ENSEMBLE

EUROPEAN COMMISSION

HORIZON 2020
H2020-ART-2016-2017/H2020-ART-2017-Two-Stages
GA No. 769115

ENSEMBLE

ENabling **S**af**E** **M**ulti-**B**rand p**L**atooning for **E**urope

Deliverable No.	D3.2	
Deliverable Title	Operational (non-automotive grade) ITS-G5 communication, supporting the multi-brand truck	
Dissemination level	Public	
Written By	Jan de Jongh, Jacco van de Sluis, TNO	07-07-19
Checked by	T3.1 Task lead: Antoine Schmeitz, TNO WP3 Leader and project technical coordinator: Dehlia Willemsen, TNO	11-07-19

Approved by	Coordinator: Marika Hoedemaeker, TNO	16-07-19
Status	Final, approved by EC	05-08-2022

Please refer to this document as:

de Jongh, J. and van de Sluis, H.J.D. (2019). *Operational (non automotive grade) ITS-G5 communication, supporting the multi-brand truck.* D3.2 of H2020 project ENSEMBLE, (www.platooningensemble.eu)

Acknowledgement:

Jeffrey Panneman, Daan Ravesteijn, Antoine Schmeitz, Ramon de Souza Schwartz, Geert Verhaeg, Harry Wedemeijer, Dehlia Willemsen, Bastiaan Wissingh, Wieger IJntema, TNO.

TABLE OF CONTENTS

Revision history

Version	Date	Author	Summary of changes	Status
0.9	07/07/2019	Jan de Jongh, Jacco van de Sluis (TNO)	Created	Prepared [for review/approval]
1.0	16/07/2019	Jan de Jongh (TNO)	Review comments: - Authors - Task 3.1 Lead - WP3 Lead - Project Coordinator	Final [for EC submission]

FIGURES

1. EXECUTIVE SUMMARY

1.1. Context and Need of a Multi Brand Platooning Project

Context

Platooning technology has made significant advances in the last decade, but to achieve the next step towards deployment of truck platooning, an integral multi-brand approach is required. Aiming for Europe-wide deployment of platooning, 'multi-brand' solutions are paramount. It is the ambition of ENSEMBLE to realise pre-standards for interoperability between trucks, platoons and logistics solution providers, to speed up actual market pick-up of (sub)system development and implementation and to enable harmonisation of legal frameworks in the member states.

Project scope

The main goal of the ENSEMBLE project is to pave the way for the adoption of multi-brand truck platooning in Europe to improve fuel economy, traffic safety and throughput. This will be demonstrated by driving up to seven differently branded trucks in one (or more) platoon(s) under real world traffic conditions across national borders. During the years, the project goals are:

- Year 1: setting the specifications and developing a reference design with acceptance criteria
- Year 2: implementing this reference design on the OEM own trucks as well as perform impact assessments with several criteria
- Year 3: focus on testing the multi-brand platoons on test tracks and international public roads

The technical results will be evaluated against the initial requirements. Also, the impact on fuel consumption, drivers and other road users will be established. In the end, all activities within the project aim to accelerate the deployment of multi-brand truck platooning in Europe.

1.2. Abstract of this Deliverable

Following the specifications for multi-brand truck-platooning outlined in Year 1 of the ENSEMBLE project, the V2X part of the reference design and implementation is described in this document. The V2X part consists of an ITS-G5 communication unit running a modified Linux™ operating system, ETSI GeoNetworking and BTP (Basic Transport Protocol). Additionally a test tool ('ENSEMBLE V2X Test Tool') is described that has been designed and realized for testing the communication unit (in particular against other implementations).

2. INTRODUCTION

In this Chapter, first the reference design specification and its top-level decomposition are introduced. Next, the scope for the V2X Sub-System and its interfaces are defined. Finally, an outline of the remaining chapters of this document is given.

2.1. Specification of the Reference Design

2.1.1. Input Documents

The following documents serve as (direct) inputs to the specification of the reference design and implementation:

- ENSEMBLE Deliverable D2.2, V1 *Platooning use-cases, scenario definition and Platooning Levels*, Final version 19-12-2018 (pending EC approval).
- ENSEMBLE Deliverable D2.8, *Platooning protocol definition and Communication strategy*, Final version 12-12-2018 (pending EC approval).
- ENSEMBLE Deliverable D2.9, *Security framework*, Final version (pending EC approval) 11-06.2019.

ENSEMBLE Deliverable D2.2 introduces a system overview (e.g., in terms of operational, tactical, strategic and service layers), platoon levels and use cases. In particular, the use cases serve as important input for identification of (state-changing) required information flows and interactions between manoeuvring, control and communication. The deliverable constrains the scope of the project to its defined Platooning Level A (simply stated: Longitudinal Automation only with a time gap of at least 0.8s @ 80 km/h, and at most seven trucks in a single platoon).

ENSEMBLE Deliverable D2.8 presents the communication strategy (following ETSI TC ITS ENs, mostly) and ENSEMBLE-specific platooning protocol definition in terms of messages and message sequences. In particular, the document provides an ASN.1 description of the platooning messages (Protocol Data Units, or PDUs). Deliverable D2.8 was submitted in December 2018.

ENSEMBLE Deliverable D2.9 describes the ENSEMBLE v2x security framework. Unlike many other ITS protocols and services, intra-platoon *confidentiality* plays a dominant role, next to the - in ITS - more commons security functions like *integrity*. In order to realize the required level of confidentiality, *symmetric-key encryption* is deployed in ENSEMBLE (at the platoon level), next to per-message integrity (through signing and certificate checking) and pseudonym changes (for privacy).

Deliverable D2.9 was submitted for approval to the EC in June 2019. It was decided to use the platooning protocol to also implement *Key Management* (*key updates)*.

After the publication of D2.2, D2.8 and D2.9, new insights required changes to the original protocol definition reported in the deliverables. These changes can be found on SharePoint in the T2.3 folders.

2.1.2. Other Input Sources

Apart from the project-specific input documents referenced above, a number of other sources direct the design of the Reference System:

- ETSI (and other standardization bodies) documents referred to directly or indirectly in the project deliverables. Examples include reference to ETSI TS 103097 v1.3.1. for security (certificates, secured messages) formats.

- Documentation on previous work and projects, in particular on V2X system organization. For instance, in many communication-based systems, including in ITS, there is a natural interface boundary between ‘Transport Layer’ (Layer 4 of the ISO/OSI Reference Model) and higher layers. In ENSEMBLE, the transport protocol of the day is BTP (Basic Transport Protocol), so it makes sense to design a communication device that supports the ETSI protocol stack up to BTP, but *not* beyond because of the divergence in higher layers due to the wide range of applications to be supported.

- Existing hardware and software used in previous projects. This motivates the choices for PC Engines™ APU (Accelerated Processing Unit) boards, Linux™ and (for instance) the *btpsap* software described in later chapters.

3. TOP-LEVEL LAYER DECOMPOSITION OF THE REFERENCE SYSTEM

3.1. Introduction

Somewhat extending the layered organization of ENSEMBLE as introduced in D2.2, the following decomposition of the reference system is introduced:

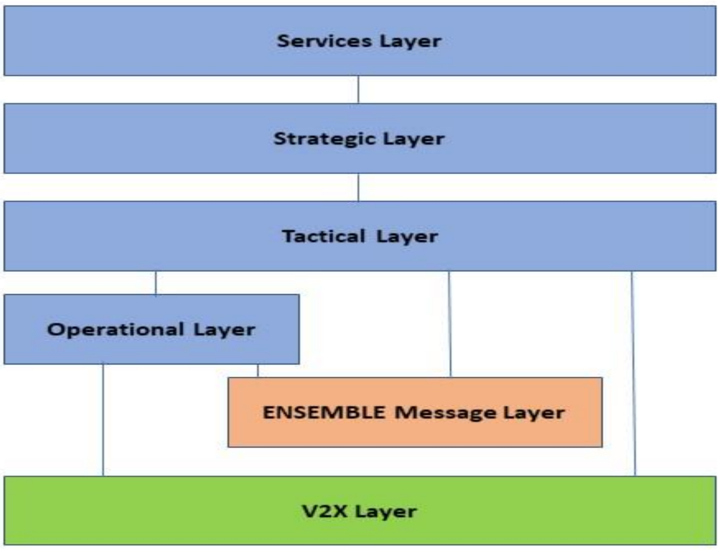


Figure 1: Decomposition of the ENSEMBLE Reference System.

Compared to the figure in D2.2, which decomposes the ENSEMBLE application into multiple interacting layers, this decomposition *adds* two more layers:

- The **ENSEMBLE Message Layer** takes (***only***) care of encoding and decoding (ASN.1-specified) ENSEMBLE-specific messages. It is used by the Operational Layer and by the Tactical Layer for interfacing with the V2X Layer with ENSEMBLE-specific messages. The main motivations for splitting these functions into a separate layer are:

  - *Both* the tactical *and* operational layers require the message encoding and decoding functions (as is also clear from the figure). This makes it difficult or at least unnatural to include message encoding and decoding into either one of them.

  - The (ENSEMBLE) ASN.1 message definitions continue to be subject to discussions and changes at the present time. By isolating the encoding and decoding functions, it is expected that the system design is more robust to future changes in the ASN.1 definitions.

  - Almost always, the decoding and encoding ASN.1 specified messages require the purchase and use of third-party ASN.1 tooling. By using a 'thin' ENSEMBLE Message Layer, the effects of switching between ASN.1 tooling vendors is easier to deal with.

- The **V2X Layer** takes care of interfacing to a medium through transmission and reception of messages using ITS-G5[1] and GeoNetworking/BTP in an *ENSEMBLE-independent way*.

  - This decision is in line with current practices in which the *communication functions* are (can be) implemented separately from applications using them. Another way of saying is, that the V2X Layer is application agnostic and can therefore be used by many applications simultaneously.

  - Since the V2X requirements (in view of their scoping as described above) are far more stable than the application (ENSEMBLE; platooning; truck platooning) specifications, the V2X Layer can be split-off as a separate design and implementation activity.

  - The definition of the V2X Layer is in line with commercially available On-Board Units like from Cohda Wireless/NXP MK IV/V as well as with research-oriented approaches like TNO's APU-based OBU.

3.2. ENSEMBLE Reference System Overview

The ENSEMBLE Reference System consists of the following parts:

- The Tactical Layer design (of the reference system) described in D3.1. (And, roughly speaking, its implementation is in D3.3.) The 3.1 component is beyond the scope of the current document, but included for proper scoping between D3.1 and D3.2.

- The V2X Sub-System design and realization described in D3.2 (the current document).

[1] Cellular communications is out of scope for this document. It is left open whether or not cellular communications in ENSEMBLE will be part of the V2X Layer or of a separate functional block, but it is not described in the current document either way.

- The ENSEMBLE Message Library described (to the extent useful) in D3.2.
- The ENSEMBLE V2X Test Tool described in D3.2 (the current document).

In the figure below, the scope of the current deliverable D3.2 is shown. Essentially this constitutes (small) parts of the operational and tactical layers, and the complete ENSEMBLE Message Library and V2X Layer.

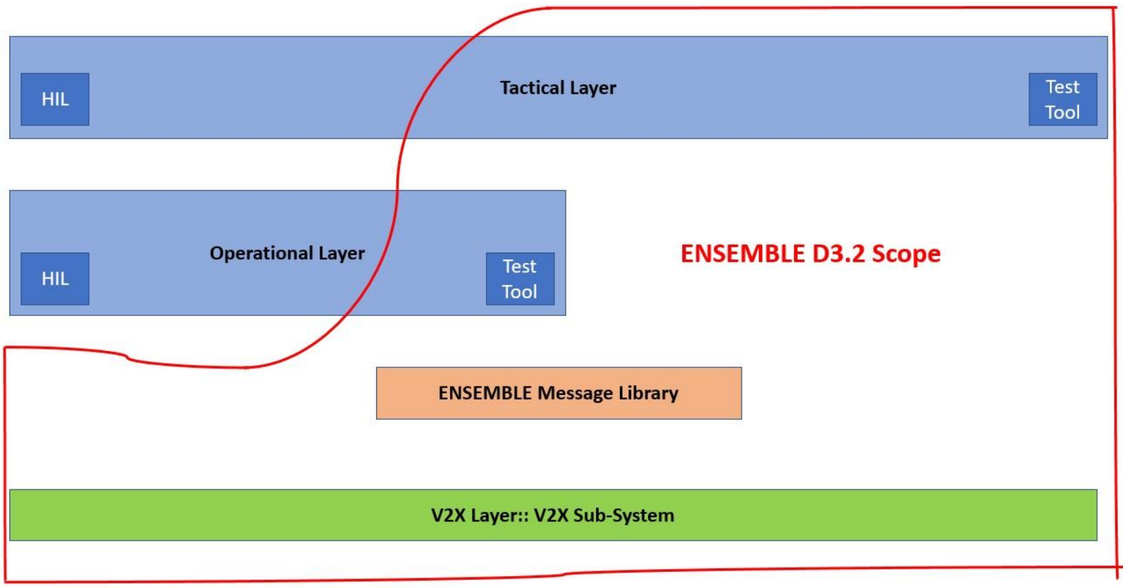


Figure 2: Scope of ENSEMBLE D3.2.

In the next two sections, the V2X Sub-System and the V2X Test Tool are described in more detail, respectively. Due to its (relative) simplicity, the ENSEMBLE Message Library is not described in more detail in the sequel. In the end, this layer merely encodes and decodes messages formatted according to the ENSEMBLE (ASN.1) specifications. Moreover, due to (ASN.1 compiler) license restrictions, the Message Library cannot be distributed as a general-purpose library, but only bundled in an application.

4. ENSEMBLE V2X SUB-SYSTEM IMPLEMENTATION

4.1. Introduction

The V2X Sub-System (in the ENSEMBLE Reference Implementation) features communication services based upon ETSI ITS-G5/GeoNetworking/BTP as well as Positioning and Time services. In this Chapter its architecture and main interfaces are described.

4.2. Overview

The V2X Sub-System consists of:

- An APU-based embedded platform with SSD and PCIe slots;
- A modified Linux kernel;
- A GPS-based PoTi (Positioning and Time) sub-system;
- BtpSap: A Java™ based implementation of ETSI GeoNetworking, Basic Transport Protocol (BTP) and security (signing and verification).

The V2X Sub-System hardware is depicted in Figure 3 and its architecture is shown in Figure 4.

Figure 3: Hardware of the V2X Sub-System (of the ENSEMBLE Reference Implementation).

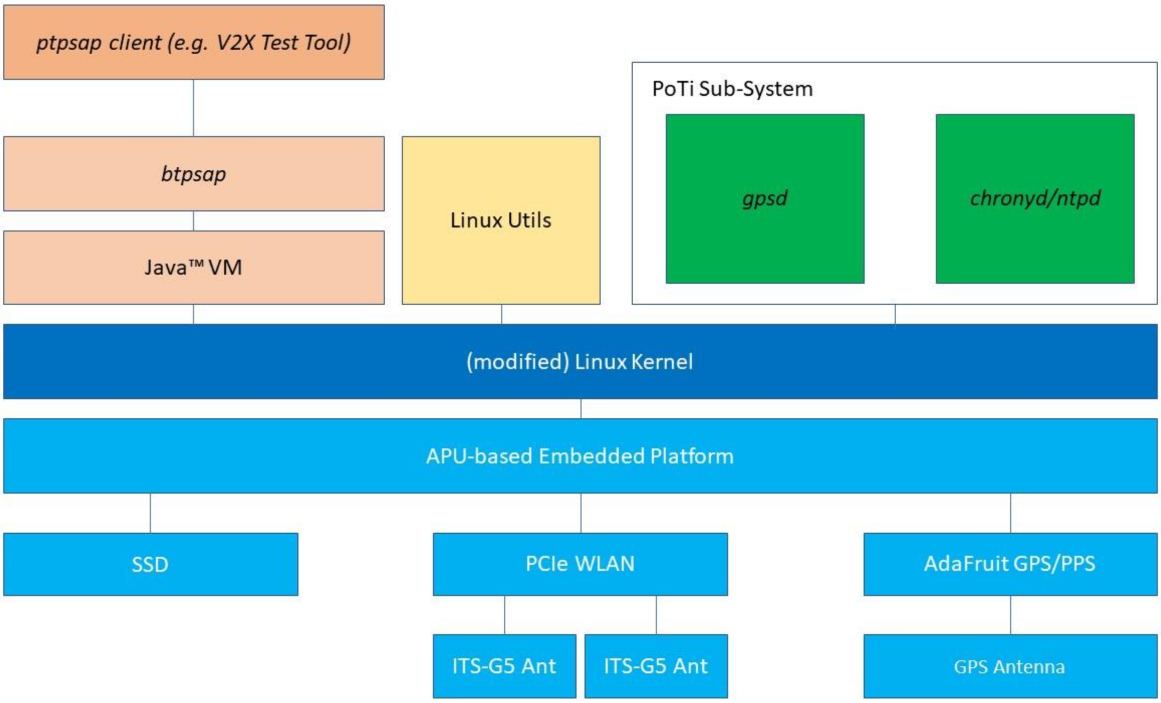


Figure 4: Architecture of the V2X Sub-System (of the ENSEMBLE Reference Implementation).

In the next sections, the individual hardware and software components as shown in Figure 4 are described in more detail.

4.3. Hardware

The V2X Sub-System consists of an APU-based hardware platform from PC Engines™. Although many alternative and equally suitable boards exist, this APU is chosen for reasons of special requirements (like availability of a Solod-State Disk, SSD, General-Purpose I/O, GPIO, and PCIe slots), low cost and previous experience.

Within the APU enclosures, several additional hardware components are added:

- A suitable GPS/PPS receiver (AdaFruit™) connected to one of the (configured) internal UART (Universal Asynchronous Receiver/Transmitter), “serial”, interfaces present on the GPIO (General Purpose Input/Output) pins, completed with pigtail and SMA (SubMiniature version A) mount.
- An SSD (solid-state drive).
- An Atheros™-based WLAN interface with (dual, for spatial diversity) pigtails and R-SMA mounts.

4.4. Linux Kernel

The V2X system is equipped with a modified Linux kernel. The modifications specifically address support for ITS-G5 (or, more accurate, the OCB mode of IEEE 802.11, also known as IEEE 802.11p). The required modifications to the kernel (in fact, to the ath9k driver and related user-space utilities) were implemented by the Industrial Informatics Research Center of Prague University (CTU-IIG), and are available from *https://github.com/CTU-IIG/802.11p-linux*.

Note that certain additional changes to the CTU-IIG Linux kernel configuration are applied in order to support (all of) the APU hardware.

4.5. PoTi Sub-System

The V2X Sub-System is equipped with GPS-based Positioning and Time (PoTi). At its heart is the more or less Linux-standard GPS Daemon, or *gpsd*. The GPS Daemon takes care of interpreting position and time updates from the AdaFruit™ GPS Receiver, the latter of which is connected to one of the UART interfaces of the APU. This itself provides the system with position and time estimates that are sufficient for (plain) GeoNetworking operations.

However, in this approach, without additional measures, the time estimates are only useful as “timestamps” of the position estimates from the receiver. In other words, one knows with high accuracy the time at which the position was estimated, but there is a relatively high and unpredictable latency between obtaining the estimate at the receiver and its reception in the Linux kernel. This means the time estimates *cannot* be used for synchronization of the system time.

In order to circumvent aforementioned problem, a special signal available at the receiver, the so-called Pulse-Per-Second (PPS) signal, is monitored by one of the GPIO pins on the APU, and its edges trigger the relevant kernel drivers. This way, a very high precision is obtained in synchronization of the system time with GPS. The high precision is required for instance for CAM time stamping and for certain security functions. Once a so-called 3D-Fix has been obtained by the GPS Receiver, the (Linux) system time will be synchronized with the GPS time, either through *ntpd* or through *chronyd*, depending on the deployment.

The Position and Time estimates are maintained by *gpsd* and are available to applications either running *on* the APU, or connected *to* it. The estimated upper bound on the deviation between GPS time and Linux system time is better than 100 μs. (Without PPS, the estimated accuracy is on the order of 100 ms). Note, however, that the current configuration of the PoTi subsystem uses a 1 Hz update rate from the GPS receiver; insufficient for the high-frequency positioning requirements in ENSEMBLE. The V2X Sub-System implemented by TNO (as is) *cannot* meet the ENSEMBLE requirements on positioning. It is only suitable for time synchronization and low-frequency position updates (as required for GeoNetworking, for instance). In other words, it meets the requirements of a communication device, but *not* of a high-accuracy sensor. In most practical cases, this implies the need for a separate, high accuracy, high update-rate GPS (like RTK-GPS based solutions) next to the GPS used in the PoTi sub-system.

4.6. BtpSap: Implementation of GeoNetworking and BTP

The *btpsap* 'daemon' is a Java™ based application implementing most of ETSI GeoNetworking (GN) and BTP and implementing a BTP SAP (Service Access Point) interface over TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). The GN and BTP implementations are derived from Alex Voronov's *geonetworking* repository, *https://github.com/alexvoronov/geonetworking*, and were extended with (TNO proprietary) ETSI security functions. The TCP/UDP-based protocol (specification) is open-source, and available from *https://github.com/jandejongh/btpsap-common-free*.

Note that since Java™ does not support raw sockets, an additional daemon, *udp2eth*, is required to directly read from and write to the IEEE 802.11p interface, and communicating with *btpsap* through UDP datagrams. The daemon is available from *https://github.com/jandejongh/udp2eth*.

4.7. Reverse-SSH Service

Although *disabled by default*, for reasons of monitoring and maintenance, the V2X Sub-System is equipped with a *reverse-SSH* service. Once enabled and connected to the internet , the V2X Sub-System service sets up a SSH (Secure Shell) tunnel to a TNO-hosted server.

5. ENSEMBLE V2X TEST TOOL

5.1. Introduction and Architecture

The Ensemble V2X Test Tool (EVTT) is part of the ENSEMBLE Reference Implementation. Unlike the V2X Sub-System described in Chapter 4, the EVTT is ENSEMBLE *specific*. Its main purpose is to interact (on-site) with implementations of the ENSEMBLE platooning protocol for reasons of testing (e.g. at an OEM site).

The EVTT is Java™ based, and it implements the *btpsap* client-access protocol as introduced in Section 4.6, allowing it to control one or more APU-based V2X Sub-Systems. As long as there exists an IPv4 connection to the V2X Sub-System(s), and applicable (permissive) firewall settings are active, the EVTT can run anywhere and on any system supporting the Java Virtual Machine. The latest version of the EVTT is available from the ENSEMBLE SharePoint (under *WP3 - Platooning technology\Task3_1\TNO CU for OEM Testing*).

The EVTT functionality consists of the following three concepts:

- A Virtual ITS Station (V-ITS-S), interacting with other ITS Stations. Note that the latter may be *virtual* ITS stations as well as "real" ITS stations from which messages are received.
- A V2X Test, attempting a test-specific interaction between a Virtual ITS Station and another (virtual) ITS Station.
- A Graphical User Interface (GUI).

These concepts are visualized in Figure 5 (DUT = Device Under Test) and described in more detail in the next two sections.

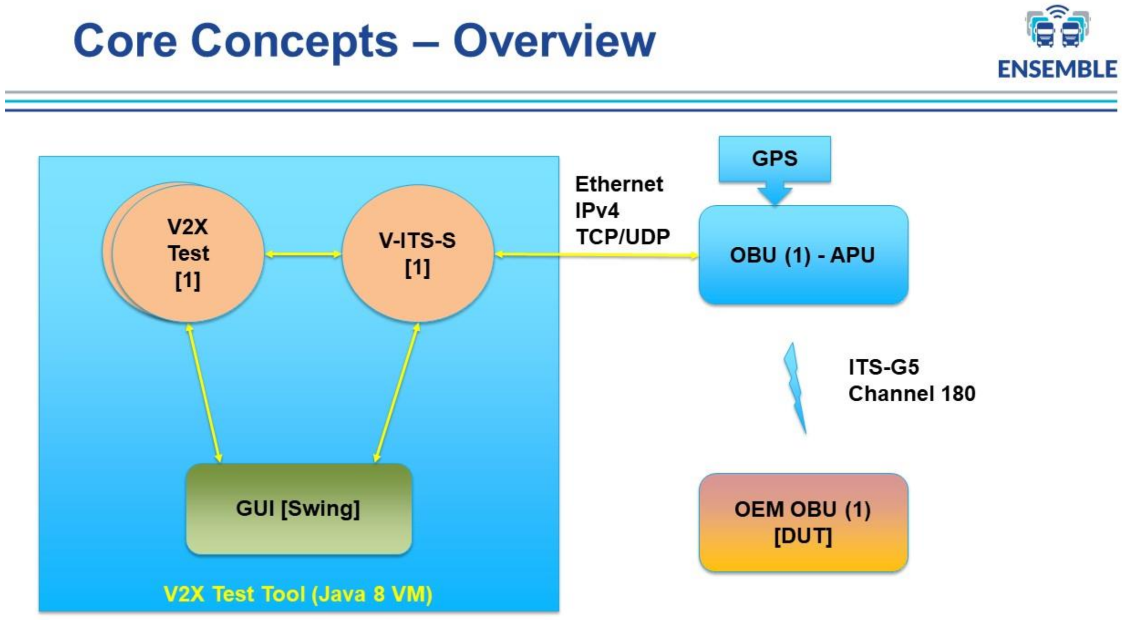


Figure 5: ENSEMBLE V2X Test Tool Architecture.

5.2. Virtual ITS Station

In the V2X Test Tool, a Virtual ITS Station (V-ITS-S) is an active object that represents an ITS Station. This may be a truck, car, or even a Road-Side Unit. V-ITS-Ss can be created at will in the test application, but each of them requires:

- A specific On-Board Unit (also known as Communication Unit) that allows the V-ITS-S to communicate through ITS-G5 with other (virtual) ITS Stations. Alternatively, several simulated communication units can be chosen from (i.e., *not* necessarily requiring a physical communication unit or even over-the-air transmissions).

- A PoTi feed that provides the V-ITS-S with a notion of position and time.

- Various configuration options related to the VITSS behaviour (like sending Cooperative Awareness Messages (CAMs), maintaining a Local Dynamic Map from received CAM messages, and which (truck-) platooning protocol to adhere to.

The basic set-up for testing is the TNO TEST CU (i.e., Communication Unit for testing) in combination with a test tool running on a separate computer. On the computer the tool needs to be installed. It will run on any operating system as long as it supports Java applications. The test tool is provided as an executable file and so it will run directly when opened.

For the physical connection to the TEST CU an ethernet cable is needed. There are two simple ways to connect to the tool:

- Via a self-provided switched network with a DHCP (Dynamic Host Configuration Protocol) server. The TEST CU eth0 port (outermost left port) is configured for DHCP. Use the IP address provided by your own DHCP server to connect the Test tool (from a laptop connected in the same subnet).

- Via Static IP addressing. Connect the TEST CU via eth2 (outermost right port) with the laptop/computer with the test tool on it. Use this IP address to connect the tool:

    - IP address is 192.168.1.28 (mask /24), on eth2 port of TEST CU.

As stated previously, the test tool uses the concept of a virtual station (V-ITS-S). Multiple V-ITS-S can be opened and configured. Within the V-ITS-S pane, see Figure 6, multiple sub-windows are available for specific configuration settings. Some of these windows also have check boxes which need to be selected to activate them. In these checkboxes, as a rule, the colours RED and GREEN represent ERROR/INACTIVE and OK/ACTIVE, respectively. Below short descriptions of the items available in the specific sub-panes (Application, CU, Position and Time, …) are given.

Figure 6: Virtual ITS station pane of test tool.

At the bottom of the screen there are three windows. On the left side there is a Tx Log window, which will list the messages send out by the TEST CU. A clear button is available to clean the log window. On the right side a Rx Log window is available which will list the messages received by the TEST CU. In the middle a Local Dynamic Map window is available which will list an overview of the current active stations within communication range, based on received CAMs. The other windows are:

1) **Application:** some specific platoon related configuration settings,
   a. Platooning ITS-AID= 31415
   b. BTP port nr CAM= 2001
   c. BTP port nr PIM= 3004 (NB not used)
   d. BTP port nr PMM= 3005
   e. BTP port nr PCM= 3006
2) **CU [Btp-Sap]:** configuration window of the connection to the TEST CU
   a. Host= IP address or host name of TEST CU
   b. Port = 36095 (default port number used for BTP-SAP connection
   c. Connection Status= green (=connected)
3) **Position and Time:** configuration window for PoTi related items. Different modes are possible for testing, like using static coordinates, or using the TEST CU GPS receiver combined with an external antenna.
   d. Mode= OFF, STATIC, GPSD (is default)
   e. GPSD Host= IP address or host name of TEST CU
   f. GPSD port= 2947 (default port number)
   g. Latitude [degrees]= used for static position
   h. Longitude [degrees]= used for static position
   i. Altitude
   j. Heading
   k. Velocity

l. Follow distance[m]= use a constant distance based on received PoTi info

m. Status

4) **Vehicle core data**

a. Name= name of vehicle

b. Station ID= unique identifier of ITS station

c. Station Type= Heavy_Truck (ITS station type)

d. Vehicle length= (in meters)

e. Vehicle width= (in meters)

5) **Vehicle extended data**

a. Gross Combination Vehicle Weight= (in kg)

b. Power-to-Mass Ratio= (in W/kg)

c. Brake capacity= (m/s^2)

6) **CAM [Tx]**

a. Active

b. Rate= transmit rate of CAM messages (in Hz)

c. S4P extensions[2]= use standard or extended CAM

d. Advertize Platoonable= selected is True

e. Advertize Joinable= selected is True

f. Status= Observed status of CAM Transmissions

In the menu a V-ITS station can be created (New) or the current selected can be ended (Destroy).

Logging Pane

The tool has a logging pane which can help with debugging and troubleshooting during the tests, or when setting up the TEST CU for testing.

[2] In future revisions of the V2X Test Tool, we intend to replace this option with ‘ENSEMBLE Extensions’.

5.3. V2X Tests

In the V2X Test Tool, next to creating and starting Virtual ITS Stations in the XXX pane, the user can create and manage (start/stop) so-called V2X Tests in the test pane. A V2X Test is a programmatic sequence applied to one or more Virtual ITS Stations with expected results for the test to pass. In the V2X Test Tool, an arbitrary number of tests can be run simultaneously, the instantiation of which requires the following parameters:

- The type of test;
- The test parameters, including the identification of the Virtual ITS Station to which the test applies.

In Figure 7 Test Management window of the V2X Test Tool is shown, which allows the user to create, start, stop and destroy test instances. In Figure 8, we explain the admissible values for the Test Status as appearing in the test pane for a running test.

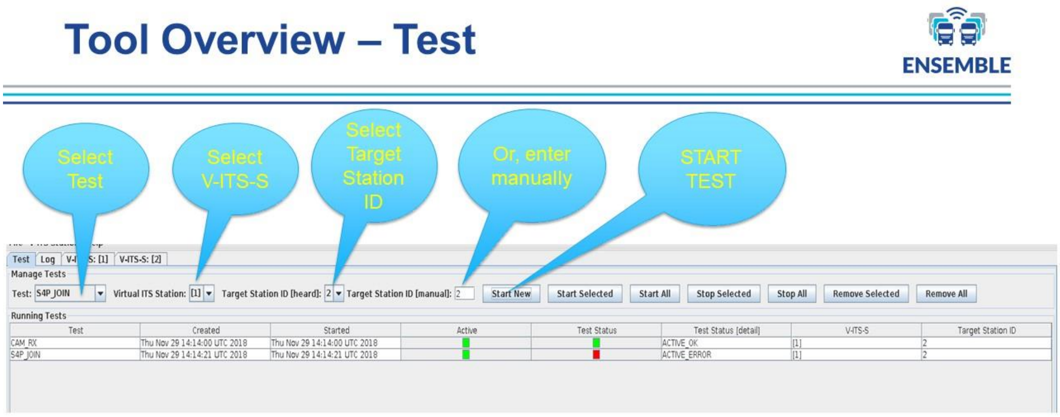
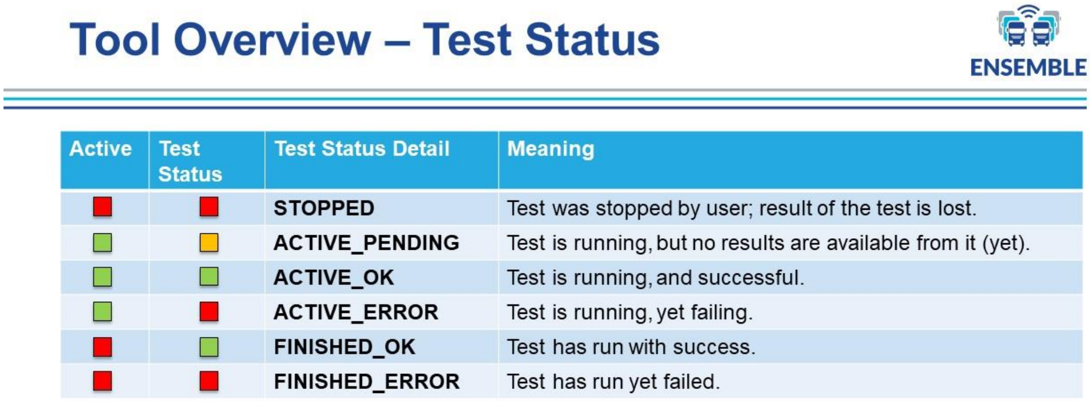


Figure 7: ENSEMBLE V2X Test Tool Test Management.



Active	Test Status	Test Status Detail	Meaning
Red	Red	**STOPPED**	Test was stopped by user; result of the test is lost.
Green	Orange	**ACTIVE_PENDING**	Test is running, but no results are available from it (yet).
Green	Green	**ACTIVE_OK**	Test is running, and successful.
Green	Red	**ACTIVE_ERROR**	Test is running, yet failing.
Red	Green	**FINISHED_OK**	Test has run with success.
Red	Red	**FINISHED_ERROR**	Test has run yet failed.

Figure 8: ENSEMBLE V2X Test Tool Test Status.

6. CONCLUSIONS

In this document the V2X component of the ENSEMBLE Reference Implementation for truck platooning (D3.2) is described. The component consists of:

- An APU(Accelerated Processing Unit)-based hardware unit with a single ITS-G5 interface, GPS-based positioning and time, and an implementation of ETSI GeoNetworking and BTP (with required security).
- The ENSEMBLE V2X Test Tool for on-site testing of ENSEMBLE implementations against a (TNO) baseline implementation.

In future activities in the ENSEMBLE project, the V2X component will be deployed for rigorous testing, for instance in the context of D3.3 (Generic open-source RCP-level reference implementation of the Tactical Layer) and the ENSEMBLE Communication PlugTest.

Finally, although this deliverable is written as communication protocol version independent, it is remarked that future updates of mainly the Test Tool software have to be conducted, as the ENSEMBLE communication protocol is expected to be subject of changes after this deliverable. These changes will be maintained using a proper software version control.

7. BIBLIOGRAPHY

Vissers, John, et. al., *V1 Platooning use-cases, scenario definition and Platooning Levels*, ENSEMBLE Deliverable D2.2, Final version 19-12-2018 (pending EC approval).

Atanassow, Boris and Sjöberg, Katrin, *Platooning protocol definition and Communication strategy*, ENSEMBLE D2.8, approved 12-12-2018.

Atanassow, Boris, et. al., *Security framework*, ENSEMBLE Deliverable D2.9, Final version 11-06-2019 (pending EC approval).

8. GLOSSARY

8.1. Definitions of terms used in ENSEMBLE

Term	Definition
Platoon	A group of two or more automated cooperative vehicles in line, maintaining a close distance, typically such a distance to reduce fuel consumption by air drag, to increase traffic safety by use of additional ADAS-technology, and to improve traffic throughput because vehicles are driving closer together and take up less space on the road.
Platoon Automation Levels	In analogy with the SAE automation levels subsequent platoon automation levels will incorporate an increasing set of automation functionalities, up to and including full vehicle automation in a multi-brand platoon in real traffic for the highest Platooning Automation Level. The definition of "platooning levels of automation" will comprise elements like e.g. the minimum time gap between the vehicles, whether there is lateral automation available, driving speed range, operational areas like motorways, etc. Three different levels are anticipated; called A, B and C.
Requirements	Description of system properties. Details of how the requirements shall be implemented at system level
Convoy	A group of two or more vehicles driving together in the same direction, not necessarily at short inter-vehicle distances and not necessarily using advanced driver assistance systems
Truck Platoon	A truck platoon may be defined as trucks that travel together in convoy formation at a fixed gap distance typically less than 1 second apart up to 0.3 seconds. The vehicles closely follow each other using wireless vehicle-to-vehicle (V2V) communication and advanced driver assistance systems
Specifications	Description of system properties. Details of how the requirements shall be implemented at system level
Scenario	A scenario is a quantitative description of the ego vehicle, its activities and/or goals, its static environment, and its dynamic environment. From the perspective of the ego vehicle, a scenario contains all relevant events. Scenario is a combination of a manoeuvre ("activity"), ODD and events

Manoeuvre (“activity”)	A particular (dynamic) behaviour which a system can perform (from a driver or other road user perspective) and that is different from standing still, is being considered a manoeuvre.
ODD (operational design domain)	The ODD should describe the specific conditions under which a given automation function is intended to function. The ODD is the definition of where (such as what roadway types and speeds) and when (under what conditions, such as day/night, weather limits, etc.) an automation function is designed to operate.
Event	An event marks the time instant at which a transition of a state occurs, such that before and after an event, the system is in a different mode.
Use case	Use-cases describe how a system shall respond under various conditions to interactions from the user of the system or surroundings, e.g. other traffic participants or road conditions. The user is called actor on the system, and is often but not always a human being. In addition, the use-case describes the response of the system towards other traffic participants or environmental conditions. The use-cases are described as a sequence of actions, and the system shall behave according to the specified use-cases. The use-case often represents a desired behaviour or outcome. In the ensemble context a use case is an extension of scenario which add more information regarding specific internal system interactions, specific interactions with the actors (e.g. driver, I2V) and will add different flows (normal & alternative e.g. successful and failed in relation to activation of the system / system elements).
Operational layer	The operational layer involves the vehicle actuator control (e.g. accelerating/braking, steering), the execution of the aforementioned manoeuvres, and the control of the individual vehicles in the platoon to automatically perform the platooning task. Here, the main control task is to regulate the inter-vehicle distance or velocity and, depending on the Platooning Level, the lateral position relative to the lane or to the preceding vehicle. Key performance requirements for this layer are vehicle following behaviour and (longitudinal and lateral) string stability of the platoon, where the latter is a necessary requirement to achieve a stable traffic flow and to achieve scalability with respect to platoon length, and the short-range wireless inter-vehicle communication is the key enabling technology.
Tactical layer	The tactical layer coordinates the actual platoon forming (both from the tail of the platoon and through merging in the platoon) and platoon dissolution. In addition, this layer ensures platoon cohesion on hilly roads, and sets the desired platoon velocity, inter-vehicle distances (e.g. to prevent

	damaging bridges) and lateral offsets to mitigate road wear. This is implemented through the execution of an interaction protocol using the short-range wireless inter-vehicle communication (i.e. V2X). In fact, the interaction protocol is implemented by message sequences, initiating the manoeuvres that are necessary to form a platoon, to merge into it, or to dissolve it, also taking into account scheduling requirements due to vehicle compatibility.
Strategic layer	The strategic layer is responsible for the high-level decision-making regarding the scheduling of platoons based on vehicle compatibility and Platooning Level, optimisation with respect to fuel consumption, travel times, destination, and impact on highway traffic flow and infrastructure, employing cooperative ITS cloud-based solutions. In addition, the routing of vehicles to allow for platoon forming is included in this layer. The strategic layer is implemented in a centralised fashion in so-called traffic control centres. Long-range wireless communication by existing cellular technology is used between a traffic control centre and vehicles/platoons and their drivers.
Service layer	The service layer represents the platform on which logistical operations and new initiatives can operate.
Leading truck	The first truck of a truck platoon
Following truck	Each truck that is following behind a member of the platoon, being every truck except the leading and the trailing truck, when the system is in platoon mode.
Trailing truck	The last truck of a truck platoon
Ego Vehicle	The vehicle from which the perspective is considered.
Platoon cohesion	Platoon cohesion refers to how well the members of the platoon remain within steady state conditions in various scenario conditions (e.g. slopes, speed changes).
Platoon formation	Platoon formation is the process before platoon engaging in which it is determined if and in what format (e.g. composition) trucks can/should become part of a new / existing platoon. Platoon formation can be done orchestrated real time or non-real time.Platoon candidates may receive instructions during platoon formation (e.g. to adapt their velocity, to park at a certain location) to allow the start of the engaging procedure of the platoon.
Platoon engaging	Using wireless communication (V2V), the Platoon Candidate (single vehicle or existing platoon) sends an engaging request to the platoon target (single vehicle or existing platoon) in front. When conditions are met the system starts to decrease the time gap between the trucks to the platooning time gap.
Platoon disengaging	The ego-vehicle can decide to leave the platoon, to split the platoon into 2 new platoons (only a following truck can perform this action) or all the vehicles within the platoon can be decoupled (dissolved). When conditions are met the truck(s) starts to increase the gap between the trucks to a safe non-platooning gap. The

	disengaging is completed when the gap is large enough which is depends on the operational safety based on vehicle dynamics and human reaction times is given.
Platoon dissolve	All trucks are disengaging the platoon at the same time in a coordinated fashion.
Platoon split	The platoon is split in 2 new platoons who themselves continue as standalone entities.
Emergency brake	Brake action with an acceleration of <-4 m/s2
Cut-in	A lane change manoeuvre performed by vehicles from the adjacent lane to the ego vehicle's lane, at a distance close enough (i.e., shorter than desired inter vehicle distance) relative to the ego vehicle.
Cut-out	A lane change manoeuvre performed by vehicles from the ego lane to the adjacent lane.
Cut-through	A lane change manoeuvre performed by vehicles from the adjacent lane (e.g. left lane) to ego vehicle's lane, followed by a lane change manoeuvre to the other adjacent lane (e.g. right lane).
Steady state	In systems theory, a system or a process is in a steady state if the variables (called state variables) which define the behaviour of the system or the process are unchanging in time. In the context of platooning this means that the relative velocity and gap between trucks is unchanging within tolerances from the system parameters.
Platoon candidate	A truck who intends to engage the platoon either from the front or the back of the platoon.
Target Time Gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds, agreed by all the Platoon members; it represents the minimum distance in seconds allowed inside the Platoon.
Time gap	Elapsed time to cover the inter vehicle distance by a truck indicated in seconds.
Legal Safe Gap	Minimum allowed elapsed time/distance to be maintained by a standalone truck while driving according to Member States regulation (it could be 2 seconds, 50 meters or not present)
Fail-safe	A fail-safe in engineering is a design feature or practice that in the event of a specific type of failure, inherently responds in a way that will cause no or minimal harm to other equipment, the environment or to people.

8.2. Acronyms and abbreviations used in ENSEMBLE

Acronym / Abbreviation	Meaning
ACC	Adaptive Cruise Control
ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ACSF	Automatically Commanded Steering Function
ADAS	Advanced driver assistance system
ADR	Agreement concerning the International Carriage of Dangerous Goods by Road
AEB	Automatic Emergency Braking (System, AEBS)
ASIL	Automotive Safety Integrity Level
ASN.1	Abstract Syntax Notation One
BTP	Basic Transport Protocol
C-ACC	Cooperative Adaptive Cruise Control
C-ITS	Cooperative ITS
CA	Cooperative Awareness
CAD	Connected Automated Driving
CAM	Cooperative Awareness Message
CCH	Control Channel
CS	Cyber Security
CSF	Corrective steering functions
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
DSRC	Dedicated Short-Range Communications

DUT	Device-Under-Test
EC	European Commission
EMC	Electromagnetic Compatibility
ESF	Emergency steering function
ESP	Electronic Stability Program
ETSI	European Telecommunications Standards Institute
EU	European Union
FAD	Fully Automated Driving
FCW	Forward Collision Warning
FLC	Forward Looking Camera
FSC	Functional Safety Concept
GN	GeoNetworking
GNSS	Global Navigation Satellite System
GPIO	General Purpose I/O
GPS	Global Positioning System
GRVA	Working Party on Automated/Autonomous and Connected Vehicles
HAD	Highly Automated Driving
HARA	Hazard Analysis and Risk Assessment
HIL	Hardware-in-the-Loop
HMI	Human Machine Interface
HW	Hardware
I/O	Input/Output
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITC	Inland Transport Committee

ITS	Intelligent Transport System
ITS-S	ITS Station
IVI	Infrastructure to Vehicle Information message
LDWS	Lane Departure Warning System
LKA	Lane Keeping Assist
LCA	Lane Centering Assist
LRR	Long Range Radar
MAP	MapData message
MRR	Mid Range Radar
MVC	Modular Vehicle Combinations
OBD	On-Board Diagnostics
OBU	On-Board Unit
OS	Operating system
ODD	Operational Design Domain
OEM	Original Equipment Manufacturer
OSI	Open Systems Interconnection (model)
OTA	Over the air
PAEB	Platooning Autonomous Emergency Braking
PDU	Protocol Data Unit
PMC	Platooning Mode Control

PPS	Pulse-per-Second
RCP	Remote Control Parking
RSU	Road Side Unit
SAE	SAE International, formerly the Society of Automotive Engineers
SCH	Service Channel
SDO	Standard Developing Organisations
SIL	Software-in-the-Loop
SMA	SubMiniature version A
SOTIF	Safety of the Intended Function
SPAT	Signal Phase and Timing message
SRR	Short Range Radar
SSD	Solid State Disk
TF	Task Force
TS	Technical Specification (in the context of standardisation, e.g., ETSI)
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to any (where x equals either vehicle or infrastructure)

VDA	Verband der Automobilindustrie (German Association of the Automotive Industry)
VECTO	Vehicle Energy Consumption Calculation Tool
V-ITS-S	Virtual ITS Station
VMAD	Validation Method for Automated Driving
WIFI	Wireless Fidelity
WP	Work Package
WP.1	Working Party 1 - Global Forum for Road Traffic Safety
WP.29	Working Party 29 - World Forum for Harmonization of Vehicle Regulations