# APPSTACLE

(ITEA 3 – 15017)

open standard APplication Platform
for carS and TrAnsportation vehiCLEs

## Deliverable: D 2.1

SotA Research with regard to Car2X Communication, Cloud and Network Middleware and corresponding Security Concepts

### Work Package: 2

Service Enablers in Intelligent Networks

### Task: 2.1

Car2X communication, Evaluation of Existing Radio Technologies and Wireless Communication and Corresponding Security Concepts.

| | | | |
|---|---|---|---|
| **Document Type:** | Deliverable | **Classification:** | Internal |
| **Document Version:** | final | **Contract Start Date:** | 01.01.2017 |
| **Document Preparation Date:** | 31.08.2017 | **Duration:** | 31.12.2019 |

ITEA3
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

EUREKA

# History

| Rev. | Content | Resp. Partner | Date |
|---|---|---|---|
| 0.1 | initial document structure | Laaroussi Zakaria | 12.06.2017 |
| 0.2 | Introduction | Laaroussi Zakaria | 28.06.2017 |
| 0.3 | added the history of self-driving cars | Laaroussi Zakaria | 05.07.2017 |
| 0.4 | added V2X section and communication scenarios | Laaroussi Zakaria | 13.07.2017 |
| 0.5 | added Requirements section | Laaroussi Zakaria and Sowmya Ravidas | 17.08.2017 |
| 0.6 | added Access control section | Sowmya Ravidas | 25.09.2017 |
| 0.7 | added Connectivity section | Alexios Lekidis | 30.11.2017 |
| 0.75 | added demonstration site baseline section | Jussi Haapola | 12.12.2017 |
| 0.8 | added Security section and attacks on connected cars | Laaroussi Zakaria | 15.12.2018 |
| 0.9 | added Network Security section | Alexios Lekidis | 10.01.2018 |
| 1.0 | addressing review comments | Jussi Haapola | 25.01.2018 |
| 1.1 | links to other deliverables and removed Appendix | Alexios Lekidis | 29.01.2018 |
| 1.2 | links between sections and conclusion | Laaroussi Zakaria | 29.01.2018 |

# Contents

**History** ii

**Summary** vi

**List of Abbreviations** vii

**1 Introduction** 1

1.1 Document Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

**2 State of the Art** 3

2.1 Corporation work on automotive area: from history to present . . . . . . . . . . 3

2.2 Technical frameworks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.3 Vehicle to everything . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.3.1 Vehicle-to-cloud . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.3.2 Vehicle-to-infrastructure . . . . . . . . . . . . . . . . . . . . . . . . 7

2.3.3 Vehicle-to-vehicle . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.3.4 Vehicle-to-pedestrian . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.3.5 Scenarios . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.4 Connectivity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.4.1 5G architecture and network layers . . . . . . . . . . . . . . . . . . 12

2.4.2 Pre-5G radio access technologies in V2X . . . . . . . . . . . . . . . 16

2.4.3 IoT Technology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.5 Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.5.1 Attacks on connected cars . . . . . . . . . . . . . . . . . . . . . . . 25

2.5.2 Network Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

2.5.3 Access control in IoT . . . . . . . . . . . . . . . . . . . . . . . . . . 32

**3 Usecases** 49

3.1 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

3.1.1 V2X communication services . . . . . . . . . . . . . . . . . . . . . 49

3.1.2 IoT Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

3.2 Use cases for ITS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

3.2.1 Platooning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

3.2.2 Over-The-Air updates . . . . . . . . . . . . . . . . . . . . . . . . . 52

3.2.3 Emergency warnings . . . . . . . . . . . . . . . . . . . . . . . . . . 53

3.3 Available communications solutions state of the art demonstration site baseline for use cases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

3.3.1 Notes on cellular V2X and its availability for demonstration purposes . . 56

**4 Conclusion** 58

# List of Figures

1 Vehicle to everything (V2X) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vi

2.1 Vehicle to everything communications styles. . . . . . . . . . . . . . . . . . . . . . 5
2.2 Information exchange on road construction warning. . . . . . . . . . . . . . . . . 6
2.3 Categorization of vehicular cloud services. . . . . . . . . . . . . . . . . . . . . . . 8
2.4 A 3GPP use-case. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.5 Main scenarios for V2C communication. . . . . . . . . . . . . . . . . . . . . . . 11
2.6 V2V commnuications enabled. . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.7 5G reference architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.8 Communication overview in the 5G fronthaul. . . . . . . . . . . . . . . . . . . . 14
2.9 3GPP network layers as introduced in release 8 (3G communication) . . . . . . 15
2.10 Available 5G deployment scenarios. . . . . . . . . . . . . . . . . . . . . . . . . 16
2.11 Scenarios for LPWAN deployment in NB-IoT. . . . . . . . . . . . . . . . . . . . 17
2.12 Reference architecture in LoRa . . . . . . . . . . . . . . . . . . . . . . . . . . 18
2.13 IoT Network Stack and Standards . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.14 ITS architecture. RSU, road side unit; OBU, on-board unit; AU, application unit; ECU, electronic control unit; TPM, trusted platform module. . . . . . . . 25
2.15 Communication layers of C-V2X architecture . . . . . . . . . . . . . . . . . . . 28
2.16 Types of IoT Architectures. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
2.17 Reference Architecture for Access Control Mechanisms . . . . . . . . . . . . . . 37
2.18 Example of Token-based Authorization Mechanism . . . . . . . . . . . . . . . . 38
2.19 Framework presented in [86]. . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
2.20 IoT-OAS Framework [67]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
2.21 Framework presented in [154]. . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
2.22 Authorization process described in [103] . . . . . . . . . . . . . . . . . . . . . . 45
2.23 Authorization process described in [98] . . . . . . . . . . . . . . . . . . . . . . 45
2.24 Authorization framework presented in [142]. . . . . . . . . . . . . . . . . . . . . 47

# List of Tables

2.1 NB-IoT and LTE-M differences . . . 18
2.2 Protocol Properties . . . 20
2.3 Operating System Properties . . . 22
2.4 Attacks on V2X . . . 26
2.5 Existing attacks and their detection by existing IDS . . . 31
2.6 Analysis of existing authorization frameworks for IoT . . . 40

3.1 IoT Requirements . . . 51

# Summary

This deliverable is the first deliverable of the APPSTACLE Work Package 2 " Service Enablers in Intelligent Network ". It contains the results of T1.2:"State-of-the-Art with regard to Car2X Communication, Cloud and Network Middle-ware and corresponding Security Concepts.

The first part concern a state of the art research. In this part we gave a full description of a new technology dubbed V2X as well as it's components, as those components enable the communication between the vehicle and other entities, we highlights the different existing wireless communication and radio technology to enable such technology(V2X) in addition to introduce the up-coming ones such 5G, enabling the different communication protocols will bring a lot of attacks and security threats against vehicles for that we described the efforts done by different industries and entities to secure connected vehicles in addition to some research projects for the same aim, we proposed a survey on the attacks against connected vehicles. The deliverable is closed by a collection of use-cases from connectivity perspective that will be implemented by the different APPSTACLE partners as well as services requirements for the V2X technology defined by the different standardization organization such European Telecommunications Standards Institute(ETSI) and 3rd Generation Partnership Project(3GPP).

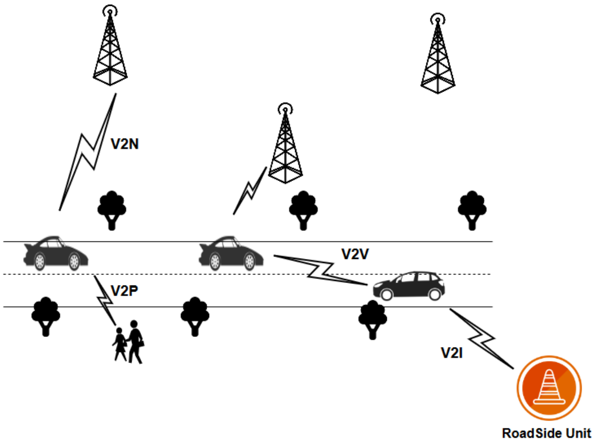


Figure 1: Vehicle to everything (V2X)

# List of Abbreviations

| | |
|---|---|
| **3GPP** | 3rd Generation Partnership Project |
| **2G** | $2^{nd}$ Generation Mobile Communications |
| **3G** | $3^{rd}$ Generation Mobile Communications |
| **4G** | $4^{th}$ Generation Mobile Communications |
| **5G** | $5^{th}$ Generation Mobile Communications |
| **5GPPP** | 5G Infrastructure Public Private Partnership |
| **5GTN** | 5G Test Network |
| **6LoWPAN** | IPv6 over Low-Power Wireless Personal Area Networks |
| **ABAC** | Attribute-Based Access Control |
| **ACL** | Access Control List |
| **ACK** | Acknowledgment |
| **AGA** | Automotive Grade Android |
| **AGL** | Automotive Grade Linux |
| **AI** | Artificial Intellingence |
| **AUTOSAR** | AUTOmotive Open System Architecture |
| **ARQ** | Automatic Repeat ReQuest |
| **BBU** | BaseBand Unit |
| **Car2X** | Car-to-Everything |
| **CITS** | Cooperative Intelligent Transportation System |
| **CoAP** | Constrained Application Protocol |
| **CoCapBac** | Community driven Capability-Based authorization framework |
| **CP-OFDM** | Cyclic-Prefix Orthogonal Frequency Division Multiplexing |
| **CPRI** | Common Public Radio Interface |
| **C-V2X** | Cellular Vehicle-to-Everything |
| **CVC** | Connected Vehicle Cloud |
| **DAC** | Discretionary Access Contro |
| **DoS** | Denial of Service |
| **DOT** | Department of Transportation |
| **DSRC** | Dedicated Short Range Communications |
| **E2E** | End-to-End |
| **ECDSA** | Elliptic Curve Digital Signature Algorithm |
| **ECU** | Electronic Control Unit |
| **EDGE** | Enhanced Data rates for GSM Evolution |
| **eDRX** | Extended Discontinuous Reception |
| **e-NodeB** | Evolved Node B |
| **EPC** | Evolved Packet Core |
| **ETSI** | European Telecommunications Standards Institute |
| **E-UTRAN** | Evolved Universal Terrestrial Radio Access Network |
| **GPRS** | General Packet Radio Service |
| **GPS** | Global Positioning System |

| | |
|---|---|
| **GSM** | Global System for Mobile Communications |
| **HARQ** | Hybrid Automatic Repeat ReQuest |
| **HTTP** | Hyper Text Transfer Protocol |
| **I2O** | In-to-Out |
| **IaaS** | Infrastructure as a Service |
| **IdP** | Identity Provider |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IoT** | Internet of Things |
| **IoV** | Internet of Vehicles |
| **IP** | Internet Protocol |
| **IPMaaS** | Identity and Policy Management as a Service |
| **IPv4** | Internet Protocol version 4 |
| **IPv6** | Internet Protocol version 6 |
| **ISM** | Industrial, Scientific, and Medical |
| **ITS** | Intelligent Transportation System |
| **LAN** | Local Area Network |
| **LLN** | Low Power Lossy Network |
| **LPWA** | Low Power Wide Area |
| **LPWAN** | Low Power Wide Area Networks |
| **LRWPAN** | Low-Rate Wireless Personal Area Networks |
| **LTE** | Long Term Evolution |
| **LTE-M** | LTE for Machine Type Communication |
| **LTE-M2** | LTE Cat M2 for Machine Type Communication |
| **LTE-MTC** | LTE-Machine Type Communication |
| **LTE-V2X** | Long Term Evolution - Vehicle-to-Everything |
| **MAC** | Medium Access Control |
| **MACo** | Mandatory Access Control |
| **MBZ** | Map of Black Zones |
| **MEC** | Mobile Edge Computing |
| **MQTT** | Message Queue Telemetry Transport |
| **MTU** | Maximum Transmission Unit |
| **NACK** | Negative Acknowledgment |
| **NB-IoT** | NarrowBand IoT |
| **NGCN** | Next Generation Converged Network |
| **NR** | New Radio |
| **O2I** | Out-to-In |
| **OEM** | Original Equipment Manufacturer |
| **OS** | Operating System |
| **OSI** | Open Systems Interconnection |
| **PaaS** | Platform as a Service |
| **PAN** | Personal Area Network |
| **PAP** | Policy Administration Point |
| **PDCP** | Packet Data Convergence Protocol |
| **PDN** | Packet Data Neywork |
| **PDP** | Policy Decision Point |
| **PDU** | Protocol Data Unit |
| **PEP** | Policy Enforcement Point |

| | |
|---|---|
| **PIP** | Policy Information Point |
| **QoE** | Quality of Experience |
| **RAN** | Radio Access Network |
| **RBAC** | Role-based Access Control |
| **RF** | Radio Frequency |
| **RFID** | Radio-Frequency Identification |
| **RH** | Radio Head |
| **RLC** | Radio Link Control |
| **RPL** | IPv6 Routing Protocol for Low-Power and Lossy Networks |
| **RSU** | RoadSide Unit |
| **SAaaS** | Software and Actuation as a Service |
| **SaaS** | Software as a Service |
| **$\mathbf{S^2aaS}$** | Sensing as a Service |
| **SDU** | Service Data Unit |
| **SDVC** | Software-Defined Vehicular Cloud Architecture |
| **SMS** | Short Messaging Service |
| **SN** | Sequence Number |
| **TB** | Transport Block |
| **TCP** | Transmission Control Protocol |
| **UE** | User Equipement |
| **UDP** | User Datagram Protocol |
| **UMTS** | Universal Mobile Telecommunications System |
| **VO** | Virtual Object |
| **VNF** | Virtualized Network Functions |
| **WAN** | Wireless Area Network |
| **WAVE** | Wireless Access in Vehicle Environments |
| **WDM** | Wavelength Division Multiplexing |
| **WiFi** | Wireless Fidelity |
| **WSN** | Wireless Sensor Network |
| **XACML** | eXtensible Access Control Markup Language |
| **XMPP** | Extensible Messaging and Presence Protocol |
| **V2C** | Vehicle-to-Cloud |
| **V2I** | Vehicle-to-Infrastructure |
| **V2N** | Vehicle-to-Network |
| **V2P** | Vehicle-to-Pedestrian |
| **V2V** | Vehicle-to-Vehicle |
| **V2X** | Vehicle-to-Everything |
| **VC** | Vehicular Cloud |

# 1 Introduction

The purpose of this document is providing a state of the art research with regard to Car-to-Everything (Car2X) Communication, Cloud and Network Middleware and corresponding Security Concepts this description it has been done in Task2.1 from the Work-package 2. The research into self-driving vehicles has been carried out since 1920 [11], with the first self-driving vehicle prototype appearing in 1980 [108], the second prototype being made by Mercedes in 1987. These prototypes focused on how to make new innovations regarding the engine and the body design of the vehicle. Recently, many companies, such as Google, Tesla, Mercedes, Bosch, Volvo, Nissan and Audi [95, 26, 20, 180], have started producing prototypes of self-driving vehicles in the automotive area. From another side, governments play the main rule in making the revolution in automotive vehicles. Indeed, many countries started have already providing the needed environment for testing autonomous vehicles in public traffic. For instance, the United Kingdom invested millions for establishing an environment where manufacturers can test their vehicles, whereas the United States of America issued new laws authorizing driverless vehicles. In Europe, many facilities have been arranged for that enable the evaluation of autonomous vehicles, these facilities allow experimentation with autonomous vehicles in real transportation systems [5, 33].

Nowadays, the vehicle design is an extremely complex product, embedding more than 100 Electronic Control Units (ECUs), as well as approximately 100 million lines of code for managing different vehicle functionalities [136, 191, 62]. The vehicle design keeps evolving which compels manufacturers to increase the on-board processing power of new vehicles. Thanks to this processing power, a technology dubbed “autonomous vehicle", has emerged. This technology will enable the vehicles to travel between cities without any human intervention.

To deal with the threats to passenger safety, the vehicle to everything (V2X) technology sprang up. Such technology enables the communication between a vehicle and other entities aiming at reducing the environmental impacts, increasing the traffic efficiency, improving the road safety, providing additional benefits to travelers, and improving the Quality of Experience (QoE)[40]. This technology can be defined through four areas: 1) vehicle-to-vehicle (V2V) 2) vehicle-to-network (V2N) 3) vehicle-to-infrastructure (V2I), and 4) vehicle-to-pedestrian (V2P). Such technologies make it easier to move from a starting point to a given destination and makes the journey more enjoyable in today’s connected vehicle, by sharing data in real time to avoid accidents, coordinate traffic and to make the vehicle more aware of its environment.

Autonomous vehicles can be divided into five levels:

- **Level 1**: The driver has full control of the vehicle.
- **Level 2**: The vehicle has a set of functionalities that can operate in an autonomous way without cooperation.
- **Level 3**: The vehicle becomes more intelligent by enabling the coordination between variant functionalities provided in level 2.
- **Level 4**: The vehicle ensures some self-driven functionalities.

- **Level 5**: The vehicle is able to take decisions and drive by itself without human intervention.

To construct a powerful intelligent transportation system (ITS), where vehicles will reach the fifth level of autonomous vehicles, vehicles should have the needed communication, storage, intelligence, management of time and latency in critical situations and the learning capabilities to anticipate the driver's intention, in order to ensure safety and improve the QoE. Such features cannot be achieved by autonomous vehicles with the software architectures currently available. Therefore, the future software architectures need to be designed in a way to support V2X technology, and the defined features, and to address the numerous issues on the roads. Our work will focus on designing an E2E architecture, aiming to meet the requirements of such architecture. The E2E architecture requires three important layers: *i*) the in-vehicle platform, *ii*) the service enablers in intelligent networks, and *iii*) the cloud IoT platform. Autonomous vehicles form a large research area from different perspectives, such as In/Ex-vehicle Connectivity, Internet of Thigns IoT, Internet of Everything (IoV), Cloud and orchestration, security and Artificial Intelligence (AI) notably machine learning. In this Document we attempted to present the research on self-driving vehicles from connectivity and security perspective in a more structured way.

## 1.1 Document Structure

This deliverable is organized as follows: Chapter 2 addresses the state of the art regarding V2X. The Chapter explores the works that have been done in the automotive area in Section 2.1 and Section 2.2 describes the current open and standardized architecture frameworks for the automotive industry. The Section 2.3 identifies the different types of connectivity required for autonomous vehicles and discusses their impact on deployable communications technologies. The Section further addresses the communications scenarios that occur in V2X domains. Section 2.4 highlights the mobile technologies leading to the $5^{th}$ generation (5G) mobile system for the connected cars as well as the scenarios of 5G communication. The Section 2.5 covers the different challenges and limitations facing the autonomous vehicles and summarizes the existing approaches to secure these vehicles.

The Chapter 3 begins with Section 3.1, which presents the requirements for V2X communications technologies, services, and IoT requirements in addition to the existing software architectures dedicated to autonomous vehicles. The Section 3.2 further describes the use cases to be addressed for automotive communications within the APPSTACLE WP2. Three use cases for ITS are described and the Section 3.3 describes the current state of the art of the demonstration sites regarding support for V2X communications on the use cases.

# 2 State of the Art

## 2.1 Corporation work on automotive area: from history to present

Although the expectation for autonomous vehicles may seem impossible for people to believe, drivers will soon be totally replaced. Companies like Ford, Mercedes, and Tesla are already testing and preparing the ground for this in the near future [68]. Although, it is important to remember when the technology started. Indeed, the first driverless vehicle were prototyped way back in the 1920s, even though they were not the self-contained models we are witnessing today [39], the vehicles were driverless but were dependent on some specific inputs.

The earliest model was developed by Houdina Radio Control in 1925 [12]. The concept was that a main vehicle was controlled by a second one, which was following it at a close distance, as it moved along the roads of New York and Milwaukee [31]. Later in the 1950s, a remote auto-driving functionality was deployed using an electrical impulse. The idea was practical due to the introduction of RCA lab detector circuits on highways. The sensor helped in following the direction, velocity, and location of other vehicles on the route to inform the autonomous vehicle [39]. The university of Ohio pursued the development of the project in the 1960s with the help of the US Bureau of Public Roads and other corporations [39]. In UK, the transport and road research laboratory conducted similar projects, which were intended to add inputs to the autonomous vehicle.

The efforts to construct autonomous vehicles have moved from academic and research prototypes to industrial models by the manufacturers of vehicles within the last ten 10 years. The industry players like BMW, Volkswagen, and General motors (GM) all have new models on the roads [39]. Also, in 2009, Google launched a self-driving vehicle able to cruise using Google maps to find the desired locations. This self-driving Google vehicle uses inputs from the radar system to detect objects, pedestrians, and other vehicles in its vicinity [19]. The data are then processed to plan a safe drive on the road.

Evidently, the journey to a fully automated driverless vehicle will soon be realized. Other manufacturers and designers joined Google and the main industry player in their efforts to test new models and add functionalities [28]. An example is Tesla Motors which released an updated version of the model S vehicle which had an autopilot capability [19]. This version allowed the vehicle to follow a lane and to switch paths to mimic overtaking and parking. Such capabilities made the S model famous among the autonomous models.

Additionally, Ford have done more than simply autonomous vehicles, the company announcing its simply plans to unveil a Ford Fusion Hybrid [41]. The forthcoming platform will have a LIDAR sensor deploy in a unique way to allow a 360-degree pattern scan [19]. The chief program engineer of the company, Chris Brewer, insists that the system will have high resolution sensors to scan its environment and determine where objects exist within its path [182]. Besides, the design comprises two cameras on the roof to detect objects and inform about the traffic lights. The radar will also help the scanning of objects at night [41]. Another player who

has shown an interest in the autonomous vehicles is the Microsoft corporation. However, the company is not ready to build such vehicles, but rather to power driverless vehicles. According to Microsoft's head of business development, Peggy Johnson, the company will continue to build the software to assist the vehicles [37]. The corporation is ready to produce an operating system dedicated solely to vehicles as added value compared to what other companies are working on, such as Ford and Tesla.
Companies for instances involved in autonomous vehicle technologies. At the EU level, there are several research projects such as 5GCAR [1] headed by Ericsson with more than 10 partners working together with the objective of developing a 5G V2X system architecture. This project will focus more on the connectivity provided to the vehicle.

## 2.2 Technical frameworks

With the technology industry getting more and more complex, the automotive industry is experiencing the same phenomena. With the growth of connected vehicle and the urge to manage accidents and improve infotainment within vehicles, a number of initiatives have emerged to make this a reality. Most of these initiatives are open source and their main objective is to create open and standardized software architectures for the automotive industry. These initiatives include AUTOmotive Open System Architecture (AUTOSAR), Automotive Grade Linux (AGL), GENIVI, and Automotive Grade Android (AGA) and other Open-source platforms that are well described in D1.1 Specification of In-car Software Architecture for Car2X Applications and D3.1 Specification of Data Management, Cloud Platform Architecture and Features of the Automotive IoT Cloud Platform. Since the autonomous vehicle is expected to have the capability of evaluating the environment and make decisions, an E2E-secured architecture that grants V2X communication in the most optimized manner is mandatory. Undoubtedly, vehicles have become a major part of the Internet of things (IoT).For that, having the sensing capabilities in addition to the V2X technology enabled within the self-driving vehicles is mandatory to avoid accidents, increase the QoE and provide an efficient driving experience. In the coming section we will try to define the V2X technology and their components and how this technology is contributing on building the future vehicle.

## 2.3 Vehicle to everything

The U.S. Department of transportation has already issued a standards, requiring that all new vehicles should be compliant with V2X setups by 2023. Currently, some companies have already met this condition, six years before the deadline. One example of a vehicle model that already meets this requirement is the 2017 Cadillac CTS that is in-built with DSRC short-range radio communication gadgets, which relay data such as heading, speed, and Global Positioning System (GPS) locations [76]. The most fundamental advantage of vehicles possessing the ability to communicate with another entity is safety. One obvious thing is that when a vehicle has information regarding the input (e.g, speed or position) of another vehicle, the two should never collide [116].

In addition to safety, there are other benefits of V2X technology such as enhanced traffic management, driver assistance, law enforcement and emergency services. Nevertheless, the system is also susceptible to several limitations such as breakdown and increased cost of installation for the consumer.

Vehicle to everything communication will transform the transportation systems by improving the efficiency in controlling the traffic and averting accidents. However, it has numerous obstacles, for instance, in cases where smartphones are used; the reliability will be decreased since mobile gadgets are energy-limited and they use constrained batteries. Once the battery is empty, the pedestrian or motorist can no longer have access to services. Also, connecting a vehicle to everything should avoid mobility issues [46], improve safety [9], keep the vehicle aware of the environment traffic and road conditions [40] without influencing the comfort of passengers [40].

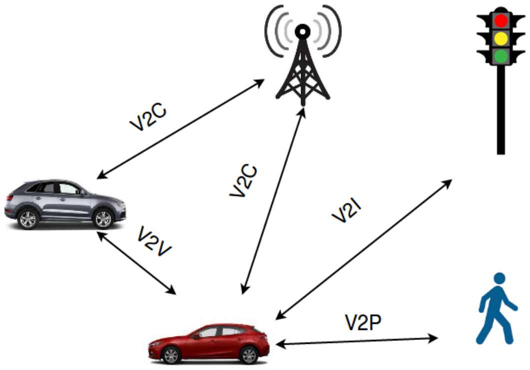


Figure 2.1: Vehicle to everything communications styles.

Figure 2.1 above depicts the communication in the V2X design as being bidirectional. For example, a vehicle can send messages to a cloud as well as to other entities. Similarly, the V2V and V2P links enable message transmissions based on the broadcast technology, either between vehicles or users on the road [40]. Indeed, V2X seeks to satisfy the following scenarios in a self-driving vehicle:

- ***Safety and reliability:*** the technology allows a vehicle to sense an emergency brake light, identify blind spots, change lanes, and provide a warning about upcoming collisions [9].
- ***Situational awareness:*** it alerts the vehicle about longer latency requirements such as hazardous road conditions [40].
- ***Mobility issues:*** the technology helps the vehicle in supporting inter-modal travel, power constraints, and complex security matters [40]. For example, the vehicle requires V2X while parking or sensing a toll system [46].
- ***Auxiliary comfort:*** it describes the conditions where more processing power is required for route planning, map disseminating, fleet managing, etc [40].

Indeed, for such use cases, V2X utilizes the Wide Access Network (WAN), Wireless Fidelity (WiFi), and Wireless Access in Vehicle Environments (WAVE) [40]. Moreover, LTE-based V2X is even available in some vehicles since it is a new technology [120]. Besides, V2X is the central technology in the development of vehicles, due to the fact that it requires spatial orientation and human equivalent perception of various situations [121]. Moreover, the wide range sensors in the vehicle coordinate with the V2X system to meet the working scenarios which drivers encounter daily [132].

Notably, the V2X technology has different components which help the vehicle in attaining the capabilities of a real driver [53]. For that reason, the vehicle should communicate with different components as humans do. Actually, vehicles can share data using direct wireless connectivity or radio technology.

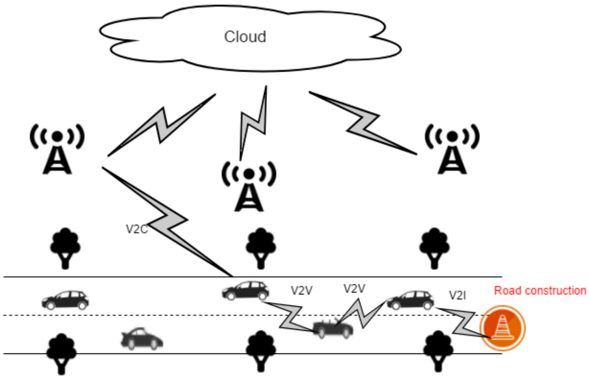


Figure 2.2: Information exchange on road construction warning.

The figure 2.2 presents a real use case for connected vehicles where a vehicle is warned about construction work in the road, and how the message is transmitted from the infrastructure to the vehicle, then to the cloud. In fact, this is an advantage because enabling such technology, V2X helps the human driver in obtaining information about obstacles, traffic, accidents and constructions on the road much quicker than using the vehicle sensors alone.

Notably, the V2X technology brings a new ecosystem dubbed "the (IoV)" where the connected vehicle is a part of IoT and where the goal is to connect things in a smart way [45]. The IoV refers to the IoT frameworks reserved for vehicles. Soumaya K. et al.[69] look into the growth of the IoV with the purpose of designing a new IoV architecture dedicated to vehicles. This is carried out by merging cloud computing and IoT aspects. The requirements for smart cities for the next generation of vehicles (standards, V2X Hardware, etc.) defined by the NTT innovation institute motivated them to study those requirements and the research progress for the connected vehicle. The IoV architecture in [69] is based on the results of two reports [3] [130]. The auto 3.0 report [130] made by NTT defines the next steps of designing the vehicle of the future by enabling the V2X technology and the automatic detection of data generated by vehicles. In the auto 4.0 report, the vehicle of the future is meant to have self-driving capabilities [3]. Indeed, such architectures dedicated for vehicles require security for the IoV as a main component (e.g., to detect a Denial of Service (DoS) attack) [181].

In this section, we shed light on existing vehicle communications, also the research and efforts made to enable such technologies with the purpose of connecting the vehicle to other entities.

### 2.3.1 Vehicle-to-cloud

Vehicle-to-cloud refers to the ability of the vehicle to connect to a cloud and access pertinent data for the completion of the task or request [96] [101]. A cloud refers to the generalized delivery of services hosted over the Internet, allowing the consumption of computer resources, like virtual machines, storage, or software applications, in a similar approach as one would consume a household utility[96] [119]. The Internet of Things (IoT) has allowed the creation of connected vehicles through the use of “a novel multi-layered vehicular data cloud platform out" through the use of the IoT and cloud computing [96]. By utilizing cloud services, vehicles are provided with the ability to access real-time data on information that are relevant to the situation in which the drivers of the vehicle may find themselves. As this data is provided to the owners of the vehicles, big data is collected from those vehicles on the traffic patterns and other concerns surrounding the vehicle, using that information to be able to provide other passing vehicles with similar information on traffic patterns, weather patterns, etc [96]. Vehicle-to-cloud communication offers the functionality to link vehicles and other gadgets through the cloud without affecting security. Vehicles are securely linked with the help of a bi-directional communication link that supports multiple protocols as well as interfaces like: MQTT, CoAP, HTTP, and SMS Shoulder Tap.

Status data and information are gathered and relayed to the Connected Vehicle Cloud (CVC), where it undergoes normalization, storage, aggregation, and is merged with other information from other sensors and systems. This data is distributed (partially or fully) to the CVC applications as well as to actors who have appropriate access rights. The firmware update functionality gives the OEM the opportunity to conduct over-the-air updates of firmware and software on-board units [65].The CVC serves as the cache that relays software updates to many vehicles. Furthermore, it offers business rules as well as a scheduling functionality that allows the OEM to manage the software file to be offered to vehicles. A new Paradigm, dubbed Vehicular Cloud (VC) defines the capability of merging cloud services with the vehicle networks [135]. This paradigm brings new services[134, 52, 71, 152] reserved for vehicles. Those services are listed in figure 2.3. Many work in the literature have tried to design architectures based on the VC paradigm. In [104], the authors highlighted the limitations of the VC and proposed a software-defined vehicular cloud architecture (SDVC) with the intention of addressing these limitations. The SDVC has two main components: the first component, dubbed “Data plane", considers all data exchanged between vehicles and RSUs, the second one, dubbed “Control plane", collects the data, provides predictions about the mobility and creates the necessary resources. The colossal amount of generated data can help in building prediction frameworks. Indeed, Ke Zhang et al.[192] proposed a Multi-access Mobile Edge Computing (MEC) offloading framework for vehicular networks based on a prediction strategy for task-file uploading in addition to enable the V2V communication in the offloading strategy. In order to avoid the latency and delay, the data are exchanged between vehicles instead of transmitting them to the cloud. One required parameter that must be taken into account to enable such communication it’s the high availability and accessibility of data. A probabilistic strategy for content caching on the edge of 5G networks is proposed in [127] while [171][170] propose a solution, dubbed “Follow-Me Cloud," where cloud services follow mobile users.

The V2C protocol will be used in the Update over the air usecase that will be developed in the Appstacle project within the WP2.

### 2.3.2 Vehicle-to-infrastructure

Vehicle-to-infrastructure is a feature that allows vehicles to communicate with infrastructures, such as the municipal traffic system to realize, an intelligent prediction solution regarding traffic conditions. One of the models incorporating V2I are the Audi A4, Q7, as well as the road vehicle models developed from 1st of June, 2016. Moreover, the entire 1,300-traffic signals managed by the Regional Transportation Commission of Southern Nevada have the infrastructure that

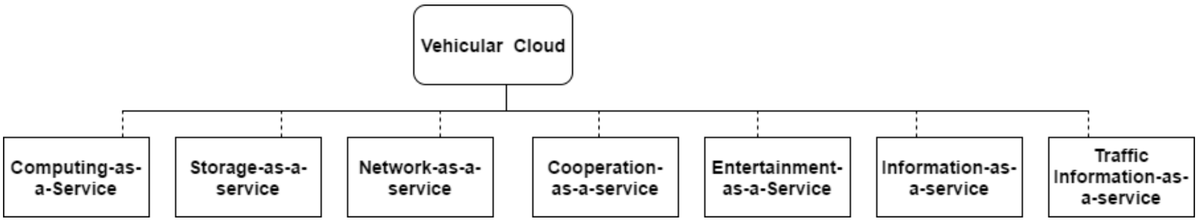


Figure 2.3: Categorization of vehicular cloud services.

can communicate with these vehicles. One use case where this technology uses the advantage of V2I is to avoid drivers from waiting for lights changes by giving them the real-time updates. This informs the drivers the length of time they will spend at intersections.

The V2I system also announces the arrival of an era where vehicles communicate with different city entities, and vice versa, aiming at mutual benefit. As municipalities make instant data traffic available, relevant information can be provided to motorist′ dashboards. Traffic management officials across the world are confident that V2I will lower congestion around major towns.

Dedicated short range communication (DSRC) is a critical for vehicle-to-infrastructure, vehicle-to-vehicle, and vehicle-to-roadside communications [54]. DSRC is a key enabler for the (ITS) [54]. The ITS is considered to be an advanced application that provides the services associated with transport and traffic management, ensuring that the computer within the vehicle, the tools available to the user, and the presentation modes of information are all working toward the goal of increasing the level of information held, safety, and generally providing the means to apply that knowledge in a manner that allows for increased efficiencies and effectiveness on the user side[47].

### 2.3.3 Vehicle-to-vehicle

The V2V communications refers to the transmission of data between two vehicles using wireless communications [84] [83]. The ultimate goal of this particular type of technology is to provide the driver with real-time updates to prevent accidents on the road through the communication of position and speed data across an ad-hoc mesh network. Depending on the capabilities of the vehicle, the result would either be a notification to the driver or the on-board computer for overtaking over to implement evasive maneuvers that prevent accidents [84][83]. V2V communication gives a vehicle the ability to send information to enhance safety, travel times, and has a central role in autonomous driving. Whereas there is a high possibility that V2V communication will be available on the market, there is also a possibility that they will undergo a fast transition to V2V communication with these technology appearing as a basic attribute in upcoming vehicle models. Indeed, the United States and Europe move forwards the deployment of V2V, with the objectives of using such technology to exchange information, bypass accidents and build a cooperative intelligent transportation system (CITS) [176].

The V2V provides vehicles with the capability of peer-to-peer communication in an endeavor to forewarn motorists about an impeding accident and avoid crashes. It employs DSRC to facilitate the communication between vehicles and relays data such as braking status, direction, speed, and location [64]. The U.S Department of Transportation (DOT) asserts that the technology is powerful unlike its predecessors. The radars, camera sensors and the radio technology in V2V technology can work up to a distance of 300 meters, and is not impaired by obstacles or other vehicles [186].

The technology is instrumental in warning motorists of impeding dangers, specifically while changing lanes or turning at junctions. Moreover, the DOT also reports to vehicles with an automated driving system which is poised for even greater benefits in the reinforcement of V2V data thus reducing the risks of accidents. According to the DOT, the application of V2I and V2V could lower by about 80% the incidence of non-impaired accidents. The agency further claims that once V2V communication is deployed, it will offer a 360-degree situational consciousness on the highway and will increase vehicle safety [75].

The V2V technology takes into consideration the personal privacy of motorists. The DOT asserts that no confidential information regarding the vehicle or driver will be transmitted using V2V. There are also other efforts to ensure that the system is secure from a cyber security perspective to make sure that the information sent is safeguarded from all types of digital attacks.

The V2V protocol will be used in the platooning usecase that will be developed in the APPSTACLE project within the WP2.

### 2.3.4 Vehicle-to-pedestrian

Vehicle-to-Person, or Vehicle-to-Pedestrian (V2P), refers to the ability of an individual other than the owner/operator of the vehicle to be able to use the vehicle as a wireless hotspot [51] [43] [117]. The primary purpose behind this connectivity is to increase the safety of pedestrians, allowing vehicles to transmit safety warnings to pedestrians and decrease the likelihood of an accident [43] [117]. The goal is to ultimately be able to use this technology to effectively communicate the speed, location, and trajectory of a vehicle to a pedestrian's connected device as a means of decreasing the likelihood of an accident [43][117]. Current limitations associated with cellular technology have made its application difficult due to the energy constraints associated with the cellular technologies available today [117]. Currently, pedestrians are allowed to exchange safety communication with vehicles via mobile gadgets like smartphones to avert V2P collisions [51]. Nonetheless, unlike vehicles, the mobile gadgets are energy-limited because they operate with constrained batteries. Furthermore, the dependability of V2V safety applications can be increased avoiding the collision between security messages relayed from the mobile gadgets. In 2014, the V2P terminal system was assembled with the help of terminal equipment such as smartphones, dedicated terminals for positioning, as well as the 700 MHz-band. In the trials that were conducted on the public roads in Nagoya and Yokosuka, the system illustrated a proper 700 MHz-band arrangement between the on-board terminal and the pedestrian terminal even while making use of Bluetooth [169].

Information distribution and collection, assisted by using Web technology instead of the Mobile Phone Network [38], concentrates on danger forecast and avoidance. A Web-based system creates an efficient scheme for gathering, analyzing, and distributing the next-generation probe data to attain early execution of a setup. This setup gathers information from all types of vehicles and all types of roads [168]. For instance, as an aging society, the U.S. can expect more mobility scooters and electric wheelchairs on the roads, which would most likely result in more accidents. A motion detection system, which detects positional field intensity, and GPS for assessing the level of proximity between pedestrians, cyclists, and electric wheelchairs who are all in transit, would inform all road users of looming danger [57].

### 2.3.5 Scenarios

Since the driverless technology needs to be safe on the roads, the V2X architecture comes into play. As defined in the previous section, the V2X idea enables the communication between various cars on the road for the sake of efficiency. Since it helps in realizing the objectives of reducing the environmental impact, increasing the traffic efficiency, improving the road safety, and providing additional benefits to the travelers, we tried in this section to introduce the different scenarios related to one of the most important types of communications which is the V2C communication.

- Lack of bandwidth/throughput.
- Out-Of-Coverage zones.
- Service persistence on higher mobility.
- Connection continuity and stability.

***Out-of-Coverage:***

When the car is not connected, we should gather information about running services in order to know which services should remain running and which ones should be stopped based on several parameters such as, location, time, etc. We intend to build a framework which handles these issues through a prediction system which learns from the service usage history of the users. In addition, the framework is intended to build a map based on connectivity loss data (location of connectivity loss, location of connectivity recovery) dubbed “Map of black zones" (MBZ). This map will allow, identify and prepare in advance the services that are more likely to be used when the car is in an out-of-coverage zone.

***In-Coverage:***

When the car is connected, it can use different generations of 3GPP for its connection. Each generation is characterized by different requirements, such as bandwidth and latency. Compatibility issues can arise in cases when the used services require a higher (i.e in terms of bandwidth) 3GPP generation than the one the car is using. So, when the car is connected, we should identify the requirements of services and prepare them in advance (when the needed 3GPP generation is available) to avoid this problem and improve the QoE. Figure 2.4 depicts the case where the car goes from one 3GPP generation to another.

***O2I-Coverage:***

This scenario concerns the transition from out-of-coverage zone to in-coverage zone. The connection here is bootstrapped and based on defined service priorities, some services will be first to get updates from the Internet. Once the connectivity is restored, the exact location is added to MBZ.

***I2O-Coverage:***

As opposed to the latter scenario, this scenario concerns the transition from in-coverage zone to out-of-coverage zone. The connection loss location is stored in-car and uploaded to the cloud once the connection is restored. This will help in building MBZ.

***Partial-Coverage:***

This is a special case combining in-coverage and out-of-coverage scenarios. In such a scenario, the V2V communication is enabled. For instance, if vehicle A is out of coverage and vehicle B is within coverage, A will communicate with B to get updates from the Internet.

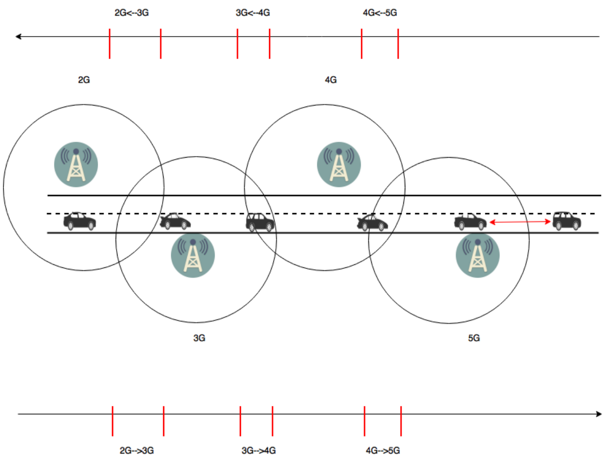


Figure 2.4: A 3GPP use-case.

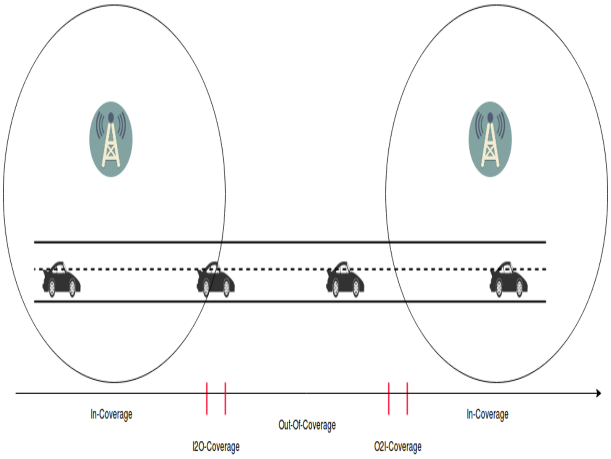


Figure 2.5: Main scenarios for V2C communication.

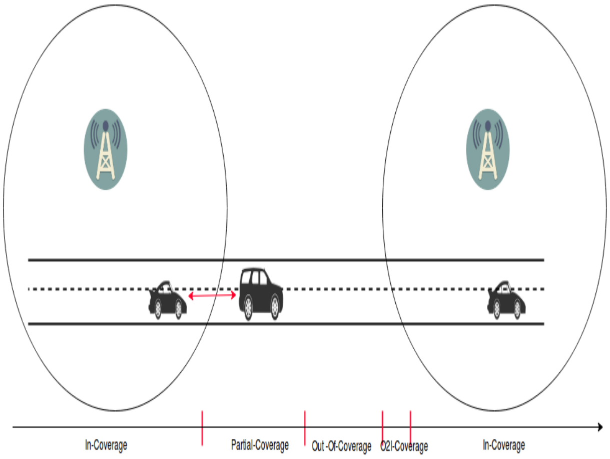


Figure 2.6: V2V commnuications enabled.

The presented technology describe how the vehicles should be aware of many events related to the network (hand offs operations, link usage, latency, etc), traffic (traffic jams, accidents, traffic hazard warnings, eco driving etc), and pedestrians. This motivated the next section where we listed the services requirements defined by 3GPP and ETSI to handle the different events.

## 2.4 Connectivity

In deliverable D1.1: "Specification of In-car Software Architecture for Car2X Applications" 3 main categories of are identified:

1. 5G radio access technologies
2. Pre-5G radio access technologies
3. Non-cellular technologies providing wireless access

As the third category was thoroughly presented in the same deliverable, this document will focus on the other two categories, namely, 5G radio access technologies in Section 2.4.1 and Pre-5G radio access technologies in Section 2.4.2.

### 2.4.1 5G architecture and network layers

Since 5G communication is in process of being standardized, we hereby focus on describing the main concepts that will be part of 5G communication as the initial prototypes and test trials. The 4G technology first introduced the Evolved Packet Core (EPC), to be the main framework for providing converged voice and data on IP networks. This allowed the cellular communication to evolve from the circuit-switched domain into a packet-switched architecture such as the one used in GPRS/UMTS. The EPC of course can be deployed as software only components and it can be virtualized, but at the same time is not flexible enough and is being deployed as a monolithic centralized component that can only cater one use case. Therefore, it faces great challenges related to the performance as well as the latency in the communication. However, as technologies move towards IoT additional use cases for LTE are arising. In order to cope with these challenges 3GPP decided to improve the EPC architecture by the Release 15 that proposes the first set of 5G standards and the maturing of LTE-V2X specifications [160].

Figure 2.7 illustrates an architectural view of 5G communication. The Figure divides the network into two parts, the user data part (also known as the user plane) and the signaling part (also known as the control plane). This separates their concerns as well as makes the scaling independent. The control plane is supported by the 5G transport network and the latter by the Radio Access Network (RAN). Additionally, the former is further divided into fronthaul and backhaul packet networks. Backhaul is the linkage between a base station and the core wired network, and is often fiber or coax, and in some cases broadband, proprietary wireless links. In most cases the backhaul network is supported by wired communications to enable less communication latency. The front hall network provides the connection between the cell tower radio itself (Radio Head or RH) and the mobile network control backbone (the Baseband Unit or BBU) and CPRI is a well-known standard for this interconnection.

The control plane includes additionally the communication with the IoT application servers and the EPC. Furthermore, the EPC comprises by two gateways the Serving and the Packet

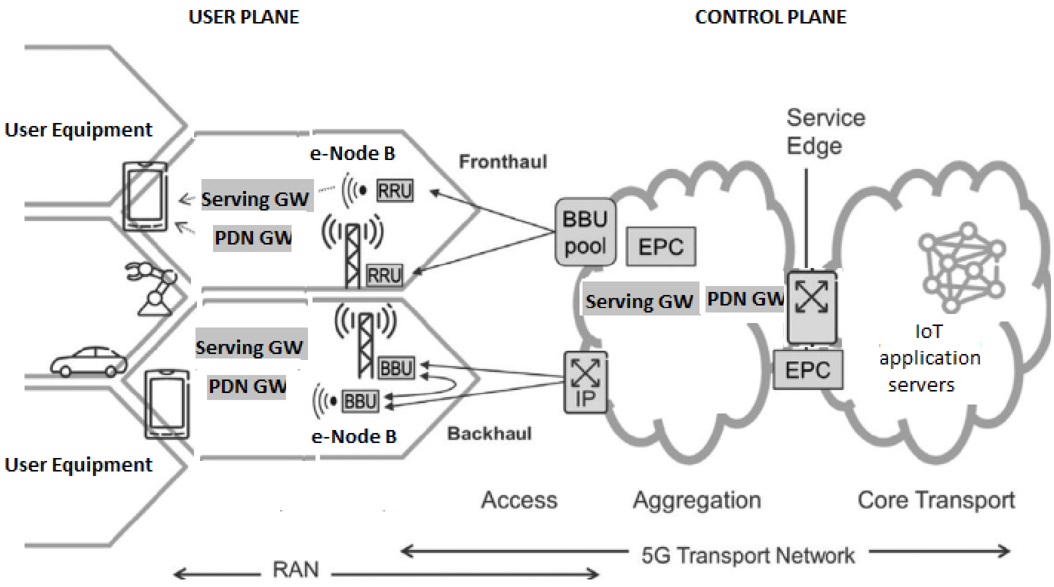


Figure 2.7: 5G reference architecture

Data Network (PDN) gateway, serving as the Control Plane for the network. The former is responsible for routing the incoming and outgoing IP packets and the latter serves as a connection point between the EPC and the external IP networks, called as PDN. The PDN gateway routes packets to and from the PDNs and performs various functions such as IP address / IP prefix allocation or policy control and charging. The Serving gateway is logically connected to the PDN gateway and even though 3GPP describes them independently, in practice they may be combined in a single hardware by network vendors.

Packets in 5G communication are exchanged between the cloud (application servers of Figure 2.7) and the control plane. The control plane is afterwards using the standard TCP, UDP and IP protocols to exchange packets with the user plane, as illustrated in more detail in Figure 2.8. The BBU is used to form the evolved (Evolved Node B (e-NodeB) in Figure 2.7) the main point responsible for the transmission/reception of IP packets to/from the control plane. The BBU also performs packet demodulation as well as amplification to transmit them to the User Equipment (UE), which denotes the end devices used for communication. Connectivity between the user equipment UE and the core network is provided by the E-UTRAN. The E-UTRAN is a collective term for the network and equipment that connects mobile handsets to the public telephone network or the Internet.

The user plane contains the e-NodeB and UE consists of three sub-layers: Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC) and Medium Access Control (MAC), that are illustrated in Figure 2.8. In the same figure we can also observe data exchange through UDP between the UE and the Serving gateway, which serves as a relay for the EPC. The exact encoding and format of the exchanged data is provided in the following part of the section.

We will hereby focus on describing the individual protocol layers that support the communication according to the previous 3GPP standards, as 5G is currently in the process of being standardized. This is accomplished by starting with the highest applicable layer, which here is the IP. 3GPP initially described as a part of release 8 of 3GPP (3G standard) that cellular com-

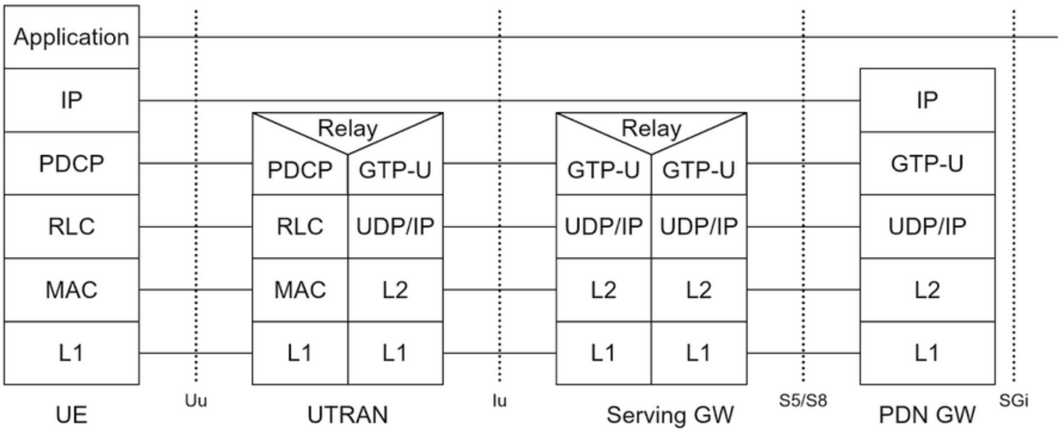


Figure 2.8: Communication overview in the 5G fronthaul.

munication will use IPv4 addressing scheme to identify the individual nodes on the network. However, the expanding use of IP addresses created the need of having an IPv6 addressing scheme to equally increase the number of devices that could be connected to the network.

In the lower layer (layer 2) of the OSI architecture we encounter the Packet Data Convergence Protocol (PDCP) layer (Figure 2.7), which is responsible for header compression and decompression of IP data of layer 3 as well as the transfer of data (user plane or control plane). Specifically, PDCP includes also a mechanism to remove the IP header (Minimum 20 bytes) from Protocol Data Unit PDU, and replace it with a token of 1-4 bytes [78]. The PDCP is also responsible for the Maintenance of Sequence Numbers (SNs), in order to organize and synchronize the number of packets received on the network. The particular steps that PDCP follows in order to transmit the data over a radio interface are described thoroughly in [78].

Another protocol that is found in the same layer is the Radio Link Control (RLC) layer, which is responsible for the segmentation of the IP packets known as (PDUs) into smaller units known as Service Data Units (SDUs), error correction through ARQ, concatenation, segmentation and reassembly of the segmented SDUs to form the PDUs on the receiving end. The RLC also handles the ARQ protocol. Additionally, the RLC layer can offer error-free delivery of data by its mechanism of requesting retransmissions of erroneous RLC PDUs. Specifically, for each incorrectly received PDU, the RLC requests a retransmission. The need for a retransmission is indicated by the RLC entity at the receiving end to its peer RLC entity at the transmitting end by means of status reports.

The MAC layer is responsible for the mapping between logical channels and transport channels. Additionally, on the sending side it is responsible for the PDU construction from multiplexing of MAC SDUs into a Transport Block (TB), which is delivered to the physical layer on transport channels. Likewise, on the receiving side it is responsible for recovering the MAC SDUs by demultiplexing the transmitted PDUs delivered from the physical layer on transport channels. Apart from the multiplexing/demultiplexing mechanism the MAC layer consists of a Hybrid ARQ (HARQ) mechanism, which includes the transmission and retransmission of TBs on the sending side as well as the reception of ACK/NACK on the receiving side. The MAC layer can include up to up to eight HARQ processes in parallel to enable continuous transmission while previous TBs are being decoded. Each of these processes perform a TB blocking send by stops and awaits feedback from receiver. When a NACK or nothing is received, transmitter

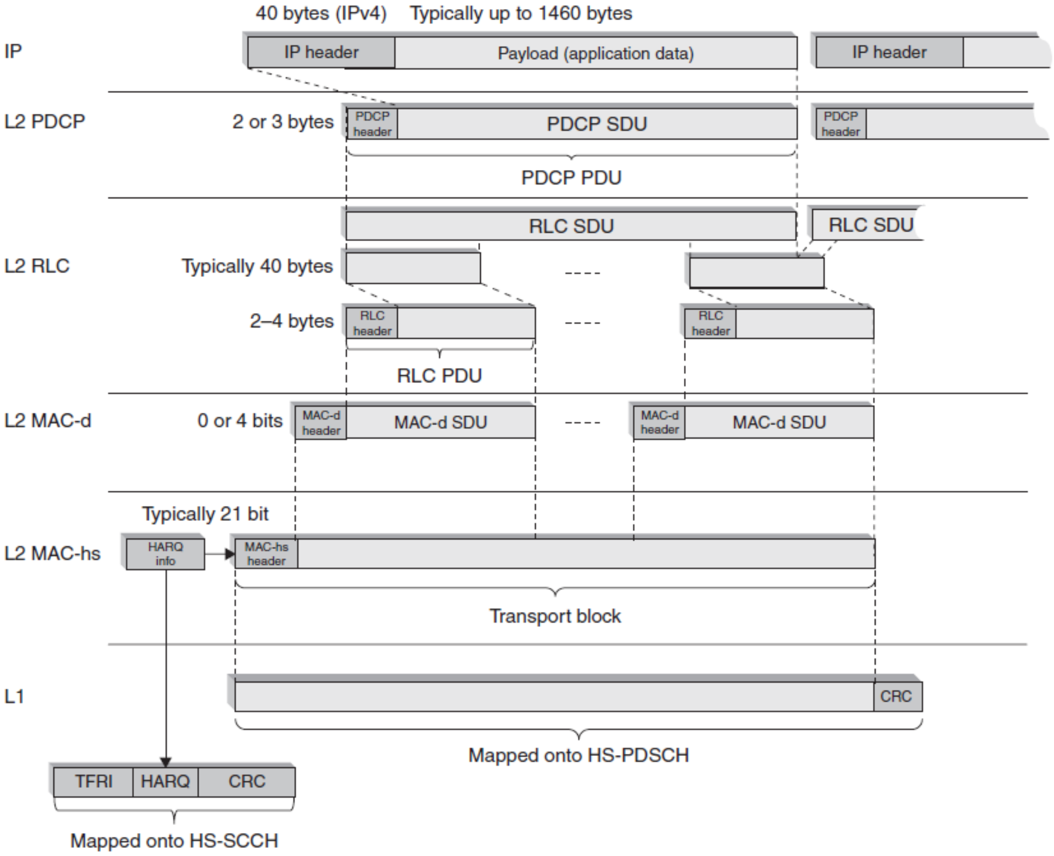


Figure 2.9: 3GPP network layers as introduced in release 8 (3G communication)

retransmits TB. Finally, the MAC layer supports logical Channel prioritization.

Packets in cellular communication are exchanged by the physical layer using radio interfaces, which use modulation techniques such as the cyclic-prefix orthogonal frequency division multiplexing (CP-OFDM) [156]. The physical layer is responsible for operations such as coding, spreading and data modulation, as well as modulation of the radio-frequency carrier.

Apart from the legacy EPC-connected devices in a standalone LTE environment, 5G communication is scheduled to support Next Generation Converged Network (NGCN) connected devices. This network offers flexibility and scalability, and it ensures efficient use of available bandwidth. The underlying reason behind this is the use of Wavelength Division Multiplexing (WDM) [79] in the physical layer that is optimized for high bandwidth in the core. WDM is enabling multiple point-to-point interconnections to the Network Terminal Equipment to create a 'mesh and branch' topology.

### Scenarios for 5G communication

The 5G technology aims in supporting Standalone and Non-Standalone New Radio (NR) operation (with work for both starting in conjunction and running together). Non-standalone NR in this context implies using LTE as control plane anchor. Standalone NR implies full control plane capability for NR. Standalone and Non-Standalone NR operation (with work for both starting in conjunction and running together). The following scenarios are provisioned to be supported in 5G communication:

1. Standalone LTE, EPC connected - legacy
2. Standalone NR, NGCN connected
3. Non-Standalone/"LTE assisted", EPC/NGCN connected
4. Non-Standalone/"NR assisted", EPC/NGCN connected

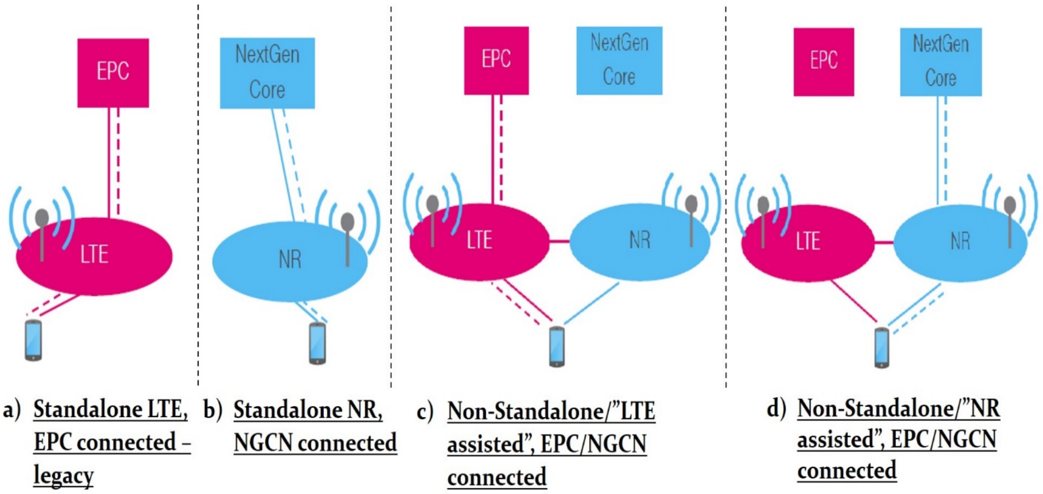

EPC
LTE
NextGen Core
NR
EPC
NextGen Core
LTE
NR
EPC
NextGen Core
LTE
NR
a) Standalone LTE, EPC connected – legacy
b) Standalone NR, NGCN connected
c) Non-Standalone/"LTE assisted", EPC/NGCN connected
d) Non-Standalone/"NR assisted", EPC/NGCN connected

Figure 2.10: Available 5G deployment scenarios.

**These scenarios are illustrated in Figure 2.8 with an explicit focus on the connection to the UE.**

## 2.4.2 Pre-5G radio access technologies in V2X

### NarrowBand IoT

Apart from the 3GPP existing 4G and ongoing 5G standards, 3GPP is also providing standards for cellular Low Power Wide Area Networks (LPWAN) communication mechanisms and functionalities targeted in low-end devices, such as the ones used in the IoT devices. A representative example in this category is Narrowband-IoT (NB-IoT) or often referred to as LTE-M2 [150].

The NB-IoT provides low energy consumption, small volumes of data and transmission over large distances or deep within buildings. It is based on release 13 of 3GPP and operates at even lower bandwidths (180 kHz/channel) and lower data rates (20 kbps) in the licensed LTE spectrum. Mobility is sacrificed in favor of better indoor coverage and support for larger number of devices. NB-IoT is managed by cellular operators with expected costs and regulations on access to this network.

The 3GPP offers three scenarios for LPWAN deployment in NB-IoT (illustrated in Figure 2.8), which are namely In-Band, Guard-Band and Standalone. Specifically, In Band makes use of the same resource block in the LTE carrier of the existing LTE network. Furthermore, guard-band deployment uses the unused blocks within the LTE carrier guard band and standalone deployment utilizes new bandwidth in comparison to existing technologies (e.g. GSM, LTE).

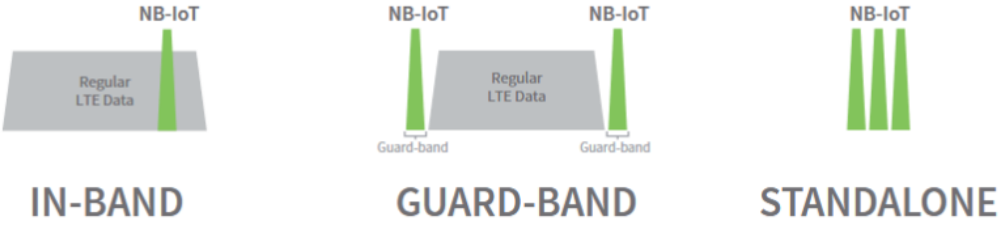


Figure 2.11: Scenarios for LPWAN deployment in NB-IoT.

### LoRaWAN

LoRaWAN [123] is an LPWAN or Low-Power Wide-Area (LPWA) openly defined network protocol that manages communication between gateways and end-devices with the following features: (1) establishing encryption keys for application payloads and network traffic, (2) device to gateway pairing assignments, and (3) channel, power and data rate selection. The devices in LoRaWAN can be of three types:

1. bi-directional end-devices with downlink followed by uplink, as for example sensor end devices,
2. bi-directional end-devices with transmission slots scheduled for downlink, as in the case of actuators, and
3. always-on bi-directional devices, which is intended for low-resource devices to ensure low-latency such as gateways or servers.

A reference view of the LoRaWAN architecture is provided in Figure 2.9. The figure illustrates the communication between the LoRaWAN Server, the gateways as well as the end devices. The gateways are responsible for maintaining radio connectivity as well as may act as transparent bridge on the network. Furthermore, they ensure seamless network upgrade. Additionally, the LoRaWAN Server is responsible for maintaining association with end node, configuring data rates, removing duplicates and the handling security and access control interfaces with applications. Finally, the LoRaWAN end device in the system has a network communication and application encryption key. All packets are transparently sent from gateways to a LoRaWAN server without any local decryption to limit the potential risk of compromised clients and gateways (Figure 2.9).

### LTE for machine type communications

LTE-MTC or LTE-M is an LPWA technology standard based on 3GPP's Release 13 specification. It specifically refers to LTE Cat M1, suitable for the IoT. Even though both NB-IoT and

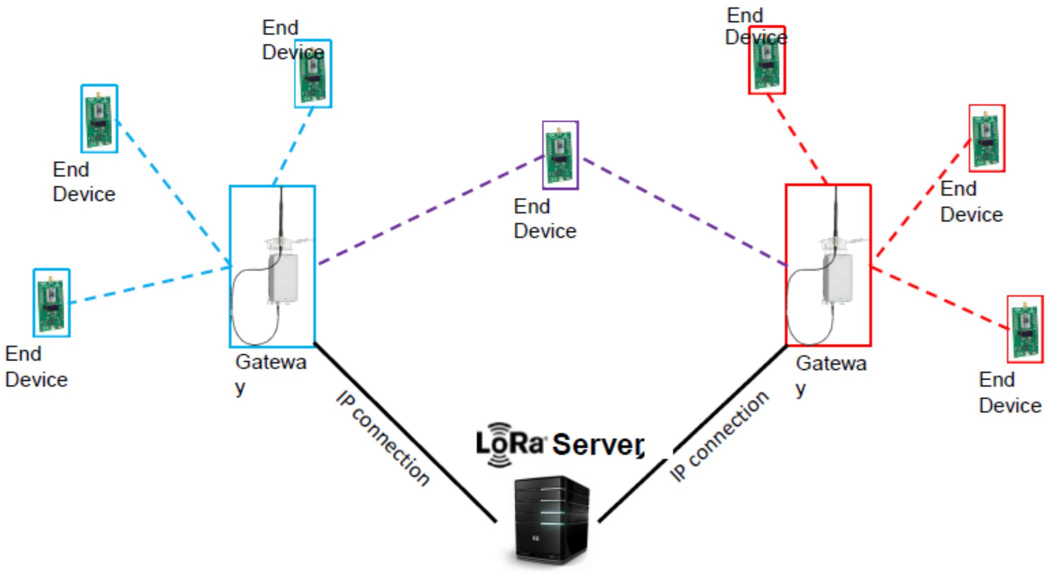


Figure 2.12: Reference architecture in LoRa

LTE-M use LTE and aims in enabling low-power communication in IoT devices, their main differences are in terms of throughput, mobility, power, latency, and cost.

Table 2.1 provides insight on the difference in technology characteristics for NB-IoT and LTE-M. The main differences are found in terms of mobility as well as the technology design. Specifically, NB-IoT does not provide mobility when change from one cell to another and the UE have to perform idle rejoin. This introduces a power penalty for moving devices). NB-IoT is good for sending small and subsequent messages, whereas LTE-M is used to send sequences of messages, such as data streams. Additionally, LTE-M and NB-IoT have also a difference in the power saving mode that the support. In particular, LTE-M supports several power saving modes (e.g: deep-sleep or wake-up only periodically while connected), whereas NB-IoT supports the Extended Discontinuous Reception (eDRX) [112] mode, which allows reduced power consumption for devices that are awake and remain connected.

| Technology characteristics | NB-IoT | LTE-M |
|---|---|---|
| Bandwidth | <250 kbps (Half Duplex) | 384 kbps-1 Mbps (Half or Full Duplex) |
| Coverage | 20 dB | 15 dB |
| Mobility | No | Yes |
| Designed for | Message-Based Communication | IP-Based Communication |
| Power saving | eDRX [112] | Deep-sleep/periodic wake-up |

Table 2.1: NB-IoT and LTE-M differences

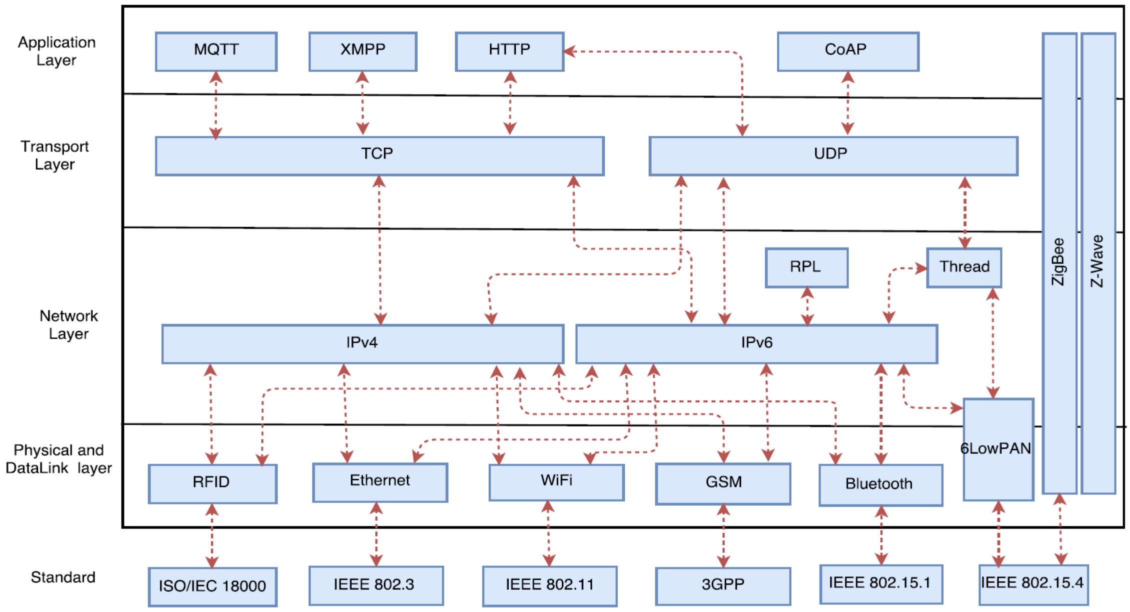


Figure 2.13: IoT Network Stack and Standards

### 2.4.3 IoT Technology

IoT has led to the design and development of new technology tailored to low-power and resource-constraint IoT devices. This section presents an overview of the technologies that enable IoT and, in particular, communication protocols and operating systems that are commonly used in IoT systems.

#### Network Stack

Communication protocols and their relations are usually represented using a network stack. A network stack is represented in layers for easier design and evaluation. Each layer represents different functions and offers different methods of data handling. Traditional network stacks usually consist of five layers [165] or seven layer networks [194]. In this work, we use a four layer model that resembles the traditional five layer network model. It comprises the application layer, transport layer, network layer and data link & physical layer. The data received from the application layer is segmented by the transport layer. These segments are encapsulated into packets by the network layer. The packets are further encapsulated as frames by the link layer and these frames are converted to signals by the physical layer. In order to provide abstraction from the low-level technical details, we combine datalink & physical layer into one layer.

Figure 2.13 presents the network stack describing the protocols commonly used in IoT environments. Moreover, we relate the standards upon which the physical & datalink layer protocols are defined. In the figure, layers are separated by solid lines. Arrows indicate that a given protocol is built on top of another protocol or built on a given standard. The technical properties of protocols, namely data transfer rate, communication range and power consumption, are reported in Table 2.2. In the remainder of the section, we review the protocols used in each layer.

| Protocol | Data Transfer Rate | Range | Power Consumption |
|---|---|---|---|
| **RFID** | 4-640 kb/s | 10 cm-200 m | < 15 mW |
| **Bluetooth** | 1-3 Mb/s | 10 m-100 m | 500-1000 mW |
| **Bluetooth Low Energy** | 1-3 Mb/s | >100 m | 100-500 mW |
| **Ethernet** | 1 Mb/s-100 Gb/s | 1 m-40 km | > 300 mW |
| **WiFi** | 2 Mb/s-1.3 Gb/s | 100 m-30 km | 500 mW-10 W |
| **GSM** | 0.5-1.6 Mb/s | 100 m-35 km | > 700 W |
| **ZigBee** | 300 b/s - 2 Mb/s | 10 m-20 km | 1-1000 mW |
| **ZWave** | 100 kb/s | 30-40 m | 1 mW |
| **6LowPAN** | 300 b/s - 2 Mb/s | 20 m - 20 km | 1-1000 mW |

Table 2.2: Protocol Properties

**Physical layer & data link layer.** Several protocols have been used in the physical layer & data link layer within IoT environments. We classify them based on the network type they support: Local Area Network (LAN), Personal Area Network (PAN) and Wide Area Network (WAN). The PAN protocols commonly used in IoT are Radio-Frequency Identification (RFID) [25], Bluetooth [42], ZigBee [23] and ZWave. RFID [25] is largely used within IoT environments to identify devices [161]. RFID is based on the ISO/IEC 18000 standard and defines the communication between tags and readers. RFID tags are attached to IoT devices for identification whereas RFID readers consist of a two-way radio transmitter-receiver that sends a signal to the tag and read its response. Bluetooth is a short-range wireless technology which was initially standardized by IEEE as IEEE 802.15.1 standard [42], which is used to exchange information over short distances using short-wavelength radio transmissions. Bluetooth divides data into packets and transmits them. The Bluetooth Special Interest Group has released a version of Bluetooth protocol, called the Bluetooth low energy[1] (or Bluetooth Smart), for low energy devices. This version enables low power transmissions and consumes around 0.01-0.50 W in comparison to the classic Bluetooth protocol that consumes around 0.5-1.0 W.

The IEEE 802.15.4 standard [36] is intended for low-rate wireless personal area networks (LRWPAN), which currently can also adapt to high data rates. A protocol based on LRWPAN specification is Zigbee [23]. Although this protocol builds on LRWPAN, it has additional components for the network and application layers. The Zigbee standards offer support to other protocols in the network, transport and application layer. Similar to Zigbee, the Z-Wave[2] protocol also works on low-frequency radio bandwidth. It is proprietary protocol and is not build on any specific standard. It provides the complete network stack from the physical layer to the application layer.

Among LAN protocols, traditional technologies such as Ethernet and WiFi are often used in IoT. Ethernet [35] or IEEE 802.3 protocol transmits data as frames, where each frame has the source and destination addresses. WiFi is based on the standard IEEE 802.11 [27] that allows gateway devices to transmit information using radio-waves over high speed Internet connections. Modern computing devices such as smartphones and tablets typically support WiFi. Typically, the devices connect to the Internet via wireless access points although this may be different for other enhancements such as 802.11p.

For WAN, cellular technologies are often used in IoT environments. The GSM [44] is the most commonly used digital cellular technology primarily used for transmitting data voice and services based on 3GPP specification of GSM. GSM describes the protocols for second

[1] https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy

[2] http://www.z-wave.com/

generation (2G) networks which is used for mobile communication and includes GPRS and EDGE. Applications such as vehicular tracking that require long range communication, are often based on GSM. Another standard supporting higher-speed 3GPP communications is the third-generation UMTS. Recent work has also standardized a fourth-generation of 3GPP communications called Long Term Evolution (LTE), while current work is focused on standardizing fifth-generation to support lower latency and improved resource utilization for IoT devices [140].

**Network layer.** The most commonly used network layer protocols in IoT are IPv4 and IPv6. These protocols are variations of the Internet Protocol (IP) [13], both used to identify devices on the Internet based on unique addresses. IPv4 uses 32-bit addressing schemes whereas IPv6 uses 128 bits addressing scheme. These addressing schemes are used in IoT to identify a group of sensor devices geographically [92]. IPv6 requires a Minimum MTU (Maximum Transmission Unit) size of 1280 bytes, whereas the IEEE 802.15.4 link layer allows a maximum frame size of 127 bytes.

The IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [16] is built on top of the LRWPAN specification. 6LoWPAN uses encapsulation and header compression mechanisms to transmit IPv6 packets over IEEE 802.15.4 networks, thereby creating a mapping between the link layer and the network layer. This protocol aims to support IP for low power IoT devices.

Thread[3] is another network layer protocol specifically designed for home automation, which is based on IPv6 and 6LoWPAN. Thread allows peer-to-peer communication of devices over a local wireless mesh network. Such IP-based mesh networks allow devices to connect and communicate with each other easily. Since it is based on 6LoWPAN, it supports low-power IoT devices.

IoT systems are also reusing the architectures and protocols of Wireless Sensor Networks (WSN), in order to extend them through the addition of web resources. This allows to facilitate software reusability and application development. A particular type of WSN IoT network architecture that is used in IoT is low-power and lossy network (LLN). In such networks, devices and routers have memory and processing constraints. Moreover, routers typically support low data rates and are unstable. RPL [21] is a IPv6 routing protocol for LLN that efficiently routes multipoint-to-point (from devices to a central point), point-to-multipoint (from a central point to devices) and point-to-point (between the devices) traffic. This protocol uses Destination Oriented Directed Acyclic Graphs that enables traffic to move up to the root or down to leaf nodes in the graph.

**Transport layer.** Transmission Control Protocol (TCP) [14] and UDP [146] are widely used protocols in the transport layer for IoT. TCP is connection-oriented and UDP is connection less. This distinction makes TCP more reliable than UDP as TCP guarantees that all packets are delivered. However, it is not scalable for small data transmissions in IoT devices [190]. UDP is more suitable for real-time communication where delay is not tolerated.

**Application layer.** In the traditional Internet protocol stack, the most common protocol in the application layer is HTTP. It can be run over both TCP and UDP. However, HTTP is verbose and complex, and adds a significant parsing overhead. This may not be suitable for constrained devices. Moreover, HTTP inherits the limitations of the protocol at the transport layer on

[3] https://threadgroup.org/

| Operating System | Kernel Size | RAM | ROM | Protocols |
|---|---|---|---|---|
| **Contiki OS** | 3.876 kB | 2-10 kB | 30 kB | IPv4, IPv6, RPL, TCP |
| **TinyOS** | 0.4-1 kB | < 1 kB | < 4 kB | 6LoWPAN |
| **RIOT OS** | 1.5 kB | 1.5 kB | 5 kB | IPv6, UDP, 6LoWPAN, RPL |
| **MynewtOS** | 6 kB | 8 kB | 64 kB | Bluetooth Low Energy, Thread |

Table 2.3: Operating System Properties

which it runs [190]. To overcome these limitations several application layer protocols have been developed. Their goal is to enhance communication between middleware and application layer by minimizing the overhead.

Constrained Application Protocol (CoAP) [32] is one of the most commonly used protocols for IoT devices and is based on the client-server model. It runs over UDP and performs asynchronous message exchanges. CoAP has low header overhead and hence simplifies the process of parsing [164]. HTTP and CoAP can also be used in association with representational state transfer architecture [18] that makes it possible to access the resources of an IoT device through an uniform resource identifier. Another commonly used application layer protocol is Message Queue Telemetry Transport (MQTT) [30], a lightweight messaging protocol on top of the TCP/IP protocol. It is mainly used for communication with remote locations where network bandwidth can be limited. MQTT is based on the publish-subscribe paradigm, where the sender (i.e., the publisher) transfers the message to a broker who distributes the messages to the interested clients (i.e., subscribers). One of the commonly used MQTT broker is Mosquitto [8]. Since MQTT runs on top of TCP, it may not be suitable for applications that require real-time processing. Another application layer protocol commonly used in IoT is Extensible Messaging and Presence Protocol (XMPP) [29]. This protocol is used for streaming extensible markup language elements and real-time exchange of structured data. The main features of XMPP are extensible messaging and presence protocol[4]. Extensible messaging implies an open standard application used for sending real time messages between clients. The presence protocol implies that the protocol indicates the server about the status of the client whether it is online, offline or busy.

### Operating Systems

An operating system (OS) manages resources, schedules process and executes application services. In addition to this, it also implements the network stack. In this section, we present an overview of the main operating systems that have been developed for IoT environments. These OSs usually require less memory and are suitable for low-power and memory-constraint devices. The properties of these OSs in terms of kernel size, memory and suitable protocols are reported in Table 2.3.

**Contiki OS [4]**: Contiki OS is written in C programming language with a kernel size of 3.876Kb. Contiki OS provides support for IPv4 and IPv6 both implemented using uIP. uIP is a TCP/IP protocol stack typically used for micro-controllers. The IPv6 stack contains the 6LoWPAN header compression. Contiki OS also implements another protocol stack called Rime. Rime is a lightweight protocol stack for low power wireless networks that aims for reliable transmission. The advantage of Rime is that it provides support to the applications to implement protocols that are not already present in the Rime network stack. Contiki OS also

[4] https://xmpp.org/about/

provides support for RPL which is implemented as ContikiRPL [174].

**TinyOS [118]**: TinyOS is written in nesC [88] and has interfaces for sensing, actuation, storage and routing. The kernel size is around 0.4-1 kB and is smaller compared to Contiki OS. TinyOS implements a network stack based on 6LoWPAN. In addition, TinyOS provides mechanisms for data collection and dissemination[5]. The mechanism for data collection aims to collect data from the network and transmit it to a central node such as a base station. The dissemination mechanism aims to deliver data to every node in the network in a reliable way to prevent packet loss. TinyOS also provides a mechanism for reliable routing and transmission of data called TYMO[6]. TYMO is a varient of DYMO [15] (dynamic mobile ad hoc network on-demand) routing protocol used for point-point routing in mobile ad hoc networks.

**RIOT OS [55]**: RIOT OS can run on 8-bit, 16-bit and 32-bit processors. It supports the use of programming languages such as C and C++ enabling the use of built-in libraries. RIOT OS provides support for IPv6, 6LoWPAN, RPL, UDP and CoAP. Some of the works on RIOT OS also implements Named Data Networking protocol stack [162]. Named Data Networking focuses on a data centric communication model rather than the commonly used host centric approach.

**MynewtOS[7]**: MynewtOS is a real-time operating system for IoT written in C programming language. The size of the OS kernel is 6 kB. MynewtOS supports the Bluetooth Low Energy protocol at the physical & datalink layer and the Thread protocol at network layer along with basic IP support. This OS also provide a remote monitoring tool for the configuration, upgrade and management of IoT devices[8].

the availability of multiple WAN and LAN connections as well as the dynamicity of the V2X environment increase the possibility of architectural vulnerabilities as well as security threats, To detect the threats as well as to provide mechanisms of protection against them we need to employ security mechanisms for the V2X environment.

## 2.5 Security

The global technology offered through V2X raised many challenges related to security. The purpose of this section is to explore some of the identified security threats and solutions to connected cars. The vehicle to cloud, vehicle to infrastructure, vehicle to pedestrian, and vehicle to vehicle threats and solutions are explored as means of documenting the existing research on security, allowing a greater understanding of current issues within the field and the role of society to address those areas of concern.

The underlying goal and purpose associated with the implementation of new technologies is noble. In the case of connected cars, the goal is to provide increased safety on the road, allowing for decreased accidents, decreased infrastructure cost, decreased delays in traffic, and a general efficacy increase to the transportation process [89]. Unfortunately, as with many technologies, the potential for exploitation of those technologies is great[89]. The use of DSRC allows for the spontaneous communication between the vehicle, other vehicles, other humans, and other communication devices without the ability of the user to effectively limit the data being sent[89]. As a result, one of the largest security areas of concern is that of privacy; while

[5] http://tinyos.stanford.edu/tinyos-wiki/index.php/Network_Protocols

[6] http://tinyos.stanford.edu/tinyos-wiki/index.php/Tymo

[7] https://mynewt.apache.org/

[8] https://cwiki.apache.org/confluence/display/MYNEWT/Apache+Mynewt+Project

machines are able to communicate with one another, the network can be tapped into by others without authorization, allowing for the collection of data on the habits of a specific individual, ranging from anything from their driving practices to their routes, most frequent stops, and general habits[89][148]. Developers of the technology are aware of the adverse uses potential of it. However, the technology has continued to move forward, being viewed as a necessary risk, due to the ability to increase the safety of transportation as a whole[148]. Researchers have looked into different security architectures for implementation within the vehicles, including Public Key Infrastructure based solutions that require the presence of a particular certification for data access,[89].

Further challenges and security risks arise when one explores the potential to manipulate data going to or from the vehicle through the modification of a single sensor, causing a vehicle to transmit false information or to result in the reception of false information from another vehicle[124]. This has the potential, if used improperly, to allow for increased congestion, traffic manipulation, and even can lead to the potential of major or minor accidents [124]. While the implementation of RFID technologies has been recommended as means of working to decrease the risk associated with this concern, there are other potential problems, including not just the electronic manipulation of sensor data, but the potential for the sensor to fail mechanically as well due to the lack of energy constraints related to sensors in vehicles, which could result in a sensor overload. Alternatives are considered, including the use of Bluetooth and ZigBee technologies, allowing the operation of industrial, scientific, and medical (ISM) frequency and the operation on a radio spectrum, decreasing the likelihood of interruption [124].

The group, Network on Wheels is a Germanic research project completed in tandem with the different manufacturers, suppliers, research institutes, and universities who are supported by the German government [81]. The goal of this organization is to work to identify the different security threats that could occur within the connected vehicle, allowing for the creation of a vehicular communication system that will utilize ad hoc networks principles with the combination of LAN technologies in order to provide “infotainment" and security applications within the context of the vehicular environment [81]. While researchers have not yet identified the measure of success of this organization, the presence of the organization indicates that the companies who are working on implementing the technologies are aware of the potential security threats associated with the use of the technology and, even if such solutions are not immediately present at this time, steps are being taken to work to reduce the amount of vulnerabilities in vehicles in order to create a network that allows for server-oriented communications using a 3G or 5G network that requires a subscription for data transmission, storage, and used as a means of additionally layering security options [40][187][81].

However, the connected cars just like the other devices connected to the Internet can be a potential target for hackers. Actually, authors in [185] survey the attacks within the vehicle and some protection methods. As the connected cars are target for hackers via the communication with Internet, the in-vehicle and inter-vehicle communication can also be a target for hackers. Authors in [105][113] introduce the security and attacks toward such vehicles, also Mahmoud H. et al. [77] they made a state of the art research regarding the different attacks against connected cars and research studies that have been done to avoid such threats, they look from security perspective to the in-vehicular network arriving to the cloud platforms, giving an example by the Ericsson’s Connected Vehicle Cloud (CVC). In addition, the authors in [111] they survey the integrity and the different cryptographic systems used for authentication in vehicular communication related to V2X technology, also they extend a description about the role of the vehicular public-key infrastructure in the credential and identity management for

VC. Also, in [114], it is proven that they can attack the vehicle by accessing the in-vehicle network. In addition, some research projects focused on securing the connected cars and their communication with the cloud [10][6]. The 5G V2X approach will provide high availability and low latency, but should also provide a secured environment to the deployment of such technology. The 5G Infrastructure Public Private Partnership (5GPPP) [34] highlights the challenges facing the future of connected cars using the 5G radio technology. Based on those challenges, they define the security requirements as well as the research studies that should be taken into consideration (Identity Management, Privacy, Update the cryptographic algorithms, etc.) to avoid those issues [2].

### 2.5.1 Attacks on connected cars

#### Scope

As defined in the ITS architecture 2.14, the three main communications domain are the in-vehicle domain, V2X domain and the infrastructure domain.

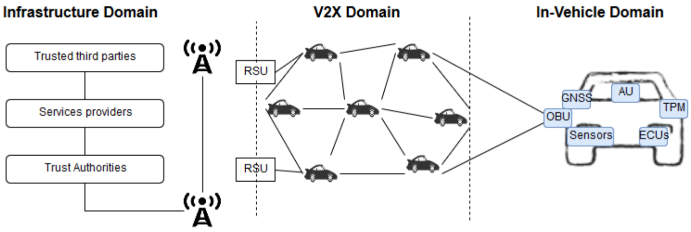


Figure 2.14: ITS architecture. RSU, road side unit; OBU, on-board unit; AU, application unit; ECU, electronic control unit; TPM, trusted platform module.

The attacks on these domains are different and well listed in table 2.4, the ones related to the In-vehicle domain are described in D1.1 and the ones related to the infrastructure domain are described in D3.1. Connected cars are the future of vehicular transportation, not only from the convenience point of view but also from a personal safety aspect. Having your car connected to the network and through that indirectly to all other vehicles, items of infrastructure, pedestrians etc. offers a completely new way of securing the driver and decreasing the amount of traffic accidents. But increasing the connectivity of the car simultaneously increases the risk of it being attacked through the network. Without proper security precautions not only specific cars, but the whole network, is under threat and could be compromised by a skilled attacker. There are numerous examples of connected cars being remotely controlled, either by so called CHTs (Car Hacking Tools) or fully remotely through the cellular network, Bluetooth or Wi-Fi connected to it. This document will focus more on the fully remote access through connectivity, it being the focus of the project in general, but the CHT will also be presented as it also poses a serious threat to the end user and should be taken into consideration. The fact is that in the future every car will be a "mobile server" that could be accessed roughly in the same way that modern servers nowadays can be attacked. This poses a serious threat to the fundamental idea of creating a network of connected cars, the idea of someone being able to access your car's functions, i.e. breaks, engine, infotainment system, in worse case scenario while at high speeds, it could lead to grave consequences. In Section 3.5 of D1.1: Specification of In-car Software Architecture for Car2X Applications, we present some of the known attacks against connected cars carried out to date, what the consequences could have been and what actions were taken

| Types of attacks on V2X | Examples |
|---|---|
| Confidentiality | Eavesdropping, information gathering, bogus information, sharing, traffic analysis, location spoofing |
| Privacy | Location tracking, identity disclosure |
| Integrity and data trust | Message fabrication/Suppression, information forgery, masquerade, replay, delation, man in the middle attack. |
| Authenticity and identification | Sybil attack, impersonation, masquerading, replay attack, GPS spoofing, tunneling, key/certificate replication, message modification/alteration, message tampering |
| Availability | Denial of service, jamming, broadcast tampering, spamming, black hole attack |

Table 2.4: Attacks on V2X

to fix the problems that allowed the attackers to access the system as well as some specific examples on traffic infrastructure potentially affecting the connected car network being hacked, meaning further restrictions to the communication defined within the APPSTACLE project. Taking the possibility of attack in regard when defining V2X communication is crucial, when thinking about threats. If an attacker is able to get into the car's systems, through which the V2X communication also is conducted, it offers him/her the possibility to tamper with that as well, which could, in the worse case scenario, tamper with the whole infrastructure of the connected car network. Corruption of the systems upon which this network and communication is built could prove fatal in many such regards. The purpose is to raise awareness regarding system architecture and security planning as well as present a state of the art survey of some attacks and what to especially think about when designing a connected car system in addition to some future directions and plausible threats in the future based on the practical examples presented.

### Attacks on V2X

#### Consequences to V2X communication

The attack surfaces critical to all kind of V2X communication are, of course, such that they, as functional, provide some kind of compatibility what comes to V2X. Effectively, almost every system in the car can prove critical in this aspect, given that the systems communicate with one another, but some have direct consequences, i.e. They can directly affect the V2X communication in ways that would compromise the safety of the driver. These include, but are not excluded to;

- Attacks on infrastructure, giving false information to the vehicles reading it (road, signs etc.).

- Machine vision, falsifying the perception of other components in the network.

- In-vehicle devices, gaining access to control systems that control critical functions for either communication or driver safety.

- LIDAR and RADAR, falsifying the function of the LIDAR system, leading to false information.

- Odometric sensors, map poisoning and GPS, falsifying the positioning.

- Other vehicles, interfering with V2V communication.

- Attacks on the CAN bus.

In addition to these, all of the other attack surfaces described could, in one way or another, have indirect effects on the communication. These threats are to be taken very seriously, given that, with the security standards many cars yet today exercise, many of these targets aren't too hard to access by an experienced attacker and that the consequences from even the slightest attacks could be fatal. Petit and Shladover further identify attacks on the GNSS (Global Navigation Satellite Systems) as the most severe, due to the feasibility and impact of these attacks. They also mention some standardization work done by ETSI for defining mitigation techniques for cyber attacks affecting V2X communication.

### 2.5.2 Network Security

As the usage of V2X automotive connectivity grows and becomes common in many architectures, a lot of issues are brought to the surface [82]. Addressing this problem becomes a reali challenge if we consider the different forms of V2X automotive connectivity that are found in today's systems, such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) short-range communication, as well as vehicle-to-Internet communication via an embedded modem that can communicate over cellular networks as described in Section 2.4 for wider-range communications.

The attacks in V2X systems that were presented in Section 2.5 can be divided in the following categories:

- Attacks against end user's confidential information
- Attacks on network accessibility or availability
- Authorized messages that give false warnings

It is well known fact that Serving Data Network (SDN) in 5G communication (Section 2.4) can bring a lot of benefits because of decoupling the control from the data plane 2.15. But still there is a vulnerable relation between SDN and DDoS attacks. SDN itself may be a target of Denial of Service DDoS attacks. Network capabilities, such as global view of the network, dynamic updating of forwarding rules and so on, can facilitate DDoS attacks detection, but the separation of the control plane from the data plane leads to emerging new types of attacks on the network availability [110]. For instance, an attacker can use the characteristics of SDN to launch DDoS attacks against the control, infrastructure and application layers of an V2X system.

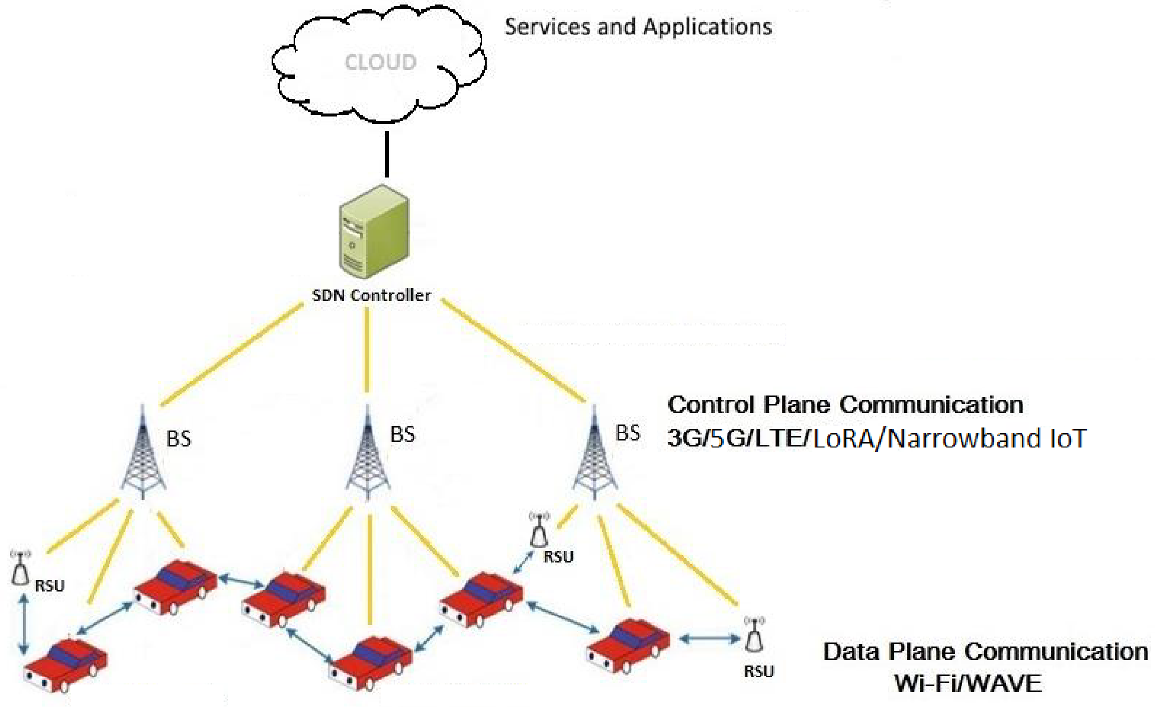

CLOUD
Services and Applications
SDN Controller
BS
BS
BS
Control Plane Communication
3G/5G/LTE/LoRA/Narrowband IoT
RSU
RSU
RSU
Data Plane Communication
Wi-Fi/WAVE

Figure 2.15: Communication layers of C-V2X architecture

### Threat analysis

Adversaries target vulnerable vehicles or RSUs in order to gain access to to the rest of the V2X network and the different V2X functionalities. To this end, an attack may target the V2X network, the individual nodes (e.g. vehicles, RSUs) or the exchanged data between the nodes. Adversaries can even navigate themselves to the core in-vehicle network with methods that are discussed in D1.1: Specification of In-car Software Architecture for Car2X Applications. Hereby, we discuss the parameters and means that characterize an adversary in a V2X environment:

1. Membership: An adversary can be either an insider, meaning an authenticated member of the network or an outsider, meaning an intruder. In the insider can we consider entities that have the internal network key.

2. Motivation: The adversary may have malicious means, in order to interrupt and harm the system's operation or can be rational, in a way of seeking personal profit from an attack on the system
3. Method: An attack on a V2X system can be active by generating packets or signals as well as passive where the adversary is only eavesdropping
4. Scope: The adversary may perform an attack by controlling a local part of the network (certain vehicles or RSUs) as well as an extended part of the network, in which case the attack may have a more sever impact as he/she controls several entities that are scattered across the network

The available network security mechanisms are divided in the following categories:

- Physical Security [183]: Requires of tampering protection mechanisms and tamper proof devices
- Digital signatures and Certificates: Fast implementation in comparison to other techniques, but often the performance requirements for the key computation can only be met when dedicated hardware is provided [128]. Additionally, an V2X system entity (e.g. vehicle, RSU) can still send valid compromised messages when it is infected.
- Firewall Gateway [184]: Signature-based detection, which requires frequent rule update
- Honeypot [178] : Vast effort and cost in providing a simulated though realistic functionality of the entire V2X system as a vulnerable target for an attacker
- Software security:
  - Intrusion detection systems (IDS) [151]
  - Intrusion prevention systems (IPS) [179]

From the presented mechanisms IDS and IPS mechanisms are considered the most prominent [9], since they use advanced detection logic to also allow detection and protection against zero-day attacks. Attacks are presented in the form of malicious and unauthorized activity. Detection is achieved by network monitoring the V2X activity and detection of suspicious and anomalous behavior. The detection is usually based on deep packet inspection and analysis as well as analysis of the data exchange flow. In this way it allows protection against zero-day attacks. An IDS can operate in the level of the host application (Host-based IDS) as well as the communication network (Network-based IDS). Both categories require an initial configuration phase, where they learn the normal system behavior, in order to create a baseline in order to detect anomalies

### Intrusion Detection Systems (IDS)

The main characteristic of V2X communication is its collaborative Internet of Things (IoT) nature. The variety of entities that synthesize the Internet of Vehicle (IoV) environment (e.g.

[9] https://www.lbmcinformationsecurity.com/blog/ids-and-ips-101-how-each-system-works-and-why-you-need-them

vehicles, RSU's etc) use different types of communication protocols and connections, which cannot be monitored by an IDS that resides within a single vehicle. Such IDS is called standalone and is deployed to monitor the behavior of in-vehicle networks (see APPSTACLE deliverable D1.1). To be able of providing end-to-end network security in an IoV environment two further categories of IDS systems should be defined:

- Cooperative IDS: Cooperative IDS are characterized by cooperation between neighboring nodes to detect the intrusion, if detection is unaccomplished individually. This cooperation is realized by exchanging information or alerts in a distributed setting. The major problem for this type of IDS is that they cause degradation of network performance by traffic exchanged between IDS agents. Cooperation between the IDS based on techniques different as mobile agents and neural networks.
- Hierarchical IDS: To remedy the lack of cooperation between different IDS proposed for ad hoc networks, an alternative method has been proposed for intrusion detection. This approach is based on the division of the network into a set of groups (clusters) each having one cluster Head determined by a cooperative algorithm between nodes. Hierarchical intrusion detection Systems try to reduce the cooperation between nodes by dividing the network into clusters. In this case the cooperation is carried out between the elected cluster Head and each of members of the same cluster, as is the case in ad hoc multilayer networks. Therefore to however is

**Signature-based**: REST-Net [173] is an Intrusion Detection System that enables ADAS to detect fake Messages by monitoring and analyzing Beacon data. It uses patterns (or rules) to define invalid actions of users and thereby detect an adversary.
**Behavior-based**: In [151] the authors propose a Misbehavior Detection System (MDS) that relies not only on the protocol-specific actions of nodes but also on the data these nodes provide. The MDS is a behavior-based IDS that operates on each node of a VANET network and compares its behavior to the average behavior of the other nodes (including the node running the MDS). This is realized by measuring the entropy, a "measure of how much coincidence a given data-set contains. The more coincidence it comprises, the higher the entropy it contains." MDS is particularly effective in detecting DDoS and frame injection attacks. Similar work that is also based on entropy detection DoS and frame injection attacks is described in [131].

IDS approaches in literature also considers active detection requiring a collaborative IDS scheme by all the neighboring nodes [91]. In such a scheme they periodically broadcast beacon packets, to update all the positions and identities, in order to measure the similarities between them. Even though the detection rate on Sybil attacks for such approach is good, it is active i.e. involves the periodic transmission of messages and requires the constant update of neighbor vehicles. Additionally, if the attack duration is shorter than a cetain detection threshold it may not be detected. Similar work in this direction has been also focusing in the detection of impersonation attacks [147]. The detection is based in comparing received signal strength with the position of a node.

A relatively new aspect that is considered in modern day vehicles is autonomous driving. The authors in [48] demonstrate the vulnerabilities and traditional threats in vehicles that are connected in a V2X setting, such as viruses, Denial of Service attacks and wormhole attacks. To address this problem they propose a solution that is based on Artificial Neural Network (ANN) and specifically a supervised learning of neural network architecture. Even though

the detection rate is high with a low alarm rate, the is can also isolate malicious self- driving vehicles, the IDS is not tested in more sophisticated attacks, such as [97].
**Hybrid**: Sedjelmaci et al. in [158] demonstrate AECFV, a Hybrid IDS that combines a rules-based decision technique with a behavior-based Support Vector Machine (SVM) detection technique to detect anomalies and zero-day attacks.

| Attack class | Description | Communication types | Associated IDS categories |
|---|---|---|---|
| Denial Of Service [143] | Network flooding with broadcast messages | V2V/V2I | Signature-based: [125]<br>Behavior-based: [131]<br>Hybrid: [48] |
| Frame injection [97] | False information or vehicle indications | V2V | Behavior-based: [151], [131] |
| Masquerade / impersonation attack [66] | Spoofing or frame replay | V2V | Behavior-based: [147] |
| Frames dropping (e.g. blackhole attack) [175] | Preventing legitimate frames from being received by the V2X nodes | V2V | Hybrid: [48] |
| Sybil Attack [93] | Control multiple vehicles to exchange wrong V2X data | V2V | Behavior-based: [91] |

Table 2.5: Existing attacks and their detection by existing IDS

### Discussion

Additional IDS for V2X communications can be derived by extending already existing IDS for Mobile Ad-hoc NETworks (MANET) systems, such as the MobIDS (Mobile IDS) [109]. The detection process of MobIDS is based on monitoring the flow of packets of an individual node. Another well-proven IDS that can be possibly adapted for the domain of V2X communications is the Snort IDS [56]. An initial step towards this direction has been done by the authors in [129], where the IDS traffic is compared to rules already established in a Wireless Personal Area Network (WPAN).

Through our work we have identified the following major design challenges of V2X IDS, based on which the research directions on network security for V2X should be identified:

**Dynamic architecture**: Nodes can be added or removed at any moment in a V2X system. Such architectural scheme allows node mobility, however the frequent changes in the network topology make it difficult to detect attacks
**Remote connections**: remote wireless hotspots or cellular connections cannot be controlled, which brings potential attack vectors. For instance, an adversary can set a fake AP, which can be accessed by legitimate users as well as adversaries. Through this AP the adversaries can

cause data leakage towards their system. Through the existing IDS the legitimate nodes cannot be distinguished from the malicious

**Detection only on wireless protocols**: To the best of our knowledge there is currently no IDS for cellular V2X communications, as it very hard to monitor the messages that are exchanged between each Base Station and the vehicles/RSU's (Figure 2.15) due the extended communication range and the variety of communication mechanisms or protocols that are used (e.g. 3G, LTE, 5G, LoRA, Narrowband IoT).

**Common benchmark**: Regarding the NIDSs presented above, it would be interesting to have a way to assess them using a common benchmark. The majority of results in the literature derive by simulated environments, which may not be applicable in real V2X scenarios.

**Incident response**: A significant open problem to intrusion detection for V2X is the response to the detected events or alarms. A method that is effective against DoS attacks is the use of IPS systems [179], in order to block non-legitimate traffic. However, such method may not effective against more sophisticated attacks and therefore its use has the underlying risk of blocking legitimate traffic in case of false-positives detection.

### 2.5.3 Access control in IoT

While offering attractive opportunities and new business models, IoT opens several security and privacy issues. In this section, we focus on one of the main security challenges in IoT, namely securing IoT-related data while ensuring the functioning of the system. The protection of sensitive data and resources is typically addressed through access control. In particular, access control aims to restrict access to data and resources to authorized entities. Firstly, we describe cloud-based IoT and the IoT architecture types. Later, we explain the access control models and a reference architecture for discussing authorization in IoT.

#### Cloud-based IoT

The last years have seen a growing interest in the application of cloud computing as a middleware for IoT systems. With the increase in IoT devices and the recent trends in big data, cloud-based IoT can offer an efficient solution for data storage and computation. In this section, we provide an overview of cloud computing along with its service models and discuss its application to IoT systems.

#### Cloud Computing

According to National Institute of Standards and Technology [133], cloud computing enables an on-demand network, storage, application and other services without any management effort. A number of service models have been proposed for cloud computing [70]:

**Software as a Service (SaaS):** Cloud providers offer application software as a service to end-users. The application software is installed and operated by the cloud provider in the cloud and end-users access the application software from web clients.

**Platform as a Service (PaaS):** Cloud providers offer a development environment as a service to end-users. Users can develop and modify cloud-based applications while the underlaying hardware and software are managed by the cloud provider.

**Infrastructure as a Service (IaaS):** Cloud providers offer computing infrastructure, e.g. virtual machines and other resources, as a service to end-users. End-users can access the infrastructure through APIs and create their own virtual cloud without being required to maintain the underlying physical infrastructure.

Apart from these models, several other service models such as security as a service [177], trust as a service [139] and data as a service [172] have been proposed to leverage the benefits of cloud computing. The growing interest of cloud computing in IoT has also led the definition of novel service models specific to IoT environments. These service models focus on the integration of cloud computing and IoT to provide new applications and services. Some of the cloud-based IoT service models are Sensing as a Service ($S^2$aaS) [144, 163], Sensing and Actuation as a Service (SAaaS) [72] and Identity and Policy Management as a Service (IPMaaS) [193].

**Sensing as a Service ($S^2$aaS):** Cloud providers offer sensor resources as a service to application nodes. Application nodes use these resources to provide services to end-users.

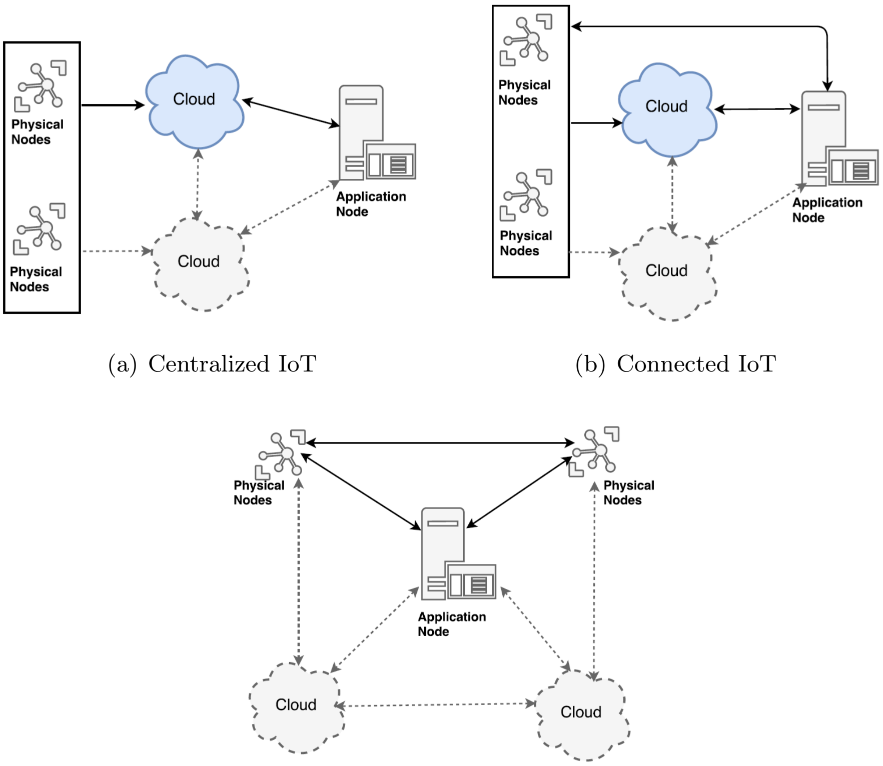
**Sensing and Actuation as a Service (SAaaS):** Cloud providers offer abstraction of sensor and actuator resources as a service to application nodes. The abstraction is achieved through a virtualization of physical nodes which makes an application node interact directly with the virtual counterpart of a physical node.

**Identity and Policy Management as a Service (IPMaaS):** Cloud providers offer management of user identity and security policies as a service to the owners of physical node. The owners of physical nodes can customize security policies for the protection of their resources.

Some applications of IoT such as vehicular safety do not tolerate any delay in sensing an event and decision making. Therefore, the latency of communication between IoT devices and middleware has to be minimal. This need has led to the definition of new computing models. *Fog computing* [167] has been proposed to bridge the gap between the physical layer and middleware. The idea underlying fog computing is to move computation services to the edge of the network such as in gateways, which are referred to as fog nodes [167]. This way, decision making can be performed in the nearest fog edge, i.e. in the network layer. *Mist computing* [122] is another computing model proposed to decrease the communication latency. In such a model, less computationally intensive tasks are pushed to the edge of the network and are performed by physical nodes. This way, the complexity of data processing is partially managed by the devices themselves. It is worth noting that cloud, fog and mist computing are not mutually exclusive. For instance, mist computing can be combined with cloud computing and fog computing where the data processed by physical nodes can be passed on to fog nodes or to the cloud for further processing.

### Cloud-based IoT Architecture Types

Many researchers [60, 85, 149] have recognized the benefits of exploiting cloud computing for IoT systems, as it could offer solutions for connectivity and interoperability between IoT applications. This has led to the definition and design of several IoT architectures that employ cloud computing as a middleware. Although these architectures can vary based on the application domain, they can be classified into three main types based on the connectivity between physical nodes, middleware and application nodes [61, 153]. An overview of the types of IoT architecture is presented in Figure 2.16.



(a) Centralized IoT

(b) Connected IoT

(c) Distributed IoT

Figure 2.16: Types of IoT Architectures.

**Centralized** Physical and application nodes are required to communicate with each other through a cloud. This means that if an application node wants to retrieve resources from a physical node, a connection has to be established using the interfaces provided by the cloud. Thus, physical nodes are usually passive entities that send data to the cloud without processing.

**Connected** Physical nodes posses the ability to process information and forward it to the cloud. In addition, physical nodes can provide this information directly to application nodes. This means that application nodes can directly connect with physical nodes through the interfaces they provide.

**Distributed** Every node can communicate with each other. This means that every node has the potentiality to process information and provide services. Note that, differently from other IoT architectures, the distributed IoT architecture does not require a cloud, although its use can facilitate the communication between nodes and service provisioning.

It is worth noting that a middleware can consists of a federation of clouds that can interact with each other. These clouds may be owned by different service providers. In such scenarios, a cloud may offer and share information to other clouds [154]. These federations of clouds ensure a unified middleware for physical and application nodes. In all the above architectures, federation of clouds can participate and can connect physical and application nodes in different trust domains.

In this section, we investigate the requirements that access control systems for IoT environments should meet. Moreover, we will investigate the design principles underlying existing access control systems for IoT environments. The design of an access control system typically comprises three main concepts [155]: *policy*, which defines the (high-level) rules according to which access control is regulated; *model*, which provides a formal representation of the access control policy and its evaluation; *mechanism*, which defines the low level implementation of the control imposed by the policy as formalized in the model. In this work, we mainly study existing access control systems for IoT environments with respect to the access control model they employ and the architecture underlying their mechanism. The last aspect is particularly interesting as it makes it possible to assess how the control is spread across the IoT architecture. To this end, we present an overview of the most popular access control models and introduce a reference architecture for access control systems.

### Access Control in IoT

While offering attractive opportunities and new business models, IoT opens several security and privacy issues. In this work, we focus on one of the main security challenges in IoT, namely securing IoT-related data while ensuring the functioning of the system. The protection of sensitive data and resources is typically addressed through access control. In particular, access control aims to restrict access to data and resources to authorized entities.

In this section, we investigate the requirements that access control systems for IoT environments should meet. Moreover, we will investigate the design principles underlying existing access control systems for IoT environments. The design of an access control system typically comprises three main concepts [155]: *policy*, which defines the (high-level) rules according to which access control is regulated; *model*, which provides a formal representation of the access control policy and its evaluation; *mechanism*, which defines the low level implementation of the control imposed by the policy as formalized in the model. In this work, we mainly study existing access control systems for IoT environments with respect to the access control model they employ and the architecture underlying their mechanism. The last aspect is particularly interesting as it makes it possible to assess how the control is spread across the IoT architecture. To this end, we present an overview of the most popular access control models and introduce a reference architecture for access control systems.

The identified requirements, access control models and reference architecture will provide the baseline for the comparison of existing access control systems that have been proposed for IoT environments and, in particular, for cloud-based IoT environments.

### Access Control Models

Several access control models have been proposed in the literature. In this section, we present an overview of the most popular access control models.

**Discretionary Access Control (DAC)** [90]: DAC is based on the notions of ownership and delegation. Intuitively, an entity (called owner) is allowed to explicitly specify the access rights a subject can have over the objects under its control. Moreover, subjects can delegate their rights to other subjects on their discretion. Although many variations have been proposed, DAC is generally considered an identity-based access control model where access rights are granted based on the identity of subjects. DAC policies are usually represented using an access matrix, where subjects are represented on the rows, objects on the columns and each entry provides the access rights that the subject has on the object. Access matrices are often sparse and, thus, not efficient for policy representation. To this end, various implementations of access matrix have been proposed [155]: authorization table, access control list (ACL) and capability list. An authorization table only stores the non-empty entries of the access matrix. In an ACL, each object is associated with the list of subjects and actions that they can perform on the object. Capability lists are similar to ACLs, but in this case access rights are stored for subjects. Specifically, each subject is associated with the list of objects and actions that the subject can perform on a given object (so called capabilities).

**Mandatory Access Control (MACo)** [115]: Differently from DAC, access rights with MACo are based on a set of systems rules defined by a centralized authority rather than at the discretion of the object's owner. These rules are typically defined based on security labels associated to subjects and objects. Thus, similarly to DAC, MACo is considered an identity-based access control model.

**Role-based Access Control (RBAC)** [157]: RBAC relies on the notion of role to simplify the specification and management of access rights within an organization. A role comprises the set of permissions needed to carry out a certain job function within an organization. Users are assigned to roles and inherit the permissions assigned to the roles they have.

Roles are often organized in a role hierarchy based on the organization structure. Intuitively, a role hierarchy defines the inheritance of permissions between roles.

**Attribute-based Access Control (ABAC)** [102, 189]: ABAC is a general-purpose access control model in which access rights are constrained with respect to the attributes of entities (subjects and objects), actions and the environment. The use of attributes makes ABAC extremely flexible and expressive. Among the others, environment attributes allow ABAC to be context-aware, thus making it suitable for a variety of applications including IoT.

### Reference Architecture and Policy Evaluation

An authorization mechanism defines the low level implementation of the control within the system. It can be logically decomposed into key components based on the authorization process. Here, we introduce a reference architecture for authorization mechanisms, which resembles the eXtensible Access Control Markup Language (XACML) reference architecture [24], to study how the policy evaluation process is spread across the entities in the IoT environment by existing authorization solution for IoT. Our reference architecture comprises four main components:

**Policy Enforcement Point (PEP)** provides an interface with the system and is responsible to enforce access decisions.

**Policy Decision Point (PDP)** evaluates access requests against access control policies and determines whether a request should be granted or denied.

**Policy Administration Point (PAP)** acts as the policy storage and offers facilities for policy management.

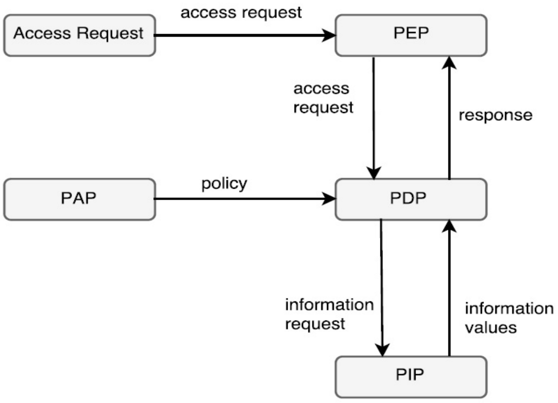


Figure 2.17: Reference Architecture for Access Control Mechanisms

**Policy Information Point (PIP)** denotes the source of information needed for policy evaluation.

Figure 2.17 shows the interaction between these components. Upon receiving an access request from an application, the PEP forwards the request to the PDP for evaluation. The PDP fetches the policies from the PAP and evaluate the request against those policies. If additional information is required for the evaluation of the access request, the PDP queries the PIP. The PDP evaluates the request against the policies and returns a response specifying the access decision to the PEP, which enforces the decision.

An important aspect for determining whether an authorization mechanism satisfies the requirements for access control in IoT is when policy evaluation is performed. For instance, it has a significant impact on the overall architecture of the authorization mechanism (i.e., where its components are deployed) and, thus, can influence the resources (ACR2) and time (ACR3) needed for policies evaluation. It has also an impact on other requirements, e.g. the ability of accounting for context information in decision making (ACR7).

We have identified three main classes of authorization mechanisms with respect to the time when policy evaluation is performed. Policies can be evaluated for every request at *run-time*, *off-line* where authorizations are precomputed, or policy evaluation can be *hybrid* where part of evaluation is done off-line and part is done at run time. Next, we present some common mechanisms for these three policy evaluation schemes.

**Run-time policy evaluation:** In a run-time policy evaluation mechanism, policy evaluation is performed at request time. Frameworks that support run-time evaluation typically employ an authorization mechanism that resembles the reference architecture presented above. Upon receiving an access request, the PDP evaluates the request against the policies made available by the PAP and returns the outcome of policy evaluation (i.e., an access decision) to the PEP for enforcement.

**Off-line policy evaluation:** In an off-line policy evaluation mechanism, the requester posses assertions on its credentials, access permissions and other attributes. These assertions are

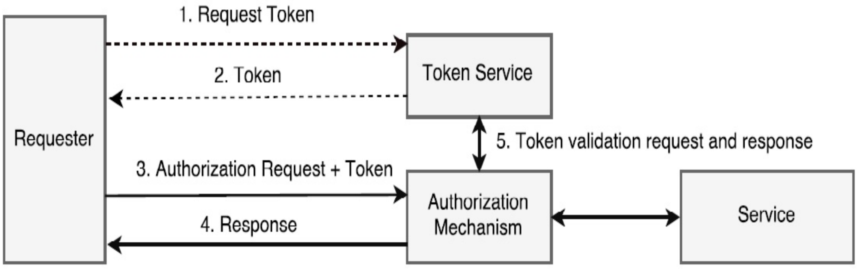


Figure 2.18: Example of Token-based Authorization Mechanism

typically provided by the resource owner or a trusted party. Frameworks that support off-line policy evaluation typically use a token service that issues tokens encoding the assertions possessed by an entity. An example of token-based framework is depicted in Figure 2.18 where a requester wants to access a service. Before being able to issue an access request for the service, the requester should obtain a token proving that it is authorized to access the service from the token service. This token is passed to the authorization mechanism along with the access request, which checks the validity of the token. In some cases, the verification of the token may require communication with the token service, if the authorization mechanism requires the token to be validated. It is worth noting that many existing token-based frameworks use the OAuth 2.0 protocol [22] due to its lightweight nature.

**Hybrid policy evaluation:** Hybrid policy evaluation lies in between run-time and off-line policy evaluation. Specifically, part of policy evaluation is performed off-line, for instance using a token service that asserts the attributes and permissions of an entity within the IoT environment. At request time, apart from validating the assertion, the authorization mechanism also evaluates the context or other additional constraints before making the access decision.

### Analysis of Authorization Mechanisms for IoT

Several frameworks and architectures have been proposed in the literature to enable authorization in IoT. In this section, we review existing access control mechanisms for IoT and analyze them with respect to the requirements.

### Evaluation Criteria

For the analysis of existing authorization for IoT, we have identified a number of evaluation criteria. These criteria aim to provide the basis for an assessment of the similarities and differences amongst existing authorization frameworks for IoT, as well as guide in finding open problems. These criteria can be grouped in three main categories.

The first category is used to assess the purpose of the proposed framework and the assumptions underlying the IoT environment. This category includes of type of *IoT architecture*, *service model* and *resource location*. The type of IoT architecture provides insights on the capabilities of nodes and their interconnections. The service model indicates the purpose of the middleware. In particular, we identify which service model presented in Section 2.5.3 is offered by the proposed framework. Moreover, we determine where resources are hosted to gain clarity on the purpose of proposed frameworks and to analyze them with respect to privacy requirement.

The second category encompasses criteria concerning the properties of authorization mechanisms. In particular, we identify the *access control model*, *context-awareness*, *policy evaluation time*, and location of the *PDP* and *PEP*. The identification of the adopted access control model is particularly important because scalability and dynamicity requirements depend on the type of access control model adopted. We also assess the ability of an authorization framework to support the specification of context-aware policies. Another important property of an authorization mechanism is when policy evaluation takes place (Section 2.5.3). This aspect can be used to determine the impact of policy evaluation on the IoT systems, especially in terms of performance and latency. Moreover, we assess where PDP and PEP lies, e.g. in physical nodes, middleware or both. While this depends on the type of IoT architecture and on the capabilities of physical nodes, this criterion is important to analyze the performance of authorization solutions.

The last category includes the *application domain* for which an authorization mechanism has been proposed and its *key characteristics*. The application domain might provide additional constraints and assumption for the proposed framework. Key characteristics are used to highlight the uniqueness of a framework with respect to other frameworks.

### Access Mechanisms for IoT

An overview of our literature review with respect to the evaluation criteria introduced in the previous section is reported in Table 2.6. For the sake of exposition, frameworks have been grouped with respect to the type of IoT architecture for which they have been proposed (Section 2.5.3). Within each group, existing frameworks are categorized with respect to the policy evaluation process.

**Authorization Mechanisms for Centralized IoT Architectures** In a centralized IoT architecture, the authorization mechanism is typically deployed in the middleware. For example, policy management and other authorization functions are handled by a centralized service within the cloud [166]. Such frameworks are particularly suitable for IoT systems comprising resource constrained devices, as the computation required for authorization is performed in the middleware.

Several authorization solutions for cloud-based middleware have been proposed [58, 63, 107]. For instance, Kaluvuri et al. [107] propose an XACML-based framework that offers authorization as a service in multi-cloud environments. This framework provides users a single point to manage their policies to be enforced across multiple clouds. Similarly, Bernabe et al. [58] and Calero et al. [63] propose an authorization service tailored to multi-cloud environments based on RBAC. Although these frameworks provide an approach to handle authorizations within cloud environments, their suitability for IoT is not proven. When applied to IoT environments, authorization solutions must also comply with the IoT requirements as stated in Table 3.1. Next, we review authorization solutions specifically designed for IoT centralized architectures.

A number of existing authorization frameworks based on a centralized IoT architecture perform policy evaluation at run-time [49, 50, 138]. Neisse et al. [138] propose an authorization framework for Mosquitto. While traditional MQTT implementations usually lack efficient policy enforcement capabilities, the proposed framework aims to provide Mosquitto with run-time support for policy enforcement by integrating SecKit [137] in the Mosquitto broker. SecKit is a model-based security toolkit that provides a PDP and a PEP as well as a means for gathering

| | IoT Architecture | Service Model | Resource Location | Access Control Model | Context Aware | Policy Evaluation | PDP | PEP | Application Domain |
|---|---|---|---|---|---|---|---|---|---|
| Neisse et al. [138] | Centralized | IPMaaS | Physical Node | — | Yes | Run-time | Middleware | Middleware | — |
| Alshehri et al. [49] | Centralized | SAaaS | Virtual Objects | ABAC | Yes | Run-time | Middleware | Middleware | — |
| Alshehri et al. [50] | Centralized | SAaaS | Virtual Objects | ACL<br>RBAC<br>ABAC | Yes | Run-time | Middleware | Middleware | — |
| Fremantle et al. [86] | Centralized | IPMaaS | Physical Node | — | No | Hybrid | Middleware | Middleware | — |
| Fernandez et al. [80] | Centralized | IPMaaS | Application Node | — | No | Hybrid | Middleware | Middleware | — |
| Cirani et al. [67] | Centralized | $S^2$aaS | Middleware | — | No | Hybrid | Middleware | Middleware | — |
| Seitz et al. [159] | Connected | IPMaaS | Physical Node | ABAC | Yes | Run-time | Middleware & Physical Node | Physical Node | — |
| Salonikias et al. [154] | Connected | IPMaaS | Physical Node & Application Node | ABAC | Yes | Run-time | Middleware | Middleware | Transport Infrastructure |
| Ye et al. [188] | Connected | IPMaaS | Physical Nodes | ABAC | Yes | Run-time | Middleware & Physical Node | Physical Node | Wireless Sensor Network |
| Hussein et al. [103] | Connected | IPMaaS | Physical Node | Capability | No | Off-line | Middleware | Physical Node | — |
| Hernandez-Ramos et al. [98] | Connected | IPMaaS | Physical Node | Capability | Yes | Off-line | Middleware & Physical Node | Physical Node | — |
| Gusmeroli et al. [94] | Connected | IPMaaS | Physical Node | Capability | No | Hybrid | Middleware | Physical Node | — |
| Garcia et al. [87] | Distributed | — | Physical Node | RBAC | Yes | Run-time | Physical Node | Physical Node | Medical Sensor Network |
| Dorri et al. [73] | Distributed | — | Physical Node | ACL | No | Run-time | Block-chain | Block-chain | Smart-home |
| Ouaddah et al. [142] | Distributed | — | Physical Node | — | Yes | Hybrid | Block-chain | Physical Node | — |

Table 2.6: Analysis of existing authorization frameworks for IoT

and managing context attributes. Policy decisions are made by SecKit and a mechanism called security plugin is used to extend and embed the PEP in Mosquitto. This allows Mosquitto to have custom authorization functions and enforce the decision before users can publish messages.

Alshehri et al. [49] propose an authorization framework based on ABAC to provide SAaaS. This framework is built on top of an IoT architecture that encompasses an object layer, a virtual object (VO) layer, a middleware layer and an application layer. VO-to-VO communication is based on the publish-subscribe paradigm: resources produced by an object (e.g., a sensor) are captured by a VO and are published to topics that other VOs can subscribe to. VO-to-VO communication is regulated by access control policies deployed in the middleware, which specify which VOs are authorized to publish and subscribe to a certain topic. In a recent work [50], the authors have extended this framework by proposing authorization mechanisms for operational and administrative purposes tailored to VO-to-VO communication. The operational access control mechanism is used to regulate subscriptions and publishing to topics, whereas the administrative access control mechanism is used to configure the operational access control. These mechanisms support policies specified in three access control models, namely DAC (implemented through ACLs), RBAC and ABAC.

Other researchers propose the use of a token service to precompute (part of) the authorization process. Fremantle et al. [86] propose an authorization framework that uses OAuth 2.0 [22] to regulate messages exchanged via MQTT. The framework encompasses physical nodes, two intermediate nodes (an MQTT broker and a token service) and a middleware that provides an authentication and authorization service as shown in Figure 2.19. Upon receiving a request from a physical node, the token service creates an OAuth token, the authorization mechanism validates the token and, based on the outcome of the validation, the MQTT broker grants or denies the physical node the permission to publish/subscribe to a certain topic. In the token generation phase, when the token service receives a token request from a physical node, it redirects the physical node to the authorization and authentication service for authentication. If the authentication is successful, the token service creates a token, which is sent to the physical node. In the authorization phase, when the physical node requests to publish resources on a certain topic, it sends the token to the broker. The broker interacts with the authorization service to validate the token. Along with the outcome of the validation, the authorization service sends the scope in which the token is valid. Intuitively, the scope of a token specifies the permissions associated to a particular token (e.g., the token's holder is only allowed to publish on a certain topic). The broker uses the scope to verify the request from the physical node and determines whether that node is allowed to publish on the topic. We can observe that the PEP lies within the broker and PDP resides partly in the authorization service and partly in the broker. The framework has been implemented using an Arduino board[10] for physical nodes, a Mosquitto MQTT broker and a web tool for the creation of tokens, and a WSO2 identity server[11] acting as the authentication service. Based on this implementation, the authors provide recommendations, such as reducing the token size, to efficiently use OAuth with MQTT.

A token-based framework resembling the reference architecture of Section 2.5.3 is presented in [80]. The framework encompasses an authorization service deployed in the cloud, which provides one PEP for each application node and a centralized PDP and PAP. In addition, the framework encompasses an identity provider (IdP) that acts as a token service. The IdP

[10] https://www.arduino.cc

[11] https://wso2.com/identity-and-access-management

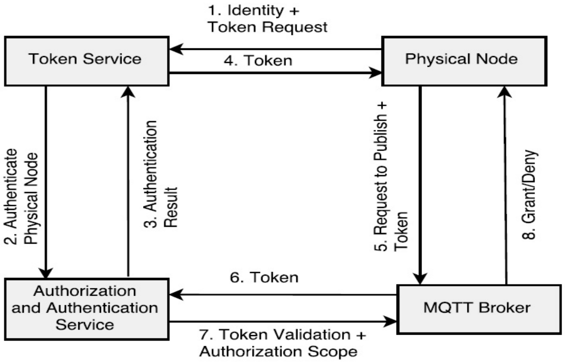
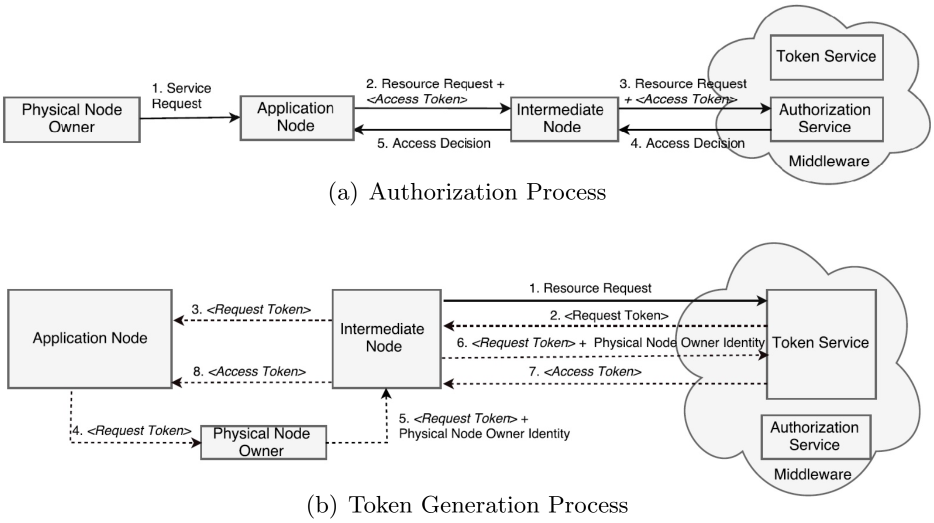

1. Identity + Token Request
Token Service
4. Token
Physical Node
2. Authenticate Physical Node
3. Authentication Result
5. Request to Publish + Token
8. Grant/Deny
Authorization and Authentication Service
6. Token
MQTT Broker
7. Token Validation + Authorization Scope

Figure 2.19: Framework presented in [86].



(a) Authorization Process

(b) Token Generation Process

Figure 2.20: IoT-OAS Framework [67].

manages the credentials of physical and application nodes. Every node is required to register itself with the IdP. When requesting a service provided by an application node, a physical node should provide a token proving its identity. Such a token can be obtained from the IdP. Upon receiving the token, the access request is passed to the authorization service for token validation and policy evaluation (the authorization service may be required to communicate with IdP for token validation). The decision, either permit or deny, resulting from policy evaluation is returned to the corresponding PEP.

Another token-based framework for centralized IoT architectures is proposed by Cirani et al. [67]. This framework, called IoT-OAS, provides an $S^2$aaS service model and uses an external authorization service based on OAuth 2.0. In particular, the framework encompasses a dedicated intermediate node and a middleware for authorization purposes. The middleware provides a token service and an authorization service. The intermediate node – a network broker or a gateway – hosts the resources of physical nodes. When the owner of a physical node wants to access a service from an application node, the application node requests the intermediate node the access to the resources of the physical node. To provide access to resources, the intermediate node expects an *access token* with the request. If an access token is present, the intermediate node forwards the request to the authorization service, which performs policy evaluation as depicted in the authorization process of Figure 2.20(a). If an access token is not provided along with the request, or if the access token has expired, a new access token is generated by binding the identity of the owner of the physical node to the application node offering the requested service. The binding process is performed through a series of token exchanges as shown in the token generation process of Figure 2.20(b). Initially, the application node obtains a *request token* that contains its identity. The application node redirects the owner of the physical node along with the request token to the intermediate node. The owner of the physical node authenticates to the intermediate node and provides its consent for the application node to access its resources. The intermediate node requests the token service to exchange the request token with an access token, which is then provided to the application node. In this framework, physical nodes and the intermediate node are not required to implement the OAuth logic, which is implemented by the authorization service in the middleware. A prototype of this architecture has been deployed on different platforms within Cooja[12], a Contiki OS simulator. This prototype shows that outsourcing the authorization process can benefit the functioning of IoT systems due to the easiness to set up an external authorization service. However, the experiments prove that the use of an external authorization service causes substantial energy consumption in physical nodes, leading a reduced battery lifetime of physical nodes.

**Authorization Mechanisms for Connected IoT Architecture** In a connected IoT architecture, physical nodes are assumed to possess some processing capabilities and, thus, they are able to take part in the authorization process. In particular, intensive computations are typically performed in the middleware or in intermediate nodes, while physical nodes may perform additional check based on local conditions and be responsible for policy enforcement.

A number of authorization frameworks for connected IoT architectures such as [159, 188] perform policy evaluation at run-time. Among these frameworks, the one proposed in [159] is based on SAML [17] and XACML [24], and uses CoAP for message exchange. Here, physical nodes provide resources that must be available only to authorized application nodes. Apart from physical and application nodes, the architecture involves a middleware that makes access decision on behalf of physical nodes. The policies are, however, enforced by a PEP that resides in physical nodes. First, the application node requests the middleware to access the resources of a physical node. The middleware evaluates the request and responds with a SAML assertion, i.e. a digitally signed statement asserting the access decision. The application node sends a CoAP request to the physical node along with this assertion. The physical node verifies the assertion as well as local conditions enabled through XACML obligations, thereby partially performing policy evaluation. The feasibility of this framework has been demonstrated by a prototype implementation deployed in an Arduino board.

Salonikias et al. [154] propose an authorization framework for IoT based on fog computing following an ABAC model (Figure 2.21). The framework is defined for a vehicular infrastructure including physical nodes, roadside units acting as application nodes and a cloud. Since applications are latency sensitive, the framework extends the capability of cloud to the roadside unit area by employing fog nodes. Instead of communicating with the cloud, physical nodes communicate with fog nodes to update local information and get access to services offered by application nodes without delay. Fog nodes can further communicate with the cloud to update or request information. This framework provides multiple PDPs and PEPs spread across fog

[12] http://www.contiki-os.org

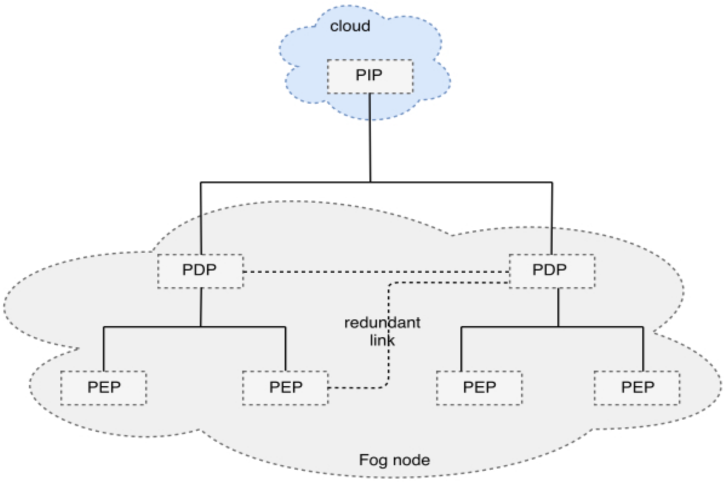


Figure 2.21: Framework presented in [154].

nodes. The PIP, which also encompasses the PAP, resides in the cloud. In particular, the PIP stores policies and attributes that are made available to the PDPs. Physical nodes are linked to a PEP and each PEP is linked to multiple PDPs; all PDPs are linked to the PIP. Moreover, PDPs are linked with each other (dotted line in Figure 2.21), in order for a PDP to receive the updated policies if a connection between the PDP and the PIP cannot be established.

Ye et al. [188] propose an authorization framework for WSNs. The framework follows an ABAC model for policy specification and encompasses application nodes, physical nodes, a gateway and an attribute authority. Physical nodes form small subnets where each subnet is managed by a gateway. The attribute authority creates and manages attribute information of nodes, such as resource and environment attributes. The attribute authority is also responsible to issue attribute certificates to nodes, asserting their attributes. Before accessing a resource, an application node is required to register with the gateway. The gateway verifies if the identity attributes of the application node are valid. Once the registration is successful, the application node can request resources to a physical node by providing its attribute certificates. The physical node verifies whether there is a sufficient overlap between its attributes and the ones of the application node.

Authorization frameworks tailored to connected IoT architectures that support off-line policy evaluation are proposed in [98, 103]. These frameworks are typically based on capability tokens. A capability token can be a key, a certificate or an assertion that uniquely references a resource and the capabilities (access rights) of the requester on the resource [94]. A token is usually generated by a trusted intermediate node. When an application node requests access to the resources of a physical node, it also sends the capability token for validation. Physical nodes verify the validity of the token and possibly local conditions, and enforces the access decision.

Hussein et al. [103] present a community driven capability-based authorization framework (CoCapBac) for IoT. The architecture is centered on the notion of communities, i.e. groups of physical nodes that work together towards a common goal, and access rights are defined based on community norms. The architecture also encompasses a middleware and a certification authority for authorization purposes. The PEP is implemented in powerful physical nodes within the community, called CO gatekeepers, and the PDP resides in the middleware.

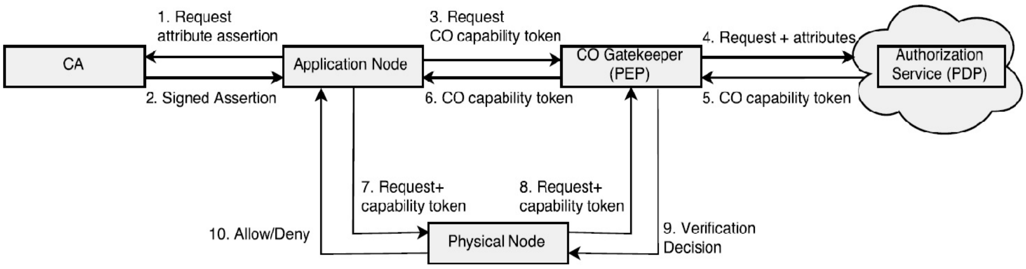


Figure 2.22: Authorization process described in [103]

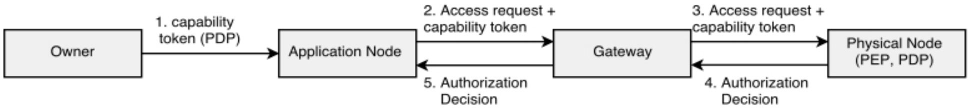


Figure 2.23: Authorization process described in [98]

There can be multiple CO gatekeepers within a community. CoCapBac relies on capability tokens that are used by CO gatekeepers to determine whether application nodes are allowed to access a resource. The capability token generation process and its verification is depicted in Figure 2.22. During the token generation process, the certificate authority asserts the attributes of application nodes. When an application node holding the assertion requests a CO gatekeeper to access a physical node's resources, the CO gatekeeper forwards this request to the middleware. The PDP evaluates the request and generates a CO capability token. Upon receiving the response from the PDP, the CO gatekeeper forwards the token to the application node. Later, the application node requests access to resources along with the capability token to the physical node. The physical node forwards the request to the CO gatekeeper to validate the token and enforce the outcome of policy evaluation.

Another capability-based framework for connected IoT architectures is proposed in [98]. In this framework, the owner of a physical node can issue a capability token to application nodes as shown in Figure 2.23. The capability token is signed with an elliptic curve digital signature algorithm (ECDSA) [106] to verify its integrity. The application node sends a CoAP request to the physical node via a gateway, along with the capability token. The physical node verifies the signature of the token and, thus, the legitimacy of the application node. Moreover, the physical node verifies context conditions. The outcome of policy evaluation is enforced by the physical node and a response with the access decision is sent to the application node that requested access through a CoAP response. The framework has been implemented and deployed on physical nodes that use Contiki OS. In recent works [99, 100] under the SMARTIE project [59], the authors have used optimizations on ECDSA to make it lightweight in order to comply with computational requirements of IoT.

The framework presented in [94] also proposes a capability-based authorization framework. The framework encompasses application nodes, physical nodes acting as the PEP, an external PDP and an external capability and revocation service. The capability service assigns capabilities to application nodes and the revocation service revokes those capabilities that are no longer valid. The framework also focuses on the delegation of capability from one application node to another. An application node sends a request along with its capability token to access a resource to a physical node. The physical node validates the token and checks if all necessary fields in the token (e.g., identity of the application node, validity of the token) are present. If the application node has delegated its capability from another node (which we refer to as the parent node), the physical node verifies if the parent node is capable of delegating its access rights. Access requests are, then, forwarded to the PDP that verifies whether the application node is allowed to perform the requested operation and its capabilities are not in the revocation list. Although the capability token provides information on the privileges of an application node, the PDP checks if its capabilities are still valid at the time of the request, thereby following a hybrid evaluation model. The decision from the PDP is enforced within physical nodes.

**Authorization Mechanisms for Distributed IoT Architectures** In a distributed IoT architecture, physical nodes are assumed to be capable to perform intensive computations and can exchange data with each other without requiring a middleware. Thus, in authorization solutions for these architectures, physical nodes are typically responsible for policy evaluation and enforcement.

A decentralized authorization framework for IoT is proposed in [87]. This framework aims to address challenges specific to medical sensor networks, such as emergency access. A medical sensor network encompasses several patient area network hubs, where each patient are network hub comprises physical nodes that are typically medical sensors. These sensors are owned by a user and are attached to her body for collecting medical information. The proposed framework is based on RBAC and takes into account the context and the health condition of the user for decision making, where a user's health condition is determined based on attributes such as heart rate and blood pressure. Policy decision and enforcement are made within physical nodes themselves. When an application node requests access to a physical node, access is granted based on the health condition of the user. For example, if the user's health condition is critical, access is given to any doctors or medical staff without the need for authorization.

Recent years have seen an increasing interest in the adoption of block-chain technology [145] as underlying technology for access control solution [126], and in particular within IoT [73, 74, 142]. For instance, Ouaddah et al. [142] present a privacy-preserving authorization framework for distributed IoT called FairAccess. This framework uses block-chain technology to efficiently manage authorization policies regulating the access physical nodes' resources and to evaluate access requests issued by application nodes against those policies. Each node has a wallet that encompasses a PEP and a PAP, and stores the node's credentials. Communications between an application node's wallet and a physical node's wallet are recorded in the block-chain as transactions. There are two type of transactions: *grant access* and *get access*. During *grant access*, an application node has to obtain an access token from the physical node in order to prove that the former has access rights on the resources; in *get access*, the application node can use the token to access the resource. Intuitively, a token is an assertion representing the access rights defined by the physical node for the application node on a particular resource. The process to generate the *grant access* transaction is presented in Figure 2.24. Here, the physical node defines an access control policy using the application node's identity and resource identifier, and generates a token encrypted with the public key of the application node. The physical node's wallet transforms the access policy into a script (called *locking script*) that locks the token to the access control policy. The physical node's wallet generates a *grant access* transaction signed using its private key and broadcasts the transaction to other nodes in the network. The miners, which are powerful intermediate nodes in the network, act as the PDP

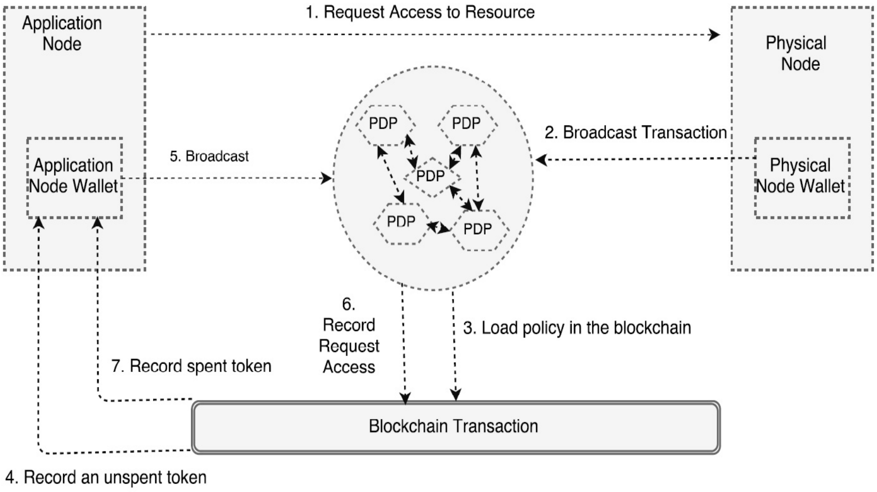


Figure 2.24: Authorization framework presented in [142].

and evaluate these transactions. If the transaction is valid, an unspent transaction is recorded in the block-chain and an unspent token is added to the application node's wallet, indicating that the physical node has granted permission to the requester; however, to access the resource, the application node has to decrypt the token. In *get access*, the application node's wallet decrypts the token and generates an unlocking script for the locking script generated in *grant access*, thus indicating that it satisfies the access policy specified by the physical node. The application node's wallet generates a *get access* transaction and broadcasts it to the PDPs. The PDPs validate the transaction and add it to the block-chain. The application node can now make a request to the physical node along with the token. The physical node validates the token and checks whether the *get access* transaction is present in the block-chain, which proves that the application node has generated an unlocking script and thereby satisfies the access control policy. The final decision whether granting or denying access depends on the local context of the physical node. The framework also supports transactions concerning the delegation and revocation of access rights. Delegation is accounted for using an approach similar to the one described above, but in this case the unspent token is recorded in the wallet of the node to which the permission has been delegated. Revocation is achieved by issuing a *grant access* transaction with a new set of permissions that overrides the previous rights. A proof of concept implementation of this framework is presented in [141] using Raspberry Pi[13] and a bitcoin network. However, the results show that this approach is not suitable for real-time applications due to the long time required by the miners to evaluate transactions.

An authorization framework for smart-home environment based on block-chain is proposed in [73]. This framework encompasses physical nodes (such as thermostats) and a local block-chain managed by powerful physical nodes acting as miners. The local block-chain stores the authorization policies as an access control list and acts as both PDP and PEP. An application node requests data (such as home temperature) to the miner. The miner forwards the request to the local block-chain that performs policy evaluation at run-time. If the access is granted, the miner retrieves the data from the physical node and provides it to the application node. The request made by the application node is recorded as a transaction in the block-chain.

[13] https://www.raspberrypi.org

The framework also allows the storage of data in the cloud. In this case, each transaction corresponding to a data storage in the cloud is also recorded in the block-chain. The miners and physical nodes from several smart-home networks can form an overlay network where they are grouped in clusters. Each cluster has a cluster head that maintains a block-chain storing the transactions of miner-to-miner communication. The same authors show how a similar solution can be employed in the automotive domain [74]. Here, the cluster heads manage a public block-chain that stores the communication between vehicles as transaction. Privacy sensitive information is stored within the vehicle, and the hash of this information is stored in the block-chain. This allows vehicles to verify the integrity of information.

# 3 Usecases

## 3.1 Requirements

### 3.1.1 V2X communication services

This section covers the LTE services requirements defined by 3GPP. Referring to the ETSI report [7], the applications for serving the ITS services define three basic classes: *i*) road safety, *ii*) traffic efficiency, *iii*) other. These services can be divided into two main classes of safety/non-safety services. The expectation of this exchange is described by the requirements below.

**Latency/ Reliability Requirements:**
The Evolved Universal Terrestrial Radio Access Network E-UTRAN should have the capability of:

- Maximum E2E delay of 1000 ms: Communication between the vehicle and a Cloud provider supporting V2C services.
- Maximum latency of 100 ms: Communication between the different entities of the V2X technology via direct link communication or via an RSU, and it should be 20 ms latency for some particular usage.
- Supporting high reliability without requiring application-layer message retransmissions.

**Message Size Requirements:**
The E-UTRAN should have the capability of:

- Sending a periodic broadcast to vehicles with variable message payloads of 50-300 bytes, not including security-related message components.
- Sending notifications to vehicles with variable message payloads which can be up to 1200 bytes, not including security-related message components.

**Frequency Requirements:**
The E-UTRAN should have the capability of:

- maximum frequency up to 10 messages by vehicle per second.

**Range Requirements:**
The E-UTRAN should have the capability of:

- Providing enough response time to the vehicle by supporting a sufficient a communication range.

**Speed Requirements:**
The 3GPP network should have the capability of:

- Transferring messages between vehicles supporting the different entities of V2X technology, while the vehicle is in motion with a speed range variation between 250 km/h and 500 km/h.

**Security Requirements:**
The 3GPP network should have the capability of:

- Providing authorization for vehicles supporting the V2X application to enable V2X communication when served by E-UTRAN.
- Providing integrity for vehicles while communicating with a V2X application.
- Providing pseudonymity and privacy for a vehicle while using the V2X application, to guarantee the identity of the vehicle.
- Providing pseudonymity and privacy for a vehicle using a V2V/V2I application, to guarantee the identity of the vehicle within a different region.

### 3.1.2 IoT Requirements

In an IoT system, the architectural layers should work together to achieve a common goal. However, IoT environments are typically characterized by stringent constraints that the IoT system should satisfy in order to guarantee smooth system functionality. These constraints impose restrictions also on the security mechanisms that can be employed for the protection of the system and data. The requirements presented in Table 3.1 have been obtained from a study of the literature and the current state of affair of IoT systems.

The Internet of Things (IoT) is a network of machines, physical things, devices, and people to facilitate communications and connectivity to relay data so that intelligent services and applications can be created. Cellular technologies have found a solid platform and huge potential as main the players for IoT because they offer ubiquitous coverage, to reinforce IoT, 3GPP is developing several related solutions as well as creating a profusion of GSM-based and LTE-based proposals, for the next section the different cellular technologies and wireless communication(3GPP and 3GPP access for IoT) used to enable the different V2X components are highlighted .

The use cases defined by ETSI are divided into four classes; active road safety, cooperative traffic efficiency, cooperative local services and global Internet services. For these four groups, different applications and facilities apply, for which the use cases are defined. This chapter will briefly present the use cases specified by ETSI and more in detail discuss some of them that directly affect the V2X connectivity and are safety-critical in some aspect. The following 16 are specific V2X use cases presented in PRE-DRIVE C2X project.

- Road work warning
- Stop sign violation
- Traffic jam ahead warning
- Car breakdown warning
- Stop vehicle warning
- Approaching emergency vehicle
- In-vehicle signage
- Regulatory and contextual speed limits
- Traffic info and recommended itinerary

| Category | ID | Requirement |
|---|---|---|
| Scalability | GR1 | The IoT architecture must be scalable to handle the devices. |
| | GR2 | The IoT architecture must be scalable to handle the data generated by the devices. |
| Interoperability | GR3 | All components of the IoT architecture must be able to interact with each other directly or indirectly. |
| Performance | GR4 | The latency of information exchange between the nodes must be minimal. |
| | GR5 | The IoT architecture must ensure that the communication is minimal on the device side. |
| | GR6 | The IoT architecture must ensure that the computation is minimal on the device side. |
| Reliability | GR7 | The IoT architecture must guarantee certain level of reliability and availability of its components. |
| Dynamicity | GR8 | The IoT architecture must handle the dynamicity of the nodes. |
| Openness | GR9 | The IoT architecture must be flexible to allow third party services to access the data. |
| Secure Data Management | GR10 | Data must be stored in the device or in a trusted middleware. |
| | GR11 | Only authorized services must have access to the data. |
| Privacy | GR12 | Data must be managed without compromising user privacy. |
| Network Security | GR13 | The IoT architecture must guarantee the confidentiality and integrity of the communication. |

Table 3.1: IoT Requirements

- Limited access warning
- Decentralized floating data
- Greenlight optimal speed advisory
- Vehicle software provisioning and update
- Fleet management
- Local electronic commerce
- Insurance and financial services

## 3.2 Use cases for ITS

### 3.2.1 Platooning

**Idea:** The user of the communication channel will send messages at a high frequency. The communication channel from transmit node until receive node is available twice to incorporate redundancy in communication. The communication aspects measured during this test are 'message reception rate' and 'message latency'.

**Rationale:** This user story is applicable to Connected Adaptive Cruise Control or Platooning scenarios. In these scenarios, the distance between vehicles is small. Therefore, vehicles need to be able to respond to each other very quickly. Otherwise the safety of the vehicles and its passengers will be in danger. A high Quality of Service of the communication channel is very import but is also depending on the other users of the same communication channel. The requirements imposed onto the communication channel are:

- Redundancy in the communication channel: To improve the reliability of the wireless communication channel, two communication nodes are present per vehicle.
- High message frequency: A high message frequency is needed to assure that vehicles can react automatically and timely on behavior of other vehicles. In this user story a message is communicated through both communication channels every 25 milliseconds.

**Test performance indicators:** To test the communication of the channel the following performance indicators are measured:

- Message reception rate: Every message is sent twice, once per transmission node. And every message that is sent can be received twice, once per reception node. As a result every message can be received in total from 0 to 4 times. The message reception rate indicator shows how many times a message is received 0 times, once, twice, three or four times.
- Message latency: Latency is measured between the moment a message is received from the in-car platform at the transmission node until this message is sent to the in-car platform at the reception node. The message latency indicator creates buckets at a width of 1 millisecond. Per bucket the number of times the latency is between the bucket boundaries is counted. For example: The first bucket counts the messages with latencies from 0 milliseconds up to but no including 1 milliseconds. Every message with a latency between these two limits increases the bucket counter with 1. Please note that the reception of a message from 0 to 4 times is only applicable if both communication channels are interoperable. In order to achieve communication redundancy it is also possible to use non-interoperable communication channels, like a combination of 5G-V2X and 802.11p.

### 3.2.2 Over-The-Air updates

**Idea**:

The move towards software-based connected cars means that the systems have to be capable of provisioning and updating this software on demand, preferably over the air. In V2X scenarios this update process should be performed from RSUs to vehicles or even between vehicles in movement. When a vehicle that has to receive an update stops near a RSU, or similarly drives alongside another vehicle capable of transmitting the update patch, this data have to be sent to the vehicle so that the update process can be initiated. The patch is transmitted to vehicles from RSUs, base stations or even other vehicles spontaneously.

**Connectivity**:

Firstly, this puts requirements on the software itself. The connectivity is the ways through the patches are sent. However, this opens many doors to malicious software threats, so the software and connectivity need to be able to recognize these threats and act accordingly. In addition, the connectivity has to efficiently enable the transmission and reception of these patches also in highly congested areas or to moving nodes. In these terms, scheduling capabilities and availability represent important criteria in addition to security.

In order to test and achieve reliable and operative results, the main goal is to build a connectivity real test-field, by means of the setup of a testbed. More precisely, the idea is to deploy a complete vehicular network infrastructure, within a limited urban area, in which the aforementioned scenarios/use cases can be reproduced. From the empirical point view, this hands-on approach can allow investigating connectivity from different perspectives. Comparing different technologies, both at the application layer and radio network interface, can facilitate the understanding of different performance tradeoff. Furthermore, verifying whether emerging networking concepts (e.g., network slicing) can be used for enhancing connectivity performance is another main objective of these tests.

As regards the evaluation and comparison of different radio interfaces, the objective is to try understanding whether communications through LPWAN network interfaces can be more efficient than communications done by means of Broadband Cellular Networks, and vice versa – from this latter point of view, the considered use case clearly plays a crucial role. Car-to-car communications is also a key issue that wants to be assessed through the testbed.

Using the outcome from empirical evaluation as input for the extension of vehicular network simulators is a further side activity that potentially may be carried out in parallel with the empirical investigation. In fact, one possible limitation about carrying out performance test on a real testbed is due to the lack of resources that allow accomplishing analyses on a very large scale and with different network connectivity conditions.

### 3.2.3 Emergency warnings

**Idea:** In V2X environments, alerts and warnings are often transmitted from RSU's or between vehicles to indicate the conditions on the physical environment (e.g. weather indications) or indicate information about road conditions (e.g. traffic jams). This user-story aims in analyzing the communication aspects of such scenarios as well as testing requirements, such 'reliability', 'message latency', 'message reception rate' and 'network security'.

From the communications perspective there are two main components in the scenario: direct V2V communications for low-latency emergency warnings to vehicles within immediate area of effect, such as an obstacle (rocks, ice, etc.) and V2C emergency warnings for vehicles on later approach to the afflicted area. For the former direct communications using e.g. IEEE 802.11p or C-V2X (modes 3 or 4) is required to guarantee the emergency warning reaches relevant destinations at a time. The communications should also be broadcast in nature with sufficient redundancy to be received with small finite number of transmissions. For the latter delay does not play such a high role but the, reliability of the information as well as continued existence ob the obstacle in question.

**Rationale:** This user story is applicable to Collaborative Perception and Awareness scenarios. In such scenarios, the warnings and alerts should be received in time and concern the automation of reactions in the case of road traffic, extreme weather conditions and detected obstacles in proximity of the car such that it brakes in time. This user story allows to analyze and test different connectivity requirements and in the same time provide interfaces and services to the V2X environment that are currently considered partially.

**Test performance indicators:** To test the communication requirements the following performance indicators are defined:

- Reliability: The tests for this requirement shall be focused in the presence of connectivity between the different entities in V2X communication (Section 2.4). Specifically, in the presence of wireless connectivity it shall be possible to connect through 802.11p communication, whereas in its absence connections should be cellular through 5G communication.
- Message latency: This indicator tests the the warnings and alerts that should be received with low-latency, as they often allow V2X communication entities to know beforehand for critical situations that will be encountered in the road or in the physical environment
- Message reception rate: The tests should aim in satisfying the requirement that the rate of the exchanged data shall not overcome a certain threshold. When this requirement is not satisfied, memory overflow buffer overflow in the vehicles that are communicating. This will compromize the transmission of emergency warnings between the vehicles
- Network security As the exchanged data may contain sensitive information, security aspects should be considered to avoid that they are spoofed or malformed. Therefore, a vital requirement to be tested is the secure data transmission for all types of V2X communication (Section 2.4)

## 3.3 Available communications solutions state of the art demonstration site baseline for use cases

As current state of the art available at the University of Oulu premises we categorise the possible communications demonstrations equipment into four sectors: in active development, available in the next six months, discontinued but available, and designed but not produced. The last sector will available with some effort, if required for the APPSTACLE purposes.

### In active development

The active development can be further categorised in a number of work items: e-NodeB support, end device support, EPC support, MEC support, IoT platform supports and other RF transceiver solutions. The LTE-advanced e-NodeB support covers macrocells at bands 7 and 42. Furthermore, at band 28 there is macrocell operations with Cat NB1 and Cat M1 (NB-IoT) provisions as well as LTE advanced. In addition, the same antenna tower hosts a LoRaWAN base station with development ongoing for integration to become natively incorporated to the 5G core network. All these networks are connected to the Internet using university shared Panoulu network 1 Gbps connection.

The end device support consist of multiple technologies. LTE Cat 4 and higher connections to the University of Oulu network are covered by numerous types of mobile phones and USB dongles. The 5G test network (5GTN) has its own subscriber identity module card base, which can be expanded if required. The NB-IoT devices currently operational are Cat NB1, based on Huawei chipsets and support power saving features, such as power save mode and eDRX. With regards to the LoRaWAN devices the currently utilised units are RN2483 433/868 MHz LoRaWAN radio transceivers with over 300 units constantly deployed and operational.

The EPC supports both Cat NB1 and Cat M1 devices in addition to higher category LTE-advanced devices. The EPC is based on the Nokia AirFrame concept. Virtualisation is achieved using OpenStack (Cloudband) and there are VNF for mobility management entity (MME) and gateway (GW). Further on there exists a virtualized infrastructure manager and VNF manager. The second instantiation of the 5GTN EPC is based on OpenEPC, provided by Core Network Dynamics. The OpenEPC currently supports 3GPP release 12 functionality with secure support for both trusted and untrusted non-3GPP device integration to the core. Release 13, with NB-IoT support will be available during the first quarter of 2018. Furthermore, network slicing as a service is possible throughout the EPC and radio access network.

The MEC support exists in hardware. The software is currently being developed. It can be used to process and distribute information to users for example, in the emergency warning use case of section 3.2.3.

IoT platform support comes in a number of instantiations. ThingWorx is currently used for easy graphical presentation of collected IoT device data. It also enables control of IoT devices. The devices communicate data to an MQTT server from which the data is automatically exported to ThingWorx. The Nokia IMPACT IoT platform is currently being integrated to the 5GTN and it offers a secure, standards based, simplified IoT platform that supports for example NB-IoT and LoRaWAN technologies. From a hardware perspective, the University of Oulu has a modular IoT platform for which there currently exist around ten different sensor, processing, and peripheral extension boards with around ten various transceiver chipsets. Those modular boards can be combined into a platform instantiation in any desired combination, including simultaneous use of multiple radio frequency (RF) transceiver chipsets.

Integrated with the modular IoT platform, as other RF solutions in active development, are the system on a chip CC2650 at 2.4 GHz, the AT86RF233 at 2.4 GHz, and the DWM1000 at 3.5 – 6.5 GHz. The CC2650 supports Bluetooth low energy, IEEE 802.15.4 as well as a few proprietary operational modes. The AT86RF233 is an IEEE 802.15.4 compliant transceiver with an additional 2 Mbps rate. As it was produced prior to IEEE 802.15.4t-2017 standard completion, the 2 Mbps mode is not likely to be fully standard compliant. The DWM1000 radio module is an IEEE 802.15.4-2015 compliant ultra wideband transceiver produced by Decawave and it operates on 500 MHz channels. It is mostly a short-range communications board obtaining less than 20 m range communications.

IEEE 802.11 is available at the University of Oulu premises and in many locations at the city of Oulu through the already 14-year operational Panoulu network. It is an open wifi network, not directly tied in with the 5GTN but solutions may be configured to access the 5GTN.

### Available soon

Currently, significant effort is put into including the 5G new radio to the 5GTN. The 5G new radio will have operational capabilities from 6 GHz to millimeter wave range and the plan is to have it integrated in the test network early 2018. Initially, it will be connected to the newest 4G core network. Once the 5G core network becomes operational the new radio will be integrated to it.

Another new solution is the integration of a dual chip LTE-M & NB-IoT solution into the modular IoT platform. The expected completion of the integration is first half of 2018.

### Discontinued, but still available

There are a number of RF solutions available, for which conversion to the current generation of the IoT hardware platform have not been made. However, as there still are operational previous generation IoT hardware platforms those technologies may be considered for APPSTACLE project purposes. One of the RF solutions is the CC110x sub-GHz radio transceiver. It has been implemented in two versions: the 433 MHz and the 868 MHz bands. Another radio module is the RN4020 Bluetooth Smart. Finally, the radio module LE910 is an LTE Cat 3 radio that could be used. Although the Cat 3 module supports up to 102 Mbps downlink and 51 Mbps uplink data rates the data interface bus for the module is only about 150 kbps, hence making its utilization limited.

#### Designed but not produced, available with some effort

The radio module WF121 IEEE 802.11 b/g/n compatible chipset and can be produced to the IoT hardware platform if needed. The radio module SX1272 is another LoRaWAN radio transceiver that can be produced if the currently used LoRaWAN chipset proves unsatisfactory at some point.

### 3.3.1 Notes on cellular V2X and its availability for demonstration purposes

Cellular V2X (C-V2X) communications were specified in 3GPP release 14. It has two V2V communications modes of interest to APPSTACLE project, namely Mode 3 for scheduling and interference management of V2V traffic, assisted by e-NodeBs via control signaling and Mode 4 for distributed scheduling. In Mode 3 the e-NodeB assigns the resources being used for V2V signaling in a dynamic manner, whereas in Mode 4 the user equipment carry out sensing with semi-persistent transmission. These modes are in direct competition with IEEE 802.11-2012 intelligent transportation systems specifications (IEEE 802.11p-2010) and it would be highly desirable to carry out performance comparison between these two technologies. Unfortunately, there are no C-V2X end user devices available at the moment. They are currently being developed by the manufacturers and the first models should be available around mid 2018. The e-NodeBs also must be upgraded to support especially Mode 3 functionality. Such support is not on the OpenEPC short-term roadmap and investigation is ongoing when it will be available for the Nokia AirFrame technology.

‘

# 4 Conclusion

This Deliverable presents a comprehensive state of the art research with regard to Car2X Communication, Cloud and Network Middleware and corresponding Security Concepts. It highlights the potential of autonomous vehicles for the enhancement of the Intelligent Transportation System, for improving the driver experience and safety, as well as for the different available/up-coming radio technologies and wireless communication protocols in addition to research work conducted to cope with the potential security threats. The efforts done within this deliverable are a part of the task 2.1 in WP2 and it also serves as survey of communication enablers to be used for the communications between the other WPs (In-vehicle platform and the cloud IoT platform) of the APPSTACLE project. We described the V2X technology as a way of exchanging information and critical details between the vehicle and other entities (vehicle, infrastructure, pedestrian, and cloud), and the potential of this new technology in building the autonomous vehicles of the future. The driving experience in the near future will be characterized by intensive use of technologies and communication systems. The V2X communications will transform the way drivers and others make use of roads. However, even though these protocols confer comfort and efficiency, they also face major security and safety challenges. In this vein, a survey is presented on the available security solutions that protect the connected cars from different threats and attacks.

# Bibliography

[1] : *5G Communication Automotive Research and innovation (5GCAR).* https://5gcar.eu/

[2] : *The 5G Infrastructure Public Private Partnership.* https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf

[3] : *AUTOMOTIVE 4.0 Sensing the road ahead for tier 1 suppliers.* https://emea.nttdata.com/uploads/tx_datamintsnodes/Whitepaper_Automotive_Tier1_final_single.pdf

[4] *Contiki: The Open Source OS for the Internet of Things.* – http://www.contiki-os.org/

[5] : *Driverless cars take to the road*

[6] : *E-safety Vehicle Intrusion Protected Applications (EVITA).* https://www.evita-project.org/

[7] : *LTE Service requirements for V2X services.* http://www.etsi.org/deliver/etsits/122100122199/122185/14030060/ts122185v140300p.pdf

[8] : *Mosquitto: An Open Source MQTT v3.1/v3.1.1 Broker.* https://mosquitto.org/

[9] : *V2X communication for autonomous driving: Techical brief.* http://www.auto-talks.com/wp-content/uploads/2016/06/V2X_Communication_for_Autonomous_Driving_Technical_Brief.pdf

[10] : *"Secure Vehicle Communication (SeVeCOM).* http://www.sevecom.org/

[11] Car to Car Communications and Traffic Safety. In: *The Milwaukee Sentinel. Google News Archive* (1926), December

[12] : *Phantom Auto to Be Operated Here: Driver-less car to be demonstrated about city street next saturday - controlled entirely by radio.* https://news.google.com/newspapers?id=PthNAAAAIBAJsjid=yYoDAAAAIBAJpg=6442,3879017hl=e
June 1932

[13] : *Internet Protocol.* Darpa Internet Program Protocol Specification. 1981. – URL https://tools.ietf.org/html/rfc791

[14] : *Transmission Control Protocol.* DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION. 1981. – URL https://tools.ietf.org/html/rfc793

[15] : *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4.* Network Working Group, Request for Comments: 4728. 2007. – URL https://tools.ietf.org/html/rfc4728

[16] : *Transmission of IPv6 Packets over IEEE 802.15.4 Networks.* Request for Comments: 4944. 2007. – URL https://tools.ietf.org/html/rfc4944

[17] : *Security Assertion Markup Language (SAML) V2.0 Technical Overview.* OASIS Security Services TC. 2008. – URL https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[18] : *REST.* W3 Semantic Web. 2011. – URL https://www.w3.org/2001/sw/wiki/REST

[19] : *Autonomous cars through the cars.* https://www.wired.com/2012/02/autonomous-vehicle-history/. June 2012

[20] : *Nissan car drives and parks itself at Ceatec.* OCTOBER 2012

[21] : *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.* Internet Engineering Task Force (IETF) Request for Comments: 6550. 2012. – URL https://tools.ietf.org/html/rfc6550

[22] : *The OAuth 2.0 Authorization Framework.* Internet Engineering Task Force (IETF) Request for Comments: 6749. 2012. – URL https://tools.ietf.org/html/rfc6749

[23] : *ZIGBEE SPECIFICATION.* ZigBee Alliance Board of Directors. 2012. – URL http://www.zigbee.org/download/standards-zigbee-specification/

[24] : *eXtensible Access Control Markup Language (XACML).* OASIS Standard. January 2013. – URL http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[25] : *Information technology – Radio frequency identification for item management – Part 6: Parameters for air interface communications at 860 MHz to 960 MHz General.* ISO/IEC 18000-6:2013. January 2013. – URL https://www.iso.org/standard/59644.html

[26] : *Toyota sneak previews self-drive car ahead of tech show.* JANUARY 2013

[27] : *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Standards Association. 2013. – URL http://standards.ieee.org/about/get/802/802.11.html

[28] : *A brief history of autonomous vehicles technology.* July 2014

[29] : *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence.* Internet Engineering Task Force (IETF): Request for Comments: 3921. 2014. – URL https://www.ietf.org/rfc/rfc3921.txt

[30] : *MQTT Version 3.1.1.* OASIS Standard. 2014. – URL http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[31] : *Self-Driving-Vehicles by 2025....a Death Knell for Auto Manufacturing?* http://www.engineering.com/AdvancedManufacturing/ArticleID/9357/Self-Driving-Vehicles-by-2025a-Death-Knell-for-Auto-Manufacturing.aspx. January 2014

[32] : *The Constrained Application Protocol (CoAP).* Internet Engineering Task Force (IETF): Request for Comments: 7252. 2014. – URL https://tools.ietf.org/html/rfc7252

[33] : *UK to allow driverless cars on public roads in January.* July 2014

[34] : *5G Automotive vision.* `https://5g-ppp.eu/`. October 2015

[35] : *IEEE Standard for Ethernet.* IEEE Std 802.3-2015, IEEE Computer Society. 2015. – URL `http://standards.ieee.org/about/get/802/802.3.html`

[36] : *IEEE Standard for Low-Rate Wireless Networks.* IEEE Std 802.15.4-2015, IEEE Computer Society. 2015. – URL `http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf`

[37] : *Microsoft wants to power self-driving cars, not build one.* https://www.theverge.com/2016/6/3/11850214/microsoft-self-driving-car-software. June 2016

[38] NB-IOT enhancements Work Item proposal. (2016), June. – URL `http://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=713571`

[39] : *The road to driveless cars: 1925 - 2025.* http://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/12665/The-Road-to-Driverless-Cars-1925–2025.aspx. July 2016

[40] V2X cellular Solutions. In: *5G Americas* (2016), October

[41] : *What ford is doing right with the fusion hybrid autonomous car.* https://venturebeat.com/2016/12/29/what-ford-is-doing -right-with-the-fusion-hybrid-autonomous-car/. December 2016

[42] : *Bluetooth Core Specification: The building blocks of your Bluetooth device.* Bluetooth SIG. 2017. – URL `https://www.bluetooth.com/specifications/bluetooth-core-specification`

[43] Connected vehicle cloud:Under the hood. (2017). – URL `http://archive.ericsson.net/service/internet/picov/get?DocNo=28701-FGD101192`

[44] : *Digital Cellular Telecommunications Sytem (Phase 2+) (GSM): GSM/EDGE Radio Transmission and Reception.* 3GPP TS 45.005 version 14.0.0 Release 14. 2017. – URL `http://www.etsi.org/deliver/etsi_ts/145000_145099/145005/14.00.00_60/ts_145005v140000p.pdf`

[45] Ahmed, K. J. ; Lee, M. J.: Secure, LTE-based V2X Service. In: *IEEE Internet of Things Journal* PP (2017), Nb. 99, P. 1–1. – ISSN 2327-4662

[46] Aijaz, Amer ; Bochow, Bernd ; Dötzer, Florian ; Festag, Andreas ; Gerlach, Matthias ; Kroh, Rainer ; Leinmüller, Tim: Attacks on inter vehicle communication systems-an analysis. In: *Proc. WIT* (2006), P. 189–194

[47] Alam, Muhammad ; Ferreira, Joaquim ; Fonseca, José: *Introduction to Intelligent Transportation Systems.* P. 1–17. In: Alam, Muhammad (Publisher) ; Ferreira, Joaquim (Publisher) ; Fonseca, José (Publisher): *Intelligent Transportation Systems: Dependable Vehicular Communications for Improved Road Safety.* Cham : Springer International Publishing, 2016. – URL `http://dx.doi.org/10.1007/978-3-319-28183-4_1`. – ISBN 978-3-319-28183-4

[48] Alheeti, Khattab M A. ; McDonald-Maier, Klaus: Hybrid intrusion detection in connected self-driving vehicles. In: *Automation and Computing (ICAC), 2016 22nd International Conference on* IEEE (Organ.), 2016, P. 456–461

[49] Alshehri, Asma ; Sandhu, Ravi: Access Control Models for Cloud-Enabled Internet of Things: A Proposed Architecture and Research Agenda. In: *Proceedings of International Conference on Collaboration and Internet Computing* IEEE (Organ.), 2016, P. 530–538

[50] Alshehri, Asma ; Sandhu, Ravi: Access Control Models for Virtual Object Communication in Cloud-Enabled IoT, 2017

[51] Anaya, Jose J. ; Merdrignac, Pierre ; Shagdar, Oyunchimeg ; Nashashibi, Fawzi ; Naranjo., JoseE: Vehicle to Pedestrian Communications for Protection of Vulnerable road Users. In: *IEEE Intelligent Vehicles Symposium, Jun 2014, Dearborn, Michigan, United States* (2014), P. pp.1–6

[52] Arif, S. ; Olariu, S. ; Wang, J. ; Yan, G. ; Yang, W. ; Khalil, I.: Datacenter at the Airport: Reasoning about Time-Dependent Parking Lot Occupancy. In: *IEEE Transactions on Parallel and Distributed Systems* 23 (2012), Nov, Nb. 11, P. 2067–2080. – ISSN 1045-9219

[53] Armknecht, F. ; Festag, A. ; Westhoff, D. ; Zeng, K.: Cross-layer Privacy Enhancement and Non-repudiation in Vehicular Communication. In: *Communication in Distributed Systems - 15. ITG/GI Symposium*, Feb 2007, P. 1–12

[54] Arslan, Sibel ; Saritas, Muzeyyen: The effects of OFDM design parameters on the V2X communication performance: A survey. In: *Vehicular Communications* 7 (2017), P. 1 – 6. – ISSN 2214-2096

[55] Baccelli, Emmanuel ; Hahm, Oliver ; Gunes, Mesut ; Wahlisch, Matthias ; Schmidt, Thomas C.: RIOT OS: Towards an OS for the Internet of Things. In: *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on* IEEE (Organ.), 2013, P. 79–80

[56] Beale, Jay ; Baker, Andrew R. ; Esler, Joel: *Snort: IDS and IPS toolkit.* Syngress, 2007

[57] Becker, D. ; Schäufele, B. ; Einsiedler, J. ; Sawade, O. ; Radusch, I.: Vehicle and pedestrian collision prevention system based on smart video surveillance and C2I communication. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, P. 3088–3093. – ISSN 2153-0009

[58] Bernabe, Jorge B. ; Perez, Juan M M. ; Calero, Jose M A. ; Clemente, Felix J G. ; Perez, Gregorio M. ; Skarmeta, Antonio F G.: Semantic-aware multi-tenancy authorization system for cloud architectures. In: *Future Generation Computer Systems* 32 (2014), P. 154–167

[59] Bohli, J. M. ; Skarmeta, A. ; Moreno, M. V. ; García, D. ; Langendörfer, P.: SMARTIE project: Secure IoT data management for smart cities. In: *Proceedings of International Conference on Recent Advances in Internet of Things)*, IEEE, 2015, P. 1–6

[60] Botta, Alessio ; De Donato, Walter ; Persico, Valerio ; Pescapé, Antonio: Integration of Cloud Computing and Internet of Things: a Survey. In: *Future Generation Computer Systems* 56 (2016), P. 684–700

[61] Bouij-Pasquier, Imane ; Ouahman, Abdellah A. ; El Kalam, Anas A. ; Montfort, Mina O. de: SmartOrBAC security and privacy in the Internet of Things. In: *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of* IEEE (Organ.), 2015, P. 1–8

[62] Busnelli, Andrea: *Car Software: 100M Lines of Code and Counting.* https://www.linkedin.com/pulse/20140626152045-3625632-car-software-100m-lines-of-code-and-counting. June 2014

[63] Calero, Jose M A. ; Edwards, Nigel ; Kirschnick, Johannes ; Wilcock, Lawrence ; Wray, Mike: Toward a multi-tenancy authorization system for cloud services. In: *IEEE Security & Privacy* 8 (2010), Nb. 6, P. 48–55

[64] Camps-Mur, D. ; Garcia-Saavedra, A. ; Serrano, P.: Device-to-device communications with Wi-Fi Direct: overview and experimentation. In: *IEEE Wireless Communications* 20 (2013), June, Nb. 3, P. 96–104. – ISSN 1536-1284

[65] Camps-Mur, Daniel ; Pérez-Costa, Xavier ; Sallent-Ribes, Sebastií: Designing Energy Efficient Access Points with Wi-Fi Direct. In: *Comput. Netw.* 55 (2011), September, Nb. 13, P. 2838–2855. – URL http://dx.doi.org/10.1016/j.comnet.2011.06.012. – ISSN 1389-1286

[66] Chim, Tat W. ; Yiu, SM ; Hui, Lucas C. ; Li, Victor O.: Security and privacy issues for inter-vehicle communications in VANETs. In: *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09. 6th Annual IEEE Communications Society Conference on* IEEE (Organ.), 2009, P. 1–3

[67] Cirani, Simone ; Picone, Marco ; Gonizzi, Pietro ; Veltri, Luca ; Ferrari, Gianluigi: IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios. In: *IEEE Sensors Journal* 15 (2015), Nb. 2, P. 1224–1234

[68] Coppola, Riccardo ; Morisio, Maurizio: Connected Car: Technologies, Issues, Future Trends. In: *ACM Comput. Surv.* 49 (2016), Oktober, Nb. 3, P. 46:1–46:36. – URL http://doi.acm.org/10.1145/2971482. – ISSN 0360-0300

[69] Datta, S. K. ; Haerri, J. ; Bonnet, C. ; Costa, R. Ferreira D.: Vehicles as Connected Resources: Opportunities and Challenges for the Future. In: *IEEE Vehicular Technology Magazine* 12 (2017), June, Nb. 2, P. 26–35. – ISSN 1556-6072

[70] Dillon, Tharam ; Wu, Chen ; Chang, Elizabeth: Cloud computing: issues and challenges. In: *Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA),* IEEE (Organ.), 2010, P. 27–33

[71] Dinh, Hoang T. ; Lee, Chonho ; Niyato, Dusit ; Wang, Ping: A survey of mobile cloud computing: architecture, applications, and approaches. In: *Wireless communications and mobile computing* 13 (2013), Nb. 18, P. 1587–1611

[72] Distefano, Salvatore ; Merlino, Giovanni ; Puliafito, Antonio: Sensing and actuation as a service: A new development for clouds. In: *Proceedings of International Symposium on Network Computing and Applications* IEEE (Organ.), 2012, P. 272–275

[73] Dorri, Ali ; Kanhere, Salil S. ; Jurdak, Raja: Blockchain in Internet of Things: Challenges and Solutions. In: *arXiv preprint arXiv:1608.05187* (2016)

[74] Dorri, Ali ; Steger, Marco ; Kanhere, Salil S. ; Jurdak, Raja: BlockChain: A distributed solution to automotive security and privacy. In: *arXiv preprint arXiv:1704.00073* (2017)

[75] Duan, Y. ; Borgiattino, C. ; Casetti, C. ; Chiasserini, C. F. ; Giaccone, P. ; Ricca, M. ; Malabocchia, F. ; Turolla, M.: Wi-Fi Direct Multi-group Data Dissemination for Public Safety. In: *WTC 2014; World Telecommunications Congress 2014*, June 2014, P. 1–6

[76] Eichler, S.: Performance Evaluation of the IEEE 802.11p WAVE Communication Standard. In: *2007 IEEE 66th Vehicular Technology Conference*, Sept 2007, P. 2199–2203. – ISSN 1090-3038

[77] Eiza, M. H. ; Ni, Q.: Driving with Sharks: Rethinking Connected Vehicles with Vehicle Cybersecurity. In: *IEEE Vehicular Technology Magazine* 12 (2017), June, Nb. 2, P. 45–51. – ISSN 1556-6072

[78] ETSI: *Evolved Universal Terrestrial Radio Access (E-UTRA), Packet Data Convergence Protocol (PDCP) specification*

[79] Farjady, Farshid ; Parker, Michael C.: *Wavelength division multiplexing.* 15 Jan. 2002. – U.S. Patent No. 6,339,664

[80] Fernández, Federico ; Alonso, Álvaro ; Marco, Lourdes ; Salvachúa, Joaquín: A model to enable application-scoped access control as a service for IoT using OAuth 2.0. In: *Proceedings of Conference on Innovations in Clouds, Internet and Networks* IEEE (Organ.), 2017, P. 322–324

[81] Festag, A. ; Noecker, G. ; Strassberger, M. ; Lübke, A. ; Bochow, B. ; Torrent-moreno, M. ; Schnaufer, S. ; Eigner, R. ; Catrinescu, C. ; Kunisch, J.: *'NoW – Network on Wheels': Project Objectives, Technology and Achievements*

[82] Festag, Andreas ; Baldessari, Roberto ; Zhang, Wenhui ; Le, Long ; Sarma, Amardeo ; Fukukawa, Masatoshi: Car-2-x communication for safety and infotainment in europe. In: *NEC Technical Journal* 3 (2008), Nb. 1, P. 21–26

[83] Festag, Andreas ; Fussler, Holger ; Hartenstein, Hannes ; Sarma, Amardeo ; Schmitz, Ralf: Fleetnet: Bringing Car-to-car Communication into the Real World, 2004

[84] Festag, Andreas ; Hessler, Alban ; Baldessari, Roberto ; Le, Long ; Zhang, Wenhui ; Westhoff, Dirk: Vehicle-to-vehicle and Road-side Sensor Communication for Enhanced Road Safety, 2008

[85] Fox, Geoffrey C. ; Kamburugamuve, Supun ; Hartman, Ryan D.: Architecture and measured characteristics of a cloud based internet of things. In: *Prooceedings of International Conference on Collaboration Technologies and Systems* IEEE (Organ.), 2012, P. 6–12

[86] Fremantle, Paul ; Aziz, Benjamin ; Kopeckỳ, Jacek ; Scott, Philip: Federated identity and access management for the internet of things. In: *Proceedings of International Workshop on Secure Internet of Things (SIoT)* IEEE (Organ.), 2014, P. 10–17

[87] Garcia-Morchon, Oscar ; Wehrle, Klaus: Modular context-aware access control for medical sensor networks. In: *Proceedings of ACM Symposium on Access Control Models and Technologies* ACM (Organ.), 2010, P. 129–138

[88] Gay, David ; Levis, Philip ; Von Behren, Robert ; Welsh, Matt ; Brewer, Eric ; Culler, David: The nesC language: A holistic approach to networked embedded systems. In: *Acm Sigplan Notices* Volume 38 ACM (Organ.), 2003, P. 1–11

[89] Goldacker, Matthias Gerlach Andreas Festag Tim Leinmuller G. ; Harsch, Charles: Security Architecture for Vehicular Communication.

[90] Graham, G S. ; Denning, Peter J.: Protection: principles and practice. In: *Proceedings of the May 16-18, 1972, spring joint computer conference* ACM (Organ.), 1972, P. 417–429

[91] Grover, Jyoti ; Laxmi, Vijay ; Gaur, Manoj S.: Sybil attack detection in VANET using neighbouring vehicles. In: *International Journal of Security and Networks* 9 (2014), Nb. 4, P. 222–233

[92] Gubbi, Jayavardhana ; Buyya, Rajkumar ; Marusic, Slaven ; Palaniswami, Marimuthu: Internet of Things (IoT): A vision, architectural elements, and future directions. In: *Future Generation Computer Systems* 29 (2013), P. 1645–1660

[93] Guette, Gilles ; Ducourthial, Bertrand: On the Sybil attack detection in VANET. In: *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on* IEEE (Organ.), 2007, P. 1–6

[94] Gusmeroli, Sergio ; Piccione, Salvatore ; Rotondi, Domenico: A capability-based security approach to manage access control in the internet of things. In: *Mathematical and Computer Modelling* 58 (2013), Nb. 5, P. 1189–1205

[95] Hartmans, Avery: *How Google's self-driving car project rose from a crazy idea to a top contender in the race toward a driverless future.* OCTOBER 2016

[96] He, W. ; Yan, G. ; Xu, L. D.: Developing Vehicular Data Cloud Services in the IoT Environment. In: *IEEE Transactions on Industrial Informatics* 10 (2014), May, Nb. 2, P. 1587–1595. – ISSN 1551-3203

[97] Heijden, Rens W. van der ; Lukaseder, Thomas ; Kargl, Frank: Analyzing Attacks on Cooperative Adaptive Cruise Control (CACC). In: *arXiv preprint arXiv:1710.05789* (2017)

[98] Hernandez-Ramos, Jose L. ; Jara, Antonio J. ; Marin, Leandro ; Skarmeta, Antonio F.: Distributed capability-based access control for the internet of things. In: *Journal of Internet Services and Information Security* 3 (2013), Nb. 3/4, P. 1–16

[99] Hernández-Ramos, José L ; Jara, Antonio J. ; Marín, Leandro ; Skarmeta Gómez, Antonio F.: DCapBAC: Embedding Authorization Logic into Smart Things through ECC Optimizations. In: *International Journal of Computer Mathematics* 93 (2016), Nb. 2, P. 345–366

[100] Hernandez-Ramos, Jose L. ; Pawlowski, Marcin P. ; Jara, Antonio J. ; Skarmeta, Antonio F. ; Ladid, Latif: Toward a lightweight authentication and authorization framework for smart objects. In: *IEEE Journal on Selected Areas in Communications* 33 (2015), Nb. 4, P. 690–702

[101] Hindle, Abram: Orchestrating Your Cloud-Orchestra. In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. Baton Rouge, Louisiana, USA : The School of Music and the Center for Computation and Technology (CCT), Louisiana State University, 2015 (NIME 2015), P. 121–125. – URL `http://dl.acm.org/citation.cfm?id=2993778.2993811`. – ISBN 978-0-692-49547-6

[102] Hu, Vincent C. ; Ferraiolo, David ; Kuhn, Rick ; Schnitzer, Adam ; Sandlin, Kenneth ; Miller, Robert ; Scarfone, Karen: Guide to Attribute Based Access Control (ABAC) Definition and Considerations. In: *NIST Special Publication* 800 (2014), P. 162

[103] Hussein, Dina ; Bertin, Emmanuel ; Frey, Vincent: A Community-Driven Access Control Approach in Distributed IoT Environments. In: *IEEE Communications Magazine* 55 (2017), Nb. 3, P. 146–153

[104] Jang, I. ; Choo, S. ; Kim, M. ; Pack, S. ; Dan, G.: The Software-Defined Vehicular Cloud: A New Level of Sharing the Road. In: *IEEE Vehicular Technology Magazine* 12 (2017), June, Nb. 2, P. 78–88. – ISSN 1556-6072

[105] Jenkins, Michael ; Mahmud, Syed M.: Security Needs for the Future Intelligent Vehicles. In: *SAE Technical Paper*, SAE International, 04 2006. – URL `http://dx.doi.org/10.4271/2006-01-1426`

[106] Johnson, Don ; Menezes, Alfred ; Vanstone, Scott: The Elliptic Curve Digital Signature Algorithm (ECDSA). In: *International Journal of Information Security* 1 (2001), Nb. 1, P. 36–63

[107] Kaluvuri, Samuel P. ; Egner, Alexandru I. ; Hartog, Jerry den ; Zannone, Nicola: SAFAX – an extensible authorization service for cloud environments. In: *Frontiers in ICT* 2 (2015)

[108] Kanade, Takeo ; Thorpe, Chuck ; Whittaker, William: Autonomous Land Vehicle Project at CMU. (1986), P. 71–80. – URL `http://doi.acm.org/10.1145/324634.325197`. ISBN 0-89791-177-6

[109] Kargl, Frank ; Klenk, Andreas ; Schlott, Stefan ; Weber, Michael: Advanced detection of selfish or malicious nodes in ad hoc networks. In: *ESAS* Springer (Organ.), 2004, P. 152–165

[110] Katsaros, Konstantinos ; Dianati, Mehrdad: A conceptual 5G vehicular networking architecture. In: *5G Mobile Communications*. Springer, 2017, P. 595–623

[111] Khodaei, M. ; Papadimitratos, P.: The Key to Intelligent Transportation: Identity and Credential Management in Vehicular Communication Systems. In: *IEEE Vehicular Technology Magazine* 10 (2015), Dec, Nb. 4, P. 63–69. – ISSN 1556-6072

[112] Koc, Ali T. ; Jha, Satish C. ; Gupta, Maruti ; Vannithamby, Rath: Extended discontinuous reception (drx) cycle length in wireless communication networks. In: *U.S. Patent Application No* 14 (2013)

[113] Kocher, Paul ; Lee, Ruby ; McGraw, Gary ; Raghunathan, Anand: Security As a New Dimension in Embedded System Design. In: *Proceedings of the 41st Annual Design Automation Conference*. New York, NY, USA : ACM, 2004 (DAC '04), P. 753–760. – URL `http://doi.acm.org/10.1145/996566.996771`. – Moderator-Ravi, Srivaths. – ISBN 1-58113-828-8

[114] Koscher, K. ; Czeskis, A. ; Roesner, F. ; Patel, S. ; Kohno, T. ; Checkoway, S. ; McCoy, D. ; Kantor, B. ; Anderson, D. ; Shacham, H. ; Savage, S.: Experimental Security Analysis of a Modern Automobile. In: *2010 IEEE Symposium on Security and Privacy*, May 2010, P. 447–462. – ISSN 1081-6011

[115] LaPadula, Len ; Bell, D E. ; LaPadula, Leonard J.: Secure computer systems: Mathematical foundations. In: *Draft MTR, The MITRE Corporation* 2 (1973)

[116] Larminie, James ; Lowry, John: *Electric Vehicle Modelling*. P. 187–216. In: *Electric Vehicle Technology Explained*, John Wiley and Sons, Ltd, 2012. – URL `http://dx.doi.org/10.1002/9781118361146.ch8`. – ISBN 9781118361146

[117] Lee, S. ; Park, J. ; Kim, D. ; Hong, Y. G.: An energy efficient vehicle to pedestrian communication method for safety applications. In: *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2014, P. 7–8. – ISSN 2165-8528

[118] Levis, Philip ; Madden, Sam ; Polastre, Joseph ; Szewczyk, Robert ; Whitehouse, Kamin ; Woo, Alec ; Gay, David ; Hill, Jason ; Welsh, Matt ; Brewer, Eric u. a.: TinyOS: An operating system for sensor networks. In: *Ambient intelligence*. Springer, 2005, P. 115–148

[119] Li, Z. ; Kolmanovsky, I. ; Atkins, E. ; Lu, J. ; Filev, D. P. ; Michelini, J.: Road Risk Modeling and Cloud-Aided Safety-Based Route Planning. In: *IEEE Transactions on Cybernetics* 46 (2016), Nov, Nb. 11, P. 2473–2483. – ISSN 2168-2267

[120] Litman, Todd: *Autonomous Vehicle Implementation Predictions Implications for Transport Planning*. `http://www.vtpi.org/avip.pdf`. MAY 2017

[121] Litman, Todd: *What's Driving the Connected Car?* `https://www.spirent.com/~/media/White%20Papers/Automotive/Connected_Car_Whitepaper.pdf`. MAY 2017

[122] Liyanage, Mohan ; Chang, Chii ; Srirama, Satish N.: mePaaS: mobile-embedded platform as a service for distributing fog computing to edge nodes. In: *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2016 17th International Conference on* IEEE (Organ.), 2016, P. 73–80

[123] LoRa Alliance: *A technical overview of LoRa and LoRaWAN.* November 2015

[124] Lu, N. ; Cheng, N. ; Zhang, N. ; Shen, X. ; Mark, J. W.: Connected Vehicles: Solutions and Challenges. In: *IEEE Internet of Things Journal* 1 (2014), Aug, Nb. 4, P. 289–299. – ISSN 2327-4662

[125] Lyamin, Nikita ; Vinel, Alexey ; Jonsson, Magnus ; Loo, Jonathan: Real-time detection of denial-of-service attacks in IEEE 802.11 p vehicular networks. In: *IEEE Communications letters* 18 (2014), Nb. 1, P. 110–113

[126] Maesa, Damiano Di F. ; Mori, Paolo ; Ricci, Laura: Blockchain Based Access Control. In: *Proceedings of IFIP International Conference on Distributed Applications and Interoperable Systems* Springer (Organ.), 2017, P. 206–220

[127] Mahmood, A. ; Casetti, C. ; Chiasserini, C. F. ; Giaccone, P. ; Harri, J.: Mobility-aware edge caching for connected cars. In: *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jan 2016, P. 1–8

[128] Malasri, Kriangsiri ; Wang, Lan: Design and implementation of a securewireless mote-based medical sensor network. In: *Sensors* 9 (2009), Nb. 8, P. 6273–6297

[129] Masmoudi, Khaled ; Afifi, Hossam: An identity-based key management framework for personal networks. In: *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on* Volume 1 IEEE (Organ.), 2007, P. 537–543

[130] MEHL, DR. R.: *THE AUTOMOTIVE INDUSTRY AS A DIGITAL BUSINESS.* `http://www.ntti3.com/wp-content/uploads/Automotive_as_a_Digital_Business_V1.03-1.pdf`

[131] Mejri, Mohamed N. ; Ben-Othman, Jalel: Entropy as a new metric for denial of service attack detection in vehicular ad-hoc networks. In: *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems* ACM (Organ.), 2014, P. 73–79

[132] Mekki, Tesnim ; Jabri, Issam ; Rachedi, Abderrezak ; Jemaa, Maher ben: Vehicular cloud networks: Challenges, architectures, and future directions. In: *Vehicular Communications* (2016), P. –. – URL `http://www.sciencedirect.com/science/article/pii/S2214209616300559`. – ISSN 2214-2096

[133] Mell, Peter ; Grance, Tim u. a.: The NIST definition of cloud computing. (2011)

[134] Mousannif, Hajar ; Khalil, Ismail ; Olariu, Stephan: Cooperation As a Service in VANET: Implementation and Simulation Results. In: *Mob. Inf. Syst.* 8 (2012), April, Nb. 2, P. 153–172. – URL `http://dx.doi.org/10.3233/MIS-2012-0136`. – ISSN 1574-017X

[135] Muhammad, Khaza N. ; Soyturk, Mujdat ; Avcil, Muhammed N. ; Kantarci, Burak ; Matthews, Jeanna: From Vehicular Networks to Vehicular Clouds in Smart Cities.

[136] N.Charette, Robert: *This Car Runs on Code.* `http://spectrum.ieee.org/transportation/systems/this-car-runs-on-code`. February 2009

[137] Neisse, Ricardo ; Fovino, Igor N. ; Baldini, Gianmarco ; Stavroulaki, Vera ; Vlacheas, Panagiotis ; Giaffreda, Raffaele: A model-based security toolkit for the internet of things. In: *Availability, Reliability and Security (ARES), 2014 Ninth International Conference on* IEEE (Organ.), 2014, P. 78–87

[138] Neisse, Ricardo ; Steri, Gary ; Baldini, Gianmarco: Enforcement of security policy rules for the internet of things. In: *Proceedings of the 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* IEEE (Organ.), 2014, P. 165–172

[139] Noor, Talal ; Sheng, Quan: Trust as a service: A framework for trust management in cloud environments. In: *Web Information System Engineering–WISE 2011* (2011), P. 314–321

[140] Osseiran, Afif ; Boccardi, Federico ; Braun, Volker ; Kusume, Katsutoshi ; Marsch, Patrick ; Maternia, Michal ; Queseth, Olav ; Schellmann, Malte ; Schotten, Hans ; Taoka, Hidekazu u. a.: Scenarios for 5G mobile and wireless communications: the vision of the METIS project. In: *IEEE Communications Magazine* 52 (2014), Nb. 5, P. 26–35

[141] Ouaddah, Aafaf ; Abou Elkalam, Anas ; Ait Ouahman, Abdellah: FairAccess: a new Blockchain-based access control framework for the Internet of Things. In: *Security and Communication Networks* 9 (2016), Nb. 18, P. 5943–5964

[142] Ouaddah, Aafaf ; Elkalam, Anas A. ; Ouahman, Abdellah A.: Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In: *Europe and MENA Cooperation Advances in Information and Communication Technologies.* Springer, 2017, P. 523–533

[143] Parno, Bryan ; Perrig, Adrian: Challenges in securing vehicular networks. In: *Workshop on hot topics in networks (HotNets-IV)* Maryland, USA (Organ.), 2005, P. 1–6

[144] Perera, Charith ; Zaslavsky, Arkady ; Christen, Peter ; Georgakopoulos, Dimitrios: Sensing as a service model for smart cities supported by internet of things. In: *Transactions on Emerging Telecommunications Technologies* 25 (2014), Nb. 1, P. 81–93

[145] Pilkington, Marc: Blockchain technology: principles and applications. In: *Research Handbook on Digital Transformations.* Edward Elgar Publishing, 2016, Chap. 11, P. 225–253

[146] Postel, Jon: User Datagram Protocol. 1980 (768). – RFC

[147] Pougajendy, Jayashree ; Parthiban, Arun Raj K.: CDAI: a novel collaborative detection approach for impersonation attacks in vehicular ad-hoc networks. In: *Security and Communication Networks* 9 (2016), Nb. 18, P. 5547–5562

[148] Rabadi, N. M. ; Mahmud, S. M.: Privacy Protection Among Drivers in Vehicle-to-Vehicle Communication Networks. In: *2007 4th IEEE Consumer Communications and Networking Conference*, Jan 2007, P. 281–286. – ISSN 2331-9852

[149] Rao, BB P. ; Saluia, Paval ; Sharma, Neetu ; Mittal, Ankit ; Sharma, Shivay V.: Cloud computing for Internet of Things & sensing based applications. In: *Proceedings of International Conference on Sensing Technology* IEEE (Organ.), 2012, P. 374–380

[150] Ratasuk, Rapeepat ; Mangalvedhe, Nitin ; Zhang, Yanji ; Robert, Michel ; Koskinen, Jussi-Pekka: Overview of narrowband IoT in LTE Rel-13. In: *IEEE conference on Standards for Communications and Networking (CSCN)*, 2016

[151] Raya, Maxim ; Papadimitratos, Panagiotis ; Aad, Imad ; Jungels, Daniel ; Hubaux, Jean-Pierre: Eviction of misbehaving and faulty nodes in vehicular networks. In: *IEEE Journal on Selected Areas in Communications* 25 (2007), Nb. 8

[152] Refaat, Tarek K. ; Kantarci, Burak ; Mouftah, Hussein T.: Virtual Machine Migration and Management for Vehicular Clouds. In: *Veh. Commun.* 4 (2016), April, Nb. C, P. 47–56. – URL `https://doi.org/10.1016/j.vehcom.2016.05.001`. – ISSN 2214-2096

[153] Roman, Rodrigo ; Zhou, Jianying ; Lopez, Javier: On the Features and Challenges of Security and Privacy in Distributed Internet of Things. In: *Computer Networks* 57 (2013), Nb. 10, P. 2266–2279

[154] Salonikias, Stavros ; Mavridis, Ioannis ; Gritzalis, Dimitris: Access control issues in utilizing fog computing for transport infrastructure. In: *Proceedings of International Conference on Critical Information Infrastructures Security* Springer (Organ.), 2015, P. 15–26

[155] Samarati, Pierangela ; Vimercati, Sabrina C. de: Access control: Policies, models, and mechanisms. In: *International School on Foundations of Security Analysis and Design* Springer (Organ.), 2000, P. 137–196

[156] Sandell, Magnus ; Beek, Jaap van de ; Börjesson, Per O.: Timing and frequency synchronization in OFDM systems using the cyclic prefix. In: *International Symposium on Synchronization*, 1995, P. 1995–15

[157] Sandhu, Ravi S. ; Coyne, Edward J. ; Feinstein, Hal L. ; Youman, Charles E.: Role-based access control models. In: *Computer* 29 (1996), Nb. 2, P. 38–47

[158] Sedjelmaci, Hichem ; Senouci, Sidi M.: An accurate and efficient collaborative intrusion detection framework to secure vehicular networks. In: *Computers & Electrical Engineering* 43 (2015), P. 33–47

[159] Seitz, Ludwig ; Selander, Göran ; Gehrmann, Christian: Authorization Framework for the Internet-of-Things. In: *Proceedings of International Symposium on A World of Wireless, Mobile and Multimedia Networks* IEEE (Organ.), 2013, P. 1–6

[160] Seo, Hanbyul ; Lee, Ki-Dong ; Yasukawa, Shinpei ; Peng, Ying ; Sartori, Philippe: *LTE evolution for vehicle-to-everything services.* IEEE Communications Magazine, 2016

[161] Sethi, Pallavi ; Sarangi, Smruti R.: Internet of Things: Architectures, Protocols, and Applications. In: *Journal of Electrical and Computer Engineering* 2017 (2017)

[162] Shang, Wentao ; Afanasyev, Alex ; Zhang, Lixia: The design and implementation of the NDN protocol stack for RIOT-OS. In: *Globecom Workshops (GC Wkshps), 2016 IEEE* IEEE (Organ.), 2016, P. 1–6

[163] Sheng, Xiang ; Tang, Jian ; Xiao, Xuejie ; Xue, Guoliang: Sensing as a service: Challenges, solutions and future directions. In: *IEEE Sensors journal* 13 (2013), Nb. 10, P. 3733–3741

[164] Sheng, Zhengguo ; Yang, Shusen ; Yu, Yifan ; Vasilakos, Athanasios ; Mccann, Julie ; Leung, Kin: A survey on the IETF protocol suite for the Internet of Things: standards, challenges, and opportunities. In: *IEEE Wireless Communications* 20 (2013), Nb. 6, P. 91–98

[165] Socolofsky, T. ; Kale, C.: A TCP/IP Tutorial / Network Working Group. 1991 (1180). – RFC

[166] Soldatos, John ; Kefalakis, Nikos ; Hauswirth, Manfred ; Serrano, Martin ; Calbimonte, Jean-Paul ; Riahi, Mehdi ; Aberer, Karl ; Prakash Jayaraman, Prem ; Zaslavsky, Arkady ; Podnar Zarko, Ivana u. a.: OpenIoT: Open Source Internet-of-Things in the Cloud. In: *Proceedings of the Workshop on Interoperability and Open-Source Solutions for the Internet of Things*, Springer, 2015, P. 13–25

[167] Stojmenovic, Ivan ; Wen, Sheng: The fog computing paradigm: Scenarios and security issues. In: *Proceedings of Federated Conference on Computer Science and Information Systems* IEEE (Organ.), 2014, P. 1–8

[168] Sugimoto, C. ; Nakamura, Y.: Provision of information support by pedestrian-to-vehicle communication system. In: *2008 8th International Conference on ITS Telecommunications*, Oct 2008, P. 160–163

[169] Sugimoto, C. ; Nakamura, Y. ; Hashimoto, T.: Development of Pedestrian-to-Vehicle Communication System Prototype for Pedestrian Safety Using both Wide-Area and Direct Communication. In: *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*, March 2008, P. 64–69. – ISSN 1550-445X

[170] Taleb, T. ; Ksentini, A.: Follow me cloud: interworking federated clouds and distributed mobile networks. In: *IEEE Network* 27 (2013), September, Nb. 5, P. 12–19. – ISSN 0890-8044

[171] Taleb, T. ; Ksentini, A. ; Frangoudis, P.: Follow-Me Cloud: When Cloud Services Follow Mobile Users. In: *IEEE Transactions on Cloud Computing* PP (2016), Nb. 99, P. 1–1. – ISSN 2168-7161

[172] Terzo, Olivier ; Ruiu, Pietro ; Bucci, Enrico ; Xhafa, Fatos: Data as a service (DaaS) for sharing and processing of large data collections in the cloud. In: *Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on* IEEE (Organ.), 2013, P. 475–480

[173] Tomandl, Andreas ; Fuchs, Karl-Peter ; Federrath, Hannes: REST-Net: A dynamic rule-based IDS for VANETs. In: *Wireless and Mobile Networking Conference (WMNC), 2014 7th IFIP* IEEE (Organ.), 2014, P. 1–8

[174] Tsiftes, Nicolas ; Eriksson, Joakim ; Dunkels, Adam: Low-power wireless IPv6 routing with ContikiRPL. In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks* ACM (Organ.), 2010, P. 406–407

[175] Tyagi, Parul ; Dembla, Deepak: Performance analysis and implementation of proposed mechanism for detection and prevention of security attacks in routing protocols of vehicular ad-hoc network (VANET). In: *Egyptian Informatics Journal* (2016)

[176] Uhlemann, E.: The US and Europe Advances V2V Deployment [Connected Vehicles]. In: *IEEE Vehicular Technology Magazine* 12 (2017), June, Nb. 2, P. 18–22. – ISSN 1556-6072

[177] Varadharajan, Vijay ; Tupakula, Udaya: Security as a service model for cloud environment. In: *IEEE Transactions on network and Service management* 11 (2014), Nb. 1, P. 60–75

[178] Verendel, Vilhelm ; Nilsson, Dennis K. ; Larson, Ulf E. ; Jonsson, Erland: An approach to using honeypots in in-vehicle networks. In: *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th* IEEE (Organ.), 2008, P. 1–5

[179] Verma, Karan ; Hasbullah, Halabi ; Kumar, Ashok: Prevention of DoS attacks in VANET. In: *Wireless personal communications* 73 (2013), Nb. 1, P. 95–126

[180] Vincent, JAmes: *Mercedes-Benz and Bosch team up for the latest self-driving car partnership.* April 2017

[181] Wang, Jian ; Liu, Yanheng ; Gao, Wenbin ; Wang, Jian ; Liu, Yanheng ; Gao, Wenbin: *Securing Internet of Vehicles Using TCM*

[182] Weinmiller, Jost ; Schläger, Morten ; Festag, Andreas ; Wolisz, Adam: Performance Study of Access Control in Wireless LANs IEEE 802.11 DFWMAC and ETSI RES 10 Hiperlan. In: *Mob. Netw. Appl.* 2 (1997), Juni, Nb. 1, P. 55–67. – URL `http://dx.doi.org/10.1023/A:1013255927445`. – ISSN 1383-469X

[183] Wolf, Marko ; Daly, PW: *Security engineering for vehicular IT systems.* Springer, 2009

[184] Wolf, Marko ; Weimerskirch, André ; Paar, Christof: Secure in-vehicle communication. In: *Embedded Security in Cars* (2006), P. 95–109

[185] Wolf, Marko ; Weimerskirch, André ; Wollinger, Thomas: State of the Art: Embedding Security in Vehicles. In: *EURASIP Journal on Embedded Systems* 2007 (2007), Nb. 1, P. 074706. – URL `http://dx.doi.org/10.1155/2007/74706`. – ISSN 1687-3963

[186] Wu, X. ; Miucic, R. ; Yang, S. ; Al-Stouhi, S. ; Misener, J. ; Bai, S. ; Chan, W. h.: Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System. In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sept 2014, P. 1–7. – ISSN 1090-3038

[187] Wunder, G. ; Jung, P. ; Kasparick, M. ; Wild, T. ; Schaich, F. ; Chen, Y. ; Brink, S. T. ; Gaspar, I. ; Michailow, N. ; Festag, A. ; Mendes, L. ; Cassiau, N. ; Ktenas, D. ; Dryjanski, M. ; Pietrzyk, S. ; Eged, B. ; Vago, P. ; Wiedmann, F.: 5GNOW: non-orthogonal, asynchronous waveforms for future mobile applications. In: *IEEE Communications Magazine* 52 (2014), February, Nb. 2, P. 97–105. – ISSN 0163-6804

[188] YE, Ning ; Zhu, Yan ; WANG, Ru-chuan ; Malekian, Reza ; Qiao-min, Lin: An Efficient Authentication and Access Control Scheme for Perception Layer of Internet of Things. In: *Appl. Math* 8 (2014), Nb. 4, P. 1617–1624

[189] Yuan, Eric ; Tong, Jin: Attributed Based Access Control (ABAC) for Web Services. In: *Proceedings of the IEEE International Conference on Web Services* IEEE Computer Society (Organ.), 2005, P. 561–569

[190] Zanella, Andrea ; Bui, Nicola ; Castellani, Angelo ; Vangelista, Lorenzo ; Zorzi, Michele: Internet of things for smart cities. In: *IEEE Internet of Things journal* 1 (2014), Nb. 1, P. 22–32

[191] Zax, David: *Many Cars Have a Hundred Million Lines of Code*. `https://www.technologyreview.com/s/508231/many-cars-have-a-hundred-million-lines-of-code/`. December 2012

[192] Zhang, K. ; Mao, Y. ; Leng, S. ; He, Y. ; ZHANG, Y.: Predictive Offloading in Cloud-Driven Vehicles: Using Mobile-Edge Computing for a Promising Network Paradigm. In: *IEEE Vehicular Technology Magazine* 12 (2017), June, Nb. 2, P. 36–44. – ISSN 1556-6072

[193] Zhou, Minqi ; Zhang, Rong ; Zeng, Dadan ; Qian, Weining: Services in the cloud computing era: A survey. In: *Universal Communication Symposium (IUCS), 2010 4th International* IEEE (Organ.), 2010, P. 40–46

[194] Zimmermann, Hubert: OSI reference model–The ISO model of architecture for open systems interconnection. In: *IEEE Transactions on communications* 28 (1980), Nb. 4, P. 425–432