



D2.1

Report on Detailed Specification of Use Cases

Topic	SU-ICT-01-2018 - Dynamic countering of cyber-attacks
Project Title	Artificial Intelligence-based Cybersecurity for Connected and Automated Vehicles
Project Number	833611
Project Acronym	CARMEL
Contractual Delivery Date	M06
Actual Delivery Date	M06
Contributing WP	WP2
Project Start Date	01/10/2019
Project Duration	30 Months
Dissemination Level	Public
Editor	FICOSA
Contributors	I2CAT, T-SYS, ATOS, ALTRAN, 8BELLS, UBIWR, CLS, GFX, SID, 0INF, UCY, UPAT, AVL, PANA

Version	Date	Remarks
1.0	15/11/2019	Table of Content Ready
2.0	20/12/2019	Initial Version Ready
3.0	25/01/2020	Partners Contributions Received
4.0	10/02/2020	First Complete Draft Ready
5.0	26/02/2020	SAB and Internal Review Received
6.0	05/03/2020	Technical Manager Review Received
7.0	11/03/2020	Final Version

DISCLAIMER OF WARRANTIES

This document has been prepared by CAMEL project partners as an account of work carried out within the framework of the contract no 833611.

Neither Project Coordinator, nor any signatory party of CAMEL Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied,
 - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
 - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the CAMEL Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

CAMEL has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 833611. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

DISCLOSURE STATEMENT

"The following document has been reviewed by the CAMEL External Security Advisory Board as well as the Ethics and Data Management Committee of the project. Hereby, it is confirmed that it does not contain any sensitive security, ethical or data privacy issues."

Table of Contents

List of Figures.....	6
List of Tables.....	8
List of Acronyms	9
Executive Summary	11
1 Introduction	12
1.1 Stakeholders	13
1.2 Definitions	14
1.2.1 Role of AI/ML in the Context of CARMEL	14
1.2.2 Overview of CARMEL Anti-Hacking Solution: General Architecture and Functionality ..	16
2 Pillar 1 - Autonomous Mobility	21
2.1 Context.....	21
2.2 Scenario Description	23
2.2.1 Physical Adversarial Attacks.....	23
2.2.2 Attack on the Camera Sensor.....	27
2.3 Data Collection/Selection Methodology	31
2.3.1 Synthetic Dataset for Traffic Signs	31
2.3.2 Publicly Available Dataset for Traffic Signs	32
2.3.3 Publicly Available Datasets Featuring the Raw Sensor Camera and LiDAR Data	33
2.4 Use of Artificial Intelligence and Machine Learning	34
2.4.1 Physical Attack on Traffic Signs	34
2.4.2 Attack on Camera Sensor.....	35
2.5 Validation Methodology.....	37
2.5.1 Role of Simulation in the Autonomous Mobility Pillar	38
2.5.2 Comparison of Physical and Simulation Environment.....	38
2.5.3 Use of Simulation Framework within Pillar 1	39
2.5.4 Simulation Tool	40
2.6 Use of the Anti-Hacking Platform.....	42
2.7 Functional Requirements	44
3 Pillar 2 – Connected Mobility	47
3.1 Context.....	47
3.2 Scenarios Description	50
3.2.1 Location Spoofing Attack	50
3.2.2 Attack on the V2X Message Transmission	52
3.2.3 Tamper Attack of Vehicle's OBU	53
3.3 Enabling Infrastructure and Overview of Cyberattacks	55
3.3.1 Cooperative Cars	57
3.3.2 V2X Infrastructure	59
3.4 Data Collection and Selection Methodology	64
3.4.1 Location Spoofing Attack	64

3.4.2	Attack on the V2X Message Transmission	66
3.5	Use of Artificial Intelligence and Machine Learning	67
3.5.1	Location Spoofing Attack	67
3.5.2	Attack on the V2X Message Transmission	71
3.6	Validation Methodology	72
3.6.1	Location Spoofing Attack	73
3.6.2	Attack on the V2X Message Transmission	74
3.6.3	Tamper Attack of Vehicle's OBU	76
3.7	Use of the Anti-Hacking Device	77
3.8	Functional Requirements	80
4	Pillar 3 - Electromobility	82
4.1	Context	82
4.2	Scenarios Description	84
4.2.1	Smart Charging Abuse	84
4.2.2	EV Scheduling Abuse	85
4.3	Enabling Infrastructure	86
4.3.1	Platform Infrastructure	86
4.4	Data Collection and Selection Methodology	89
4.4.1	Meter Values and CDRs	89
4.4.2	Smart Charging	90
4.4.3	GreenFlux Database	91
4.5	Use of Artificial Intelligence and Machine Learning	93
4.5.1	Smart Charging Abuse Scenario	93
4.5.2	EV Scheduling Abuse Scenario	96
4.6	Validation Methodology	97
4.6.1	Smart Charging Abuse	97
4.6.2	EV Scheduling Abuse	98
4.6.3	Scenario Validation Overview	99
4.7	Use of the Anti-Hacking Device	101
4.8	Functional Requirements	103
5	Non-Functional Requirements	104
6	Conclusion	105
	References	107
	Annex 1	111

List of Figures

Figure 1: CAMEL Project Structure	12
Figure 2: Machine Learning Pipeline	17
Figure 3: Anti-hacking Device Software Architecture	17
Figure 4: Anti-hacking Device Hardware	18
Figure 5: Conversion of Tensorflow model for use with Edge TPU	19
Figure 6: Society of Automotive Engineers (SAE) Automation Levels	21
Figure 7: Appearance perturbations on Traffic Signs	23
Figure 8: Adversarial Attacks on Traffic Sign	24
Figure 9: High-level Description of the Roles in the Scenario	26
Figure 10: Targeted Perturbation Attack (either physical or digital) on the Camera Feed	28
Figure 11: Concept of using ML in the Attack on the Camera Sensor	35
Figure 12: Adversarial training	36
Figure 13: Comparison of the physical and simulated process starting with the sensors up to the anti-hacking device software (AHDS)	38
Figure 14: Simulation environment workflow and AHDS deployment in physical environment	39
Figure 15: Data generation acceleration and diversity increase by creating individual functions within the simulation tool	40
Figure 16: From Training to Deployment in the Autonomous Car	43
Figure 17: Integration of anti-hacking device with autonomous vehicle	43
Figure 18: Training and validation in the simulation environment	44
Figure 19: Overview of the attack surface of the Connected Mobility Use Case	48
Figure 20: Attack scenarios contemplated in the connected mobility pillar	50
Figure 21: A possible implementation for the location spoofing attack	51
Figure 22: Block Diagram of the satellite-based location integrity check application	52
Figure 23: Security techniques at the OBU level	53
Figure 24: The layered architecture of OBU security	54
Figure 25: Cooperative car equipment overview	57
Figure 26: Overview of Secure Multi-Technology V2X Telecommunications Infrastructure	59
Figure 27: Certificate chain in CAMEL PKI infrastructure	60
Figure 28: CAMEL PKI Infrastructure	61
Figure 29: ETSI MEC Framework	61
Figure 30: Interoperability between radio technologies	63
Figure 31: Interoperability between regions of interest	63
Figure 32: Vehicle Bicycle Model Representation exploiting specific on-board measurements, i.e., steering angle sensor, yaw rate gyroscope and wheel speed sensor	64
Figure 33: SOOP Implementation	65
Figure 34: Location Spoofing Detection Architecture	67
Figure 36: Example of VANET	68
Figure 36: Random VANET topology	69
Figure 37: High-level architecture of fusion of GPS and LIDAR/RADAR data	70
Figure 38: Data flow for the attack on V2X message transmission scenario	72
Figure 39: Structure of secured and unsecured frames	74
Figure 40: Tamper attack of vehicle's OBU sequence flow	76
Figure 41: Use of anti-hacking device for GPS spoofing attack detection	78
Figure 42: Use of anti-hacking in simulation environment	78
Figure 43: Model.Connect Simulation setup for the Intrusion Detector	79
Figure 44: Step 2 implementation of the Model.Connect with intrusion detection	79
Figure 45 The increasing trend of EV deployment in different countries [98]	82
Figure 46: A high level depiction of the entities getting engaged in the smart charging scenario [96]	83
Figure 47: High-level architecture of the GreenFlux Charging Solution	86
Figure 48: Possible connections between CP and GSOP	87
Figure 49: Charge Point System Architecture	88
Figure 50: Message flows between involved parties	90
Figure 51: Smart charging message flow	91
Figure 52: An overview of the Abuse Detection workflow	94

Figure 53: An example of standard deviation applied on normal distribution	95
Figure 54: A graphical representation of density-based functions	95
Figure 55: Overview of Isolation Forest Algorithm output	96
Figure 56: Information flows for Synchronous Algorithm at iteration $k \geq 1$. Left: Aggregator broadcasts time slot ordering to EVs. Right: Summation of intermediate charging profiles are forwarded to aggregator	98
Figure 57: Block Diagram describing optimal decentralized algorithm	99
Figure 58: Use of anti-hacking device in electromobility use case	102
Figure 59: Autonomous Driving Solution Scheme approximated as in IoT	105
Figure 60: Model.CONNECT ADAS toolchain	112
Figure 61: VSM configuration in Model.CONNECT	113
Figure 62: VSM vehicle simulation in testbed mode	114
Figure 63: VTD simulation video	114
Figure 64: AVL DRIVE	115
Figure 65: Web UI	116
Figure 66: Information exchange flow	117

List of Tables

Table 1: CARMEL Project Stakeholders	14
Table 2: List of CARMEL Scenarios on Autonomous Mobility	23
Table 3: The Traffic Sign Attack Scenario Definition	25
Table 4: Overview of the Physical Adversarial Attack Scenario	27
Table 5: The Attack on the Camera Sensor Scenario Definition	29
Table 6: Overview of the Camera Sensor Attack Scenario	30
Table 7: Offline Variants of the Data Records	31
Table 8: Publicly Available Dataset for Traffic Signs	33
Table 9: Publicly Available Dataset for the raw Sensor Camera and LiDAR Data	34
Table 10: Some sensors included in CARLA and their configurable parameters	41
Table 11: Validation methods of Physical Adversarial Attack	41
Table 12: Validation methods of Attack on Camera Sensor	42
Table 13: Overview of attacks in the connected mobility domain	49
Table 14: Overview of scenarios to be examined in Pillar 2	50
Table 15: OBU Tampering Attack characteristics	55
Table 16: Overview of Malicious Attacker types	56
Table 17: Overview of building blocks for the Connected Mobility pillar	57
Table 18: Data sources for ML algorithm training	66
Table 19: Validation methods of scenarios in pillar 2	73
Table 20: Overview of Location Spoofing Attack scenario	74
Table 21: Overview of the Attack on the V2X Message Transmission Scenario	76
Table 22: Overview of Tamper attack of vehicle's OBU Scenario	77
Table 23: Description of the entities engaging the smart charging scenario	84
Table 24: An overview of the ChargingPoint table of the GreenFlux's database	92
Table 25: An overview of the Charge Detail Records table of the GreenFlux's database	92
Table 26: An overview of the Connections table of GreenFlux's database	93
Table 27: An overview of the Meter Values Table of GreenFlux's database	93
Table 28: Overview of Validation for Pillar 3	99
Table 29: Overview of the Smart Charging Abuse Scenario	100
Table 30: Overview of the EV Scheduling Abuse Scenario	101

List of Acronyms

ADAS	Advanced Driver Assistance System
AES	Advanced Encryption Standard
AI	Artificial intelligence
API	Application Programming Interface
APP	Application
ASIL	Automotive Safety Integrity Level
CAM	Cooperative Awareness Message
CAN	Controller Area Network
C-ITS	Cooperative Intelligent Transport Systems
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
CP	Charging Point
CPO	Charging Point Operator
CPS	Cyber-physical System
CSE	Cryptographic Services Engine
DAC	Discretionary Access Control
DENM	Decentralized Environmental Notification Message
DRM	Digital Rights Management
DSO	Distribution System Operator
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECU	Embedded Control Unit
EKMS	Electronic Key Management System
EN	European Norms
ENISA	European Union Agency for Cybersecurity
eSE	embedded Secure Element
eNVM	embedded Non-volatile Memory
ETSI	European Telecommunications Standards Institute
EV	Electrical Vehicle
FOTA	Firmware Over-the-Air
Gateway	A VCU that connects multiple networks within a car
GDPR	General Data Protection Regulation
GNSS	Global Navigation Satellite System
HMI	Human Machine Interface
HSM	Hardware Security Module
HW	Hardware
IC	Integrated Circuit
IoT	Internet-of-Things
IP	Intellectual Property
ISO	International Standardization Organization
IT	Information Technology
ITS	Intelligent Transport Systems
IVS	In-Vehicle System
JTC1	ISO/IEC Joint Technical Committee 1 Information Technology
LAN	Local Area Network
LTE	Long-term Evolution (4th generation Mobile Internet)
M2M	Machine-to-Machine
MAC	Media Access Control
MCU	Micro Controller Unit
MEC	Multi-access Edge ComputingML
NN	Neural Network
NWI(P)	New Work Item (Proposal) standardization
OBU	On-Board Unit
OCPP	Open Charge Point Protocol
OEM	Original Equipment Manufacturer
OS	Operating System
	Machine Learning

OSS	Operating System Security
OTA	Over-The-Air
PDU	Protocol Data Unit
PHY	PHYsical Layer
PIA	Privacy Impact Assessment
PKI	Public Key Infrastructure
RBAC	Role-Based Access Control
ROP	Return-oriented programming
RSA	Asymmetric Encryption Algorithm
RSU	Roadside Unit
RTK	Real-Time Kinematics
RTOS	Real Time Operating System
SC	Sub-Committee
SDO	Standardization Organization
SE	Secure Element
SIL	Safety Integrity Level
SOOP	Signals Of Opportunity
SSL	Secure Sockets Layer
SW	Software
TC	Technical Committee
TCB	Trusted Computing Base
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TR	Technical Report
TRL	Technology Readiness Level
TS	Technical Specification
UC	Use Case
UNECE	WP.29 World Forum for Harmonization of Vehicle
UWB	Ultra-Wide Band
VCU	Vehicle Control Unit
VDS	Vehicle and Driver Status
VPN	Virtual Private Network
V2X	Vehicle-to-X, where X stands for either Vehicle or Infrastructure
WAVE	Wireless Access in Vehicular Environments
WG	Working Group
WP	Work Package

Executive Summary

This deliverable will report in detail on the use cases to be deployed within the project. It will identify main limitations of current technologies and specify actors, scenarios, the flow of actions and measurable improvements over the current state-of-the-art.

D2.1 consists of five sections, section 1 introduces the CARMEL stakeholders and provides some useful definitions useful to understand better the activities of the project. That includes the role of Machine Learning (ML) and Artificial Intelligence (AI) in the context of the project as well as an overview of the CARMEL anti-hacking solution. Section 2 focuses on the first pillar of the project, autonomous driving. After introducing the context of autonomous driving, this section focuses on some selected scenarios for the project. Section 2 provides details about data collection and selection methodology, the use of ML and AI and the role of anti-hacking solution in the framework of selected scenarios. At the end, section 2 presents some related functional requirements related to pillar 1. In the same way, section 3 provides details about the second pillar of the project, connect cars. Scenario description, enabling infrastructure, data collection and selection methodology, use of ML and AI in the context of selected scenarios, and the use of anti-hacking solution are the topics presented in section 3. This section also provides some technical requirements related to the connected car pillar. Section 4 follows the same path for the electromobility case, pillar 3. Scenario description, enabling infrastructure, data collection and selection methodology, the role of ML and AI in the context of selected scenarios as well as the role of anti-hacking solution are presented under section 4. Similar to the two previous sections, section 4 ends with some functional requirements related to the electromobility. Section 5 concludes the document and provides some extra interesting cybersecurity related scenarios.

1 Introduction

Vehicles are becoming smarter and “greener” through connectivity and artificial intelligence, and cybersecurity is emerging as a new concern. CARMEL’s goal is to proactively address modern vehicle cybersecurity challenges applying advanced Artificial Intelligence (AI) and Machine Learning (ML) techniques, and also to continuously seek methods to mitigate associated safety risks.

D2.1 will define in detail the selected use cases that are going to be validated by CARMEL. D2.1 will review the already published state of the art, analyse their limitations, and at the end propose the CARMEL solution to fill the identified gaps. This deliverable will also collect the use case requirements in terms of CARMEL showcases. As such, we will formulate this input as functional requirements to drive the definition and design of the CARMEL architecture. T2.1 will also provide the basis for the definition of an evaluation matrix.

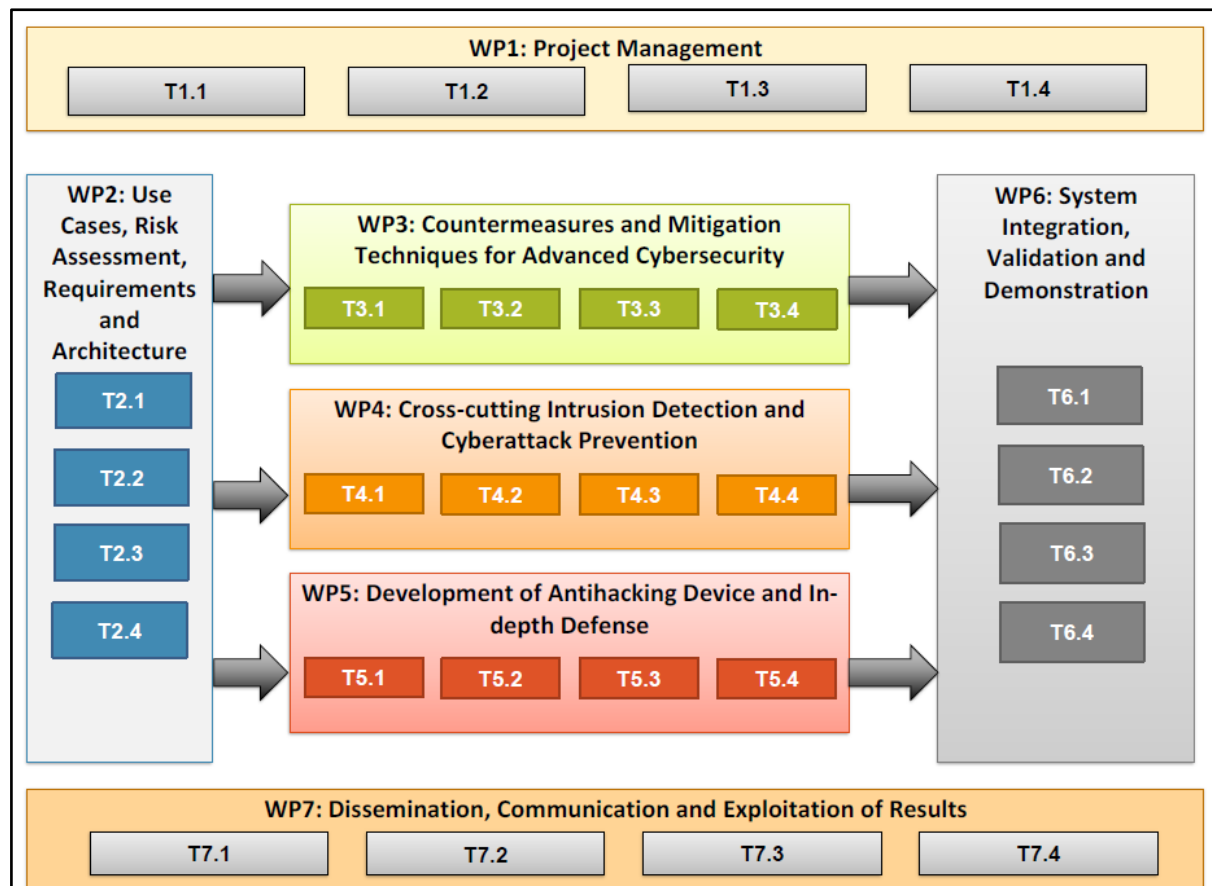


Figure 1: CARMEL Project Structure

D2.1 has been organized around three main pillars of the future mobility:

- Pillar 1 – Autonomous Mobility
- Pillar 2 – Connected Mobility
- Pillar 3 – Electromobility

At each pillar, first we will review the context, i.e. give a general overview of the future mobility form while highlighting the security, safety and privacy protection requirements. Next, we will present the scenarios that CARMEL will specifically focus on. Then, we will present the requirements of the scenario. That includes information about the data collection/selection methodology as well as the use of AI/ML techniques. In the validation section, we will identify the testing and validation methodologies of the scenarios. Some of the scenarios will be validated using the simulation tools, while others will be showcased as CARMEL real-life demonstrations. As a main innovation part of CARMEL, the project

promises to deliver an Anti-hacking solution. The role of Anti-hacking solution will be also highlighted for each pillar.

Before moving on into CAMEL pillars, first let's focus on the D2.1 stakeholder and some definitions.

1.1 Stakeholders

CAMEL considers the needs of the entire cyber-security and automotive value chains, ranging from: (a) the general public that uses digital communications and future automotive products, (b) the cyber-security solution providers, AI and ML methods developers, etc., (c) the infrastructure providers, represented by telecommunication infrastructure providers (telecom operators, ISPs), cloud service providers, and organisations with small, medium and large scale infrastructures that require a low-cost cyber-security investment, (d) vehicle manufacturing industry, i.e. automotive companies, equipment, system and solution providers for automotive industry, etc. CAMEL further considers the needs of policy makers in EU & Member States for informed decisions regarding the security of modern infrastructures for future vehicle industry. Additional benefits are considered in the case for standardisation, other special interest groups, open source communities and researchers/academics. CAMEL stakeholders are listed on Table 1.

Target Group	Main Players	Impact/Market Opportunities
Technology Suppliers	Automotive suppliers and partners, automotive integrators, vehicle engineering companies, vehicle manufacturers, charging station component suppliers, manufacturers and integrators, in-vehicle charging infrastructure component suppliers, manufacturers and integrators, payment systems providers, ICT providers, Telematics/data management companies, cybersecurity companies	Significant cost reduction due to enhanced security features Improved situational awareness, decision support and remediation Penetration testing methods developed / tested over intelligent and modern testbeds
Service providers	EV charge sellers, local EV charge service companies, charging stations owners, specialised consulting companies, mobility service providers, automotive dealers and the aftermarket sector, insurance companies	New cyber security services / products Simplification of their entry to new markets Development of business models for cyber-security services / products in future networks, especially for the automotive vertical
Operators	Telecom operators, road operators, charging station network operators, logistics operators	
Research, Academia & Open Source Communities	Researchers and academics from universities, research centres and R&D industry departments, open source communities	Novel detection methodologies Open access to an operational environment that allows validation of situational awareness & cyber-security advances in an environment closely resembling actual operations Ensuring research integrity & credibility by providing medium scale testing Extension to available open source solutions, maturing them in terms of security Contributions to Open Source Threat Intelligence Platform &

		Open Standards For Threat Information Sharing (MISP)
Authorities	Public authorities on mobility and cybersecurity, policy makers, regulators, national, local and international governmental agencies (ENISA), cities, GEAR2030, ministries, judiciary systems, national security agencies, national data protection authorities	Novel improvements and recommendations
Standards organisation bodies	Institute of Electrical and Electronics Engineers (IEEE), National Institute of Standards and Technology (NIST), Society of Automotive Engineers (SAE), International Organisation of Standardisation (ISO), AUTOSAR, ETSI (TC Cybersecurity Group, ITS, MEC), 3GPP, CEN/CENELEC	Service and data protection methods, specifications for cyber-security aware services
Networks & Platforms	Automotive	5GAA, Car to Car – Communication Consortium (C2C-CC), Connected Motorcycle Consortia (CMC), Auto-ISAC, Auto Alliance, Global Automakers
	Mobility	ERTICO, ALICE
	Cybersecurity	ECSO, CSIRTs Network
	Telecommunications	5GPPP, 5G Infrastructure Association (5GIA)
	ICT	BDVA
End-users	General public, commercial fleets, public fleets, drivers, passengers, society, related associations, fleet customers, vehicle customer	Better overall situational awareness and cybersecurity protection in future vehicles Improved protection of systems and data even in cases where endpoints are not sufficiently fortified

Table 1: CAMEL Project Stakeholders

1.2 Definitions

Before moving to the main part of this document which is the detailed explanation of CAMEL pillars and the associated use cases, in this section, we would like to provide some definitions and guidelines that are useful to a better understanding of the CAMEL solution and its role on three future mobility forms.

CAMEL focuses on an artificial intelligence-based cybersecurity solution in the context of the connected and automated vehicles. Therefore, it is worth to first focus on the role of AI/ML in the context of the project.

1.2.1 Role of AI/ML in the Context of CAMEL

The goal of CAMEL is not the improvement of general pattern classification methods, but the project tries to demonstrate the use of AI/ML-based classification methods in detection and possibly mitigation

of dynamic cyber-attacks on the system/data in the context of future mobility. Therefore, in the following paragraphs, some general aspects of AI/ML are defined while the necessity for implementing simulation is explained.

In many applications in mobility, particularly those addressed in CARMEL, due to the diversity of the data streams the patterns of interest cannot be reliably classified by explicit programming. In such cases, if sufficiently large amounts of example (training) data are available or feasible to obtain, the most usual approach is to employ ML (typically using Neural Networks - NNs).

In CARMEL we focus on detecting “anomalies” in data streams using ML. In this context, the pattern class “anomaly” may infer “attack”. The classification ideally works on the basis of:

- The well-defined nominal state of the systems in terms of its typical data stream patterns
- More or less known patterns of attacks that - ideally but are not guaranteed to - cause detectable changes in the data streams.

Mere deviation of data stream patterns (inferring deviation of the system away the nominal state) may not deliver sufficiently accurate classification (false+/false-). Therefore, the more anomaly patterns are known beforehand, the better. In other words, labelled training data is necessary combined with a ML method that is suitable for supervised learning.

Typically, “very large” amounts of data are required to train a NN. This is the downside of avoiding the explicit modelling of the problem. The term “very large” is highly application dependent and up to *a-posteriori* and subjective judgement. In bolder terms: beforehand, one never knows how much and which data is necessary to produce a satisfactory solution.

When we talk about data, for sure it covers a wide range of option, from real-world data to synthetic data. In CARMEL’s context it is clear from the beginning, that even for a set of selected and limited scenarios/use cases it is infeasible to collect sufficient volume of real-world data for:

- Characterisation of the system nominal state
- Generation of sufficient volume of real(istic) anomaly (attack) patterns to train a NN up to the point that it can provide an acceptably low rate of false+/false- for all realistic scenarios within the use-case.

This is due to the fact that we are at the dawn of the next generation mobility and the required volume of data collected from the proper sources, in particular related to the cybersecurity issues, are not much available. In addition, compared to synthetic generation, the recording of real/physical data is very personnel-, time- and cost-intensive. For example, section 2.3.1. lists publicly available image sources of traffic scenarios. It is part of the tasks in CARMEL to analyse the suitability of these data bases. Such databases are produced by different teams, sensors setups (e.g. focal lengths and other distortions, backgrounds, country specifics, etc.) and subsequent processing. It means they might not exactly serve the purpose of CARMEL.

Therefore, CARMEL decided to employ a simulation tool whose purpose is twofold: first for evaluation related tasks and second to generate (additional) synthetic training data of “appropriate realism” in both nominal and attack patterns. Furthermore, intelligent combination methods will be developed to combine real and synthetic data to an even larger set of training data that is supposedly more realistic and captures elements of the “real thing” in a better way than pure simulation/synthetic data can. Note: The generation of synthetic data requires the actual human *understanding* of the problem, i.e. model assumption on the patterns, either attack or nominal. This is a task that CARMEL will carry out in the future activities to produce proper data sets and ultimately suitable AI/ML solutions. Nevertheless, it is worth to note that the resulting realism of the synthetic data of course depends on:

- The capability of the simulation tool,
- The human effort and understanding invested in creating the synthetic environments intelligently and automatically,
- Available computing power,
- The necessary degree of realism to demo the core innovation (AI-driven alarm system).

CAMEL will work on these points to provide the best possible results. It is also worth to note that, training can be done on synthetic data [3] or on hybrid data [4][5]. A comparison of the outcome of both can also reveal the quality (invested human intelligence and problem understanding) of the combination method:

Real data + Synthetic data => Hybrid data

where the “+” symbolises a potentially complex mechanism that combines both data types. It might be a good approach to develop a realistic CAMEL solution empowered by AI/ML techniques to detect and mitigate cyber-attacks. In the case of image classification for example, the realism of synthetic scenario images is often obvious to judge by eye, less so in communication patterns. Remark: The label “Realism” in above context implies model assumptions and is a subjective property. It is exactly this above-mentioned human understanding of the problem at hand that defines the previously used term “similarity” between data (sets). Therefore, the similarity between, for example, a) publicly available data sets and b) the real-world situation in CAMEL demos are not mathematically deducible but require empiricism and experience.

Application of ML/AI implies the hope that the nature of the NN of choice (or whichever approach) is capable of some generalisation to input data that is not identical to the training data set. Anomaly detection via measurement of complex systems (or stochastic input data) that deny accurate model assumptions by huge or even infinite state spaces are a challenge. One can never tell if the most extensive measurement or simulation will ever capture sufficiently the interesting features of the system behaviour, overfitting and underfitting may go undetected, or the NN architecture may not provide the right degrees of freedom. To achieve at least some level of confidence the available data is split into a training and test set (more than one partitioning possible) and are also interchanged. A result may be that the data is not representative w.r.t. the features of interest. Which these features are is often not obvious to tell since the system cannot be modelled explicitly. CAMEL will put efforts to find the best trade off to obtain a reasonable result, thus, verifies the possibility of employing AI/ML solutions on the future automotive sector.

1.2.2 Overview of CAMEL Anti-Hacking Solution: General Architecture and Functionality

The CAMEL anti-hacking solution is an important part of the project innovation. In this section, we have a deeper look on the general architecture and functionalities of it.

The anti-hacking device is a physical controller that is integrated into the car and acts as an attack detection device. In the Autonomous Mobility scenario its task is to run pre-trained ML models that work on the sensor data to detect anomalies that might point to malicious attacks. Additionally, the anti-hacking solution might be used for different functions in the context of the CAMEL project, i.e. if needed it can ensure security for an embedded application platform. In this case, the software layer of the solution might be employed only. Further details about this approach will be presented in the rest of this document.

The anti-hacking device is connected to the busses in the car carrying the sensor data. It passively monitors the bus traffic (e.g. CAN bus frames) and extracts the raw sensor data.

Figure 2 shows the ML pipeline where raw data, e.g. from the CAN bus is pre-filtered and aggregated to make it suitable for the following machine learning stage to detect threats and attacks. Any security-relevant events are then forwarded to the visualization and mitigation components in the car.

The ML knowledge base (model) is pre-loaded into the anti-hacking device. The model will have been created offline on a more powerful system based on simulated and real-world training data (on the above section we discussed touched upon this subject. In the following activities of CAMEL more inputs in this regard will be produced and reported).

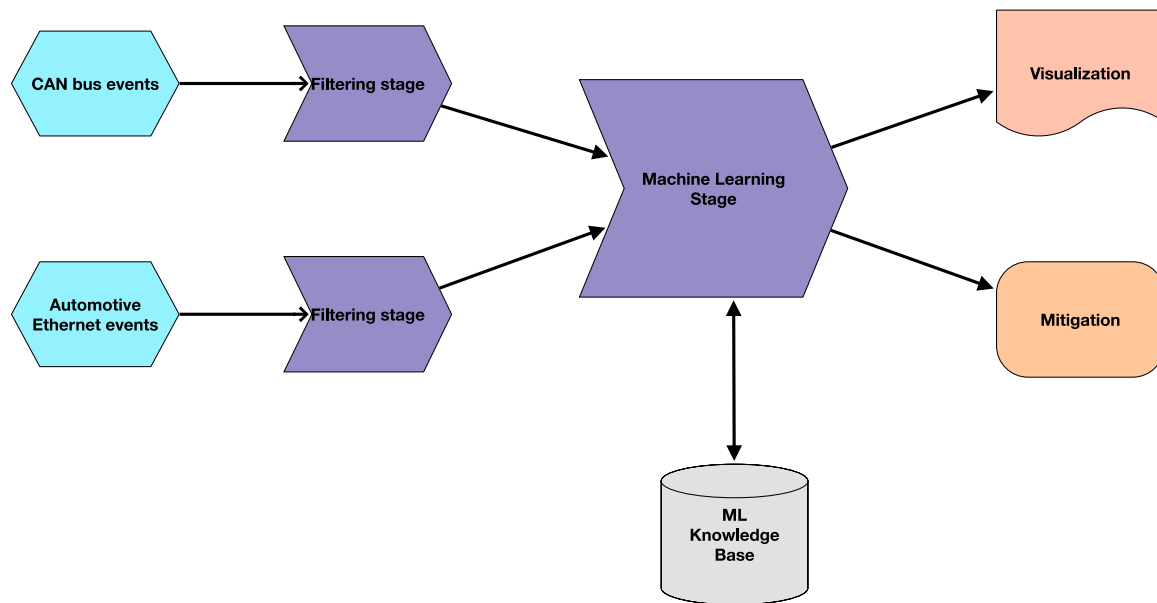


Figure 2: Machine Learning Pipeline

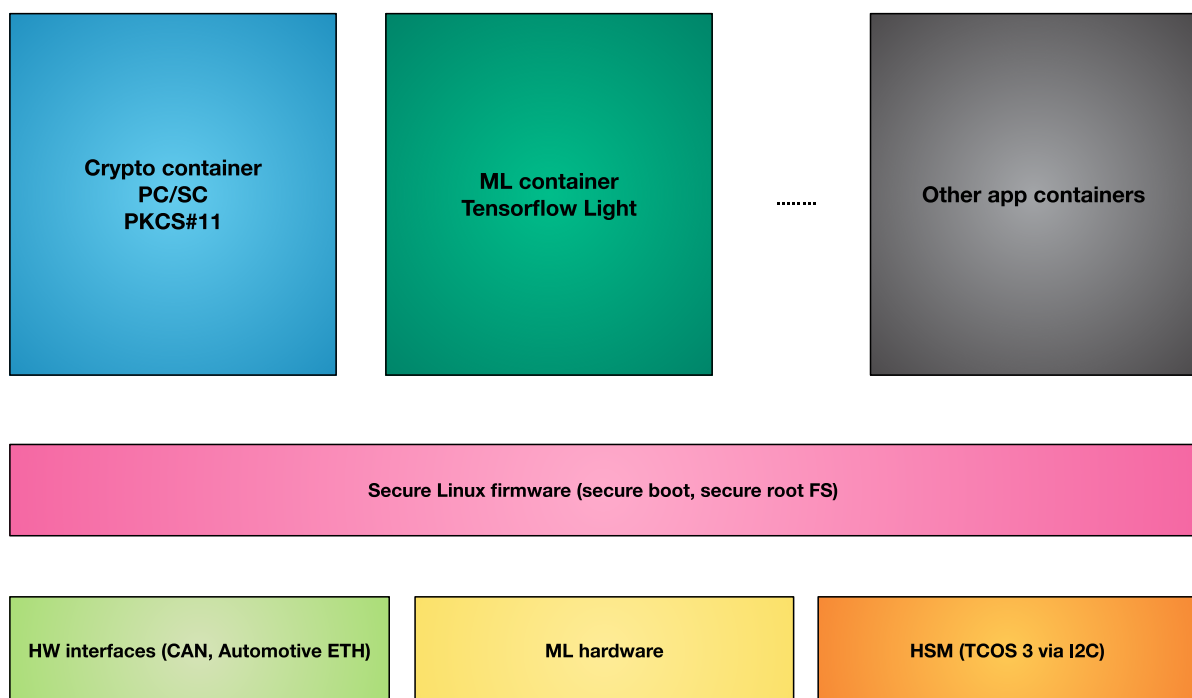


Figure 3: Anti-hacking Device Software Architecture

Figure 3 shows an overview of the software and hardware architecture of the anti-hacking device. From bottom-up the following components make up the anti-hacking devices:

- **HW Interfaces:** The anti-hacking device will be connected to the in-car systems via appropriate interfaces used in the automotive industry such as the CAN bus or Automotive Ethernet connections. For integration into development and simulation frameworks standard Ethernet will also be supported.
- The anti-hacking device will also support machine learning (ML) hardware. Since the anti-hacking device is based on the Coral Dev Board the Tensorflow Lite Processing Unit (TPU) is

the hardware element to support ML. For a development and simulation configuration the Coral USB Accelerator will also be supported.

- HSM (hardware security module): To provide security-related functions of the anti-hacking device the hardware will integrate a Secure Element or HSM in the form of a TCOS (Telekom Card Operating System) embedded smartcard module that supports secure storage of private keys and different cryptographic operations.
- The anti-hacking device itself is based on an NXP Freescale i.MX8 processor that supports security functions such as hardware-assured boot.
- On this security hardware runs a Yocto-based firmware layer (a Linux embedded meta distribution).
- On top of this firmware substrate Docker-based application-specific containers can be loaded. Out-of-the box there will be crypto containers supporting the security functions of the anti-hacking device. ML workloads will be also be implemented as containers that have access to the underlying ML hardware as well as the crypto functions exported by the crypto container.
- The anti-hacking device could also act as a secure run-time environment for other functions as needed by the different use cases.

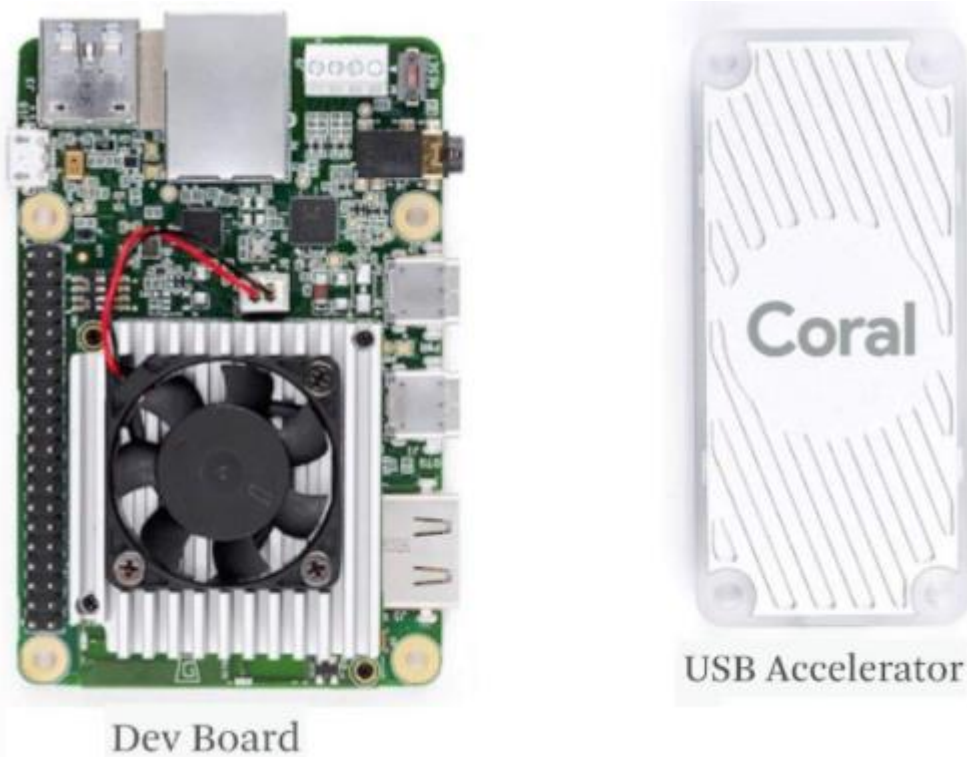


Figure 4: Anti-hacking Device Hardware

Figure 4 shows a picture of both the final target hardware - the Coral Dev Board¹ - as well as the solution for development and simulation - the USB Accelerator.

The Coral Dev Board has the following hardware specifications:

- CPU: NXP i.MX 8M SOC (quad Cortex-A53, Cortex-M4F)
- GPU: Integrated GC7000 Lite Graphics.
- Coprocessor: Google Edge TPU.
- RAM: 1GB LPDDR4.

¹ <https://coral.ai/docs/dev-board/get-started/>

- Flash memory: 8GB eMMC.
- Connectivity: Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz) Bluetooth 4.1.
- Dimensions: 48 x 40 x 5mm.

The i.MX8 SOC includes advanced security features such as HAB (high-assurance boot) and CCAM (Cryptographic Accelerator and Assurance Module) that will support the security features of the Anti-hacking device. The firmware for the i.MX8 SOC will be created using the Yocto environment which is an industry-standard toolkit to create custom embedded firmware images in a reproducible manner. Our build process will support signed bootloaders and Linux kernel in order to prevent tampering with the anti-hacking device software and configuration.

The Coral Dev Board also has many connectivity options integrated on the board:

- Ethernet port (can be used for IP-based connections in a simulation and test environment, or to attach Automotive Ethernet adapters if needed)
- GPIO and I2C ports (used for connecting the HSM module, can be used for other purposes as well)
- USB port (used in the project to connect USB-to-CAN-bus converters)
- Wireless connectivity - Wi-Fi and Bluetooth

The Edge TPU processor integrated into the Coral Dev Board supports the execution of Tensorflow Lite models, performing 4 trillion operations (tera-operations) per second (TOPS), using 0.5 watts for each TOPS (2 TOPS per watt). The same Edge TPU is integrated into the USB Accelerator stick, so similar performance can be expected in the Anti-hacking Device simulation environment.

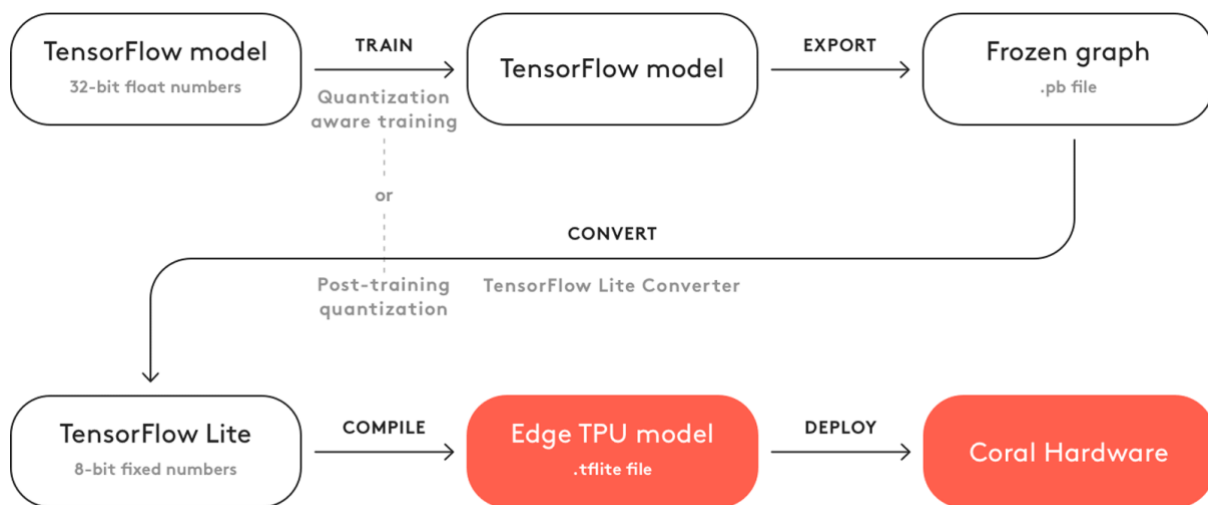


Figure 5: Conversion of Tensorflow model for use with Edge TPU²

Figure 5 shows how TensorFlow models created by a machine learning process (e.g. running in the cloud or on project hardware) can be converted for use with either Coral Dev Board or the Coral USB Accelerator.

The I2C ports of the Coral Dev Board will be used to connect an HSM (hardware security module) based on the TCOS (Telekom Card Operating System) specification to act as an embedded Secure Element (eSE) and security anchor for the Anti-hacking device. The HSM is meant to support the following functions:

- Authentication of the Anti-hacking device for remote provisioning and updates
- Provide support for other CARMEL use cases that need HSM functionality

² Source: [compile-workflow.png](#)

- Authentication of the anti-hacking device against central systems such as Automotive SOC (Security Operations Centre) for event reporting and alerting

Now that we have a better understanding of CARMEL definitions let's move on to the three mobility pillars targeted by the project.

2 Pillar 1 - Autonomous Mobility

2.1 Context

Automated driving systems were developed to automate, adapt and enhance vehicle systems for safety and improved driving. Most road accidents occur due to human error, and automated systems use input from sensors like video cameras to reduce human error by issuing driver alerts or controlling the vehicle. Such systems have become common in modern cars, with automobile manufacturers integrating these systems in their cars. There are six levels of automation as shown in Figure 6. When it comes to Advanced Driver Assistance Systems (ADAS), the highest level (5) corresponds to full automation where the automated functions control all aspects of the car, and the lowest level (0) where the driver controls all aspects of the car.

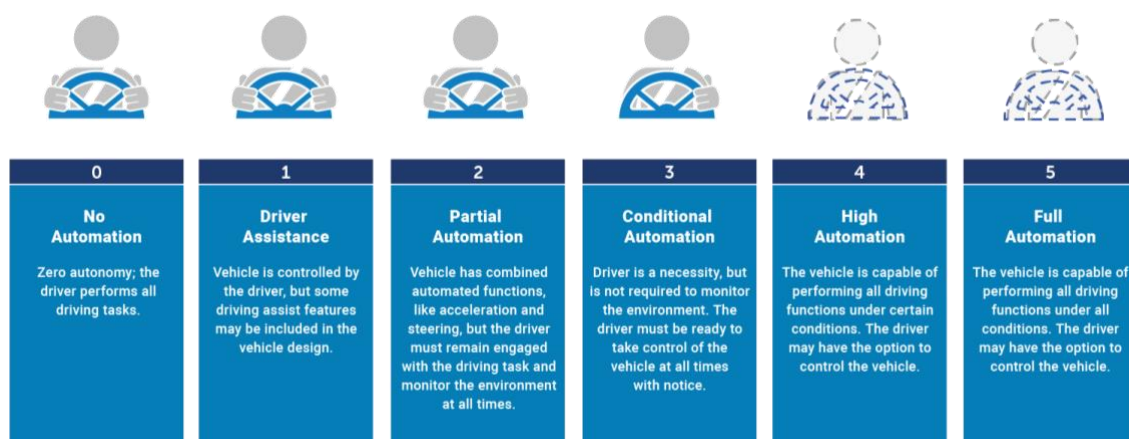


Figure 6: Society of Automotive Engineers (SAE) Automation Levels

Recently, these systems have attracted increased attention within academia, and the academic community has begun to investigate the systems' robustness to various attacks. Recent studies [6] [7] [8] showed that ADAS alerts and notifications can be spoofed by applying adversarial machine learning techniques to scene structural elements (e.g. traffic signs, objects, etc.)

Adversarial attacks seek small perturbations of the input causing large errors in the estimation by the perception modality. Attacking perception functions using adversarial examples is a popular way to examine the reliability of learning approaches for data classification [9]. The key to all such attacks is that the change to the image should be minor yet have a large influence on the output. Adversarial examples typically involve small perturbations to the image that are not noticeable by the human eye. The adversaries are shown to work even when a single pixel is perturbed in the image [6]. Although these attacks reveal limitations of deep networks, they are hardly replicated in real-world settings. For instance, it is rather difficult to change a scene such that one pixel captured by a camera is perturbed in a specific way to fool the network. However, recent work on [10] [11] demonstrate that adversarial examples can also work when printed out and shown to the network under different illumination conditions. [12] shows that adversarial examples can be 3D printed and are misclassified by networks at different scales and orientations. [13] constructs adversarial glasses to fool facial recognition systems. [14] show that stop signs can be misclassified by placing various stickers on top of them.

Apart from the adversarial attacks, which involve scene modifications on the physical layer, within the Autonomous driving, the vulnerability of the Perception Engine is also an important issue to address. CAMEL focuses on this point thanks to the perception engine contributed by Panasonic Automotive Europe. The Perception engine has to be secured against a variety of cyber-attacks at the sensors layer with the help of proper approaches for detecting the attacks and mitigating them.

In line with what is described above, CAMEL in the framework of autonomous mobility (pillar 1) believes the following scenarios presented in Table 2 are the most important cases to be addressed.

Note that the presented scenarios are selected based on the CARMEL consortium knowledge, available resources and showcasing capacity.

Description	
1	Adversarial attack on traffic signs: This is an attack on the physical layer. It assumes disturbance of the visual appearance of structural elements of the scene like the traffic signs. According to this attack, minor changes might be introduced, e.g.: stickers attached on the traffic signs in such a way that they might be marginally observable by the human eye but disturbing the scene perception output. The cyber-attack detection & mitigation engine will get as input the traffic sign topology from the perception engine and will assess the occurrence of cyber-attack.
2	Adversarial attack on lane/parking markings: As the above, this is also an attack on the physical layer. It is oriented towards distorting the appearance of lane/parking markings. The change could involve distortions in a multitude of appearance characteristics, e.g.: shape/length/colour. This attack should introduce minor changes in such a way that they could be marginally detectable by the human eye but finally affecting the output of the scene perception engine. The cyber-attack detection & mitigation engine should detect the occurrence of the cyber-attack and perform restoration in case that the restored version is derived with high confidence.
3	Attack on the Camera Sensor Layer: This scenario would involve a cyber-attack based on activating some malicious software which got installed during the software update process. Throughout this use-case the camera sensor could be attacked in a number of different ways, which could vary between adding noise lying on specific bands of the frequency spectrum/ introducing morphological deformations/ on the whole or parts of the image.
4	Attack on the Camera Sensor Layer by de-synchronizing the data: Throughout this scenario, the cyber-attack will be geared towards disturbing the association between the captured frames and the timestamp assigned to them. This will cause the failure of the perception engine, as all the architectural modules performing stochastic filtering on the scene observations will be affected by error. This use case should study the potential and the limitations of the cyber-attack detection and mitigation engine in assessing and recovering the failures.
5	Attack on the Camera Sensor by a remote agent: In addition to the aforementioned scenario, the cyber-attack detection and mitigation engine will be used to detect and mitigate the camera signal distortion in the case that a malicious remote agent interferes with the test vehicle by knowing the IP of the processing unit and sharing some erroneous data. More specifically, this use case will assume that the remote agent sends via V2X communication: time zone/ daylight related data in order some sensor parameters (e.g.: gain/exposure time) to be tuned accordingly.
6	Attack on the LiDAR sensor: Apart from the camera, cyber-attacks on the LiDAR sensor is another important issue. As in use cases 3-5, the attack will involve triggering malicious software through either a remote agent or some date-related software update process. The malicious software could distort multiple attributes of the LiDAR signal which could

vary by either adding noise to the measured data or changing arbitrarily some of the sensor configuration parameters (e.g.: scanning frequency).

Table 2: List of CAMEL Scenarios on Autonomous Mobility

2.2 Scenario Description

CAMEL scenarios on Autonomous Mobility can be classified into two big categories: physical adversarial attacks and attacks on the camera sensor. In this section, we will present them in more details. When referring to physical adversarial attacks we will consider attack scenarios where changes in the physical world will cause the cyber system in the autonomous vehicle to misbehave. Such is the example of physically manipulating traffic signs. Camera sensor attacks refer to the scenario where an attacker manages to access critical vehicle systems and manipulate directly the camera image. In such scenarios, detection and mitigation techniques will also utilize multiple additional sensor inputs such as lidar.

2.2.1 Physical Adversarial Attacks

Deep learning solutions are used in several autonomous vehicle subsystems in order to perform perception, sensor fusion, scene analysis, and path planning. State-of-the-art and human-competitive performance have been achieved by ML on many computer vision tasks related to autonomous vehicles [15]. Nevertheless, over the last years it was demonstrated that ML solutions are vulnerable to certain visual attacks [16] that can cause the autonomous vehicles to misbehave in unexpected and potentially dangerous ways, for example on physical modification of the environment and especially traffic signs [17] [18].

It is considered that these attacks and modifications are physically added to the objects themselves. The traffic signs were selected as the main target domain of this scenario for several reasons discussed below:

- The relative visual simplicity of road signs.
- Road signs exist in a noisy unconstrained environment with changing physical conditions such as the weather, lighting, distance and angle of the viewing camera,
- Road signs play an important role in transportation safety.
- A reasonable threat model for transportation is that an attacker might not have control over a vehicle's systems but is able to modify the objects in the physical world that a vehicle might depend on to make crucial safety decisions.

In this scenario, the autonomous vehicle is expected to drive from a starting location to a given destination following a specified path. Throughout this path, certain traffic signs will be physically modified. An example could be the stop or turn left/right signs due to their important role in transportation safety. Figure 7 demonstrates an example of a physical attack from a real graffiti at the left and from an engineered attack aiming to make the ML system fail but most humans would not consider it suspicious [19].



Figure 7: Appearance perturbations on Traffic Signs

Figure 8 represents the related scenario. The attacker modifies physical traffic signs. The autonomous vehicles with all the available sensors and mainly the camera aims to detect the attacks, provide notifications to the operator through HMI and to the other connected vehicles. Furthermore, the improved robust ML deep learning models should be able to overcome and not be affected by these attacks

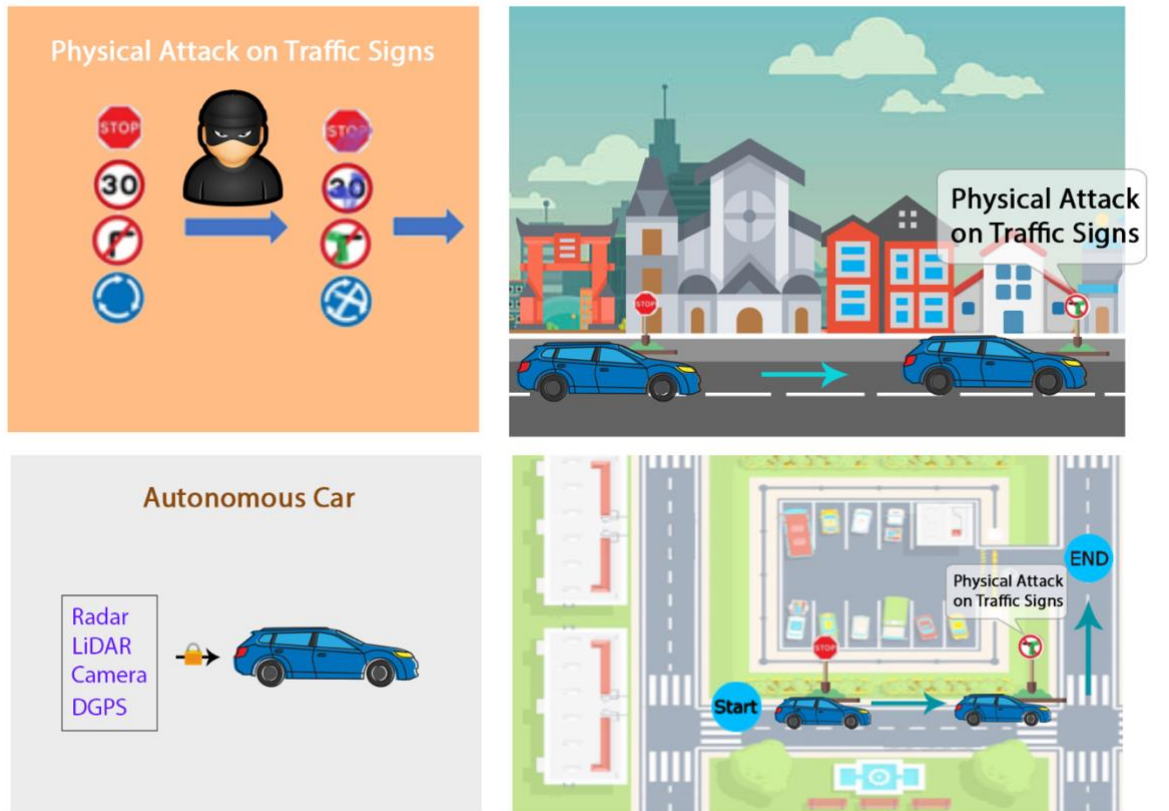


Figure 8: Adversarial Attacks on Traffic Sign

The components existing in the traffic sign physical attack scenario are as follows:

- **Autonomous vehicle:** An autonomous vehicle embodying a multitude of on-board sensors (cameras, ultrasonic, GPS, Lidar, radar) and AI providing sufficient information related to the vehicle localization, the surrounding obstacles and the possibility of collision.
- **Visual sensors:** all the available vision related sensors (e.g. Camera, LiDAR, etc.) will be considered in this scenario. In practice the camera sensor will be utilised, as it is the main sensor for scene understanding and traffic sign detection and recognition.
- **Path planning system (PPS):** PPS is a basic framework which defines the objective for the autonomous vehicle to move from one place to another. To achieve this, the PPS must choose a path and adjust to obstacles, terrain, and changing conditions to reach its destination safely. Note: The use case does not require/have access to the autonomous vehicle PPS to work.
- **Attacked traffic signs:** Real traffic signs will be modified and placed at the test area. Regarding the simulation models of traffic signs with attacks, they will be placed in the virtual environment.

The possible CARMEL components to be integrated and the required functionalities from them are listed below:

- **Machine Learning component for sign attack detection (anomaly):** ML models trained to detect attacks on traffic signs will be integrated to this scenario. The models will be based on

various state-of-the-art architectures to detect anomalies. More details are available in section 2.3.2.

- **Robust ML model for sign attached**: ML models trained to overcome such attacks will be integrated into the architecture.

Although adversarial attacks on the traffic signs comprise a very interesting use case for evaluating the potential and the limitations of CARMEL's solution on addressing cyber-attacks, the possibility of reproducing this use case on the test area with Panasonic's vehicle remains to be verified based on the input of the test area operator. Given the fact that the test area is managed by a third-party service provider, the consensus of the operator on distorting the appearance of existing traffic sign structures needs to be provided. However, this scenario will be extensively investigated in the simulator as the flexibility provided there by the simulation environment will allow detailed analysis on the precision of ML module in detecting and mitigating the attacks.

Table 3 describes the traffic sign attack scenario while Figure 9 shows the roles of the actors identified for this use case.

Use case	Scenario ID and Title	Priority level
Autonomous Vehicles – Traffic Sign Physical Attack	Detection and reaction to physical attacks on traffic signs	High
	Robustness to physical attacks on traffic signs	High

Table 3: The Traffic Sign Attack Scenario Definition

- **Vehicle operator/ passenger** – a person responsible to operate the vehicle in the case of a not fully automated one, monitoring the environment and the vehicle behaviour. They are responsible for receiving notifications from the CARMEL platform and taking the necessary measures to react to the physical attacks.
- **Connected Vehicles** – a list of other vehicles connected to the current one and the corresponding operators or passengers. They are responsible for receiving related notifications and acting accordingly.
- **Cyber-attacker** – a person conducting the physical-attack either randomly or considering adversarial permutations on physical objects such as traffic signs

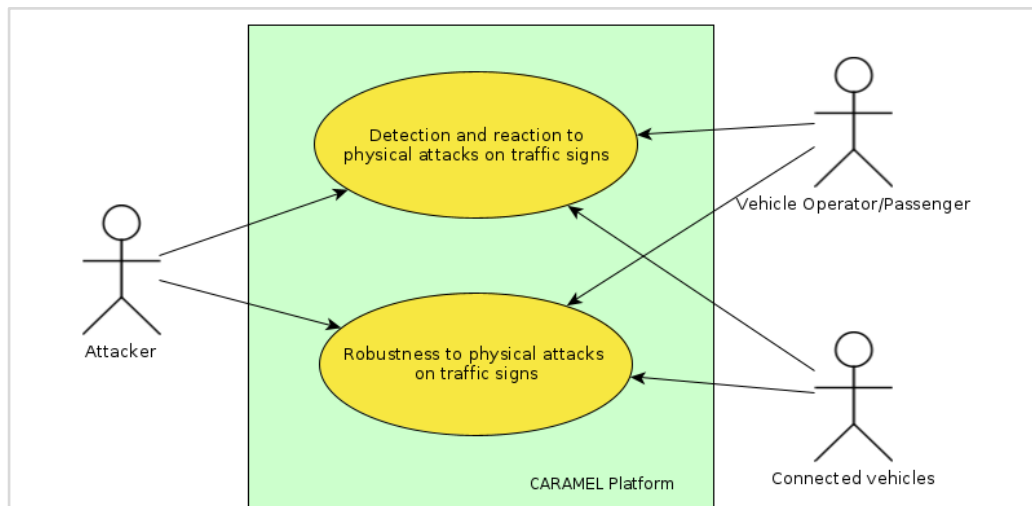


Figure 9: High-level Description of the Roles in the Scenario

Table 4 summarizes the physical adversarial attack and identifies the evaluation criteria.

Scenario Name	Detection of physical attacks on traffic signs
Related Pillar	Autonomous Vehicle Simulator
Scenario Description	The scenario deals with two kinds of attack: attacker vandalises traffic signs i.e. some random graffiti that hides a different part of the sign or a coordinated attack such as generating ML based image to cover the signs.
Brief Description	The autonomous vehicle moves in the test area. Certain traffic signs have been physically modified in order to influence the driving behaviour and planning of the autonomous vehicle. CARMEL's platform is operating in parallel to the driving system of the autonomous vehicle without influencing the decision-making module. When the vision-related sensor and the ML components of CARMEL detects a physical attack, a corresponding notification will be displayed to the vehicle operator or passenger.
Challenges	<ol style="list-style-type: none"> 1. Ability to detect physical attacks on traffic signs 2. Improved robustness on physical attacks
Assumptions & Pre-Conditions	<ol style="list-style-type: none"> 1. Datasets (real and synthetic) for traffic signs are available 2. The camera and vision sensors are properly calibrated for both the real and simulated cases
Goal (Successful End Condition)	The physical attack on the traffic signs is successfully identified without affecting the driving behaviour and the decision-making processes of the autonomous vehicle.

Involved Actors	<ol style="list-style-type: none"> 1. Attacker 2. Vehicle operator/ passenger
Scenario Initiation	An autonomous vehicle from a given location is instructed to drive to a selected end destination.
Main Flow	<ol style="list-style-type: none"> 1. Selection of starting and ending location. 2. Data acquisition mainly from the camera sensor. 3. Physical attack detection for traffic signs 4. Robust model deployment in parallel with step 3 5. Operator notification
Evaluation Criteria	<p>CARMEL platform detects the attacked signs and notifies the vehicle operator allowing them to take appropriate remedial actions.</p> <p>Metrics related to the detection and recognition accuracy such as F1 score, Precision and Recall will be considered.</p>

Table 4: Overview of the Physical Adversarial Attack Scenario

As discussed in Table 2, apart from the adversarial attacks geared towards physically distorting the appearance of some of the scene structural elements, CARMEL also targets to study the potential of cyber-attack detection techniques in estimating the occurrence of attacks on the sensor signal, camera being the most possible candidate. Throughout the subsequent subsection, the possible methods for signal distortion will be investigated along with methods for mitigation.

2.2.2 Attack on the Camera Sensor

Besides physical attacks that can induce erroneous cyber-system behaviour there is also the possibility that the camera data can be manipulated directly thus eliciting false algorithmic inferences. This can cause an AI-based perception module/controller to make incorrect decisions, such as when an autonomous vehicle fails to detect a lane/ parking marking and results to a collision.

To realize the autonomous driving functions, it is highly needed to localize and classify the obstacles in the vicinity of the vehicle. This process in contrast to image classification tasks necessitates identifying the location of all objects within the image. The last few years there has been a growing concern on the cyber-security of perception modules for object localization such as object detectors and object segmentation [6] [20], because DNNs are known to be vulnerable to adversarial examples (AEs) as shown in Figure 10. Compared to the image classifiers, the object detectors and image segmentation models are more challenging to attack, as the AEs need to mislead not only the label predictions but also the object existence prediction (whether there is an object). Adversarial attacks can be performed on the machine vision algorithm and video/image processing algorithm used for object detection (road, obstacles, road signs, etc.) by altering the image captured by the camera. More importantly, unlike classifiers that are always working on stationary images, object detectors are commonly applied in an environment where the relative position between the objects and the camera may keep changing due to the relative motion of both, e.g. object detectors on fast-moving autonomous driving vehicles.

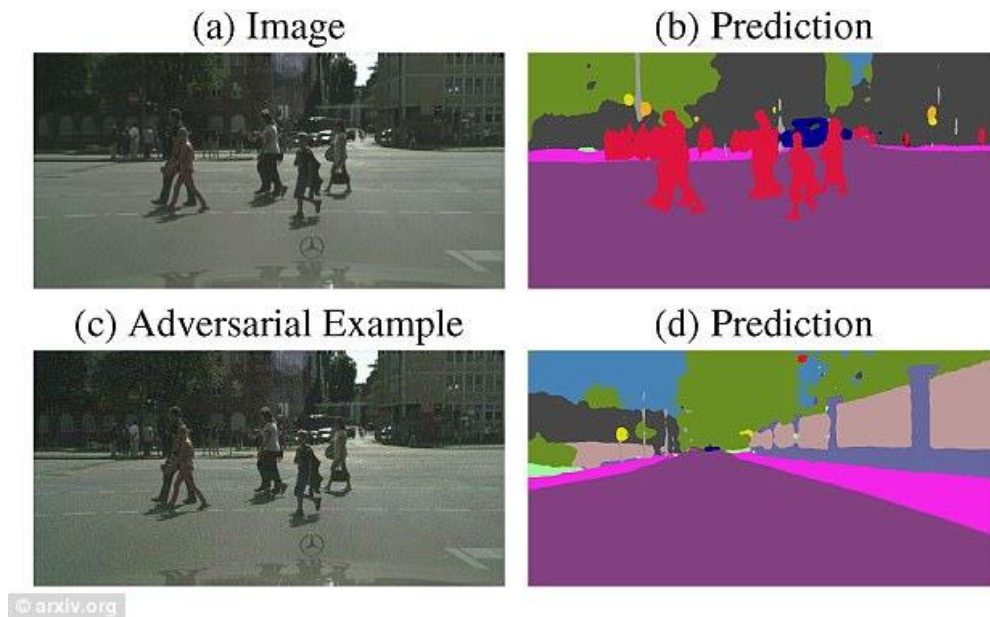


Figure 10: Targeted Perturbation Attack (either physical or digital) on the Camera Feed

Recently, it has been shown that it is possible to generate robust AEs to attack the state-of-the-art object detectors [21] and driving models [22] used in the real world. Attackers can succeed with these tactics even if they don't know the details of how the target neural net was constructed or trained. Adversarial tampering can be extremely subtle and hard to detect, even all the way down to pixel-level. Typically, two forms of attacks are considered. Hiding Attack (HA) which makes the perception module failing to recognize an entity in the environment, and Appearing Attack (AA), which makes the perception module mis-recognize the AE as a different element specified by the attacker.

This scenario studies the impact of attacking the camera signal and the disturbance that it produces on the perception module output. It is considered that such a scenario provides another vector of attack that can be used by mischievous parties to make the vehicle perception engine to misbehave [7] by either: 1) misidentifying or not detecting certain objects 2) failing to spot specific markings and 3) failing to detect lane lines amongst others. Hence, it can cause collision. Figure 10 demonstrates the use case of an HA attack, where the pedestrians were hidden by noise corrupted camera signal.

In this scenario, the autonomous vehicle will move along two predefined points on a path and at a given time instance the camera signal will be attacked in order to introduce malicious problems on the perception engine output (e.g., disturbing the detection of lane/parking markings or hiding objects lying within the field of view). The previous figure also demonstrates one such example where the pedestrians are masked causing the vehicle to steer forward since there is no obstacle detected.

To address the detection of such an attack, the output of the remaining sensor modalities can be used [23]. We assume that the attack is carried out on the image captured by the camera. Hence, additional modalities such as LiDAR can be used to detect potential discrepancies [8]. For example, if a side camera fails consistently to detect a parking spot whereas the front camera has detected the slot in the predefined location in the map and the LiDAR sensor also confirms the absence of obstacles in the same location then this can be considered as flagged incident and this information will be communicated through the Human Machine Interface (HMI) to the driver.

Table 5 describes the adversarial attack on the camera sensor to hide/appear objects. The components existing in the projected patterns attack scenario are as follows:

- Autonomous vehicle: A simulated vehicle with the basic autonomous functionality features
- Camera sensors: The camera sensor will be utilized which is the main sensor for detection and perception

- LiDAR sensor: Used to provide an additional modality to the perception module. This enables the detection of attacks on the visual sensor.
- Path planning system: Will follow a predetermined route.
- Altered Images: Images with added perturbations. The images from the vehicle camera stream will be collected and used to generate malicious perturbations. These perturbations will be injected in the vehicle field-of-view.

Table 6 summarizes the adversarial attack on the camera sensor scenario and identifies its evaluation criteria.

Use case	Scenario ID and Title	Priority level	Related requirements
Autonomous Vehicles – Camera Sensor Attack	Environment Hiding Attack	High	Attacker has inserted malicious code within the vehicle that can cause the adversarial attack
	Environment Appearing Attack	Medium	

Table 5: The Attack on the Camera Sensor Scenario Definition

Scenario Name	Adversarial attacks on camera sensor
Related Use Case	Autonomous Vehicle
Scenario Description	First, an attacker may analyse public freely available object detectors to find attacks. This phase is usually offline. The second phase is the attack deployment where the perturbations are applied on one of the camera sensors that are on board. The demonstration of this scenario will involve the attack application phase within simulation context.
Brief Description	The autonomous vehicle is expected to drive from a starting location to a given destination following a specified path. At a given time instance the image of the vehicle will be tampered through a specific perturbation intended to cause the perception module to misbehave (e.g., either detect objects that are not truly present or hide objects that are within the field of view).

Challenges	<ol style="list-style-type: none"> 1. Ability to detect adversarial attacks on the camera signal that disturb the detection of important elements in the scene. 2. Improved robustness on adversarial attacks.
Assumptions & Pre-Conditions	<ol style="list-style-type: none"> 1. Datasets of environment structures (e.g., parking markings, pedestrians, vehicles, driving lines) in realistic driving conditions are available 2. The camera image sensors are properly calibrated 3. The camera feed can be accessed, altered and then can be fed back to the perception module. Otherwise, the attack scenario will follow the flow of the physical attack.
Goal (Successful End Condition)	The attack on the camera image is successfully identified and if possible, without affecting the driving behaviour and the decision-making processes of the autonomous vehicle.
Involved Actors	<ol style="list-style-type: none"> 1. Attacker 2. Vehicle operator
Scenario Initiation	An autonomous vehicle from a given location is instructed to drive through a predefined area.
Main Flow	<ol style="list-style-type: none"> 1. Selection of starting and ending location. 2. Data acquisition mainly from the camera sensor. 3. Adversarial attack detection targeting a perception module. 4. Operator notification
Evaluation Criteria	The adversarial detection within the CARMEL platform will detect the attack and notify the vehicle operator allowing them to take appropriate actions. The evaluation will be performed both in the simulator and on the real vehicle.. Metrics related to the detection accuracy will be considered.

Table 6: Overview of the Camera Sensor Attack Scenario

One of the crucial factors in efficiently detecting and mitigating cyber-attacks is to properly parameterize and tune the learning modalities employed in the CARMEL architecture. Since the implicit definition of the learning schemes would require complex analytical expressions which is difficult to be proven as convergent, we will try to solve this part of the problem by using Machine Learning. In this case, the parametrization of the learning scheme will be automatically defined during the training phase. Thus, proper selection of the training sets so as them to closely resemble the characteristics of the data encountered on real scenes, is very important. Section 2.3 discusses the data collection strategy to be followed as well as the augmentation techniques that the consortium will adopt in order to suppress overfitting cases and lack of generalization.

2.3 Data Collection/Selection Methodology

The dataset used for training will be based on 3 major pillars: (a) synthetic dataset, (b) real dataset, (c) augmented data. Throughout the following subsections the criteria for structuring the dataset will be extensively discussed.

2.3.1 Synthetic Dataset for Traffic Signs

Table 7 describes the offline variants of the data records that can be created by customizing the simulation tool. The headings for each column are described as follows:

- Attack Class: type of attack.
- Attack Variation: manipulation methods used to perform attack.
- Location: refers to various settings and conditions e.g. location, world, junction, straight etc.
- Time of Day: day or night, can also be expanded to include dusk.
- Weather: different weather conditions
- Signs/Duration: presents the duration in number of frames and the amount of possible signs used for each dataset.

Note: It is expected to have around 10000 samples for each data class.

Attack class	Attack Variation	Location	Time of Day	Weather	Signs / Duration
Normal (no attack)	None	> 3 Locations	Day / Night	Sun / Cloudy / Rain	20 signs for 30 frames
Attacked with Noise	Gaussian Noise, Masking / Obstruction	> 3 locations	Day / Night	Sun / Cloudy / Rain	20 signs for 30 frames
Adversarial Attack	ML generated attack	>3 locations	Day / Night	Sun / Cloudy / Rain	20 signs for 30 frames

Table 7: Offline Variants of the Data Records

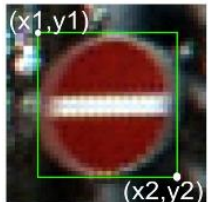
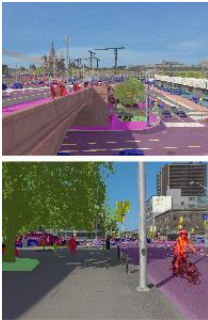

Additionally, the data generation method should support different types of cameras (e.g. wide field of view) and customizable cameras so as to resemble the target position of the camera on the real vehicle.

In order to produce datasets embodying information from both camera and LiDAR sensors, we can use simulators, e.g. CARLA [37], Using such simulators it will become feasible to export high-quality, synchronized LIDAR and camera data with object annotations reflecting real-life sensor arrays.

Apart from the volume of visual/LiDAR data produced by the simulation environment, CARMEL's Adversarial Attack detection engine will be benefitting by incorporating some publicly available dataset in the training process. Throughout the subsequent subsection we are going to provide a brief overview of the available datasets along with its characteristics and possibility to be incorporated in the training. More specifically, Table 8 presents numerous visual datasets on traffic signs, while Table 9 summarizes the available datasets produced by the automotive sensing community, which apart from the camera, incorporate other sensors as well.

2.3.2 Publicly Available Dataset for Traffic Signs

Table 8 shows the list of publicly available datasets for traffic signs. These datasets will be used to train the machine learning models and to generate the attacks described in the section above.

Dataset	Description	Sample
LISA Traffic Sign [24]	<p>The LISA Traffic Sign dataset is a US traffic signs dataset that contains set of videos and annotated frames.</p> <ul style="list-style-type: none"> • 47 US sign types • 7855 annotations on 6610 frames. • Sign sizes from 6x6 to 167x168 pixels. • Images obtained from different cameras. Image sizes vary from 640x480 to 1024x522 pixels. • Some images in colour and some in grayscale. • Full version of the dataset includes videos for all annotated signs. • Each sign is annotated with sign type, position, size, occluded (yes/no), on side road (yes/no). 	
Mapillary Global [25]	<p>The dataset contains a diverse street-level image with bounding box annotations for detecting and classifying traffic signs around the world.</p> <ul style="list-style-type: none"> • 100,000 high-resolution images (52,000 fully annotated, 48,000 partially annotated) • Over 300 traffic sign classes with bounding box annotations • Global geographic reach of images and traffic sign classes, covering 6 continents • Variety of weather, season, time of day, camera, and viewpoint 	
The German Traffic Sign Recognition Benchmark (GTSRB) [26]	<p>GTSRB is a multi-class, single-image classification challenge.</p> <ul style="list-style-type: none"> • Single-image, multi-class classification problem • More than 40 classes • More than 50,000 images in total • Large, lifelike database • Reliable ground-truth data due to semi-automatic annotation • Physical traffic sign instances are unique within the dataset (i.e., each real-world traffic sign only occurs once) 	


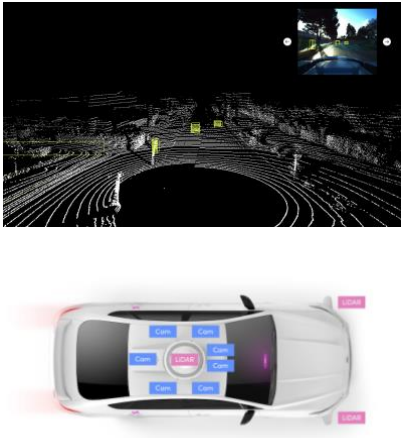

<p>The German Traffic Sign Detection Benchmark (GTSDb) [27]</p>	<p>GTSDb is a single-image detection assessment for researchers with interest in the fields of computer vision, pattern recognition and image-based driver assistance.</p> <ul style="list-style-type: none"> • a single-image detection problem • 900 images (divided in 600 training images and 300 evaluation images) • division into three categories that suit the properties of various detection approaches with different properties • an online evaluation system with immediate analysis and ranking of the submitted results 	
---	---	---

Table 8: Publicly Available Dataset for Traffic Signs

2.3.3 Publicly Available Datasets Featuring the Raw Sensor Camera and LiDAR Data

Table 9 shows the list of publicly available datasets for raw sensor camera and LiDAR data. These datasets will be used to train the expected machine learning models and to generate the attacks described in the section above.

Dataset	Description	Sample	Link
Lyft	<p>Main features:</p> <ul style="list-style-type: none"> • Up to 7 cameras • Up to 3 lidars • Over 55,000 3D annotated frames • A drivable surface map • An HD spatial semantic map • 4,000 Lane segments • 197 Crosswalks • 60 Stop signs • 54 Parking zones • 8 Speed bumps • 11 Speed humps 		Lyft Data
Kitti	<p>Kitti contains a suite of vision tasks built using an autonomous driving platform. The full benchmark contains many tasks such as stereo, odometry, 3D object detection, 3D tracking, etc.</p>		Kitti Data

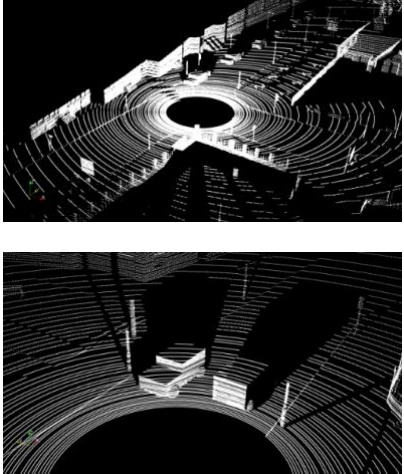
Motion Distorted Lidar	<p>This is a large-scale dataset generated using CARLA, aiming to dynamic object detection.</p> <p>Main features :</p> <ul style="list-style-type: none"> • Town 1 (2.9 km of drivable roads with 90 vehicles) • Town 2 (1.9 km of drivable roads with 60 vehicles) 		Motion Distorted LiDAR Data
------------------------	---	--	---

Table 9: Publicly Available Dataset for the raw Sensor Camera and LiDAR Data

2.4 Use of Artificial Intelligence and Machine Learning

2.4.1 Physical Attack on Traffic Signs

The Machine Learning (ML) algorithms will be used for the detection and identification of attacks or anomalies in traffic signs. Various research has demonstrated that recognition-based models are vulnerable to engineered visual attacks [16]. Therefore, additional ML models should be utilised to identify such attacks. Following are the key area where ML will be used:

1. Anomalies Detection:

Auto Encoder (AE) has been a popular option for detecting anomalies [28]. When an anomaly is defined as a one-class classification problem, AE can be trained to re-create the dominant class of training dataset. Once, AE has learnt to re-create the class it has seen before, it is expected that any new and novel classes are re-created poorly. By measuring how poorly the network performs on unseen data, it is then possible to classify the new point as an abnormal or normal class. Similarly, Generative adversarial network (GAN) has been extensively used in the identification of anomalies within the image and videos [29].

2. Mitigation processes

ML can be trained to recover cyber-attacks or reconstruct the missing or obstructed part of the images [30] [31]. This can be particularly important to improve the robustness of the model. Such models can be used to reconstruct the traffic signs that are vandalised or obstructed. Likewise, other useful attributes such as improvement of road layouts can be carried out using similar techniques [32].

3. Generation of simulated attacks

Several adversarial approaches have been proposed to generate attacks in the form of small perturbations to images that remain almost imperceptible to human vision but such attacks can cause detection-base model to significantly decrease the detection and recognition performance [33] [34]. ML will be used to generate such simulated attacks.

The basic goal of the use of machine learning in addressing adversarial attacks on the physical layer is two-fold: (a) at first the detection of the attacked patterns needs to be tackled while at a subsequent step (b) we are also interested in producing via ML a multitude of attack patterns and use case scenarios that result in resembling real world attacks. The increased data volume should improve the overall performance of the ML model. In order the enhance the generalization capability, data augmentation will be considered. Thus, training should involve first train the model with synthetic data and then fine-

tune it with real-world data. This process is necessary as a model trained with only synthetic data won't perform as good as real data.

Further to defining the usage of ML in addressing the attacks on the physical layer, we will also summarize its role in tackling sensor attacks as well.

2.4.2 Attack on Camera Sensor

The Machine Learning (ML) algorithms will be used for the detection and mitigation of direct attacks on the camera sensor data. Research has shown that through multimodal sensing it is possible to detect mis-behaving sensors thus detecting potential attacks. Additional AI/ML models will also be utilised to identify such attacks from image source and either discard the image sensor or attempt to reconstruct the input image. Figure 11 presents the concept of using AI/ML in the autonomous vehicle case in the attack on the camera sensor. In summary, the main components utilizing the AI/ML are the following:

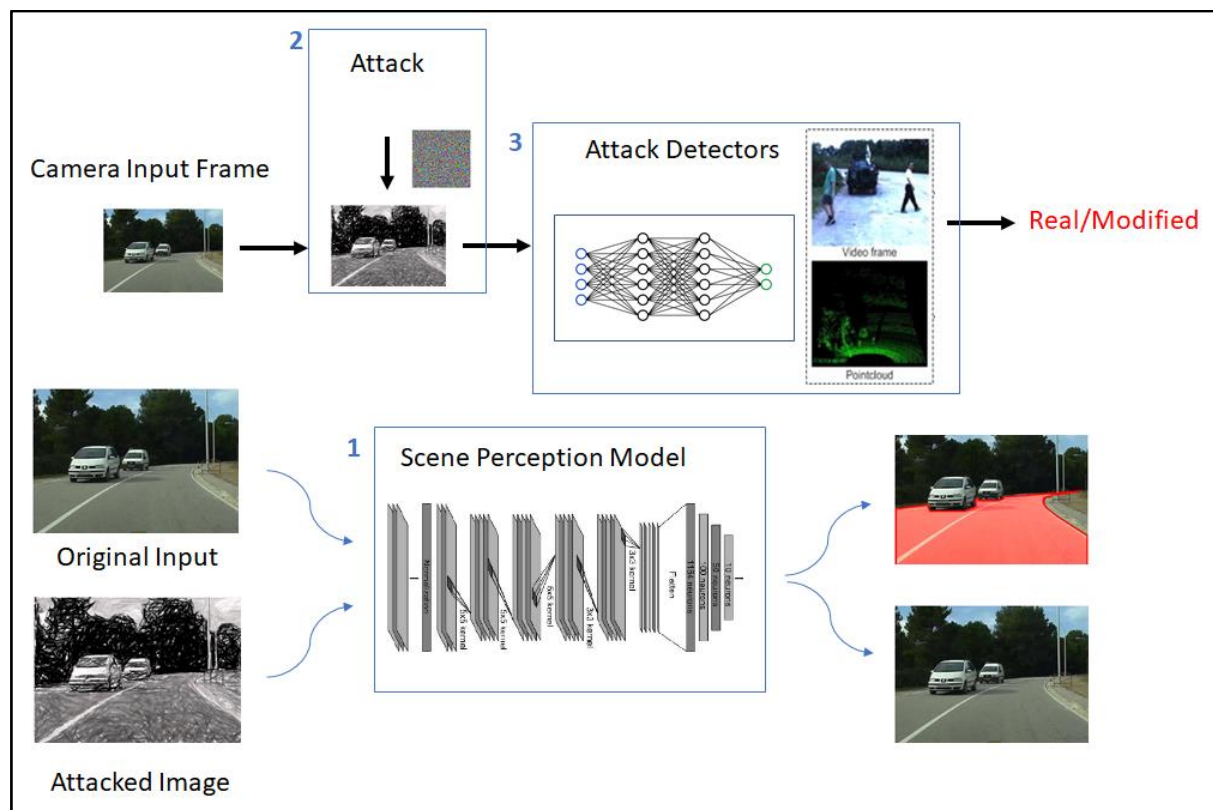


Figure 11: Concept of using ML in the Attack on the Camera Sensor

1. Scene Perception Model

A model that will utilize AI/ML to detect a specific environment object (road area, parking markings, vehicle, pedestrian, lane line). For the purposes of CARMEL, we consider such models as given and will only train such models if the underlying functionality is not available.

2. Attack on the Image

It will be used to simulate attacks or adversarial examples as presented in [35]. They are deliberately calculated perturbations to the input that can result in an error in the output from the perception model. Autonomous vehicles developed nowadays lack robustness to adversarial conditions. They can be directed by expert knowledge of the attacker of underlying the AI/ML perception model to be more effective. There are many methods that aim to compromise the integrity of ML/DL models. Most of them rely on gradient in order to fool the models.

Regarding the adversarial examples a perturbation is calculated by approximating an optimization problem given in Equation (1) iteratively until the crafted adversarial example gets classified by ML classifier in targeted class.

$$x^* = x + \underset{dx}{\operatorname{argmin}}\{\|d\|: f(x + d) = t\} \quad (1)$$

- x is the correctly classified sample
- x^* is an adversarial sample
- d is the perturbation
- f is the classifier
- t is the targeted class

3. AI/ML-based Attack Detectors

The goal is here to improve the robustness of the environment perception module through the use of AI/ML models to reduce the impact of the attack (e.g., denoising autoencoder [36]).

Different AI/ML techniques will be investigated to Improve the robustness of the environment perception module and reduce the impact of the attack (e.g., denoising autoencoder [36]).

To make Autonomous Vehicles less vulnerable to any attacker, it is necessary to develop adversarial robust ML/DL solutions. Adversarial training as shown in Figure 12 or input reconstruction, that adversarial samples will be cleaned to transform them back to legitimate ones will lead to more robust methods.

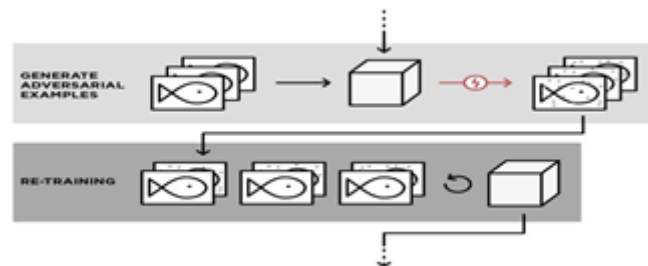


Figure 12: Adversarial training

Additionally, combining multiple defence strategies can clean some of the adversarial perturbations. Another solution would be to modify the whole ML/DL model and its parameters learned from the data, either using Network Distillation or Gradient Regularization.

Various defences have been proposed to mitigate the effect of adversarial attacks. These defences can be grouped under three different approaches:

- Modifying the training data to make the classifier more robust against attacks, e.g., adversarial training which augments the training data of the classifier with adversarial examples.
- Modifying the training procedure of the classifier to reduce the magnitude of gradients, e.g., defensive distillation.
- Attempting to remove the adversarial noise from the input samples based on the concept that correctly classified examples tend to have greater maximum SoftMax probabilities than erroneously classified and out-of-distribution examples

A defence strategy which uses a WGANs is proposed in [50]. WGAN is trained on legitimate (unperturbed) training samples to “denoise” adversarial examples and combat both white-box and black-box adversarial attacks against classification networks.

Besides the presented above cases which relies on the ML solutions developed using a single data source, there are possibilities to employ ML solutions leveraging multiple data sources. CARMEL

considers these types of solutions important and in the following we will present a case where multi-modal fusion is used for anomaly detection.

Multi-sensor data fusion is the process of combining observations from a number of different sensors to provide a robust and complete description of an environment or process of interest. Data fusion finds wide application in many areas of autonomous vehicles such as object recognition, environment perception, road detection, etc. Current approaches for multiple tasks related to autonomous vehicles use either cameras, LIDAR, Radar or other sensors. Cameras can work at high framerates and provide dense information over a long range under good illumination and fair weather. However, being passive sensors, they are strongly affected by the level of illumination. A passive sensor is able to receive a specific amount of energy from the environment, light waves in the case of cameras, and transform it into a quantitative measure, such as an image. Clearly, the process depends on the amplitude and frequency of the light waves, influencing the overall result, while a reliable system should be invariant with respect to changes in illumination [38]. LIDARs sense the environment by using their own emitted pulses of laser light and therefore they are only marginally affected by the external lighting conditions. Furthermore, they provide accurate distance measurements. Based on this description of benefits and drawbacks of these two sensor types, it is easy to see that using multiple sensors might provide an improved overall reliability.

Recently, a variety of 3D detectors that exploit multiple sensors have been proposed. A multi-task multi-sensor detection model that jointly reasons about 2D and 3D object detection, ground estimation and depth completion has been proposed in [39]. Pointwise and ROI-wise feature fusion are applied to achieve full multi-sensor fusion, while multi-task learning provides additional map prior and geometric clues enabling better representation learning and denser feature fusion. Moreover, F-PointNet [40] uses a cascade approach to fuse multiple sensors. Specifically, 2D object detection is done first on images, 3D frustums are then generated by projecting 2D detections to 3D and PointNet [41][42] is applied to regress the 3D position and shape of the bounding box. Furthermore, object localization from a frustum in LiDAR point cloud has difficulty dealing with occluded or far away objects as LiDAR observation can be very sparse. MV3D [43] generates 3D proposals from LiDAR features, and refines the detections with ROI feature fusion from LiDAR and image feature maps. AVOD [44] further extends ROI feature fusion to the proposal generation stage to improve the object proposal quality. However, ROI feature fusion happens only at high-level feature maps. Furthermore, it only fuses feature at selected object regions instead of dense locations on the feature map. To overcome this drawback, ContFuse [45] uses continuous convolution [46] to fuse multi-scale convolutional feature maps from each sensor, where the correspondence between image and bird's eye view (BEV) spaces is achieved through projection of the LiDAR points. However, such fusion is limited when LiDAR points are very sparse. To address this issue, other methods predict dense depth from multi-sensor data, and use the predicted depth as pseudo LiDAR points to find dense correspondences between multi-sensor feature maps.

It is clear enough that multi modal fusion in the field of autonomous vehicles is important for a robust and complete description of an environment to be provided. CAMEL will focus on AI and multi modal fusion techniques in order to achieve the aforementioned task. To elaborate more on the solution below we will present some details about the generation of adversarial attacks and a deep learning (DL) solution for detecting attacks.

For each AI/ML model the most appropriate data generation process and datasets will be selected from Table 9 and will be used for training. Following are the type of datasets that will be used to train the ML model:

1. Real-world publicly available datasets
2. Augmented datasets - Real dataset augmented with synthetic datasets
3. A synthetic dataset will be constructed either from both a simulation environment as well as any available real-world data to capture different weather conditions, traffic, etc.

2.5 Validation Methodology

In this section we will explain the methodology that CAMEL will use to validate the project outcome. In broad terms there will be two major paths: simulation and real-life demonstrations. In the following we will highlight why a simulation practice is needed and how it will complement the real-life demonstrations. Note: given the limited duration and resources, not all validation cases of CAMEL will be carried out in field trials.

2.5.1 Role of Simulation in the Autonomous Mobility Pillar

The goal of CARMEL is to demonstrate the use of AI/ML-based classification in an alarm system in the context of the automotive cybersecurity domain on the basis of dynamic system/sensor data and other suitable data streams around the vehicle's communication patterns. This section explains the need for large amounts of data that are not available and/or generatable on a real-world basis in a context such as CARMEL. For example, recording different weather conditions is out of scope and not reproducible. Therefore, as discussed in the beginning of this document the use of synthetic data is inevitable. The literature shows that models based on synthetic or hybrid data detect patterns in the real world. For this reason, CARMEL will use synthetic data generated within a simulation tool in addition to freely available datasets and datasets captured by the sensor setup of the demo vehicle. The advantage of creating synthetic data is that, in contrast to freely available data sets, the obtained data can be adapted specifically to the application, i.e. sensor data from the real car are as "similar" as possible.

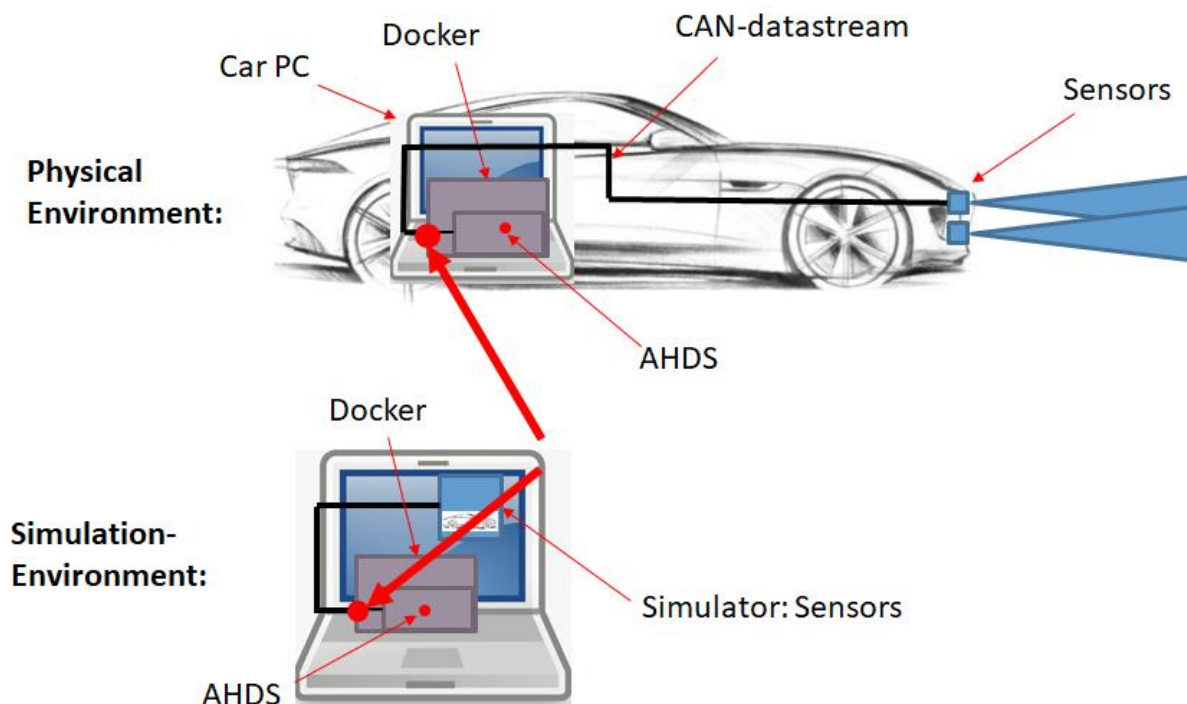


Figure 13: Comparison of the physical and simulated process starting with the sensors up to the anti-hacking device software (AHDS)

2.5.2 Comparison of Physical and Simulation Environment

Figure 13 shows the process for the physical and the simulation environment. In the physical environment, sensors generate data, which is transferred to an intermediate, experimental data (if available) via a CAN-DataStream. The Car-PC is connected to the anti-hacking device. The anti-hacking device AHD contains a docker container into which the anti-hacking device software AHDS is deployed. In the simulation environment, the entire pipeline is run on a computer. Sensor data is generated in the simulation tool. The docker container and the software of the AHD receive the generated data. From the moment the data reaches the docker container, the simulation and physical execution in a real car are identical. The anti-hacking software cannot distinguish between physical and simulated data and can therefore be exchanged between the environments.

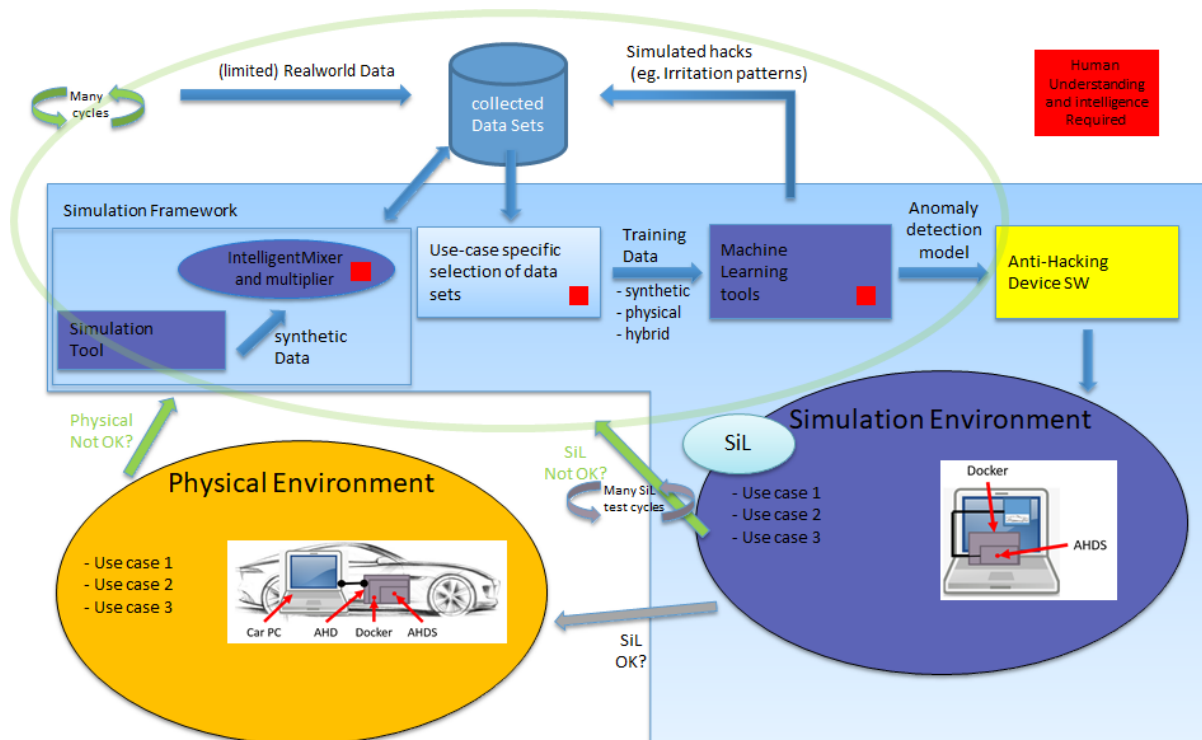


Figure 14: Simulation environment workflow and AHDS deployment in physical environment

2.5.3 Use of Simulation Framework within Pillar 1

Figure 14 shows a workflow within and around the simulation framework. There are three major cycles. First the basic cycle for adjusting the basic settings. Here, data is collected in a pool consisting of limited real-world data and synthetic data generated in the simulation tool. Furthermore, hack simulations generated via ML are collected. The data is mixed and multiplied. Based on the collected data use-case specific data sets are compiled and used as training data for training ML models. The training data can consist of real, synthetic or hybrid data. CARMEL trains different ML-models, on the one hand models for generating hacks, where the generated data are collected in the data pool and, on the other hand, models for the detection of anomalies. The latter model is introduced in the second cycle, the Software-in-the-Loop cycle. Within the cycle, synthetic data can be adapted and remixed and duplicated. Datasets can be compiled in a use case- and scenario- specific manner, and the hyperparameters of the neural networks can be modified. The second cycle extends the basic cycle, where the trained anomaly detection model corresponds to the AHDS and is tested in the simulation environment. If the model does not deliver satisfactory results, the parameters mentioned above are adjusted again on the basis of the evaluation. If the test delivers satisfactory results, the AHDS is transferred to the third cycle and tested in the physical environment. Depending on the result, the mentioned parameters are adjusted again.

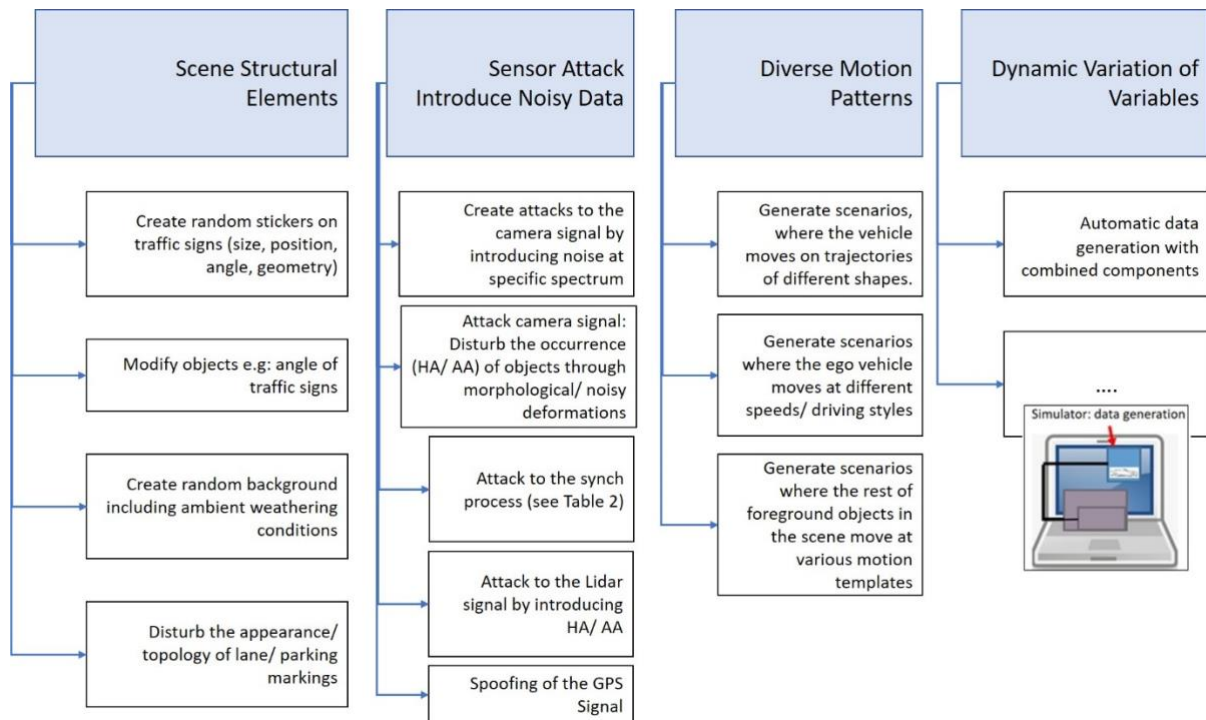


Figure 15: Data generation acceleration and diversity increase by creating individual functions within the simulation tool

2.5.4 Simulation Tool

With the technological advancements in the field of graphic engines, the simulation of scenarios with static objects such as traffic signs, semaphores, benches, buildings, etc. all of them with high quality textures has become possible. Additionally, having dynamic objects deployed in the simulated world is feasible. For example, cars, trucks, bicycles, pedestrians or moving objects can be simulated with shapes very similar to the shapes of its counterpart in the real world. It has to be possible to extend the simulation environment with the specific functions shown in Figure 15. For example, diversity can be increased by creating random stickers on street signs, varying street sign orientations by changing backgrounds and weather conditions. Further additions can be scenario specific trajectories and (random) variations within trajectories. By automatically combining and executing the above functions, many complex scenarios can be generated automatically.

Different simulation tools have been developed to support such simulated worlds, some are open source e.g. CARLA simulator [37], Microsoft AirSim [51], while others are from private companies e.g. AVL [52] and Automotive AI [53]. They were developed to facilitate easy deployment of artificial neural networks in industrial applications. The following uses CARLA as an example. This tool can be extended by specific functions as shown in Figure 15, thus simplifying data generation. A highlight of CARLA is the possibility to collect data from sensors attached to specific locations on the vehicle. CARLA allows the specification of sensor properties within certain limits. Table 10 shows configurable parameters of some sensors.

Sensor (arbitrarily many)	Configurable Parameters
RGB	Position, resolution, gamma, field of view, aperture, shutter speed, iso, extrinsic parameters
RGBD	Position, resolution, field of view, lens distortion.
Lidar	Position, range, refresh rate, field of view.

Table 10: Some sensors included in CARLA and their configurable parameters

As we now understand why and how the simulation tool would be used in CARMEL, let's focus on the scenarios explained under pillar 1 and map them into the simulation activities and/or real-life demonstrations.

The validation of the physical adversarial attack will be carried out as follows. First, we assume that the goal of the attacker is not to completely fail the detection-based system but to mislead the detection model to detect the other signs. For example, an attack on the physical stop signs using advanced neural network can mislead detection-base model detecting signs as no stop signs. Based on that, the validation and demonstration of the use case will be performed in both real and simulation environments. Metrics related to the detection and recognition accuracy such as F1 score, Precision and Recall will be considered.

Physical Adversarial Attack	Demonstration Type	
	Real	Simulation
Anomaly detection in traffic sign	To be confirmed	Yes
Robust network for pre-processing	No	Yes

Table 11: Validation methods of Physical Adversarial Attack

The attack on the camera images will be demonstrated in a real-use case where an environmental element (e.g., parking mark) will be hidden/appear based on the attack. The specific element will be decided based upon the availability of data and sensor input. As mentioned above, these data could either come from:

- Driving simulator,
- Real data (video captures)
- Augmented

Each collected data element is being provided with a timestamp. Detection techniques will mainly be based on the co-registration and processing of data from multiple sources located at different strategic points on the vehicle. If the previous module detects something suspicious, then an alert could be raised in order to forewarn the user of a possible attack. By implementing some filtering techniques, we could mitigate the attack in order to recover the malicious data and let the system decide again.

Apart from the real-use case scenario, we are also going to implement an adversarial attack using a simulator e.g. CARLA. The purpose of the aforementioned attack is to add a perturbation to the input data that can result in an error in the output of a trained model. We aim to develop adversarial robust ML/DL solutions in order to make Autonomous Vehicles less vulnerable to any attacker.

Table 12 summarizes the validation activities related to the attack on camera sensor scenario.

Attack on Camera Sensor Scenario	Demonstration Type	
	Real	Simulation
Detection and Mitigation of attack on Camera Sensor with multimodal fusion: An environment element (e.g., parking mark) will be hidden/appear.	Yes	Yes

Table 12: Validation methods of Attack on Camera Sensor

2.6 Use of the Anti-Hacking Platform

In the Panasonic autonomous car sensor data is transported over the CAN bus. The anti-hacking device, using its CAN bus interface, can intercept this traffic, extract the raw sensor data, and can analyse the data using pre-trained TensorFlow Lite models to detect possible attacks on the sensor.

Figure 16 shows how in the training phase sensor data is collected from sensors in the car. The sensor data must be classified appropriately to create training data for the ML environment. In the ML environment data scientists create models for attack and threat detection.

In order to deploy the finished models to the anti-hacking device (lower part of the diagram), the models must be converted to the TensorFlow Lite format supported by the ML TPU in the anti-hacking device.

Finally, there will be real-world testing, where the anti-hacking device in the car actually listens to the raw data on the CAN bus and uses the TensorFlow Lite model to detect attacks on the sensors.

Figure 17 shows how the anti-hacking device is integrated with the autonomous vehicle. There are basically be two interface between the car and the anti-hacking device: The first interface is the CAN bus, the other interface (yet to be defined in detail) is the interface to a visualization component in the car that indicates the attack situation (or the normal, non-attack mode) to the driver of the car.

Figure 18 shows how the anti-hacking device (target hardware or virtualized machine-based anti-hacking with Coral USB Accelerator) can be integrated into the development and simulation workflow. The main difference is that, instead of the real car with real sensors, the CARLA simulation environment will be used to simulate the car and its sensors. Since there is no CAN bus, the connection between simulation environment and anti-hacking device will be implemented using high-level REST-based interfaces. The ML software running on the anti-hacking device (in the form of a dockerized ML app) will be the same as for the real car case.

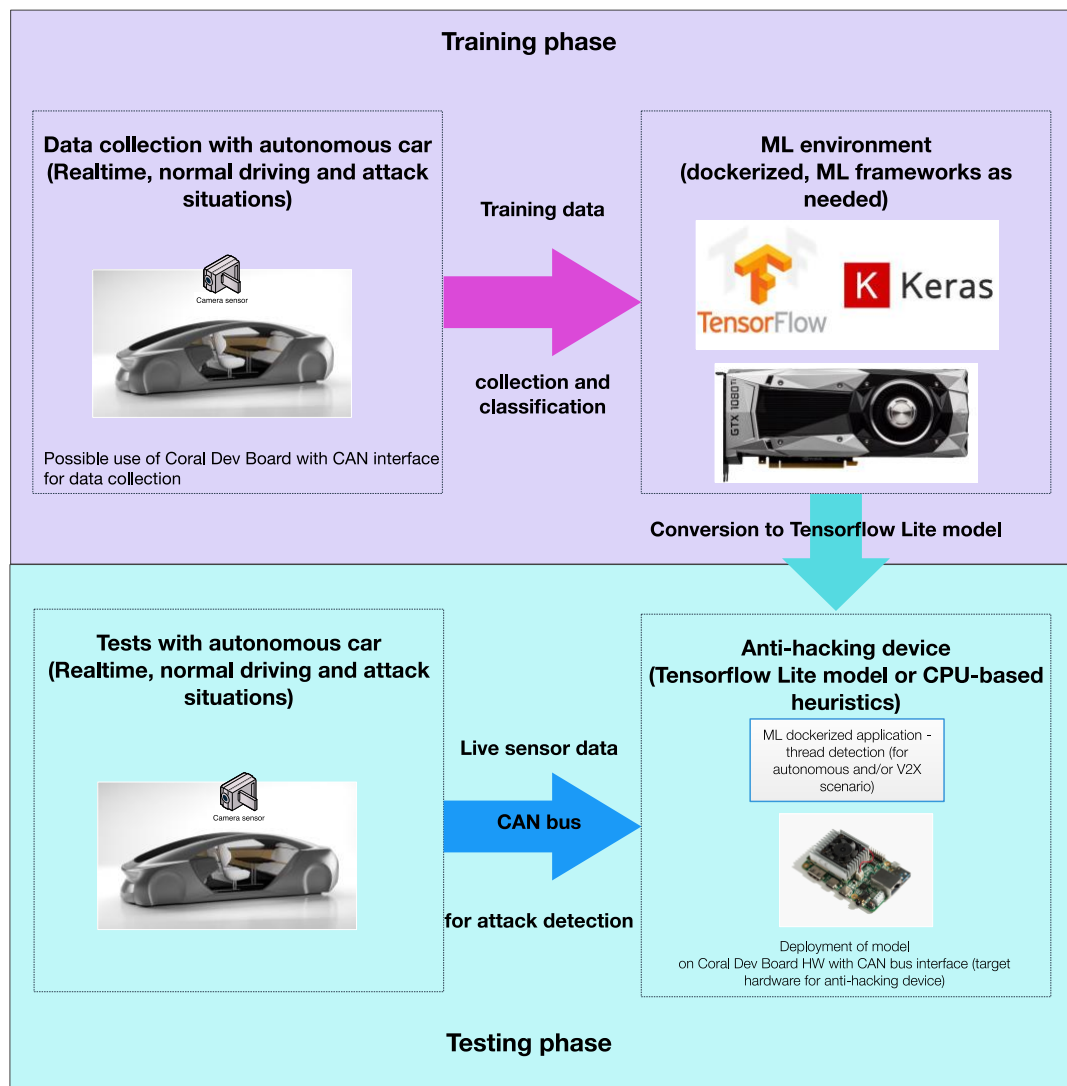


Figure 16: From Training to Deployment in the Autonomous Car

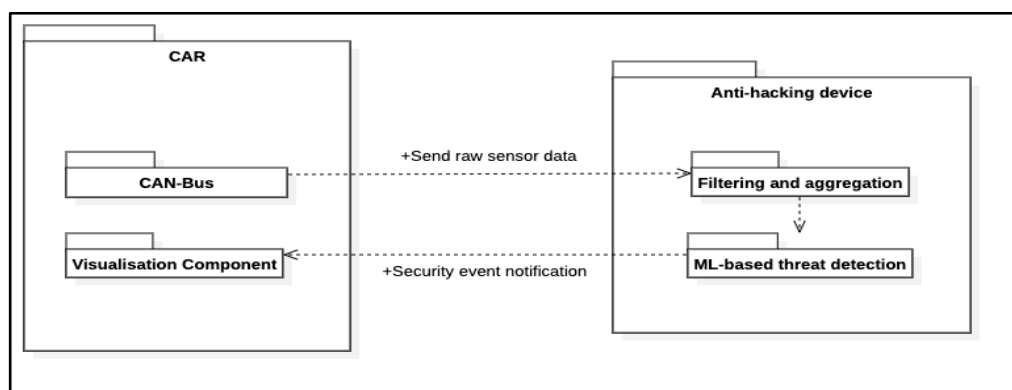


Figure 17: Integration of anti-hacking device with autonomous vehicle

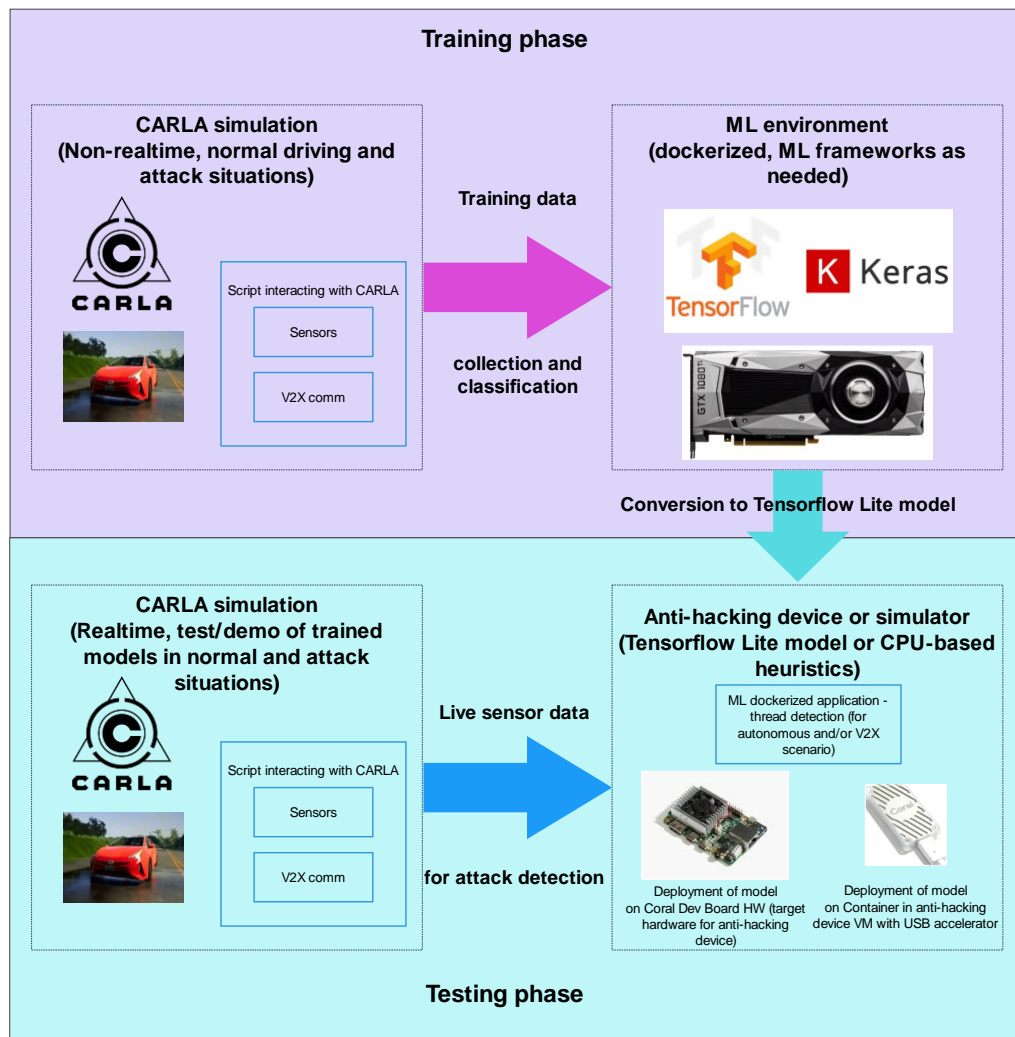


Figure 18: Training and validation in the simulation environment

2.7 Functional Requirements

In a high-level perspective, the following workflow has to be carried out in the CARMEL scenarios.

- Receive input from the car sensors
- Receive input from streamed data
- Apply attack detection ML algorithms
- Apply ML algorithms to overcome attacks (robust models)
- Push notifications

The functional requirements of the CARMEL system are presented below. The requirements ID, name, and description are given. Furthermore, each requirement is linked with other sources of the CARMEL project, such as user, security and privacy requirements

Reg ID	CRPL1-FR01
Title	Data Generation

Definition - Description	The CARMEL platform must be able to generate synthetic data from a selected simulation tool.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide an asset and data management environment. Accordingly, the simulator will support different traffic signs, weather conditions, and locations.

Reg ID	CRPL1-FR02
Title	Anomaly detection of Traffic Signs
Definition - Description	The CARMEL platform must be able to identify the anomaly in traffic signs.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide ML models to detect anomalies in traffic signs in various scenarios.

Reg ID	CRPL1-FR03
Title	Reconstruction of attacked Traffic Signs
Definition - Description	The CARMEL platform must be able to reconstruct the appropriate traffic signs when attacks are present.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide ML models to filter and reconstruct the signs.

Reg ID	CRPL1-FR04
Title	Anomaly detection on Lane/Parking Markings
Definition - Description	The CARMEL platform must be able to detect adversarial attacks resembling appearance perturbations on structural scene elements e.g: Parking/Lane Markings.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide a Learning Module aiming at detecting anomalies in Lane/Parking Markings.

Reg ID	CRPL1-FR05
Title	Reconstruction of adversarial attacks on Lane/Parking Markings
Definition - Description	The CARMEL platform must be able to detect adversarial attacks resembling appearance perturbations on structural scene elements e.g: Parking/Lane Markings.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide a Learning Module aiming at detecting anomalies in Lane/Parking Markings.

Reg ID	CRPL1-FR06
Title	Detection of Attacks on Camera Sensor based on Multi-model Fusion
Definition - Description	The CARMEL platform will be able to detect attacks on the camera sensor by cross-referencing the perception output with the LiDAR output.
Target WP	4
Priority	Mandatory
How	The CARMEL platform will provide ML models to detect any discrepancies between LiDAR and image detections.

Req ID	CRPL1-FR07
Title	Mitigate effect of attack
Definition - Description	The CARMEL platform will be able to mitigate the attacks on the camera sensor by signalling a warning alarm. Further, it will attempt to improve the image quality to mitigate the effects of the attack if possible.
Target WP	4
Priority	Mandatory
How	Once an attack has been detected the system will display a warning on the HMI with the error message that the driving environment has been compromised. It will also attempt through the use of generative models to reduce the effect of the attack on the image.

3 Pillar 2 – Connected Mobility

3.1 Context

Nowadays vehicles are provided with three different kind of communication systems:

- Communications inside the vehicle: these are communications between the different parts of the vehicle itself. Mechanic and electronic components of the vehicle as injectors, brakes, gears, the control system etc. interchange data through the CAN bus or the automotive Ethernet.
- Communication of the infotainment system: users are able to consume multimedia content through the display or vehicle's speakers, maintain voice calls or access Internet content. Different kinds of network technologies are used for this purpose: IEEE 802.11, Bluetooth, cellular networks, USB connectors.
- V2X Communications: The so called V2X (Vehicle-to-Everything) communications enable vehicles to communicate with the road infrastructure and other road users (vehicles, scooters, bikes, or pedestrians) and to have a more accurate knowledge of their surrounding environment that can improve the traffic safety and provide new Intelligent Transport Services (ITS). European Telecommunications Standards Institute (ETSI) ITS-G5 suite in Europe and the Wireless Access in Vehicular Environments (WAVE) in the US define all standards and protocols to provide numerous ITSs.

Among these types of vehicle communications, CAMEL Pillar 2 will address the functional, security and privacy issues of the V2X Communications to provide a secure environment for ITS applications. This technology is relatively new and the security issues are not yet completely studied, therefore works on this area are necessary and this project will focus part of the efforts on them:

1. The first requirement is to provide the necessary infrastructure for interoperability of radio communications bearing in mind that there are two different candidates for the radio technology, as discussed in Section 3.3.
2. The second addressed issue is the provision of a complete system that enables to verify the authenticity of the transmitted messages through a Public Key Infrastructure (PKI) that distributes and revokes certificates.
3. The third objective is to develop an anti-tamper Hardware Security Module (HSM) to store these certificates in the vehicle.
4. Next, as most ITS messages rely on the geographic position of the vehicles, CAMEL has to ensure that vehicle's position information is trustful and reliable. For this reason, a location spoofing attack detection system will be developed.
5. Finally, the fifth objective is to provide a system that enforces vehicles privacy through appropriately choosing the instants in which vehicles should change the certificate used to sign their packets to prevent being tracked.

In line with these objectives, below we list the main open issues in terms of security and privacy that will be examined through the use case discussed in this section:

- Different radio technologies for V2X communications
- Authentication of messages through a Public Key Infrastructure
- Storage of cryptographic material in the vehicle through a Hardware Security Module
- Vehicle location spoofing
- Vehicle tracking using its signature certificates

The use case presented here is revolving around the connected mobility pillar, but it also enhances the autonomous mobility domain discussed in the previous section, as it enables vehicles to communicate with their surrounding environment (emergency calling, data mining, remote services, entertainment data, geo positioning, etc.).

An overview of potential attack surfaces of the connected mobility pillar is shown in Figure 19.



Figure 19: Overview of the attack surface of the Connected Mobility Use Case

For the cooperative and connected mobility domain, amongst the top priorities is to ensure the integrity and authenticity of the exchanged information. Some of the most important security threats of this domain are described in Table 13.

Description	
1.	Attacks on backend server. An attacker can compromise a backend server and use it to attack the connected cars. An attacker may launch a DoS attack on backend servers to disrupt their services. An attacker may target sensitive data at the server or information in other parts of the cloud. For example, mobile apps are used to allow a user to query the status and control the car from his/her smartphone. Insecure APIs at the backend allow an attacker to interact with the car using falsified API requests.
2.	Attacking a car using V2X communication channels. An attacker may spoof V2X messages, tamper with transmitted data or code, attack data integrity, exploit the trust relation, gain unauthorized access to data, jam the communication channel on the protocol or RF level and inject malware or malicious V2X messages. For example, non-secure protocols such as HTTP are sometimes used for V2X communications. Even when TLS/SSL is used, if the client software does not properly check the server certificate, an attacker can launch a Man-in-the-Middle attack to steal the user's credentials to further control the car.
3.	Attacking a car by exploiting software update. An attacker may compromise the Over-the-Air (OTA) updates or local and physical software update process, manipulate the software before the update process, or even steal cryptographic keys to compromise code signing. For example, the 2014 Jeep Cherokee was remotely hacked by updating the Renesas V850 firmware to allow the compromised telematics unit to send messages directly to the ECUs on the CAN bus.
4.	Social engineering exploits vulnerabilities and weaknesses introduced by human errors. An attacker may trick an owner, operator, or maintenance engineer to unintentionally install malware or change the setting to enable an attack. An attacker may also exploit errors in system configuration or usage.

5.	Attacking vehicle interfaces and functions for external connectivity. An attacker may access and manipulate functions designed to remotely operate systems or provide telematics data, short range wireless systems and sensors, and applications with poor software security. An attacker may also utilize physical interfaces such as USB or diagnostic port, or even media connected to the car as a point of attack. For example, connected cars rely on network devices with TCP/UDP ports to interact with the outside world. Even the IP address of a connected car is protected by network separation provided by network operator, open ports and services with weak or no authentication pose security risks. An attacker can remotely scan and access the open ports and exploit the services as an entry point to the on-board system. In addition, the CAN bus can be accessed physically through the OBD port, charging station, or a mechanic's computer.
6.	Attacks on in-vehicle network or software of on-board systems. An attacker may extract data and code, manipulate vehicle data, erase data and code, inject malware, inject or overwrite existing software, disrupt system operation, and manipulate vehicle parameters.
7.	Attacks that exploit security flaws in system design. An attacker may break the encryption due to insecure cryptographic design such as lack of encryption, weak key strength, or the use of deprecated cryptographic algorithms. Bugs in software and hardware may provide the attacker exploitable vulnerabilities and means of access or privilege escalation. Poor network design such as weakness in internet-facing ports and internal network separation also pose security risks. Crypto systems in the car should last for a long period of time. Lack of crypto-agility, i.e. not being able to upgrade broken or obsolete cryptographic systems over time, may affect the whole security status.
8.	Attacks on privacy or data loss and leakage. V2X communication packets may contain identifiable information. Some of the information may be anonymized or pseudonymized. However, an attacker may still be able to intercept the V2X packets, footprint and track a car's movement over a certain period and area and re-identify the user. Personal data may be transferred to third-party service providers in V2X communications. Sensitive data from cars may be lost or leaked due to physical damage, failure of IT components, or change of ownership.
9.	Physical manipulation of on-board systems to enable an attack. Manipulation of OEM hardware or adding unauthorized devices may enable a remote attack afterwards.

Table 13: Overview of attacks in the connected mobility domain

The scenarios targeted by the project are based on the CARMEL consortium knowledge, available resources and showcasing capacity. We will examine three main attacks:

- Geolocation attacks: using a type of man-in-the-middle attack called "location spoofing" attack.
- V2X message attacks: man-in-the-middle attack and vehicle bus (through radio interface) attack.
- OBU Tampering attack: attack on key certificates storage, malicious firmware update, attack on exploiting OSS vulnerabilities.

Those three attack scenarios are shown in Figure 20 (a burglar icon is placed where the vulnerability intended to be attacked is shown).

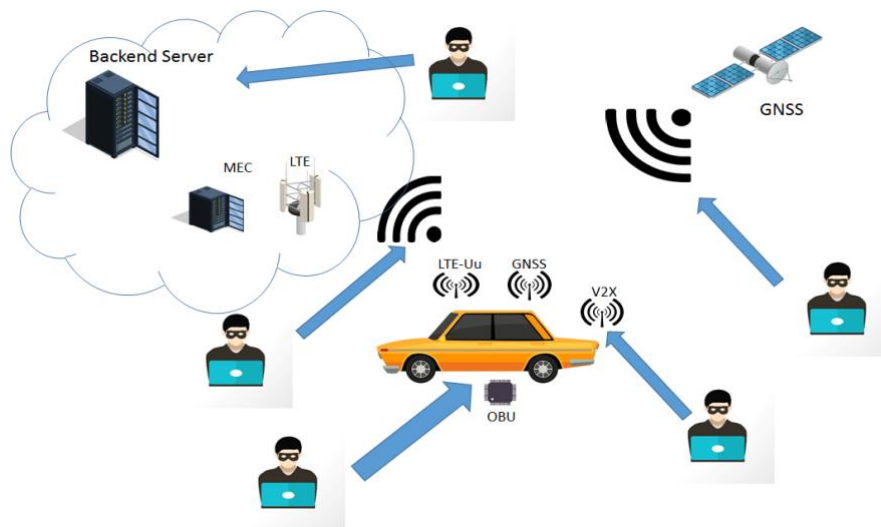


Figure 20: Attack scenarios contemplated in the connected mobility pillar

3.2 Scenarios Description

Keeping in mind the threats discussed above, a set of relevant scenarios, which are also feasible in terms of timing and budget, have been identified for Pillar 2, presented in Table 14.

No.	Scenario Name	Short Description
1.	Location Spoofing Attack	<p>The attacker is able to jam the satellite signals and the connected car does not have satellite-based location (e.g., GNSS/RTK).</p> <p>The attacker is able to spoof the satellite-based location of the connected car.</p> <p>CAMEL system aims to detect the jamming and location spoofing attack.</p>
2.	Attack on the V2X message transmission	<p>A malicious attacker transmits fake CAM and DENM messages or tries to track a specific vehicle.</p> <p>A malicious attacker tries to track a vehicle though the digital certificate used to sign transmitted messages.</p>
3.	Tamper attack of vehicle's OBU	<p>The attacker is able to tamper an OBU physically by accessing the vehicle.</p> <p>The attacker could have acquired another OBU (e.g. aftermarket sample) in order to study its vulnerability beforehand.</p>

Table 14: Overview of scenarios to be examined in Pillar 2

3.2.1 Location Spoofing Attack

A location spoofing attack attempts to deceive a GNSS/RTK receiver by broadcasting incorrect satellite signals, structured to resemble a set of normal satellite signals (e.g., GPS, GLONASS, GALILEO, etc.). These spoofed signals may be modified in such a way as to cause the receiver to estimate its location to be somewhere other than where it actually is. One common form of a location spoofing attack begins

by broadcasting signals synchronized with the genuine signals observed by the target receiver. The power of the counterfeit signals is then gradually increased and drawn away from the genuine signals.

This type of attack has already been successfully carried out in several scenarios, i.e., against boats or Unmanned Aerial Vehicles (UAVs) in references like [55][56]. Following such philosophy, in CAMEL, the attack is carried out thanks to fake satellite signals transmitted by Software Defined Radio (SDR) hardware. Figure 21 shows a possible implementation where a UAV (e.g., commercial-grade drone) is used for transmitting counterfeit signals. Other candidate implementations for demonstration include static transmitters carried by the attacker covering a specific target area, e.g., a crossroad.

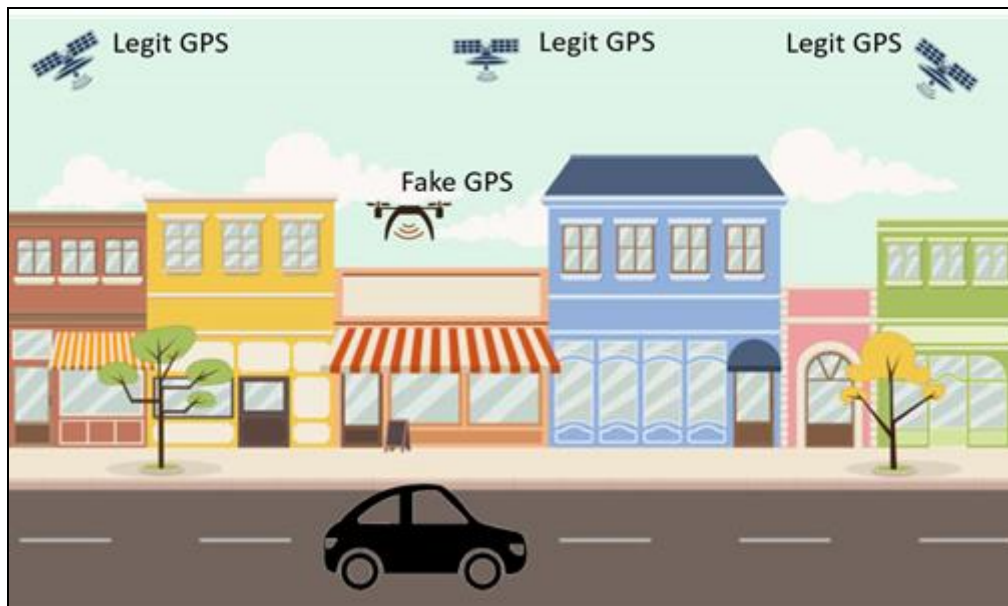


Figure 21: A possible implementation for the location spoofing attack

For this attack, the CAMEL system will be able to detect when the satellite signals are spoofed thanks to a parallel stream of vehicle locations that does not rely on satellite signals, but rather in-car measurements readily available through the vehicle's CAN bus, which we call hereinafter the CAMEL secondary location stream. Such secondary location stream is based on a Bayesian filtering technique, which consists of two basic steps: (i) the prediction step and (ii) the update step. With Bayesian filtering, the motion of the vehicle is described through the characterization of the underlying physical laws, e.g., with a bicycle model, and the prediction on the future vehicle locations is obtained through on-board sensors, e.g., with the Inertial Measurement Unit (IMU) readings. In the update step, the forecasted vehicle locations are then fused with satellite-free global location measurements.

In order to detect the location spoofing attack, in CAMEL, the secondary location stream will be compared with the obtained satellite-based locations. When the difference between the two sets of location measurements exceeds a predefined threshold, an alarm will be raised, and the location spoofing attack will be detected. The alarm will be communicated through the CAMEL connectivity infrastructure to the PKI infrastructure, which then will take appropriate countermeasures, e.g., revokes the certificate of the attacked vehicle.

The secondary location stream will be computed by an application within a container on the anti-hacking device in the CAMEL system, which will be installed inside the vehicle. Such an application will retrieve information regarding on-board sensor readings from the CAN bus and will fuse such information with satellite-free global positioning measurements thanks to a Bayesian filtering technique. An attack will be identified, and an alarm will be triggered, by comparing the coherence of the obtained secondary location stream with the location information obtained by the satellite receiver (Figure 22). To this end, the system will take the secondary location stream as the ground truth and will exploit the knowledge of its error's covariance to validate the satellite-based location. If the probability of the

measured distance between the averages of the two multivariate distributions is above a given pre-defined threshold, an alarm will be raised, i.e., a location spoofing attack will be detected.

Notably, the solution adopted by the CARMEL system is modular, and each block could be modified based on the available on-board sensors and on the available satellite-free global positioning measurements. As an example, the vehicle state model could be defined depending on the most accurate sensors present in the vehicle or on the sensors leading to the most accurate location predictions. In the same way, any satellite-free Global Positioning Measurement could be used, as far as it is possible to determine its error covariance matrix.

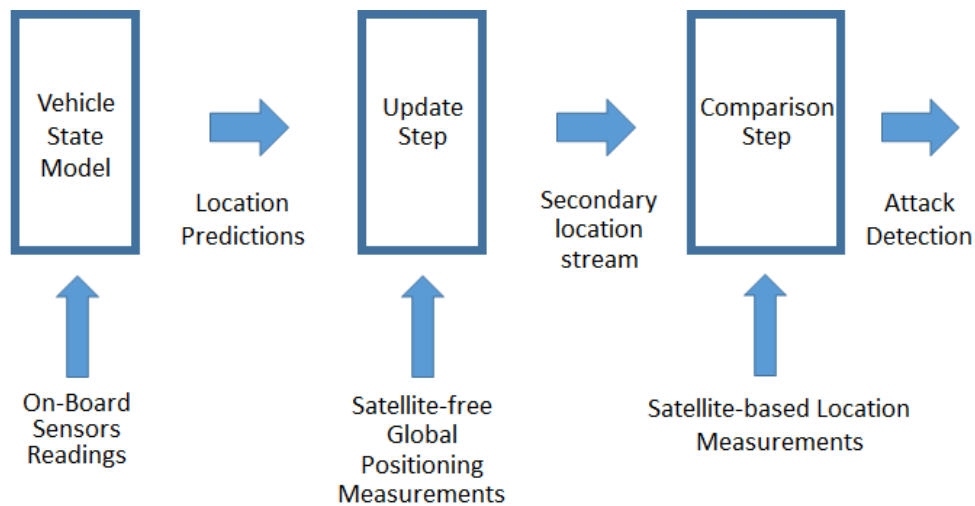


Figure 22: Block Diagram of the satellite-based location integrity check application

3.2.2 Attack on the V2X Message Transmission

This scenario deals with the security of V2X message transmission. It will be used to test the PKI architecture, the certificate distribution model and the signing functions of the transmitted messages.

Additionally, as V2X messages are transmitted digitally signed, an important aspect is to preserve privacy. This is achieved using pseudonyms instead of real identifiers for the continuous message transmission. Standards recommend changing pseudonyms at given intervals. However, knowing the position of vehicles and the interval used in pseudonym renewal, tracking by an attacker becomes trivial.

The common solution to improve privacy is to randomise the moments when pseudonyms are renewed and, optionally, insert silence periods of a few seconds. However, the naive insertion of these silence periods may affect the performance of safety applications. The authors in [57] propose a set of pseudonym renewal randomisation strategies using IEEE 802.11p radios, and study how these strategies impact the success rate of an intersection collision detection application. In CARMEL, we plan to further extend these strategies proposing an algorithm based on Machine Learning to optimize the moments when the pseudonyms are renewed.

The scenario comprises four test cases where the security in communication functionalities of the OBU will be tested and analysed. In particular, CARMEL will deploy:

- Test case 1: The attacker is a fake vehicle that generates messages with some invalid data.
- Test case 2: The attacker is a fake vehicle that sniffs and replays messages of compliant vehicles.
- Test case 3: The attacker is a compliant vehicle but supplanting identity (example: a normal vehicle sends information as an ambulance).

- Test case 4: The attacker is a vehicle trying to track another one which changes the AT using the proposed changing algorithm.

In all cases, the receiving vehicle will need to check the authenticity of the incoming messages and discard the non-compliant ones (fakes or not authorized).

To test this scenario the following elements of the architecture will be involved:

- The PKI will be used to authenticate/authorise vehicles and distribute Authorization Tickets (AT) to the OBUs, to be able to sign their own V2X messages.
- 3 types of OBUs: One receiver OBU, one transmitting compliant OBU and one attacker OBU.
- MEC (Multi-access edge computing): In the case where V2X messages are transmitted V2V, the receiving OBU will be in charge of detecting non-compliant messages. Nevertheless, in the case where messages need to be relayed using the infrastructure (in order to interoperate different radio technologies or to forward messages to other geographic areas), the MEC, before forwarding the message, will check its authenticity and will discard messages from the attacker.
- Anti-Hacking device: To compute the moment in which an OBU should change the pseudonymous certificate.

3.2.3 Tamper Attack of Vehicle's OBU

This scenario reflects the fact that even though a network vector attack can be more impactful than a physical attack, directed attacks to a given physical unit of a vehicle are also potentially dangerous. Such attacks can be directed towards a specific target negatively impacting its privacy and security while also indirectly impacting its safety.

There are three techniques, at the OBU level, which can be used to enhance the level of security, as depicted in Figure 23 .

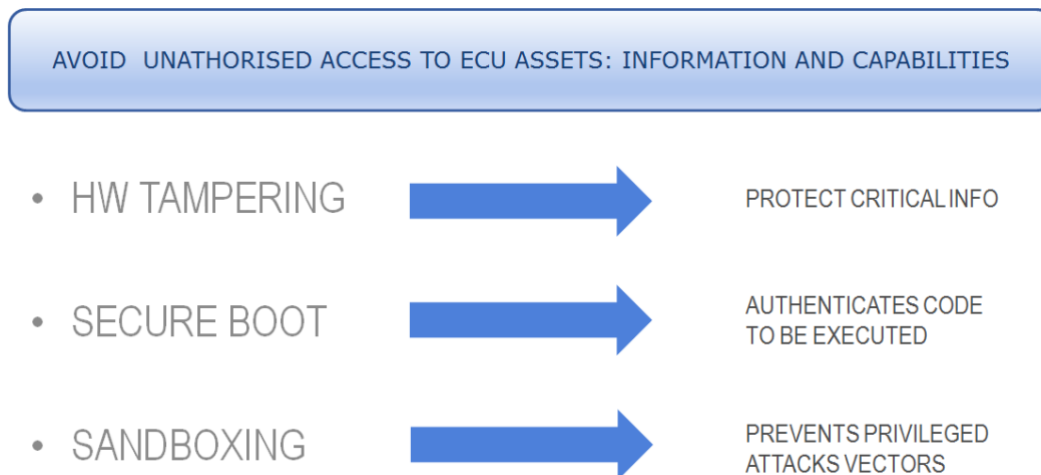


Figure 23: Security techniques at the OBU level

Hardware (HW) tampering refers to any means through which the components itself can be manipulated. The severity of the tampering can range from just naive manipulation such as breaking a seal to dangerous manipulation resulting in accessing privileged information. Secure boot is a technique which prevents non-authorized software (SW) to be executed. This technique aims to prevent tampering through malicious software execution. Sandboxing provides an additional level of security which can prevent vulnerabilities in authenticated software to be exploited. Every application is executed in its predefined and isolated sandbox environment where it cannot interact or affect the execution of other applications. This mainly prevents ROP attacks.

In the context of CARMEL, the main R&D development will be focused on HW techniques to detect and prevent tampering. Nonetheless, secure boot and sandboxing approaches will be implemented as well.

Since the OBU is the gateway to the vehicle's network communications, its protection should be the top priority so as to prevent it from becoming the weakest link in the vehicle's security chain.

In order to do so, it is necessary to consider splitting the OBU's complexity in several layers and applying a different security approach per layer, according to its specific needs.

There are five (plus one) layers to be considered, as illustrated in Figure 24:

- Hardware layer: all HW techniques to prevent HW tampering. This includes, but not limited to, HSM module.
- Hypervisor: SW layer which commonly provides an abstraction of the OS.
- OS Control Access
- Network protection
- Application Sandboxing
- OTA update ability

ECU Security – Layered Architecture 5+1

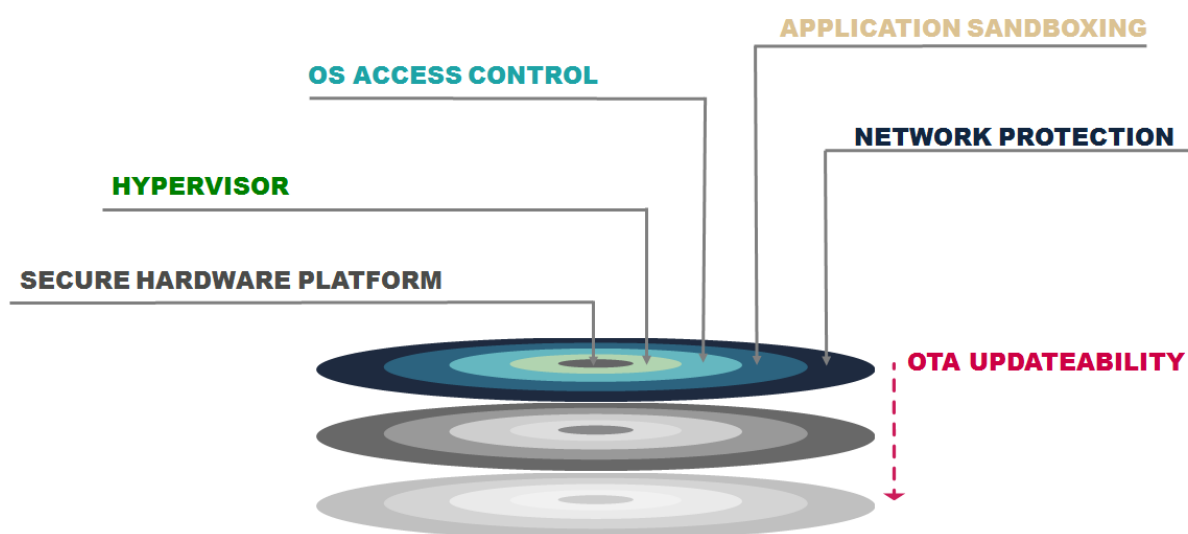


Figure 24: The layered architecture of OBU security

The characteristics of an OBU tampering attack are listed in Table 15.

Execution Steps	<ul style="list-style-type: none"> • The attacker gets hands-on access and opens the enclosure of the OBU. (OBU is disconnected) • OBU detects tampering and triggers anti-tampering mechanism:
-----------------	---

	<ul style="list-style-type: none"> ○ OBU triggers mechanism to enter in secure-state ○ Confidential information is protected against tampering
Data Flow	<ul style="list-style-type: none"> • OBU detects tampering signal • Tampering signal triggers secure-state actions <ul style="list-style-type: none"> ○ Zeroisation of private keys ○ Zeroisation of any other confidential data • OBU enters secure state • In this secure state the system will inform the PKI server, through any channel available, to revoke the certificates.
Assumptions	<ul style="list-style-type: none"> • The malicious attacker has studied the OBU (e.g. has obtained some pictures, diagrams and knows where the SE is placed). • OBU is disconnected from main supply • Every OBU has unique secure set of private keys

Table 15: OBU Tampering Attack characteristics

3.3 Enabling Infrastructure and Overview of Cyberattacks

In this section we will refer first to the type of malicious attacker we are envisioning for the kind of attacks that we will protect in CAMEL. Note that the attacker is also present in the other use cases addressed by the project, but in the Connected Mobility scenarios it gains a more active role. Table 16 presents the different types of malicious attackers.

Type of Malicious Attacker	Short Description
Cyber Criminals	Individual or teams of people who use technology to commit malicious activities on digital systems or networks with the intention of stealing sensitive company information or personal data, and generating profit
Hactivists	Hactivists are individuals or groups of hackers who carry out malicious activity to promote a political agenda, religious belief, or social ideology. According to Dan Lohrmann, chief security officer for Security Mentor, a national security training firm that works with states said "Hactivism is a digital disobedience. It's hacking for a cause." Hactivists are not like cybercriminals who hack computer networks to steal data for the cash. They are individuals or groups of hackers who work together and see themselves as fighting injustice
State sponsored attackers	State-sponsored attackers have particular objectives aligned with either the political, commercial or military interests of their country of origin. These types of attackers are not in a hurry. The government organizations have highly skilled hackers and specialize in detecting vulnerabilities and exploiting these before the holes are patched. It is very challenging to defeat these attackers due to the vast resources at their disposal.
Insider Threats	The insider threat is a threat to an organization's security or data that comes from within. These types of threats and attacks are usually committed by employees or former employees, but may also arise from third parties, including contractors, temporary workers, employees or customers. Insider threats can be further categorized into the following:

	<ul style="list-style-type: none"> ○ Malicious threats are attempts by an insider to access and potentially harm an organization's data, systems or IT infrastructure. These insider threats are often attributed to dissatisfied employees or ex-employees who believe that the organization was doing something wrong with them in some way, and they feel justified in seeking revenge. Insiders may also become threats when they are disguised by malicious outsiders, either through financial incentives or extortion ○ Accidental threats are threats which are accidentally done by insider employees. In this type of threats, an employee might accidentally delete an important file or inadvertently share confidential data with a business partner going beyond company's policy or legal requirements ○ Negligent threats are threats in which employees try to avoid the policies of an organization put in place to protect endpoints and valuable data. For example, if the organization has strict policies for external file sharing, employees might try to share work on public cloud applications so that they can work at home. There is nothing intentionally wrong with these acts, but they can open up to dangerous threats nonetheless.
--	---

Table 16: Overview of Malicious Attacker types

The infrastructure required for the deployment of the Connected Mobility use case consists of the building blocks presented in Table 17.

Component	Short Description
Malicious attacker	Any type of actor that can negatively impact the vehicle or other relevant infrastructure. See Table 16 for further descriptions of malicious attackers.
Cooperative car	<p>V2X-enabled communications car. It is capable of communicating with other cars and infrastructure relevant data such as position, speed, etc. and detect security attacks.</p> <p>It is composed of:</p> <ul style="list-style-type: none"> • OBU. • Anti-Hacking device.
Fixed infrastructure	<p>Consisting of:</p> <ul style="list-style-type: none"> • eNB: C-V2X base station. • RSU: IEEE 802.11p fixed station. • PKI servers. • MEC. <p>It has to provide:</p>

	<ul style="list-style-type: none"> Multi-Technology V2X Communications interoperability: It is foreseen that there will be vehicles and users communicating with different radio technologies. The infrastructure needs to perform the correspondent actions to enable that all messages get to all required destinations. Distribution of Revoked Certificates Lists in real time: Certificate revocation is the consequence of any misbehaviour or malicious act detection. Whenever the anti-hacking device deployed in the vehicle or any other application executed in the fixed infrastructure detect a fraudulent or misbehaving action performed by a vehicle, the system will decide if this vehicle has to have its certificates revoked. If affirmative, a new process to inform about this fact to other vehicles in the system has to be deployed. In particular, the infrastructure has to provide a mechanism to distribute Revoked Certificates Lists in real time to the rest of vehicles of the system.
Outside Infrastructure	<ul style="list-style-type: none"> Public parking Workshop Private parking

Table 17: Overview of building blocks for the Connected Mobility pillar

3.3.1 Cooperative Cars

Standard cooperative cars are equipped with an OBU (On-Board Unit) which provides all secure communications functionalities. The objective of CAMEL is to develop a completely functional OBU that complies with current security regulations, plus an “Anti-Hacking Device”, which runs processes able to detect hacking attempts and functional misbehaviours using Machine Learning algorithms and techniques (Figure 25). The Anti-Hacking device is used to counter attacks in the “Location Spoofing Attack” scenario, focusing on the vehicle's satellite-based location service.

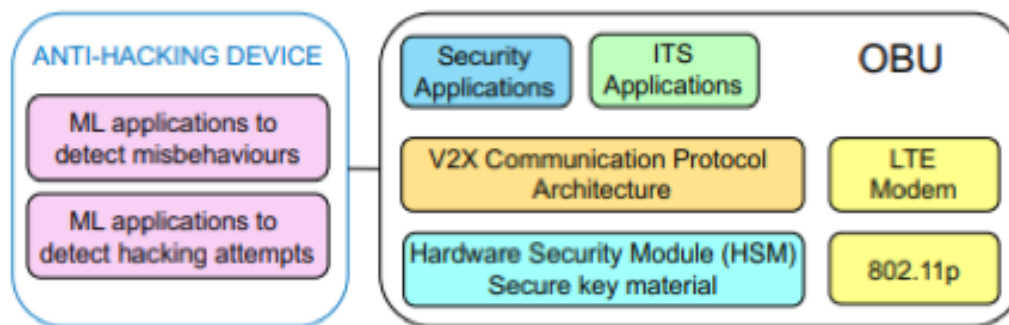


Figure 25: Cooperative car equipment overview

In the Secure multi-technology OBU architecture, we can distinguish the following main elements:

- Hardware Security Module (HSM):**

One of the possible attack vectors to V2X infrastructure is to steal sensitive data or cryptographic keys from the vehicle's OBUs. In order to counter this attack, trustworthy, unforgeable, and non-copyable identities must be established for the V2X communication partners. One way to achieve this goal is to integrate a Hardware Security Module (HSM) into the OBU that serves as a repository for private key data (for authentication and encryption purposes) as well as a cryptographic processor for sensitive operations. This fully secured hardware platform requires additional protection at different stages of the device's operation: i) booting only secure firmware that is known to be obtained by a trusted source, ii) resilience against hardware modification, iii) resilience against software modification (after booting the trusted firmware), and iv) additional protection of sensitive assets at runtime.

The HSM is the component used for preventing and countering attacks of the “Tamper Attack on Vehicle’s OBU” scenario.

- **Radio interfaces (IEEE 802.11p and LTE-Uu):**

All vehicles require to connect to the PKI servers in order to obtain the pseudonymous authorization tickets before being able to transmit ITS messages. Although it is possible to download these tickets once every several days in different places (home through a Wi-Fi connection, petrol stations, mechanical garages, ...) the most common situation is when the vehicle has a cellular interface to connect itself to the fixed network. The technology of this interface will evolve as cellular networks evolve. Therefore, as for CAMEL’s deployment, LTE-Uu is going to be used, but it can be extended to 5G NR, or 6G in a near future.

Additionally, and in order to reduce latency during ITS message transmission between vehicles, direct V2V connections are preferred than V2I. V2V connections can be performed using IEEE 802.11p or LTE-PC5 radio technologies but, nowadays, IEEE 802.11p is much more commercially available. Consequently, in CAMEL project we will use IEEE 802.11p for V2V connections. Lastly, we consider that a real road scenario contains two types of cars, those that are able to perform V2V connections (they have two radio interfaces: LTE-Uu and 802.11p), and those that are only able to connect to the fixed network (they have a single radio interface: LTE-Uu).

Currently there is not a clear radio technology to be used for V2X communications. Up to now, IEEE 802.11p has been the de facto wireless technology standard for V2X communications. It is a relatively mature technology and has already been validated by over a decade of field trials. Despite that, IEEE 802.11p, which uses Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), suffers from a high level of collisions under heavy traffic conditions, mainly due to hidden terminal situations.

Long-Term Evolution (LTE) based V2X from the Third Generation Partnership Project (3GPP) is a relatively new alternative to IEEE 802.11p-based V2X communications. The first version of LTE-V2X, also known as Cellular-V2X (C-V2X), was published in June 2017 under Release 14, which came with numerous enhancements to the existing Device-to-Device (D2D) communications in order to accommodate vehicular communications. The proposed enhancements include a new arrangement of the resource grid of the physical layer and two types of D2D channel access mechanisms: i) a mechanism coordinated by the evolved NodeB (eNB), named Mode 3, and ii) a distributed mechanism, where User Equipments (UEs) access the channel on their own, named Mode 4. Moreover, LTE-V2X employs different radio interfaces: i) interface between the vehicle and eNB, named LTE-Uu, and ii) interface between vehicles, named LTE-PC5.

Current specifications state that IEEE 802.11p and LTE-PC5 communications take place in one channel inside the unlicensed frequency band 5855 MHz - 5925 MHz named ITS-G5, while LTE-Uu uses part of the licenced spectrum assigned to the operator that owns the eNB.

Major Original Equipment Manufacturers (OEMs) are starting to roll out V2X capabilities, but there is not yet an industry-wide consensus about the best communications technology. In this regard, Volkswagen begins to manufacture a new 2020 Golf equipped with the system Car2X which uses the more mature 802.11p radios. Preliminary V2X services including the continuous dissemination of a vehicle position and speed, through the use of ETSI-G5 Cooperative Awareness Messages (CAM), or the notification of road events through the use of ETSI-G5 Decentralisation Event Notification Messages (DENM), have been demonstrated, using IEEE 802.11p, in various EU funded projects including DRIVE C2X, C-ROADs or PRESERVE. However, other OEMs supported by telecom operators and vendors, have gathered around the 5GAA to promote the adoption of the C-V2X.

At present, both the IEEE and the 3GPP, continue to enhance their support for vehicular communications. The IEEE recently launched the 802.11bd working group, which will update the physical layer of 802.11p to the one used in 802.11ac (Very High Throughput), while adding enhancements for high mobility (500 km/h), support for 60 GHz operation, and longer range (double than 802.11p) [54]. However, in order to maintain backwards compatibility, 802.11bd reuses the same channel access mechanisms than 802.11p. On the other hand, the 3GPP is currently defining in Release 16 C-V2X extensions for the 5G-New Radio technology, known as NR-V2X. This technology though does not target day-1 safety services in the ITS -G5 band, will focus on value added services using licensed spectrum, such as teleoperated driving or platooning [54].

Under these circumstances, the real scenarios that we will face in the coming years will be those where vehicles, and other road users, will use different radio technologies. Therefore, CAMEL addresses

both the interoperability of IEEE 802.11p and C-V2X technologies, and the securitisation of V2X communications. Interoperability between both technologies is implemented using infrastructure support, through the use of Multi-Access Edge Computing (MEC) the capabilities of which are described later in the section.

- **V2X Communication Protocol Architecture:**

This element contains the software package that enables the OBU to generate Facilities layer messages encapsulated on Basic Transport Protocol (BTP) and GeoNetworking protocol (GN). CARMEL is going to use the open source framework Vanetza [58], updated accordingly, so it is able to perform all security and privacy related functionalities.

- **Security applications:**

This element contains all software functions in charge to interact with the PKI infrastructure and manage the registration and authorization procedures, as well as to obtain the pseudonymous authorization tickets and store them into the HSM according to [59]. These applications will use the LTE-Uu channel to transmit their information.

- **ITS Applications:**

This element represents any ITS application that the vehicle is executing. As for the CARMEL's testbed it is planned to use applications sending and receiving Cooperative Awareness Messages (CAM) and Decentralized Event Notification Messages (DENM) messages.

3.3.2 V2X Infrastructure

Besides the internal elements in the car, in case of connected and cooperative mobility we should also take into account the fixed telecommunication infrastructure. The secure and interoperable V2X communication system is supported by a Secure Multi-Technology V2X Telecommunications Infrastructure, illustrated in Figure 26 which provides three main functionalities:

- It enables interoperability between C-V2X and IEEE 802.11p vehicles.
- It enables hosting of additional security functions embedded into the MEC infrastructure. It performs all vehicle registration and authorization through the PKI infrastructure.

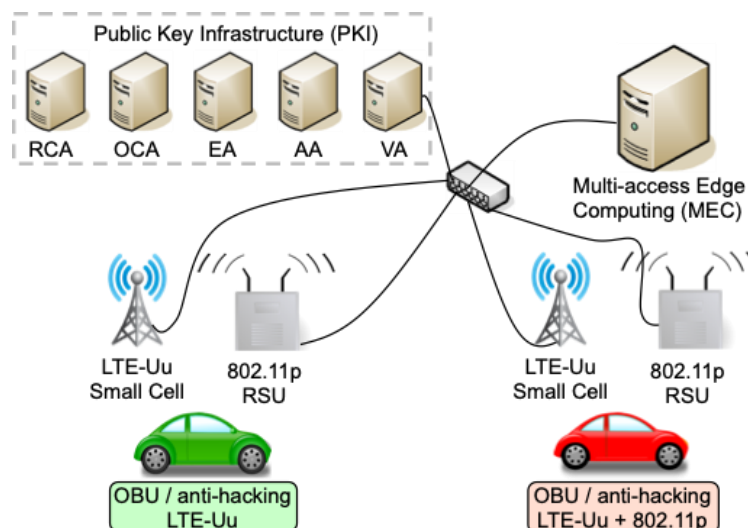


Figure 26: Overview of Secure Multi-Technology V2X Telecommunications Infrastructure

The main components of the Secure Multi-Technology V2X Telecommunications Infrastructure are:

- **Radio equipment:**

It is deployed using:

- Accelleran LTE Small Cells Base Stations (eNBs)
- IEEE 802.11p RSUs.

All of them will use a fixed network to be connected to the MEC and PKI infrastructures.

- **Public Key Infrastructure:**

This element basically comprises five different servers:

- The Root Certification Authority (RCA). This server, offline for security reasons, must be managed only by authorized personnel, and contains the root certificates for the entire PKI infrastructure.
- The Online Certification Authority (OCA). This is an online server, signed by the RCA. Its main responsibility is to sign the different lower authorities in the PKI infrastructure, described in Figure 27.
- Enrolment Authority (EA). This entity it's in charge of providing the necessary certificates at the enrolment phase. This authority can only be managed from certain authorized locations (the manufacturers, authorized points, etc.) and can provide enrolment certificates, that are used by the car to ask for pseudonym certificates (or authorization tickets).
- Authorization Authority (AA). The AA is the entity that manages the pseudonym certificates. These certificates are issued for ensuring privacy of the cars within the PKI infrastructure.
- Validation Authority (VA). The VA is present to provide a way to ask the PKI infrastructure which certificates are revoked. It provides a CRL list with the revoked certificates, along with an online service that returns the state of a specific certificate in real-time.

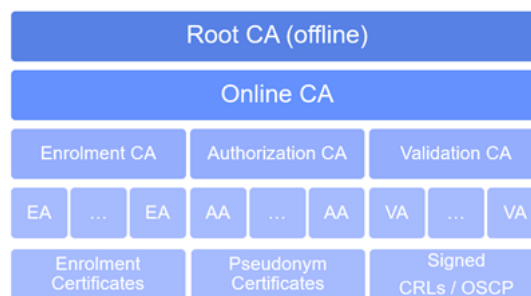


Figure 27: Certificate chain in CARMEL PKI infrastructure

In the enrolment phase, an ITS station requests enrolment credentials to an EA such that it can be trusted to function correctly by other ITS stations. In the authorization phase, an enrolled ITS station requests pseudonymous authorization tickets (AT) to an AA to get specific permissions (e.g. to access to a specific service/resource) ensuring confidentiality and privacy. Internally, the AA will ask the VA to check if the request is authorized. Finally, EA and AA can be trusted by ITS stations through validating their authenticity with the RCA [59]. This process is depicted in Figure 28.

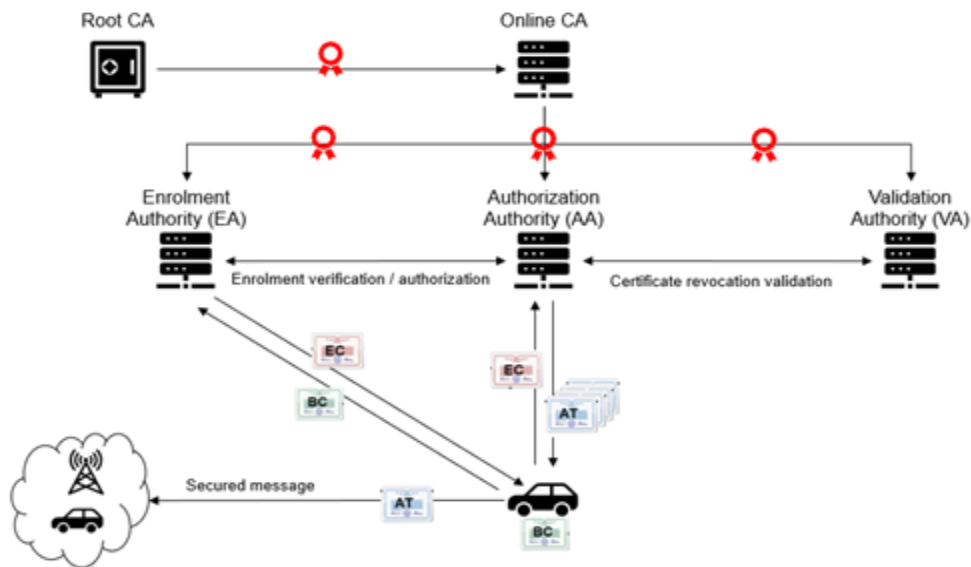


Figure 28: CARMEL PKI Infrastructure

The PKI infrastructure is the enabler to provide security to V2X message transmissions and will be the basis to build and test the “Attack on the V2X message transmission” scenario.

- **MEC infrastructure:**

The Multi-access Edge Computing (MEC) server will be deployed to accommodate the required functions to run at the edge of the network, following, as much as possible, the ETSI MEC framework standardization (Figure 29).

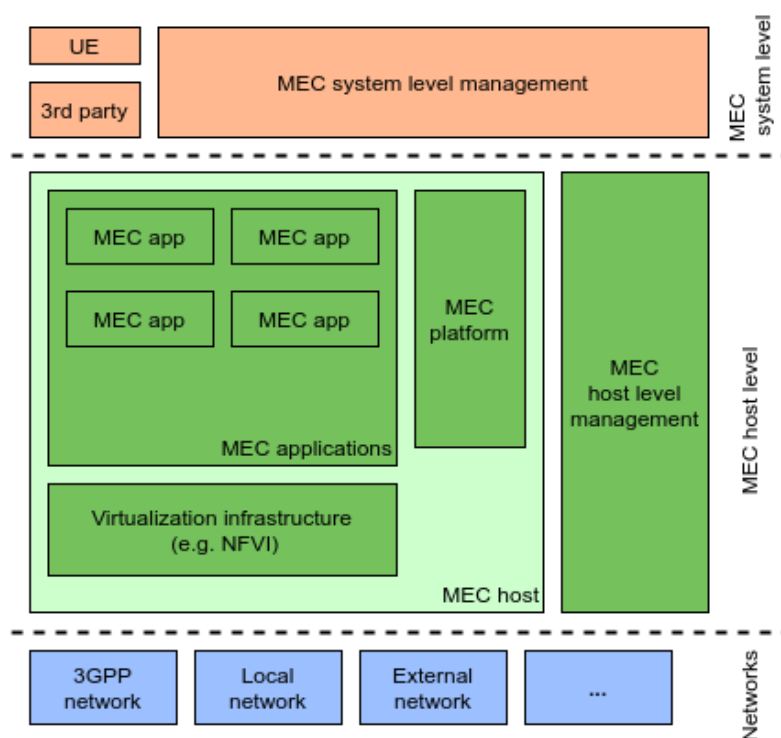


Figure 29: ETSI MEC Framework

The *MEC system level* management retains the global view of the whole MEC system, i.e. the collection of MEC hosts/servers, being the entity responsible to orchestrate and manage the MEC platform elements, the MEC app rules and requirements and the MEC app lifecycle operations.

At the *MEC host level*, the MEC host/server sits along with its associated management subsystem. The MEC host level management module is a gateway, used by the above layer orchestrator, which enables the management of the MEC host and its running apps. The MEC host is constituted by the platform and the virtualization infrastructure on top of which the functions/apps will run.

The *Networks* layer enables the physical connectivity to cellular networks, local networks and external networks (e.g. internet).

To pursue the requirements of CAMEL, it is intended to provide a framework that enables the deployment and management of MEC applications in a dynamic and flexible way, comprising:

- Dashboard module - provides a user interface to deploy and manage MEC apps, to close the gap between the user and the orchestrator.
- Orchestrator - manages MEC servers and their applications.
- MEC Server - contains compute and network resources, on top of which the MEC host will run and provide a virtualized infrastructure to run applications.

The MEC server provided in the project offers both the virtualized environment for MEC apps to be instantiated on, as well as the required V2X physical interfaces to the different underlying radio access technologies. It hosts applications for the following functionalities:

(i) Message forwarding to provide interoperability between multi-technology communications:

This functionality deals with the fact that during some transition time there will be cooperative ITS users using different access technologies. CAMEL considers IEEE 802.11p working in the Control Channel (CCH) of the ITS-G5 band (5,9 GHz) and LTE-Uu radio working in one operator's band, which should be made interoperable thorough functionalities of the infrastructure.

The MEC, relying on the different fixed radio equipment, will receive messages transmitted using one technology and will replicate them in the other technology, while conserving the original message signature.

CAMEL's is going to implement two different interoperability test cases:

- Test case 1: The message is sent by one vehicle in one radio technology and the infrastructure forwards this message in the same region with the other radio technology (Figure 30).
- Test case 2: The message is sent by one vehicle in one radio technology and the infrastructure forwards this message in another region with both radio technologies (Figure 31).

The MEC will have rules to decide which messages to forward and where to forward them. These rules will enable filtering and dropping message according to their importance, age, region of interest, type of destination vehicles, etc.

The tests described above will be focussed on two main key performance indicators:

- The increase of the transmission delay between the direct communication (V2V) and the communication that requires the infrastructure (V2I2V). Delay measurements will be performed using the GPS clock of the transmitting and receiving vehicles.
- The forwarding capacity of the system expressed in number of messages that the infrastructure is able to process. For this measurement it is necessary to transmit a high number of V2X messages. CAMEL will develop software that will run on one or two additional devices which will be able to emulate the transmission of many vehicles simultaneously.

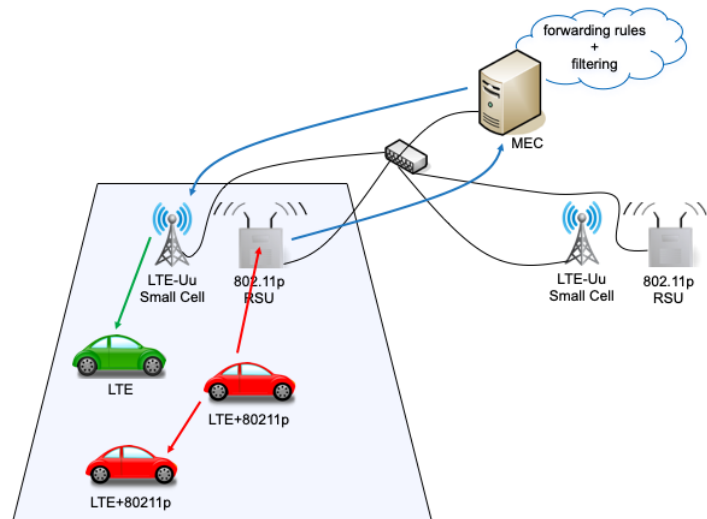


Figure 30: Interoperability between radio technologies

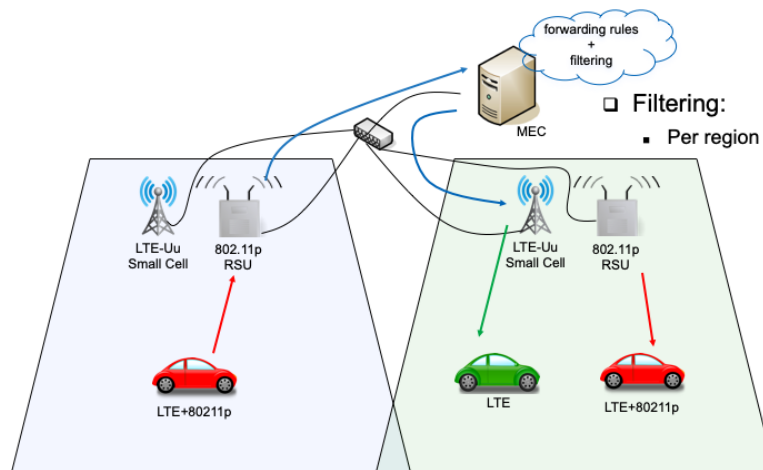


Figure 31: Interoperability between regions of interest

(ii) Security function to distribute Revoked Certificates Lists (CRL) in real time to vehicles:

The use of pseudonymous AT involves how these certificates are distributed among vehicles, how they are stored and how they are revoked. Project PRESERVE dealt with these issues and provides some proposals [60][61]. Eventually, vehicles which already have some of those valid pseudonymous AT stored, will need to have them revoked due to misbehaviour or administrative reasons. In TR 102 893 V1.2.1 (March 2017), ETSI proposes to distribute information about compromised units, but this will cause congestion in the 5,9GHz ITS-G5 channels or it will require an additional communication channel. For this reason, at the present time, the Certificate Policy from the EU C-ITS platform does not foresee revocation of single C-ITS stations. Instead a “revocation by expiry” is specified, which means that short term certificates for communication have a rather short validity time, e.g. one week, and after that defined period they are not trusted anymore [62]. In this mechanism, it is important to limit the maximum preloading time to a reasonable time span. Preloading defines how long in advance short-term certificates, which are valid for a specified period and are intended for later use, can be loaded onto the vehicle. A too long preloading period, e.g. of several years, would pose a risk to the C-ITS trust system, since these certificates cannot be individually revoked later on.

Nevertheless, in a highly reliable system, this approach is not effective and some method to distribute Certificate Revocation Lists (CRL) will soon be needed. Some authors have already published proposals, for instance: there are proposals for a versatile and low-complexity framework to facilitate the distribution of the CRL issued by the Certification Authority. Under these circumstances, CARMEL will develop a system to distribute CRL to vehicles in real time.

This functionality comprises two test cases:

- Test case 1: A process running in the MEC detects a critical situation in one vehicle (it can be a misbehaving or an administrative issue) and takes the decision to revoke its ATs. From this point, it will take all necessary actions to inform, in real time, the PKI servers and other vehicles of the system about this just revoked ATs.
- Test case 2: The Anti-Hacking device detects a misbehaving in the vehicle, it informs the MEC about this situation, and henceforth, the system acts as in test case 1.

Tests in this scenario will require triggering situations that can lead to certificate revocation in the MEC or in the Anti-Hacking device. We will use three options for this purpose. Firstly, CARMEL will implement a small function of software that will randomly select moments in which the vehicle is supposed to be under attack, secondly, when the GPS spoofing detection system detects an attack, and thirdly, when the HSM detects an attack.

All communication between OBUs and fixed infrastructure related to certificates and revoked certificates will be transmitted using the LTE-Uu channel. CARMEL will study the implications of CRL distribution on channel load and its scalability depending on the number of revoked certificates.

3.4 Data Collection and Selection Methodology

This section describes the datasets that will be used for two of the three scenarios explained in 3.2. Note that the tamper attack scenario does not rely on any dataset, so we skip it in this part.

3.4.1 Location Spoofing Attack

In the following, we describe a possible data selection technique to obtain the secondary location stream used in CARMEL to validate the integrity of the satellite-based location measurements. We assume to have available at the CAN bus the measurements from the steering angle sensor, the yaw rate gyroscope and the wheel speed sensor, respectively denoted with:

$$\alpha, \dot{\phi}, v$$

Then, it is possible to build a non-linear bicycle model of the vehicle system state following specific physical laws. A bicycle model is built under the basic assumption that the motion of a vehicle can be well approximated by a bicycle, i.e., collapsing the rear and the front axes into a single point. A representation of the model is given in Figure 32, together with the physical meaning of the assumed on-board sensors.

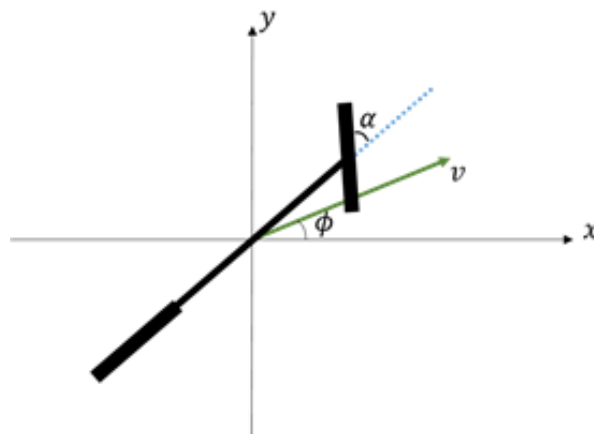


Figure 32: Vehicle Bicycle Model Representation exploiting specific on-board measurements, i.e., steering angle sensor, yaw rate gyroscope and wheel speed sensor

Given the adopted bicycle model, we describe the motion of the “vehicle” considering the involved inertial forces, e.g., the friction of the wheels on the pavement. We start describing the vehicle movement relative to its body-frame, i.e., having the x-axis directed as the heading of the vehicle:

$$\begin{pmatrix} x_{k+1}^u \\ \dot{x}_{k+1}^u \\ y_{k+1}^u \\ \dot{y}_{k+1}^u \end{pmatrix} = \begin{pmatrix} v \Delta t \\ v \\ \frac{1}{2} \left(C_f \left(\alpha - \frac{l_f \dot{\phi}}{v} \right) + C_r \frac{l_r \dot{\phi}}{v} \right) \frac{1}{M} \Delta t^2 \\ \left(C_f \left(\alpha - \frac{l_f \dot{\phi}}{v} \right) + C_r \frac{l_r \dot{\phi}}{v} \right) \frac{1}{M} \Delta t \end{pmatrix}$$

where l_f and l_r represent the distance of the front wheel and the rear wheel from the mass barycentre, respectively, M is the mass of the vehicle, C_f and C_r represent the corner stiffness of the front and rear wheels, respectively.

Given the one-step prediction of the vehicle movement in its body-frame, a simple coordinate transformation is applied to obtain a one-step prediction in the global geographic reference system:

$$\begin{pmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ y_{k+1} \\ \dot{y}_{k+1} \\ \phi_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + x_{k+1}^u \cos \phi_k - y_{k+1}^u \sin \phi_k \\ \dot{x}_{k+1}^u \cos \phi_k - \dot{y}_{k+1}^u \sin \phi_k \\ y_k + x_{k+1}^u \sin \phi_k + y_{k+1}^u \cos \phi_k \\ \dot{x}_{k+1}^u \sin \phi_k + \dot{y}_{k+1}^u \cos \phi_k \\ \phi_k + \dot{\phi} \Delta t \end{pmatrix}$$

The associated covariance of the estimated position is computed with a Bayesian Filter e.g., an Extended Kalman Filter (EKF) approach.

The EKF can also be used to fuse the obtained predicted vehicle location with some global positioning measurement. In the update step of the EKF, we assume to obtain a global positioning measurement of the vehicle through signals of opportunity (SOOP); however, any other type of global positioning measurement could be used instead.

For SOOP, we can follow the approach proposed in [63]. A passive receiver located at the vehicle scans a predetermined set of bandwidths where transmitters are normally active, e.g., LTE bandwidths, TV bandwidths or Radio bandwidths. Thanks to the average received power at the selected bandwidths, it is possible to exploit well-known path loss models and to compute the approximate distance between the passive receiver and the corresponding transmitters. Applying standard multilateration techniques, it is then possible to obtain, with some uncertainty, the location of the vehicle relative to the transmitters and, subsequently, the global location of the vehicle if the transmitters' locations are known. A possible hardware implementation of the above-mentioned solution is provided in Figure 33, where a HackRF One device passively scans multiple bandwidths. The HackRF One device is connected to a Raspberry Pi 3, where the needed software for passive scanning and position multilateration is running.



Figure 33: SOOP Implementation

3.4.2 Attack on the V2X Message Transmission

The second scenario of the connected mobility use case involves a task that requires data for the purpose of training a ML algorithm. This task is about deciding the best moment to change the AT of the V2X messages to avoid being tracked by an attacker who is listening to the sent messages.

Several datasets will be used for the training, test and validation of the ML algorithms used in this task. They are vehicular mobility datasets and many of them are openly distributed on the Internet. In principle, these open data would be enough to fulfil the requirements of this task. Although, if more specific data is needed, it would be produced by using simulators like CARLA or SUMO.

Some of the open datasets already explored are summarized in Table 18. It includes both, datasets produced both by real and by simulated data. The Crowdad dataset originates from GPS data of taxis and the NGSIM dataset was created by images taken by cameras on the streets. On the other hand, the last two datasets shown in the table were produced with simulated data originating from two different simulators (SUMO and CARLA).

Dataset	Source	Description	Size
Crowdad	Real data from GPS	Data registered: XY position of approximately 500 taxis collected over 30 days Location: San Francisco Bay area	91 MB
Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data	Real data from video cameras	Data registered: cars XY position and ID Location: <ul style="list-style-type: none"> • Southbound US 101 and Lankershim Boulevard in Los Angeles, CA • Eastbound I-80 in Emeryville, CA • Peachtree Street in Atlanta, Georgia 	1,5 GB
Vehicular mobility trace of the city of Cologne	Simulated data in SUMO	Data registered: XY position of more than 700.000 car trips during 24 hours Location: Cologne Urban area (400 square kilometres)	7 GB
Motion Distorted LiDAR Data	Simulated data in CARLA	Data registered: <ul style="list-style-type: none"> • Town 1: 2.9 km of drivable roads with 90 vehicles for 5 minutes • Town 2: 1.9 km of drivable roads with 60 vehicles during 5 minutes 	17 GB

Table 18: Data sources for ML algorithm training

The collected data will need to be curated and enriched in order to reproduce the same information that is sent by the V2X messages. For this purpose, data processing methods will be applied to include car ID, AT reference number, car position (x and y), and velocity in the training datasets.

3.5 Use of Artificial Intelligence and Machine Learning

In line with 3.4, this section skips the tamper attack because the CARMEL approach is based on HW techniques to detect and prevent it. The other two scenarios can consider the use of AI/ML techniques, as explained below.

3.5.1 Location Spoofing Attack

In this section, an approach for detecting the location spoofing attack that utilises AI/ML techniques to achieve cooperative vehicle localization is presented.

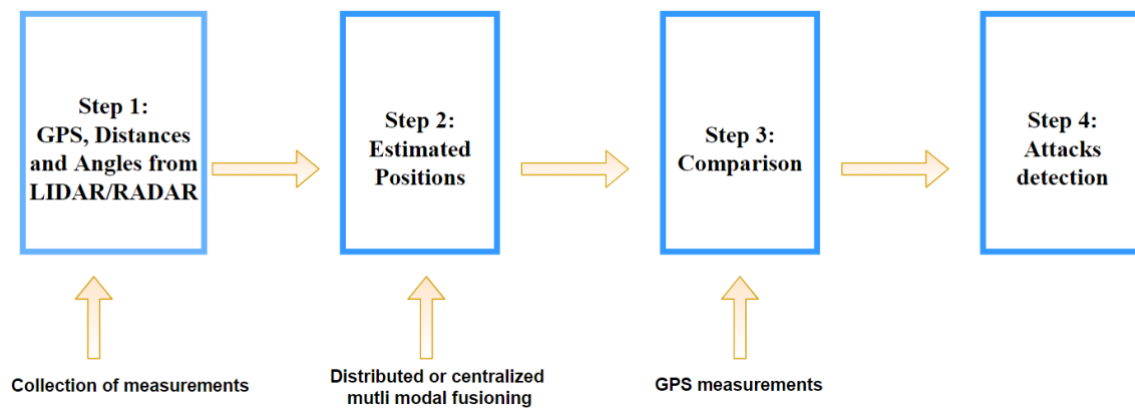


Figure 34: Location Spoofing Detection Architecture

Figure 34 represents the approach. In Step 1, each vehicle collects the useful measurements (absolute positions, relative distances, etc.). In Step 2, based on the cost function formulated by the measurements or the graph approach of the previous subsection, each vehicle estimates its position. In Step 3, the estimated positions will be compared with the GPS measurements, and if the difference is above a predefined threshold, then the attack has been detected.

We focus on the robustification of the cooperative localization approach, that can be considered as a maximum likelihood approach, assuming that the noise in the different modalities can be modelled as a normal gaussian noise. It can be considered as an unsupervised AI approach. The robustification strategies are based either: i) on estimating vehicles locations using also input from cameras and range measurement from geotagged images, ii) by imposing constraints on additional optimization variables that correspond to the GPS spoofing attacks.

Multi modal fusion (LIDAR/RADAR,GPS) between vehicles for accurate position estimation

Autonomous driving is considered to be the major framework for cooperative Intelligent Transportation Systems (cITS). cITS applications relying on 5-G based Vehicular Ad hoc NETWORKS (VANETs) assume the availability of a positioning system to provide each vehicle with accurate location information regardless of operating conditions. Although the Global Positioning System (GPS) is the most common, accessible and cheap device for vehicle localization today, it still fails to fulfil cITS application requirements (localization error lower than 1.5 m), especially in challenged environments such as long tunnels and dense urban canyons. Moreover, other vehicle motion sensors (e.g. gyroscopes, accelerometers and odometers) that can contribute to the localization process, suffer from error accumulation. Thus, relying on the fact the 5-G VANETs allow Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication capabilities, cooperative localization is considered nowadays a serious alternative to self-localization.

The main goal of cooperative localization is to exploit different sources of information coming from different vehicles within a short-range area, in order to enhance positioning system efficiency while keeping the computing cost at a reasonable level. In other words, vehicles share their location and environment information to others in order to increase their own global perception. It aims on collision avoidance/warning, cooperative adaptive cruise control, navigation, etc. It is expected to outperform self-localization, by taking advantage of sharing and fusion information coming from different sources such as sensors of multiple vehicles. The task of cooperative localization can be performed by a centralized architecture, where a single vehicle acts as a fusion centre that collects and processes the information the other vehicles sent, or by a distributed architecture where each vehicle acts as a fusion centre relying only on the information of its immediate neighbours.

Consider a 2-D region where N interconnected, via V2X, vehicles of a VANET, are moving and collecting measurements. An example of VANET, is shown in Figure 36.

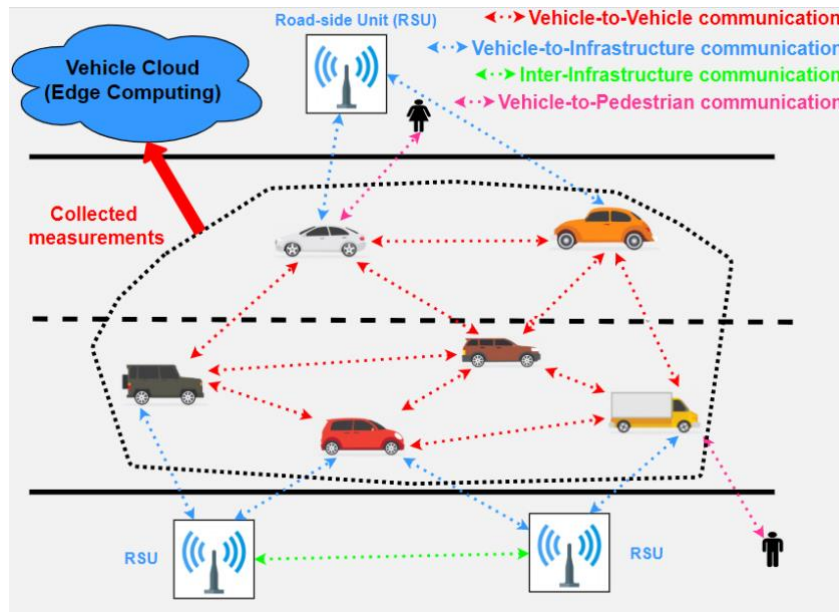


Figure 35: Example of VANET

The location of the i -th vehicle at k -th time instant is given by $\mathbf{x}_i(k) = [x_i(k) \ y_i(k)]^T$. Based on [64], each vehicle is able to know its absolute position from GPS and to measure its relative distance and angle of arrival to connected neighbouring vehicles using LIDAR or RADAR. The true relative distance z_{ij} between connected vehicles i and j is given by:

$$z_{ij}^{(k)} = \left\| \mathbf{x}_j^{(k)} - \mathbf{x}_i^{(k)} \right\|$$

The true angle of arrival $z_{a,ij}$ between neighbouring vehicles i and j is given by:

$$z_{a,ij}^{(k)} = \arctan \frac{y_j^{(k)} - y_i^{(k)}}{x_j^{(k)} - x_i^{(k)}}$$

Thus, the i -th vehicle exploits 3 measurement models:

1. Relative distance measurement: $\hat{z}_{d,ij}^{(k)} = z_{d,ij}^{(k)} + w_d^{(k)}, w_d^{(k)} \sim \mathcal{N}(0, \sigma_d^2)$
2. Angle of Arrival measurement: $\hat{z}_{a,ij}^{(k)} = z_{a,ij}^{(k)} + w_a^{(k)}, w_a^{(k)} \sim \mathcal{N}(0, \sigma_a^2)$
3. Absolute position measurement: $\hat{\mathbf{z}}_{p,i}^{(k)} = \mathbf{x}_i^{(k)} + w_p^{(k)}, w_p^{(k)} \sim \mathcal{N}(0, \Sigma_p)$

where covariance matrix Σ_p is a diagonal matrix equal to $\text{diag}(\sigma_x^2, \sigma_y^2)$

A common approach in cooperative localization is to formulate (based on the measurement models) an objective cost function $C(\mathbf{x})$ (according to Maximum Likelihood Estimation or MLE) and to minimize it with respect to locations x_i , in order to reduce the error of absolute position measurement. According to MLE, the relative or self-measurements depend only on the locations of nodes or location of the self-node involved.

Thus, the desired cost function $C(\mathbf{x})$ is given by:

$$C(\mathbf{x})^{(k)} = \sum_{i,j \in N(i)} \frac{(z_{d,ij}^{(k)} - z_{ij}^{(k)})^2}{2\sigma_d^2} + \sum_{i,j \in N(i)} \frac{(z_{a,ij}^{(k)} - z_{ij}^{(k)})^2}{2\sigma_a^2} + \sum_{i \in N} \frac{1}{2} \left[\frac{(z_{p,i}^{x,(k)} - x_i^{(k)})^2}{\sigma_x^2} + \frac{(z_{p,i}^{y,(k)} - y_i^{(k)})^2}{\sigma_y^2} \right]$$

$N(i)$ indicates the set of connected neighbours of the i -th vehicle.

For the minimization of $C(\mathbf{x})$, algorithms such as distributed ADMM [64] or cooperative gradient descent can be used. Other solution approaches, apart the previously described, include Extended Kalman Filtering [65], Bayesian Methods [66], Split covariance intersection filtering [67], etc.

Besides the minimization of a cost function, one can rely on the topology of the graph that the moving vehicles create. The nodes $v_i = [x_i, y_i]^T$ of this graph represent the true position of vehicles and the edges between v_i and v_j indicate that vehicles i and j are neighbours. An example of such graph of VANET is presented in Figure 36.

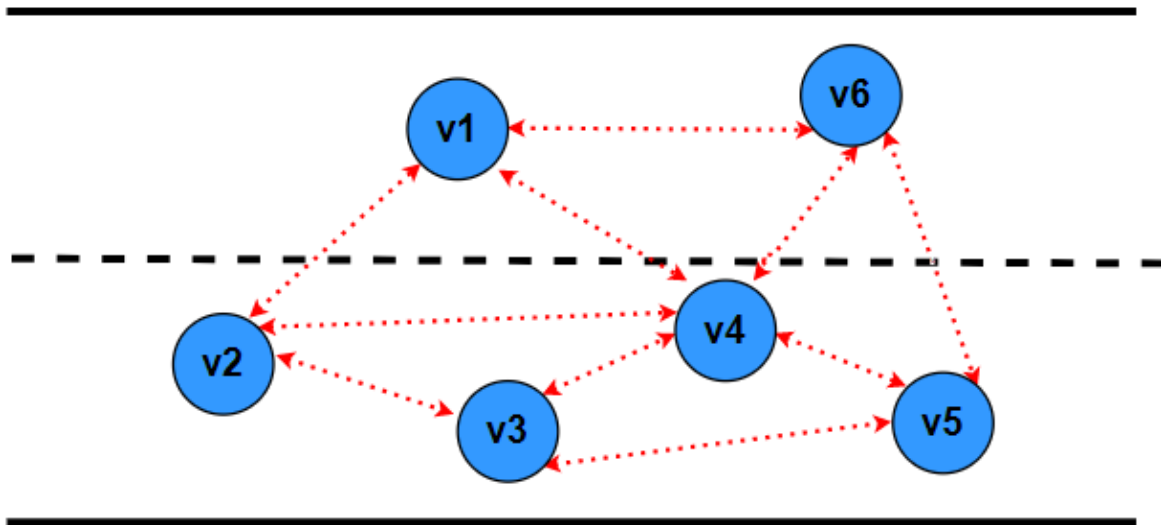


Figure 36: Random VANET topology

Given the graph, we can form the adjacency matrix A (where $A_{ij} = 1$ if i and j are neighbours, 0 otherwise) and the degree matrix D (where $D_{ii} =$ the degree of node v_i). Given these two matrices, then the Laplacian matrix $L = D - A$. Based on [68] and the Graph Signal Processing theory, we can define the differential coordinates $\delta_i = [\delta_{i(x)} \ \delta_{i(y)}]$, where:

1. $\delta_{i(x)} = (1/N_i) \sum (v_{i(x)} - v_{j(x)})$ (1)
2. $\delta_{i(y)} = (1/N_i) \sum (v_{i(y)} - v_{j(y)})$ (2)

The neighbours of v_i are v_j and N_i is the total number of neighbours of v_i . As we mentioned earlier, each vehicle is capable to measure the relative distance r_{ij} (distance of i and j) and azimuth angle/angle of arrival a_{ij} . Thus, relations (1) and (2) transform to:

1. $\delta_{i(x)} = (1/N_i) \sum (-r_{ij} \cdot \sin a_{ij})$ (3)

$$2. \delta_{i(y)} = (1/N_i) \sum (-r_{ij} \cdot \cos \alpha_{ij}) \quad (4)$$

In order to recover the true and unknown absolute coordinates $[x \ y]$ of nodes v_i , we have to solve 2 linear systems:

$$1. Lx = \delta_{(x)} \quad (5)$$

$$2. Ly = \delta_{(y)} \quad (6)$$

Consequently, given the vehicles graph, we can compute the differential coordinates relying on distances and azimuth angles, coming from LIDAR/RADAR. Unfortunately, we cannot solve the linear systems (5) and (6) because L is singular. Thus, we have to add some known anchor points, such as the positions from GPS, in order to solve the systems and restore the desired absolute coordinates. The previously described method is known as Laplacian Processing, and based on the measurement models of GPS and LIDAR/RADAR the task of cooperative localization can be fulfilled.

Robust and accurate vehicle localization plays a key role in building safety applications based on Vehicle-to-Vehicle (V2V) networks. While GPS is widely used for navigation systems, its localization accuracy poses a critical challenge for the proper operation of V2V safety networks. As a result, a hybrid solution that leverages visual and cooperative techniques for accurate position estimation should be provided.

This solution should address the challenges that exist in the literature. First of all, GPS data is often noisy due to potential attacks and exhibits significant localization errors in many urban areas. Moreover, accurate localization from imagery often relies on structure-based techniques, and thus are limited in scale and are expensive to compute. There are also appearance variations caused by changes in seasonal and illumination conditions and that makes the retrieval approaches, which try to find the most relevant database images for the query image, less efficient.

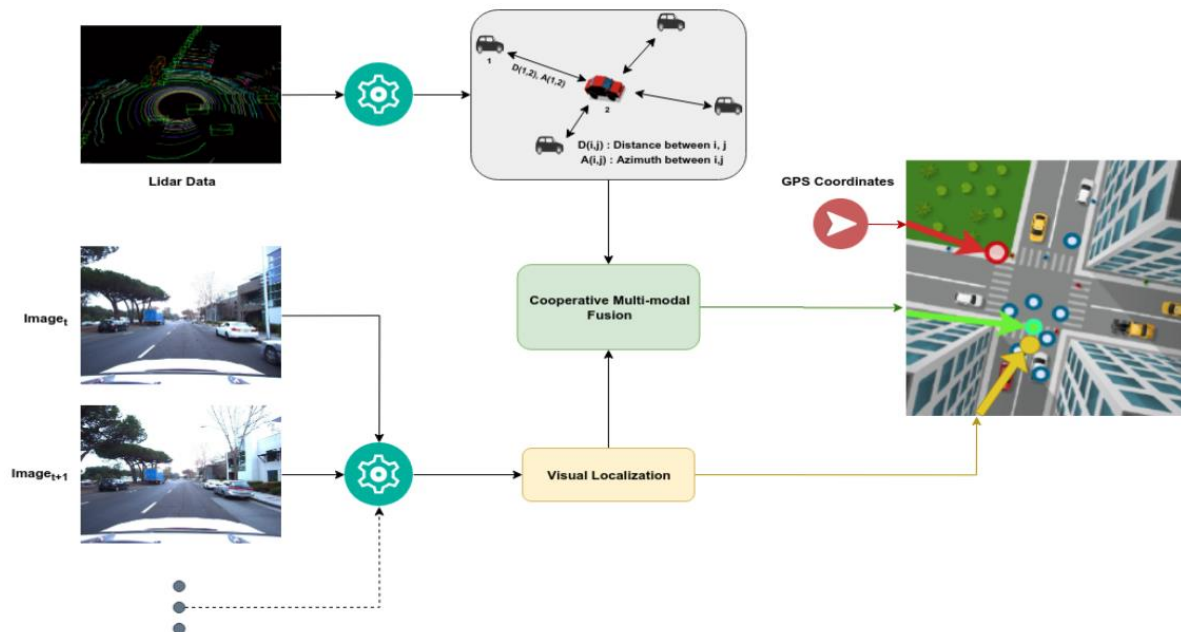


Figure 37: High-level architecture of fusion of GPS and LIDAR/RADAR data

The visual part of the proposed solution is based on [69]. In general, given a video stream of images, a hybrid visual search and ego-motion approach is applied to leverage both image representation and temporal information. As a first step, a database in which every record will include a descriptor, e.g. [70][71][72] of each image and the GPS coordinates is created. Then, given a set of images, a visual search is applied against a set of relevant geo-tagged images from the database. As a result, a ranked list of images is obtained sorted by descriptors distances. Finally, a weighted average of the GPS coordinates of the extracted descriptors yield a corrected GPS value for the query image, providing a coarse localization fix. As a final step, the visual ego-motion is used to estimate the vehicle's motion between consecutive video images. Fusing vehicle dynamics with the coarse location fixes, further

regularizes the localization error and yields a high accuracy location value. Finally, integrating both visual and the system described in the previous scenario (fusion of GPS and LIDAR/RADAR data), could lead to more accurate position estimation. In Figure 37, the architecture of the aforementioned procedure is presented.

Integration of outliers into to the cooperative position estimation

Consider the scenario described in the previous section: N interconnected vehicles are moving and collecting measurements such as GPS positions, relative distances and angles to neighbouring vehicles. If the GPS is spoofed, then the modified absolute position measurement model transforms to:

$$\bar{z}_{p,i}^{(k)} = \tilde{z}_{p,i}^{(k)} + O^{(k)}$$

where $O_{(k)}$ is sparse outlier matrix modelling the attacks on GPS at k-th time instant

In order to reduce the effects of GPS spoofing and recover the true positions of vehicles, we need to minimize the following cost function with respect to locations \mathbf{x}_i and the outlier matrix O :

$$\begin{aligned} \underset{X^{(t)}, O_p}{\operatorname{argmin}} C_{(X)}^{(t)} = & \sum_{i=1}^N \sum_{j=1}^{N(i)} (z_d[i, j]^{(t)} - \|X_i^{(t)} - X_j^{(t)}\|_2)^2 \\ & + \sum_{i=1}^N \sum_{j=1}^{N(i)} (z_a[i, j]^{(t)} - \arctan \frac{y_j^{(t)} - y_i^{(t)}}{x_j^{(t)} - x_i^{(t)}})^2 \\ & + \sum_{i=1}^N \left\| (\bar{z}_p[i]^{(t)} - o_p[i]^{(t)}) - X_i^{(t)} \right\|_2^2 + \lambda \|O_p\|_1 \end{aligned}$$

Having the estimated positions, we can compare them with GPS measurements and if the difference is above a predefined threshold, then an attack has been detected.

3.5.2 Attack on the V2X Message Transmission

This scenario considers four different types of attack already mentioned before:

- Generation of fake messages
- Sniff and replay messages of compliant vehicles
- Messages sent by a compliant vehicle with a fraudulent identity
- Tracking of vehicles by sniffing sent messages

The first three considered attacks are countered by using the pseudo anonymous authentication tickets (AT) without any need of applying AI/ML techniques. The fourth attack will be mitigated by changing the AT of the V2X messages in order to avoid the possibility of tracking the car that is sending the V2X messages. The best moment to make the change of the AT together with the best moment to send the V2X messages will be calculated by Machine Learning algorithms. Two adversarial algorithms will be trained for this purpose:

- The first algorithm will try to track the position of a car by analysing their sent and signed V2X messages.

- The second algorithm will learn when it is the best moment to change the pseudonymous certificate and to send the V2X message to fool the first algorithm.

On top of that, some restrictions will be imposed on the algorithms when changing the certificates, i.e.: limited number of changes per day, maximum time without sending any certificated V2X message, no changes allowed during critical situations, etc. The scheme of the data flow and the algorithms is represented in Figure 38.

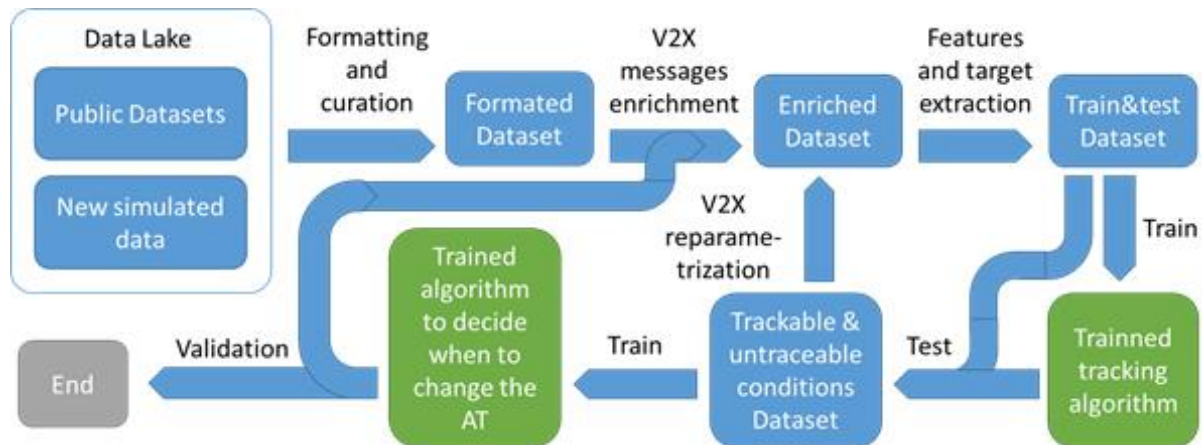


Figure 38: Data flow for the attack on V2X message transmission scenario

The data to train the ML models will be taken from several sources defined as the data lake. It will contain data from public datasets made with both real and simulated data. Additionally, if some specific data is required, new data will be simulated in traffic simulators such as CARLA or SUMO. From this data lake, the data will be integrated, reformatted and curated in order to create a consistent and exploitable dataset. This dataset will contain different dynamic scenarios with cars identified by their ID and their position (and some other data available in the data lake) for each time step.

The formatted dataset will be enriched to integrate the V2X messages mechanisms. It mainly consists in adding an AT to each car and deciding how often the V2X message with the car position (velocity etc) will be send and when the AT certificate of the V2X messages will be changed. These last two parameters are key factors to be decided since the untraceability of the car will depend on them.

A feature extraction is required to reduce the number of inputs in the ML tracking algorithm; otherwise, it will not be possible to train an algorithm that might content thousands of dimensions. That will be done taking into account that the actual position, the velocity and the time step limit the future position of a car. For example: a car with a maximum speed of 200 km/h cannot be 1 km away from the initial position after one second.

The trained tracking algorithm will be tested to identify under which conditions a car can be trackable or not by listening to the V2X sent messages. This test will be repeated for different parameters of the V2X message protocol such as the frequency at which the V2X messages are sent. These processes will generate a new dataset containing the conditions at which a vehicle is trackable and at which it is not. With this dataset, the second ML algorithm that decides when to change the AT will be trained.

3.6 Validation Methodology

This section will explain the methods used to validate each of the scenarios, summarized in the following table:

Scenario	Simulation	Demonstration
Location Spoofing Attack	YES	YES
Attack on the V2X message transmission	NO	YES
Tamper attack of vehicle's OBU	NO	YES

Table 19: Validation methods of scenarios in pillar 2

3.6.1 Location Spoofing Attack

The application for the secondary location stream and the satellite-based location integrity check will be implemented in a container that will be also tested in CARLA. In CARLA, the simulated GPS location measurements will be tampered by adding a drift to simulate a GPS location spoofing attack. In case of the SOOP solution, a new stream of data will be added to the simulator as input at the application container. Such stream of data will be computed off-line (in order to be coherent with the simulated scenario) and it will consist of at least four passively scanned bandwidths, of which the transmitter global position is known. When a GPS location spoofing attack is identified in this simulated environment, an alarm is triggered by the application container. Ideally, a visual alarm will also be presented on screen.

In a second phase, the reliability of the application for the GPS integrity check will be also shown in a dedicated demonstrator. The application container will be hosted by the anti-hacking device and the necessary measures for the secondary location stream for the vehicle are going to be retrieved from the CAN bus of a vehicle and from the necessary SDR hardware in real-time.

Scenario Name	Location Spoofing Attack
Related Use Case	Connected Mobility
Brief Description	Using SDR hardware, the attacker is able to spoof GPS satellite signals. The vehicle relies on a second location stream to identify a possible GPS location spoofing attack, based on vehicle's movement description and IMU measurements.
Challenges	1. Ability of counterfeiting legitimate GPS locations with small abnormal drifts 2. Real-time fusion of IMU measurements with GPS-free secondary location stream
Assumptions & Pre-Conditions	1. In case of SOOP implementation for the secondary location stream, it is assumed that the location of the base stations/RSUs access points passively scanned is known. 2. In case of SOOP implementation, a knowledge of the radio propagation conditions in the attacked area is assumed.
Goal (Successful End Condition)	In case of attack identification, the vehicle informs the PKI infrastructure about the attack in order to deploy the necessary countermeasures, e.g., the revocation of the attacked vehicle's certificate.
Involved Actors	Malicious attacker Cooperative car

	Fixed infrastructure Outside Infrastructure
Scenario Initiation	The cyber-attacker transmits fake GPS signals.
Novelty	In this scenario, a first implementation of GPS integrity check is presented. Furthermore, its importance lies on the ability of combining the self-understanding attack capability of the vehicle with a PKI infrastructure.
Main Flow	<ol style="list-style-type: none"> 1. The cyber-attacker alters legitimate GPS signals. 2. The vehicle exploits a secondary location stream to validate the GPS measurements. 3. Deviations between the two streams of vehicle locations are noted by the application container in the anti-hacking device. 4. The anti-hacking device recognizes the attack and informs the passengers and the PKI server. 5. The PKI server revokes real-time the certificate of the vehicle.
Evaluation Criteria	<p>The anti-hacking device detects the attack and informs the vehicle's passengers and the corresponding PKI server. (Binary evaluation)</p> <p>The anti-hacking device detects the attack within a few seconds.</p>

Table 20: Overview of Location Spoofing Attack scenario

3.6.2 Attack on the V2X Message Transmission

A major concern in V2X is how the overhead imposed by the security mechanisms affects the overall delay budget of the V2X applications. In this regard, it will be necessary to measure the transmission delay, which will be implemented using the GPS clocks of the transmitting and receiving OBUs. In fact, all V2X messages contain generation timestamp, which can be compared with the reception time at the receiver OBU.

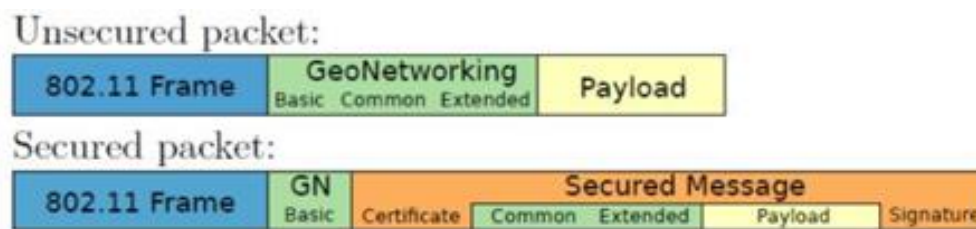


Figure 39: Structure of secured and unsecured frames

The second performance indicator is the increase of channel occupation due to the longer transmission time of secured V2X frames that contain the signature and, possibly, the AT certificate (Figure 39). CARMEL will use communication protocols analyser/sniffer to capture transmitted packets and estimate the channel load of different scenarios and situations.

The third performance indicator is the number of messages per second that an OBU can receive and process. Usually, an OBU needs to receive and process many more messages than it generates and transmits because it has to receive all messages transmitted from vehicles in the neighbourhood plus those retransmitted by the infrastructure. In order to measure the message reception capacity of the reception hardware and software of an OBU, it will require the transmission of a high number of V2X messages. CARMEL will develop software that will run on one or two additional devices, which will be able to emulate the transmission of many vehicles simultaneously.

Finally, it will be necessary to develop the application software that generates fake messages in the OBUs. This software, instead of using standard security functions from the HSM, will sign messages at software level, using Vanetza's functions.

Scenario Name	Attack on the V2X message transmission
Related Use Case	Connected Mobility
Brief Description	There are two different kind of attacks: a) A malicious attacker transmits fake CAM and DENM messages and b) a malicious attacker tries to track a specific vehicle.
Challenges	<ol style="list-style-type: none"> 1. Deploy a PKI system to register vehicles, distribute security credentials, authorize vehicles to transmit signed messages, revoke certificates and distribute lists of revoked certificates. 2. Ability to detect fake messages: not signed, signed with a non valid certificate, signed with revoked certificates, replayed and non authorized. 3. Ability to detect when a vehicle can be tracked due to the fact of using the same AT to sign messages and then select the optimal moment when the vehicle will change the current AT.
Assumptions & Pre-Conditions	<ol style="list-style-type: none"> 1. The scenario is provided with a communications infrastructure: OBUs, RSUs, Small Cells and ITS G5 communications protocol suite. 2. Vehicles are equipped with an HSM to store cryptographic material and a location system.
Goal (Successful End Condition)	<ol style="list-style-type: none"> 1. Vehicles drop fake messages and prevent safety applications from being misinformed. 2. Vehicles prevent from being tracked.
Involved Actors	Malicious attacker Cooperative car Fixed infrastructure PKI Infrastructure MEC Anti-hacking device
Scenario Initiation	<ol style="list-style-type: none"> a) The cyber-attacker transmits different type of fake messages. b) The anti-tracking algorithm
Novelty	This scenario will implement a complete security infrastructure for a vehicular communications system: PKI servers with capacity to distribute ATs and revoked certificate lists, message signature ability in the OBUs and a machine learning based algorithm to choose when a vehicle has to change the AT to avoid being tracked.
Main Flow	Case a) Fake messages: <ol style="list-style-type: none"> 1a. Normal vehicles register to the PKI system and transmit and receive messages. 2a. The attacker sends fake messages. 3a. Normal vehicles check the signature of these messages, detect that are not compliant and drop them. Case b) tracking prevention: <ol style="list-style-type: none"> 1b. Normal vehicles have a set of AT to sign messages. 2b. The antihacking device starts the algorithm to choose which is the optimum moment to change the AT. 3b. A tracking device tries to discover if messages signed with a renewed AT correspond to messages signed with a previous AT.

Evaluation Criteria	Case a) fake messages: Normal vehicles detect all non compliant messages. Case b) tracking prevention: A tracking device is not able to relate old AT with renewed AT in a high percentage (80%).
---------------------	--

Table 21: Overview of the Attack on the V2X Message Transmission Scenario

3.6.3 Tamper Attack of Vehicle's OBU

The tampering attack scenario will be validated via a real-life demonstration involving the components listed below:

Actors / stakeholders

- Car OBU
- Malicious attacker
- Vehicle owner
- OEM brand

Car OBU HW interfaces

- Ethernet Broad-R
- USB interface
- TTL interface
- Wireless Interface 802.11p
- CAN interface

The sequence of actions to be performed for the implementation of such attack is illustrated in Figure 40.

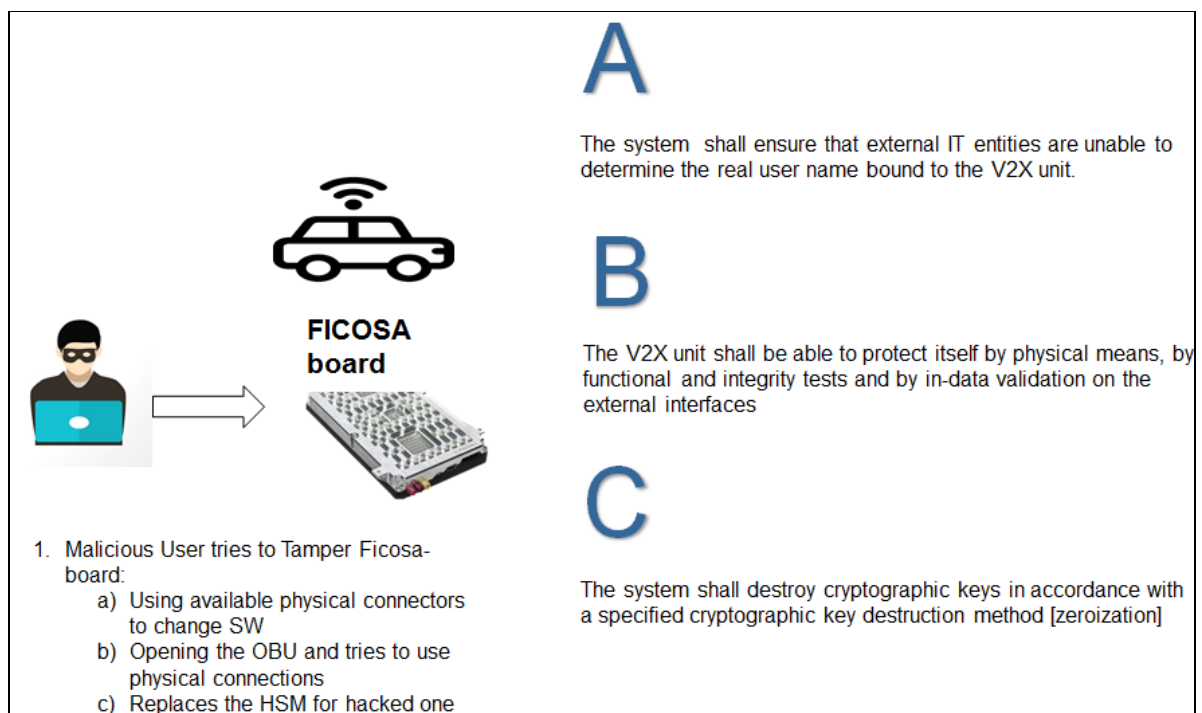


Figure 40: Tamper attack of vehicle's OBU sequence flow

Scenario Name	Tamper Attack of Vehicle's OBU
Related Use Case	Connected Mobility

Brief Description	The attacker is able to get physical access to an OBU by accessing the car. The attacker could also have acquired another OBU (e.g. aftermarket sample) in order to study potential vulnerabilities beforehand.
Challenges	1. Ability to detect the attack based on the recognition of anomalies in the physical parameters of the OBU. 2. Timely detection of the attack.
Assumptions & Pre-Conditions	1. We have collected data based on the “normal” behaviour of the vehicle. 2. There is successful monitoring of the vehicle.
Goal (Successful End Condition)	The attack has been recognized by the HDM and has informed both the rightful owner of the vehicle and the PKI server.
Involved Actors	Malicious attacker Cooperative car Fixed infrastructure
Scenario Initiation	The cyber-attacker physically accesses the vehicle and its OBU.
Novelty	This is a scenario combining physical attack on a smart vehicle with anomaly detection algorithms. Its importance lies on its capability to combine V2X communication security with real-world attacks.
Main Flow	1. The cyber-attacker physically accesses the smart vehicle. 2. The vehicle is under the control of the attacker. 3. The attacker performs the physical attack. 4. The HSM compares the features extracted from the specific attack to the normal pattern that is programmed from the factory. 5. The HSM recognizes the attack and informs the owner and the PKI server.
Evaluation Criteria	The HSM detects the attack and informs the rightful owner. (Binary evaluation) The HSM detects the attack under 1 minute.

Table 22: Overview of Tamper attack of vehicle's OBU Scenario

3.7 Use of the Anti-Hacking Device

In the connected mobility use case, the anti-hacking device will support only the vehicle location spoofing scenario; for the other two scenarios different components of the CAMEL solution will be used to detect the attacks.

The anti-hacking device hosts the container that implements the GPS integrity check algorithm. The required inputs are the on-boards sensor readings from the CAN bus and the required data to compute the GPS-free global localization measurements. In case of SOOP implementation, there should be a stream of passively received signals from the spectrum scanned by the HackRF One device. In the CAMEL scenario, the bandwidth scanned by the HackRF One covers the transmission bands of the base stations and the RSUs that will serve the demonstration area.

Then, the HackRF One device requires to be on board of the vehicle, either indirectly, e.g., via the CAN bus or any wireless technology, or directly connected to the Anti-Hacking device. Furthermore, the container that checks GPS integrity requires to input also the GPS localization measurement that will be used to detect a possible attack. When a GPS spoofing attack is identified, an alarm is triggered by

the container within the Anti-Hacking Device to the on-board passengers and the PKI infrastructure via the OBU transmission interfaces.

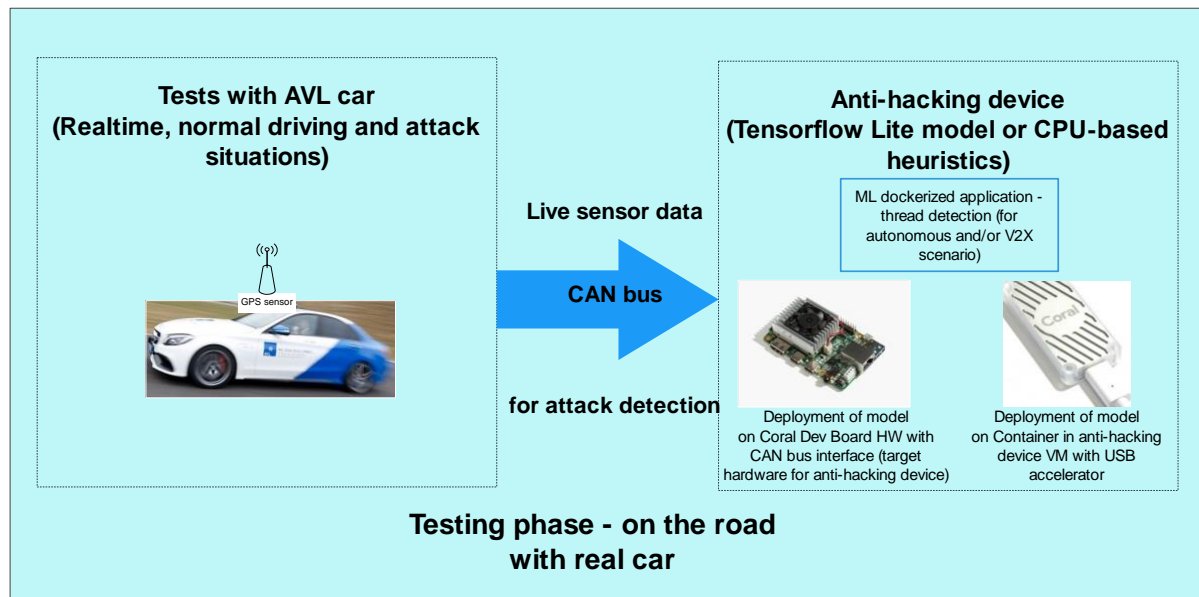


Figure 41: Use of anti-hacking device for GPS spoofing attack detection

Figure 41 shows how data from the GPS sensor in the AVL test car is fed into the anti-hacking device where it is analysed by the attack detection algorithm running in a container. Since the detection of the GPS attack will not be based on ML algorithms but will use other methods as described in previous sections, there will be no ML model creation phase. The ML capabilities of the anti-hacking device hardware are probably not used for this scenario.

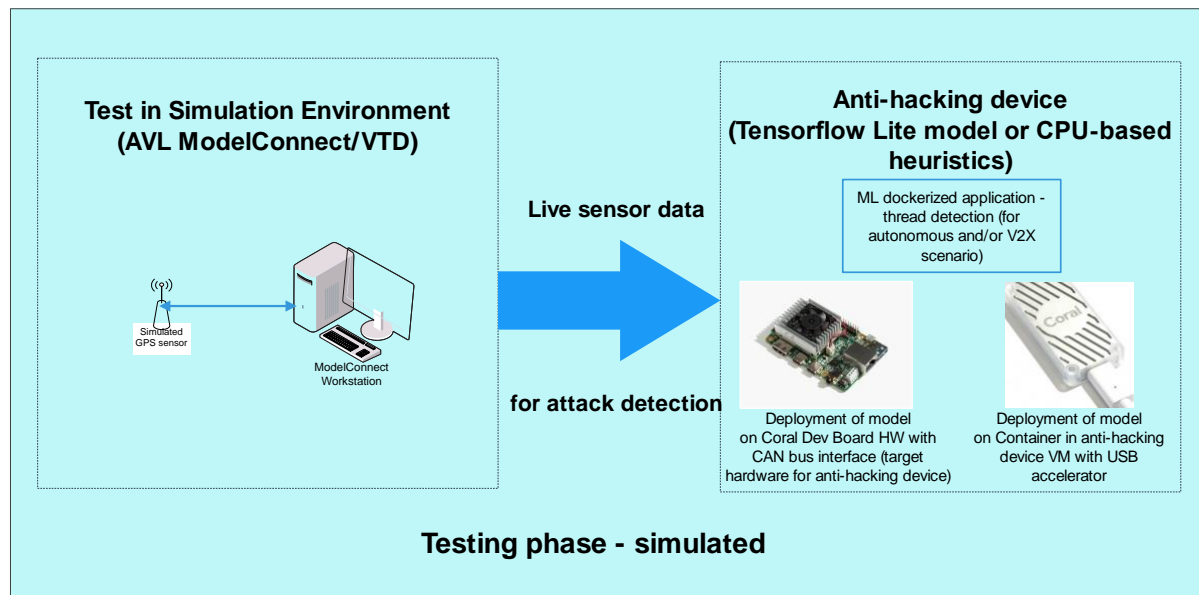


Figure 42: Use of anti-hacking in simulation environment

To support development of the GPS spoofing detection algorithm the anti-hacking device (either the Coral Dev Board or the simulated environment) can also be directly attached to the simulation

environment which will be based on AVL ModelConnect/VTD simulation software in this case (see Figure 42). In this case, higher level APIs (eg. REST APIs) will be used to facilitate the transfer of raw GPS sensor data.

In addition, AVL List GmbH – Austria has made a primary goal to contribute in the topic of GPS Spoofing. Since, AVL List GmbH has already worked in the topics like secure gateway for communications and many cyber related topics communication by using GPS signals and making it as secure as possible.

As a brief idea about making an Anti-hacking device for GPS Spoofing, AVL wants to go with the innovative approaches like Artificial Intelligence and Machine Learning. Bringing an AI enabled Anti – Hacking device for GPS Spoofing would be a promising step in making the Automated Mobility with advanced communication technologies safe and secure. Figure 43 shows the representation of the Anti-Hacking device for GPS Spoofing which is denoted as Intrusion Detection along with the In- house simulation setup. The intrusion detection will be a container inside the Anti-hacking device developed by T-Systems. The In-House simulation setup is named Model.CONNECT (see Annex I). It consists of components like VTD interface, Vehicle Dynamics, HMI Interface, GNSS sensor Simulation and Manipulation and Intrusion Detector or Anti – Hacking Device for GPS Spoofing. Monitoring and visualization of all the parameters and other supporting variables can be done in the desktop.

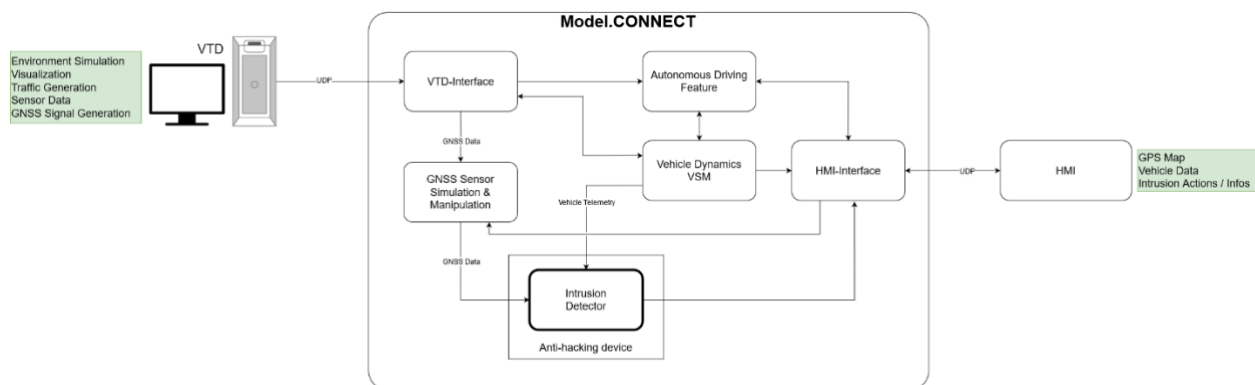


Figure 43: Model.Connect Simulation setup for the Intrusion Detector

In Figure 44 a further step in implementing the Anti-hacking device as a realistic approach an idea is proposed to get the data from the vehicle using CAN communication interface. From a GNSS Stimulator the data is passed to a GNSS receiver and from the receiver it is sent to the detector via CAN interface.

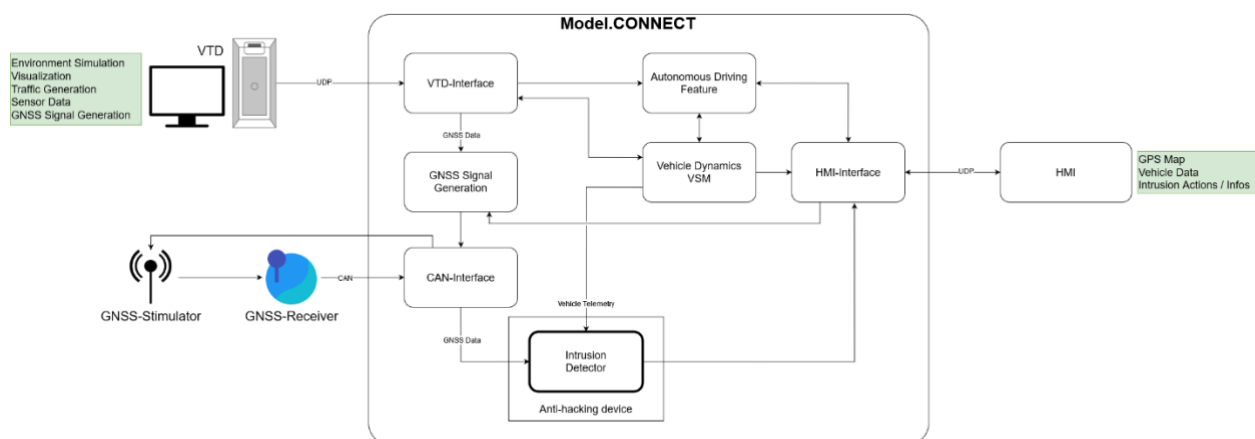


Figure 44: Step 2 implementation of the Model.Connect with intrusion detection

3.8 Functional Requirements

Reg ID	CRPL2-FR01
Title	Secure association
Definition - Description	The system shall be able to establish a communication channel between itself and another ITS station such that they can exchange messages according to negotiated security parameters.
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will implement 802.11p and C-V2X communication technologies.

Reg ID	CRPL2-FR02
Title	Identity Management
Definition - Description	The system shall support simultaneous change of communication identifiers (like station ID, network ID, MAC address) and credentials used for secure communications, within the ITS station
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will support a PKI key system, modified with pseudonyms, in order to address such issues.

Reg ID	CRPL2-FR03
Title	Replay protection
Definition - Description	The system shall verify that messages are sent/received in a consistent manner by including a timestamp in outgoing messages and by checking the timestamp of incoming messages
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will include GPS clock in the messages for time control, and will discard messages timed-out.

Reg ID	CRPL2-FR04
Title	Plausibility validation
Definition - Description	Time stamps and geo-positions shall be checked for plausibility, both on incoming and outgoing messages.
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will include algorithms capable of validating if geo-positions are plausible using a variety of methods.

Reg ID	CRPL2-FR05
Title	V2X unit self-protection
Definition - Description	The system shall be able to protect itself from spoofing and manipulation including physical and software tampering.
Target WP	3
Priority	Mandatory
How addressed	The CAMEL project will provide a hardware/software system that is able to detect if an OBU is physically attacked. If someone wants to take control of an OBU, they will have to avoid obstacles

	that CAMEL will put in their way, enabling the system to detect and protect main secrets.
--	---

Reg ID	CRPL2-FR06
Title	HSM communication
Definition - Description	The system shall be able to protect the V2X HSM interface from spoofing and manipulation either by physical or logical methods.
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will work with signed bootloaders and Linux kernels to implement resilience against hardware and software modifications.

Reg ID	CRPL2-FR07
Title	Exclusive HSM
Definition - Description	All ECC key generation, ECDSA signature generation and ECIES encryption/decryption operations shall be performed by the V2X HSM.
Target WP	3
Priority	Mandatory
How addressed	The CAMEL platform will be equipped with an HSM where private key data will be stored. It will also include a cryptographic processor for sensitive operations.

Reg ID	CRPL2-FR08
Title	Messages anomaly detection
Definition - Description	The CAMEL platform must be able to detect anomalies or forging of messages.
Target WP	4
Priority	Mandatory
How addressed	The CAMEL project will provide the PKI infrastructure that is able to detect irregularities in messages sent between vehicles.

4 Pillar 3 - Electromobility

4.1 Context

Plug-in Electric Vehicles (EVs) have demonstrated significant potential over the last years, in part due to the recent technological developments in the field of electrical engineering but also due to their potential to reduce greenhouse gas emissions and mitigate oil dependency, which is in tune with the growing environmental awareness of society. The current number of electric vehicles in EU is estimated to be over 1 million while in the global scale is over 5 million, but most importantly the trend towards EV is constantly increasing. In Figure 45, the deployment of electric cars is depicted for selected countries, showing an increasing trend over the last years with estimations predicting sales over 44 million vehicles per year by 2030 [97].

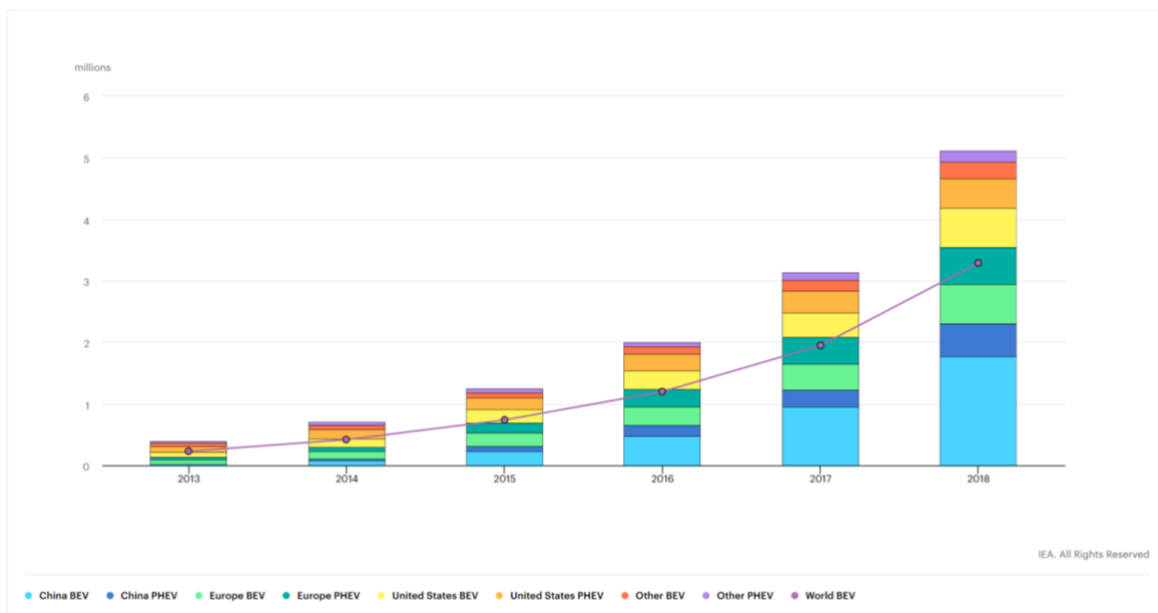


Figure 45 The increasing trend of EV deployment in different countries [98]

Plug-in EVs are considered a promising solution when it comes to reducing CO₂ emissions in the transportation sector, but they are also accompanied by a series of technical challenges such as the additional charging infrastructure that must be installed in order to accommodate them. From the Distribution System Operator's (DSO) perspective, the installation of new charging station can become a severe problem as the current power infrastructure does not support simultaneous charging of large numbers of electric vehicles.

One way to overcome this is through smart charging. Smart charging refers to a system where an electric vehicle and a charging device share a data connection, and the charging device shares a data connection with a charging operator, thus it creates a smart grid among the EVs, the charging stations and the DSO. As opposed to traditional charging, smart charging allows the charging station's owner to monitor, manage, and restrict the use of their devices remotely to optimize energy consumption.

This capability, however, puts indirectly into risk the reliability and security of the power network, as neither the charging stations have deployed security mechanisms for identifying and preventing security threats and attacks, nor the DSO have implemented security mechanisms for mitigating potential disturbances of the network due to a break-down (or a hack) of the smart charging stations.

Smart charging is complex system which requires the orchestration of a number of services such as metering and payment for energy, communication between the EV battery management system and the charge point, followed by a communication mechanism between the CP and a central management system, and finally the establishment of a communication channel between the CP and energy suppliers

(DSO, TSO, smart grids, etc.). Having in mind that the services are offered from different entities, these complex communication schemes create an environment susceptible to a number of security threats on different levels.

The co-existence of an electrical system monitored and controlled from an ICT infrastructure is an open challenge due to the heterogeneity of the involved cyber-physical systems which require the standardization of protocols and the implementation of two primary interfaces, one for electricity and another for the management of the system. In the case of the smart charging scenario, the ICT system is related to the status, authorization, metering, and billing of the EV that interacts with the system.

A high-level overview of the entities involved in the smart-charging use case are depicted in Figure 46. The DSO is responsible for the distribution of the electric power and ensures the functionality of the electricity network, the CPO takes care of the customer-end services (authentication, billing, etc.) alongside with the management of the charging points, the eMSP is responsible for setting the billing mechanism, the CP acts as the open gate to the system, and eventually there is the EV which is the end-user of the infrastructure. The roles/entities of the smart-charge use case that are described briefly above are presented in more detail in Table 23.

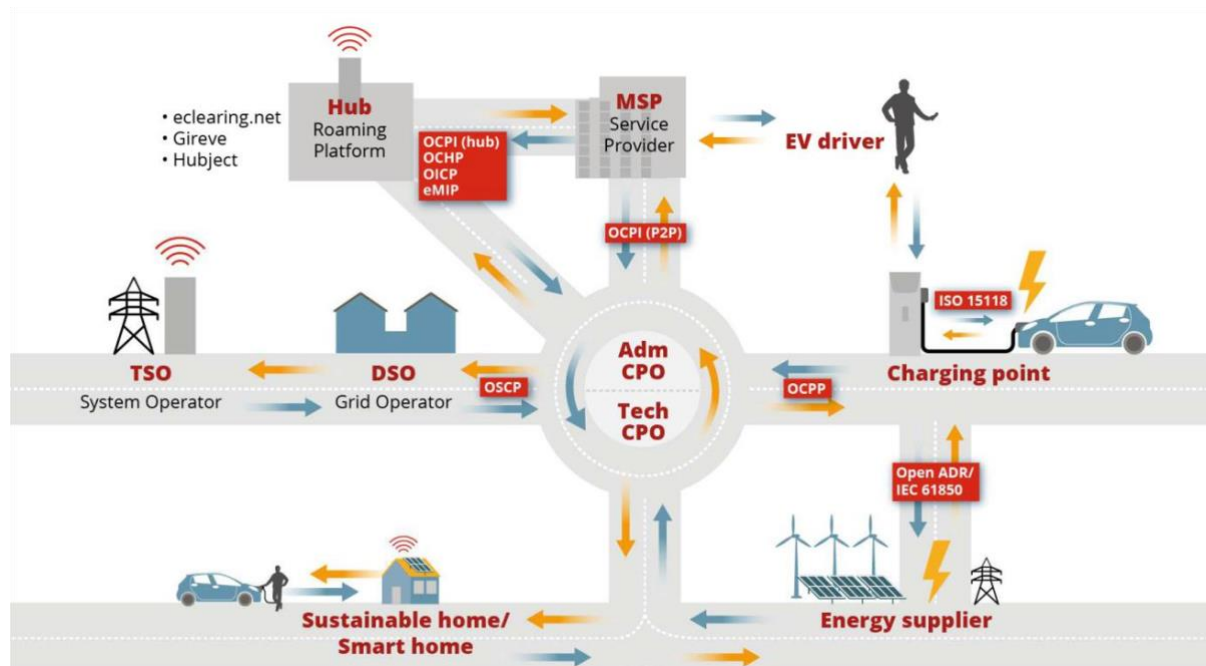


Figure 46: A high level depiction of the entities getting engaged in the smart charging scenario [96]

Entity	Description
Distribution System Operator – DSO	The distribution system operator (DSO) manages the electrical grid. The DSO does not produce any electric power but does however ensure that it is transported from the power station to the place where it is needed. The most important task of the DSO is to maintain a stable, reliable and well-functioning electricity network.
eMSP	An eMSP is a market role that offers charging services to EV drivers. An eMSP provides value by enabling access to a variety of charge points around a geographic area, usually in the form of a charge card. This means the EMSP is responsible to set up contracts with customers (owners of EV cars) and for managing customer information and billing.
Charging Point Operator - CPO	The CPO is responsible for the management, maintenance and operation of the charging stations (both technical and administrative). The role of CPO can be segmented into: 1. Responsibility for administrative operation (e.g. access, roaming, billing to eMSP etc.) and 2. Responsibility for technical maintenance, which is often done by the manufacturer. CPOs play a very important role in the EV market as they are responsible for bridging the gap between the entities

	managing and maintaining the physical electrical network – the DSOs – and all other entities: the energy providers, the customers and the eMSPs.
Charging Point - CP	Charge Points are devices where EVs get charged. Each CP contains at least one meter per socket (MID meter) owned and controlled by the CPO. This CPO meter is connected to the energy socket through which the EV gets charged and is used to measure the energy consumed by the EV. Each CP also includes a local controller (LC) with a connection (e.g.: GPRS or wire connection) to the back-office of the CPO. Among other things (e.g.: remote updates), such connection is used to authenticate the customer (EV owner) at the CPO.
Electric Vehicle - EV	Gets charged through a CP. In many cases a vehicle will charge to its maximum capacity but the vehicle can always determine its own charging profile within the range available

Table 23: Description of the entities engaging the smart charging scenario

GreenFlux provides a white-label cloud-based SaaS Service and Operations Platform which allows both CPOs and eMSP to run their EV charging business. The cloud-platform consists of three main functionalities:

- With the operator functionality, customers can directly couple charging stations, enabling them to do operational management, fault analysis, retrieve data from the charging station and provide smart charging services.
- With the service provider functionality, customers can provide EV-drivers with RFID cards and App-plugins. Data can be exchanged with other partners enabling roaming services.
- With the billing functionality, many pricing schemes for EV charging become possible and wholesale and retail billing is supported.

As a charging station operator, GreenFlux independently manages around 3,000 charging stations (both private and public) in the Netherlands while at the same time as eMSP, GreenFlux serves around 1,000 EV drivers. In addition, the GreenFlux office has a parking area with charge points that is used as a testing ground.

4.2 Scenarios Description

For the third pillar of the CAMEL project (electromobility) two scenarios will be demonstrated covering the most important cyber-attacks that exploit known vulnerabilities of the electromobility ecosystem. The first scenario is a distortion attack where the attacker gets the control of the smart-charging network and therefore can potentially cause disruptions to the electric grid through synchronized demands of energy. The second scenario examines the scheduling abuse of the electric grid adopting a decentralized solution for its mitigation. The difference between the scenarios lies in the location of the attack. In the first scenario the attacker has access to the infrastructure (hardware / software) while in the second scenario the attacker can influence the behavior of the user (or his car). The effects of both attacks however can be of similar impact.

4.2.1 Smart Charging Abuse

A cloud-based back office of a CPO communicates with a charge point via the Open Charge Point Protocol (OCPP). This standard is supported by more than 97% of the connected charging stations worldwide. The charge capacity of a charging station can be set from the cloud by means of OCPP requests. Versions 1.6 and 2.0 of this protocol support smart charging. This means that one platform can connect to a wide range of charging stations and still be able to provide smart charging services to all of them.

The charging station then in turn communicates with the EV via the IEC 61851 protocol (for high speed DC charging other standards are used, but these are usually not used for smart charging).

There are a few important observations to be made here:

- Via OCPP, the maximum charge rate for a charge point/socket can be set for a specific period
- The charge point imposes this maximum on the EV
- The EV can choose its own charge rate, as long as it is below the maximum

It is therefore *not possible* to set a specific charge rate for an electric vehicle, only the maximum charge rate can be set.

There are currently around 20,000 charging stations connected to GSOP (GreenFlux Services and Operations Platform). On average, a charging station can charge at around 11 kW. This means that someone with access to GSOP has control over charging stations with a combined capacity of around 220 MW, equal to the power output of a medium-sized power plant. It is expected that around 200,000 charging stations will be connected in 5 years, which corresponds to a potential capacity of around 2 GW. Simultaneous switching on or off of all these charging stations can lead to a pan-European blackout.

The cloud platform receives meter readings from the connected charging stations at fixed time intervals (more information on data flows in this scenario can be found in Section 3.4). These are generally very predictable. Irregularities due to a cyber-attack produce variations in these meter values and can be detected by an ML algorithm.

The protection of the European electric grid should become a priority for all entities involved in the EV ecosystem. The output of this scenario is aiming at increasing the cyber-security of the GreenFlux's platform through the integration of ML techniques for identifying anomalies in the charging patterns, and therefore minimize the exposure of both the enterprise's database and the stability of the electric grid. The scenario covers both the ICT and the electric engineering domain on an effort towards increasing the cybersecurity on what is called Energy Internet [95].

4.2.2 EV Scheduling Abuse

The influx of a large number of electrical loads originating for EVs without any coordination could prove problematic and challenging to the electrical grid. The lack of a proper coordination scheme could cause voltage magnitude drops and unacceptable load peaks even if the total penetration of EV loads does not exceed the 10% [73]. Contrarily, the control of EV loads can minimize charging costs or provide auxiliary services leveraging power electronics.

As evident from the above paragraph, the area of EV scheduling is a very active area of current research, a review of which can be found in [74]. The approaches proposed can be categorized concerning the degree of the centralization of the scheduling control. More decentralized control strategies enjoy more computational resources and enhance user privacy. In [75] a heuristic decentralized EV scheduling mechanism is proposed which is based on congestion pricing used in Internet Protocol networks. A game theoretic approach is described in [76], in which a Nash equilibrium point is proved to exist, but with the unrealistic assumption that all vehicles share the same charging request and plug in/out times. The works described in [77] and [78] propose Lagrange relaxation method, iterative optimal decentralized relaxation schemes. The coordination of the EVs is accomplished through the distribution of locational marginal prices in [79]. Furthermore, in [80] [8] is demonstrated that a feasible valley-filling charging profile is optimal for any convex charging cost and is proposed a decentralized protocol that can be interpreted as a projected gradient descent (PGD). In [81] and in [82] an ant-based swarm algorithm and a multi-agent system for the coordination of charging are proposed respectively. The issue of electrical charging is tackled in a decentralized manner via the alternating direction method of multipliers (ADMM) in [83]. Moreover, in [84] the problem of spatial coupling introduced by transformer capacity limitations is addressed by utilizing a combination of ADMM and PGD. The authors of [85] propose a real-time decentralized charging method based on dual decomposition a projected sub-gradient. With respect to unpredictable load and vehicle plug-in times, an online decentralized charging scheme is described in [86]. Its asymptotic performance is analysed under the presumption that the EV charging requirements can automatically be satisfied. Finally, in [87] by using the water filling scheme, a joint optimal power flow and EV management problem is solved.

As a result of the analysis of the literature discussed above, an EV scheduling abuse scenario will be developed and tested as described in the following subsections. The goal of this scenario is two fold:

- To present potential cyber attacks that affect the scheduling functionalities that are part of the controller installed in the electric vehicle, leading to an overload to the grid.
- To improve security at the vehicle controller via novel asynchronous decentralized approaches which are robust to cyber-failures.

4.3 Enabling Infrastructure

The infrastructure associated with this use case consists of two sides. On the one hand there is the ICT infrastructure around the platform. This consists of the (mainly software-based) facilities that are required to keep GSOP operational. On the other hand, there is the physical infrastructure of a charge points where energy transfer takes place by authorised users identified by a charge card. Both infrastructures are described in more detail in the following sections.

4.3.1 Platform Infrastructure

The GreenFlux Charging Solution runs partly on a public IP infrastructure. A high-level scheme is shown in Figure 47.

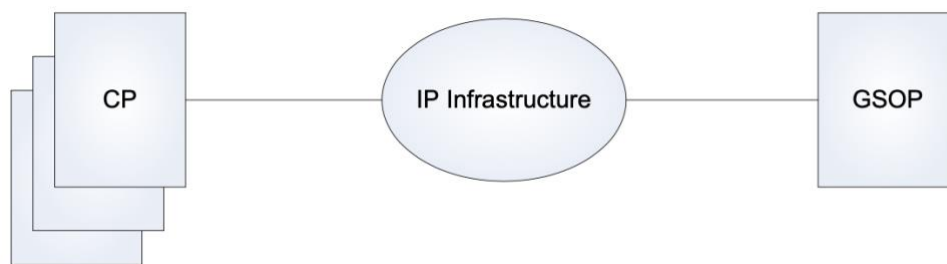


Figure 47: High-level architecture of the GreenFlux Charging Solution

- *GSOP* is the central application; it consists of a set of services that are hosted in the Microsoft Azure Cloud.
- *CP* are the ICU charging stations that are located at the charging premises.
- *IP Infrastructure* is the connecting network between the CP and GSOP. This can be either a public or a private IP network.

GSOP has the possibility to connect to other central applications, such as DSO's and charging service providers. These connections are public internet based.

GSOP

GSOP is a collection of application and storage services that are all hosted on the Microsoft Azure cloud. These services communicate with each other using HTTPS and secure authentication tokens. For Azure the Platform as a Service (PaaS) model is used where the application services run on cloud services hosted by Microsoft. Azure manages the security of the cloud services by means of various threat management activities, including intrusion detection, DDoS attack prevention and penetration testing³.

³ <https://www.microsoft.com/en-us/TrustCenter/Security/AzureSecurity>

CP

The chargepoints are controlled by an embedded device, the controller. The controller is only accessible via GSOP, using the infrastructure and the OCPP application protocol as described below.

Figure 48 illustrates the two possibilities for connections between CP and GSOP.

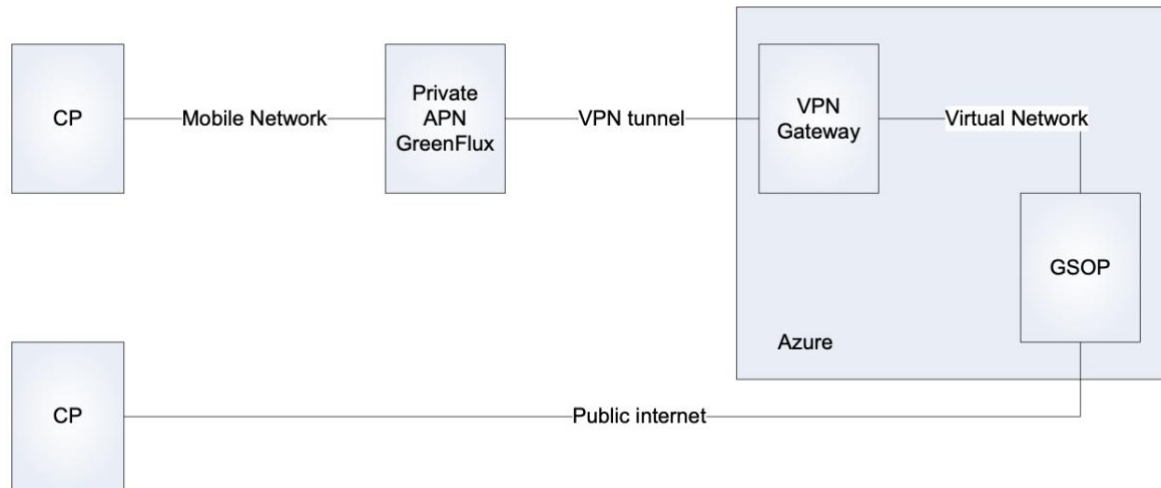


Figure 48: Possible connections between CP and GSOP

Charge points can either be connected via a private network or via the public internet.

- Private connection: The charge point is equipped with a SIM card that is configured to set up a mobile data connection to a private APN (access point). From the private APN a site-to-site VPN tunnel is available to GSOP.
- Public internet: The charge point uses the public infrastructure to reach GSOP.

The private connection is secured and encrypted by means of VPN tunnel. The public internet connection is not secure, therefore the application protocol between CP and GSOP needs to be secure.

Application Protocol

OCPP₄ is the application protocol connecting CPs and the GSOP. It uses web sockets to abstract from the underlying infrastructure. Two versions are possible:

- In a private connection infrastructure, the OCPP web socket connection is set-up without encryption (ws://).
- In a public internet infrastructure, the OCPP web socket connection is set-up with encryption: (wss://). The wss:// requires a server certificate for GSOP. This is either a GreenFlux certificate, when GSOP is deployed by GreenFlux, or a dedicated certificate for a dedicated GSOP instance deployment.

The charge point always initiates the web socket connections; it will not accept web socket connection requests.

Chargepoint Infrastructure

The Charge Point has several functions such as:

- Providing and controlling the energy to the EV using the Electric Vehicle Supply Equipment (EVSE) component
- Collecting the measurements from the meter for each charge of an Electric Vehicle.
- Identifying and authorizing EV users via user authentication component
- Enabling remote capabilities (e.g. adjustment of the maximum current allowed by the Charge Point) to the Charge Point via the Local Controller component over WAN.

Figure 49 illustrates the architecture of the EV Charging Systems that are in scope of this project.

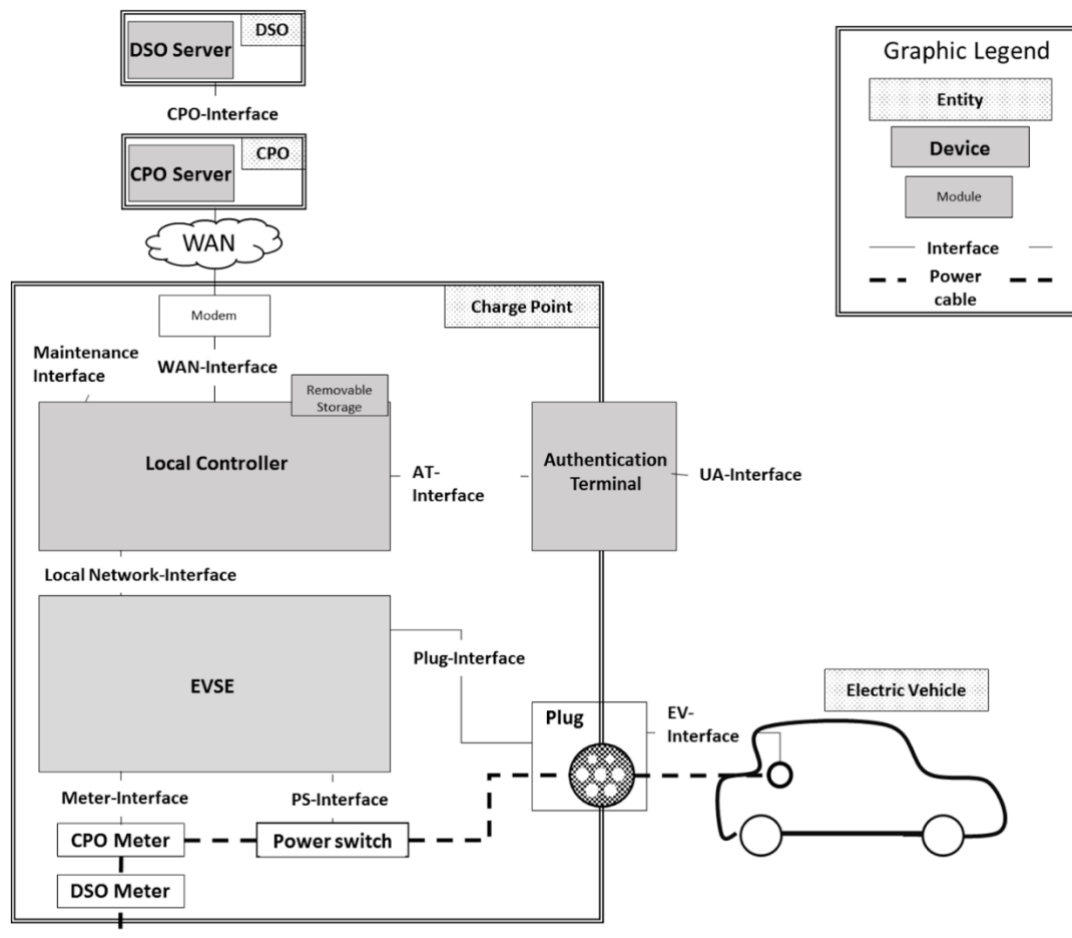


Figure 49: Charge Point System Architecture

The externally reachable interfaces of the ChargePoint are:

1. the WAN interface,
2. the Maintenance interface, and
3. the User Authentication (UA) interface

Note in particular that the internal interfaces in the Charge Point are not covered by security measures. In the current situation most of these interfaces use serial protocols with no security features. This

exclusion of these interfaces implies that the inside of the Charge Point is a trusted environment, meaning that anyone with physical access to the internal systems can compromise the Charge Point.

The Charge Point System Architecture references various items in the Graphic Legend:

- An **Entity** represents a main part of the EV charging system.
- A **Device** identifies the component included in the EV charging system. A device is can contain Modules and can have Interfaces to communicate with other devices.
- A **Module** identifies the physical part of the Device where important functionalities are to be found.
- An **Interface** defines the communication link between two Devices.

4.4 *Data Collection and Selection Methodology*

In this section the information flows and data exchanges between parties involved in the described scenarios are discussed. Parts of these messages are stored in a database and can be used for analysis.

4.4.1 **Meter Values and CDRs**

The interaction starts with the EV driver plugging in the charging cable and swiping his charge card to start a charging session. Figure 50 describes the messages flows between involved parties.

1. **Authentication:** A customer holding a valid RFID card (i.e., authentication token) uses it to authenticate himself at the CP and waits until its validity is checked. Authentication by the CP requires interaction with the CPO and eMSP. The CP extracts the UID of the card and sends an authentication request to the CPO. The CPO contacts the eMSP and checks if the customer is actually authorised to charge at such a CP. A response is then generated and traced back to the CP and to the customer afterwards.
2. **Charging:** In case the customer is allowed to charge, i.e., if authentication is successful, the cable is locked (a pin on the inside of the socket is automatically moved through the car plug). The CP starts the charging by creating an OCPP transaction session that will lock the socket until the customer decides to re-authenticate again. While in charging mode, the CP sends meter readings (MeterValues) to the CPO every 15 minutes. These MeterValues are stored in a database. When the customer wishes to unplug the cable, he has to re-authenticate himself again to the CP (using the RFID card). In most of the cases, this second authentication is performed locally at the CP and does not require any interaction with the CPO. After the session is completed a Charge Details Record (CDR) is forwarded to the eMSP. This CDR includes the customer UID and the total amount of energy charged, the CP identifier (where the charge took place) and the starting and ending time of the transaction.
3. **Billing:** After the whole process is completed (i.e. after charging the car), the eMSP bills the customer. This is usually done on a monthly basis and according to the contracted service.

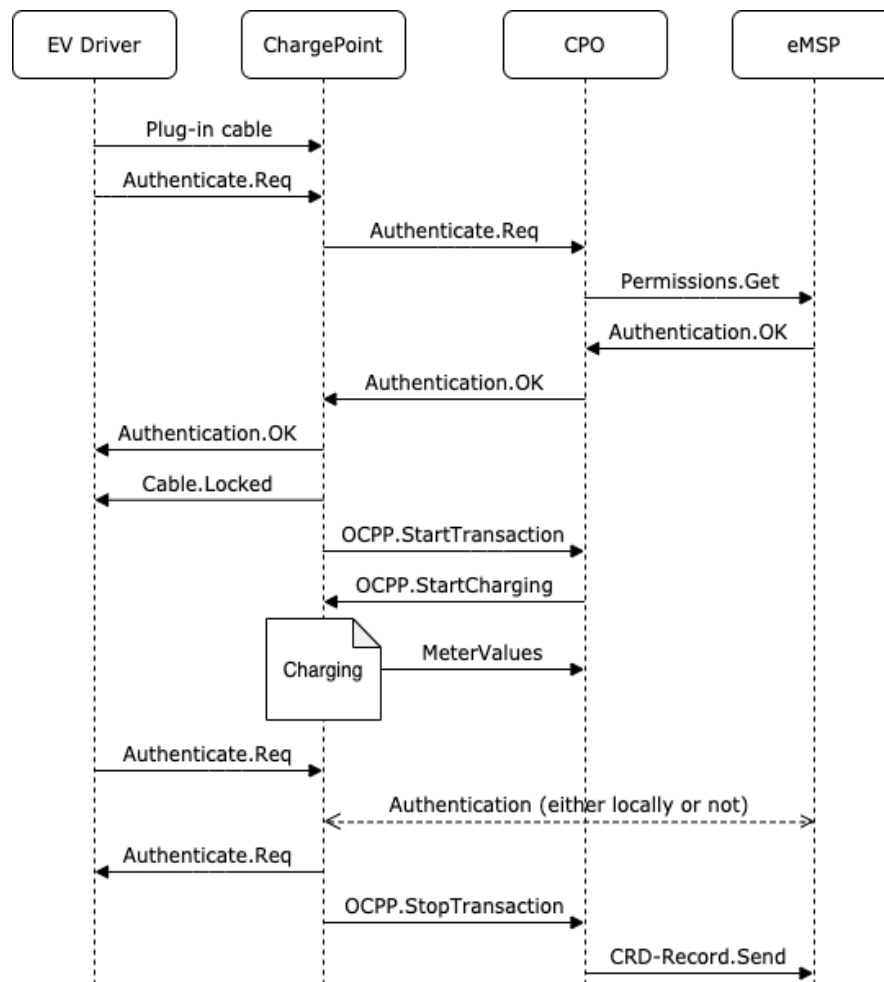


Figure 50: Message flows between involved parties

4.4.2 Smart Charging

In order to be able to forecast and divide the assigned capacities per cable and per eMSP for specific time slots of the next 24 hours, the DSO relies on some variables: historical usage data (of the DSO meter in the transformer), aggregated meter readings per CPO and per cable (from the CPO meter in the CP) and weather forecasts. The DSO makes a forecast based on the historical usage data and weather forecasts of the next 24 hours and uses the aggregated meter readings to divide the capacity per CPO and per cable. The historical usage data is obtained directly from the meter placed in the transformer serving the specific cable. Weather forecasts are retrieved through a web service designated for this purpose. Figure 51 depicts the message flows between the involved parties.

After calculating the forecasts, the DSO sends them (using the OSCP protocol) to the CPO, which in turn forwards them to the corresponding CPs (using the OCPP protocol). Upon reception of the forecasts an acknowledgement message is always sent back to the DSO. The CPs will be able to charge the cars according to the capacity dynamically assigned at each time slot.

Every 24h the CPO can then send the aggregated meter readings per cable to the DSO (the combined meter readings of all CPO meters in all CPs on a cable), such that the DSO can use those values to divide the forecasted capacity and provide them to the energy supplier for billing purposes.

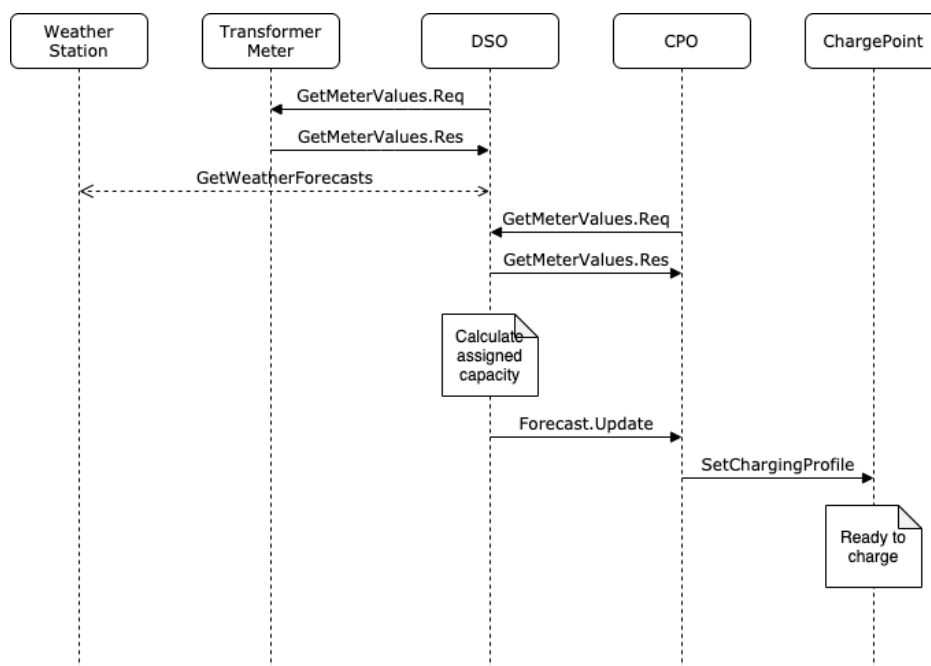


Figure 51: Smart charging message flow

4.4.3 GreenFlux Database

The term smart charging is used to stress the importance of the data exchange between an electric vehicle and a charging station, as opposed to traditional charging where the charging devices are not connected to the cloud. This connection of the charging device to the cloud allows the charging station owner (in our use case GreenFlux) to monitor, manage, and restrict the power supply of each station achieving resources' optimization and ensuring the grid from overcharging issues and a potential blackout.

The cloud-enabled solutions for smart charging allow modifications of the system's features on-the-go without the need for upgrade of the existing infrastructure, offering an effortless and secure way to add or remove features at will. A smart charging solution is a sustainable investment as any potential change on the demands can be turned into new features and added to the current infrastructure. Since the charging stations are connected to the cloud, they can be managed based on various signals (features in the database) such as: the requested volume (kWh), **local electricity consumption**, the start time of the charging, the requested amperage, the amount of other vehicles being charged or **electrical devices being used on a nearby premise**. The inclusion of heterogeneous data requires the installation of a smart charging system that can prioritize the requests from different clients and eventually create a more sustainable energy system based on renewable energy sources.

In the context of CARMEL, the GreenFlux EV charging dataset from GreenFlux's lab in Amsterdam will be used. This dataset is comprised of millions of charge sessions hosted on the cloud platform from throughout the Netherlands dating back to 2012. The database consists of four tables, each one of them representing a **unique entity** in the EV scenario, namely: the Charge Points, the Charge Detail Records (CDRs), the Connections and the Meter Values (MVs).

The table Charge Points represents the actual charging points connected to the GreenFlux platform and contain information for their geo-location features, such as address, zip code and city. as depicted in Table 24. The features of the table provide the necessary information in order to the charge points to

be located both intuitively from the security administrator (country, city, zip), but also contain information that can be used from another software (longitude, latitude) for visualization purposes.

Column	Datatype	Description
ID	PK, Int	Unique ID for Charge Points
ExternalID	Unique ID for Charge Points	External charge point identifier
Address	Nvarchar (255)	Charge point address
Zipcode	External charge point identifier	Charge point zipcode
City	Nvarchar (max)	Charge point city
Country	Charge point address	Charge point Country (NLD = Netherlands)
Latitude	Nvarchar (max)	Charge point latitude coordinate
Longitude	Charge point zipcode	Charge point longitude coordinate

Table 24: An overview of the ChargingPoint table of the GreenFlux's database

The table Charge Detail Records (CDRs), as shown in Table 25, describes the necessary details of each charging attempt such as the duration and the volume, but it also includes features from other tables as foreign key in order to express the correlation with the other entities of the grid. Therefore, every record to the database includes the unique ID of the charge card used by the EV driver, and the unique ID of the charging station. The features of duration, volume and session start/end time have the highest value for the AI algorithms, as they can offer a useful insight for the pattern of a charging session.

Column	Datatype	Description
ID	PK, int	ID for CDR
Duration	Nvarchar (50)	Duration of session
Volume	Nvarchar (50)	Volume in kWh
AuthenticationId	Nvarchar (50)	Unique charge card ID
ChargePoint_ID	FK, int	Unique Charge Point ID
ConnectorId	Nvarchar(255)	ChargePoint Connector Identifier
dStart	datetime	Session start time
dEnd	datetime	Session end time

Table 25: An overview of the Charge Detail Records table of the GreenFlux's database

The different tables in the GreenFlux's database represent different entities in the grid, and the linking of the entities in the database is achieved through the table Connections, as shown in Table 26, the goal of which is to accomplish this connection.

Column	Datatype	Description
ID	PK, int	Connection identifier
ConnectorId	int	ChargePoint Identifier Connector
ChargePoint_ID	FK, int	Unique ChargePoint ID

Table 26: An overview of the Connections table of GreenFlux's database

The last table in the GreenFlux's database is the Meter Values table, shown in Table 27, which consists of information about the actual values of the charging process. Apart from the technical details of the table, it contains the connection_ID as a foreign key in order to be linked with the other entities of the system.

Column	Datatype	Description
ID	PK, int	Meter Value identifier
Timestamp	datetime	In-session timestamp
Value	Decimal(18,2)	Measured Value
ValueType	int	Specifies unit of measurement
ReadingContext	int	(0 = Wh, 1 = kWh, 8 = Amp)
Connection_ID	FK, int	Specifies measurement or instruction

Table 27: An overview of the Meter Values Table of GreenFlux's database

GreenFlux's database also contains tables operating as linking bridges between the entities offering the necessary interconnection. These tables are not mentioned as they do not offer any valuable features that can be used into the analysis of the system under a cybersecurity scope.

4.5 Use of Artificial Intelligence and Machine Learning

4.5.1 Smart Charging Abuse Scenario

The integration of AI/ML techniques into the GreenFlux's smart charging system can secure not only the enterprise's grid but could also prevent potential catastrophic abuse of EU's electrical grid. In the scenario of the smart charging abuse, different users are synchronized (either on purpose or unintentionally) and proceed timely in connection/disconnection actions, causing an unexpected load to the electrical grid. Such actions can be prevented if AI/ML techniques are integrated into the GreenFlux's software.

The data that have been collected in GreenFlux's lab since 2012 can be used as a starting point for getting an insight of the charging stations' behaviour, extracting the attributes of a "normal" charging action and identifying suspicious actions as outliers. As an outlier we define a charging process that cannot be grouped into what is called expected behaviour (inlier), for example an EV requests a greater amount of power, or a specific car(s) is/are charged on a different station(s) from the usual ones, showing an unusual behaviour that should be further investigated.

The detection of outliers is achieved either through the application of unsupervised ML algorithms or through the definition of a series of logic rules that should cover the entire spectrum of what is characterised as "normal behaviour". In the context of CARMEL, the available data will be used, and a series of unsupervised ML algorithms will be deployed in an effort to identify actions that are potential threats to the electrical grid. As long as the objective is to identify abnormal activity that can harm the electric grid, there is no need to proceed into simulating specific cyber-attack attacks and afterwards deploying supervised classification algorithms.

The abuse detection workflow is depicted in Figure 52 offering a more intuitively approach for the application of the ML algorithms for the improvement of the EV's cybersecurity. The algorithm receives both real-time data (meter values) and historic values from the GreenFlux's database that are fed into the abuse detection system predicting if the incoming event is a legitimate or indicates a cyberthreat. In the latter case the CPO and the DSO are warned, otherwise the charging continues.

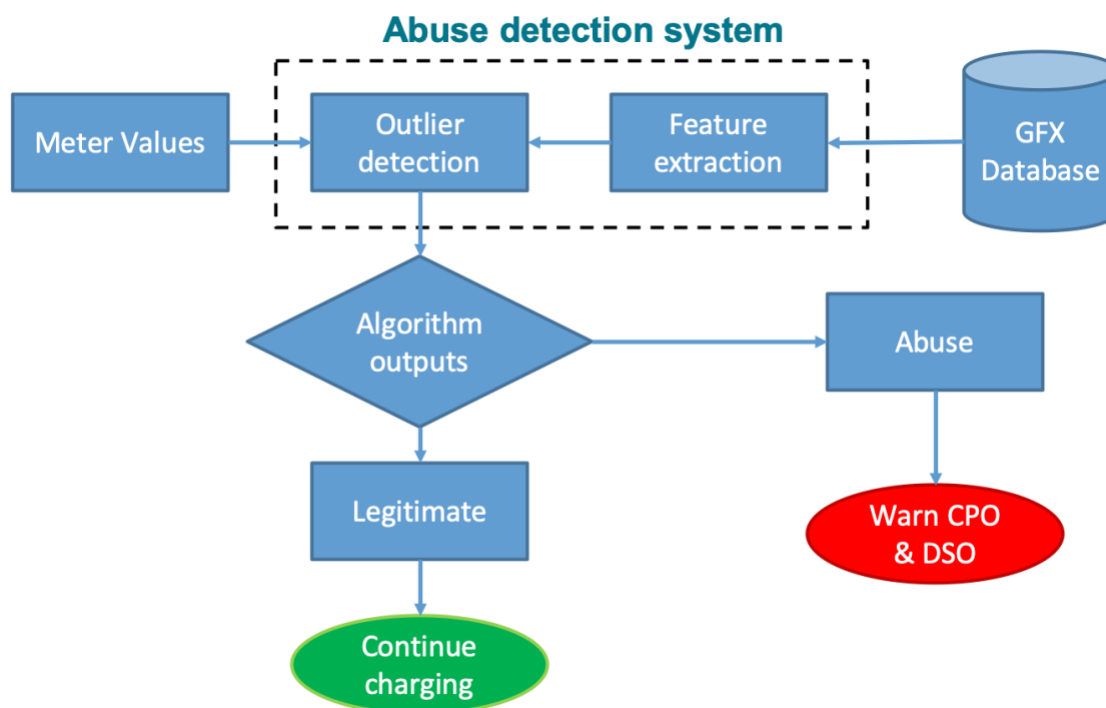


Figure 52: An overview of the Abuse Detection workflow

A series of different methods will be deployed in the collected data, aiming to not only deliver a proof-of-concept demonstration but also to provide some benchmarks for the effectiveness of different algorithms on the field of EV cybersecurity, introducing a data-driven approach for secure smart charging. The integration of AI/ML into the cybersecurity is a necessity towards the cyber-crisis management on EU level, as it provides the chance to obtain actionable insights from large amounts of data (big data), such behavioral activity otherwise unfeasible for humans to analyze [99]. On a more technical level, three different methods will be deployed in the smart charging abuse scenario:

- **Z-Score (Standard Deviation):** For different features (e.g. Volume, Duration, ReadingContext) of GreenFlux's dataset it is going to be tested if a registration (e.g. charging action) is within three

standard deviations, if not then it is considered as an outlier. Figure 53 presents a graphical representation of a normal distribution and its correlation with standard deviation. For example, the number of cars on a working day for a specific datetime can be predicted based on the mean value of previous weeks; if on a specific datetime the connected cars are more than 2 standard deviation away from the mean value, it is indicative of an anomalous behavior. It has to be mentioned that the standard deviation score is not a ML algorithm, but it is often used as a benchmark for unsupervised ML algorithms, as it is an effective approach that can be easily understood from the human perspective. A drawback of the approach is that can only be used with parametric distributions in a low dimensional feature space, thus in a future expansion of the feature collected from the GreenFlux's software it might not be a suggested approach.

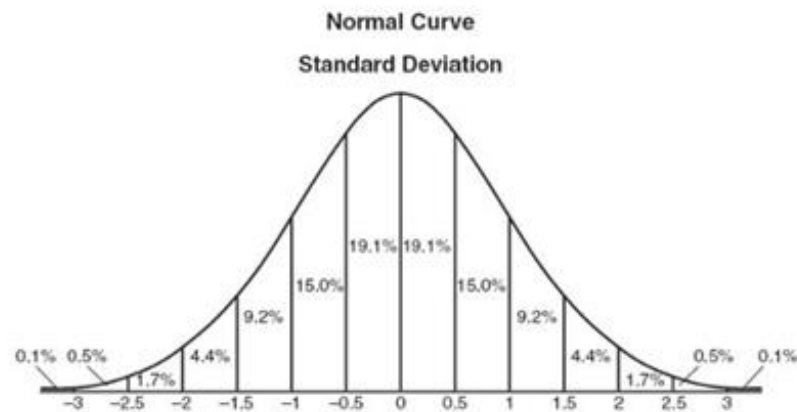


Figure 53: An example of standard deviation applied on normal distribution

- Density-based functions:** Density-based algorithms rely on the hypothesis that density of points around an outlier point is significantly lower compared to an inlier point. The rationale is based on the hypothesis that similar events (e.g. EV charging on a specific CP) have (very) similar characteristics (e.g. charging time) in contrast to abnormal activity that might indicate a cyber-threat. As depicted in Figure 54, every point/event has a specific distance to any other point/event when mapped into a 2-d representation. Based on this distance different clusters of similar events can be created, group similar events, and therefore locate outliers.

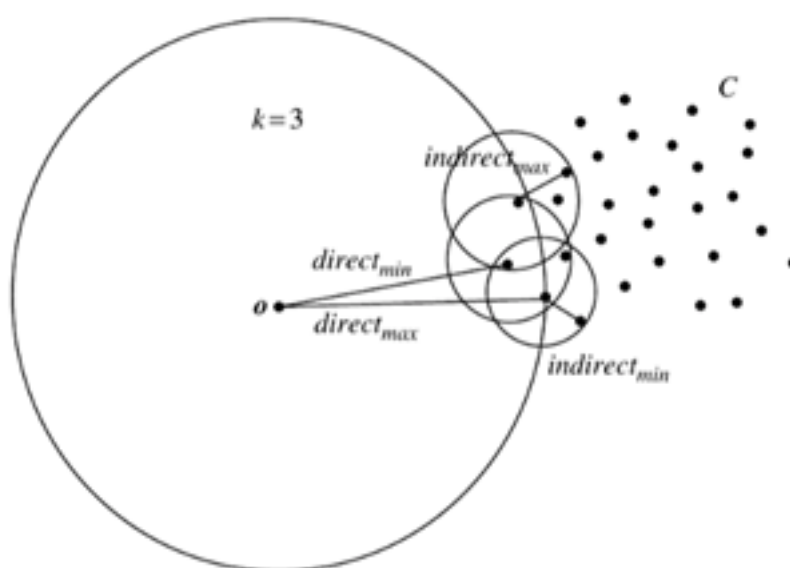


Figure 54: A graphical representation of density-based functions

The density-based algorithms also offer an intuitive graphical interpretation when applied on single-dimensional data (power requested, time of charging, etc.). The algorithms can also work on multi-dimensional data and are actually suggested for multidimensional feature space ($n > 3$), but the visualization of the results is not easily interpreted by humans. In future expansions of GreenFlux's dataset which includes more features, the density-based approach can have an even greater impact. A drawback of this approach is the arbitrarily definition of the k variable, which may cause overfitting-related problems.

- **Isolation Forest:** Isolation forest is an unsupervised learning algorithm that belongs to the ensemble decision trees family, isolating anomalies instead of creating clusters of similar events such as the density-based algorithms. The algorithm is based on the concept that anomalies have extreme values compared to normal instances and therefore it is possible to isolate them. The isolation algorithm is based on iterations that generate partitions of the data sample by randomly selecting an attribute and then selecting a split value for this attribute. This recursive partition of the sample can be represented by a tree structure named Isolation Tree, and the length of the path is the number of iterations that are required in order to isolate the path. A graphical illustration of the algorithm is depicted on Figure 55. The left side depicts the multiple iterations of the algorithms in order to isolate an inlier, in contrast to right side, where the algorithm isolates the outlier point in 5 iterations. For instance, a typical charging action does not present any specific characteristics that can easily lead to its isolation it similarly to the left part of the figure, but a charging action that requests an unusual volume can be isolated faster from the algorithm, similar to the right part of the figure.

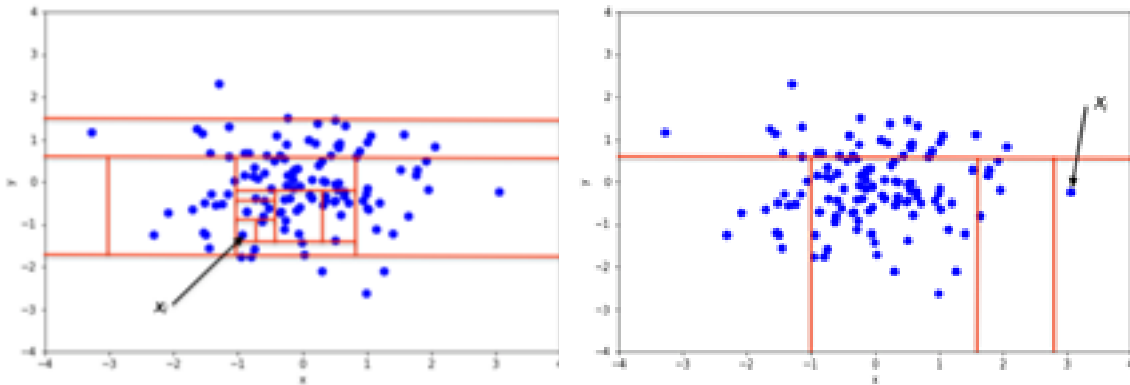


Figure 55: Overview of Isolation Forest Algorithm output

$$c(m) = \begin{cases} 2H(m-1) - \frac{2(m-1)}{n} & \text{for } m > 2 \\ 1 & \text{for } m = 2 \\ 0 & \text{otherwise} \end{cases}$$

A characteristic of the isolation forest is the algorithm's ability to perform well even if the training set does not contain any anomalous point, because the algorithm perceives the high values of a path length $h(x)$ of an outlier x instance as another data point. Therefore, there is no need for generating data that mimic anomalous behavior and integrated them into the existing GreenFlux's database.

4.5.2 EV Scheduling Abuse Scenario

The work presented in [88] tackled with the problem of EV charging scheduling and in addition examined an asynchronous charging scheme for dealing with cyber failures/attacks in communication links. More specifically the authors describe a decentralized charging protocol based on the Frank-Wolf method. The devised scheme provides fast convergence, especially during the first iterations. The overall computational time is reduced since its closed-form updates pose minimal processing requirements for the controllers of the vehicles. Conducted numerical experiments have demonstrated a 100-times speed-up advantage over existing alternatives. However, the feature that differentiates this approach

from the others is the provision for coping with communication failures between the vehicle and the data aggregator. The notion behind this is the adoption of an asynchronous variant of the charging scheme. The judicious modification of the step size produces convergence rates that keep up with the respective rates produced by the synchronous method. The work complements the work in [89], where the plain Frank-Wolfe scheme was used as a basis for constructing synchronous charging protocols complying with distribution grid constraints.

The Frank-Wolfe algorithm is a first-order optimization algorithm for constrained convex optimization. The method, which is also known as conditional gradient algorithm, selects an initial feasible solution vector and iterates for a step size $\gamma_k \in (0,1]$. The step size can be determined by selecting a proper value so that a faster convergence rate can be achieved, especially in the first iteration. But firstly we define the conditions of the problem. Supposedly we have a fleet of N EVs that have to be charged over a period of T consecutive time slots by a data aggregator. The time slots comprise the set $\mathbf{T} := \{t: t=1, \dots, T\}$. The charging rate of each vehicle at every time slot is denoted by $p_n(t)$ and its value lies between zero and a maximum value $p_{n(t)}$ that is determined mainly by the battery specifications of the vehicle. Every vehicle is charging only when is connected and thus the maximum charging rate of a vehicle n has the maximum value only for a subset \mathbf{T}_n of \mathbf{T} , during which the vehicle is connected equals to zero. Outside this subset the maximum charging rate equals to zero. At the end of time horizon T , its vehicle has consumed total energy represented by B_n which depends on the battery and the initial and final charging state of the vehicle. Therefore, for each vehicle exists a specific charging profile $p_n := [p_n(1) \dots p_n(T)]$ which lies in the set $P_n := \{p_n : p_n T = B_n, 0 \leq p_n(t) \leq p_{n(t)} \ \forall t \in \mathbf{T}\}$ which is convex and compact. The goal of the aggregator is to minimize the electricity cost by the optimal EV charging problem, proved that solving the aforementioned optimization charging problem is rendered equivalent to solving another problem ensued by the previous one by using the electricity cost $H(p)$ quadratic function. The advantage of the latter is that it can be efficiently solved utilizing the Frank-Wolfe algorithm since $H(p)$ is convex differentiable and P is convex and compact.

4.6 Validation Methodology

4.6.1 Smart Charging Abuse

The smart charging abuse scenario is based on real data that collected by GreenFlux since 2012 on its lab in Amsterdam, therefore no simulation needs to take place. The challenge that is created when unsupervised ML algorithms are applied lies on the evaluation of the algorithms' performance; a known problem in the scientific community [90][91]. The goal of the clustering algorithms is to define separations of the data based on some assumptions such that the points of a cluster are similar to some ground truth set of classes or the point satisfies the hypothesis that the points belong to the same cluster present higher similarity in compared to points outside of the cluster. In the context of CAMEL, the first hypothesis could end up on creating categories of electric charges based on the V consumption that is requested, and in the second alternative some probability measure (log-likelihood, perplexity, etc.) can be applied on the data.

Another approach that can be followed is a combination of supervised and unsupervised ML techniques. An iteration of this combination is described as follows: step 1) application of the clustering algorithms, step 2) comparison with the results of the supervised algorithms, step 3) adjusting the clustering algorithm (e.g. number of clusters), step 4) repeat step 1. In this case, it is necessary that a fair amount of data is annotated either through the experienced human annotators that can identify an outlier to the system either through the simulation of abnormal activity on the charging station.

For the smart charging abuse scenario, three different evaluation methods will be applied in order to ensure high quality to the findings of the ML techniques. The applied evaluation methods will contain both qualitative (visual inspection, manual investigation) and quantitative metrics (accuracy, precision) offering a validation framework wide enough to cover different aspects of cybersecurity in the area of EV smart charging. More technical details on the validation methods are illustrated on the next subsection, where the validation methods are expressed through tables.

4.6.2 EV Scheduling Abuse

Each EV is equipped with a controller capable of communicating with an aggregator and performing simple computation tasks, like selecting the maximum possible charge s_{nk} during cheapest time slots following the method described earlier. The controllers are presumed as nodes of a tree graph where the root is the aggregator server. This architecture matches satisfying the physical system structure of a radial information router system. According to the proposed method the aggregator optimally selects the time step and broadcasts the corresponding information to the EV controllers. Each EV controller updates its charging profile and calculates s_{nk} which forwards either to the aggregator directly or the next car in the tree (Figure 56). In addition, in [92] a Proper Generalized Decomposition (PGD) approach was proposed (Figure 57) for solving the problem of minimizing the optimal EV charging problem with a converging at most $O(1/k)$.

The previous algorithm assumes that each EV controller would be able to update its charging profile according to the current control signal. However, this does not correspond to realistic circumstances since in practical charging scenarios the EV controller may not be able to update their charging profiles synchronously. The cause of this could be a failure in a communication link or delays caused by the calculation conducted in the EV controllers. In such scenarios the step size γk has to be modified to guarantee the convergence of the Algorithm. This asynchronous variant ensures the proper functionality of the algorithm against random cyber delays which otherwise could lead to an EV scheduling abuse.

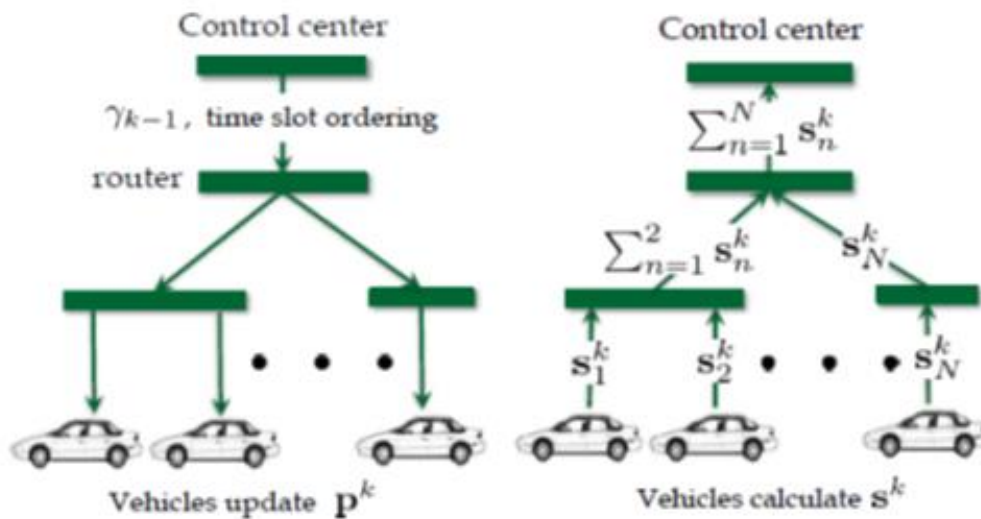


Figure 56: Information flows for Synchronous Algorithm at iteration $k \geq 1$. Left: Aggregator broadcasts time slot ordering to EVs. Right: Summation of intermediate charging profiles are forwarded to aggregator

The efficiency of the devised charging scheme will be verified by conducting a simulation including 52 EVs each with a battery capacity of 24KWh and a maximum charging power fixed to 3.45kW. Moreover actual survey data [93] will be used for determining the EV plug-in and plug-out times as well as the daily travel miles. The expected state of charge (SOC) will be set for each vehicle to 90%. The Energy needed for 100km will be defined to $E_{100} = 15\text{kWh}$ and the initial SOC will be calculated by the formula $S_{no} = 0.9 - M_n E_{100} / (100 B_n)$ where M_n are the travel miles per day for vehicle n and B_n is the battery capacity of vehicle n . Normalized base load curves with base unity 1000kW will be acquired by averaging the 2014 residential load data from Southern California Edison [94]. The time horizon will be defined in the period between 12:00 pm and 12:00 pm the next day and it will be comprised of $T = 96$ time slots. During no EV scheduling, the EVs will be assumed to start charging as soon as they plug-in and stop charging until they reach the desired SOC level.

The experimental setup regarding the EV Scheduling Abuse will be performed in a simulated environment.

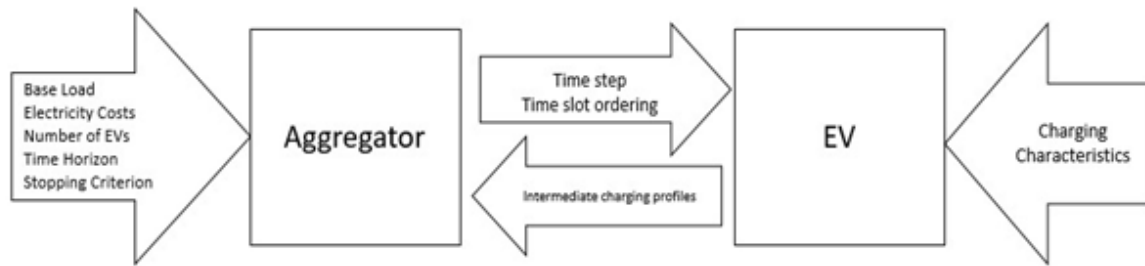


Figure 57: Block Diagram describing optimal decentralized algorithm

4.6.3 Scenario Validation Overview

Table 28 provides an overview summarising the algorithms used per scenario, their challenges and the associated metrics that are used to assess the algorithm's performance.

Scenario	Algorithms	Challenges	Metrics
Smart charging abuse	Z-score	Application in n-dimensional data	Manual evaluation
	Density-based function	Evaluation of the results	Perplexity, coherence
	Isolation forest	Training time	Combination with supervised ML algorithms
EV Scheduling Abuse	Frank-Wolfe	Charging of even a penetration 10% of EV loads will cause voltage magnitude drop and unacceptable load peaks.	Load curves
	Projected gradient descent (PGD)	Decentralized charging protocols for electric vehicles.	Convergence performance of Frank-Wolfe and PGD algorithms
		Coping with cyber failures in the communication link between the aggregator and the vehicle controllers.	

Table 28: Overview of Validation for Pillar 3

The Smart Charging Abuse scenario can be summarized as shown in Table 29 listing the flow of the actions that will take place in the GreenFlux's software in order to increase the cybersecurity of the charging stations and prevent potential cyber-threats.

Scenario Name	Smart Charging Abuse
Related Use Case	Electromobility
Brief Description	The attacker(s) occupy (physically or remotely) the available charging stations starting and proceed timely in connection/disconnection actions creating an enormous load to the electric grid.
Challenges	<ol style="list-style-type: none"> 1. Ability to detect the attack based on the recognition of anomalies in the pattern. 2. Timely detection of the attack.
Assumptions & Pre-Conditions	<ol style="list-style-type: none"> 1. We have collected data based on the "normal" behavior of the vehicle. 2. There is successful monitoring of the charging stations.
Goal (Successful End Condition)	The attack has been recognized from the GreenFlux's software and has informed both the security administrator and warns the owner of the vehicle about abnormal behavior.
Involved Actors	Malicious attacker Charging station GreenFlux Infrastructure
Scenario Initiation	The cyber-attacker(s) gain access to different charging stations.
Novelty	<p>This is a scenario combining physical attack on a smart vehicle with anomaly detection algorithms. Its importance lies in its capability to combine V2X communication security with real-world attacks.</p> <p>The field of security electromobility is a very premature scientific area, therefore few (if any) studies have been applied on real data. The results of the scenario will be a significant outcome for both cybersecurity and electromobility communities.</p>
Main Flow	<ol style="list-style-type: none"> 1. The cyber-attacker(s) gain access to charging stations. 2. There is a synchronization intermittent energy demand. 3. GreenFlux's software recognizes the anomaly. 4. GreenFlux automatically interrupts the electrical and data exchange with the specific charging stations.
Evaluation Criteria	GreenFlux's software detects the attack and isolates the charging stations. GreenFlux's software detects the attack in under 15 minutes.

Table 29: Overview of the Smart Charging Abuse Scenario

In a similar manner, Table 30 provides an overview of the measures that are being taken to set up a system capable of detecting the abuse of EV scheduling. The main challenge here is, with the available

data stream, to remotely detect an attack on connected charging infrastructure before damage can be done to the power system.

Scenario Name	EV Scheduling Abuse
Related Use Case	Electromobility
Brief Description	<p>With proper coordination scheme, EV loads can be controlled to minimize charging costs.</p> <p>Attacker(s) occupy (physically or remotely) the available charging stations. Uncoordinated charging of even a 10% penetration of EV loads will notably affect power system operation, giving rise to voltage magnitude fluctuations and unacceptable load peaks.</p>
Challenges	<ol style="list-style-type: none"> 1. EV decentralized charging protocol 2. Ensuring privacy of EV owners during charging 3. Anomaly detection
Assumptions & Pre-Conditions	<ol style="list-style-type: none"> 1. We have collected data based on the “normal” behaviour of the vehicle. 2. Provided by GFX, Southern California Edison, IEEE 123-bus feeder
Goal (Successful End Condition)	EV decentralized charging has been succeeded, while preserving the privacy of EV owners and detecting anomalies
Involved Actors	<p>Malicious attacker</p> <p>Charging station</p>
Scenario Initiation	The cyber-attacker(s) gain access to different charging stations, and possibly disabling decentralized EV charging
Novelty	<p>A decentralized charging method based on the Frank-Wolf algorithm</p> <p>The decentralized protocol requires communication only between neighbouring vehicles and preserves the privacy of EV owners.</p>
Main Flow	<ol style="list-style-type: none"> 1. The cyber-attacker(s) gain access to charging stations. 2. There is a synchronization intermittent energy demand. 3. UPAT's software recognizes the anomaly. 4. UPAT's software tries to detect anomalies, while decentralized charging is still operating efficiently
Evaluation Criteria	<p>Load curves after the proposed charging protocol</p> <p>Convergence performance of proposed algorithm</p>

Table 30: Overview of the EV Scheduling Abuse Scenario

4.7 Use of the Anti-Hacking Device

In the electromobility scenario no anti-hacking device will be physically deployed in EVs as in the other use cases. Instead, we will take advantage of the simulated anti-hacking device devised for the development phase of the other use cases and deploy it on a cloud service in a virtualized environment.

The anti-hacking device will take advantage of the advanced computing resources available in the cloud environment (eg. using data center GPUs or TPUs depending on the chosen cloud platform).

Figure 58 shows the familiar two machine learning phases in the electromobility scenario:

- In a first phase data from the GreenFlux dataset is analysed using the methodologies described in the previous sessions in order to train and develop attack detection models.
- The resulting attack and threat detection models are transferred to the anti-hacking device (here running as a cloud service). The anti-hacking is then fed test data containing normal and attack situations. The anti-hacking device will report attack situations appropriately (e.g., on an operator console).

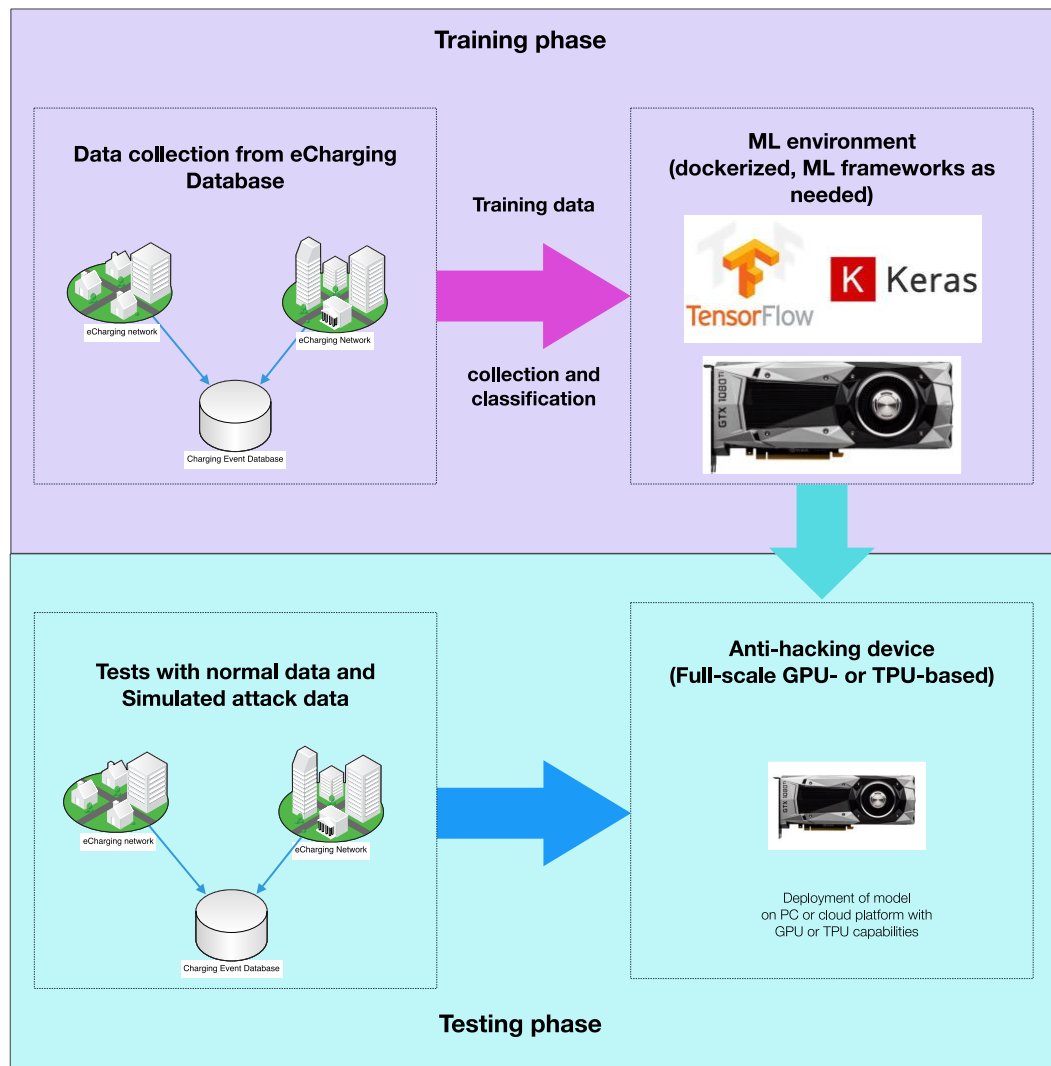


Figure 58: Use of anti-hacking device in electromobility use case

4.8 Functional Requirements

Reg ID	CRPL3-FR01
Title	Train on existing data
Definition Description	- The CARMEL platform must be able to train on existing data
Target WP	4
Priority	Mandatory
How addressed	The CARMEL platform will provide a data management environment where (anonymized) historical loading data is stored. This data is used to train ML models.

Reg ID	CRPL3-FR02
Title	Smart charging simulation
Definition Description	- The CARMEL platform must be able to simulate the behaviours of EVs that are (smart) charged and thereby generate synthetic data
Target WP	4
Priority	Mandatory
How addressed	Part of the CARMEL platform will be a simulation environment capable of generating synthetic charging data. In addition, there will be a module that injects false data into the data stream. The contaminated data will be used as input for a ML model.

Reg ID	CRPL3-FR03
Title	Anomaly detection in ChargeProfiles
Definition Description	- The CARMEL platform must be able to detect anomalies in ChargeProfiles that are sent from GSOP to the ChargePoints when charging sessions are active.
Target WP	4
Priority	Mandatory
How addressed	The CARMEL project will provide a tool that is able to detect irregularities in loading profiles that are sent by GSOP via OCPP to a charging station. Since not all messages can be checked, a trade-off must be made between accuracy and responsiveness.

Reg ID	CRPL3-FR04
Title	Detection of physical attacks
Definition Description	- The CARMEL platform must be able to detect physical attacks on charging infrastructure by recognizing incorrect or unexpected messages that are sent from a compromised charging station
Target WP	4
Priority	Mandatory
How addressed	The CARMEL project will provide a software tool that is able to detect if a charging station is physically attacked. If someone unobtrusively wants to take control of a charging station, they will have to keep sending messages to GSOP. It can be deduced from these messages whether they are sent by the charging point or are 'fake'.

5 Non-Functional Requirements

There are two main important non-functional requirements in the overall CARMEL activities, i.e. ethical related issues and data management and protection.

The ethical aspects in the CARMEL project are limited to potential interaction with data subjects as part of relevant research and in the use cases and in dealing with personal data in general.

The project addresses them by:

- An ethical assessment and an assessment if any of the planned activities require ethical opinion, authorization or confirmation;
- An ethics report explaining the result of this exercise and providing all necessary documents;
- A data management plan, which includes best practice approaches and policies to be implemented by the consortium. The data management plan is to be updated in iterations;
- For any direct interaction with natural persons which require the collection of personal data, this will be justified, and any collection will be accompanied by a specific privacy notice and if based on consent a consent form, specifically drafted for the CARMEL project by the partner 8BELLS.

Data Management Plan (DMP) is a written formal document that describes how data will be handled until the completion of the project and after it. The Guidelines on FAIR Data Management in Horizon 2020 [100] provide a set of principles and criteria that have to be addressed. Research data should become Findable, Accessible, Interoperable and Re-usable (FAIR). The CARMEL DMP will describe in detail the data that the project will collect/generate, the methodologies and standards that will be followed to make research data FAIR, the data that will be shared/made open, and how they will be curated and preserved during and after the lifetime of the project.

Data sharing in the open domain can be very beneficial to society, however, we need to balance openness on the one hand and protection of sensitive data on the other hand. As stated in the Guidelines on FAIR Data Management [100] data should be 'as open as possible and as close as necessary'. All data providers that participate in the consortium should comply with all applicable data protection or similar laws regulating the processing of any personal data.

CARMALE will provide a detailed plan of the mentioned non-functional requirements on D1.2 Ethics Framework and Data Management Plan. Agreed procedures and steps on D1.2 will be followed by all partners, securing the satisfactory achievement of the non-functional requirements of CARMEL. The Ethics and Data Management Committee of CARMEL (composed by 8Bells, T-SYS, PANA and chaired by the project coordinator) is responsible to monitor the defined framework under D1.2.

6 Conclusion

D2.1 reports on detailed specification of CARMEL use cases. The document was organized around three main pillars of the project, namely: autonomous vehicles, connected and cooperative cars, and electro mobility. For each use case D2.1 provides an extensive overview of the scenarios that will be investigated and showcased by the project. Data collection and selection processes as well as the use of AI/ML solutions are important parts of this document. In addition, D2.1 determines the validation methodology for each CARMEL scenario, i.e., whether simulation and/or demonstration activities will be executed depending on the characteristics of each use case.

The Autonomous Mobility pillar classified the two main possible attacks as physical adversarial attacks and attacks on the vehicle's camera sensor, the detection and overcoming of which can be achieved by the use of trained ML models. For the Connected Mobility pillar, the project will examine three scenarios: location spoofing, V2X messages attack and physical tampering of the vehicle's OBU. The first two will be addressed by employing AI/ML techniques, while the approach for the last one will be based on HW techniques. Finally, the Electromobility pillar examines the susceptibility to threats on the smart charging scenario, where AI/ML approaches will be implemented to secure the system.

In addition to the presented material, CARMEL also would like to include some extra cases under D2.1, especially regarding the urban mobility. In order to enhance the safety level, modern autonomous driving solutions running on urban and suburban environments build their environmental modelling part through combining multiple sources of information coming not only from the sensors on board and from the infrastructure, but also through embodying the footprint of occurrence of other agents as it is communicated through V2X. This approach attempts to approximate the safety aspect in an Internet of Things fashion, where each road user (vehicle/traffic sign/pedestrian) acts as a device broadcasting its status and seeking confirmation from the perception engines. ETSI EN 302 637-2 introduces this concept under the term cooperative awareness and assumes a unified fusion framework of building scene awareness by fusing scene elements observation through a multitude of sources, which are coming not only by the sensing modalities, thus being less vulnerable to weather and lighting conditions.

As a result of the existence of the aforementioned techniques, CARMEL considers that such cooperative awareness schemes could be subjected to cyber-attacks quite easily. Thus, the scenario could be investigated in case that the resources and the setup during the execution of WP4 of the project are sufficient. This particular direction is considered as an element of future extension as according to the consortium's knowledge, ADAS solutions embodying cooperative awareness features have not been commercialized yet. It is worth to mention that this case will not be deeply investigated/demonstrated by the CARMEL project.

Figure 59 illustrates the concept of operating a Cooperative Awareness solution in autonomous driving.

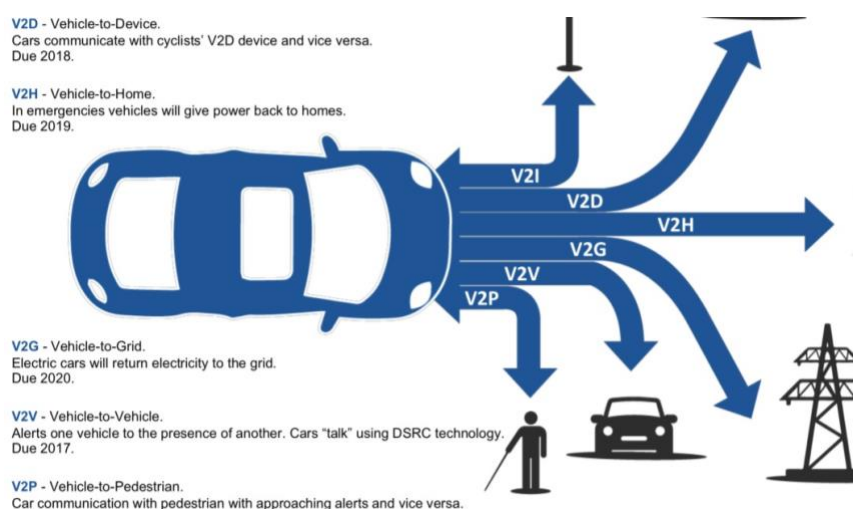


Figure 59: Autonomous Driving Solution Scheme approximated as in IoT

The scenarios described in this document were targeted by CARMEL to assess the solutions proposed by the project to overcome the main cybersecurity issues for the autonomous and connected vehicles. They are a broad representation of the potential cyberattacks that the future mobility will experience.

References

- [1] E. Stav, S. Walderhaug, and U. Johansen, ARCADE - An Open Architectural Description Framework. December 2013, SINTEF ICT
- [2] <https://www.managementstudyguide.com/desk-research.htm>
- [3] Tobin, Josh, et al. "Domain randomization for transferring deep neural networks from simulation to the real world." 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2017.
- [4] Tremblay, Jonathan, et al. "Training deep networks with synthetic data: Bridging the reality gap by domain randomization." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.
- [5] Inoue, Tadanobu, et al. "Transfer learning from synthetic to real images using variational autoencoders for robotic applications." arXiv preprint arXiv:1709.06762 (2017).
- [6] Naveed Akhtar and Ajmal Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey", 2018, IEEE Access, vol. 6, pp. 14 410–14 430.
- [7] Jiajun Lu, Hussein Sibai, Evan Fabry, David Forsyth, "Standard detectors aren't (currently) fooled by physical adversarial stop signs", 2017, arXiv preprint arXiv:1710.03337.
- [8] Jonathan Petit, Bas Stottelaar, Michael Feiri, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar", 2015, Black Hat Europe, vol. 11.
- [9] Dudi Nassi, Raz Ben-Netanel, Yuval Elovici, Ben Nassi, "MobilBye: Attacking ADAS with Camera Spoofing", 2019, <https://arxiv.org/abs/1906.09765>
- [10] Liu Weizhe, Salzmann Mathieu, Fua Pascal, "Using Depth for Pixel-Wise Detection of Adversarial Attacks in Crowd Counting", 2019, <https://arxiv.org/abs/1911.11484v1>
- [11] Ranjan Anurag, Janai Joel, Geiger Andreas, and Black Michael, "Attacking Optical Flow", 2019, Proceedings of the International Conference of Computer Vision.
- [12] Athalye Anish, Carlini Nicholas, Wagner David, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples", 2018, <https://arxiv.org/abs/1802.00420v4>.
- [13] Sharif Mahmood, Bhagavatula Sruti, Bauer Lujo, Reiter Michael K., "Adversarial Generative Nets: Neural Network Attacks on State-of-the-Art Face Recognition," 2017, arXiv:1801.00349.
- [14] Evtimov Ivan, Eykholt Kevin, Fernandes Earlene, Tadayoshi Kohno, Atul Prakash, Amir Rahmati, Dawn Song, "Robust Physical-World Attacks on Machine Learning Models", (2017), <https://arxiv.org/abs/1707.08945v5>
- [15] Hussain, R., & Zeadally, S. (2019). Autonomous Cars: Research Results, Issues, and Future Challenges. IEEE Communications Surveys & Tutorials, 21, 1275-1313.
- [16] Sitawarin, C., Bhagoji, A.N., Mosenia, A., Chiang, M., & Mittal, P. (2018). DARTS: Deceiving Autonomous Cars with Toxic Signs. ArXiv, abs/1802.06430.
- [17] Qayyum, A., Usama, M., Qadir, J., & Al-Fuqaha, A.I. (2019). Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and The Way Forward. ArXiv, abs/1905.12762.
- [18] Morgulis, N., Kreines, A., Mendelowitz, S., & Weisglass, Y. (2019). Fooling a Real Car with Adversarial Traffic Signs. ArXiv, abs/1907.00374.
- [19] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D.X. (2018). Robust Physical-World Attacks on Deep Learning Visual Classification. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1625-1634.
- [20] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, Volker Fischer, "Universal Adversarial Perturbations Against Semantic Image Segmentation", ICCV 2017.
- [21] Simen Thys, Wiebe Van Ranst, Toon Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection", 2019, <https://arxiv.org/abs/1904.08653>
- [22] Adith Bloor, Xin He, Christopher D. Gill, Yevgeniy Vorobeychik and Xuan Zhang, "Simple Physical Adversarial Examples against End-to-End Autonomous Driving Models", 2019, <http://arxiv.org/abs/1903.05157>
- [23] Cory Cornelius, Jason Martin, Shang-Tse Chen, Duen Horng (Polo) Chau, "Towards the Realistic Evaluation of Evasion Attacks using CARLA", 2019, <https://arxiv.org/abs/1904.12622>
- [24] Mogelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1484-1497.
- [25] Neuhold, G., Ollmann, T., Rota Bulò, S., & Kotschieder, P. (2017). The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4990-4999).
- [26] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). The German traffic sign recognition benchmark: a multi-class classification competition. In The 2011 international joint conference on neural networks (pp. 1453-1460). IEEE.
- [27] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., & Igel, C. (2013, August). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In The 2013 international joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.

- [28] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.
- [29] Di Mattia, F., Galeone, P., De Simoni, M., & Ghelfi, E. (2019). A survey on gans for anomaly detection. arXiv preprint arXiv:1906.11632.
- [30] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., & Efros, A.A. (2016). Context Encoders: Feature Learning by Inpainting. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2536-2544.
- [31] Jia, Y., Xing, Y., Peng, C., Jing, C., Shao, C., & Wang, Y. (2019, August). Semantic image inpainting with boundary equilibrium GAN. In Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition (pp. 88-92).
- [32] Berlincioni, L., Becattini, F., Galteri, L., Seidenari, L., & Del Bimbo, A. (2019). Road layout understanding by generative adversarial inpainting. In Inpainting and Denoising Challenges (pp. 111-128). Springer, Cham.
- [33] Akhtar, N., & Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, 6, 14410-14430.
- [34] Narodytska, N., & Kasiviswanathan, S. (2017, July). Simple black-box adversarial attacks on deep neural networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 1310-1318). IEEE.
- [35] Y. Vorobeychik and M. Kantarcioglu, "Adversarial Machine Learning." Morgan and Claypool, 2018
- [36] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, Pierre-Antoine Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders", 2008
- [37] Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." arXiv preprint arXiv:1711.03938 (2017).
- [38] J.M. Alvarez, A. Lopez, R. Baldrich, Illuminant-invariant model-based road segmentation, in: 2008 IEEE Intelligent Vehicles Symposium, 2008, pp. 1175–1180, <http://dx.doi.org/10.1109/IVS.2008.4621283>.
- [39] Ming Liang*, Bin Yang*, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In CVPR, 2019
- [40] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In CVPR, 2018. 1, 2, 6, 8
- [41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In CVPR, 2017. 2
- [42] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In NIPS, 2017. 2
- [43] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In CVPR, 2017. 1, 2, 6, 8
- [44] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. In IROS, 2018. 1, 2, 6, 7, 8
- [45] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In ECCV, 2018. 1, 2, 3, 4, 6, 7, 8
- [46] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In CVPR, 2018. 2
- [47] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014b
- [48] Wiyatno, Rey & Xu, Anqi. (2018). Maximal Jacobian-based Saliency Map Attack.
- [49] Li, D., Chen, D., Goh, J., & Ng, S. (2018). Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series. 1–10. Retrieved from <http://arxiv.org/abs/1809.04758>
- [50] Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-Gan: Protecting classifiers against adversarial attacks using generative models. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, (3).
- [51] Shah, Shital, et al. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles." Field and service robotics. Springer, Cham, 2018.
- [52] AVL, AVL ISAC 6TM, 08.01.2020, <https://www.avl.com/vehicle-simulation>
- [53] AAI Automotive Artificial Intelligence 08.01.2020, <https://www.automotive-ai.com/>
- [54] G. Naik, B. Choudhury, J.M. Park. "IEEE 802.11 bd and 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications". IEEE Access, vol. 7, pp: 70169-70184. 2019.
- [55] N.O. Tippenhauer et al., "On the requirements for successful GPS spoofing attacks.", Proceedings of the 18th ACM conference on Computer and communications security. 2011.
- [56] D.P. Shepard et al., "Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks.", Radionavigation Laboratory Conference Proceedings. 2012.
- [57] S. Lefevre, J. Petit, R. Bajcsy, F. Kargl. "Impact of V2S privacy strategies on intersection collision avoidance systems". In Proceedings of IEEE Vehicular Networking Conference (VNC), pp: 71-78. December 2013.
- [58] R. Riebl. Vanetta framework, Opensource implementation of the ETSI C-ITS protocol stack. Technische Hochschule Ingolstadt. Available online: <https://github.com/riebl/vanetta>
- [59] ETSI. TS 102 941 V1.3.1 Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. 02-2019.

- [60] M. Feiri, J. Petit, F. Kargl. "Efficient and Secure Storage of Private Keys for Pseudonymous Vehicular Communication". In proceedings of the ACM workshop on Security privacy and dependability for cyber vehicles. November 2013.
- [61] M. Feiri, R. Pielage, J. Petit, N. Zannone, F. Kargl. "Pre-distribution of Certificates for Pseudonymous Broadcast Authentication in VANET". In proceedings of IEEE Vehicular Technology Conference (VTC Spring). July 2015.
- [62] C-ROADS. "Draft report on European security mechanism". Version 1.4. January 2019.
- [63] N. Souli, P. Kolios, and G. Ellinas, "Relative positioning of autonomous systems using signals of opportunity," in 2020 IEEE 91st Vehicular Technology Conference (VTC Spring). IEEE, 2020, p. to appear
- [64] Hyowon Kim, Sang Hyun Lee, and Sunwoo Kim. Cooperative Localization with Distributed ADMM over 5G-based VANETs. 2018 IEEE Wireless Communications and Networking Conference (WCNC): Special Session Workshops, pp: 612-616.
- [65] Rajnikant Sharma, Stephen Quebe, Randal W Beard, and Clark N Taylor. Bearing-only cooperative localization. *Journal of Intelligent & Robotic Systems*, 72(3-4), pp:429–440, 2013.
- [66] A. Howard, M. Mataric, and G. Sukhatme. Putting the 'i' in 'team': An ego-centric approach to cooperative localization. in *Proc. IEEE Int. Conf. Robotics Automation*, pp: 868–892, 2013.
- [67] H. Li and F. Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp: 33–44, 2013.
- [68] Olga Sorkine. Laplacian Mesh Processing. EUROGRAPHICS 2005.
- [69] Brosh, Eli & Friedmann, Matan & Kadar, Ilan & Lavy, Lev & Levi, Elad & Rippa, Shmuel & Lempert, Yair & Fernandez-Ruiz, Bruno & Herzig, Roei & Darrell, Trevor. (2019). Accurate Visual Localization for Automotive Applications.
- [70] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, ´ and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1704–1716, Sept. 2012
- [71] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale ´ image retrieval with compressed fisher vectors. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 3384–3391, 2010.
- [72] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and ´ C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '09*, pages 19:1– 19:8, New York, NY, USA, 2009. ACM
- [73] P. Richardson, D. Flynn, and A. Keane, "Optimal Charging of Electric Vehicles in Low-Voltage Distribution Systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 268–279, Feb. 2012, doi: 10.1109/TPWRS.2011.2158247.
- [74] G. B. Giannakis, V. Kekatos, N. Gatsis, S.-J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and Optimization for Power Grids: A Signal Processing Perspective," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 107–128, Sep. 2013, doi: 10.1109/MSP.2013.2245726.
- [75] Z. Fan, "A Distributed Demand Response Algorithm and Its Application to PHEV Charging in Smart Grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1280–1290, Sep. 2012, doi: 10.1109/TSG.2012.2185075.
- [76] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized Charging Control of Large Populations of Plug-in Electric Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 67–78, Jan. 2013, doi: 10.1109/TCST.2011.2174059.
- [77] Y. Zhang, N. Gatsis, and G. B. Giannakis, "Robust Energy Management for Microgrids With High-Penetration Renewables," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 944–953, Oct. 2013, doi: 10.1109/TSTE.2013.2255135.
- [78] D. Papadaskalopoulos, G. Strbac, P. Mancarella, M. Aunedi, and V. Stanojevic, "Decentralized Participation of Flexible Demand in Electricity Markets—Part II: Application With Electric Vehicles and Heat Pump Systems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3667–3674, Nov. 2013, doi: 10.1109/TPWRS.2013.2245687.
- [79] R. Li, Q. Wu, and S. S. Oren, "Distribution Locational Marginal Pricing for Optimal Electric Vehicle Charging Management," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 203–211, Jan. 2014, doi: 10.1109/TPWRS.2013.2278952.
- [80] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, May 2013, doi: 10.1109/TPWRS.2012.2210288.
- [81] "(12) (PDF) Ant-Based Swarm Algorithm for Charging Coordination of Electric Vehicles," ResearchGate. [Online]. Available: https://www.researchgate.net/publication/258391714_Ant-Based_Swarm_Algorithm_for_Charging_Coordination_of_Electric_Vehicles. [Accessed: 04-Feb-2020].
- [82] E. L. Karfopoulos and N. D. Hatziaargyriou, "A Multi-Agent System for Controlled Charging of a Large Population of Electric Vehicles," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1196–1204, May 2013, doi: 10.1109/TPWRS.2012.2211624.
- [83] J. Rivera, P. Wolfrum, S. Hirche, C. Goebel, and H.-A. Jacobsen, "Alternating Direction Method of Multipliers for decentralized electric vehicle charging control," in 52nd IEEE Conference on Decision and Control, 2013, pp. 6960–6965, doi: 10.1109/CDC.2013.6760992.

- [84] W.-J. Ma, V. Gupta, and U. Topcu, "On distributed charging control of electric vehicles with power network capacity constraints," in 2014 American Control Conference, 2014, pp. 4306–4311, doi: 10.1109/ACC.2014.6859139.
- [85] O. Ardakanian, C. Rosenberg, and S. Keshav, "Distributed control of electric vehicle charging," in Proceedings of the fourth international conference on Future energy systems, Berkeley, California, USA, 2013, pp. 101–112, doi: 10.1145/2487166.2487178.
- [86] Q. Li, T. Cui, R. Negi, F. Franchetti, and M. D. Ilić, "On-line Decentralized Charging of Plug-In Electric Vehicles in Power Systems," ArXiv, vol. abs/1106.5063, 2011.
- [87] N. Chen, C. W. Tan, and T. Q. S. Quek, "Electric Vehicle Charging in Smart Grid: Optimality and Valley-Filling Algorithms," IEEE Journal of Selected Topics in Signal Processing, vol. 8, pp. 1073–1083, 2014, doi: 10.1109/JSTSP.2014.2334275.
- [88] L. Zhang, V. Kekatos, and G. B. Giannakis, "A generalized Frank-wolfe approach to decentralized electric vehicle charging," in 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 1105–1111, doi: 10.1109/CDC.2016.7798415.
- [89] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable Electric Vehicle Charging Protocols," IEEE Transactions on Power Systems, vol. 32, no. 2, pp. 1451–1462, Mar. 2017, doi: 10.1109/TPWRS.2016.2582903.
- [90] Campos, G.O., Zimek, A., Sander, J. et al. Data Min Knowl Disc (2016) 30: 891. <https://doi.org/10.1007/s10618-015-0444-8>
- [91] Markus Goldstein, Seiichi Uchida, A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data, PloS one (2016), <https://doi.org/10.1371/journal.pone.0152173>
- [92] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," IEEE Transactions on Power Systems, vol. 28, no. 2, pp. 940–951, May 2013, doi: 10.1109/TPWRS.2012.2210288.
- [93] P. Hu, "Summary of Travel Trends 2001 National Household Travel Survey," ORNL/TM-2004/297, 885762, Jan. 2005.
- [94] "SCE Load Profiles," SCE.com. [Online]. Available: <http://origin.sce.com/regulatory/load-profiles>. [Accessed: 05-Feb-2020].
- [95] K. Wang et al., "A Survey on Energy Internet: Architecture, Approach, and Emerging Technologies," in IEEE Systems Journal, vol. 12, no. 3, pp. 2403–2416, Sept. 2018.
- [96] Netherlands Enterprise Agency: Electric vehicle charging: Definitions and explanation, January 2019, https://www.rvo.nl/sites/default/files/2019/01/Electric%20Vehicle%20Charging%20-%20Definitions%20and%20Explanation%20-%20january%202019_0.pdf
- [97] IEA (2019), "Global EV Outlook 2019", IEA, Paris <https://www.iea.org/reports/global-ev-outlook-2019>
- [98] IEA, "Electric car deployment in selected countries, 2013-2018", IEA, Paris <https://www.iea.org/data-and-statistics/charts/electric-car-deployment-in-selected-countries-2013-2018>
- [99] Artificial Intelligence - An opportunity for the EU cyber-crisis management, Workshop, ENISA, June 2019, <https://www.enisa.europa.eu/events/artificial-intelligence-an-opportunity-for-the-eu-cyber-crisis-management>
- [100] Guidelines on FAIR Data Management in Horizon 2020. European Commission. [Online] Available: https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf

Annex 1

Model.CONNECT overview

Model.CONNECT™ is AVL's open model integration and co-simulation platform, connecting virtual and real components into one functional prototype.

Vehicle development is a team effort. The key to mastering new development tasks is understanding the whole system early on in the process and using advanced simulation techniques in both the development and the testing phase.

- How to efficiently manage development tasks such as RDE, Thermal Management, electrification or ADAS/AD?
- How to link component models into one virtual prototype?
- How to integrate hardware components with simulation models?
- How to synchronize distributed development teams, while ensuring model sharing and IP protection?

Model.CONNECT™ improves development efficiency by interlinking simulation models from different tools into one consistent virtual prototype, featuring:

- Efficient integration of existing models (Simulink, AMESim, VTD, GT, IPG, MSC ADAMS, AVL...), industry standards (FMI, XCP...) and user code (Python, C/C++, Java...)
- Exchangeability of models from different domains, across department and application boundaries
- Accurate and fast results with unique coupling algorithms
- Connecting co-simulation with real-time systems by using patented RT-synchronization technology

As a part of AVL's Open and Integrated Development Platform (IODP), Model.CONNECT™ empowers the implementation of model-based development, closing the gap between virtual and real worlds, with the following benefits:

- Ready-to-use platform for building a digital twin in a heterogeneous model landscape
- Better understanding of component interactions even during early development phases
- Sustainable and secure collaboration between different departments and development partners
- Shortening development iteration loops and improving testing efficiency by extensive usage of simulation methods in the testing environment

ADAS toolchain in Model.CONNECT

This document describes the ADAS toolchain connected in AVL Model.CONNECT™.

The Model.CONNECT project consist of following tools:

1. AVL VSM
2. Vires VTD
3. AVL DRIVE
4. MATLAB Simulink FMU
5. Other components with smaller part

The layout of the ADAS toolchain in Model.CONNECT is shown on Figure 60

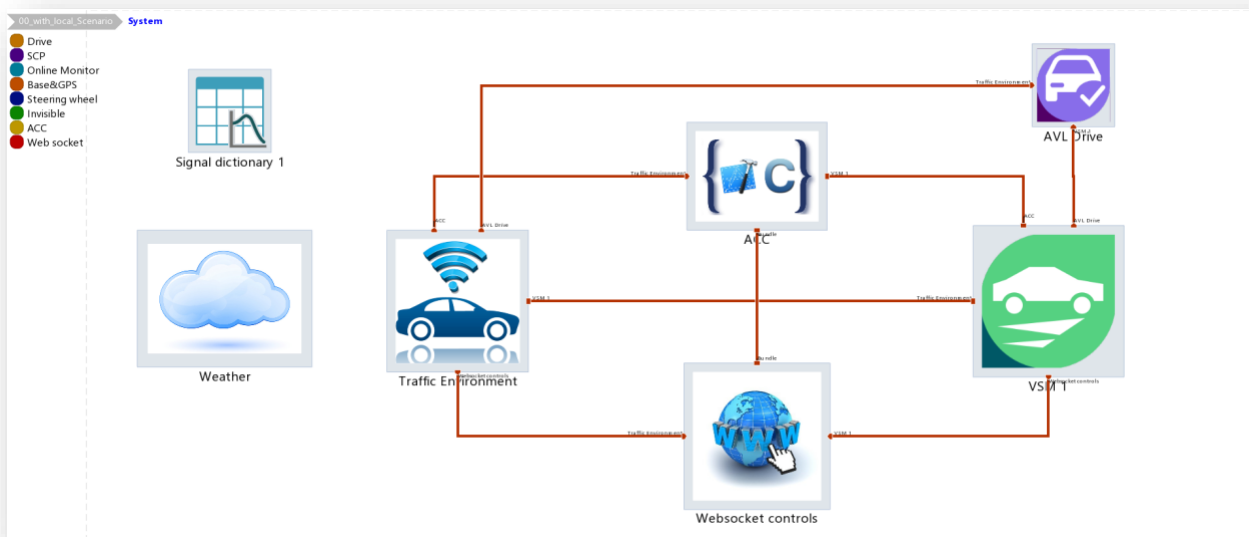


Figure 60: Model.CONNECT ADAS toolchain

AVL VSM

VSM is used to simulate vehicle dynamics. In this setup, it is useful to have a vehicle simulated in a single environment. For this purpose, a VSM installation example vehicle is used, mid-sized class vehicle. During the VSM integration in Model.CONNECT, the user has a possibility to use or exclude certain parts of the VSM vehicle.

Figure 61 shows VSM components which are possible to include or exclude from VSM simulation.

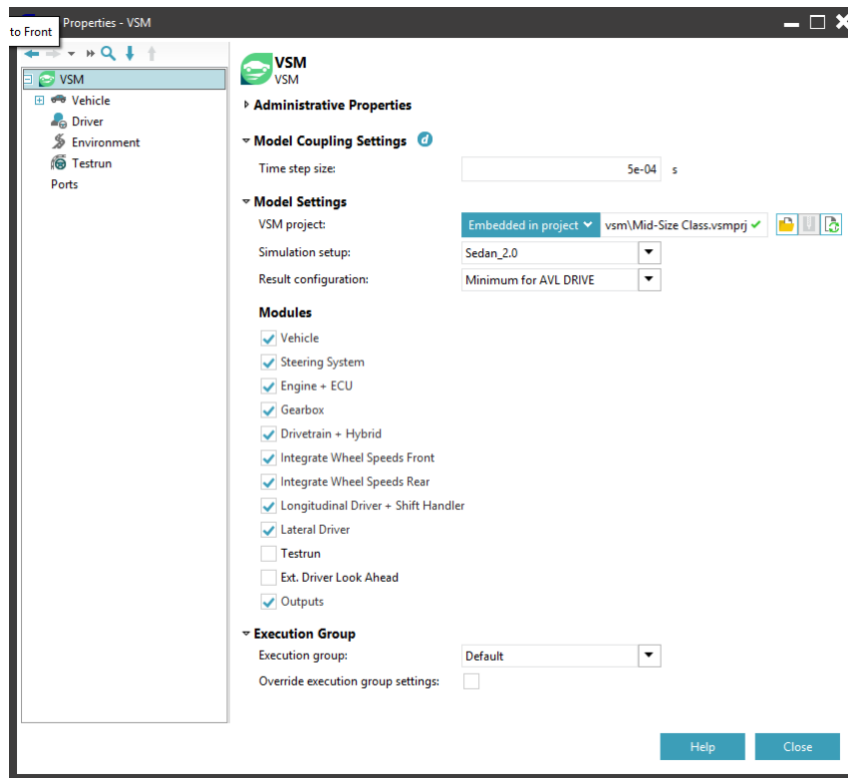


Figure 61: VSM configuration in Model.CONNECT

In this toolchain, *Testrun* and *Ext. Driver Look Ahead* are excluded since a vehicle environment is defined in VTD environment.

It is worth noting that *Longitudinal Driver* is included in VSM simulation, although other Model.CONNECT components are providing signals like load and brake signal. Adjusting the certain flags during the simulation runtime, it is possible to bypass some of the VSM internal functions and implement functions from Model.CONNECT.

Vires VTD

VTD (Virtual Test Drive) is a simulation software for sensors and environment. VTD runs only on a Linux machine, so for co-simulation with VTD, a dedicated Linux machine or a virtual Linux machine is required.

VTD provides environment and sensor readings. Environments consist of a road and its elements:

- road marks
- signs
- vehicles
- pedestrians, etc.

This model uses two sensors: front sensor and GPS sensor.

Front sensor is capable to detect maximum five objects and it will provide following information of an object:

- type
- distance
- speed
- position

- lane ID

GPS sensor provides a location of a vehicle on the road. In order to use GPS sensor, an OpenStreetMap is imported into the VTD Road Scenario.

Front and GPS sensor information are sent to the Model.CONNECT. Front sensor readings are used mainly for the ACC functionality, while GPS sensor data is used for the web interface and positioning the car on the OpenStreetMap.

A nice feature of the VTD is a video of the running simulation, as shown on the Figure 62 and Figure 63.

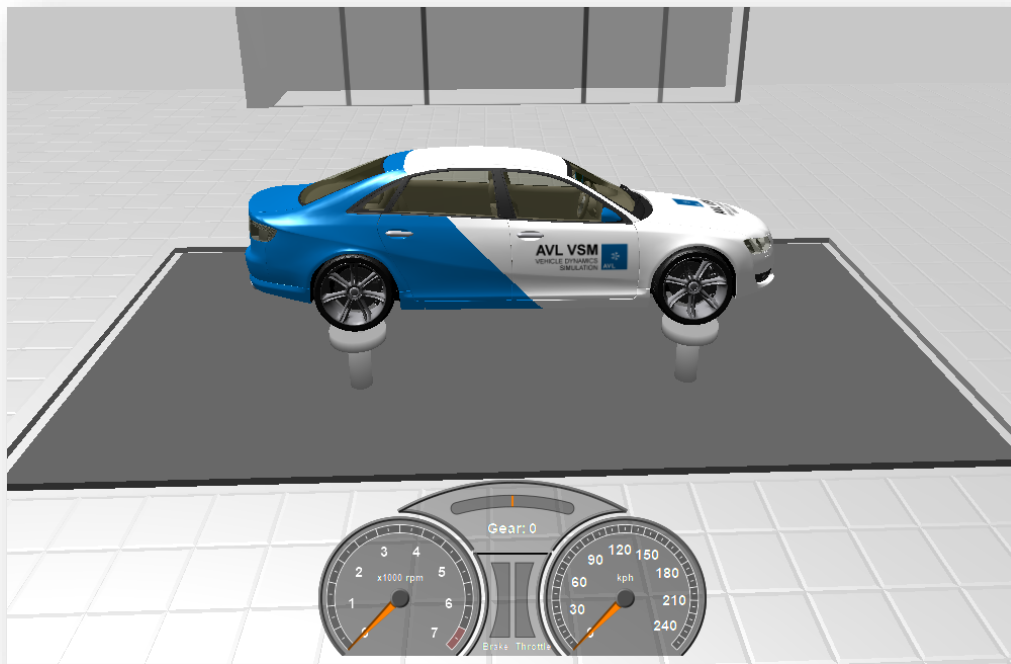


Figure 62: VSM vehicle simulation in testbed mode

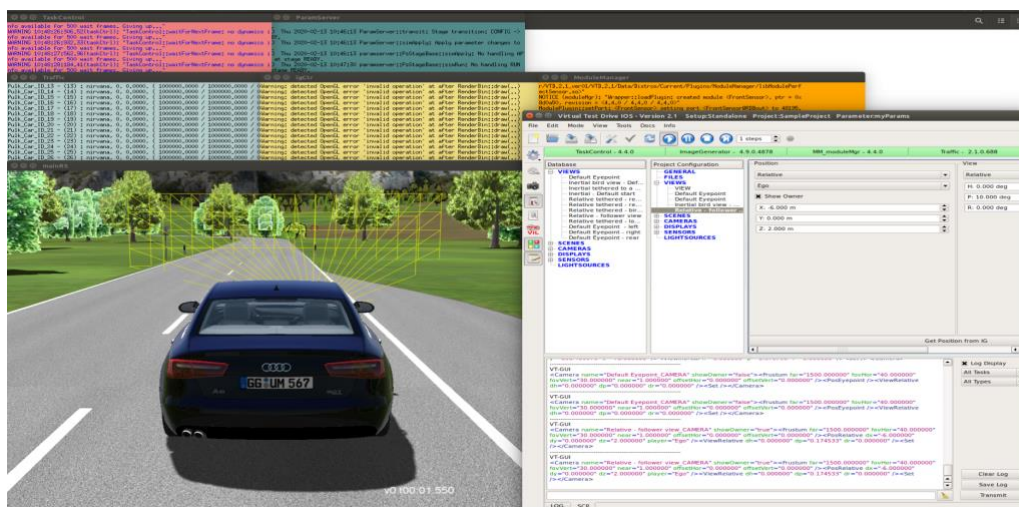


Figure 63: VTD simulation video

AVL DRIVE

Drive is a software for driveability score (see Figure 64). It scores the vehicle movements like acceleration, deceleration, gear shift, constant speed, vehicle stop, constant speed.

The score has a range from 0 to 10, where 10 means the comfort and safety are achieved in the best possible way.

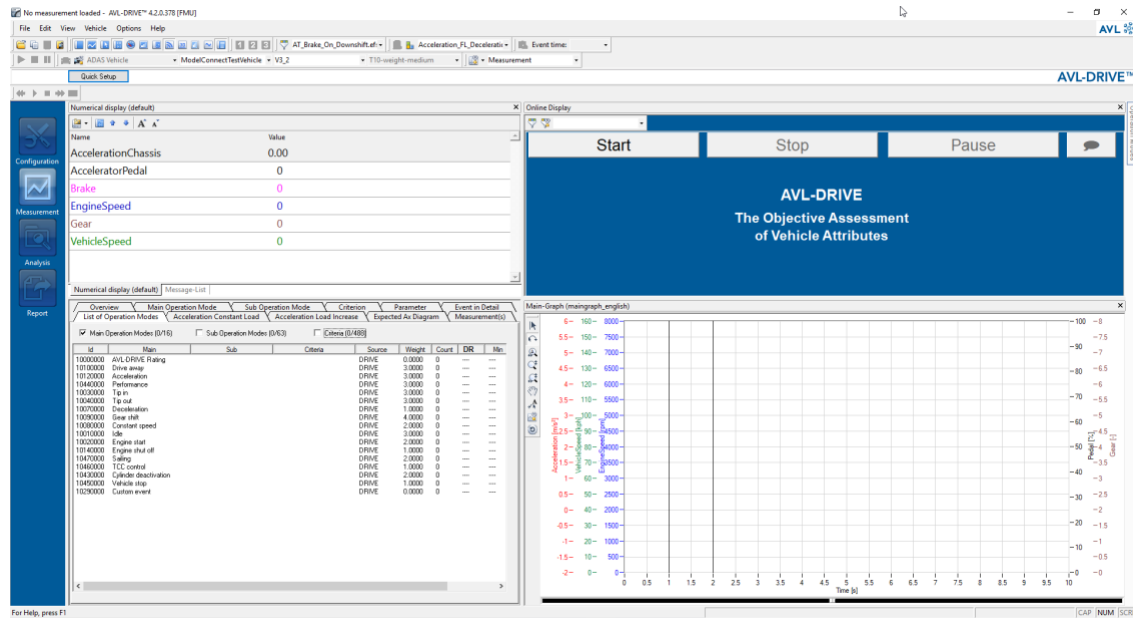


Figure 64: AVL DRIVE

MATLAB Simulink FMU for ACC

A Simulink ACC controller model is compiled as an FMU using the AVL fmi.Lab software.

The benefit of using Simulink FMU instead of the native Simulink models are:

- faster run of the model
- a MATLAB license is not required for the run

The downside is that FMU is a black box and the user can't see into the model. On the other hand, it's possible to expose parameters of the Simulink model which can be used and modified from Model.CONNECT.

Web interface

This model has a web user interface. A FMU is used for communication with the web page.

The UI is shown on the Figure 65.

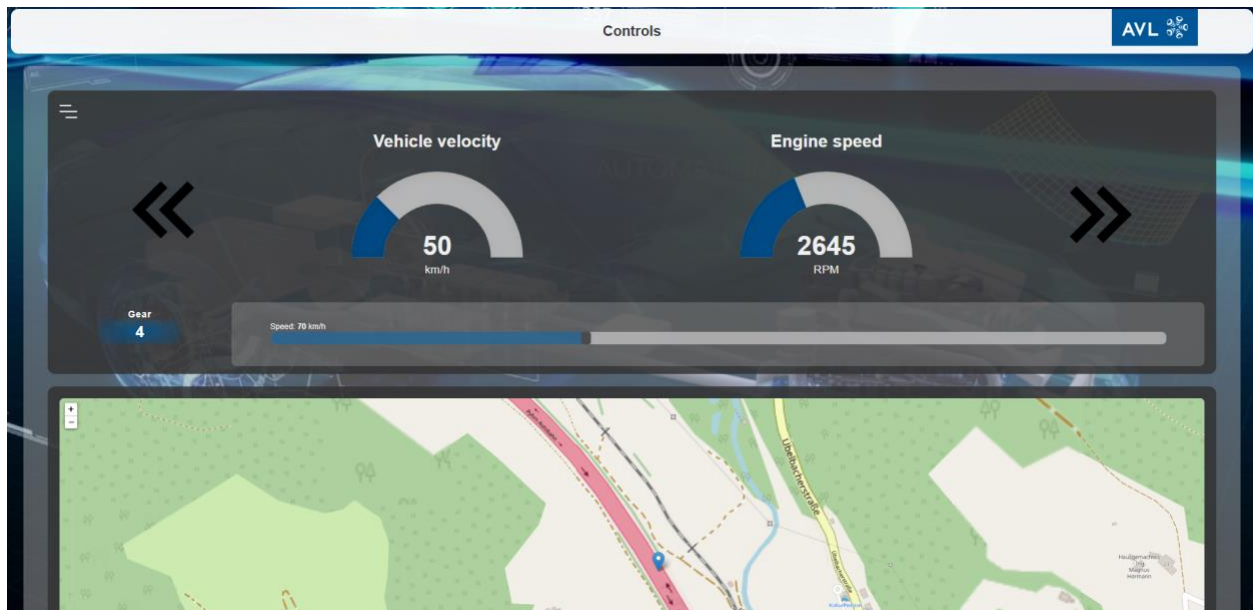


Figure 65: Web UI

The user can enable or disable ACC functionality, set ACC vehicle speed, change lanes, observe vehicle velocity and engine speed, and vehicle position.

Additionally, the user has a possibility to override all driver signals to drive the vehicle with the tablet/mobile movement. E.g., leaning tablet forward will increase load signal, leaning backward will activate brakes, and leaning left and right will rotate the vehicle steering wheel.

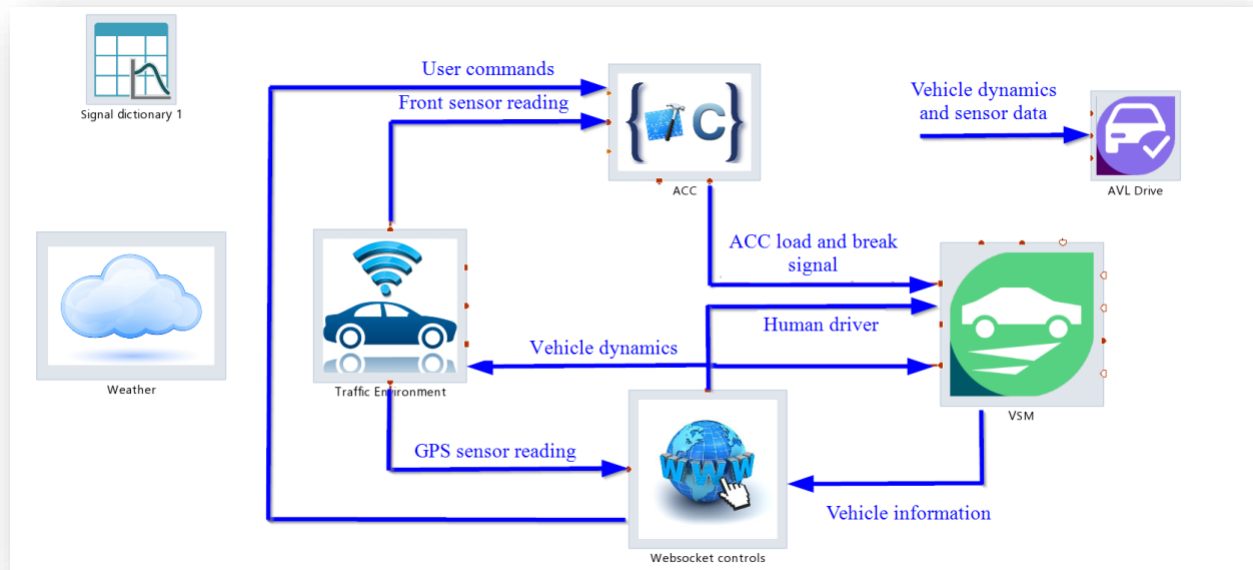
Data flow

Figure 66 shows the principle data flow. VSM provides the vehicle dynamics to the VTD, so VTD can place car on the exact position on the road. Also, VSM send the information of the vehicle state to the AVL DRIVE, ACC controller, WebSocket controls.

ACC controller receives sensor data from VTD (object around the vehicle), desired velocity from web user interface, and actual vehicle velocity from VSM. Based on this information, ACC will calculate the load and brake signal and send it to the VSM vehicle pedals.

AVL DRIVE is only on the receiving side, in order to calculate driveability score.

Additionally, a *Weather* component can be found on the system topology. It serves to send SCP commands to the VTD environment and change the weather.

**Figure 66: Information exchange flow**