sponsored by Federal Ministry of Transport and Digital Infrastructure

Deliverable D2.2: Final ConVeX System Architecture Description

Version: V1.0

January 31, 2018

Audi ERICSSON QUALCOMM swarco TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Contents

1 Scope 5

2 References 6

3 Abbreviations 8

4 High Level ConVeX System Overview 11

4.1 System Components 11
4.2 Reference Point Definition 12

5 Description of System Components 13

5.1 C-V2X Communication Platform 13
5.1.1 Overview 13
5.1.2 Architecture 13
5.1.3 Vehicle Integration of the C-V2X Communication Platform 14
5.2 In-Vehicle-System 15
5.2.1 Connection between the Car and the C-V2X Communication Platform 16
5.2.2 In-Car Systems 16
5.2.3 Human-Machine-Interfaces (HMI) 17
5.2.4 User Input 18
5.2.5 Sensors 18
5.3 ITS Roadside Unit 18
5.3.1 General Architecture Aspects 18
5.3.2 IRS functionalities 20
5.3.2.1 Processing of C-ITS Messages: 20
5.3.2.2 Management and Maintenance Functions: 20
5.3.2.3 Security Functionalities 21
5.3.3 Interfaces 21
5.3.3.1 V2I Interface 21
5.3.3.2 ICS Interface 21
5.4 System for Vulnerable Road User 22
5.4.1 General Issues and Technical Challenges 22
5.4.2 VRUE Design Options 23
5.4.3 VRUE in ConVeX 23
5.5 4G/5G Test Network 24
5.5.1 A9 Testbed 26
5.5.2 Rosenheim Testbed 27
5.6 Cloud Servers 29
5.6.1 Traffic Control Center 29
5.6.2 V2N Application Servers 32
5.6.2.1 VM in the Flight Rack 33
5.6.2.2 Physical Machine Connected to Flight Rack 33
5.6.2.3 Remote Server 33
5.6.3 Measurement Server 34

6 Communication Protocols 40

6.1 PC5 Interface 40
6.1.1 Specifics of the Physical Layer 40
6.1.2 Physical Channels 41
6.1.3 Resource Pool 42
6.1.4 Distributed Congestion Control 42
6.1.5 Other Layers: RLC and PDCP 43

6.2 Uu Interface 43
6.2.1 Overall Uu Protocol Stack 43
6.2.2 MAC Procedures 45
6.2.2.1 Dynamic Scheduling 45
6.2.2.2 Semi-Persistent Scheduling (SPS) 48
6.2.2.3 TTI Bundling 50
6.3 ITS Stack 50
6.4 ITS Applications 54
6.5 Security Aspects 56
6.5.1 Introduction 56
6.5.2 ETSI ITS Security Architecture 58
6.5.3 ITS Security Frameworks 60
6.5.3.1 Overview 60
6.5.3.2 Communication between ITS Station and EA 60
6.5.3.3 Communication Between ITS Station With AA 61
6.5.3.4 Role of CA and Root CA 61
6.5.3.5 ITS Message Formats 62
6.5.3.6 Confidentiality Protection 62
6.5.4 Security Mechanisms Supported by the C-V2X Communication Platform 63
6.6 In-Vehicle Internal Communication 64
6.6.1 CAN Communication 64
6.6.2 Ethernet 64
6.6.3 Displaying Information 64
6.6.4 Forwarding Information to the C-V2X CP 65

7 Simulation Platform 66

7.1 Introduction 66
7.2 Overview of Current Simulators 67
7.2.1 Road Traffic Simulation Frameworks 67
7.2.1.1 Simulation of Urban Mobility 68
7.2.1.2 Quadstone Paramics Modeller 69
7.2.1.3 Treiber's Microsimulation of Road Traffic 69
7.2.1.4 Aimsun 70
7.2.1.5 Trafficware SimTraffic 70
7.2.1.6 CORSIM TRAFVU 71
7.2.1.7 Vissim 71
7.2.2 Network Simulation Frameworks 72
7.2.2.1 Network Simulator (NS-3) 72
7.2.2.2 OMNET++ 72
7.3 Selection of Frameworks for the Project 73
7.4 Simulation Framework Overview and Architecture 73
7.4.1 Overall Architecture 73
7.4.2 Functional Components 74
7.4.3 Interfaces 75

8 Summary and Conclusions 76

1 Scope

This document has the objective to describe the functional architecture and the supported topology of the various system components to be implemented for the ConVeX trial. In this sense, all hardware related components that build up the trial system and the interfaces between them with the used communication protocols are included here.

It is important to consider that the functional architecture is designed to support the use cases and requirements defined in Deliverable 1.1 [1]. Some complex scenarios may still require other elements or functionalities that go beyond the scope of ConVeX. For some aspects, the document provides an outlook on what might be extended, specifically for a later commercial deployment.

After a high-level system overview, the following components and functionalities will be described in detail:

- C-V2X communication platform
- In-Vehicle-System that integrates the C-V2X communication platform into the car
- Roadside Unit being made capable of C-V2X communication
- System for a Vulnerable Road User
- 4G/ 5G Test Network architecture
- Cloud Servers for different purposes
- Communication links and used protocols
- Interfaces between different components to support exchange of information and allow to utilize functions (e.g. APIs)
- Introduction of the simulation platform used in the scope of the project

In summary, this document should clarify, how the different parts brought from the partners connect with each other and how their interworking is achieved, assuring full system integration and functionality. Note, that there is the possibility of adaptations that might come up in the course of the project.

2 References

[1] ConVeX Deliverable 1.1: “Use Cases, Requirements, Performance Evaluation Criteria”

[2] http://www.cse.unt.edu/~rdantu/FALL_2013_WIRELESS_NETWORKS/LTE_Alcatel_White_Paper.pdf

[3] http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core

[4] 3GPP TS 22.185 v.14.3.0 Service requirements for V2X services

[5] 3GPP TS 23.258 v.14.3.0 Architecture enhancements for V2X services

[6] 3GPP TS 36.211 v.14.3.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation

[7] 3GPP TS 36.213 v.14.3.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures

[8] 3GPP TS 36.321 v.14.3.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification

[9] 3GPP TS 36.323 v.14.3.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification

[10] 3GPP TS 36.331 v14.3.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification

[11] Sesia, Stefania(Author): “LTE: The UMTS Long Term Evolution: From Theory to Practice” (2nd Edition)

[12] IEEE Std 1609.2-2016, IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages.

[13] “COOPERATIVE ITS SECURITY STANDARDIZATION AND ACTIVITIES ON EUROPEAN C-ITS TRUST MODEL AND POLICY”, ETSI IoT Security WORKSHOP, 13-15 June 2016, by Brigitte Lonc, Renault.

[14] ETSI TS 102 940: "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management", V2_0_1 (2017-03) draft

[15] ETSI TS 102 637-2: "ITS Vehicular Communications; Basic Set of Applications; Specification of CAM"

[16] ETSI TS 102 637-3: "ITS Vehicular Communications; Basic Set of Applications; Specification of DENM"

[17] ETSI TR 102 893: “Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)”

[18] ETSI TS 102 731: "Intelligent Transport Systems (ITS); Security; Security Services and Architecture".

[19] ETSI TS 103 097 v1.2.1: "Intelligent Transport Systems (ITS); Security; Security header and certificate formats".

[20] ETSI TS 102 940: "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management", v1.2.1

[21] ETSI TS 102 723-8 v1.1.1 (2016-04), "Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 8: Interface between security entity and network and transport layer"

[22] ETSI TS 102 941: "Intelligent Transport Systems (ITS); Security; Trust and Privacy Management"

[23] ETSI TS 102 942: "Intelligent Transport Systems (ITS); Security; Access Control".

[24] ETSI TS 102 943: "Intelligent Transport Systems (ITS); Security; Confidentiality services".

[25] “Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems v1.0 (draft)” issued by “European C-ITS Platform - WG Security”.

[26] ETSI TS 103 301: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services".

[27] ETSI TS 102 965: "Intelligent Transport Systems (ITS); Application Object Identifier (ITS-AID); Registration"

[28] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2"

[29] IETF RFC 6347: "Datagram Transport Layer Security Version 1.2

[30] 3GPP TS 36.300: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 (Release 14)"

[31] 3GPP TS 33.303: " Proximity-based Services (ProSe); Security aspects (Release 14)"

[32] Sommer, Christoph, and Falko Dressler. *Vehicular Networking*. Cambridge University Press, 2014.

[33] Piorkowski, Michal, et al. "TraNS: realistic joint traffic and network simulator for VANETs." *ACM SIGMOBILE mobile computing and communications review* 12.1 (2008): 31-33.

[34] Sommer, Christoph, Isabel Dietrich, and Falko Dressler. "Realistic simulation of network protocols in VANET scenarios." *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007.

[35] Wegener, Axel, et al. "TraCI: an interface for coupling road traffic and network simulators." *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008.

3 Abbreviations

3GPP	3rd Generation Partnership Project
5G	5th Generation
AA	Authentication Authority
ADAS	Advanced Driver-Assistance Systems
AID	Application ID
AP	Application Processor
API	Application Programming Interface
AQ	"*Anzeigequerschnitte*" (German for: gantries)
AT	Authorization Ticket
BNetzA	Bundesnetzagentur
CA	Certification Authority
CAM	Cooperative Awareness Message
CAN	[Automotive] Controller Area Network
CAN-GW	CAN-Gate Way
CCARD	Connected Car Application Reference Design
C-ITS	Cooperative Intelligent Transport Systems
C-ITS-S	Central ITS Station
CSR	Certificate Signing Request
DEN	Decentralized Environmental Notification
DENM	Decentralized Environmental Notification Message
DMRS	Demodulation Reference Signal
DR	Dead Reckoning
DSRC	Dedicated Short Range Communication
DTLS	Datagram Transport Layer Security
EA	Enrolment Authority
EEBL	Emergency Electronic Brake Light (use case)
eMBMS	Evolved Multimedia Broadcast Multicast Services
eNB	Evolved Node B
ECU	Electronic Control Unit
ETSI	European Telecommunications Standards Institute
E-UTRA	Evolved UMTS Terrestrial Radio Access
FCD	Floating Car Data
FCW	Forward Collision Warning
FG	Functional Group
GNSS	Global Navigation Satellite System
HMI	Human Machine Interface
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HV	Host Vehicle

ICS	ITS Central Station
Id	Identification
ITS	Intelligent Transport System
IVI	In-vehicle Information
IVIM	In Vehicle Information Message
IRS	ITS Roadside station
IP	Internet Protocol
JSON	JavaScript Object Notation
KPI	Key Performance Indicators
LTE	Long Term Evolution
MAPEM	Map data Extended Message
MQTT	Message Queue Telemetry Transport
QoS	Quality of Service
PC5	ProSe Communication reference point 5
PDU	Protocol Data Unit
ProSe	Proximity-based Services
PKI	Public Key Infrastructure
pub/sub	publish/ subscribe
PVD	Probe Vehicle Data
P-ITS-S	Personal ITS Station
QXDM	Qualcomm eXtensible Diagnostic Monitor (also called QXDM Professional™)
R-ITS-S	Roadside Intelligent Transport System Station
RV	Remote Vehicle
RWW	Road Works Warning
RSU	Roadside Unit
SAE	Society of Automotive Engineers
SDK	Software Development Kit
SHA	Secure Hash Algorithm
SPATEM	Signal Phase And Timing Extended Message
SST	"*StreckenSTation"* – German: Outstation
SW	Software
SWD	Shockwave Damping Service Deployment
TCC	Traffic Control Center
TCP	Transmission Control Protocol
TCU	Telematics Control Unit
TLS	Transport Layer Security
TLS	"*Technische Lieferbedingungen für Streckenstationen*" – German term for the standard defining the connection of outstations
TVRA	Trust, Vulnerability, Risk Analysis
UE	User Equipment
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

USB	Universal Serial Bus
UZ	"*UnterZentrale"* – German: Subcenter
Uu Interface	Interface between UE and Node B/ eNode B
V2I	Vehicle to Infrastructure
V2N	Vehicle to Network
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
V-ITS-S	Vehicular Intelligent Transport System Station
VM	Virtual Machine
VMS	Variable Message Sign
VRU	Vulnerable Road User
VRZ	"*VerkehrsRechnerZentrale"* - German: Main Traffic Computer Center
VRUE	Vulnerable Road User Equipment
V-TCC	Virtual Traffic Control Centre
V-VMS	Virtual Variable Message Sign
WAVE	Wireless Access in Vehicular Environments
Wi-Fi	Wireless Fidelity

4 High Level ConVeX System Overview

4.1 System Components

This section provides an overview of the system components involved and needed for a complete V2X system, and which are used in the ConVeX trial system.

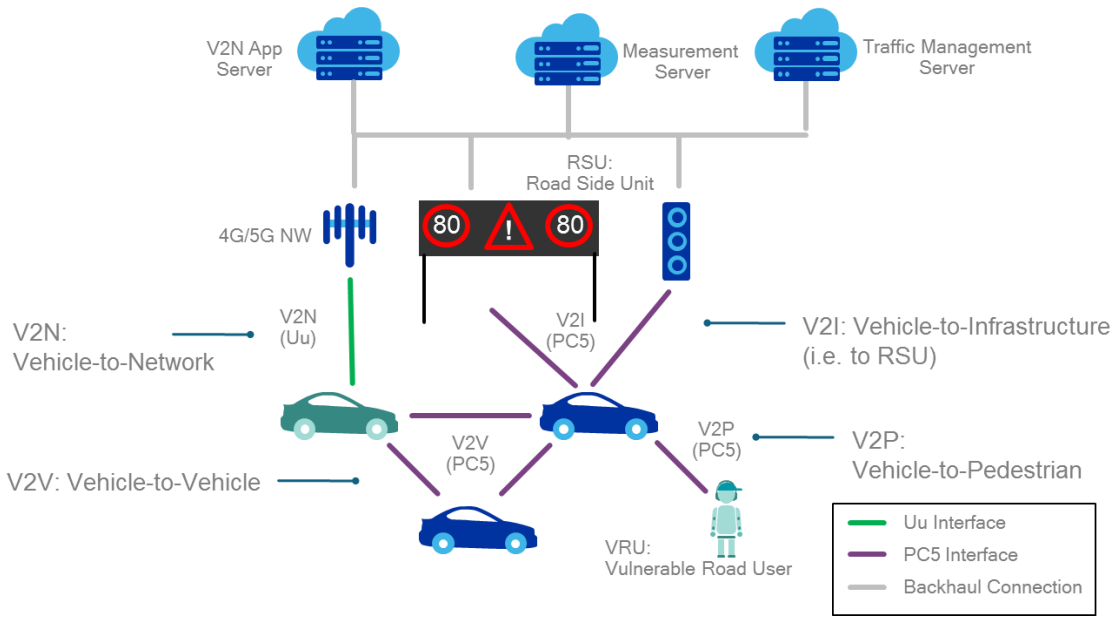

Figure 4.1-1: ConVeX High Level System Architecture

As illustrated in Figure 4.1-1, the various participants in traffic scenarios – cars and pedestrians, equipped with V2X enabled UEs are one main building block. Pedestrians can also be replaced with other road users that would need more protection than a car, i.e. also bicyclists, or motor bikers – the whole class of so called Vulnerable Road Users (VRUs). Also involved are Roadside Units (RSUs), which could be stand alone, or e.g. integrated into traffic lights, and additionally if legacy LTE is used for communication with the mobile Network, the eUTRAN and its core network is part of the system. This could also in the future be extended to the 5G mobile Network.

Further components are servers in the cloud for different purposes – here mainly the Traffic Management Server or Traffic Control Center as a whole, which would be the entity in a real deployed system, a V2N Application Server providing specific services to the traffic participants, and finally a Measurement and Configuration Server. For the purpose of the trial, this server will collect information from all entities involved in the defined use cases, provide the possibility to evaluate and aggregate the pieces of information, and calculate Key Performance Indicators. This would not be part of a real-world system. Also some security related configurations might be done with this server (like provisioning of certificates).

To enable the C-V2X communication in the ConVeX trial, the so-called C-V2X Communication Platform will be used and integrated into vehicles and RSUs, or used by the VRU.

Detailed descriptions of the mentioned system components will follow in the respective sections of this document.

4.2 Reference Point Definition

The C-V2X specifications in 3GPP release 14 include two complementary communication interfaces: PC5 (ProSe Communication reference point 5) and Uu (the legacy LTE air interface). Figure 4.1-1 illustrates where these interfaces are used.

The PC5 interface provides the possibility of direct communication between two devices and is used for vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P) and vehicle-to-infrastructure (V2I) communication. More details on PC5 can be found in section 6.1.

The LTE-Uu connection complements the system providing communication on non-direct links within the (wide area) network, e.g. for getting dedicated V2N services. This connection will also be used as backhaul-link between road-side-units (RSU) and its traffic management center, acting as transparent IP connection, and similarly it provides the connection to the Measurement Server. Details on Uu can be found in section 6.2.

Note that the 3GPP standard also foresees a variant of a V2I communication via Uu link, in this case assuming the RSU to be part of an eNode B. However, this setup is not used within this trial as it is no probable option for a future real world implementation.

5 Description of System Components

5.1 C-V2X Communication Platform

5.1.1 Overview

The C-V2X Communication Platform can be used for car and roadside unit integration and supports advanced use cases for V2V, V2I, V2P and V2N. It is supposed to be used for development and testing of trial specific use cases.

The main hardware component for the trial system is the C-V2X Communication Platform. Figure 5.1-1 shows a simplified structural and functional overview of the prototype implementation with its corresponding components. It is composed of the C-V2X Development Platform (C-V2X DP) shown on the right side linked via an Ethernet connection to the Communication Module (CCARD - Connected Car Application Reference Design) shown on the left side. In the trial setup, the CCARD is used for the V2N communication via an LTE network.

The C-V2X Development Platform consists of the Application Processor (AP) module and the C-V2X Module and peripheral components. These two modules are linked and communicate via an Ethernet connection. The C-V2X DP furthermore provides external connectivity via Ethernet, CAN and USB.

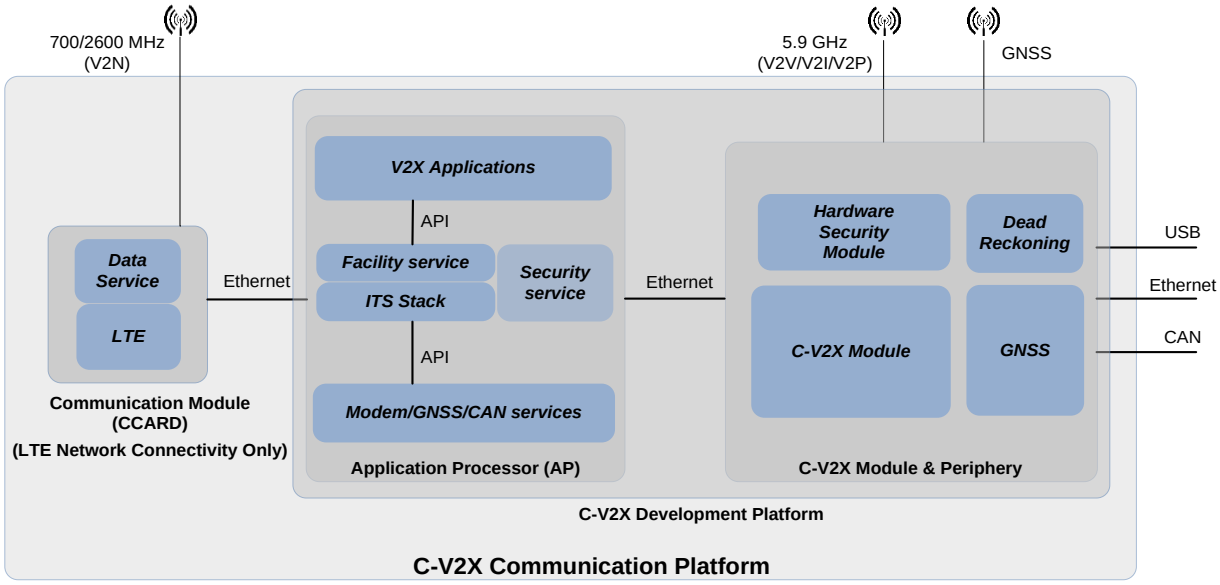

Figure 5.1-1: C-V2X Communication Platform (simplified)

5.1.2 Architecture

The AP module of the C-V2X Development Platform consists of a Qualcomm Snapdragon APQ8096 which runs the software providing the lower layer services, the ITS Stack and the individual V2X applications. Application Programming Interfaces (APIs) are provided to enable the usage of the different services. The figure shows the two most important APIs giving access to the ITS Facility layer as well as the lower entry point to e.g. use the modem functionalities. In addition, the AP also runs the security management functions which will ensure proper handling of the security credentials. The APQ8096 verifies ITS message security, evaluates position

information from the C-V2X Module, and processes the vehicle information/events coming from the CAN bus.

The C-V2X Module and peripheral components (Hardware Security Module (HSM), GNSS and Dead Reckoning) are connected to the Application Processor via an Ethernet interface. This module includes the C-V2X Module itself employing the Qualcomm Snapdragon X16 LTE modem MDM9x50, which implements various communication technologies on its respective frequency bands, and is used here to support the V2V/V2I/V2P use cases in the 5.9 GHz band. The C-V2X Module also contains functions providing desired security mechanisms in the Hardware Security Module (HSM). Positioning functionality is also supported. To achieve lane level accuracy, the module integrates a global navigation satellite system (GNSS) component as well as a position augmentation employing sophisticated algorithms for dead reckoning at 10 Hz.

As shown in Figure 5.1-1 the V2N functionality is supported by a separate Communication Module, the CCARD. The CCARD is a Telematics Control Unit (TCU), which can support a variety of features for a car. For the ConVeX trial, it is only used for supporting data service connectivity in the 700 MHz and 2600 MHz LTE bands. Again, utilizing the Qualcomm Snapdragon X16 LTE modem MDM9x50, it is capable of downlink speeds of 1 Gbps at the physical layer with support of 4x4 MIMO, 4 Downlink Carrier Aggregation and 2 Uplink Carrier Aggregation.

Note that for a later commercialization all C-V2X and legacy TCU capabilities will be supported by one hardware component.

5.1.3 Vehicle Integration of the C-V2X Communication Platform

As one example for a car integration, Figure 5.1-2 shows a potential functional split of the components of an in-vehicle prototype integration in the context of this project. In this example, the functionalities are split into “Roof Box” components allowing short distance antenna cabling for radio and communication modules and “Inside Vehicle” components realizing user-friendly access for handling of information, visualization and control of various functions and drive tests.

To achieve low RF signal attenuations, it might be important that antenna cables are as short as possible. This can be accomplished by placing the main components that require RF signals close to the roof of the vehicle. This is especially important for the higher C-V2X frequencies at 5.9 GHz. The actual implementation in the selected test vehicles may deviate from this example though, and will be described in the respective car integration section. The C-V2X Communication Platform contains RF antenna connectors for the different frequency bands of the used technologies, i.e. 5.9 GHz C-V2X, 700 and 2600 MHz LTE, and 1500 MHz GPS. For C-V2X and

LTE, two and four antenna connectors could be used, respectively. This aims to allow Receive Diversity for C-V2X and 4x4 MIMO support for LTE.

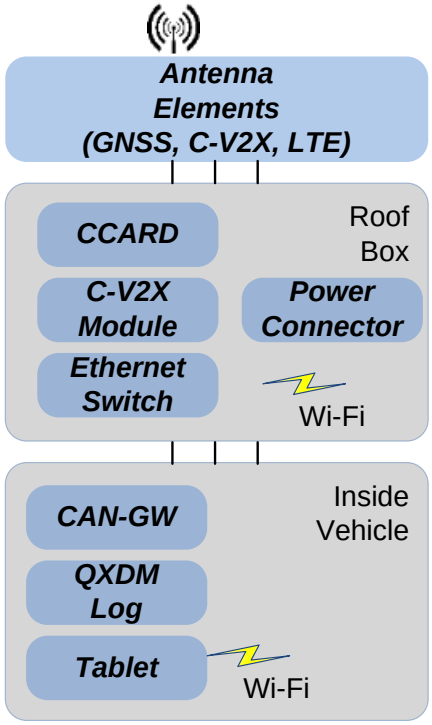

Figure 5.1-2: Example for a Car Integration of the C-V2X Communication Platform (shown without internal USB, Ethernet and CAN interfaces)

Inside the vehicle, functionality for handling of the CAN bus and other connections and interfaces may be located. Also, diagnostic logging (e.g., using Qualcomm's QXDM solution connected via USB) and other visualization tools, control of various components and elements via a laptop or tablet may be implemented or located.

For additional external access and control, the roof box can be further equipped with a Wi-Fi access point as well as an Ethernet switch as indicated in Figure 5.1-2. A power connector allows attaching to a 12V power supply.

External power supplies as well as a CAN Gateway (CAN-GW) and CAN splitter (not shown) may be needed for the integration into the vehicle. The CAN splitter provides a means to connect to different ports of the C-V2X Module to input messages for dead reckoning and other information for the considered use cases. The CAN-GW may be needed to adapt messages to the considered use case applications.

For a roadside unit integration, a similar diagram can be depicted, where no CAN bus and additional support for lane level accuracy via Dead Reckoning are needed. The RSU would basically consist of the components in the "roof box" plus the antennas, and would potentially be connected via Ethernet to a Traffic Control Center or other external components.

More details on the in-vehicle and roadside unit integrations will be described in the respective sections.

5.2 In-Vehicle-System

This chapter describes the main components, interconnection and interfaces in the vehicles which are utilized in the project. The provided vehicle model is an Audi Q7. Figure 5.2-1 shows this vehicle model.

Figure 5.2-1: Audi Q7 Vehicle

5.2.1 Connection between the Car and the C-V2X Communication Platform

The C-V2X DP is connected to the car in two ways. For base information like velocity, x- and y-acceleration, and braking interventions a CAN interface is used. For interpretation of the CAN messages a database is also provided. For additional, functional input an Ethernet interface is used. The different components can be accessed with different IP addresses.

Figure 5.2-2 shows a schematic of the vehicle integration.

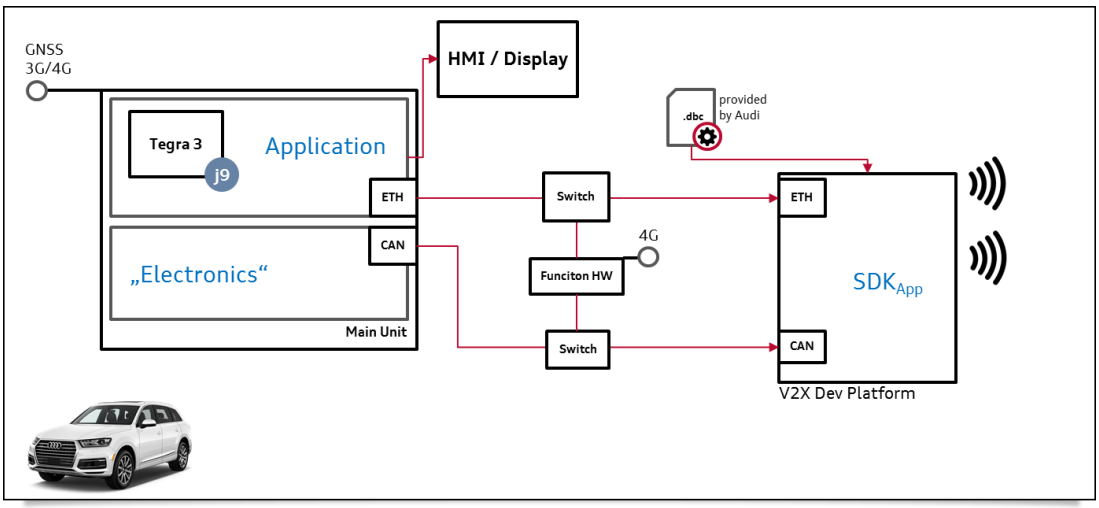

Figure 5.2-2: Schematic of the vehicle integration

5.2.2 In-Car Systems

Main Unit

The Main Unit is the infotainment ECU (Electronic Control Unit) in the vehicle. It is used to render screens for the map, the radio and entertainment functions. It contains a Tegra application processor that provides a Java Runtime Environment which includes an Eclipse/IBM j9 Java Virtual Machine called OpenJ9. It controls the Multi Media Interface (MMI) system, like the different displays and user input possibilities.

Function Hardware

The Function Hardware can be seen as an additional computer connected to the car networks. It provides the opportunity to use flexible implementations to realize Use Cases in addition to the ones provided by the V2X applications running on the C-V2X Communication Platform. In that

case, only the modem functionality of the C-V2X CP would be used. The Function Hardware supports different programming languages and can also interface to the screens in the vehicle.

Listed below are interfaces for displaying, entering and processing of functional relevant parameters and information:

5.2.3 Human-Machine-Interfaces (HMI)

The vehicles contain capabilities to present information to the driver acoustically and visually ex-factory. Additionally, there are also interfaces for input available (like buttons, touchscreens, voice command).

Displays

1) MMI (Multi Media Interface) Display:

The MMI display is located to the right of the steering wheel. It has a resolution of 1024x480 pixels. Usually this display shows multimedia applications or is used for visualization of route guidance or other navigation functions. Technically, the full resolution of the display can be used in the project to visualize information. Figure 5.2-3 displays the installation location and navigation screen for routing functions

Figure 5.2-3: MMI display - the top shows the installation location, the bottom a typical navigation screen

2) Dashboard:

The instrument cluster is located behind the steering wheel. The vehicle's velocity and the engine's rotational speed are displayed as digital read outs on the right and left side of the dashboard, respectively. The area in between is available for displaying information relevant for ConVeX Use Cases (see Figure 5.2-4). The total resolution is 1440 x 540 pixels.

Figure 5.2-4: Audi instrument cluster – the hatched area describes the available space for displaying information relvant for the Use Cases in the project.

5.2.4 User Input

For user input, both the MMI keypad located on the transmission tunnel between driver's and co-driver's seats, and the function buttons on the steering wheel can be used.

5.2.5 Sensors

The car offers a wide variety of sensors. The supported type of measurements as shown in Figure 5.2-5 can also serve as functional input parameters for ITS applications.

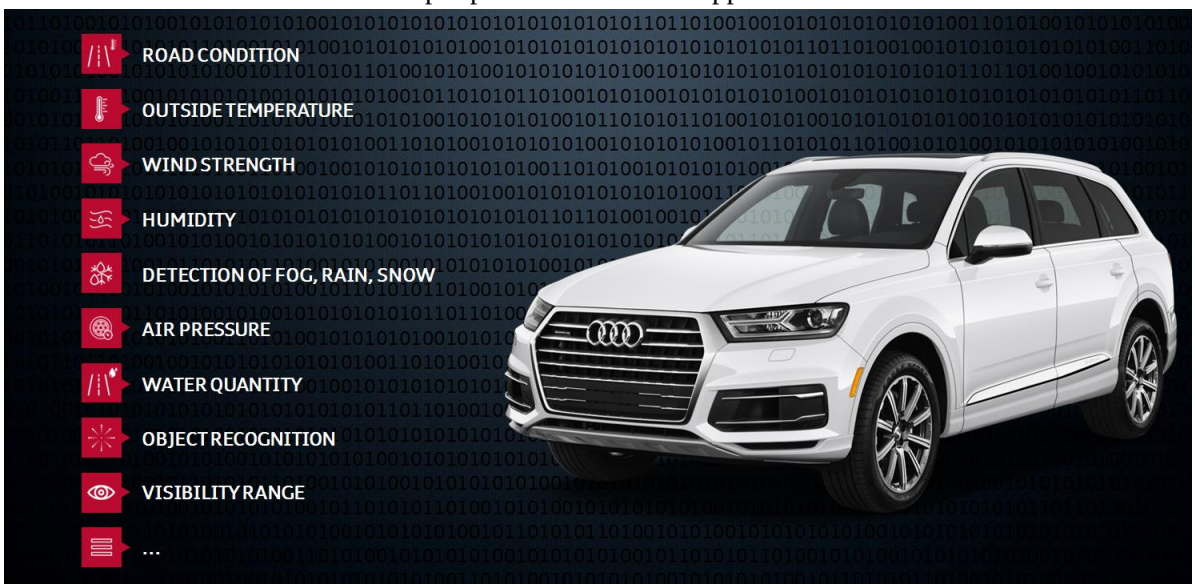

Figure 5.2-5: Overview about sensor values of an Audi Q7 vehicle

5.3 ITS Roadside Unit

5.3.1 General Architecture Aspects

A typical structure of highway cooperative ITS (C-ITS) which uses one or more ITS Roadside Units (IRS) – also referred in other sources as R-ITS-S (Roadside ITS Station) or RSU (Roadside Unit) – is given in Figure 5.3-3.

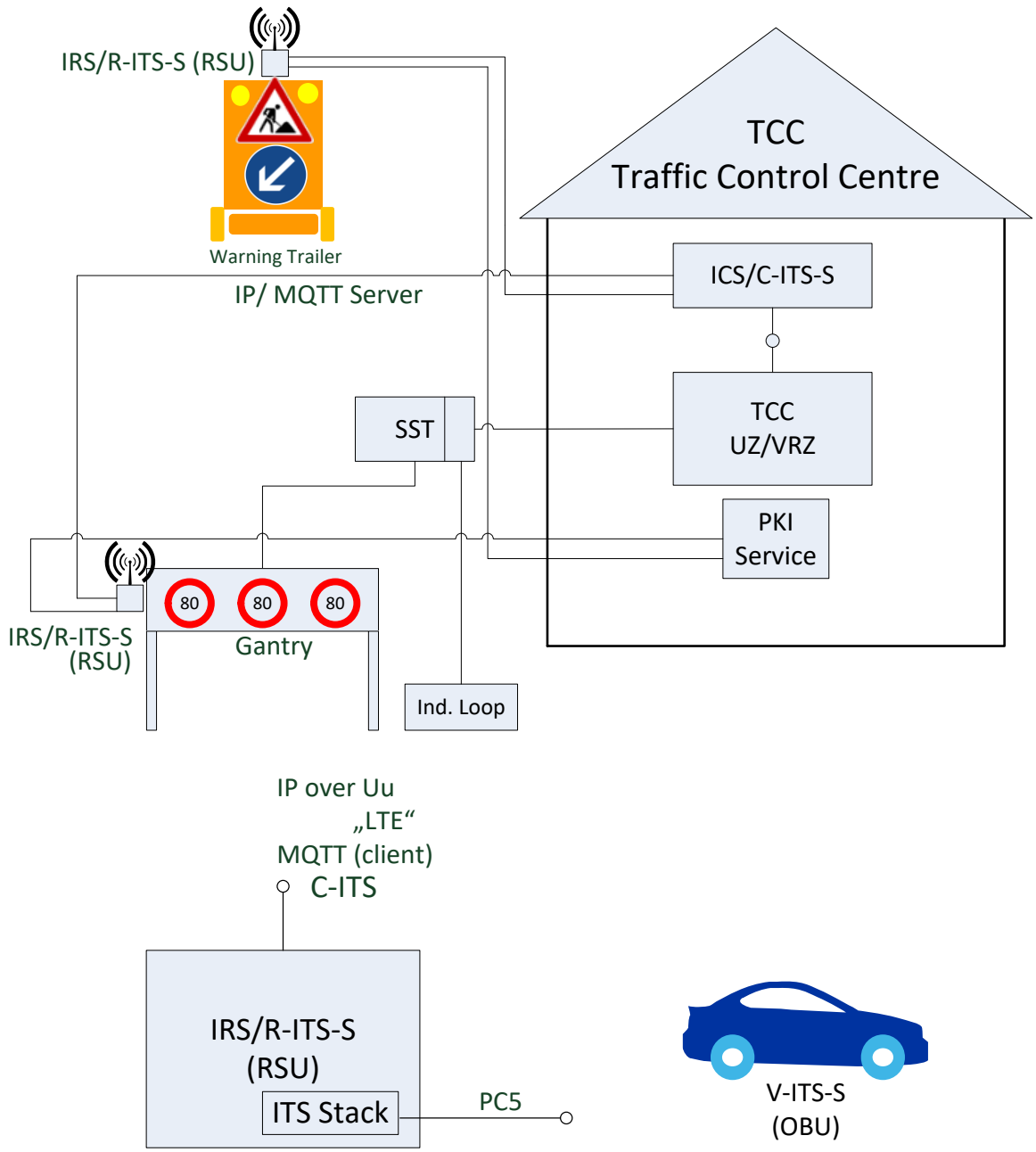

Figure 5.3-4 Structure of highway C-ITS

Corresponding to this structure, IRS serves as the communication frontend to the ITS Vehicle System - alternative references are: V-ITS-S (Vehicle ITS Station and OBU (On Board Unit)).

In such a configuration, the IRS will receive traffic management data from ITS Central Station (ICS) – also named C-ITS-S (Central ITS Station) - and after preparing and actualizing the data (DENM, IVIM), will send it to the OBU. Typically, the C-ITS-S is a part of a more complex Traffic Control Center (TCC) which can include other components such as UZ/VRZ - "*UnterZentrale/VerkehrsRechnerZentrale"* (German: Subcenter/Main Traffic Computer Center) – and components for control of the SST ("*StreckenSTationen"* – German for: Outstation) with Variable Message Sign (VMS) gantries.

In reverse direction, the IRS receives Probe Vehicle Data (PVD) from vehicles and after pre-processing, forwards it to the ICS for utilization in ITS control algorithms (e.g. for Shock Wave Damping (SWD), Road Works Warning (RWW)).

The IRS can be installed on a road works safety warning trailer or any other similar mobile solution, or inside the existing Outstations installed at the gantries on the highway (fixed solution).

In the ConVeX project, the IRS is designed to be placed in a temporary outdoor environment. For the related use cases, the IRS does not have an interface to stationary gantries, but is representing several virtual gantries containing cross-section measurement and cross-section display units. Here, the IRS is responsible for de- and encoding of C-ITS messages sent and received over the C-V2X link (PC5) and preprocessing upcoming information. Furthermore, the IRS should handle the communication with the ICS and is responsible for correct processing of traffic, management and security information.

For handling of security items (see chapter 6.5), every IRS should have a connection to the PKI (Public Key Infrastructure) Services.

5.3.2 IRS functionalities

5.3.2.1 Processing of C-ITS Messages:

Following operations should be conducted on the IRS:

1. Receive C-ITS messages – CAM, DENM from OBUs via C-V2X LTE interface. The incoming messages will be decoded and prepared for further processing.
2. Pre-process received messages:
    - check message validity – relevant message should have valid sender identification, time and spatial parameters
    - filter duplicate and not relevant messages – for reducing data amount all such messages will be rejected
    - organize message storage – for protocoling and logging purposes all relevant messages will be held on IRS for some previously determined time period
3. Organize CAM aggregation – for reducing amount of information which will be send to the ICS, some statistical data has to be created on IRS
4. Forward pre-processed information to ICS for further evaluation and processing – information should be converted into MQTT protocol format and then sent via MQTT/TCP/IP interface
5. Receive C-ITS information from ICS to build ITS messages, e.g.:
    - IVI messages for use case Shockwave Damping (SWD)
    - DENM for use case Road Works Warning (RWW)
6. Check and finalize message content - received ICS data will be controlled for validity and missing information will be completed
7. Message time and space handling – here the following operations are required:
    - organize message triggering
    - provide relevant repetition rate
    - control expiration duration parameters
    - provide relevant geographical dissemination parameters
8. Distribute actual messages to end-receiver via C-V2X interface

5.3.2.2 Management and Maintenance Functions:

Management and maintenance functions are comprised of the following operations:

1. Communication protocol handling – following protocol stacks run on IRS:
    - MQTT stack for connection with ICS
    - ETSI C-V2X stack for communication with OBU

2. Supervision of connection to ICS:
    - Establish, maintain and re-establish the connection
    - Monitor status of the link
    - Handle connection time outs
3. Organize remote IRS device management:
    - remote control & monitoring
    - software updates
    - device configuration
4. Provide data logging – old relevant (preliminary determined) data should be stored on the IRS for some (configurable) period of time
5. Provide error handling and recovery
6. Provide sufficient performance for appropriate data handling

5.3.2.3 Security Functionalities

Following security functions are required (detailed information see in Chapter 6.5 of this document):

- Authentication and authorization (device specific credentials)
- Integrity protection (signing of messages)
- Confidentiality (encryption of messages)
- Privacy (periodic change of public device IDs)

5.3.3 Interfaces

5.3.3.1 V2I Interface

This interface provides the communication between vehicles and the IRS in vicinity (in case they are equipped with OBUs supporting C-V2X). In the ConVeX project, the V2I Interface will use the LTE Rel. 14 C-V2X PC5. Further details are given in section 6.1 of this document.

5.3.3.2 ICS Interface

This interface should provide a reliable duplex connection between IRS and ICS. In the ConVeX project, the MQTT-protocol via TCP/IP will be used.

MQTT stands for Message Queue Telemetry Transport. It is a publish/subscribe model based, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal for the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are of concern.

The publish/subscribe pattern (pub/sub) is an alternative to the traditional client-server model, where a client communicates directly with an endpoint. However, pub/sub decouples a client, who is sending a particular message (called publisher) from another client (or more clients), who is receiving the message (called subscriber). This means that the publisher and subscriber do not know about the existence of one another. There is a third component, called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly.

MQTT uses subject-based filtering of messages: each message contains a topic, which the broker uses to find out if a subscribing client should receive the message or not.

In order to handle the challenges of a pub/sub system, MQTT has Quality of Service (QoS) levels. It is easily possible to specify that a message gets successfully delivered from the client to the broker or from the broker to a client. But still there is the chance that nobody subscribes to the particular topic. If this is a problem, it depends on the broker how to handle such cases. In order to mitigate the inflexibility of topics, it is important to design the topic tree very carefully and leave room for extension for use cases in the future.

In the ConVeX project, according to MQTT data exchange paradigm, the following communication model is used: on ICS the MQTT broker will be running, on IRS, correspondingly publisher and subscriber clients. Thanks to such an organization, it is possible to provide data exchange in both directions.

The data flow will be implemented as a set of pre-defined MQTT topics. Some topics will describe messages according to ETSI data models, others will serve for managing and maintenance functions. To allow generic message filtering, the topics will be built in a hierarchical manner.

Exact topic structures and communication parameters of the MQTT interface will be specified later and are out of the scope of this section.

5.4 System for Vulnerable Road User

5.4.1 General Issues and Technical Challenges

Vulnerable Road User (VRU) refers to any traffic participant other than vehicle drivers or vehicle passengers who are exposed to a very high risk of death or serious injury when getting involved into a road accident. This includes pedestrians, pedal cyclists, motorcycle riders and road workers.

We denote the C-V2X enabled portable equipment carried by a VRU as Vulnerable Road User Equipment (VRUE).

There are two different types of application scenarios for VRU:

1) Vehicle drivers receive warning alerts when VRUs are detected in their proximity to pose a risk of a collision
2) The VRU receives warning alerts

These two types of application scenarios may occur independently or in parallel. However, in real traffic scenarios the first use case appears to be more important since normally it is the vehicle that can react faster and even react automatically when using ADAS with automatic breaking. The second above use case might be hard to accomplish successfully in practice, e.g., for pedestrians carrying the VRUE in a pocket or bag, which might impact the GNSS reception, as well as PC5 sidelink communication, or simply the pedestrian pays no attention to the alerts.

However, already the first application scenario of warning vehicle drivers of VRUs in their proximity has a great potential to reduce the number of accidents and save many lives.

Besides those considerations, there are also some technical challenges involved as listed below:

- increase of V2X communication traffic due to permanently broadcasted CAM due to potentially huge number of pedestrians: VRUE likely needs intelligence to detect the environment and send CAMs only when the VRU is at risk of an accident (e.g., when getting into proximity of a street, or when receiving CAMs from cars around)
- power consumption of C-V2X functionality
- position location accuracy, due to
    - (i) non-optimal placement of GNSS antennas (e.g., when VRUE is carried in a pocket or bag)

- (ii) unavailability of information drive DR algorithms (estimated speed and direction of movement)

5.4.2 VRUE Design Options

The most obvious option for the design of a VRUE is to build a smartphone based on a C-V2X-capable chip platform. This is likely feasible at affordable extra cost for the consumer, provided the demand for such C-V2X enabled smartphones can be expected to become sufficiently large. If the extra cost of this feature is regarded reasonable and fair in view of the added value of the V2X services, it is likely that the market demand for such smart phones develops fast.

A C-V2X enabled mobile phone would be suitable for all types of VRUs. However, the additional power consumption has to be evaluated since in the foreseen "C-V2X operation" the device would all the time transmit the periodic CAM and permanently listen for transmissions of other traffic participants. This is very different to the duty cycles of a mobile phone in the legacy mobile networks.

For e-bikes and motorcycles the full integration of VRUE into respective vehicle appears to be feasible. In these cases, battery consumption is not a severe issue, and also optimization of GNSS antenna position is feasible. Potentially the integration into portable bike computers is feasible as well.

Also dedicated devices for Road Workers might be a viable option, maybe with a bulkier design that enables the usage of bigger batteries, or also for a kind of nomadic use, where the worker places the VRUE e.g. on the operated machine or even at the ground close to the actual working location.

5.4.3 VRUE in ConVeX

For the ConVeX project, the VRUE will be implemented using the same communication platform which is employed in vehicular and roadside ITS stations.

It will consist of the following items:

1) C-V2X DP providing the VRU application and P2V (and V2P) functionality
2) CCARD providing network connectivity (especially required for logging of measurement data on the remote measurement server)
3) Battery powering the CCARD and C-V2X DP
4) Antenna system
5) Tablet computer (or smart phone) connected via WiFi with the C-V2X DP providing VRUE monitoring and control functions

The first three items are planned to be placed in a small backpack. The antenna system can be designed and placed to emulate different user scenarios. For instance, for emulating of a pedestrian VRUE, an antenna system equivalent to the one integrated in a mobile phone may be employed. For emulating a C-V2X enabled bike, the antenna can be designed and positioned as appropriate for such a scenario.

5.5 4G/5G Test Network

The Ericsson 5G Testbed will be used as cellular network. It is a 3GPP-compliant platform that currently uses LTE radio, and offers several advanced 5G capabilities:

- **Network Slicing**: Parts of the network can be dedicated to and optimized for an application, and isolated from the rest of the physical network.
- **Distributed Cloud**: Due to a local breakout of the mobile core network at the radio site, cloud services can be executed closer to the user, enabling lower latencies and reducing the risk of congestion.
- **Virtualized Core Network**: All core network nodes are virtualized (HSS, MME, PCRF, Serving GW & PDN GW in Figure 5.5-1), and can thus be scaled depending on the demands, easily moved to other places, and nodes can be easily instantiated when needed, and shut down when not needed.

Figure 5.5-1 depicts the setup of the cellular network.

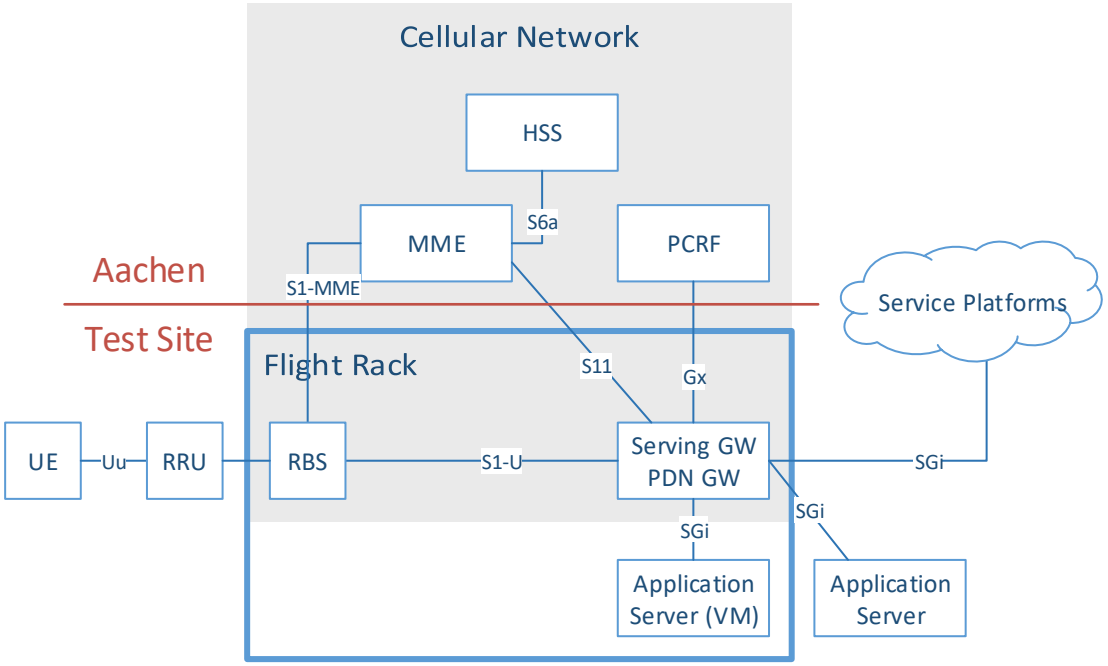

Figure 5.5-1: The cellular network is partly deployed in Aachen, and partly deployed at the test site, where it is possible to host services at a local breakout of the network.

In the testbed, the core network is mostly split into user plane (part of the network that deals with user data) and control plane (part of the network for pure signaling): The major part of the control plane resides in Aachen, from where the network is managed, while the user plane is instantiated at the test site, in a flight rack, containing several blades for instantiating the virtual core network functions that are needed. Aside from that, it contains a firewall, a switch, and the radio base band unit on dedicated hardware, plus a few O&M components.

The core parts of the control plane are:

- The Home Subscriber Server (HSS), which is basically a database of all authorized UEs, where authorization is done based on SIM cards in the UEs. For accessing the Ericsson testbeds at A9 and in Rosenheim, special SIM cards provided by Ericsson have to be used. These are configured for only these testbeds, and do not provide access to any other mobile networks.

- The Mobility Management Entity (MME), which takes care of signaling needed for mobility & security, among other things.

The core parts of the user plane are:

- The serving gateway (Serving GW or S-GW), which handles handovers between RBSs and access technologies (e.g. between LTE and UMTS).
- The packet data network gateway (PDN GW or P-GW), which is the termination point of all user data IP traffic, and the interface to external IP networks. At the PDN GW, application servers can be attached, either using virtual machines in the flight rack, or separate machines attached via ethernet cables.

Nodes of user and control plane in the core network are connected and communicate via an IPSec tunnel between Aachen and the test site.

Radio access nodes, containing both user and control plain functionality, are:

- The Radio Base Station (RBS), which is the radio access node of the network, and acts as a baseband unit, whereas modulation and power amplification is done in Remote Radio Units (RRUs). It also acts as a termination point towards the UE (User Equipment) below IP layer, while at the same time handling the backhaul communication towards the gateways, and control plane communication towards the MME. The installations of the testbeds in Rosenheim and at A9 use a setup with multiple radio units & antennas, which are all served by the same RBS in the flight rack at the respective test site. Therefore, there are several cells per test site implemented, as described further below.
- The User Equipment (UE) is the client device accessing the network, e.g., a smartphone, or in the scope of the project usually a vehicle equipped with a corresponding modem and a SIM card.

Finally, the PCRF (Policy and Charging Rules Function) is an optional control plane entity that can be used for setting up QoS (Quality of Service) rules for different services. A large number of possibilities exists for the relative prioritization of services, or defining guaranteed resources for a service. For handling the different QoS classes, the QCI (QoS Class Identifier) mechanism is used, where each QCI is associated with a number of parameters, such as a maximum packet delay budget, or relative priority handling. Among the standardized QCIs, several are designed for V2X services.

A more extensive introduction to mobile radio networks can be found at [2] or [3].

The virtualized environment also enables the flexible implementation of advanced techniques in the core network such as Network Slicing, as illustrated in Figure 5.5-2, where multiple packet gateways are instantiated and used for different services. In the current implementation at both test sites, three packet gateways are configured. Thus, three Network Slices can be used in the core network.

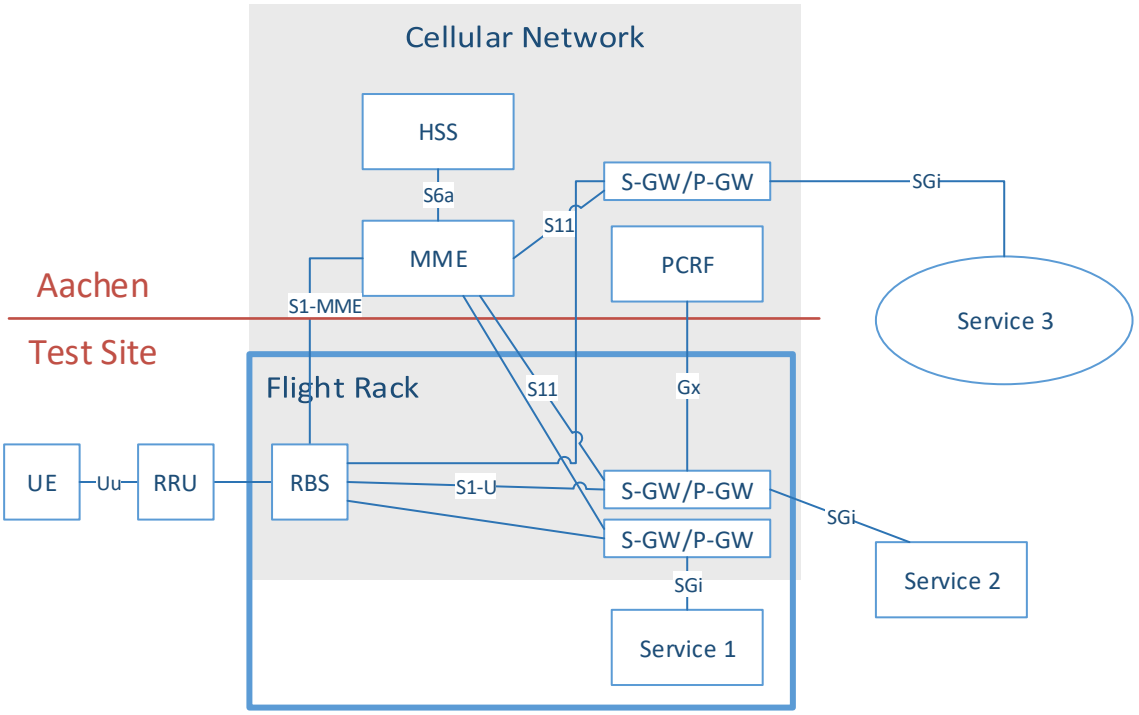

Figure 5.5-2: The network can be tailored to the service requirements, while at the same time isolating services from each other ("Network Slicing"), starting with multiple instantiations of the gateways.

The purpose of Network Slicing is to isolate services among each other, as well as to tailor the deployment to the service requirements: some services might have low latency requirements, or generate an extensive load, making a local deployment of the service endpoint in the network feasible. Other services might have less stringent requirements, and thus use a more cost-efficient central deployment. In addition to the separation in the core network for network slicing, radio resource partitioning can be used on the radio for allocating a specified share of the radio resources to a network slice dynamically.

Network services, such as VoLTE, are on the roadmap for the testbeds. For using such services, a dedicated UE integration setup is needed, and needs to be scheduled in advance with Ericsson staff. Furthermore, load scenarios are possible using a load generation feature in the testbed radio, which allows to generate load at discrete values, e.g. 30% or 60%.

5.5.1 A9 Testbed

The testbed uses Band 28b (700MHz), which has been granted as a test band by BNetzA, and which has been awarded to Telefónica Germany for commercial use in the future. The testbed provides 5MHz of spectrum for both UL and DL, without any MIMO configuration. 256QAM is only supported at the southernmost site, and at the other sites, maximum 64QAM is supported. Consequently, at the southernmost site, ~25Mbit/s of throughput can be achieved in DL. At the other sites, a maximum of ~19Mbit/s of throughput can be achieved in DL.

The A9 testbed consists of currently 6 sites, and covers approx. 30 km alongside A9 highway. The positions and directions of the testbed cells are indicated in Figure 5.5-3.

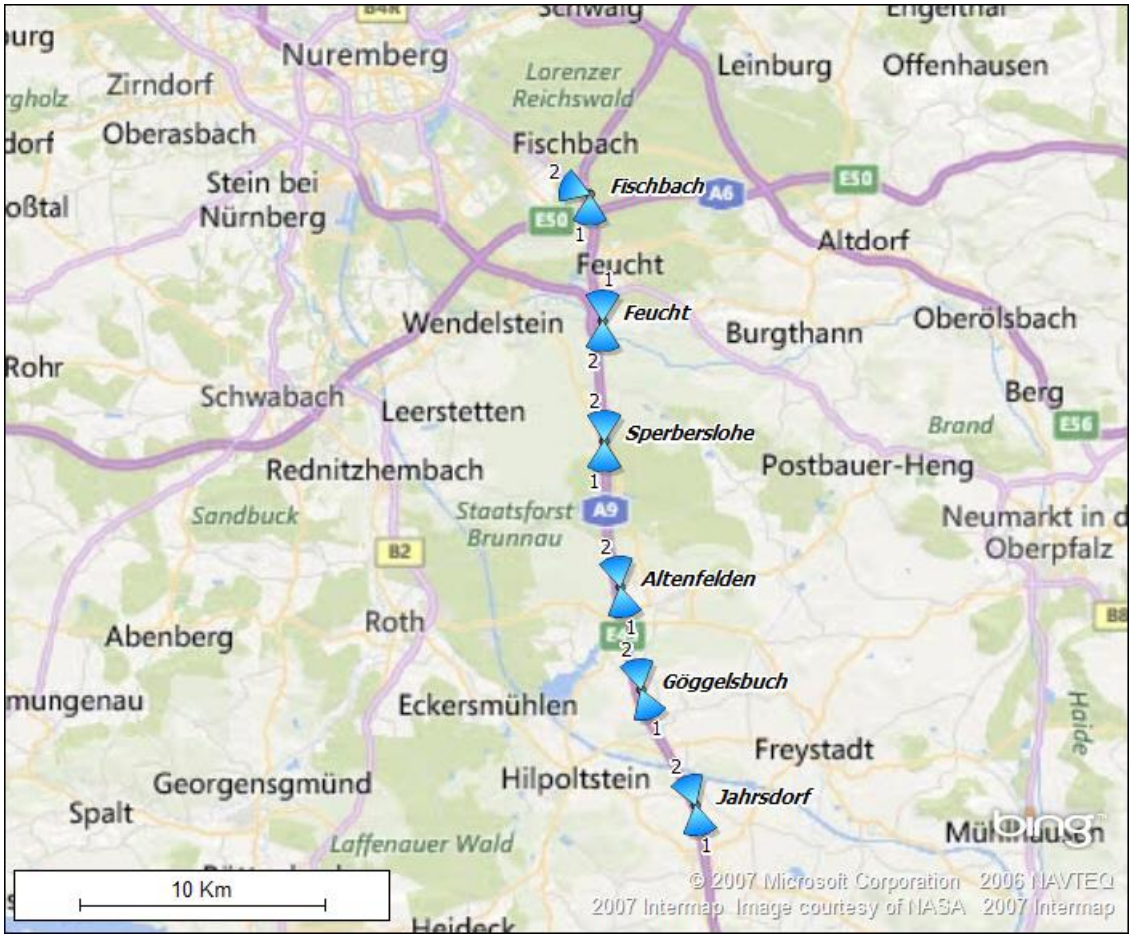

Figure 5.5-3: A9 Testbed with currently 6 sites, covering approx. 30km alongside A9 highway.

5.5.2 Rosenheim Testbed

The testbed uses Band 28b (700MHz) and Band 7 (2600MHz), which has been granted as a test band by BNetzA, and which has been awarded to Telefónica Germany for commercial use in the future. The network provides 10MHz of spectrum for both UL and DL, in each band, with 4x4 MIMO. Furthermore, 256QAM is supported. Consequently, ~200Mbit/s of throughput can be achieved in DL, in each carrier.

The Rosenheim testbed contains two sites with three cells each, and covers Rosenheim city center, where the area depends on the respective frequency band. The positions and directions of the testbed cells are indicated in Figure 5.5-4.

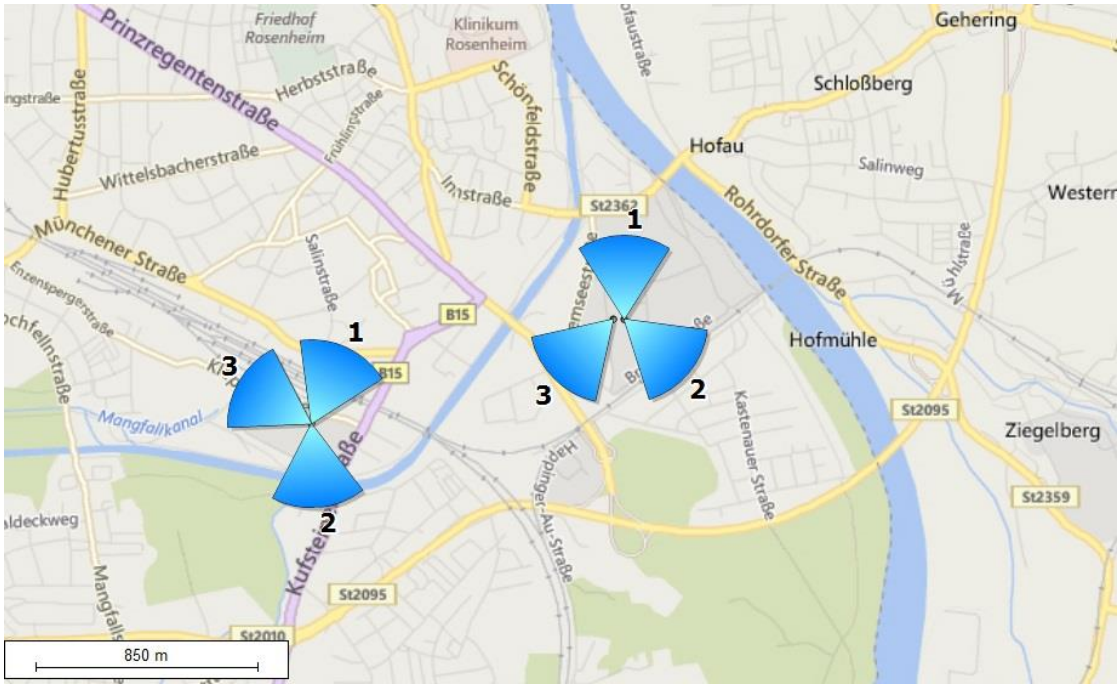

Figure 5.5-4: Rosenheim Testbed with currently two sites, with three cells each, covering the center of Rosenheim.

5.6 Cloud Servers

5.6.1 Traffic Control Center

SWARCO provides a "Virtual Traffic Control Centre (V-TCC)" and a set of "Virtual Variable Message Signs (V-VMS)" – represented by "V2X Roadside Units (RSU)"

The Virtual Traffic Control Centre acts towards the demonstration system like a Traffic Control Centre (TCC) on German Motorways which are typically structured like this:

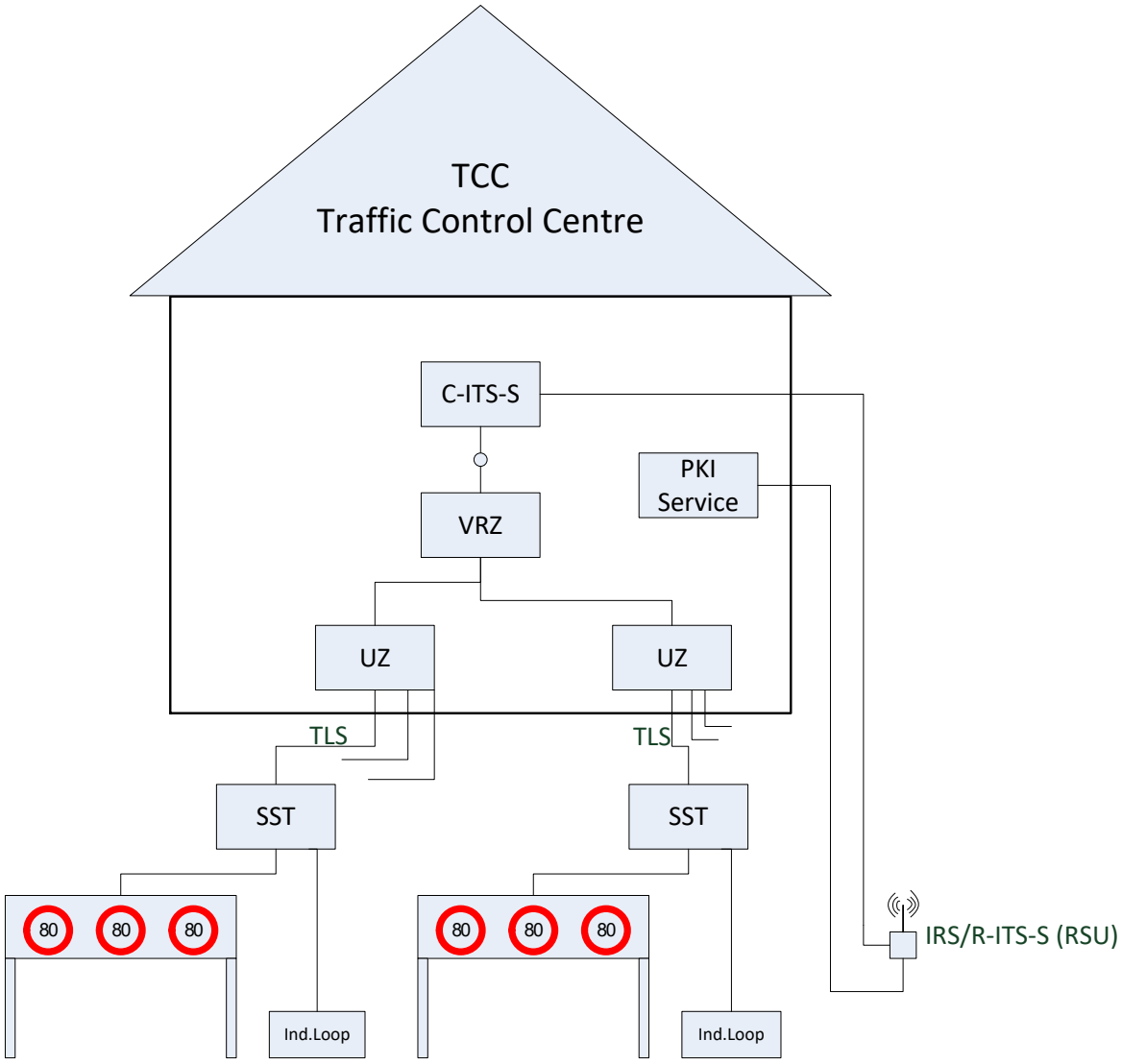

Figure 5.6.1-5 Structure of a Virtual Traffic Control Center

The Traffic Control Centre TCC is a hierarchical system composed by:

- The main integrating control centre "*Verkehrsrechnerzentrale* VRZ"
- and several sub-centres „*Unterzentrale* UZ"

The sub-centres (UZ) are linked to the field, in which all infrastructure locally (few 100m) is collected in a control cabinet "*Streckenstation* (SST)" (German for: Section Station). From there elements such as gantries "*Anzeigequerschnitte* AQ" (German for: gantries) or sensor systems such as loops are linked. All linkage in the field is achieved by the standard for connecting outstations „*Technische Lieferbedingungen für Streckenstationen*" (TLS), in which also the data is defined.

To extend cooperative systems functions, the TCC central system is extended by layer elements, having different names in different countries:

- ITS Central Station / Central ITS stations: **ICS** / C-ITS-S is the element to link the new cooperative communication units (as outstations) to the TCC
- An element for introducing the Public Key Infrastructure (**PKI**) service must be foreseen. It is likely that there will be only one (inter-)national element for this. Every cooperative ITS unit needs a link to the security / PKI system in order to get regularly updates on trusted keys for signing messages.
- To achieve traffic functions such as Shockwave Damping (**SWD**) extensions are done on "Unterzentrale UZ" level.

In the field, the extension is composed by the component:

- ITS Roadside Station / Roadside ITS stations: **IRS** / R-ITS-S (RSU)

The IRS is typically added next to a field system, using available cabling and power. Depending on a national architecture, the IRS can be added independently from the local system with an own direct link to its corresponding central component (C-ITS-S) for the traffic functions and a link to the (international) PKI.

For the demonstration, the test area (A9) is shown on the map and the map holds Variable Message Sign (VMS)-gantries, showing the sign content (speed-limits, warnings, lane closures). The virtual TCC has the functionality to "switch" the sign content (as an operator action).
The demo system supports the use cases:

IVI:

When a sign content is changed (manually or by any automatic), the new sign content is distributed to the right "V-VMS", resulting in a sending of IVI messages.

Shockwave Damping (SWD):

A SWD algorithm (running in a "simulation mode") is continuously producing recommendations for a target speed using a configured sequence of road sections as if it was a real system. The speed recommendations are distributed to the right RSUs and are sent to the vehicles as IVI messages.

Road Works Warning (RWW):

The user can generate a "roadworks" in the Virtual Traffic Control Centre (V-TCC), which is sent to the RSU and delivered to vehicles as DEN message

For the demonstration, only some relevant elements for cooperative-systems are represented in a V-TCC:

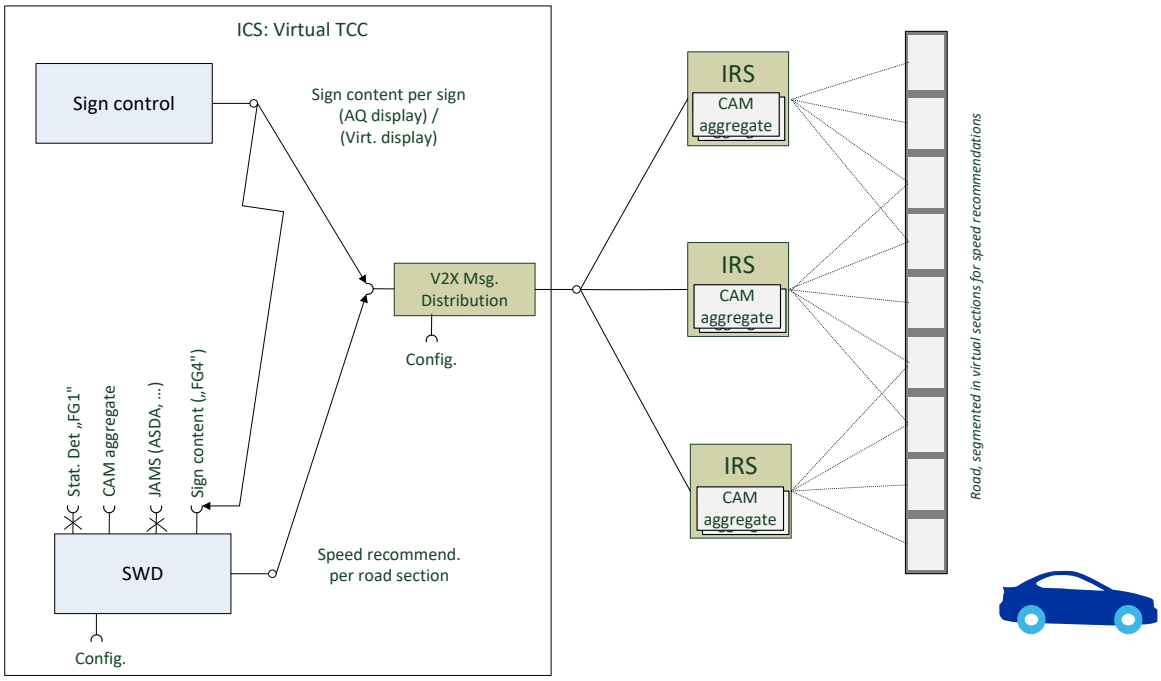

Figure 5.5.1-6 Structure of a Virtual Traffic Control Center

Figure 5.6.1-2 shows the schematic diagram of a Virtual Traffic Control Center (V-TCC). The Virtual TCC consists of following components:

- Sign Control,
- Shockwave Damping (SWD) and
- Vehicle to Everything (V2X) Message (Msg.) Distribution.

The SWD component is fed by the inputs:

- From vehicle measurements (aggregated CAM messages). The data can also be added by the simulation in ConVeX.
- Sign content (TLS “FG4” data). The Sign content originates from the gantries Sign Control and is used to determine the maximum allowed speed set by dynamic road signs. Also, potential lane closures or open hard shoulders can be deduced from this source. The data can also be fed by simulation in ConVeX.
- Detector counts (TLS “FG1” data). In ConVeX this data is not available from a real field installation. It can be added through the simulation.

Key system parameters are stored in the configuration (Config.). It includes:

- The road layout in terms of
  - Number of lanes,
  - Places of dynamic road signs / gantries
  - Places of detectors
  - Entry and exit slip roads
- Position of IRS units
- The layout of speed-recommendation-sections for SWD

The format follows the data model existing in real VRZ / UZ systems.

SWD produces speed recommendations (per section). Together with the road sign settings the information is distributed by theV2X Msg. Distribution component to the IRS, where the V2X standard messages are sent to vehicles. In summary, the distributed data comprises:

- Sign content to be shown on (virtual) dynamic road signs (equivalent to FG4 – Functional group “FG4” in the TLS Protocol: FG4 contains Sign Display) are converted to IVI messages.
- Speed recommendations per section (in form of IVI message elements)
- DENM warnings from the TCC (e.g. about roadworks)

In the direction from field-to-centre the ConVeX demonstration comprises the data:

- (Aggregated) CAM, used as detection / traffic status sensor
- DENM to inform the TCC about received warnings from vehicles

5.6.2 V2N Application Servers

In general, all data flows going over the V2N link can only leave the corresponding mobile network via a PDN gateway, so in order to leverage on the advantages of application servers close to the radio site, a PDN gateway must also be close to the radio site. The flight racks in the Ericsson test network come with multiple PDN gateways inside the flight rack, which are as close to the radio as possible. It is of course still possible to use a centrally deployed PDN gateway in Aachen, be it as a reference, or to dedicate the local PDN gateways to different services. These two options are called local breakout and central breakout, and are depicted in Figure 5.6-7. In the following, we will focus on the local breakout, although the same options can be realized using a central breakout.

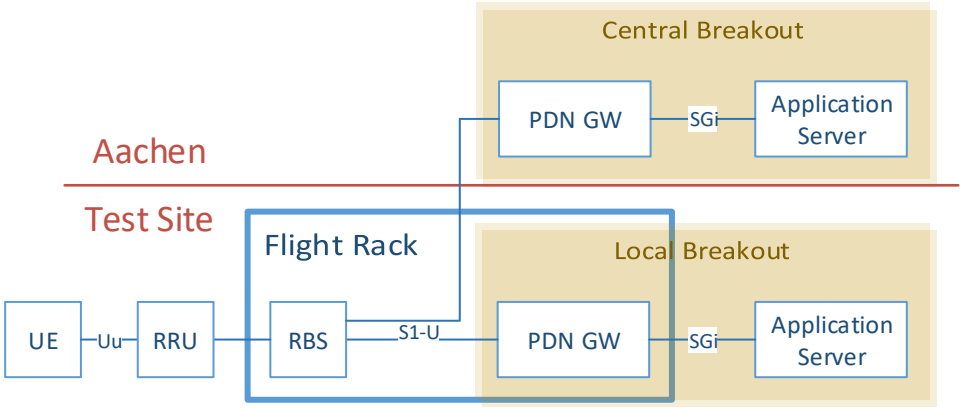

Figure 5.6-7: Local breakout vs central breakout

Three general options for deploying an Application Server reachable via the V2N link are offered by the test network. They are depicted in Figure 5.6-8, and explained in more detail in the following.

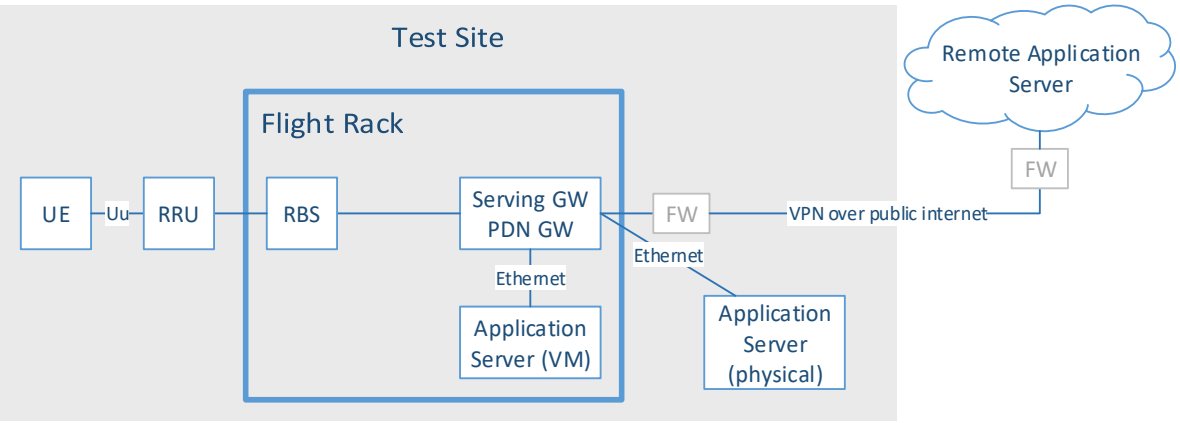

Figure 5.6-8: Application servers can either be hosted as a VM in the flight rack, or on a physical machine next to the flight rack connected via ethernet, or in a remote location with access via public internet, possibly using a VPN tunnel.

5.6.2.1 VM in the Flight Rack

An application server can be hosted in a VM inside the flight rack. These VMs are managed by Ericsson, where different operating systems are possible, as well as giving access to the VMs from outside Ericsson, but additional integration effort is needed for that. The VMs can be configured with different performance levels (number of vCPUs, RAM, storage), with the overall capacity of the hardware in the flight rack as a limitation.

5.6.2.2 Physical Machine Connected to Flight Rack

An application server can also be hosted in a physical machine near the flight rack, connected to the switch in the flight rack via ethernet. A port on the switch in the flight rack can be configured to be used for connecting external machines, where the IP configuration of that machine needs to be set up according to the VLAN configuration in the flight rack. Public internet can be made available on that machine, as well as a connection to remote sites by different partners.

In general, this option does not perform better or worse than option 1, w.r.t. communication. However, it can be used to overcome capacity limitations on the integrated data center, as well as for connecting special hardware, such as GPU systems.

5.6.2.3 Remote Server

Finally, application servers can be hosted on a remote server, e.g. located in a cloud-based service platform. Endpoints that can be reached via public internet are trivial to connect to. Aside from that option, access to remote application servers can be protected by setting up a VPN tunnel, usually IPSec. For that, the firewall inside the flight rack, which builds up the IPSec tunnel to the control plane (and additional optional components) of the network in Aachen must be configured to also build up this additional VPN tunnel, and a firewall is needed on the other side of the VPN tunnel to terminate it before the remote application server. Configuration of firewalls for IPsec tunnels need to be aligned for both ends of the VPN tunnel and shall preferably be configured by one party, where Ericsson can assist on this when requested.

Usage of this option will affect characteristics, including but not limited to latency, which shall be taken into account for considered application services.

5.6.3 Measurement Server

The C-V2X Server provides backend services to support data processing services to illustrate the C-V2X's performance, strength and advantages. It is a platform to demo C-V2X use cases and view supporting detail data. In this section the server's architecture will be described to elucidate how the collection of data, calculation of KPIs and other relevant measurements for the trial are made possible.

The C-V2X server sits in the hub position to connect all components in trials: vehicle, RSU, pedestrian and web client:

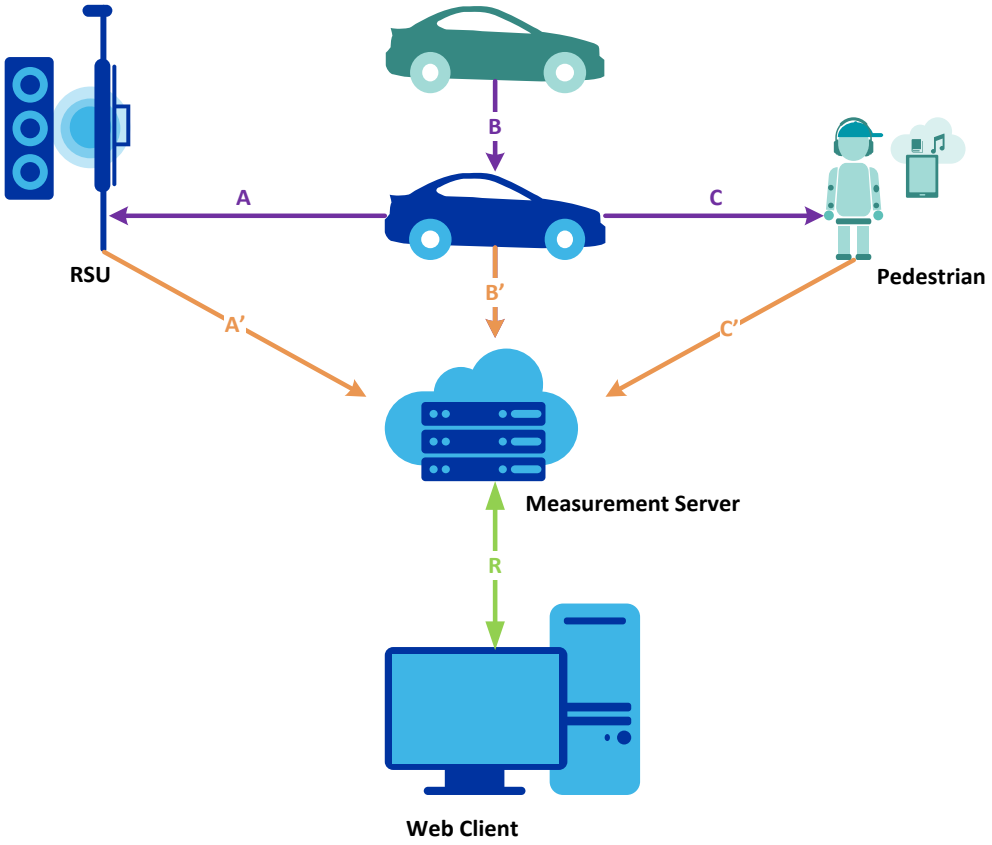

Figure 5.6-9: Measurement Server Architecture. ITS messages (A, B, C) received by any entity of the system are encapsulated and forwarded to the measurement server (A', B', C'). Same for sent ITS messages (not shown for simplicity). Web client and measurement server communicate with Requests and Responses (R)

Pedestrians, vehicles and RSUs communicate all transmitted and received messages to the server over LTE or fixed internet access, which then stores them by having an integrated database. KPIs are generated on the analytics module, also present in the server and can be seen by the user over a web client.

5.6.3.1 Functionality

The server is designed to provide certain set of functionalities, among which the minimum basic features are:

- Server provides the user with a real-time view of the system showing actual status of active devices including their locations. Filtered views based on use case and/or device and display of "flight paths" will also be available.
- Server generates KPI reports in a chart format with the possibility to apply filtering based on device and use case.

Additionally, some potential enhancements foreseen are:

- Server could provide the possibility to enable configuration of the devices.
- Server could provide use case / test configuration control such as selection of time window, start and end time, use case type, device, etc.
- Server could support query of completed use cases, as well as filtering use case data by device. User would then have the option to playback use case data getting a visualization on the map.

5.6.3.2 Architecture

Each instance connected to the server plays a specific role in the system. This subsection describes how the server interacts with those components and how the user accesses the data over the web browser / user interface (UI).

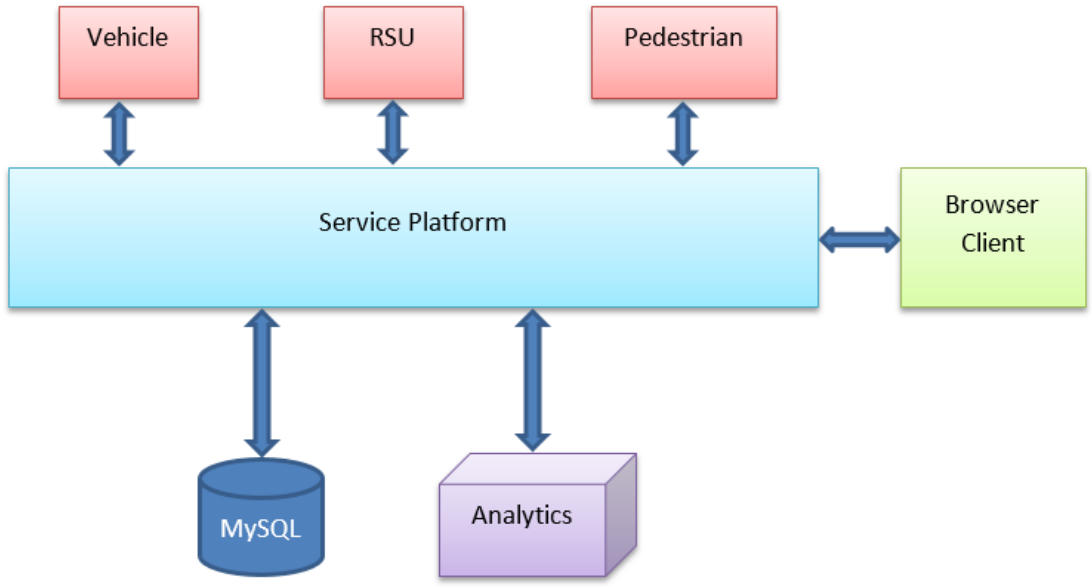

Figure 5.6-10: Service Architecture Building Blocks

Vehicle, RSU, Pedestrian

Components are gifted with the capability to communicate with the server over internet. Each one of them individually sends both transmitted and received messages to the server, as well as component-specific pre-defined alerts. These alerts and V2V, V2I, V2P communication exchanged between those components compose part of the message sent to the server in a sort of logging manner. That way the calculations necessary for the KPIs can be done.

Browser Client

This is the interface the user interacts with and is where the metrics and KPIs can be visualized. The browser client is connected to the Internet using fixed line access for instance. Measurements and KPIs are displayed on the UI per user request, i.e. by clicking on the button to generate report. Currently, the browser JavaScript makes an HTTP query to a URL on the server, which then responds with a JSON message. The Figure 5.6-3 illustrates the procedure.

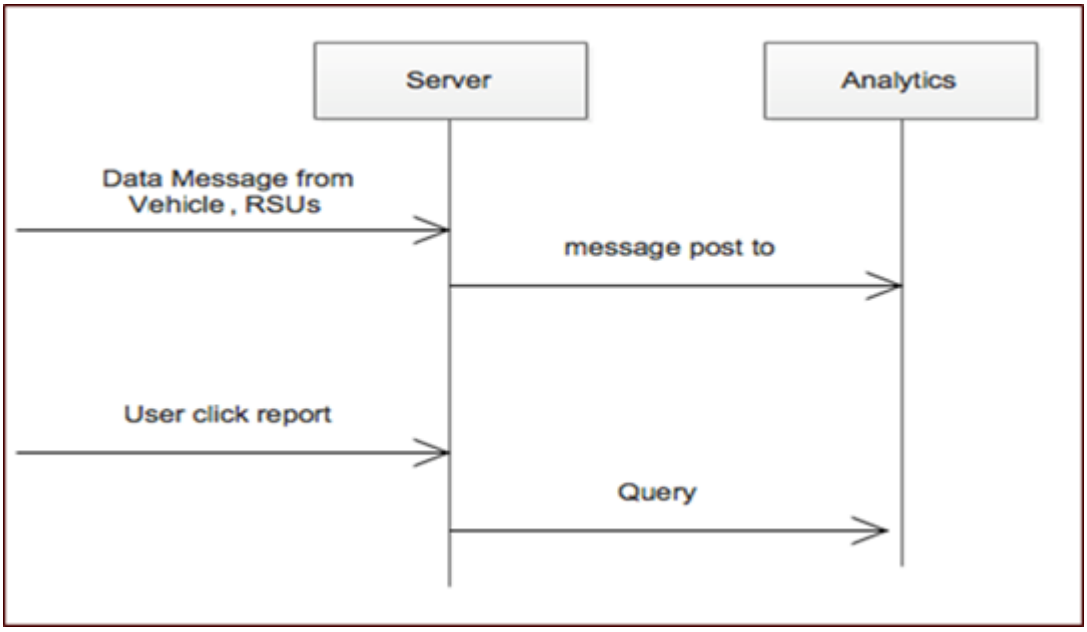

Figure 5.6-11: Sequence diagram

A preliminary example of the view is shown on the Figure 5.6-4.

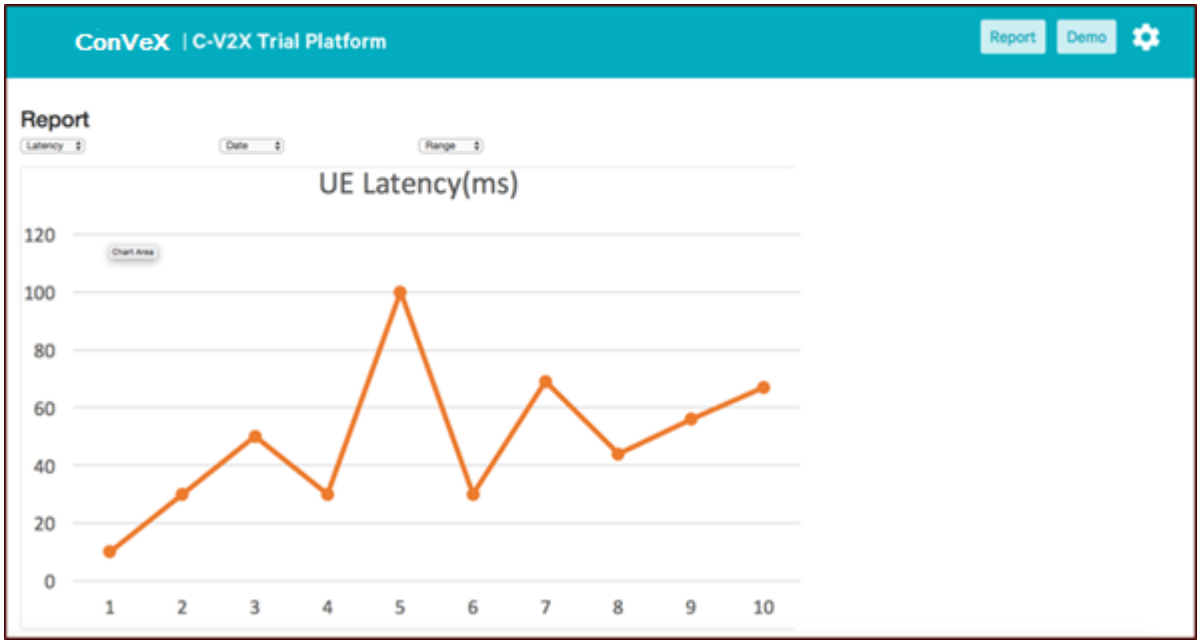

Figure 5.6-12: Latency graph on UI view

The UI will enable the view of KPIs defined in the ConVeX Deliverable D.1.1., some of which have been summarized in the Table 5.6-1:

Table 5.6-1

KPI	Description
Latency	Time difference between the moment a message at the originating application is timestamped to the instant it is received by the destination application. Originating message timestamp is an instant GPS fix calculation.

Reliability	PRR – Packet Reception Rate at the receiving application Inter Packet Arrival Rate at the receiving application
Range	Distance between Source and Destination entity for each received message at the Destination Application
Speed	Car Speed from V2X message

Analytics

An Analytics module is included to support post-processing of the data. It is composed by two submodules: Cassandra and Spark.

Cassandra is a high-performance data warehouse that stores C-V2X messages.

Spark works integrated with Cassandra, hosting some functions that enables processing of Big Data. Spark module is responsible for initial validation and post-processing of the data. It scans, transforms and performs calculations on them to then stores the results. This way, the values can be easily retrieved from existing tables and then plotted on the UI, when a query comes from the browser.

Database / MySQL

Temporary database for transmitted and received alerts such as a breaking event, switch of left/right indicator, etc. Interacts essentially with the web client to display such alerts on a map.

5.6.3.3 The Server Architecture and the V2N Communication

The server architecture shown is designed to measure KPIs related to V2V, V2I, V2P that are running on PC5. Messages sent over V2N would also need to be logged on the network infrastructure side. In the ConVeX project, this communication would happen over the LTE Uu interface, thus eNode B logging would be needed or other means implemented by Ericsson, that could provide the needed information (application running on Ericsson server).

Another application of V2N communication is the use of the network to broadcast V2N messages in a certain area by means of eMBMS, which is out of the scope of this project. Nevertheless, a possible way to “simulate” this, would be to simply send the messages on a unicast basis to the subscribed users in its coverage area, which could be evaluated.

In this sense, latency and reliability could be measured for Uu evaluation, also to allow the comparison with PC5. Moreover, testing different settings such as modifying the prioritization of V2N messages would provide better understanding of the system and help to calibrate it. However, the measurement server described in this section will not support it, but instead Uu evaluation would have to be performed on the network side as mentioned before.

5.6.3.4 Provisioning and Authorization

To avoid that data from unwanted sources is disturbing the processes of the measurement server, some level of authorization is needed.

Given the trial scope is limited to known devices and users, provisioning and authorization are designed to simplify authorization but with needed security controls:

- Admin initiates provisioning by registering a device, server will generate an access code.
- Client(device) will call server to verify with access code, and get a code64 encoded token.
- For each request from client(device), this token is attached to the messages.

- The token contains timestamp, and will expire in 24 hours (configurable). For each client such newly issued token is to be used onward.
- Server will maintain 3 past tokens, and allow those 3 tokens to be retrieved and force new tokens to be generated.

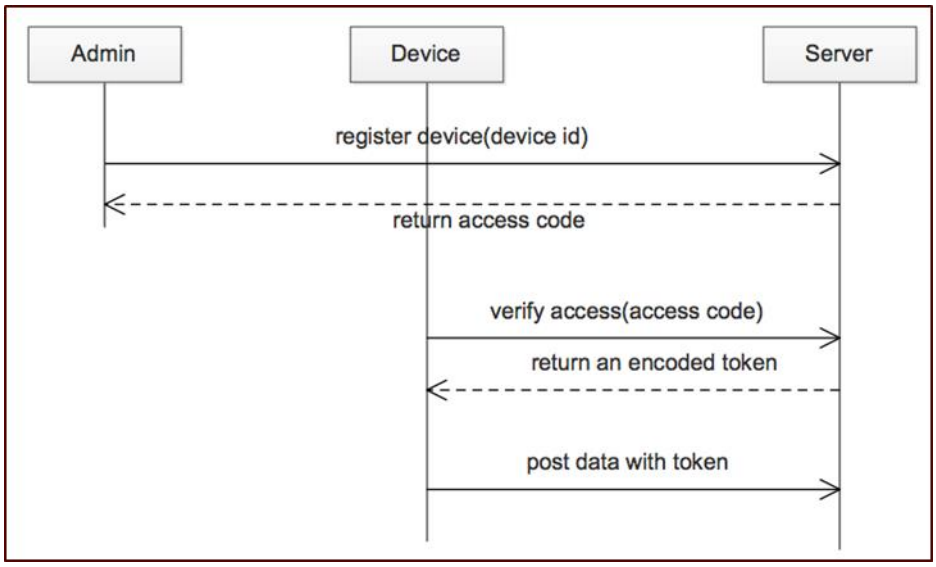

Figure 5.6-13: Registration

If the token of a device is expired or the device loses the token, it can gain access back easily by verifying with access code again.

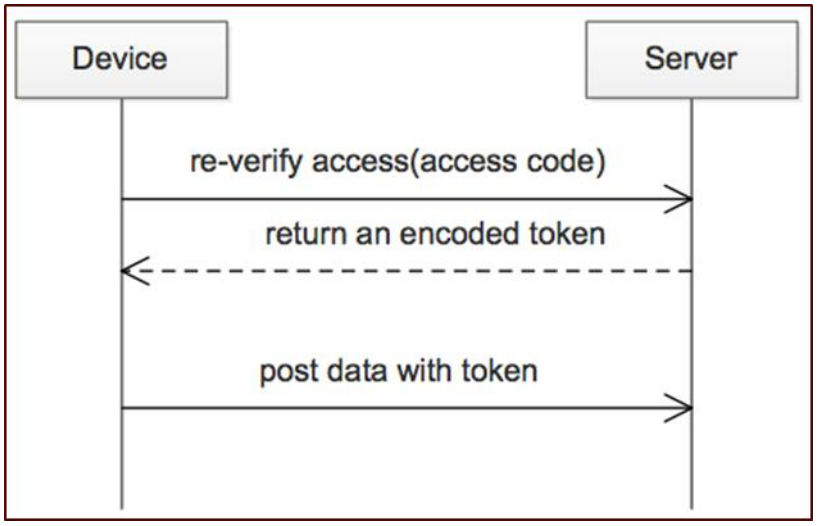

Figure 5.6-14: Re-verify Access

5.6.3.5 Messages

As mentioned previously, the server takes as input alerts and all transmitted and received messages from vehicles, pedestrians and RSUs and that is done in order to generate KPIs. Those messages are divided into two types: data messages and event messages.

5.6.3.5.1 Data Messages

Data messages are ITS messages and are used for analytics. Whenever an ITS message is transmitted, the component of the system responsible for its transmission has to notify the server that it was sent. Analog process is done when a message is received.

The data message container comprehends parameters derived from the ITS message transmitted/received itself, along with some other fields. An example of the data message content can be seen on table 5.6-2. Be aware that the content is not limited to what is shown.

Table 5.6-2

Token
Message_id (mandatory)
Current_Device_id (mandatory)
Origin_device_id
Timestamp (mandatory)
Received_timestamp
Position: **latitude (mandatory), longitude (mandatory)**, heading, accuracy
Speed
Use_case_category
Message_count (mandatory)

The server supports data messages with 100ms periodicity.

5.6.3.5.2 Event Messages

Event based messages are alerts, warnings or any other internal information from the vehicle, RSU or pedestrian. Additionally, vehicle position updates with 1s periodicity for those in motion are considered event messages.

The content of an event message can be seen in Table 5.6-3 below but is not limited to what is shown:

Table 5.6-3

Token
Device_id (mandatory)
Timestamp (mandatory)
Position: **latitude (mandatory), longitude (mandatory)**, heading, accuracy
Msg_id (mandatory)
Event_type (mandatory): warning, information, position_update
Enable (if event was enabled or disabled such as headlights on)
Speed

6 Communication Protocols

6.1 PC5 Interface

The PC5 interface was introduced in 3GPP Release 12 for the LTE direct Device-to-Device communication. It is also referred to as “Sidelink”, in contrast to the classical mobile communication from a base station to a device and vice versa, which are called “Downlink” and “Uplink”.

In Release 14, 3GPP modified and optimized the PC5 Sidelink for V2X according to the specific challenges for this kind of communication. E.g., a design target was the support of relative speeds of up to 500 km/h, high device density as well as low latency requirements.

Per definition this link utilizes TDD (Time Division Duplex), here with a complete uplink assignment, i.e., the transmissions from the UEs are in this regard seen as uplink. The system bandwidth can be 10 MHz or 20 MHz, and as operating frequency band the reserved ITS band in 5.9 GHz spectrum is foreseen (Band 47, with an overall bandwidth of 70 MHz from 5855 MHz to 5925 MHz). For the ConVeX trial, it is planned to use the upper part between 5905 MHz and 5925 MHz, also referred to as ITS-G5D band.

The assumption is that the actual payload transferred are ITS messages, which are periodic in nature with periodicities of e.g. 100 ms or 1 s, and with a size in the range of 300 bytes, plus some event-driven messages that go up to 1200 bytes [4].

The LTE standard allows two different transmission modes on PC5: the so-called “in-coverage” mode, where an eNode B is scheduling the actual transmissions of the UE (Sidelink Transmission Mode 3), as well as the “out-of-coverage” mode, where there is no eNode B scheduling involved, but an autonomous resource selection performed by the UEs (Sidelink Transmission Mode 4). For the ConVeX trial, Transmission Mode 4 will be used, which is also envisioned as the most relevant mode of operation for future real world usage. An advantage of this mode is, e.g., that it still works in areas where no eNode B coverage exists.

For the rest of this section, Transmission Mode 4 (TM4) is assumed (some parts are anyway common between TM3 and TM4).

For synchronization GNSS (e.g., GPS) is used, i.e., all UEs are synchronizing relative to the GNSS and are thus time-synchronized with each other.

6.1.1 Specifics of the Physical Layer

For C-V2X Sidelink, four Demodulation Reference Signal (DMRS) symbols per subframe are used (legacy LTE uplink transmissions use two DMRS symbols). This increase of DMRS allows a better detection, which is important to be able to cope with the potentially high relative speeds of transmitting and receiving UE.

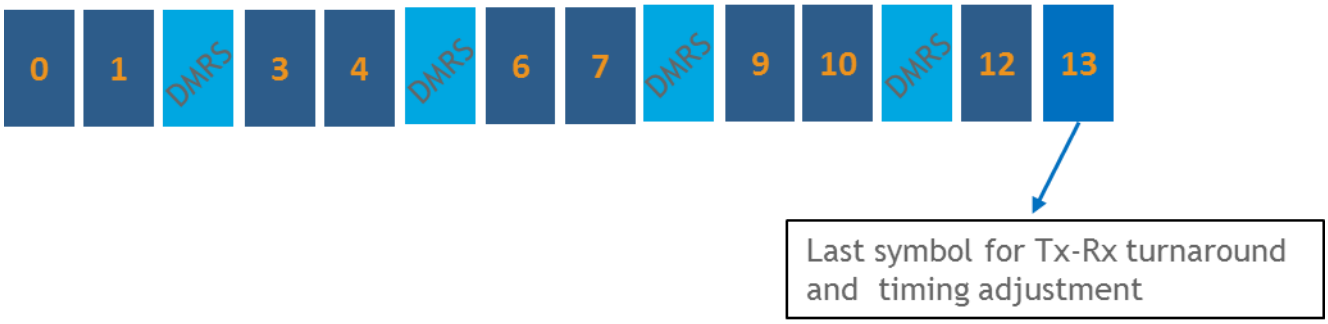

Figure 6.1-1 Structure of a C-V2X Sidelink Subframe

Apart from the position of the DMRS within the subframe, Figure 6.1-1 also shows that the last symbol in each subframe is not used for transmissions, but left empty to give time for the turnaround between sending and receiving in the next subframe.

Since the V2X messages are broadcast in nature, there is no CQI involved or used, and there is also no feedback for successful or not successful receiving of a transmission, i.e., no physical layer ACK/NACKs. It is possible to configure (via RRC) one "blind" retransmission: in that case the sender would always do a transmission followed by a retransmission of the same content.

6.1.2 Physical Channels

Two of the channels that were newly introduced for Sidelink communication are utilized for C-V2X:

The **PSCCH** (Physical Sidelink Control Channel) for the control information, and the **PSSCH** (Physical Sidelink Shared Chanel) containing the actual data part. Specifically changed for C-V2X is, that they are transmitted in the same subframe, which brings a latency reduction. 3GPP gives the two configuration options that they are adjacent in frequency or not. For the ConVeX trial, they will be adjacent (the support of the non-adjacent allocation is actually optional for a UE, and furthermore no advantages of such an assignment are currently seen).

The PSCCH has a fixed size of two RBs (Resource Blocks) and is transmitted with QPSK modulation. It carries the Sidelink Control Information (SCI) basically describing the data channel it belongs to:

- Priority in terms of PPPP value (ProSe Per Packet Priority): value between 0 and 7, with 7 representing the highest priority
- Resource Reservation: indicates the intended periodicity of the upcoming transmissions, e.g., 100ms or 1000ms. This would be determined by the higher layers, i.e., the ITS stack
- MCS value of the data
- Location of the PSSCH (starting RB in frequency space, size)
- Time gap for retransmission (i.e., in case of the setting with a blind retransmission, when in time will this happen. This way the receiver gets to know which data transmissions it can potentially soft-combine). Maximum is 15 subframes.
- Retransmission Index (indicates original transmission or retransmission)

PSCCH is transmitted with 3 dB higher power than the PSSCH, since it should not be the bottleneck on the receiving side (and if the control channel cannot be read, the data channel cannot be decoded anyway).

The PSSCH is variable in size (as described within the SCI) and can be transmitted with QPSK or 16QAM modulation (according to the MCS). As mentioned above, the actual data transported will be mainly the ITS messages.

6.1.3 Resource Pool

The Resource Pool defines locations of channels in frequency and time, that the UE can use for sending and receiving (TX pools and RX pools). In the frequency domain, subchannels can be defined. Here different options are possible in terms of number of subchannels and their size. Adjacency or non-adjacency of PSCCH and PSSCH would also be determined with the Resource Pool definition.

In principle 3GPP introduced the possibility to not use all resources for V2X to enable a mixed operation with legacy LTE, however specifically for the 5.9 GHz ITS band, such a mix is not foreseen, but rather all resources should be allocated to C-V2X. Similarly, also in the time domain multiplexing with LTE would be possible, but again not expected in this deployment scenario.

The Resource Pool definition is task of RRC – but since there is no eNodeB involvement here (eNodeB could broadcast the configuration in SIB 21), it could be either downloaded from a certain configuration server (which is a foreseen 3GPP option [5]) or preconfigured accordingly in the UE. The latter will be done for the trial.

6.1.4 Distributed Congestion Control

Since there is no higher instance for distributing the resources in TM4, the concept of distributed congestion control is implemented. That means, if a UE wants to start transmissions, it is required to sense the resource usage for 1s. During this time, it ranks the resources according to the energy received (measuring the sidelink RSSI per subchannel). It would select the ones with low energy as potential candidates for the transmission. Furthermore, the UE attempts in a 100ms window to decode the PSCCH and PSSCH, and exclude candidates, where the PSSCH-RSRP exceeds a defined threshold. Finally, it also checks the priority value (PPPP) and would avoid sending packets with a lower priority on a resource used by other users for higher priority transmissions. Note that the system is tailored for the periodic sending of ITS messages, i.e., when the UE found a good resource in frequency and time domain during the sensing period, it can be assumed that the situation stays similar for the future, as all participants would continue with their periodic transmissions using the same resources. So, also the UE that did the sensing to start transmissions found the most suitable resource and would keep using it. The actual periodicity of its intended transmissions is signaled in the Resource Reservation field (part of the SCI) as mentioned earlier. With this concept, the interference or resource usage among all UEs in the system is predictable and collisions can be avoided.

How long a resource can be kept by the UE is dependent on certain rules and parameters, including some random choices, that the UE has to do. In brief, it randomly picks a starting counter value from a defined range, and decreases the value by one with each transmission. When the value one is reached, the UE has to choose randomly a value between 0 and 1 and compare its choice with a parameter provided by the system. If chosen value is higher than the parameter, it can still keep the resource and moves back to the step of picking a new starting counter value. If the chosen value is lower, it is doing one last transmission on the selected resource and is then doing a new resource selection similar to when initially starting the transmissions. The UE now takes the information from the measurement history of the last second, i.e., it can quickly pick a new best

suited resource. This also implies that the UE continuously has to measure and evaluate the sidelink RSSI per subchannel and the PSSCH-RSRPs, and keep record of them.

Forcing the UE to reevaluate the situation after some time takes care of the dynamic nature of the traffic, e.g., if the UE is installed in a car, this might have moved to a different area in the meantime with other participants present.

Another reason why the UE would need to give up the chosen resource is, when it is actually not transmitting – either for 1s in total or a (configurable) number of opportunities. This could happen if for some reason there is no data coming from the higher layers (ITS stack). Here the idea is, that if the UE is not transmitting for a while, other UEs might have spotted that resource in time and frequency as the most suitable for their own transmissions, and if the original UE then starts using it again, a collision would be happening.

6.1.5 Other Layers: RLC and PDCP

In line with the already mentioned broadcast nature of the data transfer on PC5, RLC is running in unacknowledged mode and stays unchanged compared to legacy LTE.

PDCP as the higher transport layer would indicate for ITS messages, that non-IP data is transferred, and there would be no security functions activated – again keeping in mind that it is a one-to-many communication.

There is also the option to transport IP-based data over the PC5 interface, which might be utilized for non-ITS based data transfers and applications.

More details on PC5 being used for C-V2X can be found in the various 3GPP Release 14 Specifications (see e.g. [6] to [10]).

6.2 Uu Interface

6.2.1 Overall Uu Protocol Stack

LTE-based V2X communications can be realized over the uplink/downlink interface (Uu interface) or the sidelink interface (PC5) that is addressed in section 6.1. For downlink traffic on the Uu interface, both unicast and multicast/broadcast transport may be utilized to deliver the message within the area of interest. However, in the test network, multicast/broadcast is not supported.

For unicast V2N communication, the usefulness of LTE Uu is obvious, because a communication link from the vehicle to a server in the fixed infrastructure is necessary. However, also messages that originate from vehicles and are targeted at vehicles can, as an alternative to direct V2V communication link using the LTE PC5 link or DSRC also be conveyed via the Uu interface.

In this mode, each vehicle uses an uplink unicast connection to send V2X messages to a centralized or distributed server. In the downlink, the server can send V2X message to all vehicles located in a target geographic area. That is, each dedicated connection carries a copy of the message. In case of CAM messages, if the vehicle originating the CAM message has N neighboring vehicles, the downlink traffic volume is N times of that in the uplink. An advantage of communication via Uu interface over direct V2V communication is the longer range of the Uu uplink and downlink. A drawback maybe be increased end-to-end latencies.

The LTE stack consists of user and control planes. In the control plane, the Layer 3 is comprised of Radio Resource Control (RRC) and Non-Access Stratum (NAS) protocols. Layer 3 exists in the control plane only.

The LTE Layer 2 protocol stack is composed of three sublayers, as shown in Figure 6.2-1:

Packet Data Convergence Protocol (PDCP) layer: This layer processes Radio Resource Control (RRC) messages in the control plane and Internet Protocol (IP) packets in the user plane. Depending on the radio bearer, the main functions of the PDCP layer are header compression, security (integrity protection and ciphering), and support for reordering and retransmission during handover. For radio bearers which are configured to use the PDCP layer, there is one PDCP entity per radio bearer.

Radio Link Control (RLC) layer: The main functions of the RLC layer are segmentation and reassembly of upper layer packets in order to adapt them to the size which can actually be transmitted over the radio interface. For radio bearers which need error-free transmission, the RLC layer also performs retransmission to recover from packet losses. Additionally, the RLC layer performs reordering to compensate for out-of-order reception due to Hybrid Automatic Repeat reQuest (HARQ) operation in the layer below. There is one RLC entity per radio bearer.

Medium Access Control (MAC) layer: This layer performs multiplexing of data from different radio bearers. Therefore, there is only one MAC entity per UE. By deciding the amount of data that can be transmitted from each radio bearer and instructing the RLC layer as to the size of packets to provide, the MAC layer aims to achieve the negotiated Quality of Service (QoS) for each radio bearer. For the uplink, this process includes reporting to the eNodeB the amount of buffered data for transmission.

At the transmitting side, each layer receives a Service Data Unit (SDU) from a higher layer, for which the layer provides a service, and outputs a Protocol Data Unit (PDU) to the layer below. The RLC layer receives packets from the PDCP layer. These packets are called PDCP PDUs from a PDCP point of view and represent RLC SDUs from an RLC point of view. The RLC layer creates packets which are provided to the layer below, i.e., the MAC layer. The packets provided by RLC to the MAC layer are RLC PDUs from an RLC point of view, and MAC SDUs from a MAC point of view. At the receiving side, the process is reversed, with each layer passing SDUs up to the layer above, where they are received as PDUs [11].

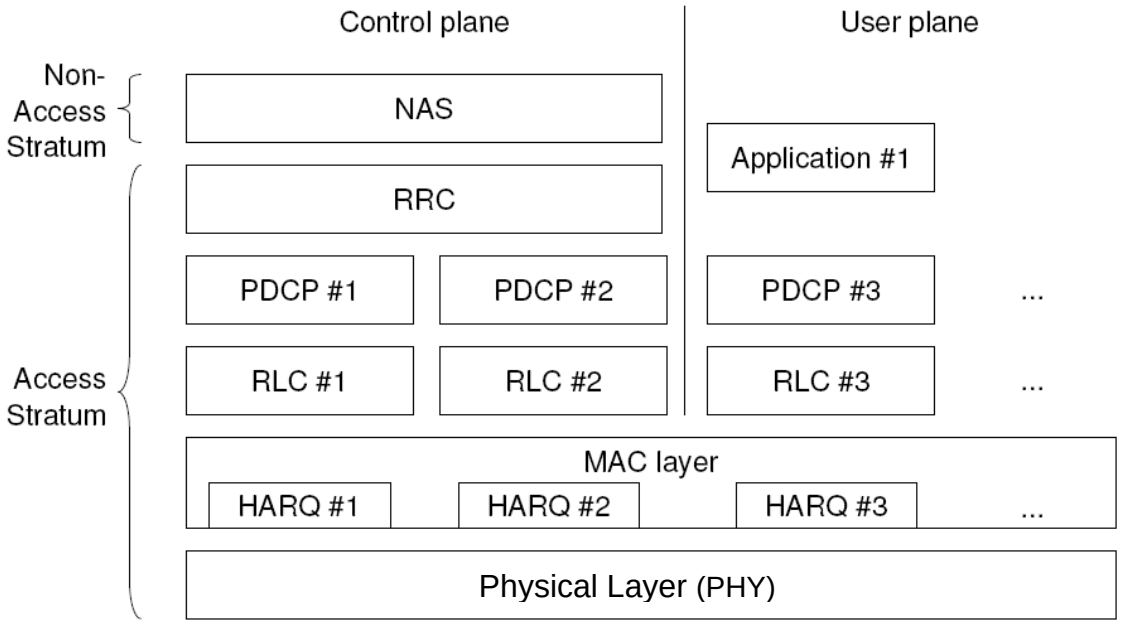

Figure 6.2-1: Overall Uu protocol stack

The **Physical Layer (PHY) r**epresents Layer 1 of the overall Uu protocol stack.

The physical layer of the LTE Uu interface uses OFDM for both uplink and downlink. For the uplink, to improve power amplifier efficiency, the variant DFT-spread OFDM is used. The channel access in the uplink and downlink is scheduled by the eNode B to avoid collisions and channel sensing delays, except for the uplink random access (RA). RA is, however, performed infrequently for V2X messages, which are largely transmitted periodically.

LTE supports bandwidth of 1.4, 3, 5, 10, 15, 20MHz and the subframe duration is 1ms. The scheduling is both in time and frequency. Thereby, for small messages that typically would not exploit the capacity of an e.g. 1ms/10MHz transport block, multiple UEs can be scheduled simultaneously in the frequency domain. The allocation unit is a resource block of 180kHz bandwidth.

6.2.2 MAC Procedures

The two fundamental scheduling modes of LTE are dynamic and semi-persistent scheduling (SPS). Transmission Time Interval (TTI) Bundling was introduced to increase uplink transmission range often limited by transmit power constraints already in Rel. 8 for Time Division Duplex (TDD) mode, followed by enhancements in the LTE Coverage Enhancement work item in Rel. 12.

6.2.2.1 Dynamic Scheduling

Figure 6.2-2 depicts the process of dynamic downlink scheduling and related delays. Data arrives in the PDCP/RLC buffer of the related bearer and the related UE(s). The scheduler then decides when to transmit the data. This decision is subject to scheduler implementation. It therefore results in an implementation dependent queueing delay d_Q. This delay especially increases if more data is queued for transmission than radio resources are available. Different bearers can have different priorities and the scheduler evaluates the priority when deciding which data to transmit. Transmissions are conducted within one TTI. In case the PDCP packet does not fit inside the scheduled resources further reassembly delays will apply until all segments of the packet are

transmitted. Downlink Control Information (DCI) is transmitted before the actual data to support the UE in finding where in frequency domain the transmission is located and what MCS and MIMO scheme is used to transmit it. The UE needs up to 3 ms to decode the packet resulting in a processing delay d_{Proc}. In the case of error free reception, the packet is passed to the RLC layer and a positive Acknowledgement (ACK) is transmitted to the eNB. Assuming no segmentation, packet transmission takes 4 ms plus queueing delay d_Q. The packet has to at least remain queued until the next TTI starts resulting in 0 to 1 ms delay with 0.5 ms on average.

The second data packet shown in Figure 6.2-2 cannot be successfully decoded at the UE. The UE therefore replies with a Negative ACK (NACK) to the eNode B after 3 ms processing time d_{Proc}. The eNode B also requires 3 ms to decode the NACK and then retransmits the packet. In this example, the packet is successfully received after first retransmission. The overall delay is therefore 12 ms + d_{Q2a} + d_{Q2b}. Further retransmissions can happen and each one adds at least 8 ms additional delay. The number of retransmission attempts is limited according to a configured maximum value. The packet is dropped after this value is reached. In case of Acknowledged Mode RLC, packets after reaching maximum retry count can be recovered by RLC ARQ. Another source of packet loss is falsely decoding a NACK as an ACK and therefore assuming a packet was successfully decoded although it was not.

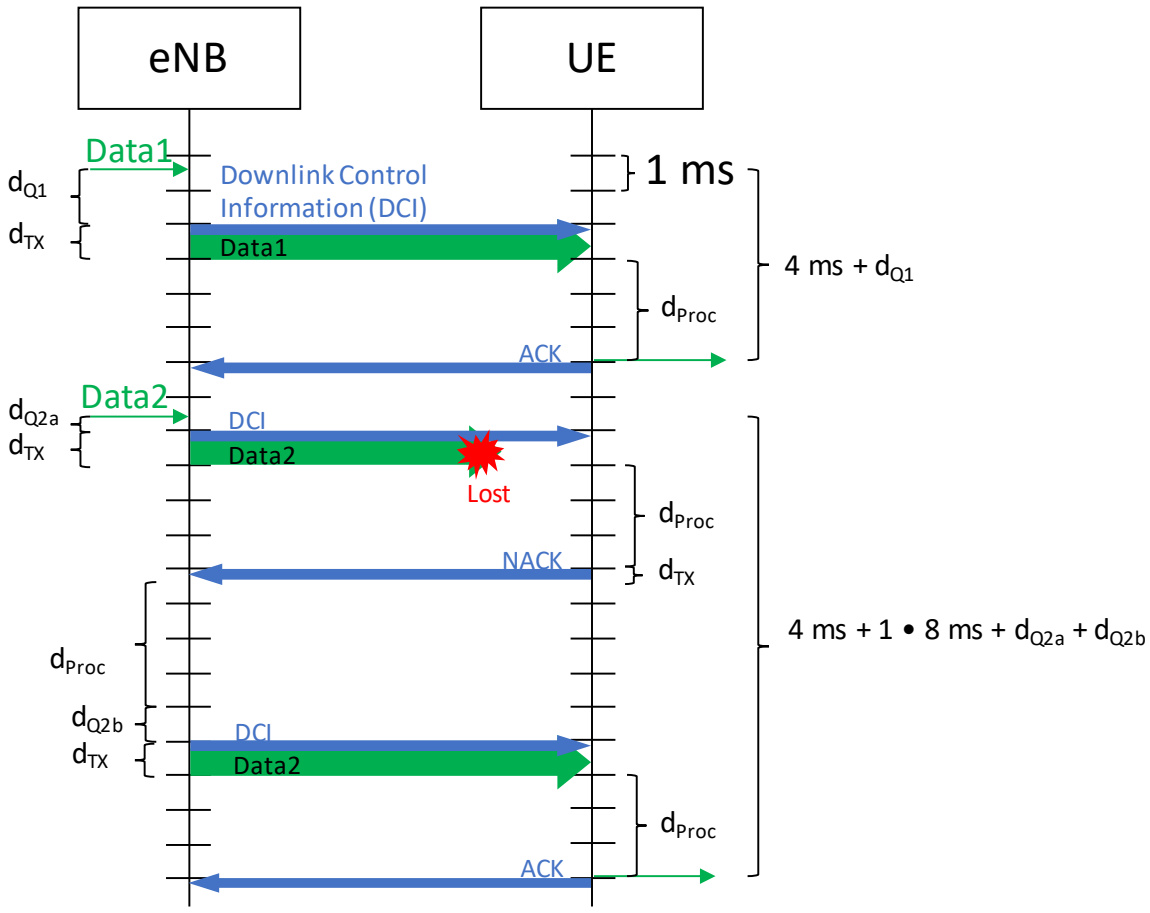

Figure 6.2-2: Dynamic Downlink Scheduling

Figure 6.2-3 presents different examples for uplink data transmission with dynamic scheduling. Two main differences compared to downlink exist:

- The scheduler in the eNode B decides when a UE is allowed to transmit. For this it must know about data being buffered at that UE.

- HARQ is done synchronously meaning that the UE must[1] retransmit 3 ms after receiving a NACK. Each retransmission therefore adds exactly 8 ms delay (see third packet in Figure 6.2-3). Everything else is the same as for the downlink and will not be further explained.

UEs transmit Buffer Status Reports (BSRs) to the eNode B to inform it about buffered data. For that the UE also needs uplink resources for data transmission. If there were no previous ongoing transmissions the UE usually does not have such resources and needs to send a Scheduling Request (SR) to the eNode B. The delay d_{SR} between data being buffered and sending SR results from waiting for dedicated control channel resources where SR can be transmitted. Those are usually present in every[2] TTI resulting in 0 to 1 ms delay with 0.5 ms on average. The SR only requests resources without containing information about the amount of data to be transmitted. The eNode B will grant enough resources through an Uplink Grant to at least transmit the BSR containing more detailed information about the amount of queued data. It is up to the scheduler to grant more resources. The eNode B needs at least the 3 ms processing time d_{Proc} to decode the SR before replying with an Uplink Grant. Further scheduling delay d_{Sched} can apply in case the eNode B has no free resources to be granted. This delay depends on the uplink scheduling algorithm. The Uplink Grant contains information which resources (PRBs) should be used for transmission 3 TTIs after the grant was received. MCS and further transmission parameters like MIMO scheme are also included. The example in Figure 6.2-3 assumes the amount of granted resources is not sufficient to transmit the whole first packet. The UE therefore also transmits a BSR to inform the eNode B about the amount of data that is still buffered. The eNode B therefore sends another Uplink Grant together with the ACK confirming the segment was successfully received. In this case total transmission time is 20 ms plus the delay to send out the SR d_{SR} and the scheduling delay d_{Q1}.

The example in Figure 6.2-3 shows a second data packet arriving before the second segment of the first one was transmitted. In this case the UE can transmit another BSR using the same resources intended for the second segment of the first packet. The eNode B responds immediately after 3 ms processing delay d_{Proc} in the same TTI the ACK for the first packet is transmitted. Total delay in this case, when no SR must be transmitted, is 12 ms + d_{Q2}.

The third packet experiences additional 8 ms delay for a retransmission. The queueing delay is now split into two parts d_{Q3a} and d_{Q3b}. The first one is equivalent to d_{Q2} and caused by waiting for an opportunity to transmit a BSR within resources granted for transmission of the second packet. The second delay d_{Q3b} comes from the fact that the scheduler in the eNode B does not have to immediately grant resources e.g. if there are no available ones. It is therefore equivalent to d_{Q1}.

[1] The standard also allows to delay the retransmission and therefore resemble asynchronous HARQ operation as done in the downlink but this is not considered default protocol behavior.

[2] Assuming the UE has not been inactive for such long time that its control channel resources (Physical Uplink Control Channel (PUCCH)) were revoked. In this case significantly longer delays apply in the range of tens of milliseconds caused by the transition from RRC_Idle to RRC_Connected state requiring access using the Random Access Channel.

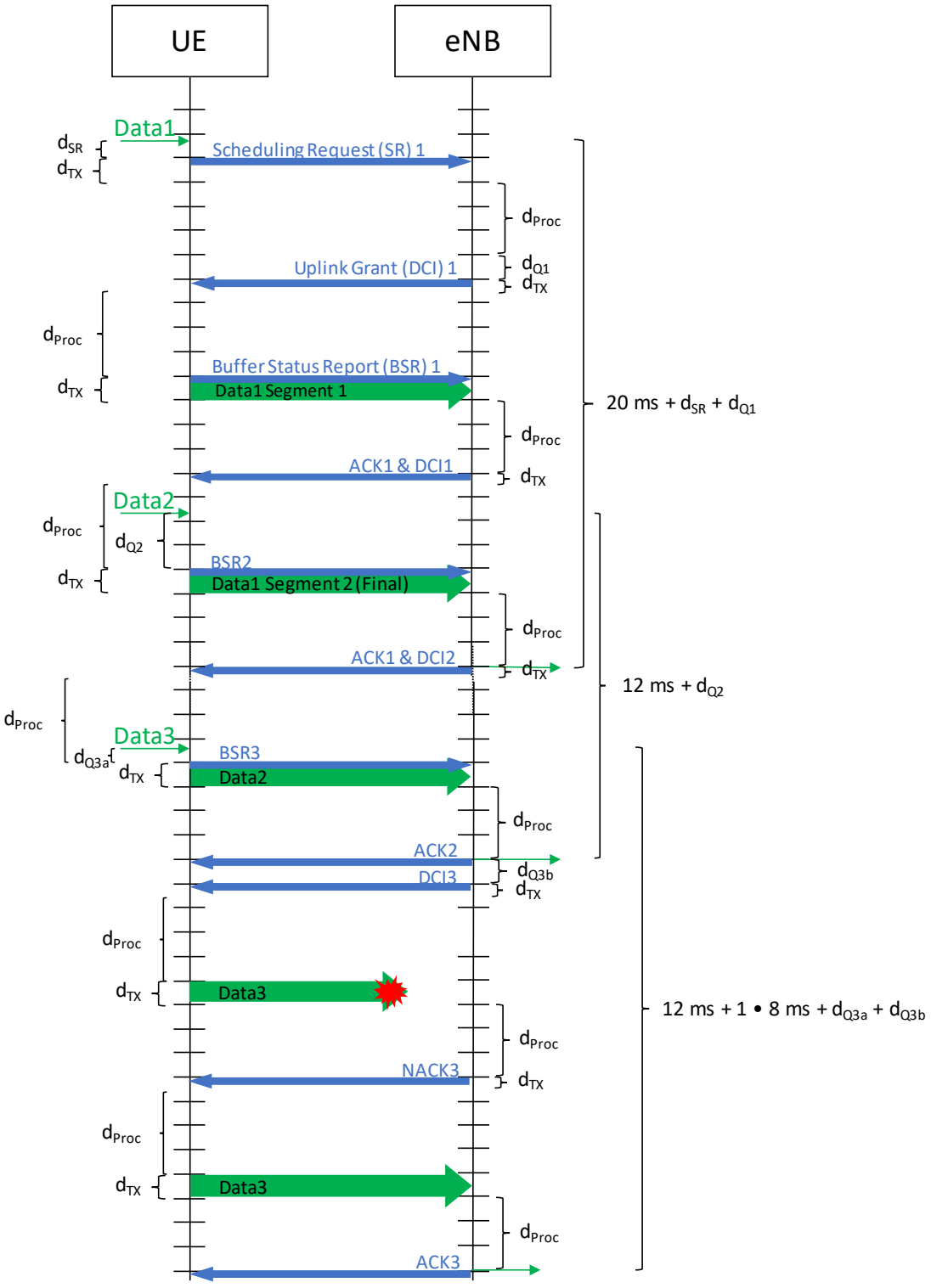

Figure 6.2-3: Dynamic Uplink Scheduling

6.2.2.2 Semi-Persistent Scheduling (SPS)

Semi-persistent scheduling (SPS) does not require signaling DCI for every packet. The same resources with same configuration (MCS, MIMO-scheme, etc.) are reserved periodically. It was originally introduced for VoIP packets generated periodically and with identical size while a person is speaking. SPS is enabled through Radio Resource Control (RRC) Layer signaling originating from the eNode B where the UE receives a separate identifier valid for SPS. The

signaling message also includes the period. If the UE receives DCI with this identifier it knows the resources are not granted once as with dynamic scheduling but remain valid until the grant is revoked or changed. Changing a grant can especially be necessary in case of changing channel conditions requiring to adjust the MCS or MIMO-scheme. HARQ retransmissions are scheduled dynamically according to the description above.

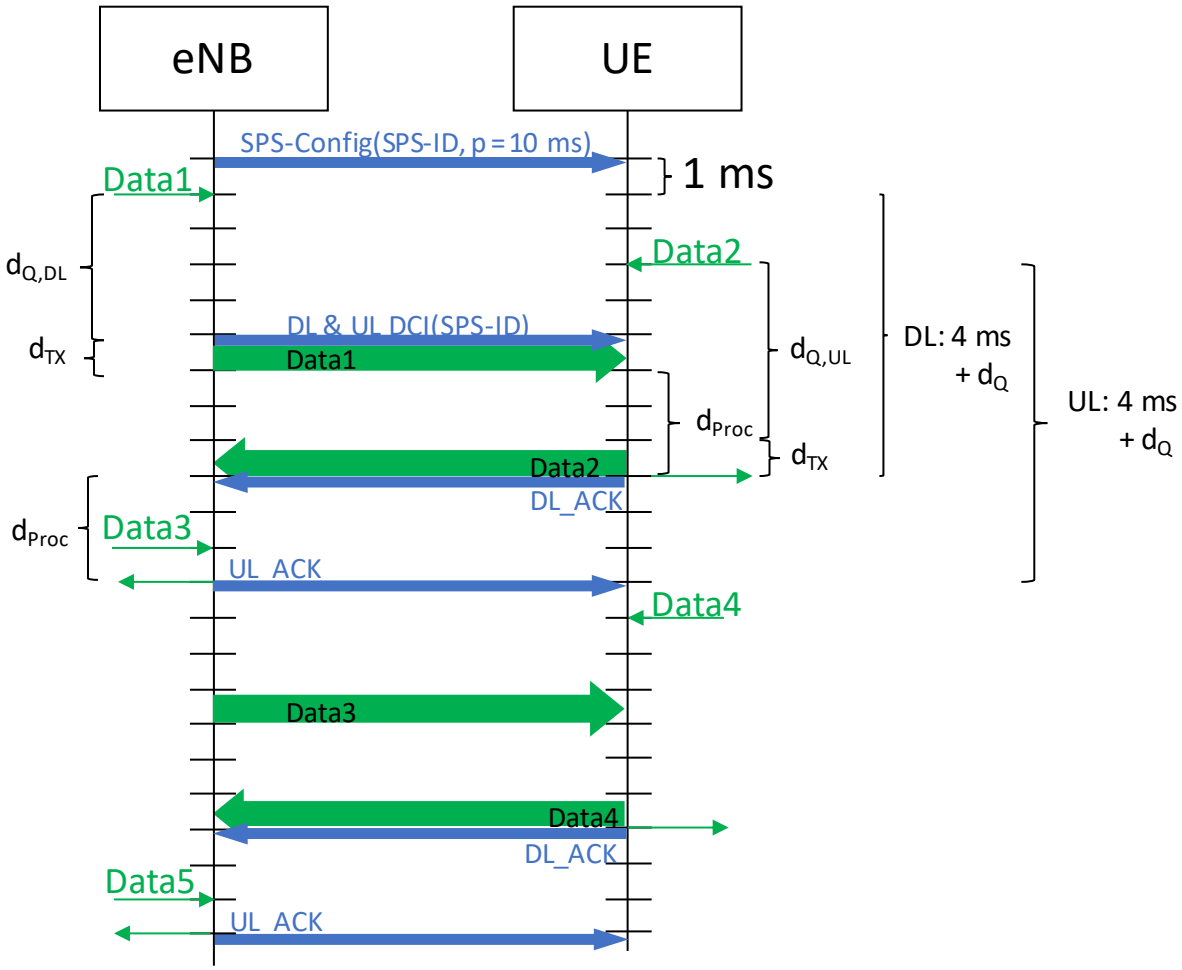

Figure 6.2-4: Semi-Persistent Scheduling (SPS)

Figure 6.2-4 shows SPS for downlink and uplink. As the first step, the RRC Layer of the eNode B transmits an *rrcConnectionReconfiguration* message to the UE including an SPS-Config element. In this example both, downlink and uplink, are configured, but it can be also done for just one direction. The RRC Layer message is transmitted using Acknowledged Mode RLC communication and confirmed by the UE through a *rrcConnectionReconfigurationComplete* message (those details are omitted in Figure 6.2-4). At this point SPS is configured but not activated yet. The UE receives a unique identifier used to distinguish between dynamic and SPS scheduling grants transmitted through DCI. In this example p = 10 ms period is used for the sake of readability. In the downlink, the grant becomes valid in the same TTI where the DCI was sent and from there on every 10 ms. In the uplink, the DCI refers to 3 TTIs later. As with dynamic scheduling, the receiver needs 3 ms processing time d_{Proc} to decode the packet. Total delay is therefore 4 ms plus queueing delay d_Q. The queueing delay depends on the offset between message arrival and reserved resources. In the uplink this offset can be assumed almost constant since the message originates from the UE. Jitter is possible in the downlink since the message can experience varying delays on its way from the source to the eNode B. This results in 0 to p ms

delay with p / 2 on average assuming uniformly distributed offsets. As with dynamic scheduling, HARQ is used introducing at least 8 ms delay per retransmission.

Permanent link adaptation as done with dynamic scheduling is not possible with SPS. The eNode B can transmit another DCI in order to reconfigure the SPS grant in case channel quality becomes worse and initially selected MCS and MIMO-scheme result in too high packet error.

A grant can either be revoked explicitly by DCI or implicitly if no data is transmitted for a certain number of consecutive periods. This number is announced in the SPS-Config element.

6.2.2.3 TTI Bundling

Overall uplink transmission power is limited to 23 dBm. The more resources are used in frequency domain the less power is transmitted on each one leading to reduced SINR at receiving eNode B and therefore high packet loss probability. TTI-Bundling is a mechanism to transmit a packet over multiple TTIs in time domain. As shown in Figure 6.2-5 less resources are then used in frequency domain allowing higher transmission power on each one. Almost the same behavior would be achieved when dynamically scheduling on the same resources in two consecutive TTIs. The difference is that signaling would have to be performed twice and RLC Layer would need to segment the packet and add headers to each segment.

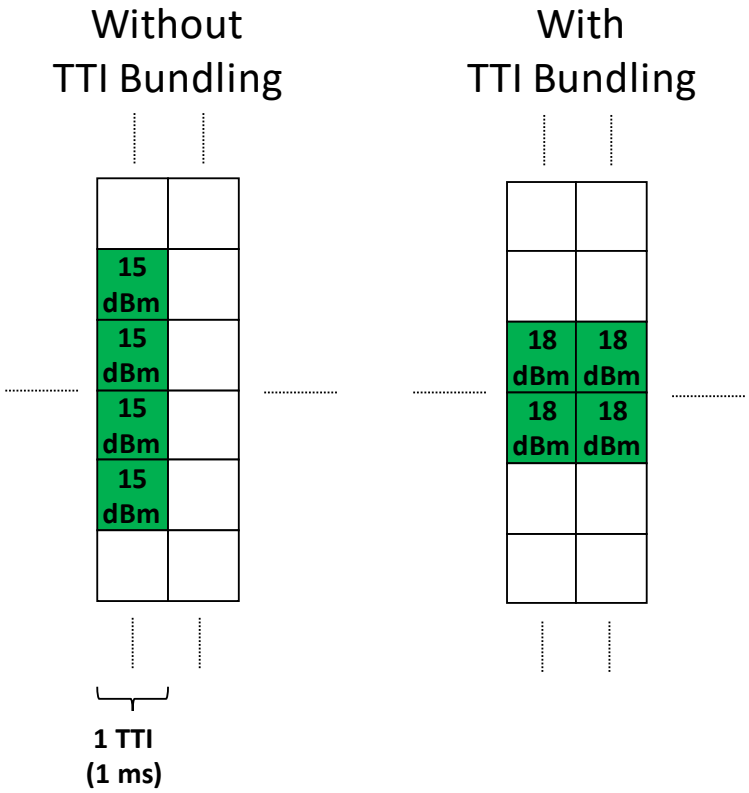

Figure 6.2-5: TTI Bundling

6.3 ITS Stack

The general structure of the ITS Protocol Communication Stack applying to the ConVeX project is illustrated on Figure 6.3-2.

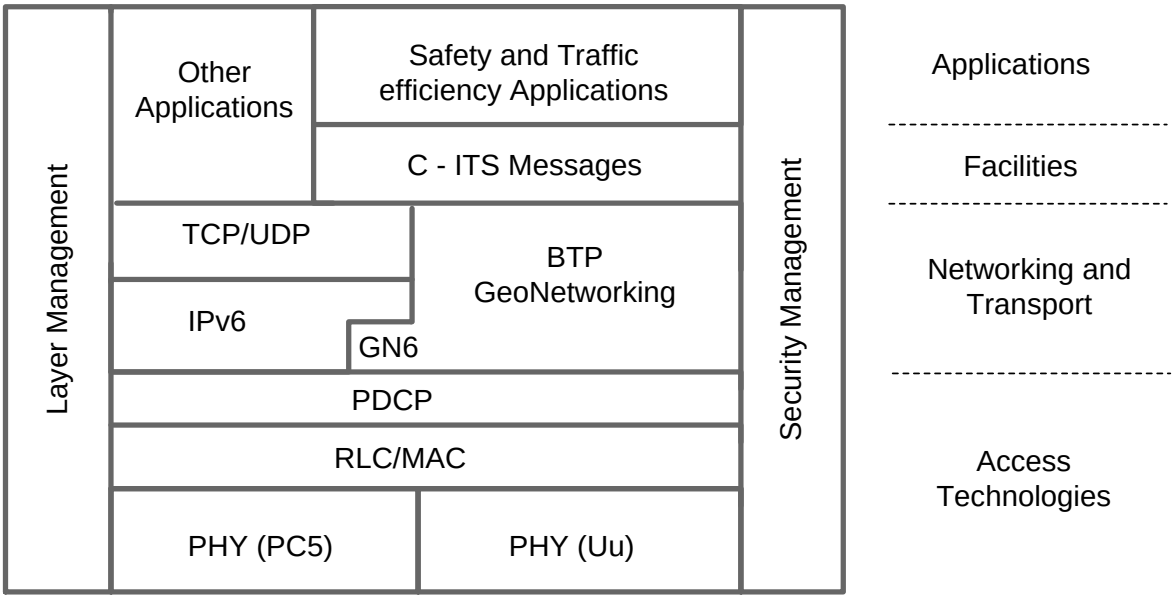

Figure 6.3-3 ITS Protocol Stack

Facilities Layer: Specifies requirements and functions supporting applications, communication and information maintenance. It covers messaging protocols, position and time management, location referencing, sensor data fusion in local dynamic map (LDM) and others. The most relevant in ConVeX environment are those for C-ITS messaging CAM, DENM, IVI.

CAM (Cooperative Awareness Message) is a periodic message that provides car status information to neighboring ITS station (RSU) or cars (OBU). It conveys critical vehicle state information to support variety of safety and traffic efficiency applications, with which the receiver can track other vehicles positions and movements. Its transmission is activated when a vehicle is in a safety-relevant context (basically, when the engine is running).

A CAM is composed of an ITS PDU header and several containers (***Fig. 6.3-2.***) that group the data fields by the role of the sender and frequency of their appearance in the message.

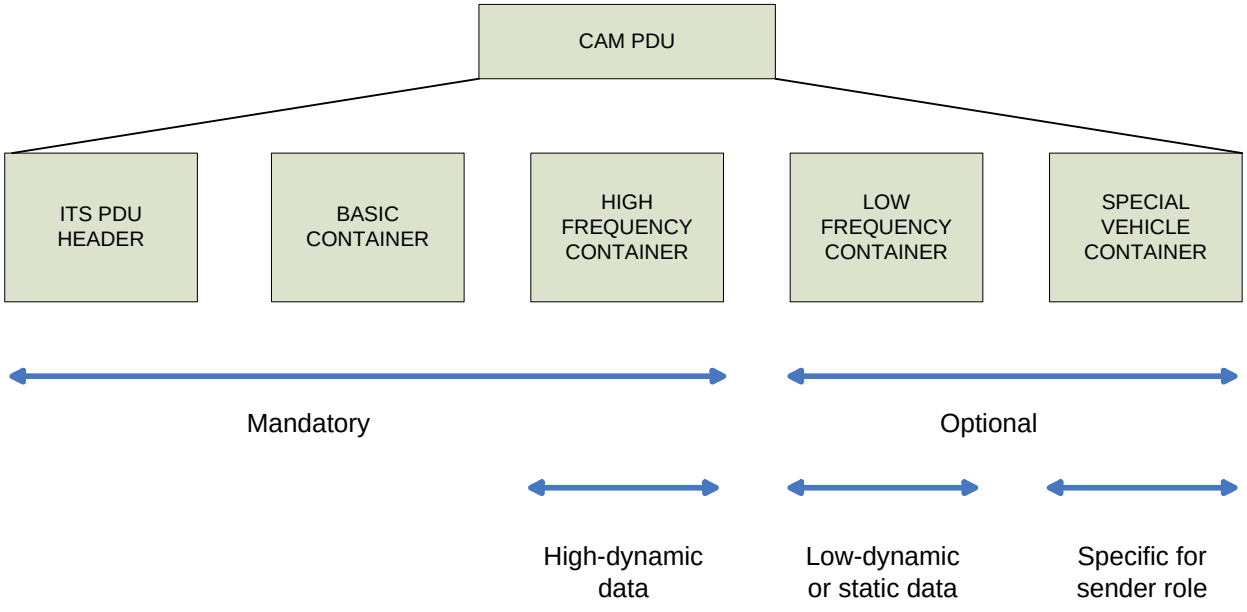

Fig. 6.3-2. CAM General Structure

The ITS Protocol Data Unit (PDU) header carries protocol version, message type, and sender address; the basic container has station type and its position. In order to reduce the size of the CAM, the high-frequency container carries mainly highly dynamic data (e.g., vehicle heading, speed, and acceleration) and is sent in every CAM. The low-frequency container has data with less safety relevance (e.g., vehicle role) or may have a large size (e.g., path history) and is therefore not always added to the CAM, but sent. The special vehicle containers are optionally added if needed for the sender's role, such as for public transport, dangerous goods, road works, or rescue.

The container concept ensures a flexible message format that can be adapted to the needs of the sending and receiving vehicle, while minimizing the load on the wireless channel.

The CAM rate is determined by CAM generation rules and can vary between the lower and upper limit of the CAM period TMin = 100 ms and TMax = 1 s (corresponds to a CAM rate of 1to 10 in 1s), controlled by the vehicle dynamics, application, and congestion status of the wireless channel. The conditions are sampled at small intervals (minimum 10 Hz).

If the vehicle dynamics exceed the predefined thresholds for heading, movement, and acceleration, a CAM is generated. A low-frequency container and special vehicle container are included if at least 500 ms has passed since the last CAM generation.

DENM (Decentralized Environment Notification Message) – an application controlled, event triggered message with safety information in geographical region.

When a vehicle detects a safety situation, the DENM protocol assigns an *action identifier* that is unique for the detecting ITS station. Unlike the CAM broadcast over a single hop, the DENM gets assigned a relevance area for dissemination and can be transported over several hops, typically through the geo-broadcast mode. Similar to the CAM, the DENM is organized in containers with a prepended ITS PDU header (Fig. 6.3-3).

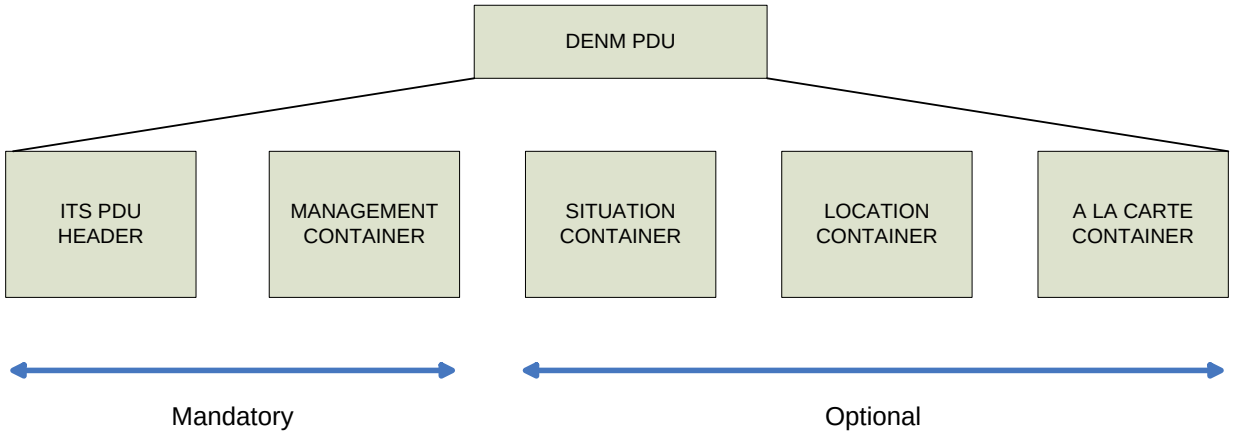

Fig. 6.3-3. DENM General Structure

The management container with fields for action identifier, detection time, event position, and so on - is mandatory, all other containers are optionally added if needed by the application. The situation container has fields to describe the event by a predefined code for the causing event as well as related events (e.g., linked events or an event history). The location container carries fields for the event speed, heading, and traces. An a la carte container can be added to transmit application-specific contents, such as for lane position, impact reduction, and road works, among others.

The DENM protocol can handle an event life cycle: an event with a specific action ID can be triggered and then updated by the originator of the DENM; the event updates are distinguished by an increasing value of a data version field. An event can also be canceled by the originator or negated by a third ITS station. The DENM protocol specification has several mechanisms for

information dissemination to keep the safety information in the relevant area during the event lifetime. The originator can repeat a DENM, typically at a lower frequency than a CAM, to ensure that vehicles entering the relevant area later can receive the information. Optionally, another ITS station than the originator can overtake the repetition of the DENM in case the originator fails to repeat the DENM (e.g., if it is broken or has left the relevant area).

IVI (In Vehicle Information or Infrastructure to Vehicle Information) – denotes a data structure that is required by different ITS services to convey information into the vehicle.

The IVI Structure (Figure 6.3-3) itself is specified as a general, extensible data structure. It is split into structures called containers. Transmitted information includes In-Vehicle Information (IVI) such as contextual speed, road works warnings, vehicle restrictions, lane restrictions, road hazards warnings, location-based services, re-routing, etc. The information in the containers is organized in sub-structures called data frames and data elements, which are described in terms of content and syntax.

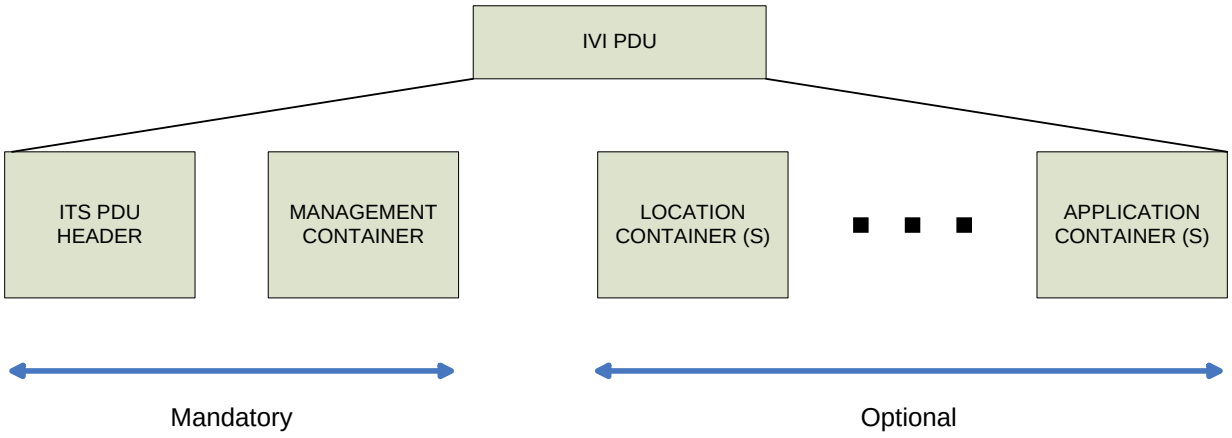

Fig. 6.3-3. General IVI Structure

The IVI Structure is intended to be encapsulated in a message with the appropriate ITS Common Header, similar to a DENM.

It shall contain a Management Container. The information in the IVI Management Container is applicable to the entire IVI Structure. This Container is mandatory and provides a receiving ITS-S with enough information to handle the IVI Structure and decide on its further processing.

The IVI Structure may contain one or more Location Container(s). The Location Container describes the essential information for applications in the receiving ITS-S: to understand how to apply the information provided by the IVI Application Containers. Location Containers may carry information relevant for different Application Containers, or carry the same content but expressed in different forms. This enables a receiving ITS-S to choose the most appropriate, supported location referencing system.

The IVI Structure may contain one or more Application Container(s). The IVI Application Container provides IVI information for use by an application. Application information is self-contained and refers to location information for its spatial validity. Application information of the same type shall not refer to overlapping Reference Zones. Each Application Container refers to zones defined in the Location Container identified by their IDs for the following usage:

1) Relevance Zone – part of the road network for which the information in Application Container is valid

2) Detection Zone – part of the road network that is passed by the vehicle in approach of the Relevance Zone

3) Driver Awareness Zone – parts of the road network on which a message is presented to inform drivers about upcoming situations

An Application Container may optionally provide information about the minimum awareness time; that is, the minimum time that the IVI should be available before the vehicle enters the Relevance Zone. This *MinimumAwarenessTime* information can be used by the receiving ITS-S to determine the appropriate Driver Awareness Zone.

The Networking and Transport layer has two columns: GeoNetworking and Basic Transfer Protocol and the other one – employs Internet Protocols, in particular IPv6 with UDP, TCP or potentially other transfer protocols such as IP mobility extensions (mobile IPv6 and its extensions for network mobility). The choice of the communication profile lies in the application. Typically, the GeoNetworking stack is used for the ad hoc communication over PC5 utilizing geo-addressing, and IPv6 for communication with IP Base infrastructure over cellular networks.

Access Technologies Layers:

As access technology LTE will be used in two flavours: the traditional LTE mobile network utilizing the Uu interface, as well as the Rel.14 C-V2X Sidelink on the PC5 interface. Apart from the slightly different Physical Layers (PHY), both use RLC/MAC and PDCP. Details are provided in the respective sections 6.1 and 6.2.

Security Protocols:

Security- and privacy-related standards enable asymmetric cryptography and changing pseudonyms. They specify mechanisms for security and privacy protection. Based on the security architecture in ETSI TS 102 940, ETSI TS 102 097 specifies private key infrastructure (PKI) enrollment and authorization management protocols, ETSI TS 102 941 confidentiality, and ETSI TS 102 942 data integrity. Detailed description of security mechanisms is addressed in section 6.5 of this document.

Management Protocols:

Management protocols mainly cover support for decentralized congestion control and communication profile management. Detailed description of management protocols is out of scope of this document.

6.4 ITS Applications

The following applications are currently intended to be developed in the course of this project. A detailed description of these services is provided in Deliverable D1.1 [1].

Warning services:

- Blind Spot / Lane Change Warning (LCW)
- Do Not Pass Warning (DNPW)
- Emergency Electronic Brake Lights (EEBL)
- Intersection Movement Assist (IMA)
- Left Turn Assist (LTA)
- Vulnerable Road User Warning (VRUW)

Advanced services:

- FollowMe Information (FMI)
- Cloud Based Sensor Sharing (CBSS)

- Shockwave Damping (SWD)
- In-Vehicle Information (IVI)
- Road Works Warning (RWW)
- See Through (ST)
- Network Availability Prediction (NAP)

The development of the warning services is planned such that the application layer software will be available for the initial field trials scheduled in early Q2 of 2018. These warning services will only require PC5 communication between vehicles and, for the VRUW use case, transmission of CAM by a VRUE which are received and processed in the C-V2X equipment of vehicular ITS stations.

Cloud Based Sensor Sharing (CBSS) requires V2N communication from one endpoint located in a vehicular ITS station to another endpoint located on an application server in the cloud.

Shockwave Damping (SWD), In-Vehicle Information (IVI) and Road Works Warning (RWW) require V2I communication between the vehicular and roadside ITS stations.

FollowMe Information (FMI) and See Through (ST) will be implemented as point-to-point unicast services between two vehicular ITS stations.

Network Availability Prediction (NAP) is an application which runs locally on a host vehicle not requiring a counterpart in another entity (vehicle, RSU or cloud server). However, this application may require a Local Radio Map, which could be downloaded using V2N communication from some cloud server.

A potential overall architecture of the application software is illustrated in Figure 6.4-1. All shown processes are running on the application processor of the C-V2X communication platform as described in Section 5.1.

The overall software architecture consists of two main parts: a set of top-level application modules and a set of shared libraries. The shared libraries implement the functionality of the ITS stack including communication with external hardware modules, such as Human Machine Interface (HMI) via Ethernet, V2X radio modems (V2N LTE Uu, V2V/I/P LTE PC5), Hardware Security Module (HSM), CAN bus and GNSS module. The communication between the application modules and shared library functions are implemented via Application Programming Interfaces (API). The shared libraries themselves employ lower-level APIs for communication with the external hardware modules.

Figure 6.4-1 shows the application software structure of a vehicular ITS station which supports all services as listed above. A roadside ITS station would require the Application Module for RSU services only. The detailed functionality of the respective Application Module for RSU services implemented in the vehicular and the roadside ITS stations can be rather different. For example, in case of the IVI service, the roadside ITS station receives traffic signage information from a Traffic Control Center (TCC) and converts it into an IVI message broadcast via the PC5 interface. In contrast, the vehicular ITS station receives these IVI messages and triggers notifications on the HMI interface. A VRUE requires a slim version of the Application Module for Warning Services only. A cloud server requires the peer function of the Application Module for V2N Cloud Services only.

Implementation details of the software architecture are currently being designed. More details of the software architecture will be described in a future update of this Deliverable when the design phase is completed.

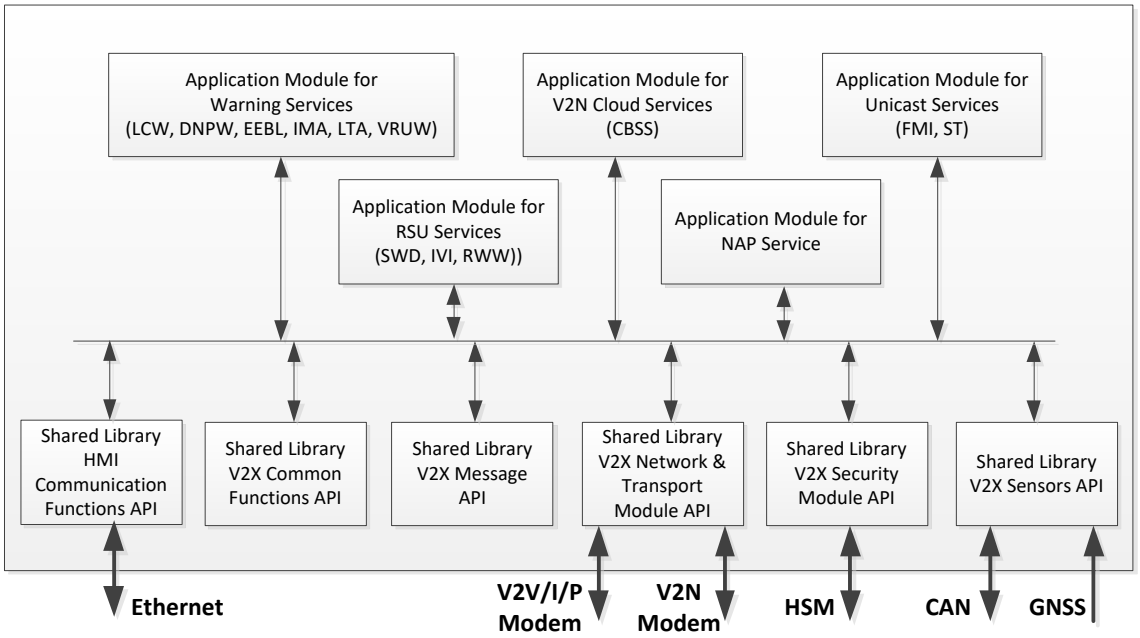

Figure 6.4-1: Structure of Application V2X Software

6.5 Security Aspects

6.5.1 Introduction

Consider the common V2X use case of vehicle prioritization at a traffic light: the traffic light ahead of a vehicle should prioritize an emergency vehicle, but not an ordinary car.

Obviously, there is a need for mechanisms to differentiate between various types of vehicles, to grant each vehicle individual permissions to indicate its specific type of vehicle and provide authorization to send specific warning messages. The security services must also comprise mechanisms to enable the receivers to securely authenticate any ITS message originators in order to validate that it can be trusted.

In a nutshell, V2X communication requires security frameworks covering the following security functions:

- Authentication and Authorization

    - The authentication function provides mechanisms to the message receiver to authenticate the identity of the originator of the message.
    - The authorization function ensures that specific ITS messages are sent only by originators, which are authorized to send the respective message.
- ITS message integrity protection
    - The integrity protection function provides means to ensure that a message cannot be modified while in transit without being detected by the receiver. This is accomplished by appending an ITS message with a digital signature.
- ITS message confidentiality protection:
    - The confidentiality protection function provides mechanisms to protect the content of a message from access by unauthorized receivers. This includes protection against eavesdropping by unauthorized parties. This can be done by encrypting ITS messages and providing the security credentials required for decryption to authorized receivers only.
- ITS message privacy protection
    - The privacy protection function provides mechanisms which hide the real identity of the originator of ITS messages to other ITS users. This prevents ordinary ITS users to be able to track the location of any individual other ITS users. This is done by using pseudonym identifiers in certificates required for Authentication and Authorization and changing these identifiers frequently.
- Security credential provisioning/management via Public Key Infrastructure (PKI)
    - This is the security service which allows the initial provision and dynamic re-assignment of security credentials required by the mechanism used for authentication, authorization and protection of integrity, confidentiality and privacy. Since public key certificates are employed as security credentials, this requires a PKI.

Many ITS services, including the majority of services considered for trialing and demonstration by this project as described in [1], are based on transmission of CAM and DENM broadcast messages which are intended to be visible by all V2X recipients. There are accordingly no confidentiality requirements associated with these messages.

There are, however, some use cases which will use point-to-point unicast communications and which will contain sensitive information of a personal or commercial nature. These services will require encryption to ensure that the information becomes visible to the intended recipient(s) only.

It is essential, that both broadcast and unicast messages are protected against alteration by means of integrity protection. Integrity protection is accomplished by appending the message with a digital signature. The signature is derived by calculating a hash (also denoted as *digest*) of the actual message using a cryptographic hash function, e.g. SHA128 or SHA256, and encoding the digest with the private key of the originator of the message using a standardized signature algorithm such as (Rivest, Shamir und Adleman) RSA, Digital Signature Algorithm (DSA) or ECDSA (Elliptic Curve DSA).

The security framework of DSRC/WAVE systems is specified in IEEE 1609.2 [12].

ETSI ITS security framework is defined in a series of specifications as illustrated in Figure 6.5-1. Note that at the present time, many of these specifications are not in a state ready for publication. For most specifications only draft revisions exist which are being worked upon in the ETSI Technical Committee ITS.

The ETSI ITS security framework adopts the concepts specified in IEEE 1609.2 [12]. The content of the ETSI documents can be summarized as follows:

- ETSI TR 102 893 [17] documents a comprehensive analysis of ITS security threats and risks
- ETSI TS 102 731 defines the high-level security architecture based on a Threat, Vulnerability, Risk Analysis (TVRA) using the results of [17]
- ETSI TS 102 940 defines security requirements and the overall security architecture of the ETSI ITS system.
- ETSI TS 102 941 defines the messages exchanged between ITS stations with Enrolment Authorities (EA) and Authorization Authorities (AA)
- ETSI TS 102 942 specifies details of Authentication and Authorization Services for CAM and DENM
- ETSI TS 102 943 specifies services recommended for confidentiality protection
- ETSI TS 103 097 defines the frame format of secured C-ITS messages including the binary format of applicable certificate representations.

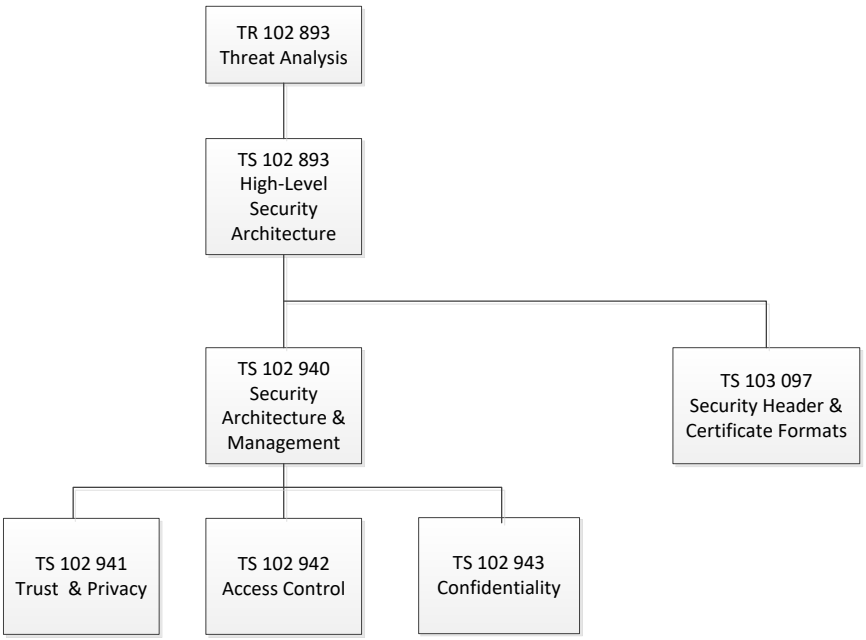

Figure 6.5-1: Specifications defining the ETSI ITS Security Framework

6.5.2 ETSI ITS Security Architecture

The ITS security architecture is comprised of the security mechanisms and management functions applied to the messages (e.g. CAM, DENM) exchanged between vehicular ITS stations, roadside ITS stations and the network via the V2X interfaces (PC5 and Uu) which are based on public key certificates, and a Public Key Infrastructure (PKI). The PKI is required to manage the security credentials employed for securing V2X communication.

A public key certificate includes the public key, information about the identity of the owner of the public key (called the subject), and the digital signature of an entity (denoted Certification

Authority) that has verified the certificate's contents (called the issuer of the certificate). If the signature is valid, and the software examining the certificate trusts the issuer, then it can use that public key to communicate securely with the certificate's subject.

As part of a certificate enrolment procedure, a secret private key is generated together with the certificate which includes the public key.

Figure 6.5-2 shows the PKI of the ETSI ITS security architecture. The PKI is comprised of the following entities:

- Enrolment Authority (EA)
  In generic PKI terminology, an EA represents a Registration Authority (RA). An RA is responsible for verifying Certificate Signing Requests (CSRs) and authorizing a CA to issue a corresponding end-entity certificate. The EA receives Enrolment Requests from ITS stations and validates their manufacturer-assigned globally unique identity. Validation of the identity of the enrolling ITS station ensures that it can be trusted to function compliant with the ITS specifications. The EA provides a proof of authentication of the ITS station [22].

- Authorization Authority (AA)
  The AA represents a Registration Authority (see definition above) as well. The AA provides an ITS station with multiple pseudonyms and it controls CSRs for pseudonym certificates from a CA. These pseudonym certificates are associated with specific authorization privileges which are assigned individually to each ITS station and include the information which ITS services the ITS station is authorized to use. The pseudonym certificates to be used to secure ITS communications are therefore also denoted as Authorization Tickets (ATs).

- Certification Authority (CA)
  A CA processes certificate signing requests (CSR) issued by the end-entities (ITS stations) and validated by the EA or AA. It may either generate the private key and CA-signed end-user certificate, or it may just generate the CA-signed certificate for a private key generated locally by the end-user.

  If this CA is not a Root CA itself, it uses for signing user certificates a CA certificate which has been signed by another superordinate CA in the certification chain of trust, to which the present CA maintains a trust relationship.

- Root Certification Authority (Root CA)
  A Root CA represents the root of trust in a certification chain. It presents itself to other entities with a self-signed root certificate. The identity given in the root certificate must be trusted, it cannot be validated by an end-entity.

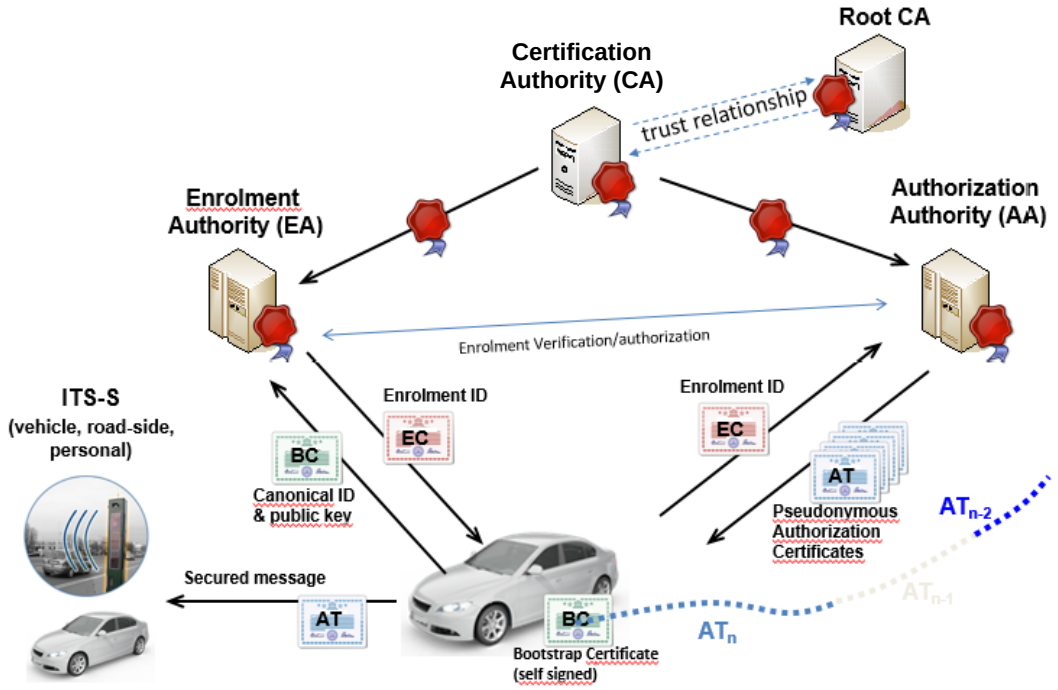

Figure 6.5-2: PKI architecture [14]

6.5.3 ITS Security Frameworks

6.5.3.1 Overview

Device manufacturers issue globally unique identifiers for their ITS stations. These identifiers are denoted *ITS authoritative identifier* or *canonical identity* [22].

TS 102 731 [18] defines the *canonical identity* as follows: identifier unique to a particular ITS-S that persists throughout the lifetime of the ITS-S and can be presented to an enrolment authority when the ITS-S requests enrolment credentials.

The roles of EA and AA are intended to separate identification/authentication from authorization which is regarded as an essential component of privacy management and provision of protection against attacks on a user's privacy [22].

However, it is nevertheless possible to delegate the EA role entirely to a manufacturer, or to keep both, EA and AA roles hosted by a single trust enabling authority.

For privacy reasons, the Authorization Authority provides an ITS station with multiple pseudonyms and the related authorization tickets (or pseudonym certificates) to be used in the ITS communication. Privacy protection is provided by changing frequently the ITS-S pseudonym and its related certificate [14].

6.5.3.2 Communication between ITS Station and EA

When an ITS station goes into operation for the first time, it sends an Enrolment Request message to an EA using a pre-configured EA identifier. To enable the EA to authenticate enrolment requests, the requesting ITS station supplies its manufacturer-provisioned certificate in the security association establishment procedure. Note that ETSI TS 102 941 [22] specifies the content of

Enrolment Request and Response messages, but the details of the used communication protocol(s) are left out of scope of the specification.

At successful enrolment with an EA, the individual ITS station is issued its specific enrolment identifier and enrolment certificate to be used in subsequent communication procedures with an AA. In addition, the requested permissions are stored at the EA: For each ITS message type, there exists a standardized ITS Application ID (ITS-AID) and a bit field containing service-specific (i.e. AID-specific) permissions (SSP). The assigned ITS-AIDs are defined in ETSI TS 102 965 [27]. The SSP determines the authorized content of messages sent by an ITS station, e.g., whether or not a CAM may contain a public transport container. The list of authorized pairs of AID and SSP is included into the pseudonym certificate employed by an ITS station.

The usage of ITS-AID and SSP is described in informative Annex B of ETSI TS 103097 [19]. An incoming secured message should only be accepted by the receiver if the payload of the secured message is consistent with the ITS-AID and SSP included in the certificate of the sender.

Table 6.5-1: ITS message types

Message Type	ITS-AID	SSP (length in byte)	Reference
CAM	36	2	[15]
DENM	37	5	[16]
SPATEM	137	2	[26]
MAPEM	138	1	[26]
IVIM	139	2	[26]

6.5.3.3 Communication Between ITS Station With AA

After successful enrolment, an ITS station sends an Authorization Request message to a AA.

An ITS station requests Authorization Tickets (ATs) via the AA identifying itself with its enrolment certificate. The AA verifies the identity of the requesting ITS station, checks its authorizations and forwards respective Certificate Signing Requests to a CA.

The AID and SSP are included into the ATs (i.e. pseudonym certificates), which are generated and signed by a CA and returned to the requesting ITS station.

To take care of compromised entities, the PKI issues a certification revocations list (CRL). The CRL as well as the EA and AA certificates are accessible from a certificate repository. A link (denoted *crlPath*) to the CRL can be included into Enrolment and Authorization response message [22]. However, due to the involved large complexity, ATs cannot be revoked. Their validity is usually limited to a short time period like several days, whereas enrolment certificates, sometimes called long-term certificates, are valid for months to years.

6.5.3.4 Role of CA and Root CA

A root certification authority represents the anchor of the certificate trust chain. There can be one or more Root CAs in the ITS PKI. Each entity inside the PKI and each ITS station needs to install the root certificate(s) in order to be able to validate certificates and certificate chains. Each CA in the given chain of trust handles certificate signing requests (CSRs) of its subordinate nodes. The root CA signs certificates of its subordinate CAs. A CA directly accessible by AAs and EAs handles CSRs issued by ITS stations. After authentication of the issuer of the CSR, the AA or EA forwards the requests to the CA and the CA signs the generated certificate.

6.5.3.5 ITS Message Formats

ITS messages can generally be represented in one of the following format options [19]:

- unprotected plain binary format (denoted *unsecuredData*),
- integrity protected binary format (denoted *signedData*),
- confidentiality-protected binary format (denoted *encryptedData*).

The *encryptedData* format is not applicable to CAM, DENM and IVIM.

In *unsecuredData* format, ITS messages of types listed in Table 6.5-1 consist of the message payload only. Payload refers to the ITS message PDUs such as e.g. CAM, DENM, and IVIM PDUs as defined in Section 6.3. In *unsecuredData* format, the payload is prefixed by a Protocol Version header.

The *signedData* format of CAM and DENM is specified in [19]. The general structure of this format is illustrated in Figure 6.5-3. A message consists of a Protocol Version, Header, Payload and Trailer fields.

Data structures specified in [19] use Abstract Syntax Notation 1 (ASN.1) and are encoded using the Canonical Octet Encoding Rules (COER) as defined in [12] and ITU-T X.696.

Figure 6.5-3: General structure of signedData message format

The protocol version field (1 byte) specifies the version of the specification the message complies with. The Header and Trailer fields have variable length and are divided into various subfields. Each subfield consists of an enumerated HeaderFieldType identifier of fixed length (1 byte) and the field value which may be structured into several header fields itself.

The Header includes in the *signer_info* field the public-key certificate of the message originator, or a hash of the certificate from which the receiver can identify the originator's certificate that has already been received and stored by the receiver.

Certificates are included only once per second into CAM headers unless the ITS station receives itself a CAM with yet unknown certificate digest. All other messages just include the digest of the certificate (8 bytes). When an ITS station receives a message including an unknown certificate digest, it is highly likely that the sender of that message also does not know the certificate of the receiver. Therefore, the receiving ITS station will include its certificate into the next CAM it sends.

The Header also includes in the its_aid field the ITS-AID described in Section 6.5.3.2.

The Trailer field includes the digital signature of the message, information about the employed signature algorithm and possible data about elliptic curve in case ECDSA signatures are employed.

Protocol version, Header and Payload fields are covered by the signature.

6.5.3.6 Confidentiality Protection

Confidentiality on the application layer is addressed in IEEE 1609.2 [12] and ETSI TS 102 943 [24].

IEEE 1609.2 specifies for encryption of individual data fields a single Public key encryption algorithm and a single symmetric-key encryption algorithm.

The asymmetric encryption algorithm is the Elliptic Curve Integrated Encryption Scheme (ECIES) as specified in IEEE Std 1363a. The only present ITS use case for ECIES defined in [12] is to encrypt symmetric keys.

The symmetric algorithm specified in [12] the Advanced Encryption Standard (AES) in Counter Mode with Cipher Block Chaining Message Authentication Code (CCM), known as AES-CCM.

ETSI TS 102 943 [24] permits any appropriate application layer encryption scheme to be employed, including the ones specified in IEEE 1609.2.

On the network layer, ETSI TS 102 943 [24] mandates that confidentiality for IPv6 services shall be provided using the Encapsulating Security Payload (ESP) protocol within IPSec. The key management is defined in TS 102 941 [22].

For one-to-one unicast communication services over the PC5 LTE sidelink interface, the PDCP layer provides functions for integrity and confidentiality protection [30], [31]. This includes functionality for key management

Use of Transport Layer Security (TLS or DTLS) is currently not addressed in [12] and [24]. However, TLS/DTLS might be a reasonable choice for end-to-end security protection of communication when using the V2N/Uu interfaces, including V2N2V communication.

6.5.4 Security Mechanisms Supported by the C-V2X Communication Platform

This section briefly describes the security mechanisms planned to be supported by the C-V2X communication platform.

CAM and DENM will comply with the ETSI ITS security header as defined in ETSI TS 103 097 [18] and described in Section 6.5.3.4 above.

All C-V2X communication modules employed in the ConVeX test system will use pre-provisioned certificates. These certificates will be installed as part of the Application and ITS stack software installation procedure.

At the present point in time, it is not decided whether or not functionality for remote re-enrolment/re-assignment of certificates will be provided. If such functionality is supported, it will likely not fully comply with ETSI PKI architecture as outlined in section 6.5.2 and instead be implemented in a more simple way where essentially all entities are combined on a single physical cloud server.

Authentication and authorization of DENM and CAM will be performed based on the end-user identities associated with the deployed certificates.

Integrity protection of messages will be applied using a suitable signature algorithm of the Digital Signature Standard recommended by the US National Institute of Standards and Technology (NIST), e.g., ECDSA with nistp256 curve and SHA256 as recommended in ETSI TS 103 097 [19]. The exact details of the signature algorithms employed by the test system are to be defined.

V2V and V2I (i.e., LTE sidelink/PC5) one-to-one unicast services supported by the ConVeX test system can be secured by using the security mechanisms specified in clause 6.5 of 3GPP 33.303 [31].

For unicast services requiring TCP/IP or UDP/IP a suitable choice for end-to-end security protection between ITS stations and the cloud and between ITS stations is to use certificate-based Transport Layer Security, using TLS 1.2 [28] or DTLS 1.2 [29].

Cryptographic functions can be executed in a secure environment using the Hardware Security Module (HSM) of the C-V2X communication platform. Any sensitive data such as private keys associated with the employed certificates can be stored securely on the HSM.

6.6 In-Vehicle Internal Communication

This chapter describes the communication protocol used between the vehicle and the communication platform.

6.6.1 CAN Communication

As described in chapter 5.2 a major part of vehicle internal information is transmitted via CAN frames. Figure 6.6-1 gives an overview about the structure of a CAN frame. It contains various bits of information (Data) and is recognized by an identifier (Arbitration Field).

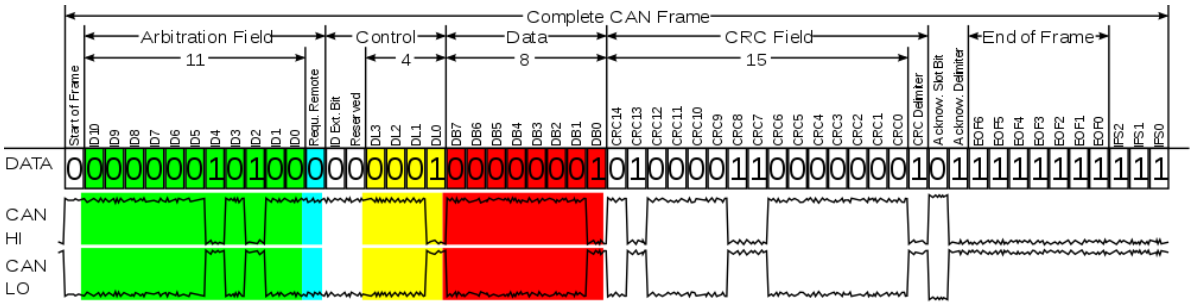

Figure 6.6-1: CAN frame [Source: Wikimedia Commons]

The payload in a CAN frame contains the actual information, which is then proprietary to automakers. The specific information of the coding of the frames and identifiers can be defined in a database. The .dbc format file specified by Vector Informatik GmbH is an example, which is intended to be used in this project. The transport layer is specified in ISO 15765-2.

Over the CAN, basic information such as speed, heading, brake operations, accelerations etc. is accessible for the communication platform. Other information can be made available via Ethernet.

6.6.2 Ethernet

Since the function ECU is connected via Ethernet to the communication platform it can support a variety of IP-based communication.

If something is to be displayed on the displays, the communication shall use Ethernet to the Function ECU. The content to be displayed should be stored on the Function ECU. A display of dynamic content that is transmitted via Ethernet is also possible. Due to potential incompatibilities, this is not recommended.

6.6.3 Displaying Information

In order to visualize an information which is specific to a use case or a function, it is necessary to do some preparation. In case a use case requires a certain image or icon this icon should be available in the deployed version of the software in the Function Hardware. Additionally, the image needs to be referenced from the use case specific components on the Connection Platform.

Therefore, adding new imagery requires adjustments on the communication and visualization components. Figure 6.6-2 gives an overview about this process.

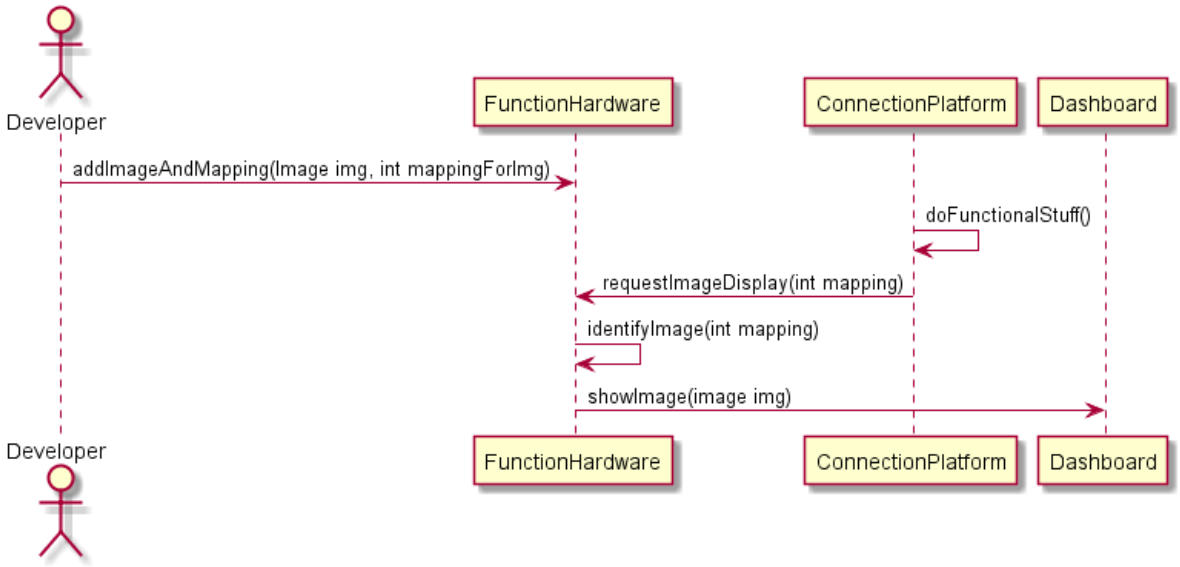

Figure 6.6-2: Sequence of displaying an image

6.6.4 Forwarding Information to the C-V2X CP

Since the Function Hardware also has access to additional information from the car, it can work as an intermediate interface between the car itself and the C-V2X CP. As mentioned in Section 5.2 it might be possible to implement additional Use Cases on the Function Hardware, and use only the modem functionality of the C-V2X DP. Figure 6.6-3 gives an overview about the sequence (here for the transmission case).

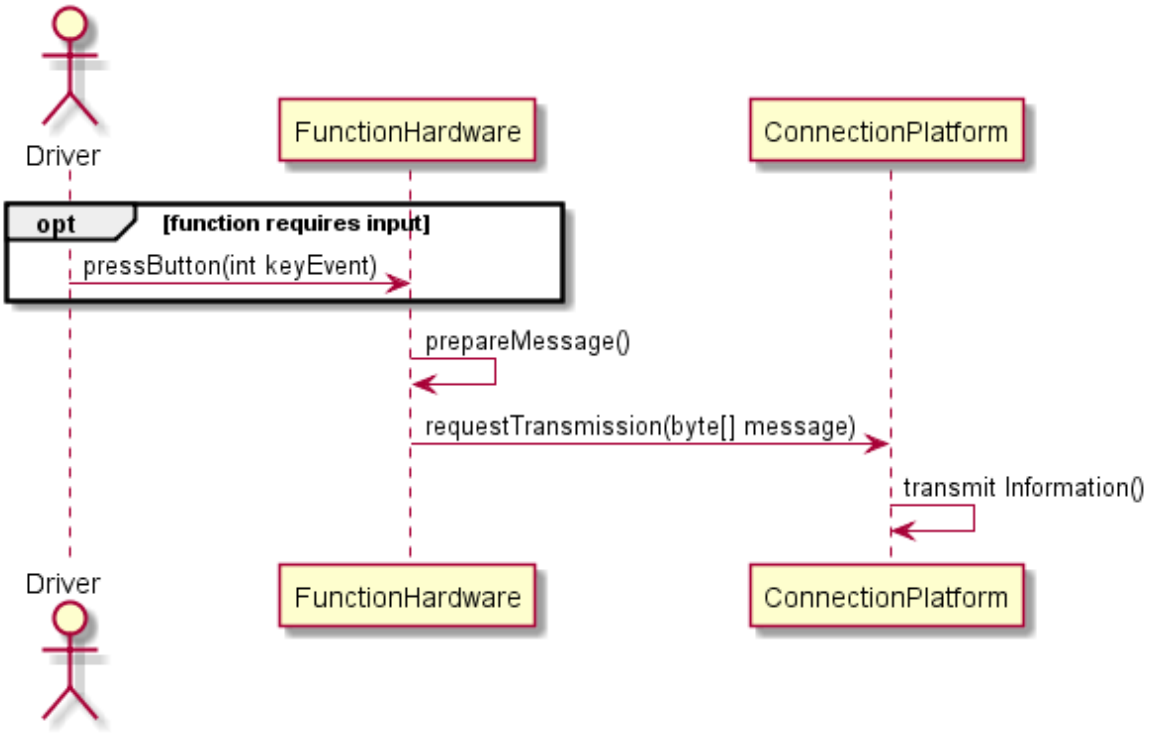

Figure 6.6-3: Sequence of sending a message

7 Simulation Platform

7.1 Introduction

In general, there are four methods to assess the performance of any system: Inspection, Analysis, Simulation and Measurement/Experiment. Though a general recommendation cannot be made regarding which method is best suited for a given scenario, it is possible to select a method based on empirical evidence and lessons learned in the field of wireless networking. This section only highlights the simulation methodology that is planned to be used as part of the project.

Existing wireless simulation techniques emulate the behavior of a given radio access technology over time with respect to parameters such as Throughput, Packet Error Rate (PER), Reliability, channel access delay etc. When applying similar methods to simulate the behavior of vehicular networking technologies, the following points need to be considered [32].

1. Simulation of vehicle's motion, which is unique to road traffic when compared to human mobility models, needs to be modeled in a more realistic (microscopic) way than the traditionally used random waypoint models. The level of detail required for realistically modeling the vehicular mobility in illustrated in Figure 7.1-1.
2. Since the purpose of vehicular communication is to improve road safety and efficiency, application dependent metrics such as CO2 emissions, Travel Time etc. need to be considered for the end-to-end simulation.

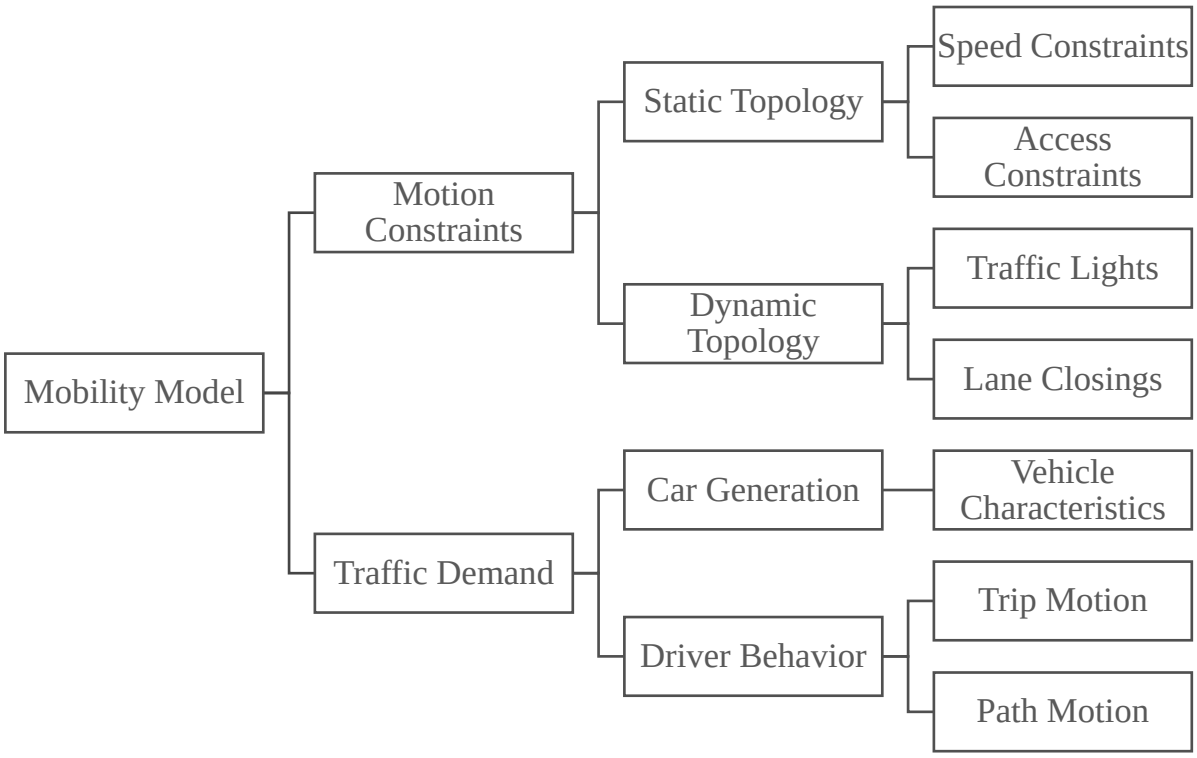

Figure 7.1-1: Microscopic Modeling of Vehicular Mobility [32]

Given the factors listed above, we can see that we need two modules interacting closely with each other – the road traffic simulation that creates and maintains the state of all vehicles in the network and the network simulation that calculates the network KPI's per given state of the vehicles. Combining these two modules into one simulator would be a fairly complex task and probably would also be very slow in terms of execution time. Secondly, we also lose on the level of detail

provided by the existing network and traffic simulators due to abstraction. Hence, we follow the approach of reuse by coupling where we integrate the existing road and network simulators by means of an interface.

Figure 7.1-2 shows the logical structure of a bidirectionally coupled simulation [33] [34]. Here two independent simulation frameworks for road and network are running in alternate phases as follows

- While the network simulation is running, it sends parameter changes to the road traffic simulation, altering driver behavior or road attributes, and influencing vehicle's routing decisions
- At regular intervals controlled by the network simulator, the road traffic simulation performs traffic computations that are based on these new parameters and sends vehicle movement updates to the network simulation

Such an approach allows us to use any of the existing frameworks as long as they offer an Application Programming Interface (API) function that can be called in order to access and change the internal parameters of the simulator.

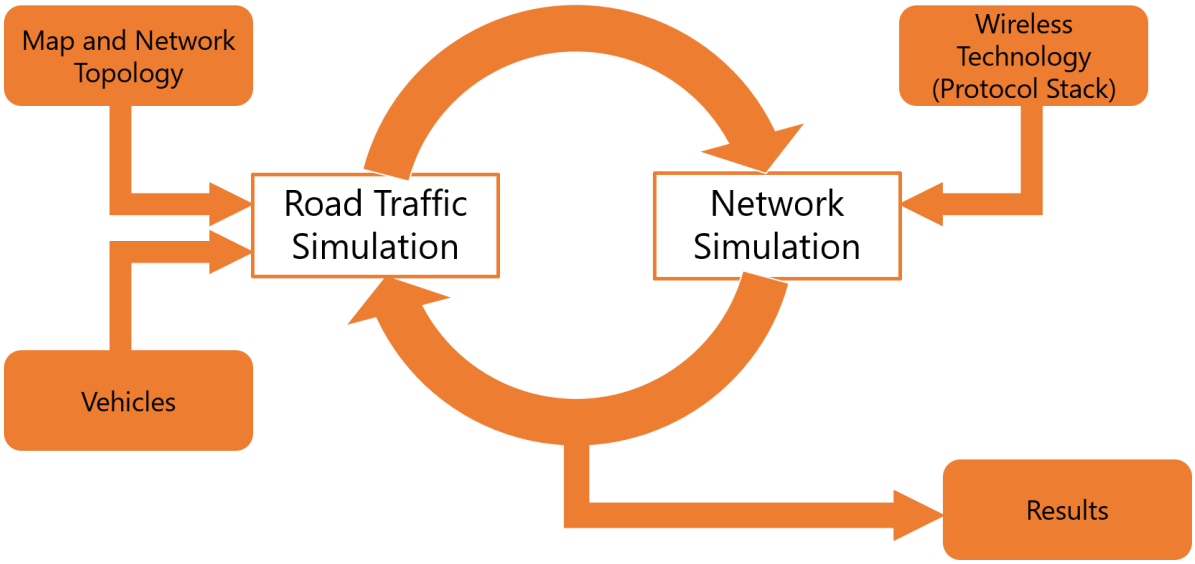

Figure 7.1-2: Bidirectionally Coupled Simulation

7.2 Overview of Current Simulators

In this section, we carry out a brief survey about the available traffic and network simulation frameworks, highlighting the pros and cons of each framework. The survey is done separately for both network and road traffic simulators

7.2.1 Road Traffic Simulation Frameworks

The following road traffic simulation frameworks are considered for survey

1. Simulation of Urban Mobility (SUMO)
2. Quadstone Paramics Modeller
3. Treiber's Microsimulation of Road Traffic
4. Aimsun
5. Trafficware SimTraffic

6. CORSIM TRAFVU
7. Vissim

The survey criteria that are considered are as follows

1. Kind of License (Opensource / Commercial)
2. Operating System Portability
3. Documentation and User Interface
4. Method of creating road networks
5. Method of creating vehicle traffic
6. GUI Quality and 3D compatibility
7. Ability to simulate very large networks
8. Simulation Output and data files
9. Supported vehicle types
10. Support for public transport and stops
11. Support for simulation of pedestrians
12. Manual configuration of traffic lights

7.2.1.1 Simulation of Urban Mobility

Table 7.2-1: SUMO

Criteria	
License	OpenSource
Operating System portability	Windows, Linux and Mac OSX
Documentation and User Interface	Well Documented & Available
Creation of Road networks	• Manual • Import from OSM and other formats • Automatic network generator
Creation of vehicle traffic	• Flow definitions and turning ratios • OD matrices • Random route generator
GUI and 3D compatibility	2D with texture customization for different vehicle types, pedestrians
Ability to simulate large networks	Yes 10,000 edges (roads) and >150,000 vehicles simulated
Simulation output and data files	• Network state dump • Lane/Edge dump • Data collection points (detectors) • Vehicle emissions • Vehicle states dump • Travel times • Traffic lights coupled output
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	Yes
Pedestrians	Yes
Traffic Light configuration manual	Yes with XML configuration file

7.2.1.2 Quadstone Paramics Modeller

Table 7.2-2: Qudstone Paramics Modeller

Criteria	
License	Commercial
Operating System portability	Windows only
Documentation and User Interface	Well Documented & Available
Creation of Road networks	Automatic network generation wizard
Creation of vehicle traffic	OD Matrices
GUI and 3D compatibility	2D and 3D including fine texture customization
Ability to simulate large networks	Yes but 3D visualization slows down the simulation
Simulation output and data files	• Queuing patterns • Demand troughs and peaks • Speed and density • Journey times • others
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	Yes (Full version)
Pedestrians	Yes
Traffic Light configuration manual	NA

7.2.1.3 Treiber's Microsimulation of Road Traffic

Table 7.2-3: Treiber's Tool

Criteria	
License	OpenSource
Operating System portability	Windows, Linux and Mac OSX
Documentation and User Interface	Well Documented & Available
Creation of Road networks	• Predefined scenarios • Build scenarious using JSON
Creation of vehicle traffic	Statistical distribution (vehicles amitted per hour from a certain intersection)
GUI and 3D compatibility	2D
Ability to simulate large networks	No
Simulation output and data files	NA
Supported Vehicle types	Cars and Trucks only
Public Transport and stops	No
Pedestrians	No
Traffic Light configuration manual	No

7.2.1.4 Aimsun

Table 7.2-4: Aimsun

Criteria	
License	Commercial
Operating System portability	Windows only
Documentation and User Interface	Not available freely
Creation of Road networks	Manual drawing using the available graphical network editor
Creation of vehicle traffic	• OD matrices • Vehicle route randomization
GUI and 3D compatibility	• 2D • 3D with reasonable quality
Ability to simulate large networks	Yes with Full version
Simulation output and data files	• Includes more than 20 different view styles of statistical information about traffic and events. • Full version only
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	Yes
Pedestrians	Yes
Traffic Light configuration manual	NA

7.2.1.5 Trafficware SimTraffic

Table 7.2-5: Trafficware Sim Traffic

Criteria	
License	Commercial
Operating System portability	Windows only
Documentation and User Interface	Well Documented & Available
Creation of Road networks	Predefined Scenarios
Creation of vehicle traffic	Automatic trip generation using TripGen2014
GUI and 3D compatibility	• 2D • 3D available on request
Ability to simulate large networks	Yes with Full version
Simulation output and data files	Full version only
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	NA
Pedestrians	Yes

Traffic Light configuration manual	Yes

7.2.1.6 CORSIM TRAFVU

Table 7.2-6: CORSIM TRAFVU

Criteria	
License	Commercial
Operating System portability	Windows only
Documentation and User Interface	Well Documented & Available
Creation of Road networks	Predefined Scenarios
Creation of vehicle traffic	NA
GUI and 3D compatibility	2D
Ability to simulate large networks	Yes with Full version
Simulation output and data files	Graphical player only. Hence no outputs available
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	Yes (Full Version)
Pedestrians	NA
Traffic Light configuration manual	NA

7.2.1.7 Vissim

Table 7.2-7: Vissim

Criteria	
License	Commercial
Operating System portability	Windows only
Documentation and User Interface	Well Documented & Available
Creation of Road networks	• Graphical Modeling and Editing • Import networks
Creation of vehicle traffic	Trip generation using statistical distributions
GUI and 3D compatibility	2D/3D and render meshes
Ability to simulate large networks	Yes with Full version
Simulation output and data files	• Data collection points • Signal control • Travel times • Delay segments • Nodes • Vehicles

	• Edges
Supported Vehicle types	Various vehicles including custom vehicle types
Public Transport and stops	Yes
Pedestrians	Yes
Traffic Light configuration manual	Yes

7.2.2 Network Simulation Frameworks

In the area of wireless networking, there are currently three simulation frameworks that are in widespread use. They are

1. Network Simulator (NS-2 and the newer version NS-3 written in C++)
2. OMNET++ (C++)

In the project, we mainly use the network simulation framework to evaluate the performance of available vehicular networking protocols. Hence, the following criteria are used for survey

1. License
2. Integration with Traffic Simulator
3. Support for 802.11p
4. Support for LTE (Including Sidelink Rel.14)
5. Documentation

7.2.2.1 Network Simulator (NS-3)

Table 7.2-8: NS-3

Criteria	
License	OpenSource
Integration with Traffic Simulator	• **Offline integration** – Mobility traces generated by SUMO can be imported into NS3 • **Online Integration** –Tools like iTETRIS or VSimRTI facilitate online interaction. But no stable interface exists for seamless integration
Support for 802.11p	Support for WAVE involves the MAC, its MAC extension (1609.4) and PHY layers.
Support for LTE	Release 12 LTE module (LENA) http://networks.cttc.es/mobile-networks/software-tools/lena/ LTE-A proximity based services (D2D) available in GitHub https://github.com/makhtardiouf/d2d
Documentation	Well documented and comprehensive

7.2.2.2 OMNET++

Table 7.2-9: OMNET++

Criteria	
License	Non Commercial Academic Use
Integration with Traffic Simulator	Supports both offline and online integration. Online integration via VEINS module using Traci interface
Support for 802.11p	**Using VEINS module**

	Dedicated models of IEEE 802.11p and IEEE 1609.4 DSRC PHY and MAC layers, including Access Categories for QoS, Wave Short Message (WSM) handling, and beaconing WAVE service announcements, as well as multi channel operation, i.e. the periodic switching between the Control Channel (CCH) and Service Channels (SCHs)
Support for LTE	**Using simulte module with some limitations such as** No Control Plane implementation Only FDD No Handovers
Documentation	Well documented and comprehensive

7.3 Selection of Frameworks for the Project

For the road traffic simulation, SUMO was selected since it is opensource and has support for multiple car following and lane changing models that would make the road traffic more realistic.

For the network simulation, both OMNET++ and NS-3 are equally competing candidates. However, OMNET++ provides seamless online integration support with SUMO (with the VEINS module and Traci interface) and hence would be the optimal choice for simulations.

7.4 Simulation Framework Overview and Architecture

In this section, we outline the logical architecture of the simulation framework that is planned to be used in the project.

7.4.1 Overall Architecture

D1.1 proposes 12 use cases covering traffic safety, efficiency and infotainment services along with corresponding scenarios (Motorway, Urban or Rural) for which the given use case is planned to be demonstrated. For the simulation of a use case, the relevant road network is constructed based on the scenario.

Figure 7.3-1 shows the overall architecture of the simulation methodology. It can be seen that the simulation is use case dependent for which a road network is constructed based on the scenario. After creating the road network, the vehicular traffic is generated based on the given vehicle types and required density distributions. The generated road network along with the traffic model is given as input to SUMO. During runtime, SUMO is called by the OMNET++ simulator Traci interface during which the input files are loaded and the list of vehicles and their states (speed, direction etc.) for a single timestep are calculated and passed on to OMNET++. OMNET++ in turn calculates the relevant network parameters (Packet statistics, link budgets etc.) based on the selected radio access protocol (802.11p, LTE etc.) and passes the turn to SUMO to execute the subsequent time step and return the vehicle states. This continues in a round robin fashion until no events necessitating the change in vehicle's flow happens. In case of such an event (e.g., accident) OMNET++ sends a flag to the SUMO to either stop the vehicles at the accident spot or reroute them using alternate routes.

Let us consider simulation flow with the help of an example use case – shockwave damping that can be applied in highway scenarios. A straight road network spanning 1-5kms with multiple segments is created along with vehicle flows from one end to the other. RSUs are placed along the length of the constructed highway that send ITS Messages to the vehicles in the proximity. The simulation is started by running the script from OMNET++ that calls SUMO using Traci interface

which loads the created road network and initializes the vehicles flow. At each time step, the vehicle states are provided by the SUMO to the OMNET++ which then simulates the flow of ITS messages between vehicles and RSU's. This process is continued until a user defined event (harsh deceleration of a randomly selected car) is injected into the SUMO by OMNET++ via the traci interface. This results in the creation of a jam (Shockwave) that is identified by OMNET++ by means of the CAMs sent out by the participating vehicles. Then the optimal speeds that result in fast mitigation of the resulting shockwave are calculated in OMNET++ which sends out these speed advisories to the oncoming vehicles by means of the RSU and also to SUMO. SUMO then applies these speed limitations on the oncoming vehicles thereby mitigating the effect of shockwave over time. Various speed regulatory algorithms can be evaluated by means of comparing parameters such as average travel time, shockwave dissipation time, road capacity over time etc. Additionally, the success probability of ITS messages when using a specific radio access protocol can also be empirically calculated.

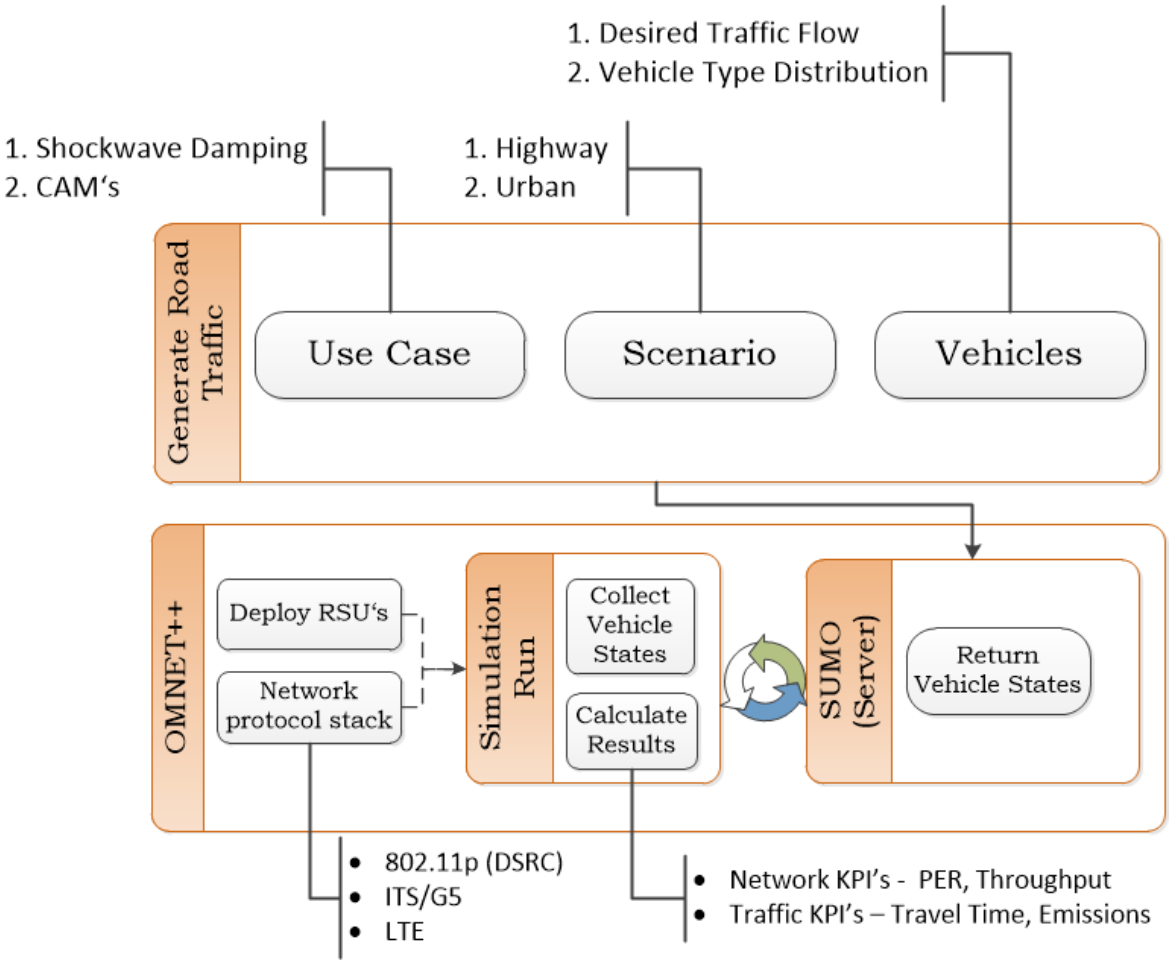

Figure 7.3-1: Simulation Architecture

7.4.2 Functional Components

The following functional components are used

1. SUMO
2. OMNET++
3. Python to generate necessary xml files for road traffic simulation

7.4.3 Interfaces

The interface between SUMO and OMNET++ is realized by **TraCI** (Traffic Control Interface) [35] that allows the user to retrieve the values of simulated objects (vehicles) and also to manipulate their behavior online. It uses a TCP based client/server architecture to provide access to SUMO. SUMO acts as the server that will listen to incoming connections (manipulations) from the clients (OMNET++) and depending upon the manipulation (e.g. Reroute request), executes the request and generates the new vehicle states. TraCI has been implemented in various programming languages such as Python, C++, JAVA and Matlab.

8 Summary and Conclusions

With this document, a description of the functional architecture and topology of the various system components is provided, that will be implemented for the ConVeX trial.

The main hardware components are explained in detail, as well as how the different parts connect with each other and how their interworking is achieved. Also, the simulation platform is introduced, that will be used for some aspects, that cannot be shown in reality, e.g., due to the limited number of real C-V2X enabled traffic participants.

It shall be noted that the functional design is tailored to support the use cases and requirements defined in Deliverable 1.1 [1]. Since there might be the opportunity to address further use cases, additional components or different implementations might be needed. Furthermore, some details or options are currently not finally decided on, and it is also envisioned that some choices are taken during the actual implementation phase. These will be documented in later deliverables, e.g. the deliverables D4.1 Roadside ITS Station Specification and D5.1 Vehicular ITS Station Specification are supposed to give insights in the final implementations for the car and RSU components.