCU flash refers to the ability to flash new controller software remotely, without a physical presence at the vehicle.

5.4 CAN Device Specifications

The table below identifies all the main CAN specifications and protocols supported by each of the proposed CAN devices. The items in **bold** highlight the main requirements and protocols requiring support on this project.

The Technalia solution is not featured in this list, as it was dropped at an early stage, due to a decision made together with Irizar, to use the AVL solution.

Table 5: Main CAN device specifications

Proposer of hardware	**AVL**	**CENEX**	**DATIK**	**Ford**
Device Name	AVL: Device. CONNECT	Influx: Rebel LT	DATIK: DCB	Vector: VN1640A
Max CAN data rate	1 Mbps	1 Mbps	250 Kbps	2 Mbps
Max CAN channels	1	2	2	4
CAN Protocol support				
ISO 11898 / 15765 (Classical CAN)	**Yes**	**Yes**	**No**	**Yes**
SAE J1939	**Yes**	**Yes**	**Yes**	**Yes**
OBD2	Yes	Yes	No	Yes
CANopen	?	Yes	No	?
CAN FD	No	No	No	Yes
CCP	Yes	Yes	No	?
XCP	Yes	Yes	No	?
LIN	No	No	No	Yes
UDS	Yes	Yes	No	Yes
Miscellaneous CAN requirements				
Remote xCU flash[1]	Yes	No	No	?
CAN-bus signal request [2]	?	No	?	?
Write back to CAN-bus [3]	**Yes**	**No**	**Yes (Static values)**	**Yes**
Remote config for device [4]	**Yes**	**Yes**	**Yes**	**?**
Data preprocessing [5]	**Yes**	**Yes**	**?**	**?**

[1] Remote xCU flash: Flash new controller software remotely, without a physical presence at the vehicle.
[2] CAN bus signal request: Request extra channels on the CAN-bus.
[3] Write base to CAN-bus: Write signals to the CAN-bus (not just read).
[4] Remote config for logger: Change/Update the configuration of the CAN device without a physical presence at the vehicle.
[5] Data preprocessing: Pre-processing of the CAN data before transmission, this could be used to perform simple calculations, also in the time domain, such as resampling.

Data preprocessing may not seem to provide so much advantage in this use-case, as all the processing can be done in the cloud. However, calculations in the time domain in particular, such as resampling, can greatly reduce the amount of data transmitted via the cellular network. This of course is only possible, if the full data resolution on the CAN-bus, is not required for the calculations in the cloud.

5.5 Combined summary of CAN Specifications

The table below identifies all the main CAN protocols supported by each of the OEMs under "Vehicle side requirements", and the equivalent specifications for the CAN device under "Proposed Hardware Specifications". The Target column identifies which are the target criteria, with colors indicating which are met.

Table 6: Comparison of vehicle CAN specifications and proposed CAN device specifications

	Vehicle side requirements			**Proposed Hardware Specifications**				**Target**
Manufacturer	FORD	TEVVA	IRIZAR	AVL	CENEX	DATIK	FORD	
Device Name				AVL: Device. CONNECT	Influx: Rebel LT	DATIK: DCB	Vector: VN1640A	
CAN1 Speed (Main vehicle CAN)	**500 Kbps**	**250 Kbps**	**250 Kbps**	**1 Mbps**	**1 Mbps**	**250 Kbps**	**2 Mbps**	**500 Kbps**
CAN2 Speed (CAN subnet)		500 Kbps	?					
CAN3 Speed (CAN subnet)			?					
CAN4 Speed (CAN subnet)			?					
Max # CAN channels	**1**	**2**	**4**	**1**	**2**	**2**	**4**	**1**
CAN Protocol support on main vehicle CAN								
ISO 11898 / 15765 (Classical CAN)	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**
SAE J1939	**Yes**	**Yes**	**?**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
OBD2	Yes	Yes	Yes	Yes	Yes	No	Yes	
CANopen	?	Yes	?	?	Yes	No	?	
CAN FD	No	No	?	No	No	No	Yes	
CCP	?	?	?	Yes	Yes	No	?	
XCP	?	?	?	Yes	Yes	No	?	
LIN	Yes	Yes	?	No	No	No	Yes	
UDS	Yes	Yes	?	Yes	Yes	No	Yes	
Miscellaneous CAN requirements								
Remote xCU flash[1]	No	No	No	Yes	No	No	?	No
CAN-bus signal request [2]				?	No	?	?	No

Write back to CAN-bus [3]				**Yes**	**No**	**Yes (Static values)**	**Yes**	**Yes**
Remote config for device [4]				**Yes**	**Yes**	**Yes**	**?**	**Yes**
Data preprocessing [5]				Yes	?	?	?	

[1] Remote xCU flash: Flash new controller software remotely, without a physical presence at the vehicle.
[2] CAN bus signal request: Request extra channels on the CAN-bus.
[3] Write base to CAN-bus: Write signals to the CAN-bus (not just read).
[4] Remote config for logger: Change/Update the configuration of the logging device without a physical presence at the vehicle.
[5] Data preprocessing: Pre-processing of the CAN data before transmission, this could be used to perform simple calculations, also in the time domain, such as resampling.

The comparison of the specifications on the vehicle side and the proposed CAN device, show that there are only 2 devices that meet the overall specifications of the whole project. AVL Device.CONNECT and possibly the Vector VN1640A.

The only question mark related to the Vector device, is the ability to configure it remotely. The Vector device is the only one in the list that does not combine the communications device with the CAN-device. Therefore, it needs to interface with a yet unspecified communications device, in order to enable wireless data transmission and potentially remote configuration also. Azure IoT hub was raised by Ford as a possibility, but too late for proper consideration. If Vector supply their own such device, remote configuration seems quite likely to be possible, however there is no indication on their website of this capability, but this does not mean that it does not exist. It could be argued that for this project, it could potentially also be acceptable if remote configuration was not possible, as long as Ford are willing to make configuration updates to the CAN device manually in this case. Clarification will be sought in later work packages, since there is a desire from Ford side to consider the Vector hardware together with the Azure IoT hub for this project.

The 2 parties that were required to give their feedback on which CAN device to use (TEVVA and IRIZAR) have stated that they will use AVL Device.CONNECT. Ford was asked to confirm their position and they did not commit to one of the devices, but both are compatible with this project, so should be suitable (once the remote configuration capability is clarified from Ford side for the Vector hardware).

6 CELLULAR/GSM COMMUNICATIONS SYSTEM ASSUMPTIONS

6.1 Overview

The cellular / GSM communications system assumptions and some other specifications related to GPS will be covered in this section. In most cases, the proposed hardware in the previous section, related to the CAN system, is integrated with the communications device. The only exception is the Ford proposed Vector VN1640A device, which they wanted to use in combination with the Azure IoT Hub. Since it was not possible to consider the Azure IoT hub due to late proposal of the device for consideration, separate stand-alone devices for communication are not considered.

6.2 Cellular / GSM communications system assumptions

Table 7: Cellular / GSM and GPS system assumptions

	AVL	CENEX	DATIK	Target
Device Name	AVL: Device. CONNECT	Influx: Rebel LT	DATIK: DCB	
Cellular generation	3G (4G LTE)	3G (4G LTE)	3G (4G LTE)	Min 4G LTE
4G bands supported	Europe, US	Europe	Europe	Europe
4G LTE category	CAT1	CAT1	CAT1 and CAT4	CAT 1
SIM card data quota	Contract dependent	4GB	2GB	Preferably Unlimited
Network Up/Down speed	5 Mbit/s / 10 Mbit/s	5 Mbit/s / 10 Mbit/s	50 Mbit/s / 150 Mbit/s	5 Mbit/s / 10 Mbit/s
GPS system	GPS	GPS	GPS	GPS
Dropout assumption (buffer size)		64 GB buffer		No buffering required
Extra encryption of communications	Yes	No (relies on mobile encryption)	Yes	Yes
SIM type	Nano SIM	?	Micro SIM	Any

Data speed and latency requirements of this project were easily achievable by supporting 4G LTE, as the 4G LTE specification far exceeded what was possible via CAN and was required on this project.

6.2.1 Data rates

Cellular 4G, otherwise known as LTE, has several categories as defined by the GSM standard to which it belongs. CAT 1 (category 1) is the lowest and must be supported in full to claim 4G LTE compatibility. This is also the most robust, as it is the fallback when even faster category speeds are not achievable.

The maximum required data rate for this project is 500 Kbps, which is the speed of the main vehicle CAN bus on the Ford vehicle. Standard CAN bus speeds top out at 1Mbps and even the next generation of CAN, CAN FD, has a maximum data rate of 5 Mbps. All of these are within the 5 Mbps limit of 4G CAT 1, so should not cause any bottlenecks in this project, as even if we saturate the 500 Kbps CAN network with signals just for transmission to the cloud, it would still be only 10% of the upload bandwidth of CAT 1 LTE (5 Mbps).

3G also a potential fallback, and is quite capable of supporting the required data rates, topping out at 3 Mbps. However, it should not be relied upon, as in many areas in Europe, it is either already very slow or is being phased out completely [1].

6.2.2 Latency

Similarly, latency is required to be very low for cellular systems (typically 15-50ms for 4G LTE), since otherwise there would be many complaints from customers regarding lag on their voice calls. Although voice calls are given priority when network quality is low, it is still very fast and unlikely to add significantly to the overall time required for the round trip communication.

The target round trip time for communications to and from the vehicle is 1s (at the target 1 Hz). This communication requires two journeys across the cellular network, so even if we take the longest expected latency, of 50ms for each journey, this will only lead to a total latency of 100ms. This is also just 10% of the total 1s available, so is unlikely to be an issue.

6.2.3 Security

4G communications are of course protected by encryption, however it has become increasingly clear that there are fundamental flaws that have been exposed with 4G that can never be fixed [2]. Therefore, to ensure ultimate security, it is advisable to further encrypt the data before transmission over the cellular network.

6.2.4 Other considerations

GSM communications are carried out in different parts of the electromagnetic spectrum, each new generation of GSM, such as 4G / LTE, will occupy a certain frequency range and this will further be subdivided into bands. Different parts of the world use slightly different bands due to many reasons.

The reason this is mentioned is that 4G bands must be supported by the cellular communication hardware, and not all hardware supports all of the bands. This is further complicated by some cellular networks being limited to certain bands, within those allowed in that region. However, since we are essentially in the Europe region for most of this project (apart from Ford who are partially in Asia) and many of the manufacturers of the hardware are based in Europe also, the frequency bands supported by the hardware will work anywhere in Europe. For AVL Device.CONNECT, it is confirmed to support US and some Asian bands also. For the other hardware manufacturers, I was not able to get this data, as it is often not published in their specifications.

As a side-note, there is a lot of overlap in the bands that are supported in each region and sometimes it is just a case of finding a cellular provider in the region that supports the bands supported by the hardware, so this is unlikely to be an issue outside certain parts of Asia and Africa where there may be fewer providers and therefore fewer bands to choose from.

6.3 Cellular / GSM communications system summary

The overall conclusion from an analysis of the GSM standard and the hardware on offer, is that if the hardware supports 4G LTE and some form of extra encryption, then it is more than capable of supporting the project.

Both AVL Device.CONNECT and Datik DCB are up for consideration if looking only at the GSM communication. However, DATIKs system does not meet the specifications of the CAN side of the system, so can not be considered for this project.

7 CELLULAR/GSM SYSTEM MANAGEMENT AND CLOUD ASSUMPTIONS

7.1 Cloud systems overview

The focus of this section is to identify the requirements of:

1. The management system for the cellular / GSM system that allows the configuration and routing of the signals from the vehicle to the cloud.
2. The calculation environment that hosts the DTs and any extra models or calculations that need to be performed in the cloud.
3. The storage requirements of this system only (excluding other cloud data storage requirements of the project).

7.2 Cloud systems architecture

7.2.1 Management system architecture

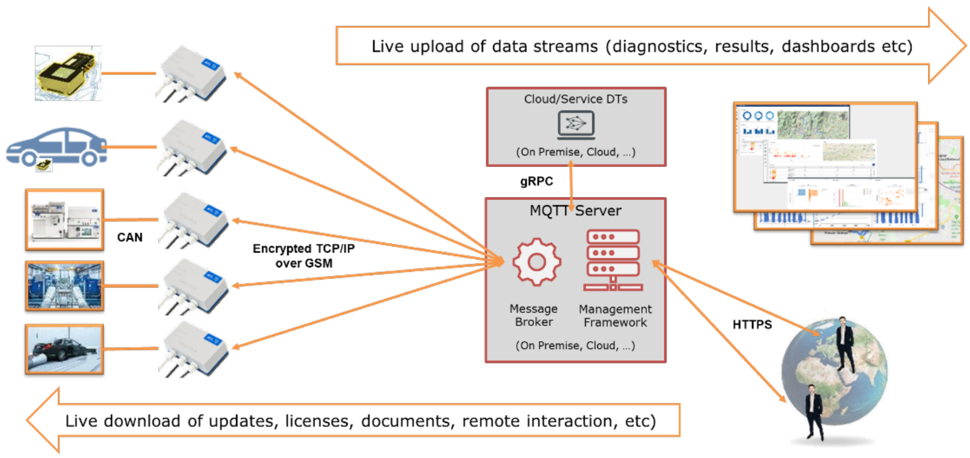

Figure 5: Cloud system architecture (focus on management system)

Figure 5 shows the full cloud system architecture, with the main focus being the management system. The calculation environment is contained within the red circle. All major components of the system are shown, including the links between them and the protocols used. Since the only viable system for this project is the AVL system, the layout is tailored to show the components utilizing the AVL Device.CONNECT Framework (the management server for Device.CONNECT).

7.2.2 Calculation system architecture

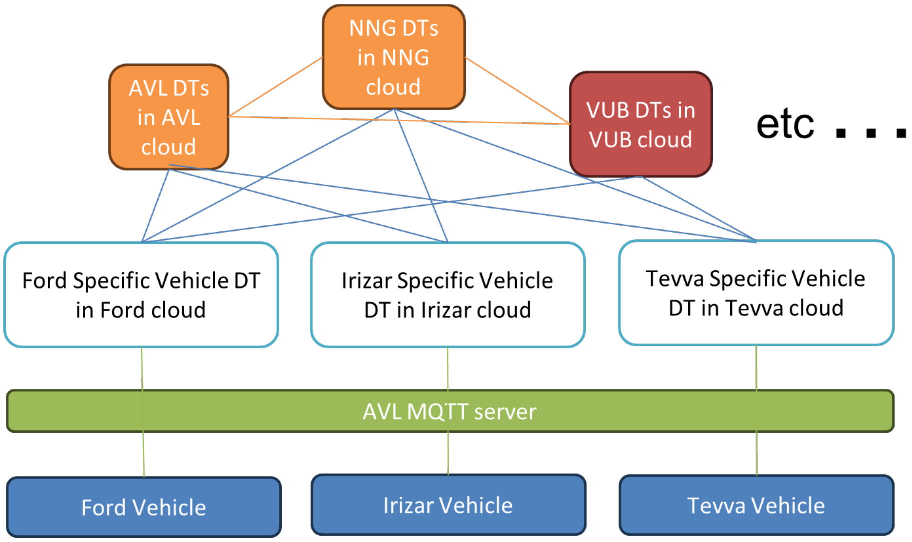

Figure 6: Cloud system architecture (focus on calculation system and specifics of this project)

Figure 6 shows the cloud system architecture with a focus on the calculation system, showing the expected final layout of the calculation system, that will likely rely on multiple cloud environments. These environments may be local clouds or even run by the same cloud provider (e.g. Microsoft's Azure), but be operated by different users. Due to privacy reasons, or because these DTs are services that are designed to be accessed via dedicated APIs, the system being designed in this project should assume that DTs could be scattered across multiple different cloud providers and even be delivered as a service via custom APIs. There will undoubtedly be latency and security issues that must be overcome, but this will ensure that there is maximum flexibility to connect to any cloud system both during the project and for the commercial rollout in future.

7.3 Cloud systems summary

In the next work package (WP4), a prototype of this system is to be demonstrated. It is a little ahead of schedule and is partially deployed internally at AVL. There are some simplifications to the system in this first step and they are detailed below.

Figure 7 below shows this simplified architecture which is the first step to showing a full working system that is able to demonstrate the required 1Hz round trip communication between the cloud and a dummy vehicle.

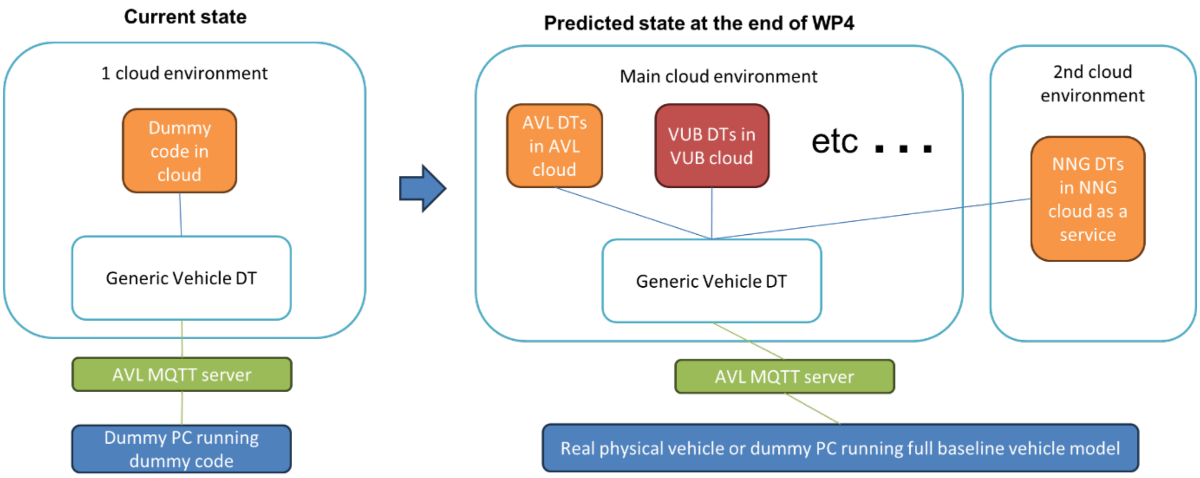

Figure 7: Cloud system architecture (focus on calculation system, showing progression from current state, to predicted state at the end of WP4)

The details below highlight the current state of the full system, as of the end of WP 3.4, and the predicted state of the system at the end of WP 4.

Current state of the cloud environments (end WP 3.4):

1. On the cloud side, simplified to just a single connection to a dummy PC, but with the full complexity on the management side, as shown in Figure 5.
2. The dummy PC will not run the full vehicle model, but be designed to generate 50 CAN signals on one channel and to accept 50 signals on the same channel from the cloud, via a USB CAN device (Peak P-CAN USB).
3. The cloud calculation code is also simplified to just do a simple arithmetic calculation on the incoming signals and to send them back to the dummy vehicle. It will be packaged in an FMU and placed in a docker container to mimic the true final system architecture.

Predicted state of the cloud environments (end WP 4):

1. Still a single dummy vehicle, but now, either a real physical vehicle or a dummy PC running the full baseline vehicle model.
2. Most DTs of the project running in a single main cloud environment.
3. At least one DT running also in a separate cloud environment as a service (probably the NNG DT).

7.4 Cloud systems VM assumptions for power and storage requirements, including cost

At this stage it is very difficult to assess exactly how much computing resource is required for the project. However, an attempt has been made to estimate these. If they turn out to be incorrect, the beauty of such cloud computing services is that it is usually possible to pay a little more, to have access to the correct resources, so it is unlikely to affect progression of the project.

1. The number of VMs (Virtual Machines) required for this project is likely to be, the number of cloud environments required for the services + 3 for OEM specific calculation systems + 1 management server for the communications.

    a. It is as yet unknown how many service DTs the project will have.
    b. 3 OEM calculation systems - 1 VM per OEM, as they probably want to run their own customized vehicle/DTs in their own cloud for security and access management reasons.
    c. 1 communications management server - The management server for Device.CONNECT (the Device.CONNECT Framework). Only one is required to interface to thousands of vehicles.

2. Which cloud environment these are deployed in.

    a. The communications management server is MQTT based for AVL Device.CONNECT Framework.
    b. The main calculation cloud system to be deployed will be assumed to be Azure, as this is the preferred system as identified by many of the project partners. However, it is also assumed that other cloud providers will be part of the project and will be incorporated as and when required.
    c. The storage systems may well reside inside of the main calculation systems, however if the amount of data is excessive, this may also be transferred to local OEM servers on premise.

3. The compute cost for these VMs is unlikely to be very high, as the calculation load is likely on the scale of a standard workstation class PC without any GPU acceleration.

4. The storage cost is also likely to be low and may even come for free as part of the package when selecting the required compute machine.

    a. This is because you will always need a working memory area for any simulation, and therefore you never rent just compute with no storage.
    b. I have estimated that storage space of 10TB is more than enough for the calculations and also to store all of the CAN system data over the 6 months of the project.
    c. Please note that this 10TB prediction does not include the data storage requirements of the “offline” storage of all logged channels via the separate logging system detailed in WP7 and briefly mentioned in section 4.2 of this report, in table 2, under item C.

8 CONCLUSION

8.1 System architecture assumptions and deployment timeline

The figure below represents the progression from the current state of the proposed system at the end of WP 3.4, through the end of WP4, then to the final state for deployment with the real vehicles.

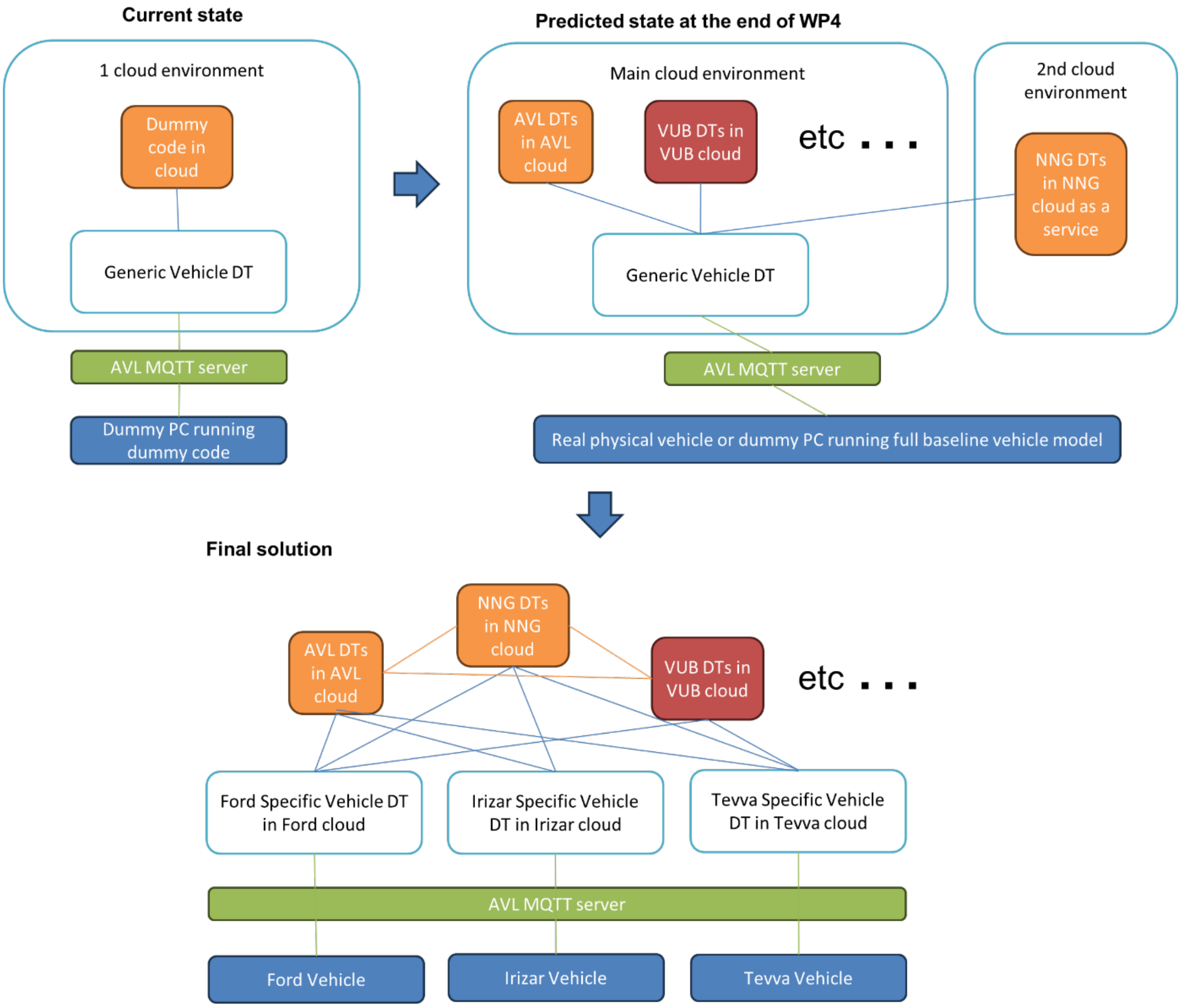

Figure 8: Evolution of the proposed system from current state to final solution

8.2 Hardware and management system overview and responsible parties

Current assumed responsible party for the hardware and management parts of the live CAN to cloud communications system were the same for all the categories below, so are grouped together for ease

- Hardware/Software provision responsible
- Hardware/Software setup responsible
- Hardware/Software maintenance responsible
- Hardware/Software cost responsible

Listed below in the following format are the responsible parties and the associated hardware / software.

Format:
Name of Hardware Device or Software
(Manufacturer of Hardware /Software)
[Responsible]

Table 8: Responsible parties for provision, setup, maintenance and costs, for the hardware and management parts of the live CAN to cloud communications system

Vehicle	Data Capture & Injection device	Comms device	Secure Routing and Management
PC (dummy vehicle) [AVL]	Device.CONNECT (AVL) [AVL]	Device.CONNECT (AVL) [AVL]	Device.CONNECT Framework (AVL) [AVL]
Ford Vehicle [Ford]	VN1640A (Vector) [Ford]	Azure IoT Hub (Microsoft) [Ford]	Azure IoT Hub (Microsoft) [Ford]
TEVVA Vehicle [TEVVA]	Device.CONNECT (AVL) [AVL]	Device.CONNECT (AVL) [AVL]	Device.CONNECT Framework (AVL) [AVL]
IRIZAR Vehicle [IRIZAR]	Device.CONNECT (AVL) [AVL]	Device.CONNECT (AVL) [AVL]	Device.CONNECT Framework (AVL) [AVL]

The items marked in Red were not able to be fully assessed, but there was a strong desire from Ford to propose this. The fallback position in WP4 will be to use the AVL solution and it will be the responsibility of Ford to assess whether their proposed system meets the requirements of the project and of their vehicle and use-case.

8.3 Cloud calculation and storage system overview and responsible parties

As alluded to in earlier sections, getting confirmation from the partners of who is going to be responsible for the setup of their cloud environments and especially the costs were a contentious issue.

The document used to capture the current thoughts of the OEMs is below. It seems that at least for the early stages of the project, all the OEMs are agreeing that they need to store their own data locally and use either AWS or Azure to host their DTs so as to ensure they had a working environment and also the ability to store data locally on their own servers. Later on there is significant uncertainty, particularly around the hosting costs and who might be responsible for setting up and maintaining such systems (as shown by the blanks in the table below).

Table 9: Responsible parties for provision, setup, maintenance and costs, for the non-cloud parts of the live CAN to cloud communications system

Vehicle	Resp.	Digital Twin Execution Cloud Environment	Data Storage Environment
Software provision responsible (who owns or hires the cloud environment)			
AVL PC	AVL	AZURE (Microsoft)	AZURE (Microsoft)
FORD	FORD	AZURE (Microsoft)	Ford own cloud
TEVVA	TEVVA	AZURE (Microsoft) or AWS (Amazon)	TEVVA own cloud
IRIZAR	IRIZAR	AZURE (Microsoft)	IRIZAR own cloud
Setup responsible			
AVL PC	AVL	AZURE (Microsoft)	AZURE (Microsoft)
FORD	FORD		
TEVVA	TEVVA		
IRIZAR	IRIZAR		
Maintenance / Execution responsible			
AVL PC	AVL	AZURE (Microsoft)	AZURE (Microsoft)
FORD	FORD		
TEVVA	TEVVA	AZURE (Microsoft) or AWS (Amazon)	TEVVA own cloud
IRIZAR	IRIZAR		
Cost responsible			
AVL PC	AVL	AZURE (Microsoft)	AZURE (Microsoft)
FORD			
TEVVA	TEVVA	AZURE (Microsoft) or AWS (Amazon)	TEVVA own cloud
IRIZAR			

For all, non-OEM specific code, there seemed to be some pressure from the OEMs for AVL to host this, but this was not accepted by AVL. AVL did however agree to host and shoulder the cost of hosting the full demo toolchain and give access to the OEMs and partners to familiarize themselves with the system, however it was made clear that no serious simulation work could be performed on the demo system and for those purpose. AVL would help to duplicate such a system at the OEMs, but any further hosting, execution and storage costs would have to be borne by other parties.

Ultimately, it is likely that the all the responsibilities for cloud computation and data storage would fall to the individual OEMs for their OEM specific models and DTs. Partners who ran their own services would pay for the hosting of those. The initial setup and maintenance would be a joint effort with those who are developing key parts of the system being responsible for either performing the initial setup and maintenance themselves, or having the responsibility to pass on the appropriate knowledge via training, so it could be performed by the individual cloud owners. The hosting and the costs of doing of the cloud environments would always stay with the owners of the clouds.

8.4 Future Work

There are some technical areas that ideally would have benefited from clarification in this WP, but this was not possible. We believe it does not affect the outcomes of this WP, but they are listed here for completeness

1. No OEM is fully able to clarify if all the required signals will be on the main vehicle CAN-bus. This is mostly because they are not able to clarify the final vehicle architecture, or on which xCU much of the custom code will run (this could be external prototyping ECUs, the main ECU or some sub xCUs). This of course makes it very difficult to know exactly which CAN lines will carry what data. For now, it is assumed that all required CAN signals will be routed to the main vehicle CAN if a direct connection to the originating xCU is not possible.
2. Ford to clarify if they will use AVL Device.CONNECT. They are not listed in this work package as a contributor, but it would be beneficial to get the same input from Ford as the other 2 OEMs.
    a. If not using Device.CONNECT, they will need to propose a communications device compatible with their proposed CAN data logging and injection device. This device will of course need to meet all the requirements of the project, as laid out in this document. They proposed Azure IoT hub (https://azure.microsoft.com/en-us/products/iot-hub) at the 11th hour and unfortunately there was not enough time to explore the capabilities.
3. An attempt was made to clarify who is doing what exactly regarding the supply, deployment, maintenance, execution and costs of the various components of this system, but as regards the final state, and especially the cloud environments, it was not possible at this stage to clarify all the responsibilities. This is not strictly part of this WP though, but still something that needs clarification sooner rather than later.

4. There was not much appetite in this work package to consider the requirements for preventative maintenance. It also appears that much of the information required to gather the information for preventative maintenance is not specified in the input-output table from WP 2.4, likely requiring many more signals to be added to the existing set, increasing processing and data transmission costs. It was therefore decided with the agreement of the partners, that full consideration of preventative maintenance would be removed from the work package and considered in a later work package, but only if interest again grew in this area. I suspect that the preventative maintenance service, if implemented, is unlikely to need any of the signals from the vehicle in real-time. This is because maintenance decisions only need to be made on a longer time-scale, for example, on a daily basis, not by the second. Combined with the potential need to transmit much more data live, it could therefore be argued that this requirement is best handled in WP 7, where they will deal with the full “offline” data logs of all logged signals from the vehicles. It should be noted that this has not been explicitly discussed with the appropriate work package leaders in WP 7, nor has particular interest in the preventative maintenance topic gained any more prominence since the initial meetings.

8.5 Next Steps

In terms of meeting the technical goals of the project, AVL are responsible for the initial baseline simulated vehicle, and the live CAN-cloud communication system. Both can be demonstrated in the new year running on Azure. We will host this demo system for others to access and also help with the required training to set up the appropriate workflow at each of the OEMs to be able to duplicate this setup. This should allow for a solid backup plan, even if others then want to transfer their setup away from Azure to another cloud provider, or even into their own local clouds.

The same can be said of the Device.CONNECT hardware and framework. Irizar and Tevva have agreed to use Device.CONNECT and we will host the management server, so there was not much resistance to this proposal. Ford are looking at alternatives, but if it does not materialize then the base setup with Device.CONNECT is always available for them to fall back on.

I believe this outlines a solid plan to ensure that all the vehicles can be equipped with the required capability, and the full cloud workflow would be demonstrated with at least 2 of the OEMs. If a different path is desired, then the required specifications of the hardware and software systems are detailed in this report and if they do not work out, then a proven fallback plan will be available.

9 REFERENCES

1. Susie S (2020). 2G / 3G Network Shutdown Status and Challenges. Retrieved on 01/12/2023 from https://www.smartviser.com/post/2g-3gnetworkshutdown

2. Syed Rafiul Hussain, Omar Chowdhury, Shagufta Mehnaz, Elisa Bertino (February 2018)
   LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE
   Conference paper
   https://par.nsf.gov/servlets/purl/10055689
   Conference presentation
   https://www.ndss-symposium.org/wp-content/uploads/2018/03/NDSS2018_02A-3_Hussain_Slides.pdf
   presented at
   Network and Distributed Systems Security (NDSS) Symposium 2018 18-21

10 ANNEX

All variables as identified by in Task 2.4 that are being transmitted or received by the vehicle via the Live CAN-Cloud communications system. Please refer to NextETRUCK - Deliverable_D2.4_v1.2_Final - Annexes 1-2-3 - Task4.3 Update v01.xlsx on the SharePoint for more details.

Signals/Parameters Specification						Physical Vehicle									Cloud,		Physical Systems		Virtual Vehicle						Cloud Modules								
Resp. (OUT)	Input by	Name	Unit	Data Type	Period (ms)	Physical Vehicle	Thermal System Controller FORD (AVL-D)	HVAC Controller FORD (AIT)	Multi-Level Ctrl IRIZAR (TEC)	BMS IRIZAR (CID)	Ext. IVI (CERTH)	IN	OUT	PAR	IN	OUT	Charging Infrastruct. (JEMA)	...	Virtual Vehicle	DT Vehicle Cabin & HVAC (AIT)	DT Cooling System Model (AVL-D)	DT Eco-Driving Feedback w/ HVAC (VUB)	BMS & HVB LF Model (CID)	...	Charge Management Platform (PANION)	Logistics Planner and Mission Assignment (CERTH)	Modular IoT and Fleet Management (DATIK)	Traffic Flow Info, Routing & Content (NNG)	DT Energy Consumption Charact. (AVL)	DT Multi-Level Control System Optimization (TEC)	DT Battery Age (TNO)	DT Charge Time Estimation (TNO)	...
AVL	AVL	sECC_OptEnergyCons_kW	kW	float						IN		1	0	0											IN	IN	IN	IN	OUT	IN			
CID	CID	vBMS_BatCellAvgTemp_C	°C	float						IN		1	0	0									OUT										
CID	CID	vBMS_BatCurr_A	A	float						IN		1	0	0									OUT										
CID	CID	vBMS_BatCurr_V	V	float						IN		1	0	0									OUT										
CID	CID	vBMS_ChrgDhrgSts	-	uint						IN		1	0	0									OUT										
AVL-D	CID	vCSM_BatCoolFlow_lpmin	l/min	float						IN		1	0	0							OUT		IN										
AVL-D	AVL-D	vCSM_BatInCoolTemp_C	°C	float						IN		1	0	0							OUT		IN										
AVL-D	AVL-D	vCSM_PWTInCoolTemp_C	°C	float			IN					1	0	0							OUT												
AVL	AVL	vECC_BatDeltaSoC_pcs	%/s	float						IN		1	0	0												IN	IN	IN	OUT	IN			
AVL	AVL	vECC_BatDeltaSoH_pcs	%/s	float						IN		1	0	0												IN	IN	IN	OUT	IN			
CERTH	CERTH	vLP_X_MissCtrlData	-	struct							IN	1	0	0												OUT							
TEC	TEC	vMLC_OptRampEffort_kNps	kN/s	float					IN			1	0	0																OUT			
TEC	TEC	vMLC_OptThermStratBMS	-	uint					IN			1	0	0																OUT			
TEC	TEC	vMLC_OptThermStratHVAC	-	uint					IN			1	0	0																OUT			
TEC	TEC	vMLC_OptThermStratVTMS	-	uint					IN			1	0	0																OUT			
TEC	TEC	vMLC_VehSpdSpRedRatio_pc	%	float					IN			1	0	0																OUT			
NNG	AVL	vTFI_ForecastAirTemp_C	°C	float			IN					1	0	0														OUT	IN				
NNG	CERTH	vTFI_ItinerData_X	-	struct							IN	1	0	0														OUT					
NNG	CERTH	vTFI_NearbyRoadSeg_X	-	struct							IN	1	0	0														OUT					
NNG	CERTH	vTFI_NewRoute_X	-	struct							IN	1	0	0														OUT					
NNG	NNG	vTFI_NewRouteOffPath_X	-	struct							IN	1	0	0														OUT					
NNG	AVL-D	vTFI_PredHeatFlow_X	W	struct			IN					1	0	0														OUT					
NNG	AVL-D	vTFI_PredVehPwr_W	W	float vector			IN					1	0	0														OUT					
NNG	AVL-D	vTFI_PredVehSpd_Kmph	km/h	float vector			IN					1	0	0														OUT					
NNG	AVL-D	vTFI_PredWeather_X	-	struct			IN					1	0	0														OUT					
NNG	CERTH	vTFI_RoadSegId_X	-	struct							IN	1	0	0														OUT					
NNG	CERTH	vTFI_RoadSegPar_X	-	struct							IN	1	0	0														OUT					
NNG	CERTH	vTFI_StreetLst_X	-	struct							IN	1	0	0														OUT					
AVL	AVL	vVEH_AirDragForce_N	N	float					IN			1	0	0					OUT										(IN)				
AVL	AVL	vVEH_RollResForce_N	N	float					IN			1	0	0					OUT										(IN)				

Figure 9: 30 Signals identified as being received via Live CAN-Cloud communication by the vehicle

Signals/Parameters Specification						Physical Vehicle									Cloud, Virt &		Physical Systems		Virtual Vehicle						Cloud Modules								
Resp. (OUT)	Input by	Name	Unit	Data Type	Period (ms)	Physical Vehicle	Thermal System Controller FORD (AVL-D)	HVAC Controller FORD (AIT)	Multi-Level Ctrl IRIZAR (TEC)	BMS IRIZAR (CID)	Ext. IVI (CERTH)	IN	OUT	PAR	IN	OUT	Charging Infrastruct. (JEMA)	...	Virtual Vehicle	DT Vehicle Cabin & HVAC (AIT)	DT Cooling System Model (AVL-D)	DT Eco-Driving Feedback w/ HVAC (VUB)	BMS & HVB LF Model (CID)	...	Charge Management Platform (PANION)	Logistics Planner and Mission Assignment (CERTH)	Modular IoT and Fleet Management (DATIK)	Traffic Flow Info, Routing & Content (NNG)	DT Energy Consumption Charact. (AVL)	DT Multi-Level Control System Optimization (TEC)	DT Battery Age (TNO)	DT Charge Time Estimation (TNO)	...
CID	CID	sBMS_BatEstSoC_pc	%	float		(OUT)				OUT	IN	1	1	0	4	0	IN										IN			IN		IN	
CID	TNO	sBMS_BatVolt_V	V	float		(OUT)				OUT		0	1	0	3	0	IN														IN	IN	
CID	TNO	sBMS_MaxChrgPow_kW	kW	float		(OUT)				OUT		0	1	0	2	0	IN															IN	
CERTH	NNG	sIVI_ActPosData_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_ItinerDataReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	NNG	sIVI_MissPathReq	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_NearbyRoadSegReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_NewRouteReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_RoadSegIdReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_RoadSegParReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_StreetLstReq_X	-	struct							OUT	0	1	0	1	0												IN					
CERTH	CERTH	sIVI_VehInfo&NavStat_X	-	struct		OUT					OUT	0	2	0	1	0										IN							
AVL-D	AVL-D	sTMS_PredComp1SpdDmd_PWM	PWM	float vector		(OUT)	OUT					0	1	0	1	0					IN												
AVL-D	AVL-D	sTMS_PredEXValv1PosDmd_pc	%	float vector		(OUT)	OUT					0	1	0	1	0					IN												
AVL-D	AVL-D	sTMS_PredFan1SpdDmd_PWM	PWM	float vector		(OUT)	OUT					0	1	0	1	0					IN												
AVL-D	AVL-D	sTMS_PredPmp1SpdDmd_PWM	PWM	float vector		(OUT)	OUT					0	1	0	1	0					IN												
AVL-D	AVL-D	sTMS_PredValv1PosDmd_pc	%	float vector		(OUT)	OUT					0	1	0	1	0					IN												
OEM	DATIK	sVEH_AccPedKickDwnBtn	-	unint		OUT						0	1	0	2	0						IN					IN						
OEM	DATIK	sVEH_AccPedPos_pc	%	float		OUT						0	1	0	2	0						IN					IN						
OEM	AVL-D	sVEH_AirAmbHum_pc	%	float		OUT	IN					1	1	0	1	0					IN												
OEM	AVL	sVEH_BatChrgReq	-	uint		OUT						0	1	0	1	0									IN								
OEM	CID	sVEH_BatCurrDmd_A	A	float		OUT						0	1	0	1	0							IN										
OEM	AVL-D	sVEH_BatInCoolTemp_C	°C	float		OUT	IN			IN		2	1	0	1	0							IN										
OEM	CID	sVEH_BatPwrDmd_W	W	float		OUT				IN		1	1	0	1	0							IN										
OEM	DATIK	sVEH_BrkPedPos_pc	%	float		OUT						0	1	0	1	0											IN						
OEM	TNO	sVEH_ExtAirAmbTemp_C	°C	float		OUT	IN					1	1	0	6	0				IN	IN	IN					IN			IN		IN	
OEM	TNO	sVEH_Timestamp_ts	ts	string		OUT						0	1	0	4	0						IN								IN	IN	IN	
OEM	AVL	sVEH_TotPwrCabAux_kW	kW	float		OUT						0	1	0	1	0													IN				
OEM	AVL	sVEH_TotPwrTrailerAux_kW	kW	float		OUT						0	1	0	1	0													IN				
OEM	DATIK	sVEH_VehEngSpd_rpm	rpm	uint		OUT						0	1	0	1	0											IN						
OEM	AVL	sVEH_VehGear	-	uint		OUT						0	1	0	1	0						IN							(IN)				
OEM	PANION	sVEH_VehMass_Kg	kg	float		OUT						0	1	0	1	0									IN								
OEM	DATIK	sVEH_VehOdo_km	km	float		OUT						0	1	0	1	0											IN						
OEM	AVL-D	sVEH_VehSpd_kmph	km/h	float		OUT	IN					1	1	0	2	0					IN						IN						
OEM	AVL	sVEH_VehTirePressList_bar	bar	loat vector		OUT						0	1	0	1	0													IN				
OEM	DATIK	sVEH_WhlTrq_Nm	Nm	float		OUT						0	1	0	1	0											IN						

Figure 10: 36 Signals identified as being transmitted via Live CAN-Cloud communication from the vehicle