CONNECT
CCAM TRUST & RESILIENCE

D6.2: Integrated Framework (Final Release), Use Case Evaluation and Project Impact Assessment

Project number:	101069688
Project acronym:	**CONNECT**
Project title:	Continuous and Efficient Cooperative Trust Management for Resilient CCAM
Project Start Date:	1st September, 2022
Duration:	36 months
Programme:	HORIZON-CL5-2021-D6-01-04
Deliverable Type:	OTHER
Reference Number:	D6-01-04 / D6.2 / 1.02 September 2, 2025
Workpackage:	WP 6
Due Date:	M36 - August 31, 2025
Actual Submission Date:	September 2, 2025
Responsible Organisation:	TRIALOG
Editor:	Antonio Kung
Dissemination Level:	PU - Public
Revision:	1.02 September 2, 2025
Abstract:	Deliverable 6.2 details the evaluation results of CONNECT's Trust Assessment Architecture when integrated/deployed in the context of the envisioned use cases unlocking the assessment of such advanced trust mechanisms as part of the end-to-end operational profile of CCAM functionalities. Building on top of the benchmarking results (per standalone component) that were documented in the previous version of this deliverable, D6.2 validates the framework's tangible impact through three core CCAM use cases: Intersection Movement Assistance (IMA), Cooperative Adaptive Cruise Control (C-ACC), and Slow Moving Traffic Detection (SMTD). Each was tested using the most appropriate methodology - from *Large-Scale Simulation* scenarios to *Hardware-in-the-Loop* and *Real-World Living Labs* environment following the development pipeline adopted in V2X security and safety engineering: from conception activities and prototyping to real-world evaluation. Our findings confirm the framework's benefits in allowing trust-aware decision making achieving all identified KPIs positioning it a stepping stone for future research towards converging security and safety which can further drive its industru-wide adoption.
Keywords:	Integrated Framework, Interfaces & APIs, Use Cases, Evaluation Results, Trust Assessment Framework, CCAM

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Editor
Antonio Kung(TRIALOG)

Contributors (ordered according to beneficiary numbers)

Anna Angelogianni, Nikolaos Fotos, Thanassis Giannetsos, Stefanos Vasileiadis (UBITECH)
Ana Petrovska, Ioannis Krontiris, Theo Dimitrakos (HUAWEI)
Vangelis Kosmatos, Pavlos Basaras, Panagiotis Pantazopoulos (ICCS)
Benjamin Erb, Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
Antonio Kung, Guillaume Mockly (TRIALOG)
Anderson Ramon Ferraz de Lucena, Alexander Kiening (DENSO)
Matthias Schunter, Sergej Schumilo (INTEL)
Konstantinos Latanis (SUITE5)
Ilias Aliferis (UNISYSTEMS)
Adam Henschke, Sadjad Soltanzadeh (UTWENTE)
Peter Schmitting (FSCOM)
Luisa Andreone, Andrea Milan (CRF)
Marco Rapelli, Claudio Casetti, Guido Marchetto, Carla Fabiana Chiasserini (POLITO)
Francesca Bassi, Ines Ben Jemma (IRTSX)
Christopher Newton (SURREY)

Disclaimer

Executive Summary

This deliverable marks the documentation of the *CONNECT* project's validation and evaluation activities, presenting the final, integrated deployment of the overarching Trust and Security Framework alongside its detailed benchmarking in the context of the envisiond use cases. The primary objective of this last evaluation phase was to move beyond the standalone component benchmarking and assess the tangible impact of our novel trust-enabling mechanisms in realistic, end-to-end Cooperative, Connected and Automated Mobility (CCAM) scenarios. By rigorously testing the framework against three diverse use cases—Intersection Movement Assistance (IMA), Cooperative Adaptive Cruise Control (C-ACC), and Slow Moving Traffic Detection (SMTD)—we have successfully demonstrated its value in enhancing the security, resilience, and overall trustworthiness of the CCAM ecosystem.

The final evaluation phase validated the more advanced and complex features of the *CONNECT* architecture, including the Federated Trust Assessment Framework (TAF) and MEC-assisted trust computations. Our validation strategy was carefully tailored to each use case: the IMA use case relied on large-scale simulation to safely test complex adversarial attacks; the C-ACC use case leveraged a Hardware-in-the-Loop (HWIL) setup to measure real-time performance; and the SMTD use case was deployed in a real-world living lab to assess its effectiveness under realistic conditions.

The results provide compelling evidence of the framework's success. Most notably, the framework proved its ability to mitigate perception-based attacks in the IMA use case by successfully identifying and excluding untrustworthy data, enabled by novel mechanisms like the Temporal-ATL for interpreting time-dependent evidence. Beyond security, the framework demonstrated high performance and efficiency, with the pull-based TAF responding to on-demand trust requests in under 10ms in the C-ACC use case. This robustness is underpinned by the federated architecture, a cornerstone of the *CONNECT* design that leverages the MEC to aggregate trust evidence, leading to earlier threat detection. The project also yielded critical lessons for future development, highlighting that the system's trustworthiness is fundamentally dependent on the quality of its evidence sources and providing solutions for inherent architectural challenges like the 'double counting' of evidence. Beyond these technical validations, a stakeholder survey confirmed the project's value-pluralist approach, demonstrating that for CCAM to be considered trustworthy, it must address a wide range of values beyond technical safety, including accountability, transparency, and fairness.

In conclusion, the *CONNECT* project has successfully designed, implemented, and validated a comprehensive, multi-layered framework that enhances security and dynamically assesses trust in cooperative mobility. By delivering on its objectives and yielding critical insights for future V2X systems, *CONNECT* has made a tangible contribution towards building a safer, more resilient, and ultimately more trustworthy CCAM ecosystem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scope and Purpose

This deliverable marks the culmination of the *CONNECT* project's validation and evaluation activities. It presents the final, integrated deployment of the *CONNECT* Trust and Security Framework, rigorously tested against three key use cases: (i) Intersection Movement Assistance (IMA) with integrated Misbehavior Detection (MBD), (ii) Cooperative Adaptive Cruise Control (C-ACC), and (iii) Slow Moving Traffic Detection (SMTD). Building upon the foundational work of previous deliverables—where core components were designed, implemented, and individually benchmarked—this document shifts the focus from standalone analysis to a holistic, end-to-end system evaluation. The primary objective is to assess the tangible impact of novel trust-enabling mechanisms within operationally realistic scenarios. These trust assessment enablers are introduced across different layers and aim at ensuring system's resilience and robustness in diverse *Connected, Co-operative and Automated Mobility (CCAM)* contexts. The selected use cases represent a diverse set of V2X interactions and mobility patterns, allowing us to validate the framework's efficacy and generalizability.

The first iteration of the framework, detailed in D6.1 [Con24b], established the architectural blueprint and technical groundwork. To this end, D6.1 [Con24b] provided an analysis of the components and their interfaces, covering the entire operational pipeline, including both synchronous (pull-based) and asynchronous (push-based) processes. The framework extends from the far-edge (vehicle) to the MEC, addressing the various and diverse needs of the CCAM ecosystem. The use case demonstrators were formerly introduced along with their initial experimental setup, implementation reports, and Key Performance Indicators (KPIs). The focus of the first round of evaluation activities was centred around the standalone Trust Assessment Framework (TAF), including in-vehicle and vehicle to vehicle (V2V) operations. In addition D6.1 also provided an initial benchmarking of key *CONNECT* components such as the *Trusted Computing Base (TCB)*, the TLEE (part of the *Trust Assessment Framework (TAF)*), and the *Distributed Ledger Technology (DLT)*. These standalone experimentations of the different components were further elaborated and extended in D3.3 [Con25c], D4.3 [Con25d] and D5.3 [Con25b].

This second and final round of experimentation significantly expands that scope. The evaluation now encompasses the more complex Federated TAF operations and MEC-assisted trust computations, which are critical for scalability and real-time performance in dense environments. A key focus of this deliverable is to evaluate the TAF's performance under a spectrum of conditions, including both nominal and adversarial scenarios designed to challenge its security and

detection capabilities. This final evaluation phase introduced significant advancements across each of the core use cases. For the Intersection Movement Assistance (IMA) use case, the experimental scope was expanded to validate performance in both small-scale and large-scale deployment scenarios. In parallel, the Cooperative Adaptive Cruise Control (C-ACC) evaluation was enhanced through the introduction of the pull-based (on-demand) trust assessment model, complementing the initial subscription-based approach. Finally, the Slow Moving Traffic Detection (SMTD) use case underwent rigorous and detailed benchmarking within a realistic living lab environment to assess its practical, real-world performance. To this end, in contrast to the other use cases, the IMA evaluation relies on simulation-based experimentation. This methodology was specifically chosen to address the unique challenges of IMA, allowing us to safely model large-scale, complex interactions and adversarial scenarios that would be impractical to test otherwise. This approach differs from the Hardware-in-the-Loop (HWIL) setup used for C-ACC, which focused on real-time performance, and the real-world Living Lab environment for SMTD, which targeted detection accuracy under realistic conditions.

Table 1.1 provides a detailed matrix mapping the specific *CONNECT* components leveraged in each use case, setting the stage for the technical deep-dive in the subsequent sections. To clarify the scope of this integrated evaluation, the core in-vehicle TAF (Standalone TAF) and its virtual counterpart (DT-TAF) are systematically evaluated across all three use cases. This comprehensive testing allows for a robust assessment of their performance across diverse CCAM scenarios, validating their benefit in real-world situations and building upon the foundational component evaluations documented in D3.3 [Con25c]. The Federated TAF, which enables collaborative trust, is specifically validated within the IMA use case (Scenario #1), as its complex intersection dynamics provide the ideal environment to test the scalability and coordination benefits of a federated model. Similarly, the *CONNECT* Risk Assessment Framework is tested within the C-ACC use case to evaluate its ability to calculate dynamic Required Trust Levels (RTLs) and adjust trust requirements in real-time based on the safety-critical nature of platoon manoeuvrers.

Central to all use cases is the *CONNECT* Trusted Component Base (TCB), which establishes the hardware-based Root of Trust (RoT) for the integrity of vehicle systems. For this system-level evaluation, we leverage simulated TCB attestation reports. This methodological choice is based on the comprehensive performance and cryptographic benchmarks already established in D4.3 [Con25d]. By abstracting the low-level cryptographic operations, we can focus squarely on evaluating the framework's end-to-end performance and its response to integrity data, without the computational overhead of re-calculating these operations. Additionally, MEC-assisted task offloading is investigated in the SMTD scenario, while the migration of a CCAM service from one vehicle computer to another is examined within the C-ACC use case. The live migration of the CCAM operational state between vehicle computers has been documented in D4.3 [Con25d]. Finally, the evaluation of an ETSI-aligned CAM/CPM encoder for the exchange of V2X messages—both vehicle-to-vehicle and vehicle-to-MEC—is presented within the SMTD use case.

Table 1.1: Use Case to *CONNECT* Components mapping

Component	IMA	CACC	SMTD	Standalone Evaluation
Standalone TAF	✓	✓	✓	✓ in D3.3
Federated TAF	✓			
DT-TAF	✓ (implicitly)	✓ (implicitly)	✓ (implicitly)	✓ in D3.3
CONNECT Risk Assessment (RA)		✓		✓ in D3.3

CONNECT TCB		✓ simulated TCB	✓Gramine-direct	✓ in D6.1 (initial evaluation and comparison with TPM) and D4.3 (final evaluation and comparison with other TEE implmentations)
CONNECT TCH	✓ simulated TCs for more agile experimentation	✓simulated TCs for more agile experimentation	✓ simulated TCs for more agile experimentation	
MEC-assisted Task offloading			✓	✓ D5.2
CAM/CPM encoder			✓	
CCAM state & operation Migration		✓ operation migration		✓ live state migration evaluation in D4.3 [Con25d]

The empirical results presented herein offer compelling evidence that the *CONNECT* framework delivers on its value propositions. The findings validate its effectiveness in enhancing the security, reliability, and overall trustworthiness of the CCAM ecosystem, demonstrating a tangible contribution to the future of secure and cooperative mobility. In addition to the technical evaluations, an ethical assessment was carried out to examine how trust dimensions have been integrated into the security engineering of CCAM systems. This assessment, detailed in Chapter 6 on Technology Assessment Modeling (TAM), utilized two complementary questionnaires. The first targeted CONNECT consortium stakeholders, aiming to investigate hypotheses regarding the influence of trust on the adoption and acceptance of CCAM services as envisioned in the project. The second questionnaire focused on external CCAM stakeholders, providing a basis for validating these hypotheses within a broader industrial and societal context. Collectively, these instruments facilitated a structured evaluation of the role of trust in both informing technical design and shaping potential adoption pathways for CCAM services.

1.2 Relation to other deliverables

The present deliverable constitutes the final output of WP6 and represents one of the concluding deliverables of the entire *CONNECT* project. It builds upon inputs from WP2, specifically D2.1 [Con23c] and D2.2 [CON23a], which define the architecture, the use cases, the functional requirements, and KPIs. Additional contributions are incorporated from all technical WPs, including WP3, WP4, and WP5, where the implementation of the framework components was completed and evaluated as part of Release B. The present deliverable directly builds on the work completed in these previous deliverables, leveraging their results to perform a consolidated and comprehensive evaluation of the framework.

The final evaluation is conducted across the three use cases, employing three complementary approaches: i) simulation-based evaluation, ii) Human-in-the-Loop (HiTL) experiments, and iii) living lab assessments. Simulation-based evaluation enables the testing of system components under controlled and reproducible conditions, providing early insights into performance, reliability, and scalability. HiTL experiments engage human participants interacting with the system to

evaluate usability, decision-making support, and trust-related behaviour in realistic scenarios. Finally, living lab assessments extend the evaluation to real-world environments, validating system performance, robustness, and interoperability under operational conditions.

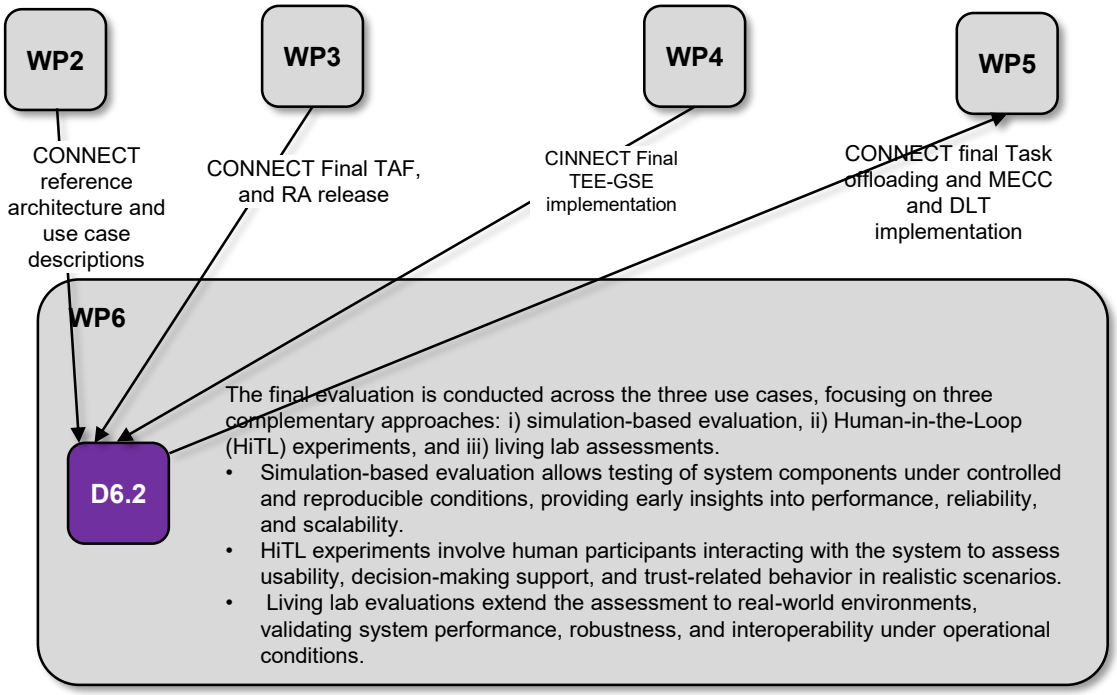


Figure 1.1: Relation to other Deliverables

1.3 Deliverable Structure

The remainder of this deliverable is structured in the following chapters: Chapter 2 first details the final integrated *CONNECT* architecture, presenting a comprehensive analysis of its core components, unit test results, and the Key Performance Indicators (KPIs) established for validation. The subsequent three chapters form the empirical core of this work: Chapter 3 presents the evaluation of the Intersection Movement Assistance (IMA) demonstrator, followed by Chapter 4 and Chapter 5, which respectively detail the results for the Cooperative Adaptive Cruise Control (C-ACC) and Slow Moving Traffic Detection (SMTD) use cases. Shifting from technical validation to broader implications, Chapter 6 then addresses the crucial ethical considerations and evaluation results pertinent to the framework's deployment. This is followed by Chapter 7, which offers a holistic discussion on the applicability of the *CONNECT* framework, synthesizing key findings and outlining the lessons learned throughout the project. Finally, Chapter 8 summarizes the main contributions of this work and concludes the deliverable.

Chapter 2

CONNECT Final Integrated Framework

2.1 Summary of Release A

As established in D6.1 [Con24b], the *CONNECT* framework is founded upon a three-layered architecture, with components operating across the vehicle (far-edge), MEC (edge), and cloud. The cornerstone of this architecture is the Trust Assessment Framework (TAF), a component designed to quantify the trustworthiness of *Connected, Co-operative and Automated Mobility (CCAM)* entities and the data they exchange. The TAF supports both node-centric and data-centric trust evaluations for all critical interactions, including intra-vehicle, V2V, and V2I. Its operation is defined by Trust Model Templates (TMTs), which are created at design time to model trust relationships, and their corresponding run-time Trust Model Instances (TMIs), which are dynamically updated with new evidence. To accommodate the dynamic CCAM environment, the TAF is designed to operate in three distinct modes: i) the Standalone TAF, ii) the Federated TAF and iii) the TAF-Digital Twin (DT). Elaborated definitions on these are available in D3.2 [Con24c] and D6.1 [Con24b].

The initial integrated version of the framework, designated Release A, concentrated on validating foundational capabilities, with a primary focus on the in-vehicle environment. In this release, the implementation focused exclusively on the Standalone TAF operating within the vehicle, with the Federated and Digital-Twin modes designated for future work. This TAF operated using predefined Trust Model Templates (TMTs) with a static Required Trust Level (RTL). From a security standpoint, the TAF was instantiated outside the secure boundaries of a Gramine SGX enclave and the underlying Trusted Computing Base (TCB) was benchmarked independently. Furthermore, the framework operated under the key assumption that the secure onboarding of all vehicle components had already been successfully completed. Finally, external communication was limited, as the shared Trustworthiness Claim (TC) only contained reports from the attestation component (AIV). This restricted MEC-layer activities to integration tasks, demonstrating message reception from vehicles without enabling comprehensive trust assessments.

More specifically, the framework explored the following definition in terms of flows:

1. **Kinematic Data Extraction:** The *Facility Layer (FL)* collects kinematic data on CCAM vehicle observations from both the *Zonal controller (ZC)* and *electronic control unit (ECU)*, including raw metrics like vehicle position, speed, and direction. This data is used to construct the vehicle's Local Perception (LP). The *Mis-behaviour Detector (MBD)*-proxy generates reports based on simulated inputs from the SUMO traffic simulation framework. The *MBD* creates a misbehavior report and forwards it to the *Trust Assessment Framework*

(TAF) for inclusion in its assessments. The Trust Model Templates define the elements for trust evaluations for specific applications. The attestation process is triggered by the *MBD*, through the *Trustworthiness Claims Handler (TCH)*, each time new kinematic data becomes available.

2. **Evidence Collection**: The TAF gathers trustworthiness evidence. For *CONNECT* the main trust sources are the *Attestation and Integrity Verification (AIV)* and the *MBD*. Evidence may be requested in two ways:

   Pull-based (Synchronous): The TAF requests evidence from sources like the AIV component on-demand.

   Push-based (Asynchronous): The TAF subscribes to evidence sources to receive automatic updates when new information is available.

3. **Attestation & Data Handling**: When new kinematic data (currently simulated) arrives, the Misbehavior Detection (MBD) proxy triggers the Trustworthiness Clearing House (TCH), which in turn requests an attestation report from the AIV. The AIV collects integrity measurements from trusted components (e.g., the in-vehicle computer's TCB) and sends a signed attestation report to the TAF for evaluation and to the TCH for potential external sharing.

4. **External Reporting**: The TCH embeds information from the AIV, MBD and TAF into the Trustworthiness Claim (TC), which can be sent to nearby vehicles or the MEC to communicate the vehicle's trustworthiness status. The receiving entities in Release A would only process the message for integration purposes, with full evaluation planned for Release B.

5. **V2X information exchange - Receival of Trustworthiness Claim from an ego vehicle**: The V2X information exchange process involves a one-way communication from a single entity, the ego vehicle, to an undefined group of receivers, including the MEC infrastructure. The process begins with the reception of a TCH message from the vehicle-based TCH to the MEC-based TCH, which includes the trustworthiness claim from the vehicle(s).

2.2 Updates in Release B and Final Integrated Framework

The progress from the initial release of the *CONNECT* framework to its final, integrated version represents a significant evolution in both capability and maturity. Release A established the foundational groundwork, focusing on the design, implementation, and standalone evaluation of core components. The final release, detailed in this section, builds upon this foundation to deliver a holistic, end-to-end system engineered for the complexities of real-world CCAM scenarios. To provide a clear and detailed overview of these advancements, Table 2.1 summarizes the key updates across the framework's primary components. The updates highlight a shift from component-level benchmarking to system-wide integration and the introduction of advanced functionalities. Key enhancements include the finalization of the Federated TAF for collaborative trust, the deep integration of the DLT for decentralized policy management, and the hardening of the TCB with advanced cryptographic mechanisms.

Table 2.1: CONNECT Framework Updates between the different Releases

CONNECT Component	*CONNECT* Release A	*CONNECT* Final Release
TAF	Focus on the standalone TAF evaluation	**Finalisation and Integration of Federated TAF in the overarching *CONNECT* framework**, leveraging NTM messages for the exchange of trust-related information. **Enrichment and enhancement of standalone TAF according to the use cases**. Management of cached data. Insertion of the tag message based on which the TAF reacted for a new ATL, associating the ATL with detectors. Enrichment of Trust Models for covering complex scenarios including IMA as part of the federated TAF and MEC-assisted task offloading. **Finalisation and testing of DT-TAF** (as documented in D3.3 [Con25c]).
Risk Assessment (RA)	First view of *CONNECT* RA framework	**Final updates of *CONNECT* RA framework with the necessary primitives for the calculation of the RTL** based on specific threat profile of trusted system
CONNECT TCB	First implementation & evaluation of *CONNECT* TCB as a standalone component	Final Updates of *CONNECT* TCB including: **Integration & evaluation of Threshold DAA scheme as part of the crypto agility layer** (see D4.3) **Finalisation of IAM & Facility Layer implementation** with ID-related operations managed by IAM provided to the Facility Layer during onboarding (i.e., *ZC* and Vehicle Computers have preshared ID keys). The Facility Layer establishes the attestation keys. **Finalisation of AIV-MEC Integration** AIV instantiated as a Confidential Container within the MEC as part of the "AI-assisted moving object detection" to attest the service container where the application is being deployed. **Elevation of the Migration Service into two modalities**: i) Migration Operation of a CCAM service from one vehicle computer to another (i.e., explored in C-ACC UC) and ii) Live Migration of state of CCAM operation from one vehicle computer to another.
DLT (incl. ABAC & ABSE)	Initial standalone evaluation of the *CONNECT* DLT framework	**Final implementation extensions** including the Security Context Broker (SCB) supporting ABSE, that intercepts and mandates data stemming from the edge in a secure and privacy-preserving manner. **Integration and evaluation of the *CONNECT* DLT in the overarching framework** with the RTL integrated within the smart contracts and the trust policy being managed at a DLT-level.
CAM encoder	Initial desings and scope definition	**Integration & evaluation of ETSI-aligned CAM/CPM encoder** for the exchange of V2X messages between the vehicles as well as between the Vehicle and the MEC (including the Traffic Control Centre service).

The culmination of these efforts is visually represented in the final integrated architecture diagram in Figure 2.1. This final architecture introduces several key clarifications and enhancements over the initial release, reflecting the functional updates detailed in the table above.

A primary update clarifies the role of the Identity and Authentication Management (IAM). Its functionalities are now explicitly shown as part of the onboarding process, where it collaborates with the Facility Layer to securely provision identities to system components, as represented by the green dotted lines. The visualization of the Digital Twin (DT) has also been refined across both

the in-vehicle and MEC environments to more accurately represent its logical boundaries: the Standalone TAF operates within the DT (i.e., in the DT-TAF mode), while task offloading occurs externally. Furthermore, the finalization of key communication interfaces is now reflected. At the in-vehicle level, the link between the application and the CAM/CPM encoder is established, while at the MEC, the interface connecting the Facility Layer, the application, and the DENM encoder is complete. At the Zonal Controller plane, the diagram now distinguishes between the two operational modalities of the Migration Service at the Zonal Controller level. Lastly, in the central cloud, the integration between the Risk Assessment framework and the DLT is now finalized, completing a critical data flow for dynamic, trust-based policy enforcement.

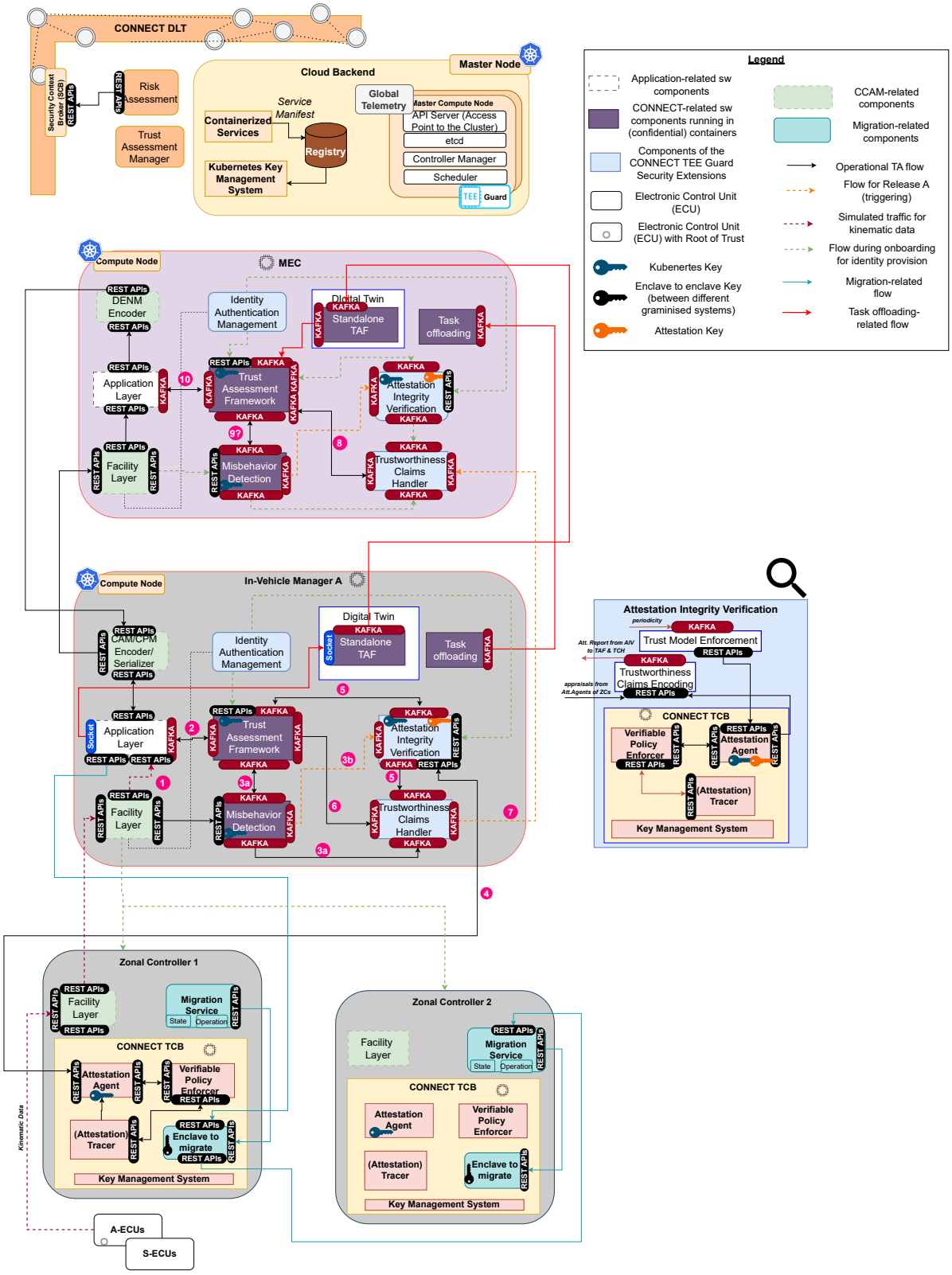


Figure 2.1: *CONNECT* Final Integrated Framework

2.2.1 *CONNECT* Final Version of TCH

2.2.1.1 Overview

The overall *CONNECT* framework supports the trust characterization of nodes and data that are internal to the V2X system (e.g., a CCAM application assesses the trustworthiness of the ECU that is responsible for the acquisition and processing of Lidar sensor data to enhance the vehicle's own perception) or external to the system (e.g., the application evaluates the trustworthiness of incoming Collective Perception Messages (CPMs) from other V2X objects, before processing them to enhance the vehicle's local perception of its surroundings). Focusing on the latter category, the overarching goal of the Trustworthiness Claims Handler (TCH) component is to aggregate all trustworthiness evidence that describe a particular V2X environment and convey them in an authentic, secure and privacy-preserving manner.

As presented in Figure 2.1, TCH is deployed in tandem with the rest of the Trust Sources running inside a V2X object that supports the CONNECT framework. The core responsibility is to convey evidence, namely Trustworthiness Claims, extracted from Trust Sources (e.g., AIV, or MBD) to external entities. This allows a TAF instance running on a remote V2X object to consume Trustworthiness Claims and form an opinion about the data that originate from that V2X source object. In order for this transmission of Trustworthiness Claims to work, there needs to be a TCH instance deployed both on the sender and the receiver V2X entity. In Figure 2.1, this is captured with the provisioning of a TCH instance on both the vehicle manager and the MEC server. Apart from the transmission of Trustworthiness Claims characterizing the target environment of a V2X object, a TCH instance is also responsible for consuming, validating, and forwarding incoming Trustworthiness Claims to its corresponding TAF instance. This creates a bidirectional TCH-to-TCH, which can be realized in different ways. In the context of the evaluations of the use cases, we have considered a dedicated channel for exchanging trustworthiness claims. In addition to that, we have also examined the case where Trustworthiness Claims are embedded within the application payload (see evaluations of task offloading scenario in D5.3 [Con25b]).

In terms of the TCH implementation roadmap, all core functionalities have been developed in its first version. These allow for the transmission of Trustworthiness Claims allowing a TAF instance to evaluate trust properties pertaining to different V2X nodes and data. The second version focused on targeted enhancements to existing capabilities that unlock a thorough and detailed second round of evaluations in the context of the *CONNECT* use cases. On the one hand, these enhancements focus on the core TCH Logic, extending the on-demand transmission of Trustworthiness Claims invoked by a third application, with capabilities to initiate the process periodically based on a predefined parameter to indicate the frequency of transmissions. In addition, as presented in the latest versions of the implemented interfaces (see Section 2.2.1.3), the encoding of the TCs has been extended to incorporate an optional tag field which allows for the end-to-end evaluation of the CONNECT TAF calculations in the context of the SMTD use case: associating a TAF trust report with a particular TC through the new tagging mechanism. On the other hand, the packaging and deployment of the TCH component is adjusted so as to allow its instantiation within a Trusted Execution Environment. This allows for the evaluation of the overhead that Trusted Computing mechanisms impose on the overall trust calculations as part of the CONNECT TAF (see evaluations in D5.3 [Con25b]). In general, details on the instantiation of TCH instances in the context of the SMTD use case are presented in 5.2.6.

In what follows, we present the implementation details of the latest version of the TCH component (Section 2.2.1.2). Finally, Section 2.2.1.3 presents the core interfaces that allow the configuration

and transmission of Trustworthiness Claims from a TCH instance.

2.2.1.2 Implementation Path Report

TCH constitutes a service that is deployed at the border of a V2X entity -e.g., vehicle, road side unit, or MEC server - and constructs trustworthiness claims about the target environment of that particular entity. The core TCH functionalities refer to the preparation, encoding/decoding and conveying trustworthiness claims to other V2X entities. In general, the implementation of the entire TCH component consists of the following three subcomponents (Figure 2.2):

- Redis database: an in-memory, key–value storage used for persisting configuration parameters (e.g., which trust sources to interact with, frequency of trustworthiness claims transmission, in-vehicle object identifiers and endpoints) relevant to the operation of the TCH.
- TCH Logic: service developed in Python, which implements the core functionalities that allow the creation and transmission of trustworthiness claims to be sent to external V2X entities. Specifically, these functionalities include the aggregation of evidence from underlying Trust Sources in the target environment where the TCH instance is deployed. For example, this involves the invocation of the AIV functionalities for the secure collection of attestation results about the correct state of the in-vehicle on-board unit. It is worth noting that this involves the data curation with respect to the collected evidence to avoid any privacy-sensitive information (e.g., the exact identifier or number of ECUs in the in-vehicle topology). As documented in the CONNECT TCB architecture in [Con25d], this includes also the provision of the threshold DAA mechanisms that enable the aggregation of attestation results to avoid disclosing any information about the ECU topology within the in-vehicle topology. Despite the fact that the invocation of the threshold DAA functionalities through the TCH Logic is implemented, the evaluations presented in the context of the use cases do not include the use of this mechanism. In addition, the TCH Logic incorporates the process of incorporating the necessary integrity and authenticity guarantees that allow the secure construction of the trustworthiness claims. This allows the receiving TCH instance to evaluate the incoming trustworthiness claims and report them properly to the corresponding TAF instance for the trust calculations taking place there.
- Kafka Client: processes that allow the interfacing of the TCH logic with external components - i.e., applications or other *CONNECT* components. Through Apache Kafka topics, applications are able to configure and request the creation of new trustworthiness claims to be broadcasted. In addition, through dedicated topics, a TCH instance is able to receive incoming trustworthiness claims, verify their authenticity, and forward them - again through Kafka - to the corresponding TAF instance in order to incorporate them as new evidence in their calculations. Aligning with the integration points on the Kafka topics convention, each TCH instance listens to a single topic - e.g., ”tch” - and expects to receive any type of message that is destined for it. The processing of the message takes place as part of the TCH Kafka Client which invokes the necessary functionality in the TCH Logic.

The TCH Logic and its corresponding Kafka Client is bundled and deployed as a dockerized image. Using Docker allows for seamless management and portability of TCH instances in the various V2X entities that are envisioned as part of the evaluation environments (see Section 5.2.6 for more details on the instantiation of TCH in the context of the use cases).

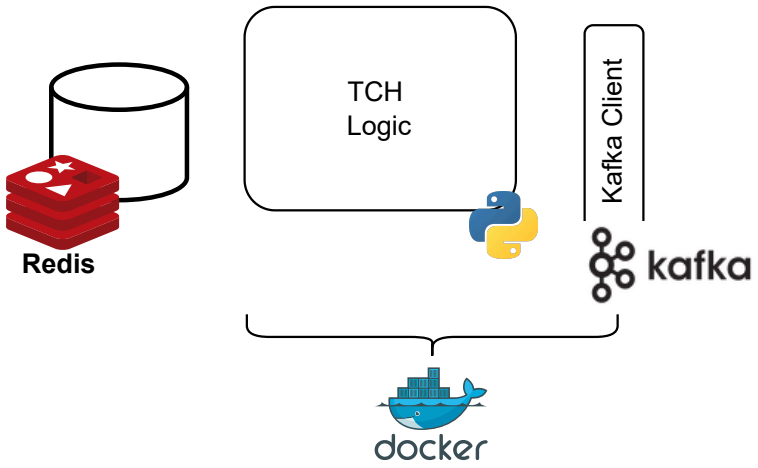


Figure 2.2: TCH implementation blueprint

2.2.1.3 Implemented interfaces

This section documents the latest version of the core interfaces that allow external components to interact with a TCH instance. First, the interfaces enable the configuration of the creation of trustworthiness claims pertaining to the environment where the corresponding TCH instance is deployed (Table 2.2). In addition, the interface reported in Table 2.3 allows for the on-demand collection and transmission of trustworthiness claims to an intended Kafka Topic specified as configuration parameter. It is worth noting that for both of the aforementioned message types are accompanied by an acknowledgement sent to the intended Kafka topic as specified in the request payload (i.e., ”responseTopic” field). Finally, the interface documented in Table 2.4 captures the receipt of trustworthiness claims into a receiving TCH instance. When a new trustworthiness claim arrives in a TCH instance, then the TCH Logic validates the authenticity of the incoming evidence; if the authenticity stands, then the TCH instance signs the tchReport field with its own cryptographic keys and forwards it to the corresponding TAF instance. This allows the corresponding Trust Source Manager to quantify a trust opinion characterizing the object id enclosed in the trustworthiness claims.

<table>
<tr><th>TCH</th><th colspan="2">Interface Technical Details</th></tr>
<tr><td>Type of Interface</td><td colspan="2">Kafka</td></tr>
<tr><td>Purpose</td><td colspan="2">Configuring TCH to prepare for capturing properties and objects for which it must generate trustworthiness claims.</td></tr>
<tr><td rowspan="2">Inputs & Outputs</td><td>Inputs</td><td>Outputs</td></tr>
<tr><td><pre>{
    "sender": "<application-id>",
    "serviceType": "TCH",
    "messageType": "TCH_INIT_REQUEST",
    "responseTopic": "application_topic",
    "requestId": "4c54a50f8e43",
    "message": {
        "query": [
            {
                "trusteeIDs": ["object-id"
                    ]
            }
        ]
    }
}</pre></td><td><pre>{
    "sender": "<tch-id>",
    "serviceType": "TCH",
    "messageType": "TCH_INIT_RESPONSE",
    "requestId": "4c54a50f8e43",
    "message": {
        "success": "Initialization
            successful"
    }
}</pre></td></tr>
</table>

Table 2.2: Interface for instantiating the logic for TC issuance on a particular object identifier

TCH	Interface Technical Details	
Type of Interface	Kafka	
Purpose	Request from TCH to issue a new trustworthiness claim	
Inputs & Outputs	Inputs	Outputs

	{
	"sender": "<application-id>",
	"serviceType": "TCH",
	"messageType": "TCH_TC_REQUEST",
	"responseTopic": "application_topic",
	"requestId": "4c54a50f8e43",
	"message": {
	"query": [
	{
	"trusteeID": "object-19",
	"requestedClaims": [
	{
	"name": "Claim_A"
	}
	]
	}
	]
	}
	}
		{
		"sender": "<tch-id>",
		"serviceType": "TCH",
		"messageType": "TCH_TC_RESPONSE",
		"requestId": "4c54a50f8e43",
		"message": {
		"success": "Initialization successful"
		}
		}

Table 2.3: Interface for requesting the issuance of a new TC

TCH	Interface Technical Details	
Type of Interface	Kafka	
Purpose	Process an incoming TC sent by a TCH instance.	
Inputs & Outputs	Inputs	Outputs

	{
	"sender": "<tch-id>",
	"serviceType": "TCH",
	"messageType": "TCH_NOTIFY",
	"tag": "<optional-tag-name>",
	"message": {
	"tchReport": {
	"trusteeReports": [
	{
	"attestationReport": [
	{
	"appraisal": 0,
	"claim": "Claim_A",
	"timestamp": "2024-11-04T12:46:48.652461Z"
	}
	]
	}
	],
	"trusteeID": "<object-id>"
	},
	"evidence": {
	"timestamp": "2024-11-04T12:46:48.658081Z",
	"signatureAlgorithmType": "ECDSA-SHA256",
	"signature": "3044022...",
	"keyRef": "object-public-key-URI"
	}
	}
	}
		None

Table 2.4: Interface for receiving trustworthiness claims

2.3 Unit Tests performed per Component

To establish a baseline of reliability and functional correctness, we executed the comprehensive suite of unit tests first specified in D6.1 [Con24b]. This systematic methodology ensures that each software module and security primitive executes its designated functions accurately prior to integration into the broader *CONNECT* framework. Table 2.5 summarizes the results of this critical

validation phase, presenting the pass/fail status for each component. Successfully passing these tests is a prerequisite for the system-level evaluations, confirming that the foundational building blocks are stable and ready for the complex interactions detailed in the subsequent chapters.

Table 2.5: CONNECT Framework—Unit, Integration Tests & Results

Component	Tested Functionality	Test Details	Result
TAF	Session handling	Creation + teardown of valid/invalid sessions	Pass, without issues encountered even in a session with large amount of data
	Trust model initialization	Valid/invalid template and parameter scenarios	Pass, successful testing of all APIs for updating the TMs including RTL coming from the RA
	TA requests	Handling with/without filters, caching settings	Pass, further optimisations where provided for the caching of the data and sharing with TLEE
	Subscription logic	Subscribe/unsubscribe, notifications	Pass and functionally tested in terms of the UCs
	AIV evidence handling (integration with AIV)	Subscribe/unsubscribe, pull/push flows	Pass for different type of evidence that AIV offers for configuration and operational correctness
	MBD evidence handling (integration with MBD)	Subscribe/unsubscribe, notification processing	Pass for different types of evidence that the MBD offers for misbehaviour detection
	TCH evidence handling (integration with TCH)	Subscribe/unsubscribe, notification processing	Pass
	V2X CPM processing	Reception and parsing via V2X	Pass
DT-TAF	Execution of the TAF computations related to a specific TAF instance in an edge node	TAF computations in the edge node are based on a replica of the TAF instance state	Pass, successful testing of all APIs for publishing TCs into both the standalone and DT-TAF
	TAF-MEC to TAF-MEC communication	Communication with other TAF instances inside MEC	Pass, leveraging KAFKA
	DT Confidentiality	Confidentiality of the TAF internal state inside the edge node	Partially achieved, successful testing of instantiation of one DT-TAF per MEC-enabled server
	DT Integrity	Integrity of the TAF internal state inside the edge node	Pass since DT-TAF was instantiated as a confidential container
MBD	Attack injection	CPM packet alteration and delivery	Pass, successful attack injection
	Detector checks	Detection in CPM fields	Pass, successful misbehaviour detection
	MR generation	JSON report creation	Pass, successful report generation
	MR transmission	TCP transmission to proxy	Pass, successful report transmission to proxy
MBD Proxy	MR parsing	Proper formatting, zero-activation check	Pass, successful parsing of the misbehaviour report
	Subscription logic (integration with TAF)	Session management for subscriptions	Pass, successful subscription to the TAF

Continued on next page

Table 2.5 – continued

Component	Tested Functionality	Test Details	Result
	Extended Perception (integration with TAF)	Retrieves and processes ATL via proxy	Pass, considering previous ATLs but as a standalone component outside of the TAF
	Session initialization (integration with TCH)	Node-ID-based session setup	Pass
	TC request generation (integration with TCH)	TC creation based on CPM sources	Pass
RA Engine	API interface testing	REST endpoint validation	Pass all interfaces successfully tested (for details see D3.3 [Con25c])
	Kafka interface testing	REST endpoint validation	Pass, asynchronous updates to the threat modelling subcomponent (e.g., new vulnerabilities associated with a particular asset) are captured by the CONNECT RA Engine and the risk graph is updated automatically.
	Attack path calculation testing	Execution of Attack Path Calculation Engine internal operation	Pass, identification of attack paths in in-vehicle topology that helps enhance the threat modelling as part of the TARA process.
	Enhancement with TARA methodology	Get TARA risk reports both in UI and as REST APIs	Pass, validated for in-vehicle topology and used for RTL calculation (for details see D3.3 [Con25c])
	Enabling RTL calculations	Successfully tested filtering capabilities of risk reports REST APIs for the RTL derivation	Pass, filter risk reports for a single context that characterizes a particular (in-vehicle) asset and security property; thus allowing for RTL calculations (for details see D3.3 [Con25c])
AIV	Scalability under load	multi-thread attestation via Gramine	Pass, successful scalability testing
	Verifiable Signing of attestation report	Correct construction of attestation key and secure interaction with the underlying tracer	Pass, successful signature construction
	Runtime integrity of AIV	Instantiate or secure launch and runtime and integrity checks	Pass, successful instantiation of AIV as a trusted application inside SGX gramine and runtime introspection of its integrity during runtime i.e., based on *CONNECT* tracing extensions of the Gramine library
	Integration with TAF	Pull and push based interactions with the TAF	Pass, without issues, supporting both types of operations

Continued on next page

Table 2.5 – continued

Component	Tested Functionality	Test Details	Result
	E2E testing of *CONNECT* TCB	Successful interactions with Attestation Agent and VPE for the secure and authorised management of KRUP	Pass, without any issues. Detailed results elaborated in D4.3.
	Secure coding of CIV	Memory safety testing - Leak/buffer overflow detection	Pass
TCH	Integration with Trust Sources	Preparing trustworthiness claims - characterizing evidence coming from in-vehicle trust sources - to external V2X entities	Passed, considering AIV and MBD as Trust Sources
	TCH-to-TCH communication	TCH on the sender V2X entity produces (and signs) trustworthines claims that are sent to the TCH entity of the receiver V2X entity for validation and forwarding to the corresponding TAF component	Passed, evaluated through a dedicated Kafka channel for TCH-to-TCH communications.
	Scalability of TC processing from a receiving TCH	Sending multiple TCs representing multiple V2X objects to a single TCH entity and testing that the TCs are successfully validated and forwarded to the connected TAF component.	Passed, considering attestation claims transmitted from multiple vehicles to a single MEC server.
	Deploying TCH under different TAF modalities	Checking that a single TCH instance is able to serve a standalone and a federated TAF instance	Passed. On the one hand, TCH is able to handle incoming TCs and forward them to a standalone TAF instance. On the other hand, TCH can extend this functionality and share - at the same time - Trustworthiness Claims characterizing its own target environment to external TAF entities that participate in a federation scheme.
Migration Service	Stateful migration robustness	CCAM operation migration	Pass (this is as a black box operation)
	Stateful migration robustness	CCAM state migration	Pass, this is part of *CONNECT* novel migration scheme, based on verifiable state management of functionalities
Task Offloading	RTSP pipeline	ffmpeg + GStreamer WebSocket playback	Pass without issues. Detailed results available in D5.3.
	TLS handshake	WebSocket/TLS handshake test	Pass without issues. Detailed results available in D5.3.
	Kafka integration	Produce/consume tests locally	Pass without issues. Detailed results available in D5.3.
	Client data parsing	Parsing and printing received metrics	Pass without issues. Detailed results available in D5.3.

Continued on next page

Table 2.5 – continued

Component	Tested Functionality	Test Details	Result
	Video streaming	WebSocket feed to GStreamer	Pass without issues. Detailed results available in D5.3.
	DeepStream RTSP/WebRTC	Overlay display, latency	Pass without issues. Detailed results available in D5.3.
	Secure interaction between TAF instances as part of the offloading		Pass without issues. Detailed results available in D5.3.
DLT	Peer certificate	Certificate generation test	Pass without issues. Detailed results available in D5.3.
	Attestation Report validation	Signature validation	Pass, successful signing by the Attestation Agent, leveraging the safeguarded attestation key and verification of the signature on the SCB
	Query functions	Query templates/reports test	Pass, this includes the successful querying of templates for both *CONNECT* components and external users
	Transaction block	Attestation pointer inclusion	Pass without issues. Detailed results available in D5.3.
	ABAC	Attribute-based access control	Pass without issues. Detailed results available in D5.3.
	Chaincode	Smart contract creation test	Pass without issues. Detailed results available in D5.3.

2.4 Evaluation in terms of Functional Specifications

In this section, we present an overview of the functional specifications evaluated as part of the system assessment. The Key Performance Indicators (KPIs) used for this evaluation were initially introduced in D2.1 [Con23c] and D2.2 [CON23a]. Each entry in Table 2.6 is identified by a unique ID and title, along with the corresponding KPI and the resulting status. Color coding is used to indicate the level of achievement: green for fully achieved KPIs, yellow for partially achieved, and red for those not achieved. Additionally, gray highlights the KPIs that were part of the Minimum Viable Product (MVP) as defined in D2.2. This structured representation enables a clear understanding of the system's progress toward meeting its functional goals.

Please note that FR.MEC.2 and FR.MEC.3 and part of FR.EC.4 that pertain to the consideration of how mobility can affect the consumption of services (i.e., trust assessment services) instantiated on the MEC were deemed obsolete considering the focus of *CONNECT* experimentation on the behavior or the trust assessment framework and the auxiliary security controls.

Table 2.6: Technical Requirements Overview

ID	Title	KPI	Status
Trust Assessment			
FR.TR. 1 TAF	Generalizability	**=3** heterogeneous CCAM use cases supported, capturing in-vehicle, vehicle to vehicle and vehicle to MEC (and vice versa) scenarios	KPI achieved & exceeded, see D3.3 where additional experiments were performed beyond the scope of the use cases

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
FR.TR. 2 TAF	Run-time Performance	$\leq$ **100 ms** delay when the standalone TAF is instantiated and executed as part of the application software stack in the target system (outside the CONNECT TEE)	KPI achieved see D3.3, even for high complexity trust models, trust assessment takes less than 50 ms
		$\leq$ **200 ms** delay when the standalone TAF is instantiated and executed within the CONNECT TEE	KPI achieved see D3.3. It shall be noted that the metric depends heavily on the employed programming language
FR.TR. 3 TAF	Scalability	$\leq$ **10 nodes** supported for small scale environment (IMA - Single TAF)	KPI achieved. TAF is able to handle more than 10 nodes as discussed in D3.3. As it pertains to the use case results further details documented in Chapter 3.
		$\leq$ **50 nodes** supported for large scale environment (IMA - vehicle or MEC; Federated TAF)	KPI achieved. TAF is able to handle more than 50 nodes in the federated TAF case, as explored in the IMA use case. Further details documented in Chapter 3.
		Time to converge on a trust decision: $\leq$ **100 ms** (outside TEE)	KPI achieved, 60-70 ms are required to derive a trust decision see D6.1 and D3.3
		Time to converge on a trust decision: $\leq$ **200 ms** (inside TEE)	KPI achieved, overhead posed by TEE is below 10% for SW-based TEE and below 40% for HW-based TEE as documented in D3.3
FR.TR. 4 TAF	Correctness	All entities are trustworthy (i.e., not compromised by an attacker)	KPI achieved by integrating CONNECT TCB into all CCAM entities safeguarding the operation of CONNECT security services; i.e., launching TAF as a trusted application (see D3.3)
		One or several entities are not trustworthy because they have been compromised by an attacker	KPI achieved, TAF successfully identifies compromised entities see D3.3 and further discussed in the context of the use cases see Chapter 3
FR.TR. 5 TAF	Robustness and Resilience	All TAF internal components to be able to be instantiated inside a TEE	KPI achieved and exceeded deemed by comparing the TAF enclavisation as a process in (gramine SGX) or as a confidential container (TDX) see D3.3
FR.TR. 6 TAF	Flexibility of Trust Sources	$\geq$ **3** different trust sources are included in the use cases	KPI achieved see D3.3.
Security			

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
FR.SR. 1 ECUs, Zonal Controllers, Vehicle, MEC	Secure and Efficient Cryptography	Size of the TCs $<$ **30 bytes**, so that it can fit into the existing security header of the standardised CAM definition.	Not achieved, as the size of the DAA claim is 90 bytes. Thus, further investigation is required, since it becomes cumbersome to be integrated in the existing standardised CAM message profiles.
		Signing and verification times for the TCs (for a typical vehicle OBU). **100 signing/verifications per second** as per ETSI specifications.	This is dependant on the employed type of signature. Currently in *CONNECT* we opted for leveraging ECC-based crypto primitives that allow also for strong privacy (i.e., user controlled linkability). This manifested in DAA or threshold DAA sign operations consuming about 20-45 ms. This adds a 50% overhead to the acceptance criteria delineated by the standards. However, there is an interplay between privacy assurances and efficiency. Weakening the privacy requirements which will result in employing more traditional primitives like ECDSA will meet these requirements in conjunction with the crypto acceleration capabilities with the underlying secure element.
FR.SR. 2 ECUs, Zonal Controllers, AIV, TCH, Key Management System, TAF, MEC	Dynamic Credential Management	Number of crypto operations (i.e., signature encryption per second) **100 signing/verifications per second**	See FR.TR.1
		Different types of crypto primitives that can be supported (for different ECU types) $>$**=3**	KPI achieved and exceeded through *CONNECT* crypto agility layer featuring 4 anonymous group type and threshold signatures; i.e., DAA based on CL and BBS signatures, ECDSA signatures and elevating this group type anonymous signatures to a threshold setting based on the integration of FROST signature scheme. *This resulted to the first of its kind implementation and benchmarking of FROST based on pairing friendly curves.*

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
		$\leq$ **30%** overhead introduced by the size of crypto structures (i.e., signatures and certificates), associated to trust related information	KPI not achieved, as the resulting overhead in the shared credential is around 50%. As was highlighted in FR.TR.1 this is again primarily due to the strong privacy assurances provided in *CONNECT* which requires the inclusion of zero knowledge proofs (ZKP) - on the ownership of specific attributes and trust properties - that can demonstrate the trust level of an entity while abstracting trustworthiness information details.
		$\leq$ **30%** overhead introduced by the trusted computing mechanisms provided by the CONNECT TEE Guard in the computational resources (i.e., CPU cycles for the crypto operation execution)	KPI achieved as all of the crypto primitives and resulting credentials can be securely managed by the *CONNECT* TEE Guard with negligable overhead
FR.SR. 3 ECUs, Zonal Controllers, Vehicle, MEC, TAF	Flexible and Reliable Key Management	**around 25% overhead** against using keys with and without key restriction usage policies in the employed TEE Guard	KPI not achieved per se; nevertheless the introduced overhead is deemed acceptable against the privacy-benefits gain. More specifically, around 19 ms are required for the creation of a signature without being predicated through a Key Restriction Usage Policy and 47 ms under the employment of such a policy. The operations are in order of ms; thus, the overhead is negligible.
		$\leq$ **60 ms**; needed for construction of different types of keys, supporting the entire lifecycle of a system (i.e., from its authentication and onboarding to its application participation and trust related evidence secure communication)	The KPI was met for the standard DAA protocol. In contrast, the Threshold DAA key generation & distribution of key shares is a one-time, setup event requiring 407-1,677 ms (for 32-128 ECUs) in a static scenario where no key re-sharing is needed. This latency, while exceeding the KPI, is well within acceptable limits for its intended purpose of securing in-vehicle zones (while also safeguarded against vehicle fingerprinting) during non-time-critical operations like group onboarding and configuration.

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
		$>$ **4** types of keys supported and maintained (i.e., identity key, integration key, authentication key, attestation key, etc.)	KPI achieved considering an extended key hierarchy not only manifesting on primitives needed for the security and privacy of the entire lifecycle of a vehicle's operation but also for enabling the verifiability of the trust assessment process. As detailed in D5.3 *CONNECT* designed a novel attribute based signcryption scheme where attributes representing discrete trust properties (simulated of up to 30 properties of interest) are safeguarded by separate keys.
FR.SR. 4 ECUs, Zonal Controllers, Vehicle, MEC, TAF	Secure Data Handling and Provenance	$<$ **2sec** to impose controlled linkability (i.e., in terms of the time needed to integrate the necessary crypto primitives such as link token, as part of a TC, so that only authorised CCAM actors, like such as OEMs, can link back to an in-vehicle system).	KPI achieved. This manifest on the use of either traditional DAA (around 600 ms for the entire process) or threshold DAA (around 850 ms) enabling controlled linkability and unforgeability
		Vehicle privacy exposure due to the communication of trust related information **FALSE**	KPI achieved, see row above
Runtime Operational Correctness			
FR.OC. 1 TCB	Common Trusted Computing Protocols	Granularity of Levels of Assurance (LoA) that can be achieved by various RoTs and essentially the common trusted computing base $\geq$ **5 LoA**;	KPI Partially achieved as the attestation scheme's design can achieve the highest level of assurance but in *CONNECT* use cases only LoA 2 was tested
		$\geq$ **3** Number of operations supported by TCB (i.e., secure storage, secure boot, key management, etc.)	KPI achieved see Table 4.11.

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
FR.OC. 2 TCB	Operational Assurance & Configuration Integrity	Time needed for the execution of the local attestation assuming the provision of authenticated runtime measurements $<$ **200ms** when the attestation process is instantiated and executed outside the CONNECT TEEs	KPI partially achieved. High-performance Vehicle Computers successfully achieved the $<$200ms target. In contrast, platforms representing more resource-constrained components failed to meet this KPI. Specifically, tests conducted on Zonal Controllers deployed on Raspberry Pi hardware, as well as on StarFive VisionFive 2 boards[1] representing low-end ECUs, consistently exceeded (almost doubled) the 200ms threshold. This performance limitation is attributed to their less powerful CPUs and slower I/O capabilities, which impact the speed of cryptographic operations and hardware bus communication. [1]
		$<$ **10% overhead**, when the attestation process is instantiated and executed inside the TEEs.	KPI achieved and even exceeded the initial goal by leveraging the cryptographic acceleration capabilities of SGX. As presented in [cTPV17], TEE performance overhead has two parts: a fixed startup cost based on size (for a 256 MB enclave, the latency is 0.5 s) and a more significant, variable runtime cost caused by I/O operations. This runtime overhead, which depends on the application's specific workload, is the dominant factor. For CPU-bound workloads that do not perform many system calls, this overhead is low, typically around 2-8% [cTPV17].[ATG+16].

Continued on next page

[1] We opted to use an FPGA as a basis for the experiments with the following characteristics Starfive Visionfive 2, a RISC-V JH7110 4 core System On a Chip (SoC), with 4-8GB of ram and a discrete GPU.

Table 2.6 – continued

ID	Title	KPI	Status
		Time needed for the construction and signing of the TCs $<$ **900 ms**	KPI achieved and exceeded, as documented in D4.3 the CIV process consumes almost 47 ms in Gramine SGX for producing the verifiable TCs. Further abstractions of the TCs to be enforced by the TCH when employing the Threshold DAA incurs an additional overhead of 461 ms, which is less than the acceptance threshold value.
FR.OC. 3 TCB	Integrity Verification of CCAM Components	Time needed for the execution of configuration integrity verification $<$ **100ms** when the attestation process is instantiated and executed outside the CONNECT TEEs	KPI achieved. Initial experimental activities indicate that devices with sufficient resources, such as Zonal Controllers and Vehicle Computers, are able to achieve the target value. In contrast, low-end devices consistently fail to reach comparable performance levels.
		$<$ **25% overhead**, when the attestation process is instantiated and executed inside the TEEs and leverages the established key restriction usage policies enabling local configuration integrity verification	KPI achieved see FR.OC.2-b.
FR.OC. 4 TCB	Chain of Trust Creation	Storage of trust-related information to the Blockchain for auditability and certifiability $\leq$ **5sec**, since this is not a real-time operation.	KPI achieved. Total time for writing an attestation report or trust policy to the DLT does not exceed 2.1 seconds as documented in D5.3.
		The AIV should be able to request transmission of fresh raw evidence by the ECUs depicting the current state (so as to be used as a trust source). $\geq$ **1 Mbit/sec** data transfer rate for the raw evidence	KPI not achieved based on the observations on the periodicity of evidence needed for increased TAF accuracy (see Section 4.3.1). Reaching such high bandwidth would require access to CAN-FD communication bus.
		Simulate low bandwidth channel with high message loss.	KPI achieved through extensive microbenchmarking and end-to-end testing in the context of the task offloading scenario as detailed in Section 5.6.1.4.
FR.OC. 5 TCB	Secure Measurement / Attribute Extraction	Efficiency of tracing and device state monitoring $\leq$ **500 ms**	KPI achieved and even exceeded. As documented in D4.3, *CONNECT* tracing and memory introspection capabilities (i.e., extracting application state) enabled through Gramine extensions require on average 41 ms.

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
FR.OC. 6 TCB	Secure Remote Asset Management and Reconfiguration Effectiveness	Update and adaptation of the trust models capturing the newly updated asset.	KPI achieved and detailed in D3.3 where microbenchmarking of TAF considered a varying volume of trust model instances.
		Deployment (and revocation of old) of new system configurations including also possible SW upgrades safeguarded by the CONNECT TEE Guard	KPI achieved. Focus was placed primary on the correct updates of the key restriction usage policy
Function Isolation and Migration			
FR.SF. 1 TAF	Dynamic awareness on potential vulnerabilities and threats and complete overview of the deployed environment	Dynamic Risk Assessment based on the identification of new threats $\leq$ **2sec** considering the identification of a new threat (by a security administrator) based on monitored evidence collected as part of a failed attestation process that indicates a possible risk.	KPI achieved even in the case for in-vehicle topology comprising 15 SW assets and 5 HW assets (i.e., Zonal Controllers) resulting in 26 relationships to be evaluated as part of the trust model and 12 attack scenarios as identified by the *CONNECT* risk assessment. In this context the whole risk assessment pipeline took 1560 ms, out of which around 70% is consumed by the internal path calculation building block (see D3.3).
		Recalculation of RTL considering the identification of new risks $\leq$ **3 sec**	N/A. Focus was shifted into vignetting ATL/RTL comparison during the trust decision process by identifying a methodology that is generic enough to accommodate both calculations with the same level of uncertainty/belied/disbelief. [2]
FR.SF. 2 TEE	Stateful Function Upgrade	Upgrading the function in one ECU $<$ **750 ms**	KPI achieved and exceeded. As reported in D4.3 the time needed in the live migration of an application state for upgrading the function in a target ECU does not exceed 549 ms at the tested scenarios.

Continued on next page

[2] This is a proposal to the 5GAA.

Table 2.6 – continued

ID	Title	KPI	Status
FR.SF. 3 TEE	Stateful Function Migration	Establishing a similar function on another box $<$ **1 sec** Note that CONNECT focuses only on the downtime/availability	KPI partially achieved. In the case of the imminent driving scenario where cashed trustworthiness evidence can be considered for the trust decision cycle, the migration process takes less than 1 sec. However, in cases where fresh evidence need to be collected, the operational migration takes around 2 seconds as detailed in Section 4.1.8.1.
MEC-Functional [3]			
FR.MEC.1 MEC	Operational requirements on (MEC) application lifecycle and application environment	about **30 seconds** for container LCM Operations [referring to Instantiation, termination, modification time]	KPI achieved and exceeded. The Application Server container of the CONNECT offloading pipeline was used to measure the latency for the instantiation of a new container. The latency for the initialization of the container itself together with the initialization of the Application Server (inside the container) was on average (over 10 samples) found equal to 420ms, with a standard deviation of 33ms.
		about **30 seconds** for container LCM Operations [referring to Instantiation, termination, modification time] when the original host avails the needed capacity.	KPI achieved and exceeded. Similar in spirit to the above, measurement on the reconfiguration of the Application Container and its restart were taken. In a series of 10 experiments, the assigned memory of the Application Server container of the CONNECT offloading pipeline, was reconfigured between values of 8Gb and 16Gb and then the container was restarted. We report an average latency of 10385ms with 13ms standard deviation. Consequently, all our LCM Operation measurements suggest that even in the case of a container downtime, the involved applications require reasonable time (i.e., in the order of tens of seconds). Those results may well reflect any CONNECT application container and its trust-related functionality.

Continued on next page

[3] All KPIs related to MEC operations were measured as part of the task offloading functionality tested in *CONNECT*.

Table 2.6 – continued

ID	Title	KPI	Status
		$<$ **20ms** E2E latency from the moment the IP ICMP Echo Request packet leaves the source host (e.g., vehicle) until the IP ICMP Echo Reply is received from the destination host (e.g., MEC) Plus $<$**15%**) expected overhead caused by the Trustworthiness Claims included in CONNECT	KPI achieved as detailed in Section 5.6.1.3. Extensive testing was done for measuring the end-to-end latency under different periodicity of TC transmissions. The end-to-end latency was measured from the moment of the Trust Assessment request so as to yield more meaningful results that can better depict the impact on the network operation. Even in the case where TCs are transmitted together with each V2X message, the incurred overhead is on average $<$ 400 ms.
FR. MEC. 4 MEC	Operational Requirements for (applications) connectivity	$<$**100 ms** inter-containers communication latency (measured as IP ICMP Echo Request packet leaving the source host (e.g., container A, host A) until the IP ICMP Echo Reply is received by the destination host (e.g., container B, host B).	KPI achieved. RTT latency was measured by issuing Ping command between the Application Server container and the Application Client container of the CONNECT offloading pipeline. Both containers are deployed on the same INTEL i9 Linux machine. Our experiment suggest an average value (out of 10 samples) of $RTT\ latency$=$0.041ms$ and corresponding Standard of Deviation=$0.013ms$. Clearly, the inter-container latency depends on numerous parameters such as the underling hardware capabilities and the complexity of the application containers; still, the provided results suggest that the involved communication overhead, potentially induced by trusted computing applications, remains negligible.
MEC-Security			
SR. MEC. 1 OEM, MNO, Service Provider, TAF, TEE Guard, Risk Assessment Engine, MBD	Security Requirements on MEC service authorization and access (authorised service access)	$<$ **2 sec** (excluding network latency) authentication latency (i.e., onboarding of the vehicle into one administrative domain managed by one single MNO) leveraging both *CONNECT* adopted mechanisms (i.e., PKI) as well as *CONNECT* developed authentication controls (i.e., leveraging the newly-developed trust extension	KPI achieved. The JOIN phase consumes 5.8 ms for the creation of the attestation keys and the the overhead imposed for the creation of the enclave as described in FR.OC.2 is maximum 8%.

Continued on next page

Table 2.6 – continued

ID	Title	KPI	Status
		$<$ **20%** overhead posed by *CONNECT* TCs that need to be exchanged during authentication so that it does not incur packet fragmentation. In the context of latency at MEC application-level [with expected value in the range of **10 to 200ms**].	KPI partially achieved. Regardless of the type of application container deployed overhead in the authentication process due to TCs is between 12-35% due to the overhead of coordination across enclaves over encrypted RPC streams.
SR. MEC. 2 OEM, MNO, Service Provider, TAF, TEE Guard, Risk Assessment Engine, MBD	Security Requirements on virtualization and containerization technology (employed -among others- at the MEC	Degree of coverage of the (defined) virtualization/containerization threats. $>$**95%** degree of coverage against the (defined) virtualization/-containerization attacks targeting both the secure launch and operational integrity of the confidential containers)	KPI achieved. The trust assessment framework was able to react (i.e., reduce the ATL) considering the threat model as defined in D2.2.
(non-functional) Privacy			
FR.PR. 2 Operators of MEC, and Operators of V2X infrastructure	Unlinkability of data provenance	TRUE	KPI achieved and evaluated as part of the standalone benchmarking in D4.3
FR.PR. 3 Operators of communication components and operators of components of trust assessment framework	Attributes related to vehicle trustworthiness should not create privacy threats with medium or high Level beyond those already in existence in the CCAM ecosystem	TRUE	KPI achieved and evaluated as part of the standalone benchmarking in D4.3
FR.PR. 5 Operators of MEC, and Operators of V2X infrastructure	Trust information and assessment lifecycle management and user acceptance	TRUE	On the users as CCAM providers (see Chapter Ethics)

2.4.1 Components mapped to Use Cases

In this subsection, we illustrate how the system components are mapped to the defined use cases. The accompanying figure presents the overall architecture, with additional annotations indicating which components are involved in each use case. This visual mapping provides a clear understanding of the functional distribution across the architecture and highlights the roles of individual components in supporting specific use case scenarios. The following chapters will delve into the specifics of each use case evaluation.

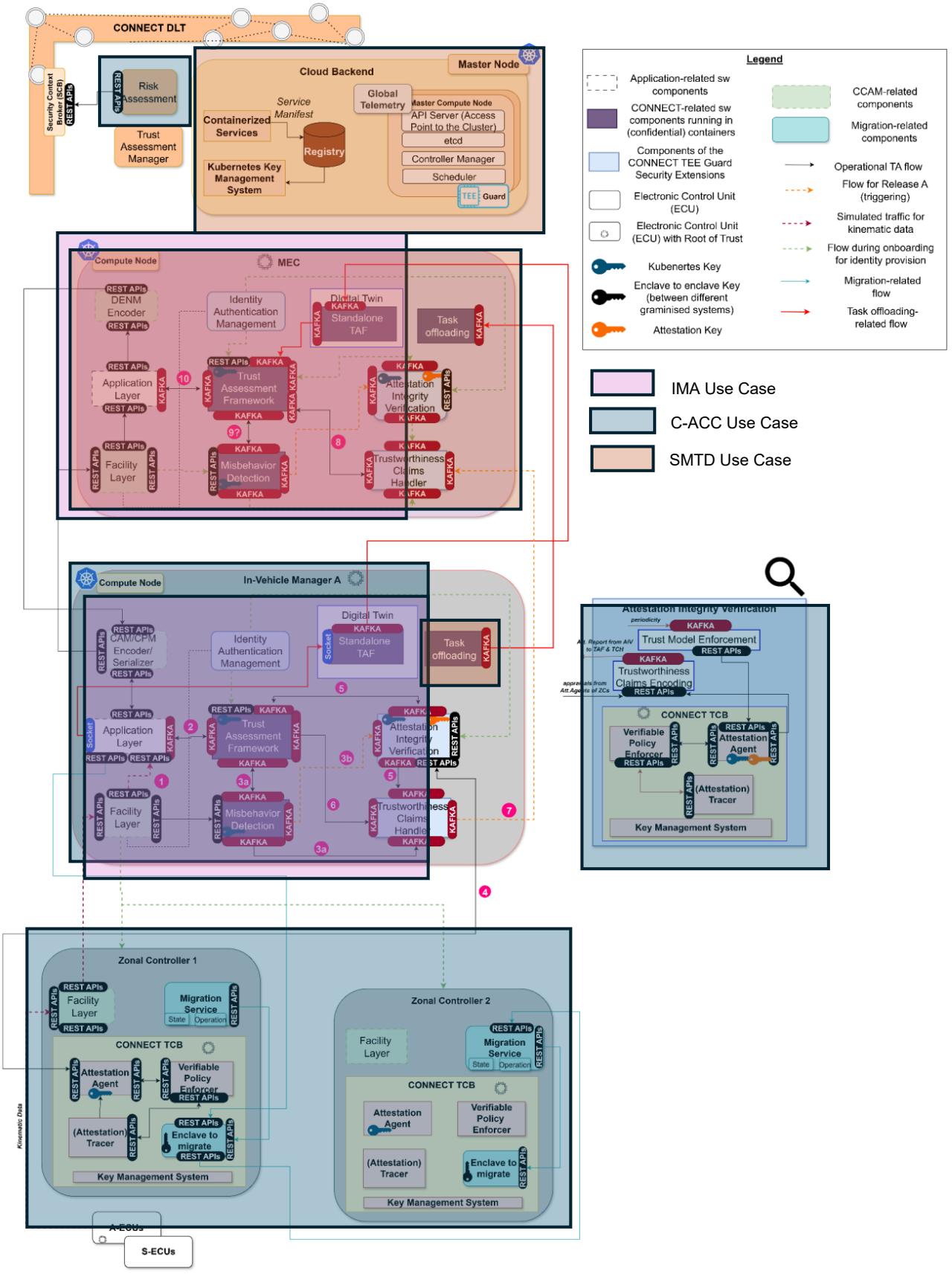


Figure 2.3: *CONNECT* Mapping the Integrated Framework to Use Cases

Chapter 3

Demonstrator #1: Intersection Movement Assistance (IMA) & Misbehaviour Detection (MBD)

Globally, traffic collisions remain among the top causes of mortality, emphasizing the importance of adopting smarter, technology-driven solutions to improve road safety. Vehicle-to-Everything (V2X) systems contribute to this goal by enabling vehicles to exchange real-time data with each other and with the roadside infrastructure, thereby enhancing **situational awareness and supporting safer decision-making on the road**. Applications such as Intersection Movement Assistance (IMA) make use of Cooperative Perception Messages (CPMs) to estimate vehicle trajectories, assess potential collision zones, and issue early warnings that help prevent accidents.

However, the reliability of such applications fundamentally depends on the trustworthiness and integrity of the exchanged data. In complex Cooperative, Connected, and Automated Mobility (CCAM) environments, malicious behavior or compromised entities can degrade the system's effectiveness if not properly detected and handled. This IMA use case focuses on the construction of the vehicle's extended perception map from kinematic data received in V2X messages, showcasing the critical role of dynamic trust evaluation. By continuously assessing the credibility of incoming data, vehicles can avoid acting on misleading or malicious information, thereby reinforcing the overall safety and robustness of CCAM applications that rely on V2X communications. Establishing such a **trust-based decision-making layer is a core enabler for secure CCAM deployments**. In this use case, we consider two scenarios each of which utilises one of the two core modalities of the TAF: the standalone scenario and the federated scenario.

In the standalone scenario, the TAF is deployed on the Ego vehicle and is responsible for evaluating trust opinions on the kinematic data contained in received V2X messages. To perform this assessment, the TAF relies on two types of trust sources: Trustworthiness Claims (TCs) that are received alongside the V2X messages and in-vehicle Misbehaviour Detection (MBD). This scenario serves to explore the interaction between heterogeneous trust sources in the trust assessment process. TCs and MBD differ in two fundamental ways. First, MBD provides evidence directly about the data under evaluation, while the TCs provide evidence regarding the entity that generated the data. Capturing their interplay requires a well-designed trust model and careful selection of fusion operators. Second, the semantics of these sources differ greatly:

negative TC evidence indicates a confirmed compromise of the system, whereas negative MBD evidence merely suggests that the data may be unreliable or misleading. This distinction leads us to recognise the need for the trust model to account for temporal patterns, such as sequences of correlated negative evidence, and highlights the critical role of time dynamics in the trust assessment process.

In the federated scenario, we illustrate the potential for performance improvement that arises from involving the infrastructure in the trust assessment process. In this configuration, the TAF on the Ego vehicle relies on locally generated MBD evidence and trust opinions from the MEC about all vehicles within its coverage area. The vehicle receives V2X messages (which are then used by the local MBD) and messages from the MEC containing these trust opinions. Unlike in the standalone case the Ego vehicle does not receive TCs from the vehicles sending the V2X messages. The MEC's TAF builds its trust opinions based on TCs from the vehicles, which are directly uploaded to the MEC when they send their V2X messages; and on MBD reports, which are uploaded to the MEC by vehicles whenever their respective MBD systems are activated. In this scenario further evidence is supplied by roadside units (RSUs) which also upload MBD reports to the MEC. This architecture enables the infrastructure to gather a broader set of trust evidence than any single vehicle could access independently. Federation is the mechanism that allows vehicles to leverage this richer set of evidence, resulting in a significant improvement in trust assessment accuracy at the vehicle level.

The evaluation of both the standalone and federated scenarios is conducted using a co-simulation platform. Following a thorough analysis of the trust assessment system's performance, we assess each scenario within a small-scale driving scene, that involves a limited number of vehicles following simple trajectories. In this controlled environment, the dynamics of data modification attacks can be precisely modelled, enabling a detailed validation of the trust assessment system within the application context. Subsequently, we extend the evaluation to a large-scale scene, where numerous vehicles undertake long trips on a real road topology, and attacks are generated randomly. This setup allows us to evaluate the trust assessment system within the application, under realistic and dynamic conditions, at a scale that still is impossible to attain in a physical testbed.

Table 3.1 summarizes the differences between the evaluation results reported in Deliverable D6.1 [Con24b] and the final release.

Table 3.1: CONNECT Framework IMA Evaluations between the different Releases

***CONNECT* IMA Evaluations in Release A**	**IMA Evaluations in Final Release of *CONNECT***
In the standalone scenario, evaluation of benchmarks on the TAF of the vehicle, concerning the use of the Trustworthiness Claims (TCs) and of the MBD system trust sources, considered in isolation.	Extension to the federated scenario. Benchmarking of the TAF of the vehicle and of the TAF of the MEC of the trust sources considered in isolation.
	For the standalone and for the federated scenarios, benchmarking of the interactions of the heterogeneous trust sources (TCs and MBD) when used jointly, in the TAF of the vehicle and in the TAF of the MEC.
	Following the introduction of the time-dependence when processing the trust sources, benchmarking of the Temporal-ATL as a trustworthiness level to be compared with the RTL.

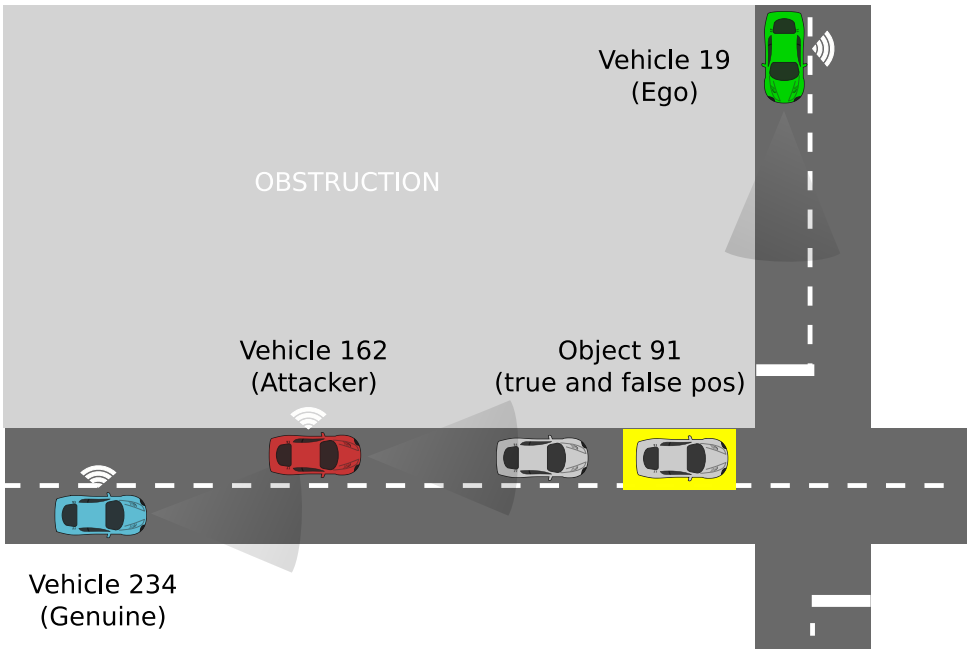


Figure 3.1: Depiction of the standalone scenario, small scale evaluation setting. The Attacker disseminates a falsified position (yellow box) of the Object in its CPMs.

	Evaluation of the trust assessment process within a driving scenario. Accomplished for both the standalone and the federated scenarios through experimentation on a small-scale scene (controlled road topology, a limited number of involved vehicles and for a moderate time duration).
	End-to-end evaluation of the application within a realistic driving scenario. Accomplished both for the standalone and the federated scenarios through experimentation on a large-scale scene (real road topology, a large number of involved vehicles performing random trips on the real road topology and a long time duration).

3.1 Standalone scenario: Perception Object Modification (#2)

3.1.1 Description

This scenario, illustrated in Figure 3.1, replicates the one introduced in Deliverable D6.1 [Con24b]. It involves three communicating vehicles (i.e., vehicles capable of sending V2X messages) and an unequipped vehicle, referred to as the Object vehicle. One of the equipped vehicles acts as the Attacker, maliciously including incorrect kinematic data about the Object vehicle in its Cooperative Perception Messages (CPMs). To illustrate this, the yellow box in Figure 3.1 denotes the false position of the Object vehicle, as disseminated by the Attacker in its CPM. The Ego vehicle utilizes its onboard system to assess the trustworthiness of the received kinematic data and to decide whether to include it in its extended perception, for consumption by the Intersection Movement Assist (IMA) application. The trust assessment, performed by the TAF, is based on two types of trust evidence: the output of the onboard Misbehavior Detection (MBD) system, and the Trustworthiness Claims (TCs) broadcast by the vehicles on the V2V radio interface along with

the transmission of each CPM. (In Deliverable D6.1 [Con24b], this is referred to as transmission of a T-CPM). The TCs contain the harmonized attributes, as defined in Deliverable [Con23b]. The harmonized attributes allow the Ego vehicle to verify that security mechanisms (e.g., secure boot) that enable the bootstrapping of trust in the transmitting vehicle are active on the sender's system.

In this scenario we will verify the ability of trust assessment to mitigate the adverse effects of attacks involving the malicious modification of kinematic data in outgoing CPMs, such as the modification of the position of perceived objects. We will compare the results in the presence of partial and complete trust evidence, and demonstrate the advantage of using multiple trust sources. The primary reason for considering this scenario is in fact to **demonstrate the capability of the trust assessment process to exploit heterogeneous trust sources**. In assessing the trustworthiness of kinematic data, the standalone TAF deployed at the vehicle combines trust evidence relating to the data itself (the MBD trust source), with trust evidence relating to the entity generating the data (the TCs evidence).

In Deliverable D6.1 [Con24b], we successfully validated the behavior of the TAF in response to TCs evidence and MBD evidence separately. However, we were unable to validate it when both trust sources interact. Specifically, the presence of non verifiable attributes in the TCs did not produce the intended trust assessment result when considered in conjunction with the MBD evidence, due to the use of the standard trust discounting operator in the trust calculations. This is detailed in CONNECT Deliverable D6.1 [Con24b], Section 4.4.2. Therefore, the first objective of this evaluation period has been to validate the interaction between TCs and MBD as trust sources, using the newest release of the TAF component, which introduces support for alternative discounting operators (see Deliverable D3.3 [Con25c], Section 4.4.1.)

Beside the fact that one trust source applies to data directly and the other to the entity generating the data, there exists another fundamental difference between the two trust sources. While the impossibility of verifying an attribute included in the TCs constitutes definitive proof of untrustworthiness of the data source, which then propagates to the data, the same cannot be said for MBD evidence, where the activation of a detector may occur due to noise and be a transient, benign phenomenon. In the first case, the temporal persistence of negative trust evidence, such as the persistence of the same non-verifiable attribute in successive TCs from the same vehicle, is trivial and expected. In the second case, however, repeated observation of negative trust evidence, such as the successive activation of a MBD check, is a signal that acquires meaning and should be incorporated into the trust assessment process.

To satisfy this important requirement, during the second evaluation phase the trust assessment framework has evolved to incorporate time-dependence in the trust evaluation process. First, a new feature of the standalone Trust Model instantiated on the vehicle's TAF has been introduced, to apply time-dependence in the evaluation process involving MBD evidence. This feature was validated through experimentation. This approach allows the ATL to gradually decrease when negative MBD evidence starts to be collected, but it has the drawback of making the ATL eventually converge to a plateau, when it is instead crucial that consecutive observations of negative MBD evidence may continuously degrade the ATL.

To address this, we introduce the Temporal-ATL, a trustworthiness level indicator designed to capture the time-dependence of the trust assessment process, which is used to apply RTL filtering rules (defined as a double threshold on belief and uncertainty) with respect to the retention of the kinematic observations in the extended perception. Conceptually, the capability of expressing Temporal-ATLs is a feature of the TAF component. In practice, since the need for this improvement arose during the experimentation phase, this feature was not included in the latest TAF release but will eventually be incorporated in a future release. The development and validation of

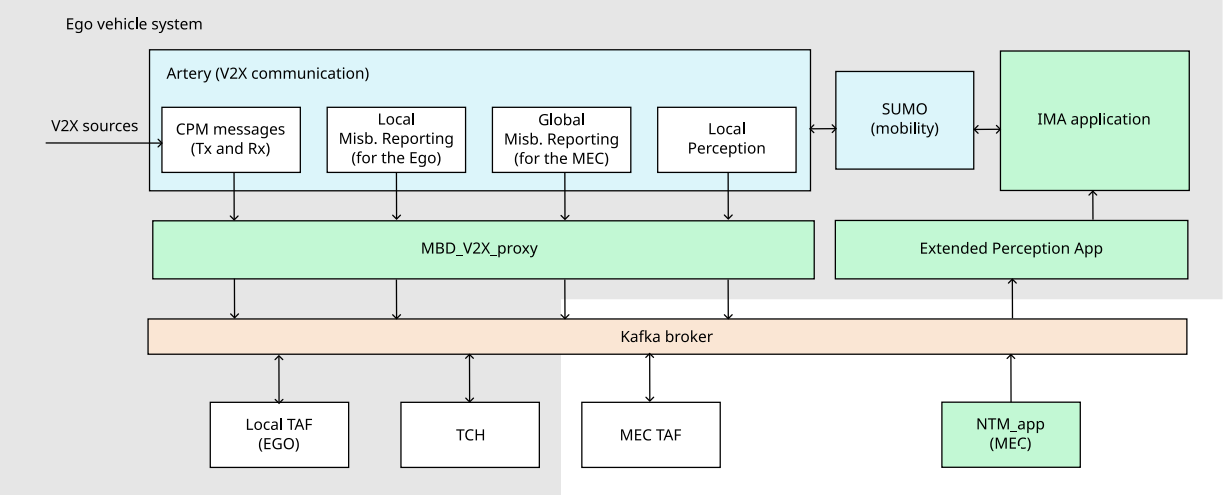


Figure 3.2: Integration of the components of the simulation platform for the IMA use case.

the Temporal-ATL feature was therefore conducted within the context of the IMA use case testbed and are documented in this deliverable.

We document the evaluation activities as follows: first, we will consider several benchmarks to characterize and validate the correct behaviour of the trust assessment within carefully crafted experiments. Initially, we will focus on the interplay between the MBD and TCs as trust sources. Subsequently, we will analyse the time-dependence in the trust evaluation process and validate the Temporal-ATL as an appropriate trustworthiness level indicator for the IMA use case.

Next, we evaluate the system in a small-scale driving scene, where a handful of vehicles interact in an experimentally controlled way. The small-scale road setup is depicted in Figure 3.1. It is used to assess the KPIs defined for the trust assessment system. The limited number of vehicles and the modest duration enable a detailed analysis of the evolution of the trust levels. Finally, we will proceed to experiment in a large-scale scene, with a large number of vehicles driving for an extended period of time, in a realistic road topology. The focus of this evaluation are the KPIs related to the driving application, and in particular to the capability of the trust assessment system to limit the persistence of erroneous kinematic data in the extended perception.

3.1.2 Testbed details

The IMA use case is evaluated using the simulation testbed depicted in Figure 3.2. The components highlighted in grey correspond to the Ego vehicle's system, while the MEC TAF and the NTM application components correspond to the MEC's system. In the standalone scenario, only the Ego vehicle's component are used.

The shared communication bus provides the infrastructure for the integration of the components. The Artery simulation frameworks generates the V2X data relative to the mobility scenario instantiated by SUMO, and the MBD information; this data may be shared over the communication bus thanks to the MBD-V2X-proxy component. For the simulations there is a TCH server that generates the TCs that are associated with the transmission of each V2X message. The generation is triggered by messaging with the MBD-V2X-proxy component. The Local TAF (at the vehicle) consumes V2X messages, MBD reports produced by the vehicle, and TCs of incoming V2X messages and outputs TAS notifications containing the ATLs of the observations contained

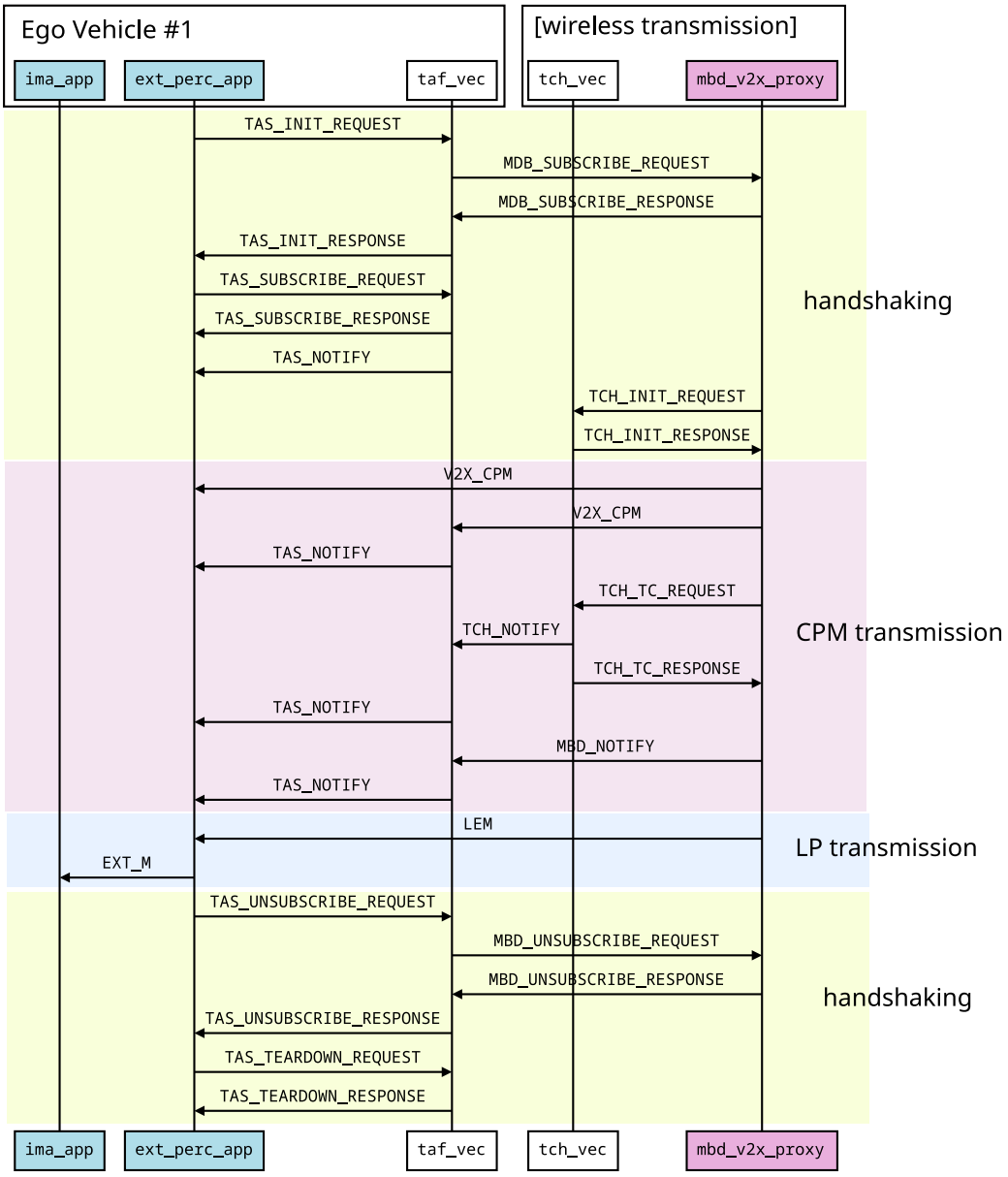


Figure 3.3: Storyline of Standalone scenario (# 2). The purple and blue bands describe the message exchange triggered by the transmission of a single CPM, including the generation of a LP message.

in V2X messages. The Extended Perception component consumes the V2X messages and the ATLs, and feeds the extended perception to the IMA application.

The functional behaviour of the overall architecture considered for the standalone scenario is illustrated in Figure 3.3, which depicts the message flow and interactions. These did not evolve with respect to what has been documented in in Deliverable D6.1 [Con24b].

For clarity, we describe in Table 3.2 the identities and roles of the simulated vehicles that will be used for benchmarking and in the small-scale driving scene depicted in Figure 3.1.

Table 3.2: Participating (simulated) Vehicles in the conducted experiments

Vehicle ID	Role and Description
19	*Ego vehicle*. This is the vehicle under test. It runs the onboard TAF and the onboard MBD system, which activates MBD checks in response to received V2X messages.
91	*Object vehicle*. This vehicle is not V2X-capable and does not broadcast its own kinematic state. Its position is inferred by other vehicles through received CPM messages.
162	*Attacker vehicle*. It behaves as a misbehaving V2X node, sending falsified kinematic data in its CPMs to misrepresent the Object vehicle's position.
234	*Genuine vehicle*. It is a legitimate V2X node broadcasting correct CPMs, used as a reference for expected behavior.

3.1.3 Trust Model

The standalone in-vehicle trust model for the IMA-MBD Scenario 2 is used by the TAF hosted on the ego vehicle. This trust model focuses on assessing the trustworthiness of kinematic data, i.e. observations, contained in the CPMs sent by other, V2X enabled, vehicles in the scenario. There is a trust model instance (TMI) for every vehicle which the ego vehicle receives CPMs from. An example TMI is given in Figure 3.4. The root node, i.e. the agent, in this TMI is the ego vehicle, V_e. The leaf node, i.e. proposition, $C_{x|x}$ represents the observation of the sender vehicle V_x on itself, which is always found in a CPM message. The leaf node, i.e. the proposition, $C_{x|y}$ represents the observation sent by vehicle V_x about perceived vehicle V_y. There could be additional propositions, i.e. trust objects, representing observations about other vehicles as reported in the CPMs sent by the vehicle V_x. Specifically, every vehicle observed by the sender and reported as an observation in a CPM, will have its own trust object in the TMI. This is the dynamic component of this trust model which is managed by the TAF at run-time and is not portrayed in the Figure 3.4. Finally, in this Trust Model Instance, there is always a trust object representing the sender vehicle V_x. (Note that observations are defined here as vehicle kinematic data, such as speed and position.)

There are trust relationships between all trust objects in this trust model instance and these trust relationships are quantified in form of trust opinions, ω_Y^X, where X represents the trustor, i.e. the trust object which assesses the trust opinion, and Y represents the trustee, i.e. the trust object whose trustworthiness is being assessed.

Consider the trust model depicted in Figure 3.4 in the context of the small-scale scene of Figure 3.1. The red node V_e is the Ego (Vehicle 19). The blue node V_x corresponds to the Attacker (Vehicle 162). The node $C_{x|x} = C_{162|162}$ represents the kinematic observation that Vehicle 162 sends about itself in the station container of its CPM; the node $C_{x|y} = C_{162|91}$ represents the observation of Object 91 that Vehicle 162 includes in the perceived object container of its CPM.

3.1.3.1 Trust Opinions

There are the following trust opinions in this TMI:

1. $\omega_{V_x}^{V_e}$ = the trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **vehicle** V_x to send messages (CPMs) whose integrity has not been compromised.

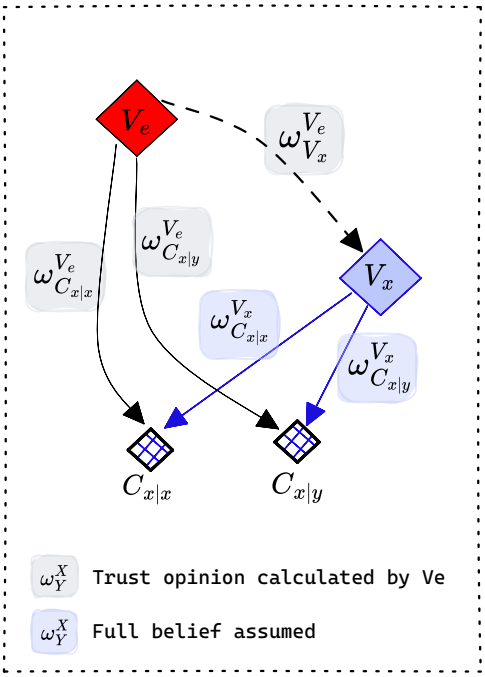


Figure 3.4: IMA-MBD Scenario 2 - Trust Model for the standalone TAF on the ego vehicle

2. $\omega^{V_e}_{C_{x|x}}$ = the trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **observation about the vehicle** V_x, contained in the most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.

3. $\omega^{V_e}_{C_{x|y}}$ = trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **observation about the vehicle** V_y, contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.

4. $\omega^{V_x}_{C_{x|x}}$ = the trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about itself**, contained in the most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.

5. $\omega^{V_x}_{C_{x|y}}$ = the trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about the vehicle** V_y, contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.

3.1.3.2 Trust Sources and Evidence

Given that this is a standalone TAF, it does not request trust opinions from external entities. The standalone TAF, however receives evidence from external trust sources, such as another vehicle. What follows is a list of all the different trust sources and evidence to be collected for each trust opinion in the trust model. Moreover, a trust quantification approach is given for each evidence.

1. $\omega^{V_e}_{V_x} \rightarrow$ the **ego vehicle** assesses this opinion based on the evidence which it receives from the sender vehicle about its own trustworthiness in form of verifiable presentations.

- **Trust Source** = TCH
- **Evidence** = Verifiable presentations - Verifiable presentations include the output of the single security controls of the vehicle computer of the sending vehicle if the security controls are not implemented (value: -1), are implemented and detected something (value: 0) or are implemented and did not detect anything (value: 1)
- **Trust Quantification Approach** = The approach for calculating the trust opinion based on the verifiable presentations is described in D6.1 [Con24b] and D3.3 [Con25c].

2. 2. $\omega^{V_e}_{C_{x|x}} \rightarrow$ the **ego vehicle** assesses this opinion based on the results of the Misbehavior Detection system running on the ego vehicle
   - **Trust Source** = MBD system on ego vehicle
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
3. $\omega^{V_e}_{C_{x|y}} \rightarrow$ the **ego vehicle** assesses this opinion based on the results of the Misbehavior Detection system running on the ego vehicle
   - **Trust Source** = MBD system on ego vehicle
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
4. $\omega^{V_x}_{C_{x|x}}$ = the ego vehicle assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.
5. $\omega^{V_x}_{C_{x|y}}$ = the ego vehicles assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.

3.1.3.3 Isolation-Actual Trustworthiness Level

$ATL_{C_{x|x}}$ is obtained in two steps: 1) first, $\omega^{V_e}_{V_x}$ and $\omega^{V_x}_{C_{x|x}}$ are discounted to obtain $\omega^{V_e;V_x}_{C_{x|x}}$, 2) then the resulting opinion $\omega^{V_e;V_x}_{C_{x|x}}$ is fused with the direct opinion $\omega^{V_e}_{C_{x|x}}$ to obtain the ATL. We are using the opposite-belief trust discounting and the cumulative fusion operators in this example.

$$ATL_{C_{x|x}} = \omega^{V_e}_{C_{x|x}} \oplus (\omega^{V_e}_{V_x} \otimes \omega^{V_x}_{C_{x|x}}) \tag{3.1}$$

Similarly, $ATL_{C_{x|y}}$ is also obtained in two steps: 1) first, $\omega^{V_e}_{V_x}$ and $\omega^{V_x}_{C_{x|y}}$ are discounted to obtain $\omega^{V_e;V_x}_{C_{x|y}}$, 2) then the resulting opinion $\omega^{V_e;V_x}_{C_{x|y}}$ is fused with the direct opinion $\omega^{V_e}_{C_{x|y}}$ to obtain the ATL. Once again, we are using the opposite-belief trust discounting and the cumulative fusion operators for this purpose.

$$ATL_{C_{x|y}} = \omega^{V_e}_{C_{x|y}} \oplus (\omega^{V_e}_{V_x} \otimes \omega^{V_x}_{C_{x|y}}) \tag{3.2}$$

3.1.4 User Story Realisation

The realisation of the following User Stories is considered in the evaluation of the standalone scenario.

[MB.US1] As the IMA application on the Vehicle, I want to be able to consume a consolidated view of the scene containing trustworthy data.

[MB.US2]: As the Vehicle I want to be able to extract an observation contained in a CPM, attribute it a Trustworthiness Level and record it in the LDM, so that it can be appropriately included in the extended perception, according to the considered RTL.

In this chapter we document the realisation of user story MB.US2 in the standalone scenario. Its evaluation is performed using the small-scale scene, as documented in Section 3.1.6.5, for two different patterns of trust evidence consistent with the perception object modification attack. Relevant to this US, benchmarking of the interaction of the trust sources is provided in Sections 3.1.6.1 and 3.1.6.2; benchmarking of the temporal evolution of the trustworthiness level as a response to the correlation in time of the collected trust evidence is provided in Sections 3.1.6.3 and 3.1.6.4.

We also document the realisation of user story MB.US1 in the standalone scenario. Its evaluation is performed using the large-scale scene, as documented in Section 3.1.6.6.

3.1.5 KPI & Acceptance Criteria

D2D2

Table 3.3: Evaluated KPIs by user stories, in the standalone scenario.

User story	KPI description	Acceptance criteria	Results
MB.US1	The consolidated view of the scene contains trustworthy data	Only sufficiently trustworthy data is used to produce the extended perception	Validated and documented in Section 3.1.6.6, with respect to different RTL values.
MB.US.2	Processing complexity until LDM update	$\leq$ **100 ms** from the reception of a new CPM and V-TC and update of the LDM table (kinematic information and ATL).	Documented in Deliverable D6.1 [Con24b]
	The ATL expressed on observations of the same physical object performed by the same V2X-node evolves correctly	When the opinion on the active V2X-node degrades, the ATL on all the observations provided by the active V2X-node degrades	Because of the evolution of the architecture since Deliverable D2.1, the opinion on the active node cannot be directly observed by the vehicle, as it is hidden inside the TAF. However, the degradation of the opinion on the vehicle may be indirectly subsumed by the reception of negative TCs evidence. This is validated and documented in Section 3.1.6.1.

		When the Local Misbehaviour Report contains an active detector on the observation of the physical object performed by the active V2X node, the ATL on the observation degrades.	Validated and documented in Sections 3.1.6.1 and 3.1.6.2.
MB.US4	The opinion of the active V2X-node evolves correctly	When non verifiable attributes are contained in the V-TCs generated by a V2X-node the opinion on the active V2X-node, recorded in the Trust Model hosted by the receiver's TAF, degrades	Because of the evolution of the architecture since Deliverable D2.1 [Con23c], the opinion on the active node cannot be directly observed by the vehicle, as it is hidden inside the TAF.

3.1.6 Evaluation of TAF behavior based on different Trust Sources

3.1.6.1 Trust sources interplay validation

In this scenario of the IMA use case the TAF on the Ego vehicle is tasked with expressing ATLs on incoming kinematic observations. To do so, it processes two types of trust evidence, namely Misbehaviour Detectors (MBD) on the kinematic observation itself; and the Trustworthiness Claims (TCs) sent by the generating vehicle along with the V2X message containing the observation. The TCs, generated by the TCH of the transmitting vehicle and included in outgoing CPMs, contain the harmonized attributes, reporting on how components of the vehicle contribute ti the trustworthiness of transmitted data. More specifically, TCs relate to six attributes:

- Secure Boot
- Application Isolation
- Control Flow Integrity
- Access Control
- Secure OTA
- Configuration Integrity Verification

In the scope of the experiments carried on in this evaluation period, we assume that either all the harmonized attributes in the TCs can be verified (*verifiable TCs*), or none can (*unverifiable TCs*).

As documented in Deliverable 6.1 [Con24b], Section 4.4.2 the major obstacle in the previous evaluation period has been the incorrect behaviour of the TAF with respect to the interplay of the two trust sources, in the context of the IMA use case needs. The issue has been addressed by introducing support for an alternative discounting operator in the TLEE, as documented in Deliverable 3.3 [Con25c].

This section then illustrates the validation of the intended behaviour of the TAF of the Ego vehicle. We consider the ATL on a received observation, evaluated by the TAF using MBD and TCs from the vehicle generating the observation, with respect the following benchmarking situations (already considered in Deliverable D6.1).

Table 3.4: Benchmarking situations.

Name	Description
Clean-MBD-Valid-TCs	ATL of the observation, based on both trust sources; the MBD does not produce detector activations; the TCs of the generating vehicle allow verification of all the harmonized attributes
Clean-MBD-Invalid-TCs	ATL of the observation, based on both trust sources; the MBD does not produce detector activations; the TCs of the generating vehicle are invalid and the harmonized attributes cannot be verified
Active-MBD-Valid-TCs	ATL of the observation, based on both trust sources; the MBD produces some detector activations; the TCs of the generating vehicle allow verification of all the harmonized attributes
Active-MBD-Invalid-TCs	Reaction of the *Ego* system when TCs are used as a trust source as well as MBD; on some observations, the MBD produces MRs with detector activation; the TCs are invalid and the harmonized attributes cannot be verified

We recall that this benchmarking evaluation failed for the ”Clean-MBD-Invalid-TCs” and ”Active-MBD-Valid-TCs” benchmarks during the last evaluation period (see CONNECT Deliverable D6.1, Section 4.4.2).

In order to validate the benchmarks in Table 3.4 we run two experiments and we track in time the ATL produced by Vehicle 19 (Ego) on the observations of Vehicle 91 (the Object) transmitted in the CPMs of Vehicle 162 (the Attacker). The ATL is a subjective logic vector; we track the values of the Belief and of the Uncertainty components. These are the values that will be compared with the RTL when deciding whether to retain kinematic observations in the extended perception, for consumption by the IMA application.

The setup of the experiments is summarized by the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91, and sends its own TCs. **Ego (Vehicle 19)**: Receives CPMs and TCs from the Attacker (Vehicle 162). Runs MBD checks on each observation of 91 sent by 162. Its TAF produces an ATL on each received observation of 91.
Trust sources	Both MBD and TCs trust evidence is available at the Ego’s TAF.
Considered metric	Evolution in time of the ATL of the observation of 91. The ATL is a subjective logic vector, consider Belief and Uncertainty components.
Expected outcome	The Belief is maximized when both the MBD and the TCs evidence is positive; the Belief heavily degrades when TCs cannot be verified, irrespective of the MBD evidence; the Belief degrades when the MBD evidence is negative and TCs evidence is positive. The Uncertainty is small if trust evidence is observed, regardless of its value.

Exp. 1: attack pattern	The Attacker (162)'s TCs can always be verified. Observations of 91 are inconsistent during intervals 15-20 seconds and 30-40 seconds; this triggers the repeated activation of MBD Check 1. In the remaining intervals no attack is carried on and no MBD check is activated. The results of Experiment 1 are depicted in Figure 3.5.
Exp. 2: attack pattern	The Attacker (162)'s TCs are always valid before second 22 so that all harmonized attributes are verified. From second 22 onwards, TCs became invalid, so that the harmonized attributes can never be verified. Moreover, observations of 91 are incosistent during the same intervals of Experiment 1, triggering the activation of the same MBD checks. The results of Experiment 2 are depicted in Figure 3.6.

As the following analysis shows, the two experiments allow all of the benchmarks to be correctly verified, including the "Clean-MBD-Invalid-TCs" and "Active-MBD-Valid-TCs" benchmarks which were unsuccessful during the previous evaluation period (see CONNECT Deliverable D6.1, Section 4.4.2).

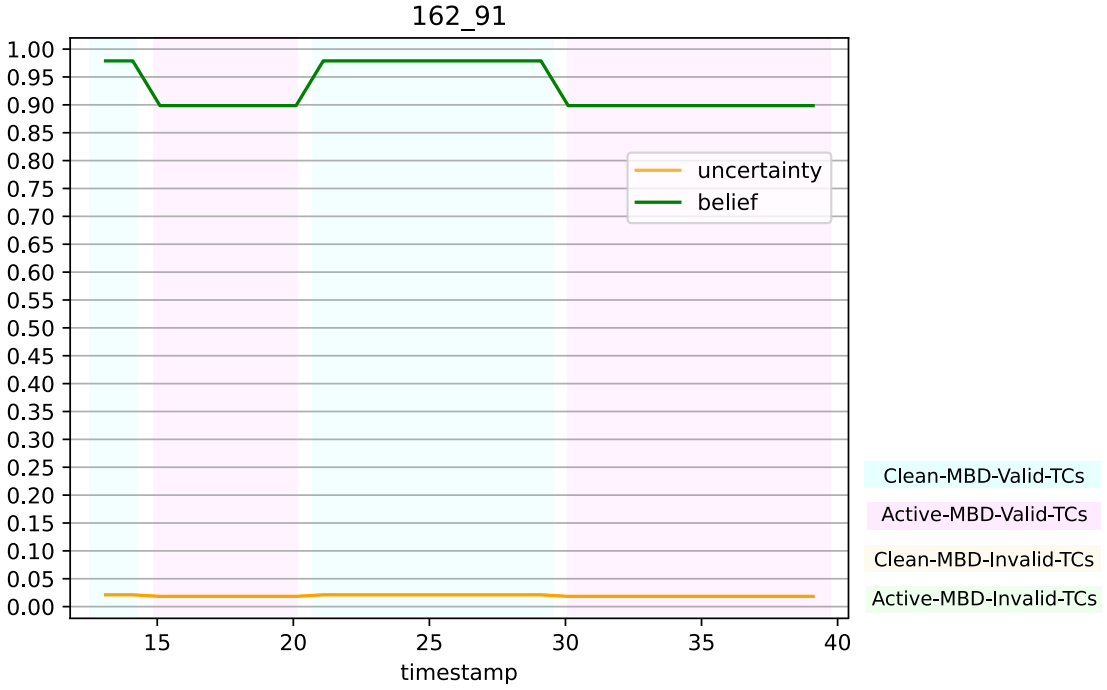


Figure 3.5: Evaluation of the Clean-MBD-Valid-TCs and Active-MBD-Valid-TCs benchmarks.

Clean-MBD-Valid-TCs: To verify this benchmark we consider Experiment 1, whose results are depicted in Figure 3.5. We are interested in the intervals during which the Attacker (162) sends valid TCs and no MBD detector is activated, which are highlighted in light blue in Figure 3.5. As expected, the Belief component of the ATL has a high value, and the Uncertainty component a very small value.

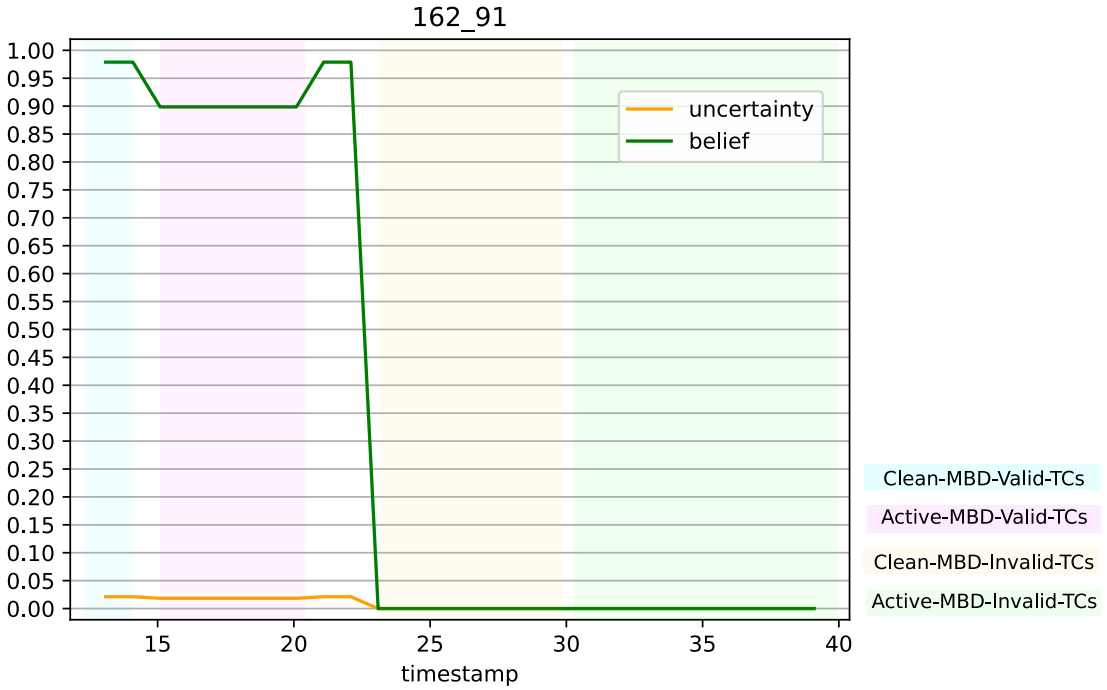


Figure 3.6: Evaluation of the Clean-MBD-Invalid-TCs and Active-MBD-Invalid-TCs benchmarks.

Active-MBD-Valid-TCs: To verify this benchmark we consider Experiment 1, whose results are depicted in Figure 3.5. We are interested in the intervals during which the Attacker (162) sends valid TCs and performs data modification attacks on the observations of 91. These intervals are marked in light violet in Figure 3.5. As expected, the Belief component decreases during these intervals, as effect of the processing of the activation of the misbehaviour detectors.

Clean-MBD-Invalid-TCs: To verify this benchmark we consider Experiment 2, whose results are depicted in Figure 3.6. We are interested in the interval during which the Attacker (162) sends invalid TCs and no MBD detector is activated, which is marked by the yellow band in Figure 3.6 (seconds 22-30). As expected, the Belief dramatically drops, even in absence of negative MBD evidence. Notice that the drop in belief is much more dramatic than what happens in case of MBD activation (light violet interval). This is intentional, and corresponds to the fact that the impossibility of verification of TCs constitutes a major indication of departure from nominal conditions of the processes responsible of V2X data extraction and transmission on the sender's system.

Active-MBD-Invalid-TCs: To verify this benchmark we consider Experiment 2, whose results are depicted in Figure 3.6. The interval in light green (from second 30 onwards) correspond to TCs which cannot be verified, and simultaneous activation of the same MBD detector on all observations from Vehicle 162. As expected, the Belief keeps its minimum value in this interval, as effect of the impossibility of verification of the TCs, which is the predominant factor.

3.1.6.2 Benchmarking MBD as a trust source

In this Section we benchmark the impact of MBD as a trust source, and in particular we assess the impact of the activation of individual detectors on the ATL on the observation.

As detailed in Deliverable 3.2, this is the list of misbehavior detectors that can be activated by the Ego system:

1. distPlau (position plausibility)
2. SpeedPlau (speed plausibility)
3. SpeedCons (speed consistency)
4. PosSpeedCons (position consistency)
5. KalmanPosCons (position consistency)
6. KalmanPosSpeedConsS (speed consistency)
7. KalmanPosSpeedConsP (position consistency)
8. LocalPerceptionVerif (local perception consistency)

To assess the impact of each individual detector's activation, we conducted an experiment in which a sequence of attacks triggers different MBD activation patterns for the same observation. The setup of the experiment is summarized in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91, and sends its own TCs. **Ego (Vehicle 19)**: Receives CPMs and TCs from the Attacker (Vehicle 162). Runs MBD checks on each observation of 91 sent by 162. Its TAF produces an ATL on each received observation of 91.
Trust sources	Both MBD and TCs trust evidence is available at the Ego's TAF.
Considered metric	Evolution in time of the ATL of the observation of 91. The ATL is a subjective logic vector, consider Belief and Uncertainty components.
Exp: attack pattern	The Attacker (162)'s TCs can always be verified. The observations of 91 are modified so that a different MBD check activation pattern is triggered at each reception. We can distinguish two series of attacks: during the first series (15-23 seconds) the activation pattern contains one MBD Check at a time; during the second series (23-34 seconds) the activation pattern contains two or more MBD Checks at a time. The results of the experiment are depicted in Figure 3.7. For each received CPM, Figure 3.7 indicates, in red, the identities of the activated MBD Checks.

Single-activation benchmark: This benchmark is verified considering the first attack series in the experiment (seconds 15-23), whose results are depicted in Figure 3.7. This interval involves the activation of a single detector on the received observation. It has the purpose

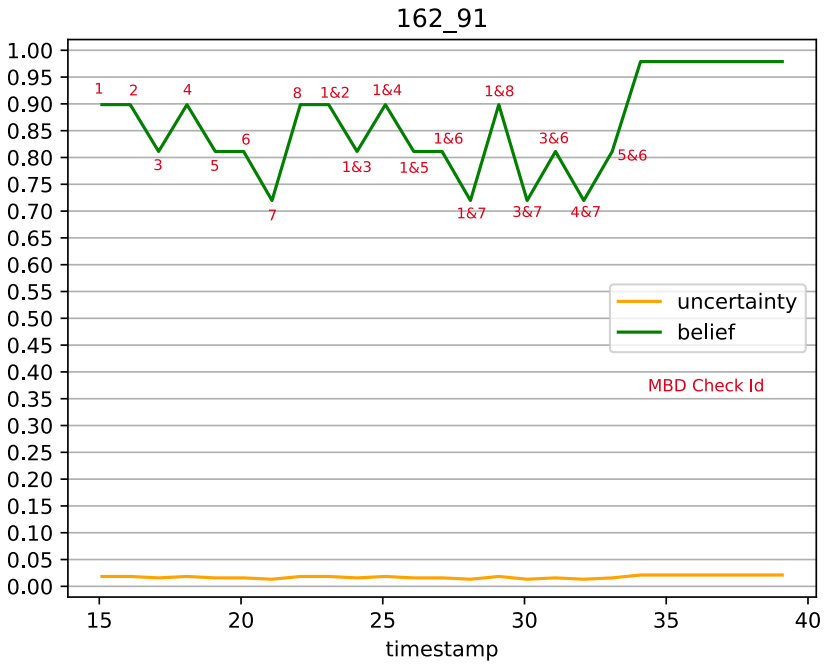


Figure 3.7: Benchmarking of the ATL attributed by the TAF to an observation, based on MBD trust evidence.

of establishing the baseline of the impact of the activation of a single detector on the ATL. As expected, check number 7 is the one providing the heaviest degradation on the ATL.

Multi-activation benchmark: This benchmark is verified considering the first attack series in the experiment (seconds 15-23), whose results are depicted in Figure 3.7. This interval is visible between 23 and 35 seconds and involves the activation of two detectors, simultaneously. As it is visible, whenever multiple detectors are activated the effect on the ATL is the same as if only the most severe of the two was activated.

The analysis brought forward the challenge of selecting an appropriate RTL for retaining observations in the IMA application's extended perception. RTL is defined via a dual-threshold system based on Belief and Uncertainty. Setting the Belief threshold above 0.9 excludes any observation flagged by any single detector, effectively mirroring conservative filtering. Lower thresholds retain observations which trigger less severe detectors, this might be episodically acceptable as explained by noise fluctuations, but fails to account for repeated activations that may indicate an ongoing attack. Evidently, to overcome this, the trust model should incorporate a temporal dimension. This dimension enables it to consider patterns of detector activations over time. Further analysis is provided in Section 3.1.7.

3.1.6.3 Testing the temporal dimension in MBD trust evidence

The temporal dimension in processing the MBD trust evidence has been introduced in the trust model for the IMA use case, as detailed in Deliverable 3.3 [Con25c]. Whenever fresh MBD evidence is made available for a known trust object, its atomic opinion resulting from MBD evidence is updated taking into account the previous value. This happens using an exponentially weighted

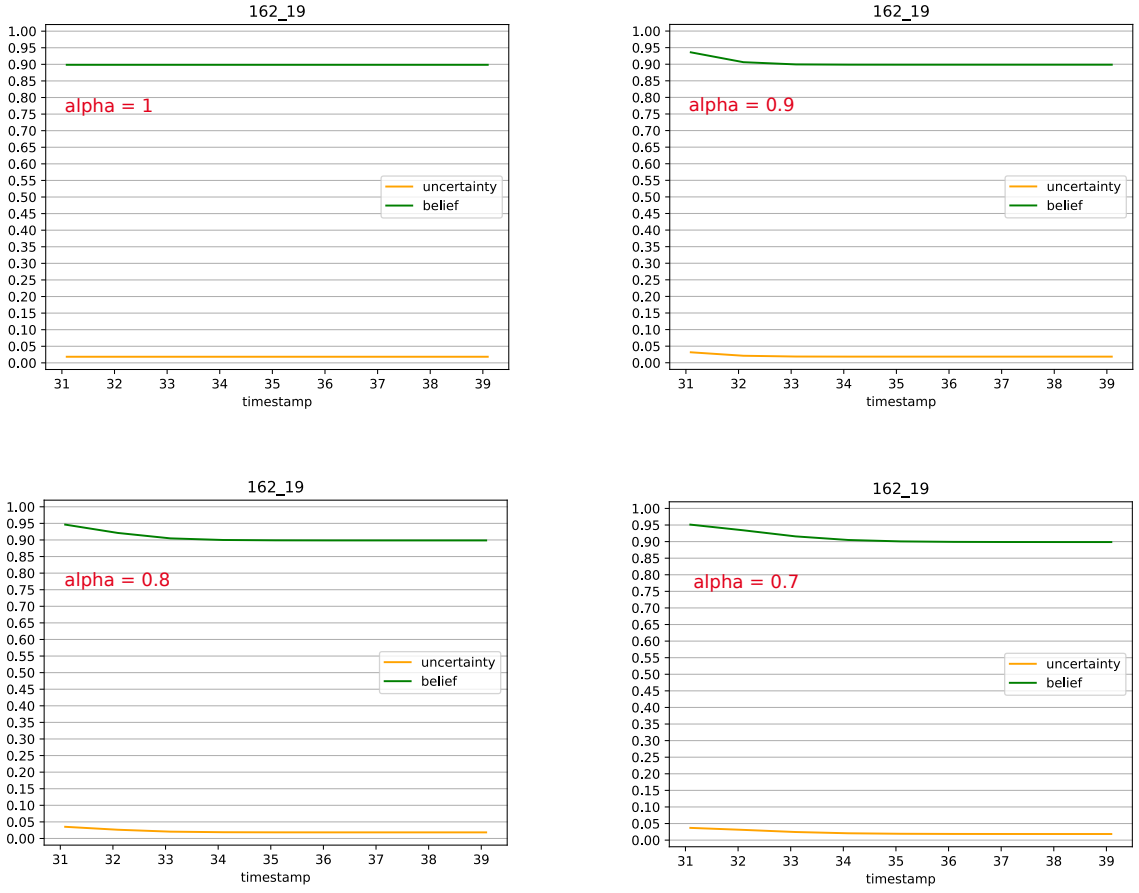


Figure 3.8: Benchmarking of temporal dimension in the usage of the misbehaviour detection trust evidence.

moving average, where the parameter α controls the influence of the previous atomic opinion on the new one. In this section we analyse the impact of adding this temporal dimension to the ATL evaluation.

To do so, we consider an experiment in which the same MBD check is activated repeatedly on the same observation over time. The setup of the experiment is summarized in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Vehicle 19, and sends its own TCs. **Ego (Vehicle 234)**: Receives CPMs and TCs from the Attacker (Vehicle 162). Runs MBD checks on each observation of 19 sent by 162. Its TAF produces an ATL on each received observation of 19.
Trust sources	Both MBD and TCs trust evidence is available at the Ego's TAF.
Considered metric	Evolution in time of the ATL of the observation of 19. The ATL is a subjective logic vector, consider Belief and Uncertainty components.
Expected outcome	Upon repeated observation of the activation of MBD checks, the Belief component of the ATL degrades. The higher the number of consecutive MBD activations, the lower the value of the Belief component output by the TAF.

Exp: attack pattern	The Attacker (162)'s TCs can always be verified. The observations of 19 are modified so that MBD Check 1 is always activated, from second 31 onwards. The results of the experiment are depicted in Figure 3.8, for varying values of the parameter α. The value $\alpha = 1$ corresponds to the case of no time-dependence.

Successive-activation benchmark Figure 3.8 illustrates the experiment for varying values of the parameter α. For $\alpha = 1$ no memory is incorporated, and all successive MBD activations are considered in isolation and produce the same Belief and Uncertainty values. For $\alpha = 0.9$ one can observe that some inertia is introduced, so that the Belief gently decreases from the full value as soon as the first MBD check is activated. Further decreasing α slows down the descent.

The tests in Figure 3.8 show the ability of the TAF to react to the temporal repetition of the activation of the misbehaviour detector, as well as the ability of tuning the rapidity of the reaction thanks to the tuning of the parameter α. Notice however that upon repetition of the activation of the same detector, the value of the ATL converges to the value obtained for a single activation with $\alpha = 1$. This is not entirely the expected outcome, because, as shown in Figure 3.8, even though there are repeated observations of the same MBD detector, the value of the Belief component converges to a pretty high value.

The temporal dimension for the MBD trust source provides some inertia in the evolution over time of the Belief component in the ATL. This is positive because it allows to set a RTL threshold on the Belief which encodes some tolerance for episodic activations of detectors, while correctly catching on the repetitive pattern.

However, the analysis puts into perspective another issue, which is the fact that the Belief component of the ATL converges to a fixed bound (corresponding to the value for $\alpha = 1$) in case of repeated activations. This is not entirely satisfactory for the use case needs, where one would expect continuous degradation of the Belief component of the ATL for repeated MBD checks activations.

3.1.6.4 Testing Temporal-ATL

In the previous round of benchmarking, our primary concern was determining how to establish meaningful RTL levels for the incoming kinematic data. The results suggest that this is a straightforward task if the influence of the TCs trust source is considered, which leads the Belief component to immediately drop to zero upon receiving negative evidence. However, this becomes considerably more challenging when negative evidence is exclusively provided by the MBD trust source.

As argued, processing of the time-dependence is crucial for handling the MBD trust source. Although the current TAF release supports some form of time-dependence with the MBD trust source, the results indicate that the ATL struggles to differentiate between a single anomaly and a sustained attack pattern. Specifically, the Belief component of the ATL does not present a monotonous degrading pattern as the consequence of a prolonged sequence of anomalous events, where repeated activations over time become more and more indicative of malicious activity.

In the context of the IMA use case, trust decisions must account for this temporal context to produce meaningful results. For this reason, we introduce a new trustworthiness level indicator, more able to provide tolerance in the case of a single isolated activation, but that reacts quickly when there are repeated activations.

The new **Temporal-ATL** is defined as a subjective logic opinion. As discussed in Section 3.1.1, the Temporal-ATL is not supported by the current release of the TAF. As a consequence, we define it here with respect to the output of the current release of the TAF, which we call Isolation-ATL, since its evaluation depends on the freshest trust evidence pieces, only.

Consider the Isolation-ATL (in the form of subjective logic opinion), with $\alpha = 1$, on a specific observation. The Temporal-ATL is a new subjective logic opinion, and is defined as follows. Let $\omega = (b, d, u, a)$ be the Isolation-ATL on the observation at instant t, and let $\overline{\omega} = (\overline{b}, \overline{d}, \overline{u}, \overline{a})$ denote the corresponding Temporal-ATL. One has the following:

$$\overline{u} = u, \qquad \overline{a} = a. \tag{3.3}$$

At the instant t, the components $\overline{b}_t$ and $\overline{d}_t$ are evaluated as functions of b_t and d_t, as well as functions of $\overline{b}_{t-1}$ and $\overline{d}_{t-1}$. Let τ be a threshold so that it exceeds the b value generated as a consequence of any MBD check activation, but below the value of b for no MBD activation. Then:

$$\overline{b}_t = \overline{b}_{t-1} - \gamma(\tau - b_t), \quad \text{if} \quad 0 \leq \overline{b}_{t-1} - \gamma(\tau - b_t) \leq 1 - \overline{u}_t \tag{3.4}$$

$$\overline{b}_t = \overline{b}_{t-1}, \quad \text{otherwise.} \tag{3.5}$$

where $0 \leq \gamma \leq 1$ is a weighting factor.

This means that $\overline{b}_t$ is obtained correcting the previous Temporal-Belief by a quantity depending on the gap between the reference value τ and the actual value of b_t of the Isolation-ATL (which depends on the identity of the last detector that has been activated). The gap will be positive in case of MBD activation, its entity depending on the severity of the activated detector; this will have the effect of decreasing $\overline{b}_t$ with respect to $\overline{b}_{t-1}$. In case of no activation, the gap takes a negative value, allowing $\overline{b}_t$ to increase with respect to the previous step.

The value of $\overline{d}_t$ is finally defined as:

$$\overline{d}_t = 1 - \overline{u}_t - \overline{b}_t. \tag{3.6}$$

To validate the Temporal-ATL, we consider two experiments. In the first the same MBD detector is repeatedly observed in time; in the second the MBD activation pattern varies. The setup of the experiments is summarized in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91 and of Vehicle 19, and sends its own TCs. **Ego (Vehicle 19)**: Receives CPMs and TCs from the Attacker (Vehicle 162). Runs MBD checks on each observation of 91 sent by 162. Its onboard system produces an ATL and a Temporal-ATL on each received observation of 91. **Ego (Vehicle 234)**: Receives CPMs and TCs from the Attacker (162). Runs MBD checks on each observation of 19 sent by 162. Its onboard system produces an ATL and a Temporal-ATL on each received observation of 19.

Trust sources	Both MBD and TCs trust evidence is available at the Egos's TAF.
Considered metric	Evolution in time of the Isolation-ATL and of the Temporal-ATL. The Isolation-ATL is a subjective logic vector, consider Belief and Uncertainty components. The Temporal-ATL is a subjective logic vector, consider the Temporal-Belief (the Temporal-Uncertainty is always equal to the Uncertainty component). The Temporal-ATL is obtained for parameters $\tau = 0.93$ and $\gamma = 0.75$
Expected outcome	Upon repeated observation of the activation of MBD checks, the Temporal-Belief degrades. The higher the number of consecutive MBD activations, the lower the value of the Belief component output by the TAF.
Exp 1: attack pattern	The first experiment concerns Ego 234. The Attacker (162)'s TCs can always be verified. The observations of 19 are modified by the Attacker, so that MBD Check 1 is always activated, from second 31 onwards. The results of the experiment are depicted in Figure 3.9.
Exp 2: attack pattern	The second experiment concerns Ego 19. The Attacker (162)'s TCs can always be verified. The observations of 91 are modified by the Attacker between seconds 15 and 35, so that various MBD checks are activated in succession. The results of the experiment are depicted in Figure 3.10.

Same MBD-check Temporal-ATL benchmark To validate this benchmark we consider the first experiment, whose results are depicted in Figure 3.9. This illustrates how the repeated activation of the same detector, however producing mild effects on the Isolation-Belief, contributes to the steady degradation of the Temporal-Belief.

Varying MBD-check Temporal-ATL benchmark To validate this benchmark we consider the second experiment, whose results are depicted in Figure 3.10. Object 91's observations activate various MBD checks between seconds 15 and 35. Here, the Temporal-Belief component is monotonously decreasing during the attack interval, as opposed to the Isolation-Belief, which oscillates depending on the severity of the last-activated MBD check. Notice that the decrease of the Temporal-Belief as a result of new evidence is proportional to the severity of the last-activated MBD check. After 14 seconds from the start of the attack, the Temporal-Belief reaches value 0. At the end of the experiment, from second 35 onwards, no MBD checks are activated anymore. This corresponds to a gradual increase of the Temporal-Belief. This shows that eventually, if no negative evidence is collected during a sufficiently long interval, the Temporal-ATL may recover. This is a desirable behaviour, useful in case of perturbations due to a transient cause, as object masking.

The conclusion is that since it is impossible to optimize the RTL threshold to tailor any possible driving scenario or attack pattern, using the Temporal-Belief we are at least confident that the RTL will be able to catch on repeated MBD checks activations, which are a marker of attacks. Therefore, the performance of the IMA application becomes much less sensitive to the choice of the RTL.

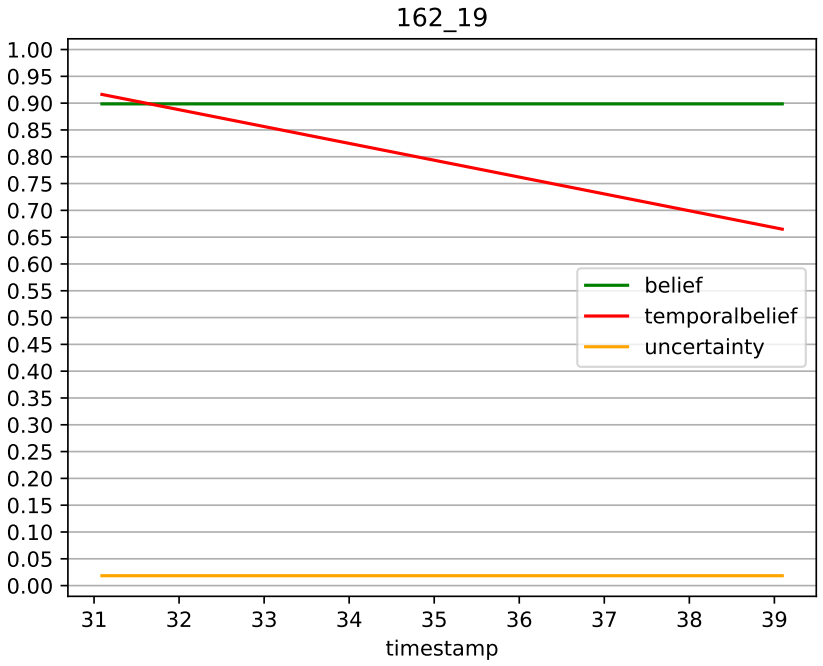


Figure 3.9: First benchmark of Temporal-ATL.

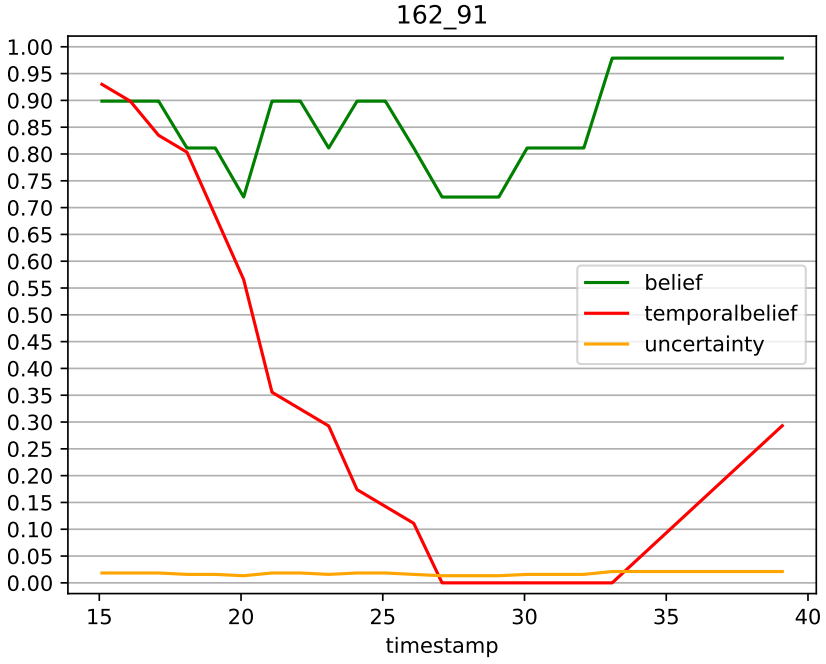


Figure 3.10: Benchmarking of temporal belief.

3.1.6.5 Small-scale scene realisation

After the benchmarking phase, we now focus on the user stories realisation, trough experimentation in driving scenarios. We begin with a small-scale scene, which allows to thoroughly inspect the outcomes to validate the intended behaviour and the KPIs belonging to the trust assessment.

The considered driving scenario is the T-junction depicted in Figure 3.1, involving three connected vehicles: an Attacker (Vehicle 162), the Ego (Vehicle 19) and a Genuine (Vehicle 234). The Object (Vehicle 91) is non connected. The detailed description of the entities involved is in Table 3.2. The Attacker performs a data modification attack by sending the erroneous position for Object 91 in its CPMs.

Table 3.9: Events in the considered storyline.

Time	Event description	Highlight
12.689 s	Vehicle 19 (Ego) receives the first CPM from Vehicle 234 (Genuine)	
13.093 s	Vehicle 19 (Ego) receives the first CPM from Vehicle 162 (Attacker)	Attack has not started yet.
16.093 s	Vehicle 19 (Ego) receives the first CPM from Vehicle 162 (Attacker) containing an attack (Series 1)	Start of Series 1 of attacks. Vehicle 19 (Ego) activates a Misbehaviour Detector.
25.093 s	Vehicle 19 (Ego) receives the last Series 1 CPM from Vehicle 162 (Attacker)	End of Series 1 of attacks. Vehicle 19 (Ego) activates a Misbehaviour Detector.
22.000 s	Vehicle 162 (Attacker) starts to produce TCs which cannot be verified by the receiver's TAF	This applies only to the test cases considering transmission of bad TCs
35.093 s	Vehicle 19 (Ego) receives the first CPM of Series 2 attacks from Vehicle 162 (Attacker).	Start of Series 2 attacks. Vehicle 19 (Ego) activates a Misbehaviour Detector.

This scenario is used to demonstrate the following user story:

As the Vehicle I want to be able to extract an observation contained in a CPM, attribute it a Trustworthiness Level and record it in the LDM, so that it can be appropriately included in the extended perception, according to the considered RTL.

The acceptance criterium evaluated using the small scale scenario is the correctness in the attribution of the ATLs to the kinematic observations. This can be verified by inspection of the evolution of the Isolation-ATL and Temporal-ATL during the experiment.

The experiment is used to demonstrate the reaction to the attack involving the modification of the position of the Object vehicle in the CPMs sent by the Attacker. We repeat the experiment in the following two conditions, both compatible with the attack: Vehicle 162 (Attacker) produces TCs such that the harmonized attributes are always verified (this is compatible with the hypothesis of a malfunctioning of the sender's system not involving a malicious compromise); and none of the harmonized attributes in the TCs is ever verified (this is compatible with the hypothesis that the attack is the result of a compromise of the sender's system).

The setup for the two experiments is summarized in the table below. The main events are described in Table 3.9.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91 and of Vehicle 19, and sends its own TCs. **Ego (Vehicle 19)**: Receives CPMs and TCs from the Attacker (Vehicle 162) and from the Genuine (Vehicle 234). Runs MBD checks on each observation. Its onboard system produces an ATL and a Temporal-ATL on each received observation. **Ego (Vehicle 234)**: Broadcasts CPMs containing observations of Object 91 and Vehicle 162, and sends its own TCs.
Trust sources	Both MBD and TCs trust evidence is available at the Ego's TAF.
Considered metric	Evolution in time of the Isolation-ATL and of the Temporal-ATL. The Isolation-ATL is a subjective logic vector, consider Belief and Uncertainty components. The Temporal-ATL is a subjective logic vector, consider the Temporal-Belief (the Temporal-Uncertainty is always equal to the Uncertainty component). The Temporal-ATL is obtained for parameters $\tau = 0.93$ and $\gamma = 0.75$
Expected outcome	Dynamic evolution of the Temporal-ATL as a reaction to the tested attack pattern.
Exp 1: attack pattern	During the first experiment the the Attacker (162)'s TCs can always be verified. The observations of 91 are modified by the Attacker in two intervals (17-25 seconds; 35-40 seconds), which causes the activation of MBD Check 1 at the Ego's. The observations of 19 are modified by the Attacker from second 35 onwards, causing the activation of MBD Check 5 at the Ego's. The Genuine vehicle's (Vehicle 234) TCs can always be verified, and no MBD Check is activated at the Ego on its observations. The results of the first experiment are shown in Figures 3.11 and 3.12.
Exp 2: attack pattern	The second experiment differs from the first only in the fact that the Attacker starts to send invalid TCs from second 22. The results of the second experiment are depicted in Figure 3.13.

During the first experiment Vehicle 162 (Attacker) is able to provide verifiable TCs for all the experiment duration.

Vehicle 234 (Genuine) sends CPMs containing observations of Vehicle 162 and of Vehicle 91 (starting from second 13). Figure 3.11 shows the evolution of the ATLs evaluated by the Ego vehicle: on the left, the ATLs of the observations of Vehicle 162 found in CPMs by Vehicle 234; on the right, the ATLs of the observations of Vehicle 91 found in CPMs by Vehicle 234. In both cases we observe that the Isolation-Belief (in green) and the Temporal-Belief (in red) correctly maintain very elevated values. As a consequence, all the observations provided by Vehicle 234 are included in the extended perception consumed by the IMA.

Vehicle 162 (Attacker) sends CPMs containing observations of Vehicle 91 starting from second 13, and of Vehicle 19 as well, starting from second 31. Figure 3.12 shows the evolution of the corresponding ATLs, as evaluated by the Ego. On the top, the two attack series are shown in light

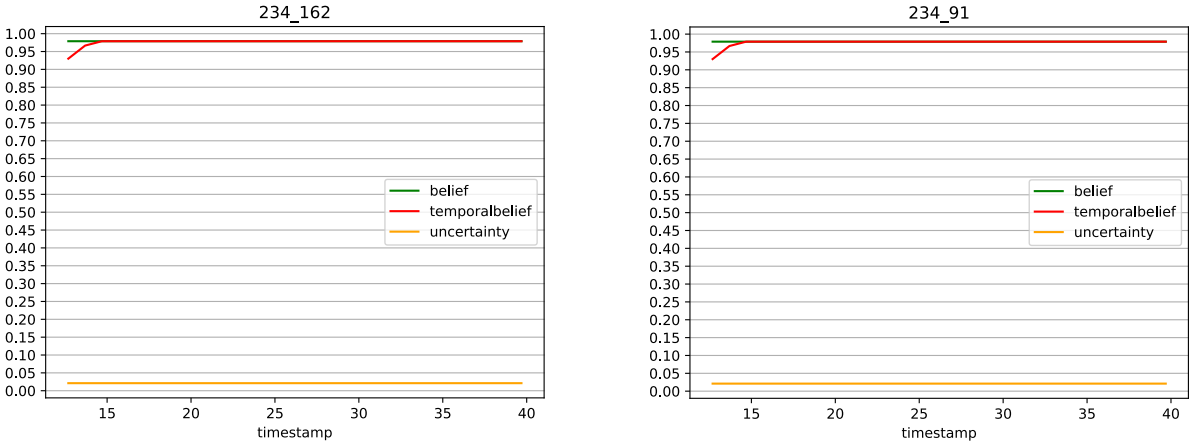


Figure 3.11: Small scale scenario: evolution of Isolation-ATL and Temporal-ATL for the observations contained in CPMs from Vehicle 234 (Genuine), as evaluated by the Ego (Vehicle 19).

blue. During the first attack phase, we observe the decrease of the Temporal-Belief. As soon as the first attack series ends, positive MBD trust evidence is collected, and the Temporal-Belief recovers, to start degrading again as soon as the second attack series starts. The bottom of Figure 3.12 shows the evolution of the ATL of the observations of Vehicle 19 contained in the CPMs by Vehicle 162 (Attacker). In this case, only the second series of attacks are involved, because Vehicle 162 does not perceive Vehicle 19 before second 31. Notice that the slope of the Temporal-Belief is steeper with respect to the case of the Temporal-Belief for the observation of Vehicle 91; this is due to the fact that the MBD detectors activated on observations of Vehicle 19 are more severe than the MBD detectors activated on the observations of Vehicle 91.

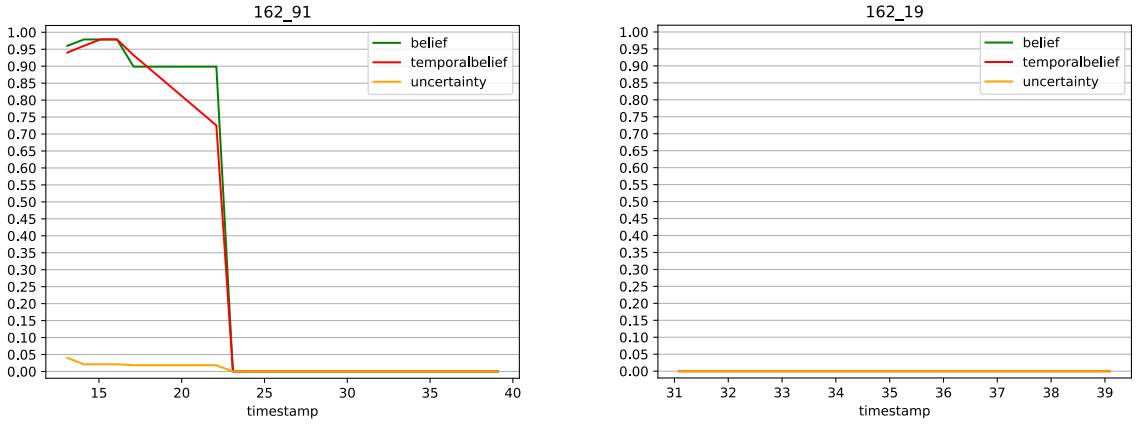


Figure 3.13: Small scale scenario: evolution of Isolation-ATL and Temporal-ATL for the observations contained in CPMs from Vehicle 162 (Attacker), when the Attacker is not able to produce verifiable TCs (from second 22).

Finally, Figure 3.13 illustrates the evolution of the ATLs on the observations contained in the CPMs sent by Vehicle 162 (Attacker), when Vehicle 162 starts to send non-verifiable TCs starting from second 22.

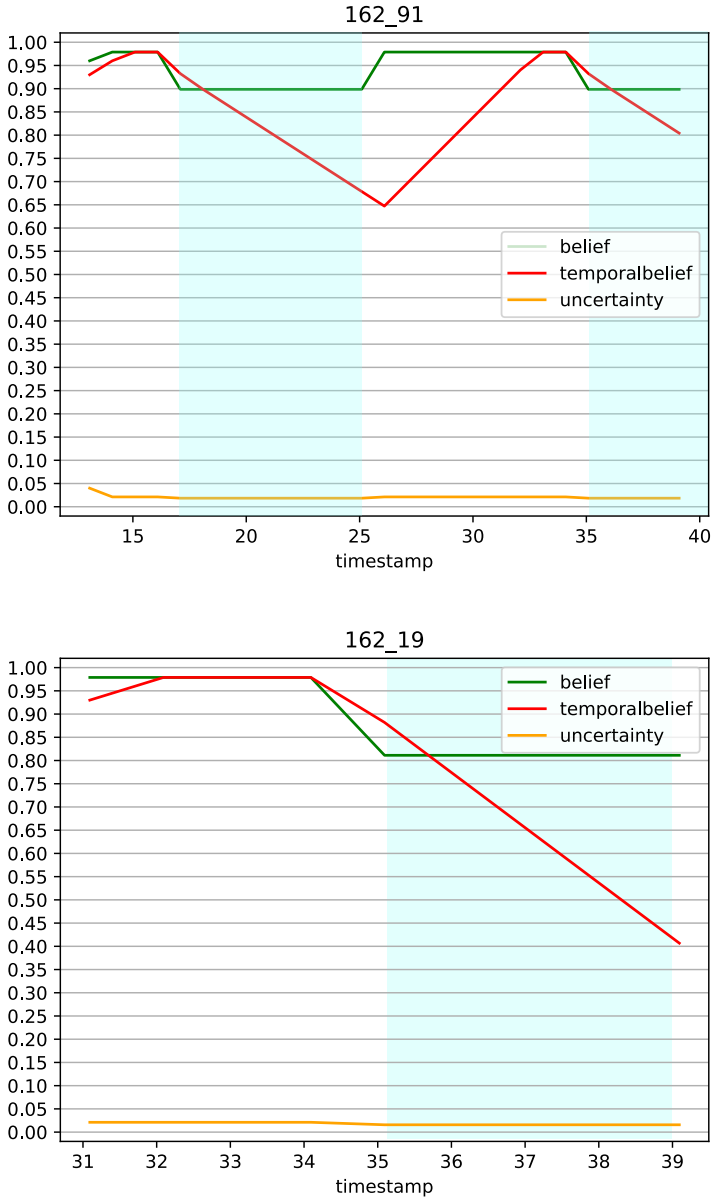


Figure 3.12: Small scale scenario: evolution of Isolation-ATL and Temporal-ATL for the observations contained in CPMs from Vehicle 162 (Attacker), as evaluated by the Ego (Vehicle 19).

3.1.6.6 Large-scale scene realisation

We move now to the large-scale scene, through which we explore the impact of the trust assessment on the contents of the extended perception consumed by the IMA application. The large-scale scene is used to demonstrate the following user story:

As the IMA application on the Vehicle, I want to be able to consume a consolidated view of the scene containing trustworthy data.

The acceptance criterium is that kinematic data whose trustworthiness level is below the RTL is excluded by the extended perception. Since the size of the large-scale scene does not allow

for the detailed inspection of the ATL traces for each considered vehicle, the evaluation will be performed through aggregate metrics. Before defining them, we proceed to describe the large-scale scene experimental setting and the considered attack models.

Experimental traces & Ground Truth dataset We consider the urban scenario of Paris Saclay (France), as shown in Fig. 3.14. It consists of a road network of size 1.24 km^2 with a stable vehicle density of 18.2 vehicles/km^2. The vehicle trips on this topology are randomly generated. All simulation settings are shown in Table 3.11. 80% of vehicles are connected, thus equipped with the Cooperative Perception service (they can both send and receive CPMs). Each equipped vehicle locally runs the MBD component. Starting from this configuration, we generate the Ground Truth dataset, composed by recording the Extended Perception snapshots of all connected vehicles, obtained when no attack is present. Each vehicle generates an extended perception snapshot every 300 ms. The extended perception snapshot contains the kinematic data of all the objects perceived by the Ego's sensors and all the objects known thanks to received CPMs. If one object's position has not been updated for longer than 2 seconds, the object is removed from the extended perception of the Ego.

Figure 3.14: Paris Saclay Network road topology

Attack model & Attack datasets The attack datasets are produced from the same experimental traces considered in the Ground Truth dataset: this means that the vehicle identities and the trips they take are the same as in the Ground Truth dataset. In an attack dataset, however, 30% of the connected vehicles are randomly selected to play the role of the Attacker. An Attacker has full priority access to sensor data and can modify sensor measurements when encoding them in its CPMs. We consider several attack types, as described in the table below, divided into two categories: alteration of the perceived object position; alteration of the perceived object speed. Each Attack dataset is formed by the collection of the Extended Perception snapshots generated by all connected vehicles, when no mitigation measure is in place. We consider a different type of attack for Attack dataset. For all attacks, the attacker nodes are always able to produce TCs containing harmonized attributes which are verifiable by the receiver.

Table 3.11: Simulation settings for the large-scale scene on the Paris Saclay Network

Simulation duration	0.5h
Penetration rate	0.8
Attacker rate	0.3
Scenario size	1.24 km^2
Vehicle density	18.2 Veh / km^2
Communication media	802.11p
Communication profile	ITS-G5
Communication type	Single Hop Broadcast
CPM interval	1 sec (fixed rate)
Front radar sensor	FoV range = 200m
	FoV angle = $\pm 20^{\circ}$

Random Position Offset (position alteration)	For each transmitted CPM, add a random noise sample to the actual distance of the perceived objects (the distance data from the sender allows the receiver to infer the absolute position of the perceived object). The distance is expressed in the vehicle's reference system (reference system with origin in the vehicle's reference point and x axis parallel to the longitudinal axis of the vehicle). The noise is obtained sampling two independent components from a Gaussian distribution $\mathcal{N}\left(0, \frac{\text{max\_sensor\_range}}{10}\right)$
Constant distance Offset (position alteration)	The attack is similar to the Random Position Offset, with the difference that each attacker will always add the same randomly generated noise sample on all attacked observations.
Random speed (speed alteration)	For each transmitted CPM, the speed of the perceived object is randomly chosen. The speed is a vector expressed on the vehicle's reference system. Both components are sampled from the uniform distribution $\mathcal{U}\left(0, \text{max\_speed}\right)$, where max_speed is chosen so that the sample is within a reasonable speed range.
Constant Speed (speed alteration)	The attack is similar to the Random Speed attack, with the difference that each attacker will always use the same randomly generated speed on all attacked observations.

CONNECT datasets Each CONNECT dataset is obtained by replaying the same configuration that generated the Attack dataset, this time having all the connected vehicles playing the role of the Ego and apply the CONNECT framework to help the generation of their Extended Perception. In particular, each connected vehicle is equipped with an oboard TAF, which can observe both TCs and MBD trust sources. A CONNECT dataset is hence parametrized by the type of attack and by the RTL value used by the vehicles to select the observations to retain in the extended perception. We consider several RTL values, where the Temporal-Uncertainty Threshold is always fixed to 0.7, and the Temporal-Belief Threshold takes a value in the set $\{0.93, 0.85, 0.75\}$.

Evaluation metrics In order to evaluate the performance of the onboard system with respect to its capability of selecting trustworthy observations to be included in the extended perception, we define a number of statistical metrics, to be applied to the considered datasets. The metrics are defined in the table below.

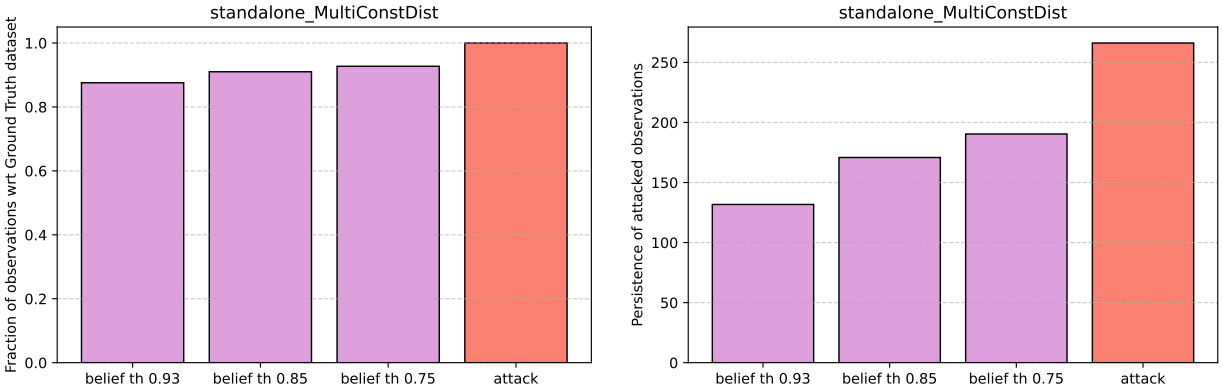


Figure 3.15: For the MultConstDist attack. On the left: proportion of observations in the CONNECT and Attack datasets, with respect to the number of observations in the Ground Truth. On the right: persistence of the attacked observations in the CONNECT datasets compared to the Attack dataset.

Number of observations	The cumulative number of observations considering all extended perception snapshots in a dataset. Used to assess the number of observations excluded by the trust decision in the CONNECT dataset, in comparison to the Ground Truth dataset.
Persistence of the attacked observations	The average (with respect to the identity of the perceived object) number of observations with altered position or speed, considering all extended perception snapshosts in a dataset. It is an indication of the proportion of attacked observations in a dataset. Used to assess the efficiency in removing corrupted observations.
Sample mean of the distance	The sample mean of the distance between the object in a snapshot of the extended perception in the CONNECT dataset and the same object in the same snapshot of the extended perception of the Ground Truth dataset. Used to evaluate the impact of the corrupted observations in the dataset on the accuracy of the extended perception, in terms of correct positioning of the objects.

Evaluation results, Constant Distance Offset attack In this section we consider the evaluation results for the Constant Distance Offset attack. The metrics, defined in the table above, are calculated comparing the Ground Truth and the Attack datasets to three CONNECT datasets, obtained setting the Temporal-Belief Threshold in the RTL to 0.93, 0.85 and 0.75, respectively. The comparison of the metrics across the CONNECT datasets allows to compare the effects of different thresholds.

We begin considering the number of observations in the datasets. The Ground Truth and the Attack datasets contain the same number of observations (70694), while the CONNECT datasets contain less as a consequence of the trust decisions. The left part of Figure 3.15 presents the number of observations per dataset, normalized to the number of observations in the Ground Truth dataset. As expected, using a higher Temporal-Belief threshold in the RTL provokes the exclusion of a larger number of observations in the CONNECT datasets. To be reassured that the excluded observations correspond to falsified data, we consider the right hand side of Figure 3.15, depicting the persistence

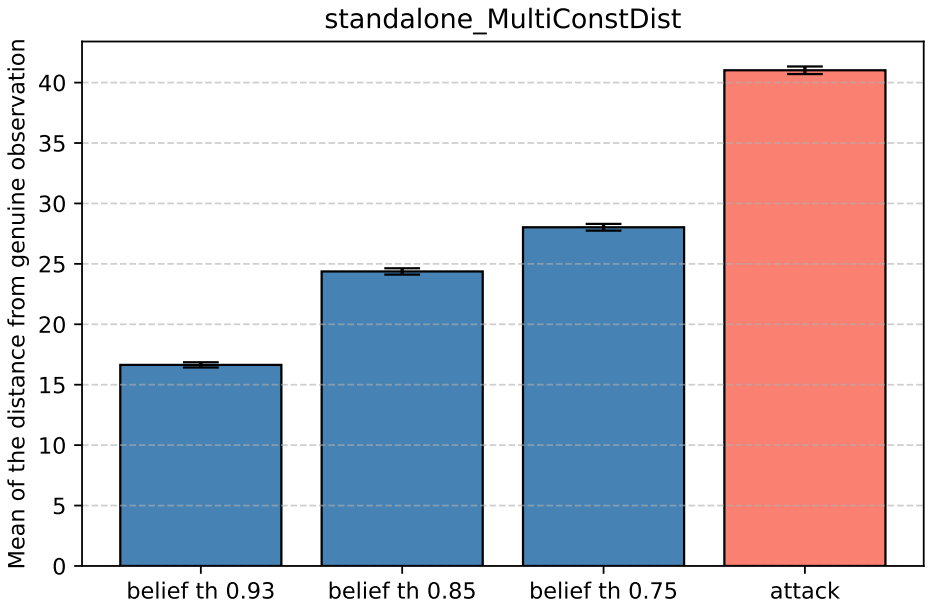


Figure 3.16: Sample mean of the distance between the same observation in the CONNECT dataset and in the Ground Truth dataset.

of the attacked observations in the dataset. The relative sizes of the bars in the persistence plot are indicative of the proportion of the attacked observations still included in the CONNECT dataset: 49% of the total attacked observations are still present in the CONNECT dataset with Belief threshold 0.93 (64% and 0.71% for Belief thresholds 0.85 and 0.75, respectively).

We now turn to consider the adverse effects of these attacked observations remaining in the CONNECT datasets. To do so, since the attack modifies the position of the perceived objects, we evaluate the sample mean of the distance between the observation in the CONNECT dataset and the corresponding observation in the Ground Truth dataset. The results are depicted in Figure 3.16. The sample mean of the distance is considerably diminished in the CONNECT datasets, with respect to the Attack dataset. This is the effect of the fact that the proportion of corrupted with respect to total observations is higher in the Attack dataset than in any of the CONNECT datasets; and more importantly, of the fact that the corrupted observations that are excluded from the CONNECT dataset are those presenting the largest deviations with respect to the Ground Truth.

Evaluation results, other attacks attack In this section we present the results of the evaluation of the large-scale scene for the set of different attacks, involving both position alteration and speed alteration. For each attack, the CONNECT dataset is generated for Temporal-Belief threshold 0.93. We summarise the results below. The Gain indicates the proportion of the attacked observations that have been removed in the CONNECT dataset, with respect to the total of the attacked observations in the Attack dataset.

3.1.7 Discussion & Critique - Lessons Learnt

3.1.7.1 Evaluation results discussion

The evaluation activity pursued a twofold objective. First, it aimed to validate the standalone trust assessment framework in the presence of heterogeneous trust sources. This was achieved through the analysis

Attack type	Attack dataset	CONNECT dataset	Gain
MultiRandomDistOffset	264.5	119.887	54%
MultiConstDist	266.129	131.725	50%
SingleConstSpeed	66.689	50.6363	24%
SingleRandomSpeed	61.916	37.922	38%

Table 3.14: Persistence of the attacked observations, for different types of attack.

of a small-scale scene, whose reduced complexity enabled an exhaustive examination of the system's outputs. As a result of this evaluation, the system's behavior was successfully validated. This in-depth analysis also yielded key insights regarding the diverse nature of trust sources and the distinct requirements they impose on the temporal aspects of trust assessment. These findings are summarized in the following subsections.

The second objective of the evaluation was to assess the effectiveness of the trust assessment system in terms of its impact on the performance of a CCAM application. The results of the evaluation on the large-scale scene demonstrate that the trust assessment system is capable of mitigating the effects of the analysed perception modification attacks on the extended perception. This evaluation also highlights the influence of the choice of the RTL on application performance, and of its sensitivity to the choice of the Temporal-Belief threshold in particular. The RTL choice needs to be performed accounting for the fact that a more restrictive RTL, while effective in removing altered data, may limit the environmental awareness, by reducing the number of known objects. In this sense, there exists a trade-off between maintaining the accuracy of the observations in the extended perception and maintaining the awareness of the existence of the objects on the road. In our experimentation activity we did not find any significative reduction in terms of awareness of the existence of objects; this may however be an effect of the chosen road topology or of the vehicle density that was considered.

Finally, a limitation of the evaluation setup for the large-scale scene is the fact that only positive TCs evidence is considered. As we have seen, MBD trust evidence is less effective in triggering the trust decision to exclude an observation, this requires repeated negative observations; this feature protects against exclusion of benign, but atypical data points. On the other hand, negative TCs evidence triggers an immediate reaction of exclusion. In this sense, the evaluation setup considered with the large-scale scene is the most challenging, and results obtained indicate the ability of the trust assessment system to remove untrustworthy data, even in absence of a clear indication of the compromise of the system that generated them.

3.1.7.2 On the heterogeneity of trust sources

Having heterogeneous trust sources is advantageous, as they can compensate for each other's limitations. In the context of the IMA use case, the presence of two trust sources allows untrustworthy data to be excluded in two ways: first, the sender vehicle may become unable to produce verifiable Trustworthiness Claims (TCs), which serve as a clear indication of a compromise. Second, the Ego system may also obtain direct evidence of data manipulation through Misbehavior Detection (MBD) activations. However, if a systematic malfunction occurs in the sender vehicle's kinematic data processing system, perhaps due to faulty sensors or components, the issue might not impact its ability to produce valid TCs. In this situation, the presence of heterogenous sources is crucial, as it still enables the Ego's trust assessment system to detect and respond to repeated anomalies in reaction to MBD evidence, enabling correct trust decisions.

During the IMA use case exploration, we examined the interplay between the two available trust sources to assess trust on kinematic data. The MBD trust source evaluates the kinematic data itself; while the TCs trust source evaluates the entity generating the kinematic data. Their interplay and their joint contributions in the trust assessment calculation are made possible by the proper design of the trust model, as well as

by the appropriate choice of the subjective logic operators selected to perform the trust model evaluation. A key lesson learned from the first experimentation period is that this is crucial.

This highlights an important point: while incorporating an additional trust source in an existing trust model may be straightforward when all considered trust sources refer to properties of the same trust node, the same process may require significantly more effort if trust sources span across multiple trust nodes. This emphasizes the importance of careful integration and consideration when expanding the availability of trust sources.

3.1.7.3 On the RTL retention in presence of time-dependent trust sources

Selecting the appropriate RTL for the IMA use case presented significant challenges. The RTL is defined based on both the Uncertainty and the Belief (or Temporal-Belief) components of the subjective logic trustworthiness level. The threshold for Uncertainty is generally straightforward because uncertainty rapidly drops as soon as trust evidence is observed. The more challenging aspect is determining the appropriate Belief threshold. Moreover, optimizing the RTL threshold for every possible driving scenario or attack pattern is impossible, emphasizing the need for the robustness of the system to imprecise RTL settings, so that correct trust decisions are eventually reached as more trust evidence is collected in time.

As demonstrated in the experimental context, the challenge arises when the decision on the Belief is based on trust sources, such as MBD, whose time dependence is critical for capturing the underlying phenomena. To address this, the Temporal-ATL has been introduced, which incorporates the necessary time-dependence for the trust evaluation process. In the context of the IMA use case, the Temporal-ATL serves as a suitable trustworthiness level indicator for RTL comparison, facilitating the correct trust decisions for the retention of observations in the extended perception. The Temporal-ATL provides inertia to manage episodic MBD activations caused by noise and prevents the Temporal-Belief from rising too quickly once perturbations subside. Additionally, it degrades the Temporal-Belief with repeated negative evidence, accurately reflecting the semantics of the MBD trust source. This aids in defining a region of acceptable RTLs for retaining observations in the extended perception. By using the Temporal-Belief, the RTL can detect repeated MBD check activations, indicating potential attacks, and making the performance of the IMA application less sensitive to the choice of the RTL. Therefore, the Temporal-ATL, which includes a temporal dimension, is crucial for accurately handling MBD data and detecting attack patterns that might go undetected with a static threshold alone.

The implementation of the Temporal-ATL, however, requires careful tuning of the parameters τ and γ, which control the dynamics of the Temporal-Belief component. The correct tuning of these parameters must be guided by use case needs. In the future, when the Temporal-ATL will be integrated into the TAF, efforts will be needed to ensure that these parameters are tunable, ensuring compatibility with specific use case requirements.

3.2 Federated scenario: Ghost Object Injection (#1)

3.2.1 Description

In CONNECT, federation refers to the mechanism that enables the collaboration of two or more instances of the TAF, which are typically not co-located, in the trust assessment process. The second scenario explored in the IMA use case highlights **the advantages of leveraging the CONNECT federated modality of the Trust Assessment Framework**. In particular, we want to explore the impact on the accuracy of the trust assessment process of the availability of a service deployed remotely, able to gather and use trust evidence that would not be otherwise available to the entity performing the trust decision. In the context of the IMA use case, federation plays a key role in unlocking the benefits of the MEC support to help the vehicle take

trust decisions on the received V2X data. In the “standalone scenario”, we assumed that the senders of V2X messages would also disseminate their Trustworthiness Claims (TCs) on the V2X radio interface and that the vehicle TAF could then use them as a source of trust. However, this assumption may not be realistic in the near term, as to be practically viable it would require a standardization process to define a new V2X message structure. The payloads of already standardized V2X messages are subject to strict size limitations, which poses a challenge to TCs integration in CAM or CPM payloads.

We now move to an alternative scenario, in which vehicles transmit their TCs to the infrastructure, similarly to how they report misbehaviour to the Misbehaviour Authority. In the federated architecture, part of the trust assessment process is then carried out by the MEC, which can process evidence (e.g., the TCs) that is not observable by the vehicle itself. Federation enables the outcomes of trust assessments by the TAF in the MEC to be shared and integrated into the trust assessments performed locally by the vehicle. The federated architecture allows for a ”trust constellation” based on types of trust evidence that may not be available to all components of the overall trust assessment framework. Or, even in the case where all types of evidence are available, the abundance of this trust evidence may vary with out the input from the MEC. In the IMA use case, we will use the MBD trust source to illustrate this phenomenon. In this case, a higher volume of MBD evidence is available at the MEC, because it collects the MBD reports of all vehicles in the coverage area. This abundance of evidence will make the trust assessment much more accurate when federation can be used, than when performed at the vehicle, only. It is worth noting here that the same trust source, the MBD, has a different semantics at the TAF in the vehicle and at the TAF in the MEC. In the vehicle, which is assessing trust on data items, MBD evidence is interpreted with respect to the property of the data of being correct. On the MEC, the TAF is assessing trust on the entities generating data, but never on the data itself, MBD evidence is interpreted with respect to the capability of the entity to provide correct data.

The federation capability was developed during the second evaluation period and was therefore not considered at all in the first experimentation period. In the context of the final release of the overall CONNECT framework, the federated trust model for the vehicle, the standalone trust model for the MEC, and the NTM application have been designed and tested. This is demonstrated in the evaluation scenario presented here, which takes place at a T-junction where three connected vehicles are in transit, as depicted in Figure 3.17. One of the vehicles performs a ghost object attack by injecting false observations of a non-existent object into its CPMs. The ghost object is indicated by the pink box in Figure 3.17. While the attack goes undetected by the onboard MBD systems of all vehicles, which cannot reliably perform MBD plausability checks against their local perception, an RSU positioned at the intersection is able to detect it, and accordingly uploads MBD reports to the MEC. Through the federation mechanism, the MEC disseminates an NTM message, which propagates a degrading trustworthiness level on the attacker node to the TAF on the vehicles. In this way, the genuine vehicles are able to exclude the data from the attacker from their extended perception. This is demonstrated in the evaluation scenario presented here, which takes place at a T-junction where three connected vehicles are in transit, as depicted in Figure 3.17. One of the vehicles performs a ghost object attack [ZJN24] by injecting false observations of a non-existent object into its CPMs. The ghost object is indicated by the pink box in Figure 3.17. While the attack goes undetected by the onboard MBD systems of all vehicles, which cannot reliably perform MBD plausibility checks against their local perception, an RSU positioned at the intersection is able to detect it, and accordingly uploads MBD reports to the MEC. Through the federation mechanism, the MEC broadcasts an NTM message, which propagates a degraded trustworthiness level on the attacker node to the TAF on the vehicles. In this way, the genuine vehicles are able to exclude the data from the attacker from their extended perception.

In the remainder of this section, we provide an updated implementation path report to document the updates performed in the testbed for supporting the benchmarking of the CONNECT trust-related artifacts in such a complex scenario. We then evaluate a series of test cases to investigate the overall trust assessment process in the federated architecture, with respect to the different trust sources available at each location. Adopting the experimentation strategy also followed in the standalone scenario, we first validate the feasibility of such trust federation/constellation in a small-scale driving scene prior to elevating the tests

in a large-scale scenario to enable the applicability and benefits of this modality in a real-world situation to be assessed.

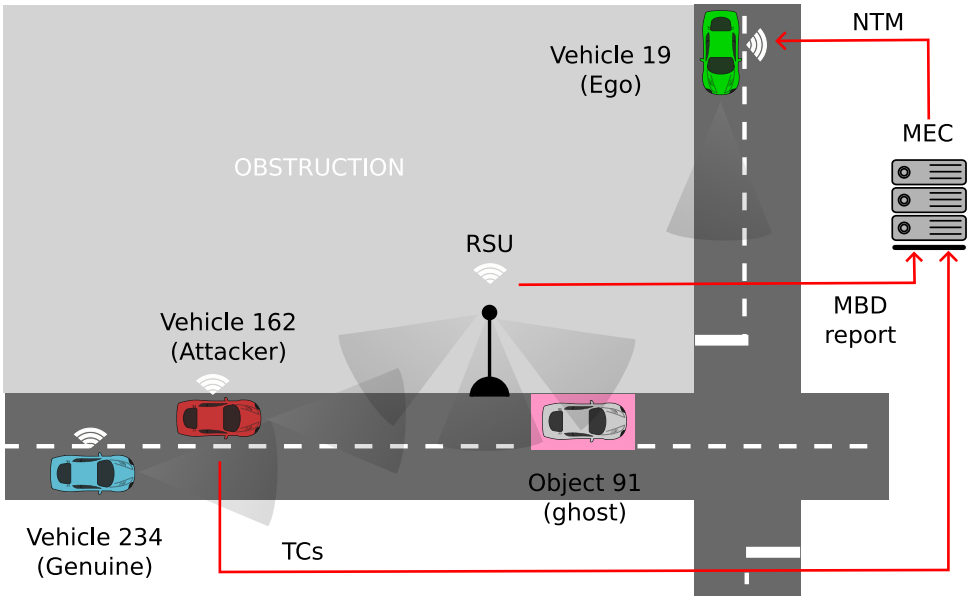


Figure 3.17: Depiction of the federated scenario, small scale evaluation setting. The Attacher inserts a ghost Object (pink box) in its CPMs.

3.2.2 Testbed Details

The simulation testbed considered for the federated scenario is again the one depicted in Figure 3.2. All of the components are considered for this scenario: the ones in the vehicle (highlighted in gray in the figure) and those in the MEC (TAF and NTM app). In the simulation testbed, the MEC is a backend entity whose communication with the vehicle is enabled thanks to the shared communication bus. From the simulation viewpoint, the RSU is seen as a stationary vehicle, with a larger field of view than a regular vehicle; the RSU's onboard system is the same as the vehicle's. The RSU does not send CPMs, but performs local MBD checks, including Check 8 (check on the consistency of received CPMs with the local perception), which is not performed by moving vehicles (because less reliable due to the motion).

The role of the TAF in the MEC is to perform trust assessment on the vehicles transmitting CPM messages within the MEC's geographic coverage area. To do so, the TAF in the MEC has the capability to process two types of trust sources: the TCs, which in this use cases are not transmitted along with the CPM messages, but are directly uploaded to the MEC, as depicted in Figure 3.17; and the MBD reports uploaded by any of the vehicles in the coverage area, whenever an incoming V2X message triggers a local MBD check. It is important to note that each MBD report identifies the vehicle responsible for the V2X message under evaluation.

Regarding the federation mechanism, an explicit communication approach has been adopted between the TAF in the MEC and the TAF in the vehicle. This is achieved through the broadcast of a dedicated message, the Node Trust Message (NTM), which conveys the trustworthiness levels of the active vehicles assessed by the MEC. This design is well suited to the IMA use case, where the outcome of the trust assessment process by the MEC is of interest for all of the vehicles in the coverage region. Broadcasting the NTM message thus emerges as an efficient solution that minimises the communication overhead. The NTM application is responsible for the creation of the NTM messages containing the trustworthiness levels of known vehicles, as received by the TAF in the MEC, and of their dissemination in a periodic fashion. An important consequence of this approach is that some delay may be induced between the moment new

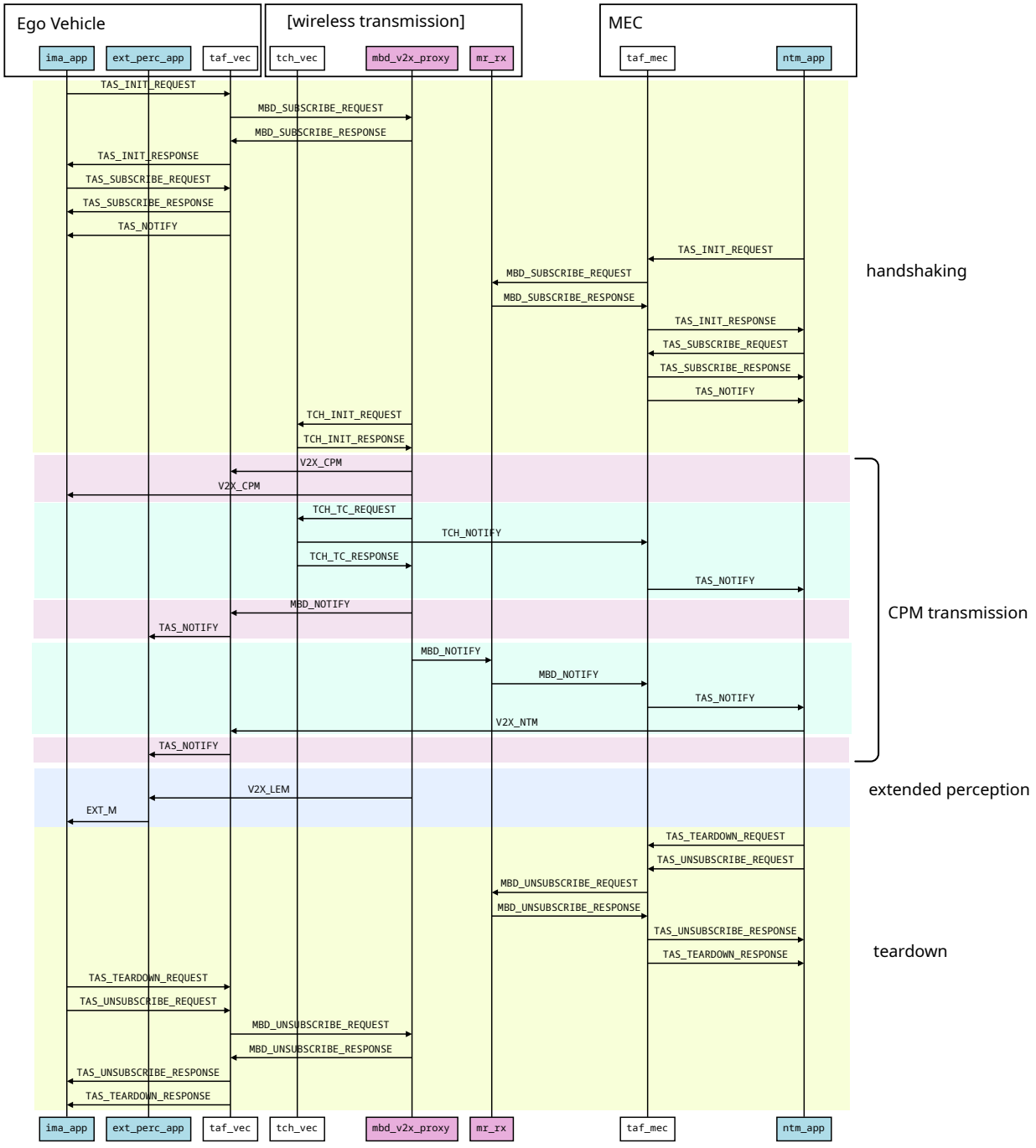


Figure 3.18: Storyline of Federated scenario (#1). The green and purple bands describe the message exchange triggered by the transmission of a single CPM. The blue band corresponds to the local perception message, which triggers the generation of the extended perception.

evidence is processed at the MEC and the moment when its informative content is conveyed to the TAF at the vehicle. This phenomenon is inherent to the periodic way in which the NTM dissemination occurs.

The functional behaviour of the overall architecture considered for the federated scenario is illustrated in Figure 3.18, which depicts the overall message flow and interactions. The sequences highlighted in yellow at the beginning and end represent the handshake and teardown procedures, respectively, required as part of the TAF configuration. These procedures occur both between the TAF and the IMA application at the Ego vehicle, and between the TAF and the NTM application at the MEC.

The message exchanges triggered by the transmission of a CPM are shown in the central part of the figure, where messages addressed to the Ego vehicle's components are highlighted in light purple, and those addressed to the MEC's components are highlighted in light green. As usual, when the vehicle receives a CPM, the testbed triggers the TCH component to transmit the TCs on the status of the transmitting vehicle. The TCs are received by the TAF in the MEC, which uses them as a trust source to update the ATL of the transmitting vehicle. The ATL update prompts the TAF to notify the NTM application.

Finally, the NTM application is responsible for generating NTM messages that include the ATLs of known vehicles. These messages are broadcast and received by the TAF in the vehicle, where they serve as a trust source for evaluating the received kinematic observations. As it will be discussed in Section 3.2.6, the trustworthiness level indicator chosen to be propagated in the NTM is the Temporal-ATL.

As done for the standalone scenario, we describe in Table 3.15 the identities and roles of the simulated entities that will be used for benchmarking and small scale experiments.

Table 3.15: Participating (simulated) entities in the conducted experiments

Vehicle ID	Role and Description
19	*Ego vehicle*. This is the vehicle under test. It runs the onboard trust assessment system and uses V2X inputs to detect misbehavior.
91	*Ghost vehicle*. This vehicle is not a real object on the road. The Attacker includes this observation inside its transmitted CPMs.
162	*Attacker vehicle*. It behaves as a misbehaving V2X node, sending kinematic data about the non-existing Object 91 in its CPMs.
234	*Genuine vehicle*. It is a legitimate V2X node broadcasting correct CPMs. It is considered as a second Ego vehicle.
RSU	*The RSU*. The Road Side Unit is deployed in the intersection, and is equipped with onboard sensors. The comparison between the RSU's own perception and the received V2X messages allows the RSU's MBD system to activate MBD check 8.

3.2.3 Trust Model

Unlike the single trust model used for the IMA-MBD Scenario 2, which is used by the standalone TAF, hosted only on the vehicle, and can produce an ATL without the need for cooperation with another TAF, this scenario features two trust models that require the cooperation of two TAF entities. One TAF is hosted on the ego vehicle receiving the CPMs from other vehicles, and the other TAF is hosted on the Mobile Edge Computer (MEC) assessing trustworthiness of all of the V2X enabled vehicles. Therefore, there are two trust models, the trust model on the ego vehicle, used for assessing the trustworthiness of observations inside CPMs, and the trust model on the MEC, used for assessing the trustworthiness of the V2X enabled vehicles sending the CPMs. The trust opinion on V2X vehicles assessed by the TAF on the MEC is sent to the TAF on the ego vehicle. The TAF on the ego vehicle requires the trust opinion generated by the MEC TAF to plug into its local trust model and perform the trust assessment.

3.2.3.1 Trust Model on the Vehicle

This Trust Model focuses on assessing the trustworthiness level of observations inside CPMs received by the ego vehicle in the Intersection Movement Assist (IMA) federated scenario and is shown in Figure 3.19. Just like in the standalone scenario, the ego vehicle uses CPMs from other vehicles as input to the IMA application. However, unlike in the standalone scenario, the ego vehicle's TAF will have the support of a TAF at the MEC, which will provide opinions about sender vehicles' trustworthiness.

- **Location of the TAF**: Ego vehicle, V_e
- **Root node / agent of the TM**: Ego vehicle, V_e, that processes the received CPMs
- **Propositions of the TM**: Observations inside CPMs, $C_{x|x}$, $C_{x|y}$, etc.
- **Other Trust Objects in the TM**: Sender vehicle, V_x, and the MEC, M

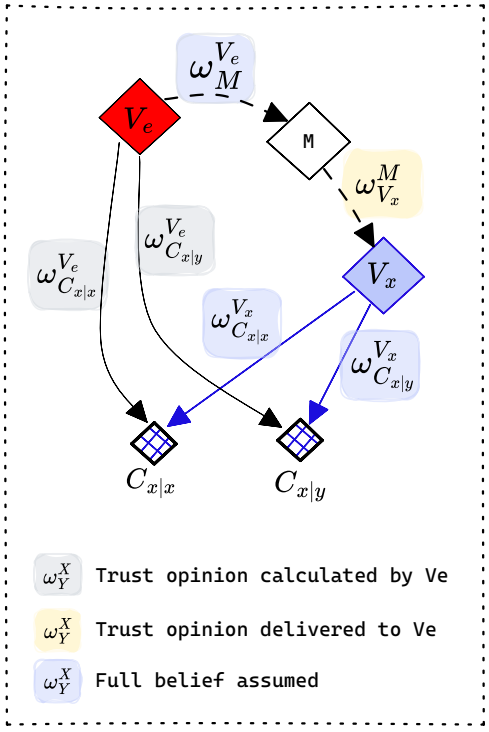


Figure 3.19: IMA-MBD Scenario 1 - Trust Model for the federated TAF on the ego vehicle

In this trust model, the ego vehicle plays the central role in assessing CPM data from other vehicles. A trust model instance is created for each sender vehicle as in the IMA Scenario 2. However, in this scenario, aside from assessing trustworthiness of the CPM observations directly by using the MBD, the ego vehicle will receive trust opinions on the sender vehicle from the MEC, as well as the evidence of MEC's own trustworthiness. For this purpose, there needs to be a TAF hosted on the MEC running its own trust model and able to interact with the TAF on the ego vehicle.

3.2.3.2 Trust Model on the MEC

The Trust Model for assessing the trustworthiness of sender vehicles by the MEC in the Intersection Movement Assist (IMA) Scenario 1 is shown in Figure 3.20. The role of the TAF on the MEC utilizing this trust

model is to build an opinion on every vehicle that the MEC receives a CPM from. To do this there is a Trust Model Instance for each of these vehicles.

- **Location of the TAF**: MEC, M
- **Root node / agent of the TM**: MEC, M
- **Propositions of the TM**: Sender vehicle(s), V_x, V_y, V_z, etc.
- **Other Trust Objects in the TM**: none

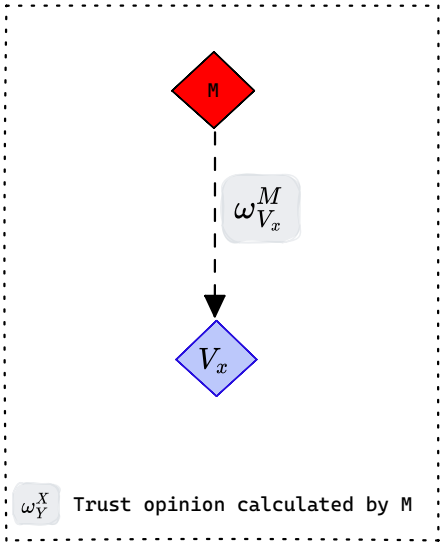


Figure 3.20: IMA-MBD Scenario 1 - Trust Model for the federated TAF on the MEC

3.2.3.3 Trust Opinions

There are the following trust opinions in the trust model instance on the ego vehicle:

1. $\omega_M^{V_e}$ = trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **MEC**, M, to not have been compromised.
2. $\omega_{V_x}^{M}$ = trust opinion assessed by the **MEC**, M, on the trustworthiness of the **sender vehicle** V_x to send messages (CPMs) whose integrity has not been compromised.
3. $\omega_{C_{x|x}}^{V_x}$ = trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about itself** contained in most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.
4. $\omega_{C_{x|y}}^{V_x}$ = trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about vehicle** V_y contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.
5. $\omega_{C_{x|x}}^{V_e}$ = trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **observation about vehicle** V_x contained in most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.

6. $\omega^{V_e}_{C_{x|y}}$ = trust opinion assessed by the **ego vehicle**, V_e, on the trustworthiness of the **observation about vehicle** V_y contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.

There are the following trust opinions in the trust model instance on the MEC:

1. $\omega^{M}_{V_x}$ = trust opinion assessed by the **MEC**, M, on the trustworthiness of the **sender vehicle** V_x to send messages (CPMs) whose integrity has not been compromised.

3.2.3.4 Trust Sources and Evidence

This subsection provides a list of all the different trust sources and evidence to be collected for each trust opinion in the trust model. Moreover, a trust quantification approach is given for each evidence.

1. $\omega^{V_e}_{M} \rightarrow$ the **ego vehicle** assesses this opinion based on the evidence which it receives from the MEC about its trustworthiness in form of verifiable presentations.
   - **Trust Source** = TCH
   - **Evidence** = Verifiable presentations - Verifiable presentations include the output of the single security controls of the vehicle computer of the sending vehicle if the security controls are not implemented (value: -1), are implemented and detected something (value: 0) or are implemented and did not detect anything (value: 1)
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the verifiable presentations is described in D6.1 [Con24b] and D3.3 [Con25c].
2. 2. $\omega^{V_e}_{C_{x|x}} \rightarrow$ the **ego vehicle** assesses this opinion based on the results of the Misbehavior Detection system running on the ego vehicle
   - **Trust Source** = MBD system on ego vehicle
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
3. $\omega^{V_e}_{C_{x|y}} \rightarrow$ the **ego vehicle** assesses this opinion based on the results of the Misbehavior Detection system running on the ego vehicle
   - **Trust Source** = MBD system on ego vehicle
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
4. $\omega^{M}_{V_x} \rightarrow$ the **MEC** assesses this opinion based on the evidence which it receives from the sender vehicle about its own trustworthiness in form of verifiable presentations, and by parsing MBD evidence contained in Misbehaviour Reports sent by vehicles and/or RSUs receiving CPMs from the sender vehicle.
   - **Trust Source** = TCH, MBD

- **Evidence** = Verifiable presentations, MBD - Verifiable presentations include the output of the single security controls of the vehicle computer of the sending vehicle if the security controls are not implemented (value: -1), are implemented and detected something (value: 0) or are implemented and did not detect anything (value: 1). The MBD includes the output of the misbehavior detectors of the messages provided by the sender vehicle.
- **Trust Quantification Approach** = The approach for calculating the trust opinion based on the verifiable presentations is described in D6.1 [Con24b] and D3.3 [Con25c]. The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c]. Both trust opinions are fused together to calculate $\omega_{V_x}^{M}$.

5. $\omega_{C_{x|x}}^{V_x}$ = the ego vehicle assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.
6. $\omega_{C_{x|y}}^{V_x}$ = the ego vehicles assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.
7. $\omega_{M}^{V_x}$ = the ego vehicle assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.

3.2.3.5 Actual Trustworthiness Level

$ATL_{C_{x|x}}$ is obtained in two steps: 1) first, $\omega_{M}^{V_e}$, $\omega_{V_x}^{M}$, and $\omega_{C_{x|x}}^{V_x}$ are discounted to obtain $\omega_{C_{x|x}}^{V_e;M;V_x}$, 2) second, the resulting opinion $\omega_{C_{x|x}}^{V_e;M;V_x}$ is fused with the direct opinion $\omega_{C_{x|x}}^{V_e}$ to obtain the ATL. We are using the opposite-belief trust discounting and the cumulative fusion operators in this example.

$$ATL_{C_{x|x}} = \omega_{C_{x|x}}^{V_e} \oplus (\omega_{M}^{V_e} \otimes \omega_{V_x}^{M} \otimes \omega_{C_{x|x}}^{V_x}) \tag{3.7}$$

Similarly, $ATL_{C_{x|y}}$ is also obtained in two steps: 1) first, $\omega_{M}^{V_e}$, $\omega_{V_x}^{M}$, and $\omega_{C_{x|y}}^{V_x}$ are discounted to obtain $\omega_{C_{x|y}}^{V_e;M;V_x}$, 2) second, the resulting opinion $\omega_{C_{x|y}}^{V_e;M;V_x}$ is fused with the direct opinion $\omega_{C_{x|y}}^{V_e}$ to obtain the ATL. Once again, we are using the opposite-belief trust discounting and the cumulative fusion operators for this purpose.

$$ATL_{C_{x|y}} = \omega_{C_{x|y}}^{V_e} \oplus (\omega_{M}^{V_e} \otimes \omega_{V_x}^{M} \otimes \omega_{C_{x|y}}^{V_x}) \tag{3.8}$$

3.2.4 User Story Realisation

The realisation of the following User Stories is considered in the federated scenario.

[MB.US5] As the vehicle, I want to obtain an updated trustworthiness level on the received data when an NTM message from the MEC updates information about the trustworthiness level of the emitter V2X-node.

[MB.US6] As the NTS Service Provider at the MEC I want to update the Trustworthiness Level of the V2X-node whenever I receive a MBD report or TCs, to deliver fresh information through the NTS service.

The evaluation of MB.US6 is documented in Sections 3.2.6.1 and 3.2.6.2 while the evaluation of user story MB.US5 is documented in Section 3.2.6.3.

The realisation of the following User Stories, already considered in the standalone scenario, is considered again for the federated scenario. The introduction of the federated architecture makes it in fact interesting to compare the evaluation outcomes of the two.

[MB.US1] As the IMA application on the Vehicle, I want to be able to consume a consolidated view of the scene containing trustworthy data.

[MB.US2]: As the Vehicle I want to be able to extract an observation contained in a CPM, attribute it a Trustworthiness Level and record it in the LDM, so that it can be appropriately included in the extended perception, according to the considered RTL.

The evaluation of user story MB.US2 in the context of the federated scenario is performed using the small-scale scene and documented in Section 3.2.6.5. Relevant to this use case, the benchmarking of the use of the MBD trust source both at the MEC and at the vehicle is documented in Section 3.2.6.4.

Finally, the evaluation of the user story MB.US1 in the context of the federated scenario is performed using the large-scale scene and this is documented in Section 3.2.6.6.

3.2.5 KPI & Acceptance Criteria

The user stories MB.US1 and MB.US2, already evaluated in the standalone scenario, are evaluated again in the federated scenario. We consider the same KPIs as before. The table below is updated to indicate where the results for the federated scenario are documented.

Table 3.16: Evaluated KPIs by user stories, in the standalone scenario.

User story	KPI description	Acceptance criteria	Results
MB.US1	The consolidated view of the scene contains trustworthy data	Only sufficiently trustworthy data is used to produce the extended perception	Validated and documented in Section 3.2.6.6, with respect with different RTL values.
MB.US.2	Processing complexity until LDM update	$\leq$ **100 ms** from the reception of a new CPM and V-TC and update of the LDM table (kinematic information and ATL).	Documented in Deliverable D6.1 [Con24b]
	The ATL expressed on observations of the same physical object performed by the same V2X-node evolves correctly	When the opinion on the active V2X-node degrades, the ATL on all the observations provided by the active V2X-node degrades	As discussed in the standalone scenario, this user story does not reflect the implemented TAF architecture, where the opinion on the V2X-node is a quantity internal to the TAF and is not disclosed explicitly to the vehicle. However, the degradation of the opinion on the V2X-node is visible in the contents of received NTM messages. This is validated and documented in Sections 3.2.6.3 and 3.2.6.4. 3.1.6.1.
		When the Local Misbehaviour Report contains an active detector on the observation of the physical object performed by the active V2X node, the ATL on the observation degrades.	Validated and documented in Sections 3.2.6.3 and 3.2.6.4.
MB.US5	The NTM message affects the trustworthiness level of the kinematic data assessed by the vehicle	The ATL associated to an observation in the LDM table degrades upon the reception of an NTM with negative ATL on the generating V2X-node	Validated and documented in Sections 3.2.6.3 and 3.2.6.4.

MB.US6	The ATL expressed by the MEC on the same V2X-node evolves correctly	The ATL to be included in the NTM are updated as a consequence of the observation of fresh trust evidence by the MEC	Validated and documented in Sections 3.2.6.1 and 3.2.6.2.

3.2.6 Evaluation

3.2.6.1 MBD as a Trust Source for the MEC-TAF

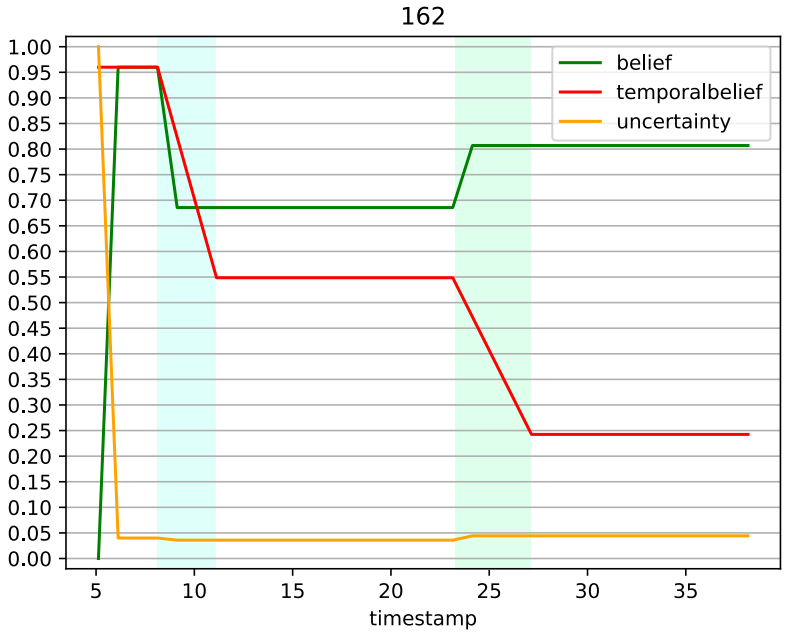


Figure 3.21: Evolution of the ATL of the node 162, as included in the NTM messages broadcast by the NTM app at the MEC.

To examine the interaction among the various components of the federated trust assessment architecture, we begin by validating the behaviour of the TAF at the MEC with respect to the MBD report as a trust source. The trust model used at the MEC handles MBD reports in a manner similar to how the standalone trust model handles MBD activations. Therefore, our evaluation considers both the Isolation-ATL and the Temporal-ATL.

We consider an experiment where Vehicle 162 (the Attacker) performs two attack phases during which two series of consecutive MBD reports containing active MBD checks for Vehicle 162 reach the TAF at the MEC. The setup for the two experiments is summarized in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Vehicle 91, and sends its own TCs to the MEC. **Reporter (RSU)**: Receives CPMs from the Attacker (Vehicle 162). Runs MBD checks on each observation and sends MBD reports to the MEC when MBD checks are activated.
Trust sources @MEC	Both MBD reports and TCs trust evidence is available at the MEC's TAF.

Considered metric @MEC	Evolution in time of the Isolation-ATL and of the Temporal-ATL on the Attacker (Vehicle 162) included in the NTM messages. The ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.96$ and $\gamma = 0.5$.
Expected outcome	Dynamic evolution of the Temporal-ATL as a reaction to the tested attack pattern.
Exp: Attack pattern	The Attacker (Vehicle 162) carries on two series of attacks. The first, between seconds 10 and 13, triggers the upload of three MBD reports; the second, between seconds 23 and 27, triggers the upload of four MBD reports. The Attacker (Vehicle 162) is always able to upload valid TCs to the MEC. The results of the experiment are depicted in Figure 3.21.

MBD as a trust source at the MEC benchmark Figure 3.21 shows the evolution in time of the ATL of Vehicle 162 at the time of the NTMs generation, as assessed by the TAF at the MEC. The interval highlighted in blue corresponds to reaction to the reception of the MBD reports of the first attack phase. The interval highlighted in green corresponds to the reaction to the reception of the MBD reports of the second attack phase.

As before, the Belief and Uncertainty component of the Isolation-ATL expressed by the TAF at the MEC are depicted in green and yellow, respectively. The Isolation-Belief component drops at the reception of the first MBD report (containing MBD Check 1), and stays constant during the remainder of the first attack phase. The value of the Isolation-Belief is updated at the beginning of the second attack phase, when a MBD report comes, with the activation of a different MBD check (MBD Check 3).

Similarly to the observation made in the standalone scenario, we consider that this behaviour does not satisfy the requirements of the IMA use case; as before, we consider the Temporal-ATL as an alternative. The evolution of the Temporal-Belief is depicted in red in Figure 3.21: during the first attack phase its value decreases, as more negative trust evidence is collected. During the interval 11.5 – 23.5 seconds no new MBD trust evidence is collected, and the Temporal-Belief stays constant, to start again its decrease as soon as the second attack phase begins, and new MBD reports about Vehicle 162 are collected.

As a consequence, the Temporal-ATL is the trustworthiness level metric chosen to be included in the NMT messages to be sent to the TAF at the vehicle.

3.2.6.2 TCs as a trust source for the TAF at the MEC

To complete the validation of the trust assessment at the MEC, we examine the response of the system to the observation of negative TC evidence. We use the same experimental setup as before, where Vehicle 162 (the attacker) carries out two attack sequences that trigger the reception of two corresponding series of MBD reports at the MEC. In this scenario, Vehicle 162 generates verifiable TCs up to second 25. After that point, it begins issuing TCs that are no longer verifiable. The setup for the experiment is summarized in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Vehicle 91, and sends its own TCs to the MEC.

	Reporter (RSU): Receives CPMs from the Attacker (Vehicle 162). Runs MBD checks on each observation and sends MBD reports to the MEC when MBD checks are activated.
Trust sources @MEC	Both MBD reports and TCs trust evidence is available at the MEC's TAF.
Considered metric @MEC	Evolution in time of the Isolation-ATL and of the Temporal-ATL on the Attacker (Vehicle 162) included in the NTM messages. The ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.96$ and $\gamma = 0.5$.
Expected outcome	Dynamic evolution of the Temporal-ATL as a reaction to the tested attack pattern.
Exp: Attack pattern	The Attacker (Vehicle 162) carries on two series of attacks. The first, between seconds 10 and 13, triggers the upload of three MBD reports; the second, between seconds 23 and 27, triggers the upload of four MBD reports. The Attacker (Vehicle 162) uploads verifiable TCs to the MEC up to second 25; after, it uploads TCs that cannot be verified. The results of the experiment are depicted in Figure 3.22.

TC as a trust source at the MEC benchmark Figure 3.22 shows the evolution of the Temporal-ATL for Vehicle 162 as included in the NTM messages generated by the MEC. For comparison, the Isolation-ATL produced by the TAF is also displayed. In both cases, the impact of the MBD reports is evident within the highlighted regions.

As expected, starting from second 25, as soon as the first negative TC from Vehicle 162 is received, both the Isolation-Belief and the Temporal-Belief drop to zero, due to the reception of the first non verifiable TCs from Vehicle 162.

3.2.6.3 TCs as a trust source at the MEC and MBD as a trust source at the Ego

After validating the trust assessment at the MEC side, we consider what happens at the Ego as effect of the federation. More precisely, we are interested in evaluating the system when the MEC and the Ego observe different trust sources: TCs and MBD, respectively.

We consider an experiment where the Ego (Vehicle 19) receives CPMs from the Attacker (Vehicle 162) which do not raise any MBD detector, locally at the vehicle. However, the Attacker is not able to upload valid TCs to the MEC. The setup of the experiment is summarized below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91, and sends its own TCs to the MEC. **Ego (Vehicle 19)**: Receives CPMs from the Attacker (Vehicle 162). Runs MBD checks on each observation.
Trust sources @MEC	The TCs trust evidence is available at the MEC's TAF.
Trust sources @Ego	The MBD trust source is available at the Ego.

Considered metric @MEC	Evolution in time of the Temporal-ATL on the Attacker (Vehicle 162) included in the NTM messages. The Temporal-ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.96$ and $\gamma = 0.5$.
Considered metric @Ego	Evolution in time of the Isolation-ATL and of the Temporal-ATL of the observations of Object 91 sent by the Attacker (Vehicle 162). The ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.93$ and $\gamma = 0.75$.
Expected outcome	The trust assessment on the Attacker propagated through the NTM messages to the Ego's TAF allows to express the correct Isolation-ATL and Temporal-ATL on the kinematic data originating from the Attacker.
Exp: Attack pattern	The Attacker (Vehicle 162) carries on two series of attacks which go undetected by the Ego's MBD system. The Attacker (Vehicle 162) uploads verifiable TCs to the MEC up to second 20; after, it uploads TCs that cannot be verified. The results of the experiment are depicted in Figure 3.23.

TCs at the MEC, MBD at the Ego benchmark This benchmark is evaluated using the experiment depicted in Figure 3.23. On the left-hand side it is depicted the evolution of the Temporal-ATL, as included in the NTM messages. On the right-hand side, it is depicted the Isolation-ATL and Temporal-ATL of the observations of Object 91, as sent by Vehicle 162 in its CPMs. The TAF at the vehicle, which can process the MBD trust source only, does not have any negative evidence on the data coming from Vehicle 162. However, as the contents of the NTM messages are processed and the trust level on Vehicle 162 gets propagated from teh MEC to the Ego, the Isolation-ATL and Temporal-ATL on the observations of Object 91 correctly degrade.

3.2.6.4 MBD as a trust source for the TAF at the MEC and at the vehicle

We now move to consider the situation when both the MEC and the Ego are allowed to observe the same trust source. We aim to examine what happens when MBD serves as a trust evidence source both at the MEC and at the vehicle. Notice that the fact that both have access to the same trust source does not imply that both the MEC and the Ego observe the same trust evidence: in fact, the MEC observes MBD reports generated by entities distinct from the Ego (as other vehicles, or the RSU).

In this benchmark, however, we are specifically targeting an experiment where the same MBD trust evidence is considered at both sides, to analyse the impact of double counting. We consider a simple experiment where Vehicle 19 (Ego) receives CPMs from Vehicle 162 (the Attacker) and begins to consistently trigger MBD checks on the observations of Object 91. The MBD activations are used as trust evidence by the vehicle TAF, and are transmitted to the MEC in the form of MBD reports, which are treated as trust evidence, as well. The setup of the experiment is summarized below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of Object 91, and sends its own TCs to the MEC. **Ego (Vehicle 19)**: Receives CPMs from the Attacker (Vehicle 162). Runs MBD checks on each observation.

Trust sources @MEC	The TCs and the MBD trust evidence are available at the MEC's TAF.
Trust sources @Ego	The MBD trust source is available at the Ego.
Considered metric @MEC	Evolution in time of the Temporal-ATL on the Attacker (Vehicle 162) included in the NTM messages. The Temporal-ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.96$ and $\gamma = 0.5$.
Considered metric @Ego	Evolution in time of the Isolation-ATL and of the Temporal-ATL of the observations of Object 91 sent by the Attacker (Vehicle 162). The ATL is a subjective logic vector, consider Belief and Uncertainty components. The parameters for the Temporal-ATL are $\tau = 0.93$ and $\gamma = 0.75$.
Expected outcome	The double counting of the MBD trusting evidence determines a sharp degradation of the Temporal-ATL at the Ego.
Exp: Attack pattern	The Attacker (Vehicle 162) carries on a series of attacks on the observation of Object 91, starting at second 13. The attacks trigger activation of MBD Check 1 at the Ego, and the upload of MBD reports from the Ego to the MEC. See Figure 3.24.

MBD at the MEC and at the Ego benchmark The left-hand side of Figure 3.24 illustrates the evolution of the Temporal-ATLs of Vehicle 162 as evaluated by the MEC in response to the MBD reports evidence, and broadcast in the NTM messages.

At the Ego vehicle, the received NTMs are processed by the Federated TAF to compute the ATLs on the observations of Object 91 received in the CPMs from Vehicle 162. Their evolution is shown in the right-hand side of Figure 3.24, both for the Temporal-ATL and for the Isolation-ATL.

We concentrate on the Isolation-Belief. Notice that the Isolation-Belief exhibits a temporal evolution (between seconds 15 and 24), which is a direct consequence of the evolution of the values in the NTMs. As soon as the Temporal-ATL of Vehicle 162 stabilizes in the NTM (left-hand), so does the Isolation-ATL on the observation of 91 by Vehicle 162 (right-hand). Note that this dynamic evolution is entirely due to the use of Temporal-ATLs as metrics in the NTM.

Consider now the Temporal-Belief evaluated by the Ego (in red on the right-hand side), which degrades in a steeper way than the Isolation-Belief (in green). This dynamic is explained by the superposition of the effects of temporal varying NTMs values, and the use of the Temporal-ATL to assess the kinematic data. It is important to remark the same trust evidence, namely the MBD reports generated by the Ego, is observed and used both by the TAF at the MEC and by the TAF at the vehicle. In this example, the steeper descent of the Temporal-Belief with respect to the Belief is due to double counting of the same trust evidence.

In conclusion, to avoid double counting of trust evidence, when MBD is considered as a trust source both at the vehicle and at the MEC, the Isolation-ATL should be used for comparison with the RTL. Importantly, the inclusion of the Temporal-ATL in the NTM messages ensures that the temporal dynamics of trustworthiness are preserved, allowing the Isolation-ATL to degrade in response to repeated MBD activations; on the other hand avoiding the use of Temporal-ATL at the vehicle prevents double counting of the same trust evidence.

3.2.6.5 Small-scale scene realisation

After benchmarking of the federated architecture, we now turn to the validation of the federated trust assessment through experimentation in a simulated driving scenario. We consider a T-junction scenario,

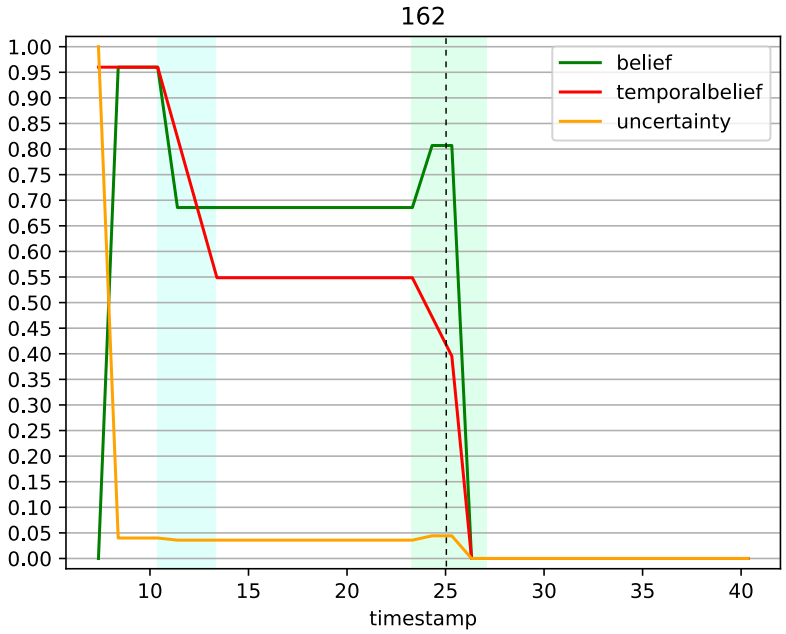


Figure 3.22: Evolution of the ATL of the node 162, as included in the NTM messages broadcast by the NTM app at the MEC.

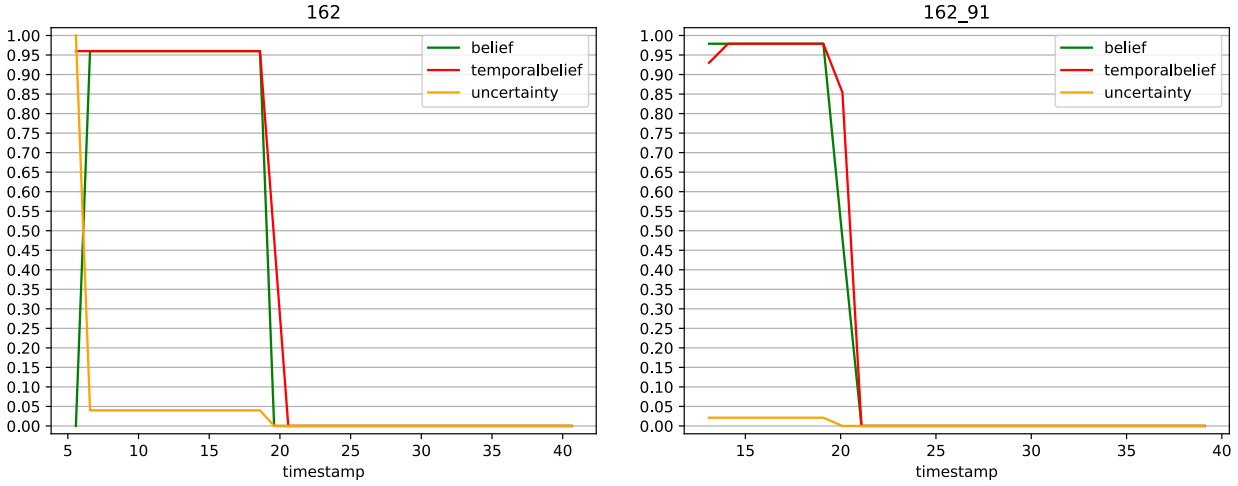


Figure 3.23: On the left, evolution of the Temporal-ATL of the node 162, as included in the NTM messages broadcast by the NTM app at the MEC. On the right, evolution of the ATL and of the Temporal-ATL on the observations of Object 91 generated by the Attacker, as evaluated by the Ego.

depicted in Figure 3.17, very similar to the one considered for the standalone case. This time, the Attacker performs a ghost object attack by including in its CPMs the kinematic data of Object 91, which in reality does not exist. The entities involved in the small-scale scene and their roles are described in Table 3.15. The main events in are described in Table 3.21.

Time	Event description	Highlight
6.093 s	Vehicle 234 (Ego) receives the first CPM from Vehicle 162 (Attacker)	CPMs already contain Object 91
13.093 s	Vehicle 19 (Ego) receives the first CPM from Vehicle 162 (Attacker)	CPMs already contain Object 91.
20.093 s	The first MBD check is activated at the RSU, and a MBD report is sent to the MEC	

22.000 s	Vehicle 162 (Attacker) starts to produce TCs which cannot be verified by the receiver's TAF	This applies only to the test case considering non verifiable TCs
25.093 s	The last MBD check is activated at the RSU, and the last MBD report is sent to the MEC	

Table 3.21: Events in the considered storyline.

The small-scale scene is used to demonstrate the following user stories:

As the vehicle, I want to update the trustworthiness level of the emitter V2X-node in the vehicle TAF whenever I receive a NTM message from the MEC

and

As the NTS Service Provider at the MEC I want to update the Trustworthiness Level of the V2X-node whenever I receive a MBD report or TCs from a vehicle. Keeping the ATL updated with fresh information allows it to deliver a beneficial NTS service.

The setup of the experiment is described in the table below.

Experiment setup	**Attacker (vehicle 162)** Broadcasts CPMs containing the observations of non-exhistent Object 91. It uploads its own TCs to the MEC. **Ego (Vehicle 19)**: Receives CPMs. Runs MBD checks on each observation. **Genuine (Vehicle 234)**: As the Ego, receives CPMs. Runs MBD checks on each observation. **Reporter (RSU)**: Receives CPMs and runs MBD checks on each observation.
Trust sources @MEC	The TCs and the MBD trust evidence are available at the MEC's TAF.
Trust sources @Ego	The MBD trust source is available at the Ego. **In the experiment configuration, no negative MBD evidence is observed at the Ego**.
Considered metric @MEC	Evolution in time of the Temporal-ATL on the Attacker (Vehicle 162), as included in the NTM messages. The parameters for the Temporal-ATL are $\tau = 0.96$ and $\gamma = 0.5$.
Considered metric @Ego	Evolution in time of the Isolation-ATL of the observations of Object 91 sent by the Attacker (Vehicle 162).
Expected outcome	Dynamic evolution of the Temporal-ATL in the NTMs and of the Temporal-ATL at the Vehicles as a reaction to the attack patten.
Exp: Attack pattern	The Attacker (Vehicle 162) carries on a series of attacks by including Object 91 in its CPMs, starting at second 13. The attacks trigger activation of MBD Check 8 at the RSU, starting from second 20, and the upload of MBD reports from the RSU to the MEC. See Figure 3.25.

In the first experiment Vehicle 162 (the Attacker) produces TCs which are always verified by the TAF at the MEC, hence constitute positive trust evidence.

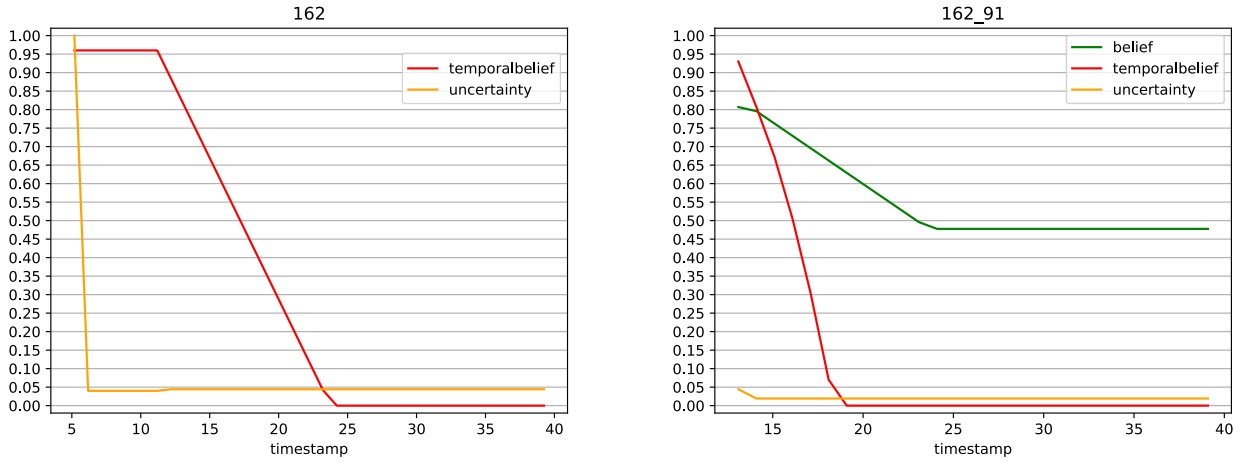


Figure 3.24: On the left: evolution of the Temporal ATL of the node 162, as included in the NTM messages broadcast by the NTM app at the MEC. On the right: evolution of the Isolation-ATL and Temporal ATL of the observation of Object 91 contained in the CPMs from Vehicle 162, as performed by the Ego.

Figure 3.25 presents, at the top, the evolution of the Temporal-ATL on Vehicle 162, contained in the NTM messages broadcast to the vehicles. As it is visible in the plot, the Temporal-Belief starts to degrade as soon as the first MBD report from the RSU is received by the MEC.

The bottom part of Figure 3.25 illustrates the evolution of the trustworthiness level on the observations of Object 91 contained in the CPMs from Vehicle 162, as evaluated by Vehicle 19 (left-hand side) and Vehicle 234 (right-hand side). Recall that neither vehicle observes MBD activations as a result of the attack.

On the left-hand side, we can verify that the ATL on the observations of Object 19 starts to decrease, as soon as the NTM message is received by the federated TAF at Vehicle 19. The Isolation-Belief keeps degrading, as a reaction to the evolution of the Temporal-Belief on Vehicle 162, and Vehicle 19 successfully excludes Object 91 from its extended perception, starting from second 21.

On the right-hand side, we observe the ATL on the observations of Object 91, as assessed by Vehicle 234. As for Vehicle 19 we observe that the ATL starts its decrease, following the evolution of the Temporal-ATL on Vehicle 162, as propagated by the NTM messages. However, we observe in the case of Vehicle 234 there is a 3 second delay in the reaction time, which is at present unexplained.

3.2.6.6 Large-scale scene realisation

As for the standalone scenario, we use the large-scale scene to explore the impact of the federated trust assessment on the accuracy of the extended perception made by the Ego.

We consider the same evaluation setting, experimental traces, and evaluation methodology as done for the standalone large-scale scene, in Section 3.1.6.6.

Experimental traces & Ground Truth dataset The same as considered in Section 3.1.6.6.

Attack model and Attack datasets The same as considered in Section 3.1.6.6.

Evaluation metrics The same as considered in Section 3.1.6.6.

CONNECT datasets Each CONNECT dataset is obtained by replaying the same configuration that generated the Attack dataset, this time having all the connected vehicles playing the role of the Ego

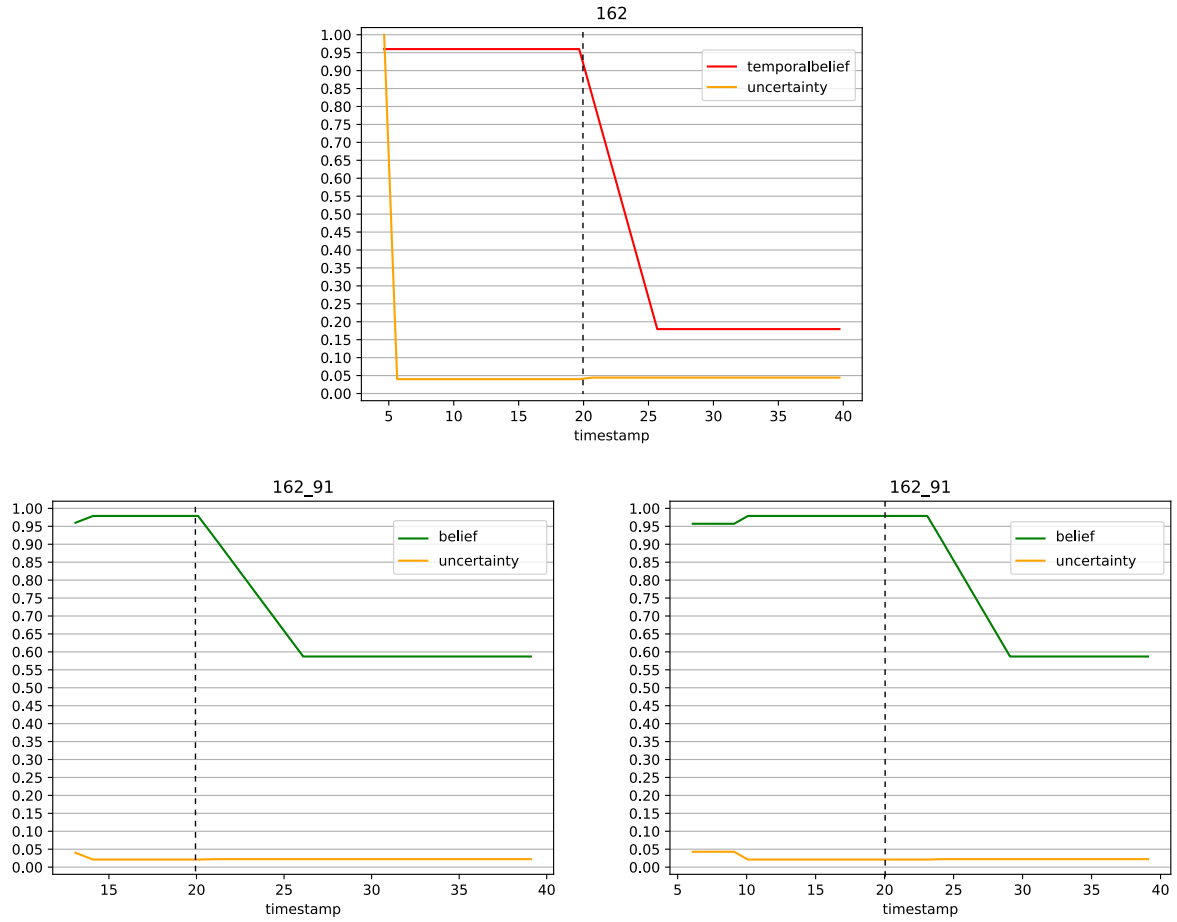


Figure 3.25: Smallscale federated scenario. At the top, the trustworthiness level of Vehicle 162 (Attacker) as included in the NTMs sent by the MEC. At the bottom, left-hand side: the evolution of the ATL of the observation of Object 91, as evaluated by the TAF on Vehicle 19. At the bottom, right-hand side: the evolution of the ATL of the observation of Object 91, as evaluated by the TAF on Vehicle 234.

and apply the CONNECT framework to help the generation of their Extended Perception. Each connected vehicle is equipped with an onboard TAF, which can observe the local MBD trust source. Moreover, the TAF at the MEC can observe TCs and MBD trust sources, and the MEC every second sends NTM messages to the vehicles. As before, the CONNECT dataset is parametrized by the type of attack and by the RTL value used by the vehicles to select the observations to retain in the extended perception. We consider several RTL values, where the Uncertainty Threshold is always fixed to 0.7, and the Belief Threshold takes a value in the set 0.93, 0.85, 0.75.

Evaluation results, Constant Distance Offset attack The purpose of this evaluation is to compare the standalone and the federated scenario, and to achieve this we consider the Constant Distance Offset attack setup, already covered in detail in Section 3.1.6.6 for the standalone scenario.

The left part of Figure 3.26 depicts the number of objects present in the CONNECT federated dataset, in comparison with the CONNECT standalone dataset corresponding to the same parameters. This simple comparison already well illustrates the higher rate of observation exclusion attained with the federated scenario. For the more restrictive RTL (Belief threshold 0.93), the federated trust assessment excludes 20% of the observations, when the total of attacked observations is 23.3%. For comparison, standalone trust assessment for the same RTL excluded only 13.3% of the observations. The increased capability of the federated setting to exclude attacked data is very visible

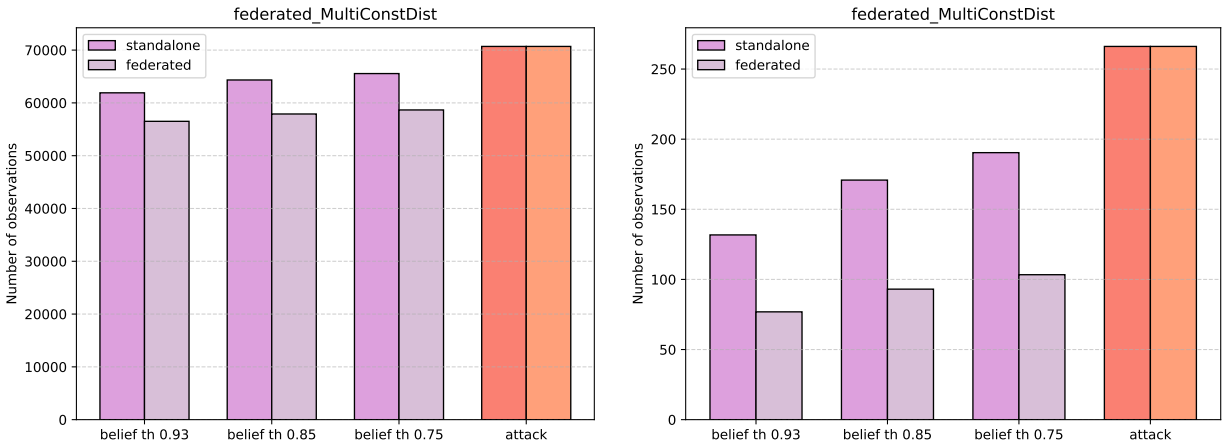


Figure 3.26: On the left, number of observations in the large scale scenario. On the right, Persistence of attacked observations in the large scale scenario.

also considering the persistence metric, in the right part of Figure 3.26. For the same level of Belief threshold, the persistence of attacked observations in the CONNECT dataset is much lower in the federated than in the standalone scenario.

We now move on to assess the effects of the altered observation that have not been excluded as a result of the trust decision. Figure 3.27 shows the sample mean of the distance, comparing the results of the standalone and federated scenario. As expected in consideration of the increased exclusion capability, the impact of the remaining altered observations is minimized by the federated setup. To sharpen the analysis, we also compare the mean of the distance, this time evaluated only on the attacked observations. The results are shown in Figure 3.28, and allows us to deduce that the federated setup is efficient in excluding those altered observations whose impact on the accuracy of the extended perception is higher.

3.2.7 Discussion & Critique - Lessons Learnt

3.2.7.1 Evaluation results discussion

In the context of the IMA use case, the performance of the CONNECT framework has been mostly assessed in terms of the accuracy of the extended perception it provides, in presence of data alteration attacks.

The comparison of the results in the standalone and federated scenarios allows us to conclude that leveraging the infrastructure for the trust assessment process, even when the trust decision takes ultimately place at the vehicle, may bring substantial gains. In our experimental activity, we concentrated in assessing the gain in the IMA use case with respect to the exploitation of the MBD trust source. As expected, the impact of this MBD trust source in enabling the early exclusion of data from attacker vehicles is multiplied when it is exploited on the MEC side, because of the increased volume of evidence available there. Another interesting effect noticeable when inspecting the results is that relying on the MEC for trust assessment of the V2X-nodes makes the overall performance much less dependent on the choice of RTL value used by the vehicle to retain observations in the extended perception. This is mainly due to the fact that, being able to observe all negative MBD evidence produced in the coverage region, the TAF at the MEC is able to produce negative opinions on attacker vehicles early on. In this way the Ego may respond correctly even to its very first interactions with the attacker.

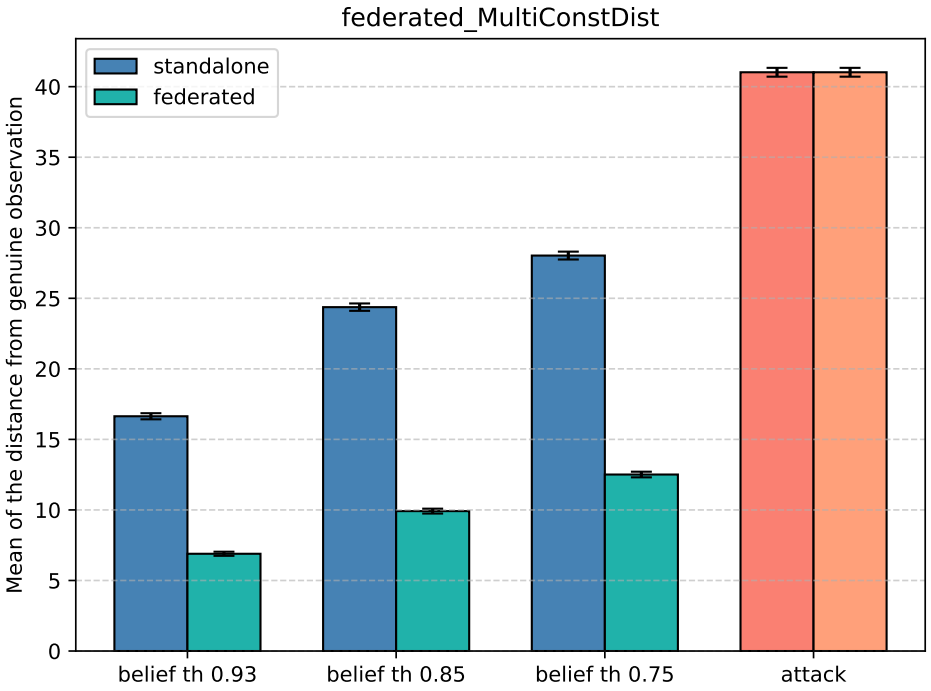


Figure 3.27: Sample mean of the distance between the observation in the extended perception and the ground truth in the large scale scenario, for the MultConstDist attack.

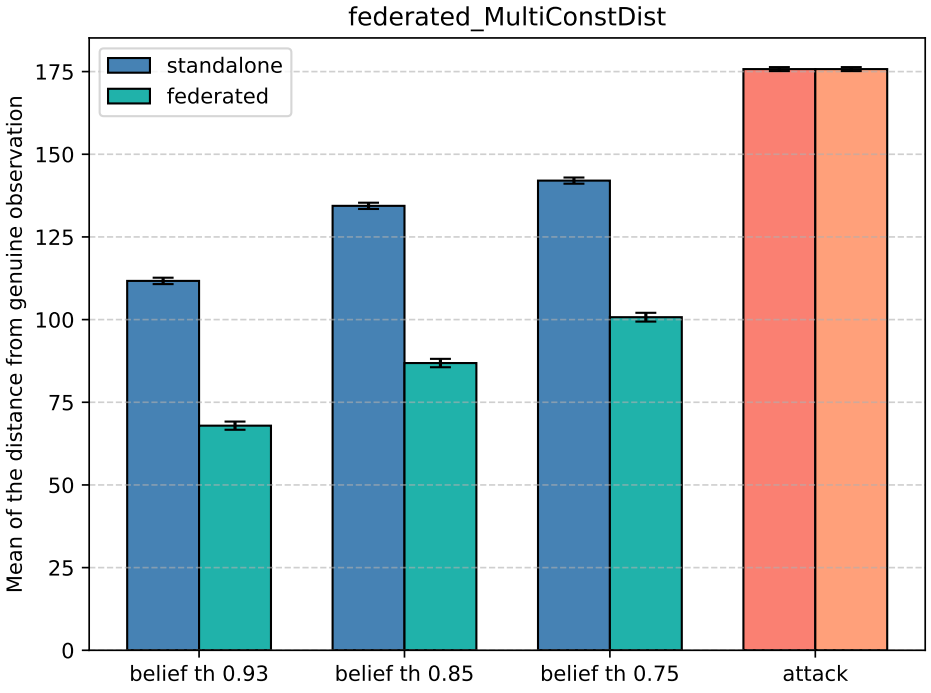


Figure 3.28: Sample mean of the distance between the observation in the extended perception and the ground truth in the large scale scenario, considering only attacked observations, for the MultConstDist attack.

As for the standalone scenario, we largely concentrated in the evaluation activity on setups considering only positive TCs evidence. This is justified by the fact that the use of TCs as a trust source impact the performance of the CONNECT framework within the IMA use case in a clear, intuitive, well understood manner, while the correct interpretation of the role of the MBD trust source required a more substantial effort.

3.2.7.2 Temporal-ATL in NTM messages

As previously discussed in the standalone scenario, the Isolation-ATL is limited in capturing situations where the key factor is the repeated occurrence of negative trust evidence. Also in the federated setup this limitation becomes evident when considering the MBD trust source, especially at the MEC. In the federated architecture, in fact, the system may collect not only the temporal accumulation of MBD evidence from a single vehicle, but also corroborating reports issued by multiple vehicles.

So being able to capture the effects of the accumulated evidence using the Temporal-ATL is crucial to the ability of the TAF at the MEC to provide meaningful and useful opinions to the TAF at the Ego vehicle.

3.2.7.3 Double counting of evidence

Benchmarking the behavior of the federated architecture when MBD is considered as a trust source both at the MEC and at the vehicle has highlighted a risk that is intrinsic to the federation setup: the potential for double counting of the same trust evidence.

The core principle of the federated architecture is to enable trust evidence gathered at different locations to contribute to a unified trust assessment. Within the IMA use case, the trust decision is ultimately performed at the vehicle. From this perspective, allowing MBD to serve as a trust source both at the MEC and at the vehicle is not only acceptable, but it is advantageous, as it allows the Ego vehicle to benefit from MBD evidence generated by other entities on the road in addition to the MBD evidence generated by itself. This is demonstrated in the small-scale scene, where two benign vehicles, unable to produce conclusive MBD evidence locally, can still leverage the negative MBD evidence generated by the RSU thanks to the NTM messages generated by the MEC on its basis.

However, this requires careful handling: we want to avoid the situation where the same piece of MBD evidence, produced locally by the Ego vehicle and sent to the MEC in the form of an MBD report, is used twice to inform the trust decision on an observation. In the context of the IMA use case, this risk is avoided by sending Temporal-ATL opinions in NTM messages, while comparing the Ego's Isolation-ATL to the RTL when making the trust decision at the Ego. In this way, we ensure that the freshest MBD evidence locally generated on the target observation is exploited by the TAF at the vehicle only. As discussed, using the NTM messages as a federating mechanism incurs propagation delays of the opinions on the V2X-nodes from the MEC to the Ego; in this way, at the moment of the trust decision, the opinion on the sending V2X-node does not account for the freshest MBD evidence, yet.

Notice that with this solution the task of encoding the reaction to repeated observations of the negative MBD trust evidence is entirely delegated to the MEC. The MEC is the natural choice for this role, as it enables the aggregation of trust evidence from multiple entities, thus maximizing its impact by participating to the trust assessment process of all connected vehicles.

These observations constitute very important learned lessons to guide the correct design of federated trust assessment constellations, whenever the same piece of trust evidence may be observed by different entities.

Chapter 4

Demonstrator #2: Cooperative Adaptive Cruise Control (C-ACC)

Cooperative Adaptive Cruise Control (C-ACC) represents a significant advancement in intelligent transportation by leveraging real-time vehicle communication to coordinate driving behaviours, thereby improving traffic flow and safety beyond the capabilities of traditional Adaptive Cruise Control [MSS+13]. This system relies on data from both onboard sensors and surrounding vehicles, exchanging critical messages to facilitate synchronized and safe operation, enhance traffic efficiency through closer vehicle spacing and synchronized speeds, and increase roadway capacity. **The trustworthiness of both technical components and the exchanged data is crucial.** The dynamic assessment of both node- and data-centric trust enables the system to react with predefined safe responses to detected anomalies, aiming to mitigate attacks. The potential for malicious or manipulated information highlights the critical need to assess the trustworthiness of C-ACC systems to safeguard passenger safety.

As detailed in both D3.3 [Con25c] and D6.1 [Con24b], a TAF can perform trustworthiness assessments in two distinct modes: i) subscription-based and ii) pull-based. In the *subscription-based mode*, an application subscribes to TAF to receive regular trustworthiness assessments. This occurs whenever a trust source changes its status. In the *pull-based mode*, however, the application must request a trustworthiness assessment from TAF on demand. This means that the application will not receive notifications when there is a change in the status of trust evidence. The pull-based mode offers the advantage of allowing the application to decide when to assess trustworthiness and whether TAF should use cached data or, when necessary, only fresh trust evidence.

The construction of the target use case revolves around **assessing the trustworthiness of in-vehicle components during runtime**; **responding to changes in the trust level of featured data sources (e.g., *electronic control unit (ECU)*)**; and **reaction time in taking appropriate actions**. Based on these changes, different instances of the C-ACC execution logic can operate on separate in-vehicle computers. In this use case, we refer to an in-vehicle computer or ECU specifically as an ”in-vehicle computer” rather than an ”on-board unit.” This distinction is made because we are solely focusing on components that are part of the vehicle itself, excluding any other elements that are not part of the ego vehicle composition. These are considered as communication units in this scenario and therefore, treated as black boxes.

In the sections that follow, we consider two driving scenarios or driving situations that differ in the time available for trust assessment, based on decision-making speed requirements, and whether cached data is used:

- **Scenario 1: Imminent driving situation** - The C-ACC functionality is used in scenarios involving imminent driving situations, allowing *cached data or the TAF's subscription API* for quick decision-making. In this scenario, we assess an imminent driving situation that requires the C-ACC item to make quick decisions. This involves evaluating the trustworthiness of the host in-vehicle computer,

where the C-ACC item is running, to ensure that it is operating on a trustworthy system. The goal is to maintain a safe distance from the vehicle ahead to prevent accidents.

To promote quick decision-making, it can request on-demand trust assessments, including cached data, to use existing trust evidence without the need for fresh evidence collection, which would lead to additional overhead. As an alternative, the C-ACC item can utilise the TAF's subscription API to continuously receive updated trustworthiness assessments when a trustworthiness status changes. With this, the provided ATL is calculated based on the TAF's internal schedule for verifying the trust sources, acknowledging that the assessment result may be based on potentially outdated information. The trust evidence includes secure boot, access control, application isolation, control flow integrity, and configuration integrity verification. These elements are derived from integrity checks conducted on the in-vehicle computers that are part of the use case architecture, as detailed in Section 4.1.2.

- **Scenario 2: Upcoming driving situation** - In this scenario, an upcoming driving situation is assumed, where the vehicle has enough time to request trustworthiness assessments based on up-to-date trust evidence, independent of the subscription notifications or cached data. Instead of awaiting updates from the TAF based on periodically checking trust evidence or using cached data, the C-ACC item can request an *up-to-date assessment* when needed.

  This scenario is important for analysing how TAF handles such a request and for measuring the time needed to provide an up-to-date ATL, including collecting fresh evidence, calculating the ATL, and sending it to the C-ACC item. Similar to Scenario 1, the trust evidence includes secure boot, access control, application isolation, control flow integrity, and configuration integrity verification. These elements are derived from integrity checks conducted on the in-vehicle computers that are part of the architecture for this use case, as detailed in Section 4.2.2.

The present deliverable builds upon the initial validation activities presented in D6.1 [Con24b], which centred on the subscription-based (push) trust assessment. The evaluation of Scenario 1, for instance, was approached in D6.1 using this subscription model, and for completeness, a summary of those findings is recapitulated in Section 4.1.8.1. Conversely, Scenario 2 was deliberately excluded from the initial evaluation phase. Its strict requirement for the most up-to-date evidence collection renders it incompatible with the subscription-based approach, which provides periodic updates rather than on-demand data. This distinction directly motivates the shift in our experimental methodology. **Consequently, the experimental focus in D6.2 transitions to the pull-based (on demand) trust assessment approach**. This methodology is not only essential for handling the on-demand data requirements of Scenario 2 but is also fully applicable to Scenario 1, thereby enabling a consistent and comprehensive evaluation of the integrated *CONNECT* framework. In this case, the TAF responds to requests for trust assessments, where the C-ACC item demands an assessment to be used for either imminent or upcoming driving situations. The main difference between these two driving situations is the potential use of cached evidence data. Further details about both scenarios can be found in the following sections, with section 4.1 and 4.2, covering their respective evaluations. The activation of C-ACC was evaluated only in D6.1, as there were no changes in the process following the trust assessment. Therefore, previous benchmarks related to the activation of C-ACC on a new host remain applicable.

The inclusion of kinematic sensors as an additional trust source was evaluated as a potential method for increasing the complexity of the trust model. However, given that the TAF's performance has already been extensively benchmarked in other complex scenarios, this line of experimentation was not pursued to avoid redundant evaluation. This decision enabled us to shift our experimental focus from re-validating established capabilities to exploring other research questions. Instead of testing how the TAF generates a trust assessment from other sources like sensor data, our evaluation shifted to i) how the TAF reacts upon receiving new trustworthiness evidence from sensors and ii) how the trust level itself behaves dynamically. Specifically, we first measured the C-ACC application's response time to a new Actual Trustworthiness Level (ATL) derived from kinematic data. This directly assesses the application-level impact of a change

in sensor trustworthiness. Secondly, we investigated the temporal dynamics of the ATL, analysing its evolution in response to a continuous stream of trust evidence. The detailed methodologies and results for these new experiments are presented in Section 4.3.

Table 4.1 highlights the main differences between the evaluation reported in Deliverable D6.1 [Con24b] and this final release, D6.2.

Table 4.1: CONNECT Framework C-ACC Evaluations between the different Releases

CONNECT C-ACC Evaluations in Release D6.1	C-ACC Evaluations in Final Release of *CONNECT*
C-ACC was designed to use subscription-based trust assessments that are applied to imminent driving situations. The C-ACC function relied on the TAF schedule to automatically receive up-to-date trust assessments related to in-vehicle computers. Notifications from the TAF were sent to the C-ACC whenever there was an update in the status of trust evidence, eliminating the need for the C-ACC to make requests. This mode is applicable only in imminent driving situations, as the application does not control when the trust assessment will be received or if cached data was used.	C-ACC was designed to utilize pull-based trust assessments that can be applied to both imminent and upcoming driving situations. In this mode, C-ACC requests a trust assessment as needed. This approach meets the requirements of Scenario 1 by allowing the use of cached data for quick trust assessments. It also addresses the needs of Scenario 2, where cached data is not permitted, and only freshly collected data is considered for the trust assessment.
AIV was emulated to evaluate TAF's behaviour using various trust source combinations. Emulating AIV allowed for better control over the type of evidence and the frequency of transmissions.	AIV collects integrity evidence by using actual controls and integrity checks from the operating system on each in-vehicle computer to generate its reports.
Benchmarks addressing subscription-based trust assessments for scenario 1 and C-ACC main function activation.	Benchmarks addressing pull-based trust assessments for both scenario 1 and 2; Benchmarks for C-ACC's response to changes in sensor data trustworthiness; C-ACC activation in different hosts was not re-evaluated because there was no change in the flow after trustworthiness assessment.

Although scenarios 1 and 2 share the same testbed and trust model, they differ in how they manage cached trust evidence. The acceptance of cached trust evidence is specified in the request message sent to TAF, which reflects the particular needs of each scenario. Detailed description, KPIs, and acceptance criteria are defined for each scenario in their respective sections, 4.1 and 4.2.

4.1 Scenario #1: Imminent Driving Situation

4.1.1 Description

In this scenario, we evaluate an imminent driving situation which requires C-ACC to make decisions quickly to avoid an accident while using the C-ACC function. Assessing trust can support C-ACC to evaluate whether an in-vehicle computer that runs the C-ACC Main Function is trustworthy or not; if considered untrustworthy, the system can trigger a migration of this function to another in-vehicle computer that has been characterised as trustworthy, against the already defined Required Trust Level (RTL).

To support such quick decision-making, it can request on-demand trust assessments using cached data to leverage existing evidence information without the need for fresh evidence collection. As an alternative, C-ACC can utilise the TAF's subscription API to receive updated trustworthiness assessments when there is

a change. IN this case, the ATL provided is calculated based on the TAF's internal schedule for verifying the trust sources, acknowledging that the assessment result may be based on potentially outdated information.

Initially, this scenario was evaluated in Deliverable D6.1 [Con24b], in the context of subscription-based trust assessments. This previous evaluation is recapitulated in Section 4.1.8.1, and it was shown that TAF responded satisfactorily under this approach. In the previous evaluation, the creation of AIV reports followed a pre-set configuration, as a design choice for allowing higher flexibility in stress testing the TAF and to consider different values for the integrity attributes. The current implementation collects actual integrity checks on the in-vehicle computers, as detailed in Section 4.1.3.

In the current evaluation, AIV can gather real-time trust evidence from the in-vehicle computers and generate a report that reflects the system's status. Additionally, the scenario is assessed using a pull-based trust evaluation method. This allows the C-ACC Migration Process to request an assessment as needed, without waiting for the TAF to send updates. In this case, TAF is enabled to consider cached data, ensuring a timely response to the application's requests.

The current workflow is illustrated in Figure 4.1. The C-ACC Migration Process begins with a pull-based trust assessment, enabling the TAF to use evidence that is either cached or recently collected, thereby preventing delays. The performed benchmarks focus on trustworthiness requests, as the C-ACC migration was evaluated in D6.1, and the migration flow remains unchanged; therefore, results are still valid and summarised in Section 4.1.8.1. We benchmarked the time taken for a request from the C-ACC Migration Process to be sent to the TAF until the request response was received. This time considers the collection of trust evidence, the AIV report when needed, and the internal TAF process.

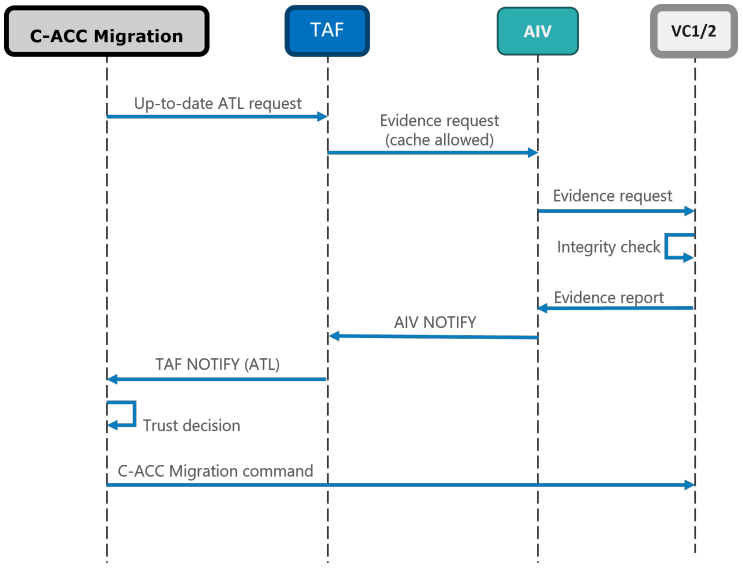


Figure 4.1: Workflow for on-demand imminent driving situation.

4.1.2 Setup - Topology - Testbed details

This use case is implemented as a hardware-in-the-loop simulation, as illustrated in the architecture shown in Figure 4.2. The architecture consists of the following components:

- **In-vehicle Computers**: They are Linux Intel-based computers with 64 GB of RAM, eight-core Intel Xeon Processor running at 2.70 GHz with 15MB Cache (67W) each. Their function is to emulate an in-vehicle computer, running the C-ACC Main Function, and send speed and brake commands to the SUMO traffic simulator. The in-vehicle computers only control the ego vehicle. The demonstrator's architecture includes two of them: VC1 and VC2. Both in-vehicle computers have an instance of the

C-ACC Main Function and can provide trust evidence to the TAF by assessing their trustworthiness status for each claimed trust evidence. In addition, VC2 hosts the AIV, TAF, and the C-ACC Migration Process. These three extra elements running on VC2 are a design decision to simplify the simulation architecture; they could have been running on any in-vehicle computer, even a third one. Both in-vehicle computers are able to control the Ego vehicle, but only one is active at any one time.

- **C-ACC item**: It is composed by two components:
  - **C-ACC Main Function**: This is responsible for C-ACC's driving functionality. It includes processing the received and perceived data, calculating the minimum safe distance between the Ego vehicle and its leading vehicle, and sending corresponding braking and acceleration commands concerning the ego vehicle in the simulation. In this simulation, we assume that the C-ACC components are secure and only focus on assessing in-vehicle trustworthiness.
  - **C-ACC Migration Process**: This component ensures that the C-ACC Main Function is executed on a trustworthy in-vehicle computer. Based on the in-vehicle computer's Actual Trustworthiness Level (ATL), the C-ACC Activation Process decides where the C-ACC main Function is running, e.g., if the in-vehicle computer VC1's ATL is not sufficient when compared with the refined RTL, the C-ACC Activation Process will look for another trustworthy in-vehicle computer to run a new instance of the C-ACC Main Function on. As previously mentioned, in this simulation, we assume that the C-ACC components are secure.
- **Trust evidence**: Instances of trust evidence indicating how trustworthy the in-vehicle computer is. They are used as a basis to calculate the ATL of each in-vehicle computer. For this use case, access control, secure over-the-air updates, secure boot, application isolation, control flow integrity, and configuration integrity verification are required in all trustworthiness assessments. Each in-vehicle computer is capable of providing an assessment for each required piece of evidence by running operating system integrity verification.
- **Traffic simulator - SUMO**: SUMO is an open-source traffic simulation tool. It simulates the vehicles involved in the demonstration and provides the vehicle's sensor data, such as speed and distance between the ego vehicle and the one ahead. Through the tool, an in-vehicle computer can send speed and braking commands to the simulated traffic, based on the C-ACC Main Function output, enabling visualisation of each vehicle's behaviours.
- **TAF**: TAF is in charge of collecting trust evidence for both in-vehicle computers, as well as calculating their corresponding ATLs. It can evaluate the trustworthiness level of each computer. Although TAF is currently located on VC2 for demonstration purposes, it can operate on any capable in-vehicle computer. In this simulation, we assume that the TAF is secure and only focus on assessing in-vehicle trustworthiness.
- **AIV**: This component is responsible for creating integrity reports about each in-vehicle computer and sending them to TAF. In the use case, AIV is accountable for collecting trust evidence when TAF requests it or upon TAF's subscription. AIV requests that each in-vehicle computer collect trust evidence and create a report. AIV reports cover access control, secure over-the-air updates, secure boot, application isolation, control flow integrity, and configuration integrity verification. AIV initiates actual integrity verification on each in-vehicle computer by sending SSH commands and by using operating system tools, such as AppArmor, to support this integrity verification. Communication with TAF is conducted via Kafka.

Two in-vehicle computers, which are capable of executing the C-ACC functionality, manage the ego vehicle in the simulation. These computers are connected to a traffic simulator called SUMO (Simulation of Urban Mobility, https://eclipse.dev/sumo/). SUMO generates sensor data for each simulated vehicle and provides full control over their behaviour, including steering and acceleration commands.

The in-vehicle computers are responsible solely for controlling the ego vehicle, while SUMO manages the other simulated vehicles. SUMO operates on an external computer and is utilised only for traffic visualisation. The remaining software components are distributed between the two in-vehicle computers. Kafka is used for communication among TAF, AIV, C-ACC components, and SUMO. Additionally, SSH is used to send commands to each in-vehicle computer, such as script executions, when a migration is triggered or AIV requests trust evidence.

While it is technically possible for the migration or activation of a new instance of the C-ACC to happen on a different vehicle rather than just between in-vehicle computers, such a scenario is not practical. This is due to concerns about accountability and latency, which are not adequate in safety-related applications.

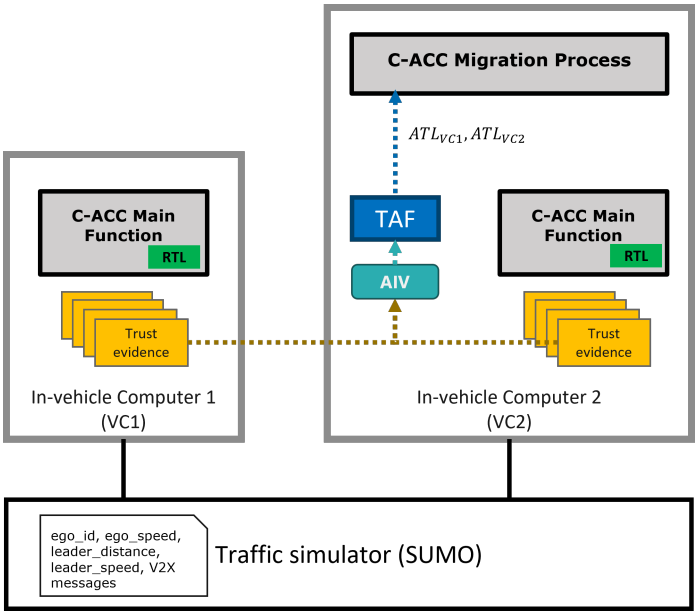
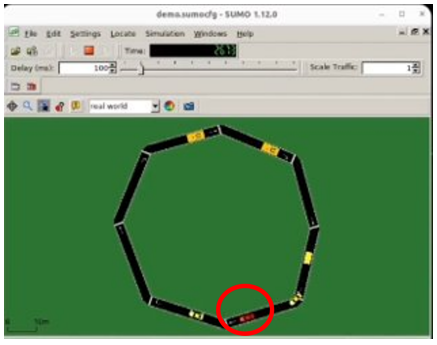
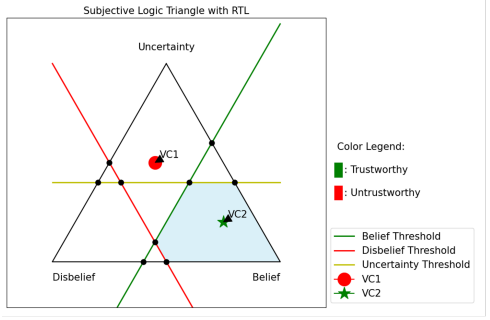


Figure 4.2: Demonstrator architecture



(a) SUMO with red vehicle controlled by VC1 or VC2.

(b) VC1 and VC2, with LED status indicators.



Trust Source Status

	ACCESS CONTROL	SECURE OTA	SECURE BOOT	APPLICATION ISOLATION	CONTROL FLOW INTEGRITY	CONFIGURATION INTEGRITY VERIFICATION
VC1	0	1	1	1	1	1
VC2	1	1	1	1	1	1

(c) Example of the dynamic trustworthiness status visualisation.

Figure 4.3: In-vehicle use case test bed.

Figure 4.3 illustrates the testbed used for simulating and evaluating the in-vehicle use case. The scenario involves three simulated vehicles operating on a circular road, which is modelled using SUMO. Each vehicle is equipped with a C-ACC Main Function that helps maintain a safe distance between them. One of the vehicles is designated as the ego vehicle, which can be controlled by two in-vehicle computers. The other two vehicles are managed by the SUMO software and scripts running on an external computer to ensure they maintain a safe distance and facilitate smooth traffic flow.

The ego vehicle is highlighted in red in Figure 4.3a and can be controlled externally by the two in-vehicle computers, VC1 and VC2, which are equipped with status LEDs, as shown in Figure 4.3b. Additionally, Figure 4.3c presents a dynamic visualisation of trustworthiness status. This visualisation displays the status of each trust source involved and indicates the trustworthiness of each vehicle through its ATL located in the subjective logic triangle. When an in-vehicle computer's ATL is within the boundaries established by the RTL, it is displayed as trustworthy in green; otherwise, it appears untrustworthy in red.

A key characteristic of this scenario is the allowance for cached data during the trustworthiness assessment. The AIV will not always be used to provide the evidence report; it will only be used once the previous AIV report is considered outdated. As a consequence, the TAF response can vary: when fresh AIV evidence is not needed, the TAF responses are faster.

4.1.3 Trust Model

The trustworthiness of each of the in-vehicle computers is assessed based on the trust model shown in Figure 4.4. The root node of this Trust Model is the TAF, and the only other two trust objects are the two leaf nodes representing the in-vehicle computers, VC_1 and VC_2. As such, there are only two trust relationships and trust opinions in this trust model, the trust relationship between the TAF and VC_1, assessed in the

form of trust opinion $\omega_{VC_1}^{TAF}$, and the trust relationship between the TAF and VC_2, assessed in form of trust opinion $\omega_{VC_2}^{TAF}$. The semantics of these trust opinions are given in the subsection below.

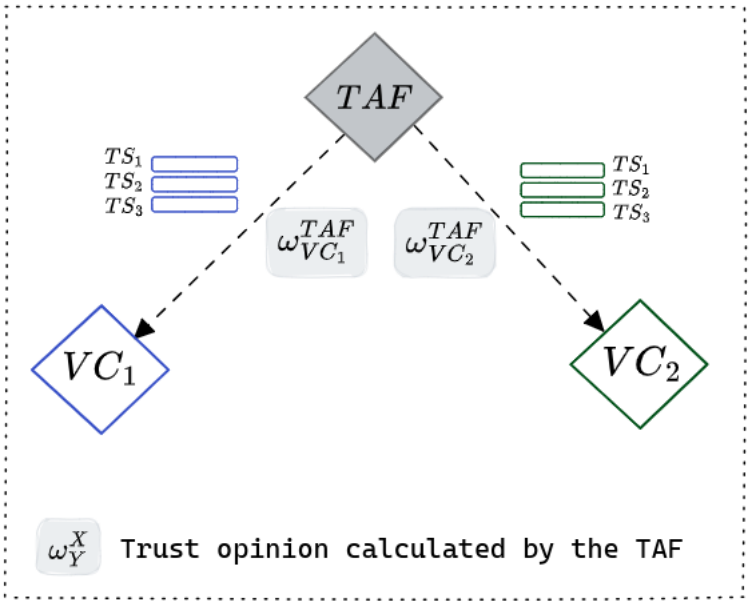


Figure 4.4: Trust Model for in-vehicle computer trustworthiness assessment.

4.1.3.1 Trust Opinions

There are the following trust opinions in this trust model:

1. $\omega_{VC_1}^{TAF}$ = trust opinion assessed by the **TAF** on the trustworthiness of the **in-vehicle computer 1**, VC_1, to not compromise the C-ACC application running on it.
2. $\omega_{VC_2}^{TAF}$ = trust opinion assessed by the **TAF** on the trustworthiness of the **in-vehicle computer 2**, VC_2, to not compromise the C-ACC application running on it.

4.1.3.2 Trust Sources and Evidence

Given that this is a standalone TAF running on the ego vehicle, concerning itself only with applications running locally and not needing external input, it does not request trust opinions or evidence from external entities. The TAF only requests evidence or subscribes to evidence from entities on board the ego vehicle.

1. $\omega_{VC_1}^{TAF}$ $\rightarrow$ the **TAF** assesses this opinion based on the evidence it receives from the AIV about the trustworthiness of the in-vehicle computer 1 in the form of a trustworthiness claim.
   - **Trust Source** = AIV
   - **Evidence** = Trustworthiness claims - Trustworthiness claims include the output of the single security controls if they are not implemented (value: -1), are implemented and detected something (value: 0), are implemented and did not detect anything (value: 1), or are not verifiable (value: -2). AIV considers actual integrity verification.
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the trustworthiness claims is described in D6.1 [Con24b] and D3.3 [Con25c].

2. $\omega_{VC_2}^{TAF} \rightarrow$ the **TAF** assesses this opinion based on the evidence it receives from the AIV about the trustworthiness of the in-vehicle computer 2 in the form of a trustworthiness claim.

    - **Trust Source** = AIV
    - **Evidence** = Trustworthiness claims - Trustworthiness claims include the output of the single security controls if they are not implemented (value: -1), are implemented and detected something (value: 0), are implemented and did not detect anything (value: 1), or are not verifiable (value: -2). AIV considers actual integrity verification.
    - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the trustworthiness claims is described in D6.1 [Con24b] and D3.3 [Con25c].

4.1.3.3 Type of Evidence Collected

In the context of this scenario, the TAF operates upon the reception of *trustworthiness claims*, capturing the current **configuration and operational state** of all viewed objects (in-vehicle computers). Configuration state is the set of writable data that enables the transition of the target system from its default state into its current state, in a verifiable manner. For instance, all evidence produced by the **Configuration Integrity Verification (CIV) process** depicting the current configuration of the entire software stack being deployed in a Vehicle Computer, builds upon the monitoring and introspection of both the *permissions* of memory regions that a process is loaded into and the structure of that memory. This is achieved by parsing a process's Executable and Linkable Format (ELF) file, the standard format for Linux binaries. By inspecting the ELF header and its associated program headers at load time, the CONNECT Tracer precisely identifies the intended boundaries of legitimate executable code segments (such as '.text'). This deep inspection of the ELF structure post-loading (as part of the traces that are then consumed by the Attestation and Integrity Verification Component, as part of the CIV process [Con25d]) ensures that memory regions are correctly mapped and that critical structures like symbol tables and relocations align with the original binary's logic. *By continuously verifying the permissions and structural integrity of these critical memory regions, the CIV can produce evidence of any detected violations that can then be interpreted by the TAF to a trust opinion on the trustworthiness of the system's current state.*

In addition to that, the AIV also - through the CONNECT Tracer - monitors whether additional safeguards such as *Control Flow Integrity (CFI)* have been activated in the system. This is done by periodically checking the configuration parameters of the underlying OS (Linux Kernel) so as to validate that they have been activated (and not tampered with) and that the target binary has been executed by invoking the correct binary source (execution path). The monitoring of the evidence produced by these two security controls can detect malevolent attempts that try to exploit memory-related vulnerabilities towards disrupting the static integrity of the code base or altering the runtime configuration and control flow of the target system.

We have to highlight, though, the need for such evidence to be monitored and provided to the TAF in a verifiable manner [Con24d, Con25d]. Hence, all tracing extensions capable of capturing such *state information* need to be part of CONNECT's Trusted Computing Base. While the CFI security control is considered trusted by-default, as it is exposed through the kernel, the Attestation Agent and Tracer components (supporting the AIV) need to be launched as trusted apps running in isolation as part of the local trusted execution environment; e.g., Gramine SGX. Considering that the overhead posed by the enclavization of these processes has been extensively evaluated [Con25d], in what follows, we emulate the integrity-related evidence produced by the CIV as a collection of *hash values* representing either the correct or incorrect state of the target device. This allowed better control of the manifestation of the TAF with pre-configured values that represent positive and/or negative attestation results, which, in turn, enabled the more focused evaluation of the TAF's behaviour.

Furthermore, we extended the AIV to also consider additional evidence that can be available through the integration of well-established Linux tools and integrity checks: Access Control, Secure Over The

Air (OTA) updates, Secure Boot, and Application Isolation. Besides information produced by the CIV and CFI safeguards, for actual verification, AIV requests that each in-vehicle computer confirms also the following: (i) To verify the claim *Access Control*, AIV requests verification of AppArmor logs, a Linux kernel security model that implements mandatory access control;(ii) To verify the claim *OTA updates*, AIV requests verification to ensure that OTA update mechanisms are securely configured and that the system is up-to-date; (iii) To verify the claim of *Secure Boot*, AIV requests a check to ensure that the in-vehicle computers have booted with Secure Boot enabled and that no unsigned kernel modules are loaded. The verification process checks whether the system is based on the Unified Extensible Firmware Interface (UEFI) and utilises mokutil, a command-line utility in Linux that manages Machine Owner Keys (MOKs) for UEFI Secure Boot; and (iv) To verify the claim of *Application Isolation*, AIV requests verification of the activation of the Uncomplicated Firewall (UFW), which is responsible for managing firewall rules. Additionally, reviewing the log file to check for any error messages.

4.1.4 Actual Trustworthiness Level

In this case, no additional steps are needed to obtain the ATLs as there are only direct trust relationships between the root node (TAF) and the leaf nodes (VC_1 and VC_2). Therefore:

$$ATL_{VC_1} = \omega_{VC_1}^{TAF} \tag{4.1}$$

Similarly:

$$ATL_{VC_2} = \omega_{VC_2}^{TAF} \tag{4.2}$$

4.1.5 Required Trustworthiness Level

Defining a Required Trustworthiness Level (RTL) is essential when establishing a solid foundation for making decisions. The RTL is determined through assessments of risk, impact, and detectability, as outlined in D3.3 [Con25c]. Although there is still a difference between RTL and ATL methodologies, and they have not yet converged into a single method, it has been observed that both approaches can be applied to this use case. This is because both ATL and RTL use the same risk assessment processes and consider the impacts of the same cybersecurity controls, reflecting the expected behaviour based on design requirements. Section 4.3.1 discusses the evaluation of RTL.

The RTL is defined for this use case to be used in both scenarios. The methods described in Deliverable D3.3 [Con25c] are used to define belief, disbelief, and uncertainty thresholds. For that, a TARA was performed, and Tables 4.2 and 4.3 summarise the main output to be used in the RTL definition. These tables are in their condensed version, presenting only the relevant output for RTL definition.

For this use case, we are interested in assessing the trustworthiness of the integrity of the in-vehicle computers while running a C-ACC Main Function; in other words, we focused on assessing the integrity of VC1 and VC2. Based on a performed TARA, the following damage scenarios were considered for this use case and scope, and their assigned impacts are shown in Table 4.2. We have considered damage scenarios that can be caused by an integrity violation in the in-vehicle computers, as this is within the scope of the trustworthiness assessment.

1. **DS.2: Loss of C-ACC functionality.**
2. **DS.3: Incorrect C-ACC behaviour.**
3. **DS.5: The in-vehicle computer is not responding as expected.**

These damage scenarios were considered as being relevant for the C-ACC functionalities, as they may impact user safety and system operation. By analysing these damage scenarios, we will define the threats that can lead to them and derive their associated risk levels. In the context of TARA, a Damage Scenario describes the consequences for road users or other stakeholders if a cybersecurity property of an asset is compromised (e.g., loss of C-ACC functionality leading to a collision). Contrarily, a Risk Scenario is a broader concept that contains the entire chain of events, including identifying the threat itself, analysing the attack path (i.e., assessing the feasibility of the attack), and linking it to the resulting damage scenario to determine the overall risk level. Therefore, the damage scenario details the ”what if it breaks?” while the risk scenario builds the ”how could it break and what would happen?” narrative. For this analysis, availability (A) and integrity (I) concerns were assessed.

Table 4.2: Impact ratings for damage scenarios (DS) related to security (S), financial (F), operational (O), and privacy (P), concerning availability (A) and integrity (I).

		Impact			
ID	**Concerns**	**S**	**F**	**O**	**P**
DS.2	A: VC1_C-ACC_Main_Function A: VC2_C-ACC_Main_Function I: VC1_C-ACC_Main_Function I: VC2_C-ACC_Main_Function A: Distance_data A: Speed_data A: Perception_data A: Speed_S A: Distance_S	Moderate	Negligible	Moderate	Negligible
DS.3	I: VC1_C-ACC_Main_Function I: VC2_C-ACC_Main_Function I: C-ACC_Actor I: Driving_command	Major	Moderate	Major	Negligible
DS.5	I: VC1 A: VC1 I: VC2 A: VC2	Moderate	Moderate	Major	Negligible

Table 4.3: Risk levels have been assigned to each risk resulting from threat scenarios (TS) related to the scope.

ID	Caused by	DS	Risk Level
R.1	TS.1: Exploitation of software weaknesses on in-vehicle computers or C-ACC_Main_Function.	DS. 2, DS. 3	3
R.4	TS.4: Tampering on C-ACC_Main_Function.	DS. 2, DS. 3	2
R.5	TS.5: Tampering on communication channels to perception data.	DS. 3	4
R.6	TS.6: Spoofing on communication channels to perception data.	DS. 3	4
R.7	TS.7: In-vehicle computers' identity spoofing.	DS. 2, DS. 3, DS. 5	3
R.9	TS.9: Denial of Service on C-ACC_Main_Function.	DS. 2	3
R.10	TS.10: Man-in-the-Middle Attack on VC.	DS. 3	3

In table 4.3, we can observe the given risk levels for the scope-relevant list of risks, and it indicates that the highest risk level concerning integrity is 4, representing the worst-case risk scenario for the assessed scope.

The methods explained in Deliverable D3.3 [Con25c] describe how to establish thresholds for belief, disbelief, and uncertainty. To determine the belief threshold (b_t), we evaluate the assigned risk levels and use the worst-case scenario to set a minimum value for b_t. The disbelief threshold (d_t) is based on the impact of each potential damage scenario and defines the maximum allowed value. Meanwhile, the uncertainty threshold (u_t) considers the capabilities for detecting a cybersecurity incident and the assigned cybersecurity assurance level. Following this approach, each threshold is defined using the proposed methodologies.

Belief threshold: To calculate the belief threshold (b_t), we utilize Equation 4.3. We assume that there is no defined belief threshold baseline b_{bt}.

$$\Delta = \frac{1 - b_{bt}}{5} = \frac{1 - 0.0}{5} = \mathbf{0.2} \tag{4.3}$$

From the list of risks in Table 4.3, we observed that the risk level assigned for the worst-case scenario ($R_{max}(F, I)$) is 4. By replacing Δ and the risk level in the equation 4.4, we get:

$$b_t = b_t + ((R_{max}(F, I) - 1) \times \Delta) = 0.0 + ((4 - 1) \times 0.2) = \mathbf{0.6} \tag{4.4}$$

Resulting in a b_t defined as **0.6**.

Disbelief threshold: According to the method in D3.3 ([Con25c]), to calculate the disbelief threshold (d_t), we observe the impacts for each damage scenario in Table 4.2 and assign weights for each impact category. For the C-ACC application, we assigned the following weights to each category: Safety = 0.5, Financial = 0.1, Operational = 0.4, and Privacy = 0.0. The weights and their corresponding mapping from the assigned impact rating to a numerical value are shown in Table 4.4 in parentheses. Equation 4.5 demonstrates the calculated weighted impact for each damage scenario.

Table 4.4: List of relevant Impact ratings, with weights

Damage scenario	S (0.5)	F (0.1)	O (0.4)	P (0.0)
DS.2	Moderate (0.5)	Negligible (0.0)	Moderate (0.5)	Negligible (0.0)
DS.3	Major (0.75)	Moderate (0.5)	Major (0.75)	Negligible (0.0)
DS.4	Moderate (0.5)	Moderate (0.5)	Major (0.75)	Negligible (0.0)

$$
\begin{aligned}
DS.2 : I_w &= \sum_{n=1}^{3} I_n W_n = 0.5 \times 0.5 + 0.0 \times 0.1 + 0.5 \times 0.4 + 0.0 \times 0.0 = 0.45 \\
DS.3 : I_w &= \sum_{n=1}^{3} I_n W_n = 0.75 \times 0.5 + 0.5 \times 0.1 + 0.75 \times 0.4 + 0.0 \times 0.0 = \mathbf{0.725} \\
DS.5 : I_w &= \sum_{n=1}^{3} I_n W_n = 0.5 \times 0.5 + 0.5 \times 0.1 + 0.45 \times 0.4 + 0.0 \times 0.0 = 0.48
\end{aligned}
\tag{4.5}
$$

As DS.5 has the highest weighted impact (I_w), we use I_w = 0.725 in Equation 4.6

$$d_t = 1 - I_w = 1 - 0.725 = \mathbf{0.275} \tag{4.6}$$

Resulting in a d_t defined as **0.275**.

Uncertainty threshold: The developed C-ACC function does not include any detectability capabilities. The C-ACC Main Function is part of a safety-critical system. Based on the analysed scope, defined damage scenarios, and assessed risk, we consider the assigned cybersecurity assurance level to be CAL 3. According to Deliverable D3.3 [Con25c] in Table 10.2, the combination of low detectability capabilities and a high required assurance level results in a low uncertainty acceptance level (UAL), which is numerically represented as 2.

To calculate the uncertainty threshold (u_t), we use Equation 4.7, where UAL is Low (2), from the previous analysis. Equation 4.7 demonstrates u_t calculation.

$$u_t = UAL_{map} = \frac{UAL}{5} = \frac{2}{5} = \mathbf{0.4} \tag{4.7}$$

Resulting in an u_t defined as **0.4**.

Final RTL's thresholds: From the calculated thresholds, the final RTL is defined by $b_t = 0.6$, $d_t = 0.275$, and $u_t = 0.4$. The graphical representation is shown in Figure 4.5. The blue area highlights the region defined by the RTL, where an ATL satisfies the requirements. This segment represents less than 15% of the triangle's total area, underscoring both the application's critical nature and the current lack of cybersecurity measures that mitigate the remaining risk.

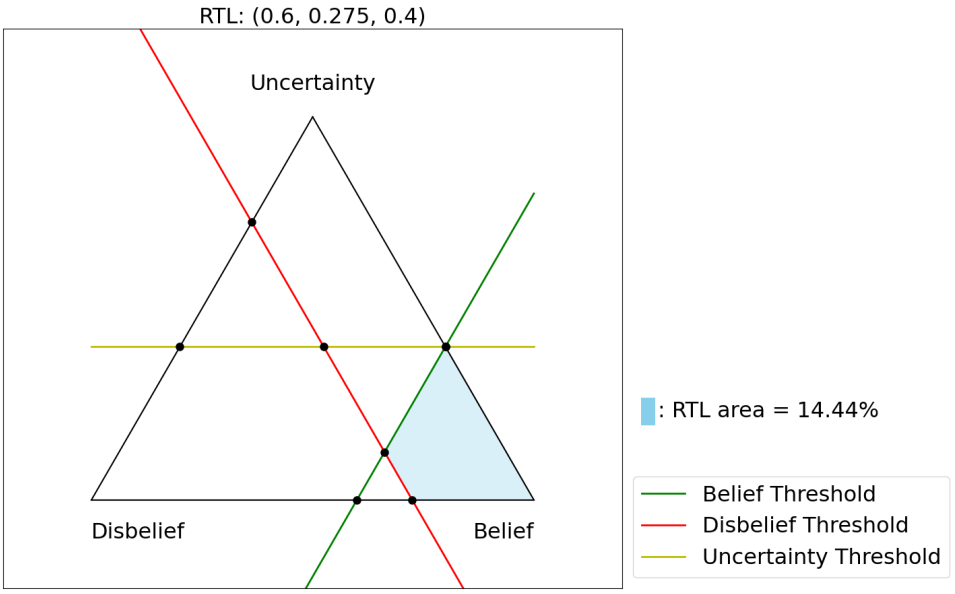


Figure 4.5: RTL representation for the C-ACC demonstrator within the subjective logic triangle.

In Section 4.3.1, the impact of the defined RTL is evaluated, considering all possible ATLs.

4.1.6 User Story Realisation

The following User Stories have been realised for this use case and evaluated in D6.1 and D6.2.

[CACC.US.1]: As C-ACC, I want to improve the safety and reliability of my function through dynamic trustworthiness assessments of how its input data items are generated (whether from local sensors or other vehicles) and how they are transferred to me. I want to be able to assess the trustworthiness to serve two different driving situations:

- Imminent driving situation: perform a quick assessment.
- Upcoming driving situation: perform a very detailed assessment with high confidence in its results to carefully prepare.

Interest: Analyse the behaviour of TAF and consider various factors, such as subscription-based or pull-based assessments and the use of cached data.

[CACC.US.2]: As C-ACC, I want to be able to compare the Actual with the Required Level of Trustworthiness for a specific node or data item as part of the overall service graph chain.
Interest: Enable the application to make cybersecurity decisions based on trust.

[CACC.US.3]: As the C-ACC Item, based on my policies, I want my C-ACC Main Component to be able to respond to changes in the trustworthiness of my execution platform (Vehicle Computer) by migrating to a more trustworthy execution platform to ensure the correctness and safety of my functionality.
Interest: Allow the application to make cybersecurity decisions based on trust, determine when to migrate to a trustworthy system, and select a new host if needed.

[CACC.US.4]: As the C-ACC Item, based on my policies, I want my C-ACC Main Component to be able to respond to changes in the trustworthiness of data items I receive from in-vehicle and vehicle-external sensors. I want to compensate for the change of trustworthiness by enacting pre-defined safe responses, such as adapting my function-internal calculations or increasing the distance to other vehicles.
Interest: Allow the application to make cybersecurity decisions based on trust, enabling it to respond to changes in the sensors' trustworthiness status, such as increasing the safe distance when the sensors are considered untrustworthy.

User story CACC.US.1 was performed in D6.1 for imminent driving situations, considering trust assessments to be received via TAF subscription regularly. In D6.2, we re-evaluate this user story considering on-demand trust assessments, where the application no longer waits for TAF to send an up-to-date ATL, but it requests it when needed.

User story CACC.US.2 appears in both evaluation phases, in D6.1 and D6.2. However, since it was benchmarked in the previous evaluation and there have been no changes to the workflow, it is not measured in this final version. Instead, it is summarised in the section on previous evaluations, as outlined in Section 4.1.8.1. This user story emphasises the need for alignment between ATL and RTL methodologies for better comparability.

As with the previous user story, CACC.US.3 is executed during both evaluation phases. However, the time taken for this process is not reassessed because the workflow remains unchanged, and no time variation was observed.

User story CACC.US.4 was not evaluated in the previous deliverable and so is only included in D6.2. CACC.US.4 does not correlate with any scenario, as it represents a stage that follows the process of trust decision-making.

Scenario 1 assesses the trustworthiness of in-vehicle computers used in immediate driving situations. This defines which in-vehicle computer a C-ACC instance can operate on, ensuring that it remains trustworthy for that function (CACC.US.1). From this user story realisation, we can observe the latency involved in requesting an assessment and receiving it from TAF.

To make a trust-based decision, we consider the realisation of CACC.US.2. Both the TAF and the application should be capable of making a decision based on a comparison of RTL and ATL. By following the CACC.US.2 realisation, we can measure both TAF and C-ACC time to effectively compare ATL and RTL, which is necessary for making an informed decision.

Once a decision to migrate is made, for CACC.US.3 realisation, it requires that the C-ACC selects a trustworthy platform to get activated on it, and the desactivation of the old instance on the old (untrustworthy) host. This realisation is used to measure the time taken, even though it is not relevant, as it happens in the background and does not contribute to downtime.

4.1.7 KPI & Acceptance Criteria

Scenario 1 is evaluated considering KPIs and criteria defined in Table 4.5. The scenario is evaluated across D6.1 and D6.2.

Table 4.5: Evaluated KPIs by user stories.

User story	KPI description	Acceptance criteria	Related benchmark & Results

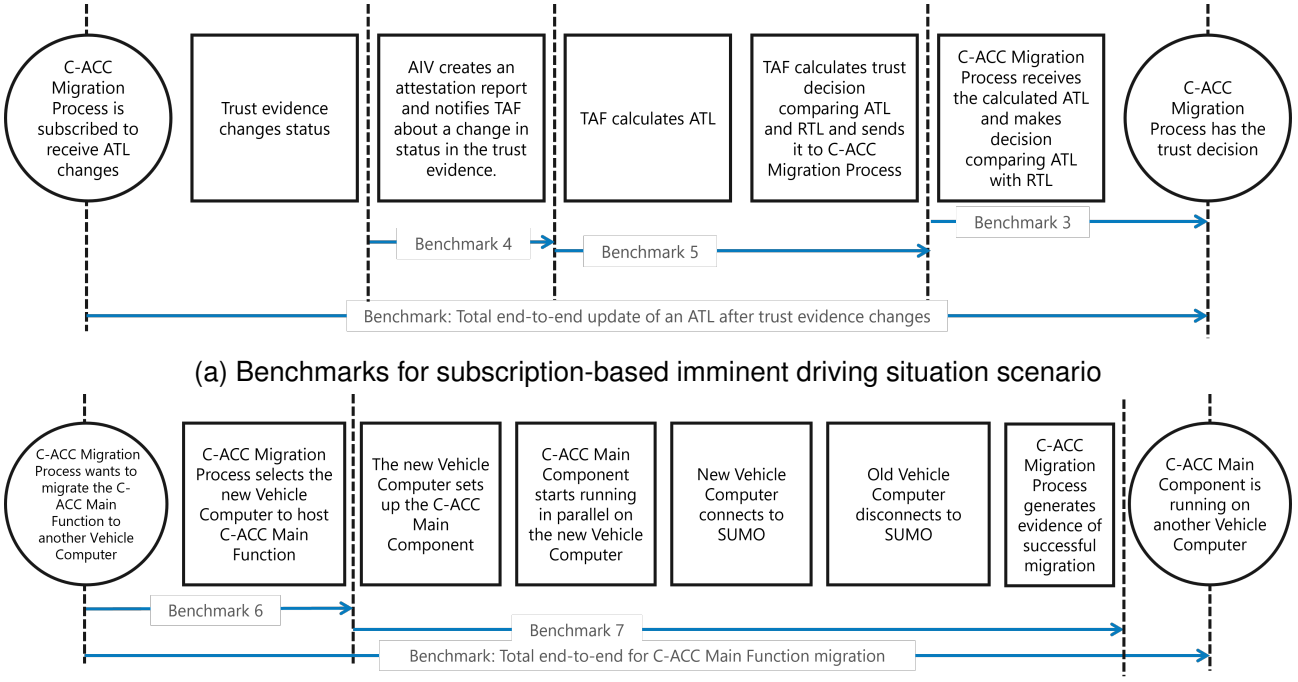
CACC.US.1	Latency for imminent driving situation	- Maximum of **100 ms** delay for the TAF to calculate an ATL and to make a trust decision (outside the CONNECT TEE). The TAF needs to be able to always send a response when a change in trustworthiness occurs, even when it is not using freshly-collected trust evidence. - Maximum of **200 ms** delay for the TAF to respond to C-ACC request when the TAF is instantiated and executed within the CONNECT TEE.	[D6.2] **1** (Section 4.1.8), [D6.1] **4**, **5** (Recap. Section 4.1.8.1). **Achieved**.
CACC.US.2	TAF's ability to compare ATL and RTL	TAF can compare ATL and RTL within a timeslot of **50 ms** (time needed for enabling a context switch between trusted and untrusted worlds).	**5** (Recap. Section 4.1.8.1). Achieved.
	C-ACC ability to compare ATL and RTL	C-ACC Main Function can calculate whether the ATL is below the RTL in less than **10ms**.	[D6.1] **3** (Recap. Section 4.1.8.1). **Achieved**.
CACC.US.3	**C-ACC Execution platform activation** (i.e., C-ACC can select another execution platform (in-vehicle computer) fulfilling the RTL as a host target, to activate a new instance of the C-ACC Main Function;)	For this selection, the target platform's ATL needs to fulfil C-ACC's RTL;	Not related to performance evaluation, but measured in D6.1 (Recap. Section 4.1.8.1). **Achieved**.
	Disabling Old Version C-ACC (i.e., C-ACC can disable the old C-ACC Main Function on the leaving platform and activate the new C-ACC Main Component on the target platform)	**TRUE**	Not related to performance evaluation, but measured in D6.1 (Recap. Section 4.1.8.1). **Achieved**.

4.1.8 Evaluation

Scenario 1 has been assessed in both D6.1 ([Con24b]) and D6.2 from different perspectives. D6.1 considered a subscription-based trust assessment and also measured the time taken to activate different C-ACC Main Function instances in different in-vehicle computers as a reaction to an in-vehicle computer trustworthiness status change. In this current implementation, D6.2 considers a pull-based trust assessment, which enables the application to receive trust characterisations on target in-vehicle computers upon request. Section 4.1.8.1 revisits D6.1 results, and 4.1.8.2 introduces the new evaluation.

4.1.8.1 Summary of Benchmarking Results for Subscription-based Trust Assessment

In D6.1 [Con24b], a subscription-based trustworthiness assessment for imminent driving situations was implemented and benchmarked, as well as the migration of the C-ACC Main Function control between in-vehicle computers. The benchmarks, illustrated in Figure 4.6, were executed, and the results can be found in Table 4.6. These benchmarks were not re-evaluated in the scope of this deliverable.



(a) Benchmarks for subscription-based imminent driving situation scenario

(b) Benchmarks for C-ACC Main Function activation on the new host

Figure 4.6: Benchmarks defined in D6.1.

- **Benchmark 3** - Measure the time taken for the C-ACC Activation Process to receive an ATL message from TAF, process it, and evaluate it by comparing it with the RTL to determine the current trustworthiness in the in-vehicle computer. Refer to Figure 4.6a.
- **Benchmark 4** - Measure the time for the AIV to construct an integrity report for each in-vehicle computer, including changes in trust evidence. Additional evaluation was reported in Chapter 7 in D6.1 [Con24b]. Refer to Figure 4.6a.
- **Benchmark 6** - Measure the time for the C-ACC Activation Process to select another execution platform based on trustworthiness level. Refer to Figure 4.6b.
- **Benchmark 7** - Measure the time for the C-ACC Activation Process to activate a new instance of the C-ACC Main Function on the new in-vehicle host. We evaluated the activation of the new C-ACC Main Function instance going from VC1 to VC2 (Benchmark 7a), from VC2 to VC1 (Benchmark 7b) and then combined all of the data regardless of direction (Benchmark 7c). Refer to Figure 4.6b.

Benchmark 5, concerning TAF measurements, is discussed in Deliverable D3.3 [Con25c] as part of the overall TAF evaluation. TAF was treated as a black box component, preventing us from conducting benchmarking within it.

We have run the test 200 times, considering 100 times the C-ACC activation from each in-vehicle computer to the other. Table 4.6, summarises the benchmark results in milliseconds (ms).

Table 4.6: Benchmark results for imminent driving situation. Time expressed in milliseconds.

Benchmark	Minimum	Maximum	Mean	Median	Std. Dev.
3	0.109	1.885	0.350	0.340	0.126
4	2.322	947.275	56.845	56.534	20.786
6*	0.000	0.021	0.001	0.001	0.001
7a	1863.494	1999.052	1834.626	1934.362	27.819
7b	1578.062	2002.937	1942.663	1947.196	45.991
7c	1578.062	2002.937	1938.664	1941.457	38.171

*The comparison time for RTL and ATL is too short, potentially allowing CPU scheduling to interfere.

In this scenario, CACC.US.1 requirement specifies a maximum 100 ms latency for the C-ACC Activation Process to receive updated in-vehicle ATL, assuming the TAF runs externally. This time shall be achieved by combining benchmarks 4 and 5. The average attestation report time from the AIV was 56.845 ms; to meet the full KPI, the TAF measurement (Benchmark 5) must perform its calculations with a mean latency below 43.155 ms. At that stage, TAF was not evaluated for this specific scenario, which is why it is not included in this summary.

4.1.8.2 Evaluation of pull-based Trust Assessment

As part of the second evaluation phase, we have measured the behaviour of the pull-based trust assessment modality when integrated in Scenario 1. The benchmark indicates the total time required by the C-ACC Migration Process to request and receive a new ATL from the TAF. The measurement is conducted through Benchmark 1, detailed below and in Figure 4.7.

Benchmark 1 - Measure the time the TAF takes to respond to the C-ACC Migration Process request for a new ATL, including the collection of evidence and ATL calculation.

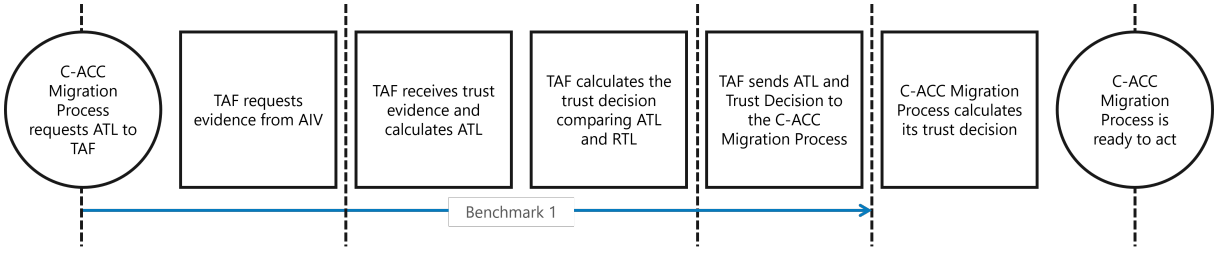
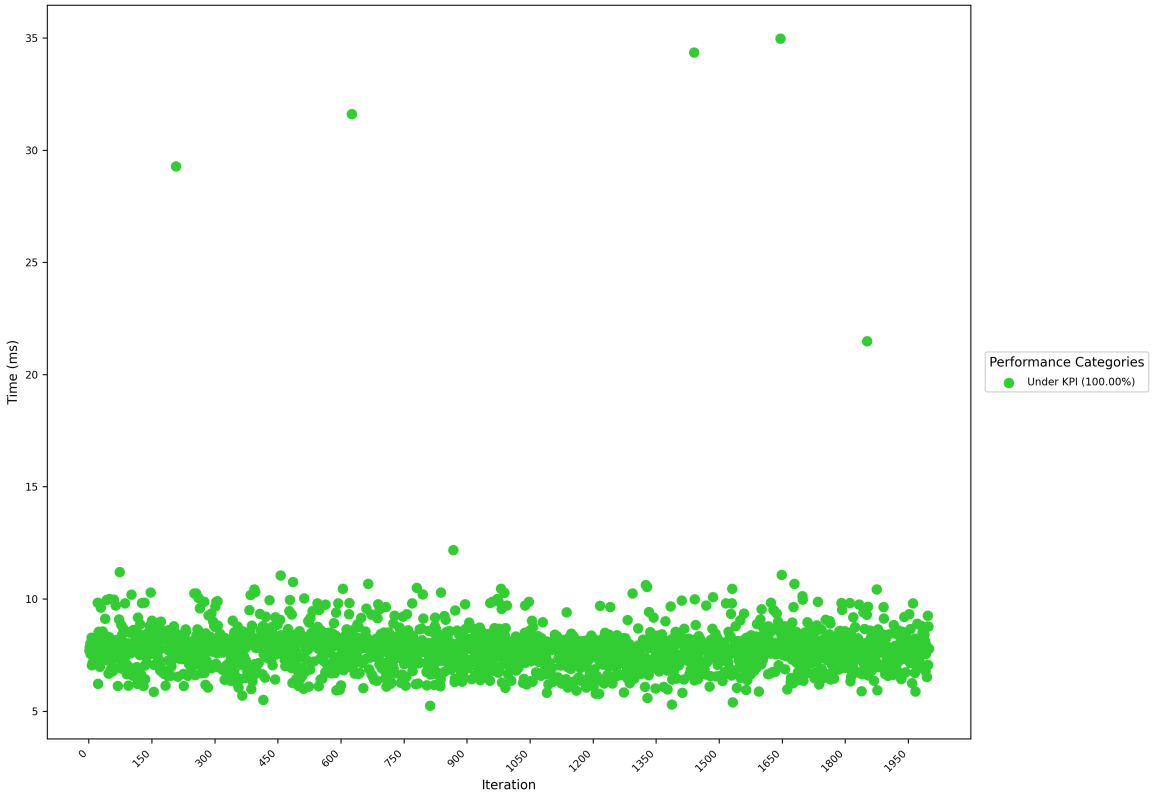


Figure 4.7: Benchmark for on-demand trust assessment.

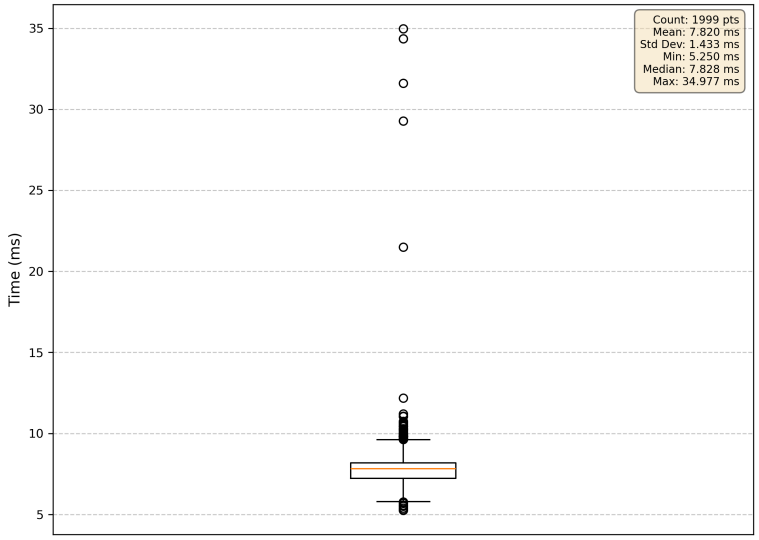
The benchmark tests the latency for imminent driving situations to ensure it meets the maximum required delay of 200 ms, as defined in D2.1 [Con23c] as a design decision, when TAF is instantiated in a trusted execution environment and 100 ms when executed as an untrusted application (i.e., not in isolation). For the given scenario, TAF is allowed to use cached data, reducing the need for evidence collection.

To assess the benchmark, the scenario is executed 2000 times with TAF running outside a TEE. For evaluating when TAF is executed within a TEE, we refer to experiments from D3.3 [Con25c]. The results are discussed in Section 4.1.9.

4.1.9 Discussion & Critique - Lessons Learnt



(a) Scatter plot of benchmark 1 when cached data is permitted without initial outlier. KPI = 100 ms



(b) Box plot for statistical distribution of the response time in imminent driving situations without initial outlier

Figure 4.8: Benchmark 1 for scenario 1- Performance analysis of the benchmark of on-demand trust assessment in imminent driving situations

The performance evaluation focuses on benchmarking pull-based trust assessments. To summarise, the C-ACC migration process requests an on-demand trust assessment from TAF and receives the current

ATL. For imminent driving situations (Scenario 1), the application allows TAF to use cached data whenever possible. However, for upcoming driving situations (Scenario 2), where there is more time to make decisions, the C-ACC application does not permit the use of cached data. In this case, TAF must request the collection of fresh data.

For scenario 1, the time between requesting and receiving a fresh ATL was measured. 2000 iterations were evaluated, and a request was sent to TAF every second. The evaluated performance is displayed in Figure 4.8.

As shown in Figure 4.8a, we can see a peak data point that falls outside the threshold during the first iteration. This occurs when TAF requests fresh evidence collection from AIV, as there is no cached data available. From the second iteration onward, all evaluated data remains below the threshold, as TAF keeps using cached data and has time to collect evidence in the background, based on periodic AIV updates or shortened AIV responses due to ongoing trust evaluation. Figure 4.8b illustrates the distribution and shows a minimal variance after excluding the outlier. On average, an ATL request took below 9 milliseconds. The results are summarised in Table 4.7.

In Deliverable D3.3 ([Con25c]), TAF was tested within a TEE, and the conclusion was that a TEE would add an average overhead of 6.59% when it is software-based, and 38.78% when it is hardware-based and this was highly dependent on the TAF implementation. Based on this evaluation, and as the same hardware and TAF were used, we assume the same overheads for the implemented use case. With this assumption, we can expect the results shown in Table 4.7.

Table 4.7: Summary of the benchmark results for imminent driving situations considering pull-based trust assessment (from sending request until receiving response with current ATL). Times expressed in milliseconds.

Benchmark	Iterations	KPI	Minimum	Maximum	Mean	Median	Std. Dev.
1	2000	100	5.250	1783.261	8.708	7.829	39.726
Software-based TEE (+6.59%)*	-	200	5.596	1900.77	9.282	-	-
Hardware-based TEE (+38.78%)*	-	200	7.286	2474.810	12.085	-	-

* The results are estimated based on previous experimentation with TAF executed within a TEE in D3.3.

Based on these results, we can conclude that the scenario is successfully meeting the KPIs. Aside from the initial request, which takes longer due to the time needed to establish SSH connectivity, pull-based trustworthiness assessment requests average 8.7 milliseconds when the TAF is outside a TEE. When the TAF is within a software-based TEE, the average response time is expected to be 9.3 ms, and within a hardware-based TEE, it is expected to be 12.1 ms.

4.2 Scenario #2: Upcoming Driving Situation

4.2.1 Description

In this scenario, we consider an upcoming driving situation in which the vehicle has enough time to request trustworthiness assessments based on current trust evidence from the TAF. As shown in Figure 4.9, rather than using the available evidence report, the C-ACC Migration Process requests an up-to-date assessment, forcing TAF to re-collect trust evidence.

This scenario is essential for evaluating how the TAF processes such requests and for measuring the time taken to generate an up-to-date ATL. It contains all the steps involved, including gathering evidence, calculating the ATL, and sending the results to the application.

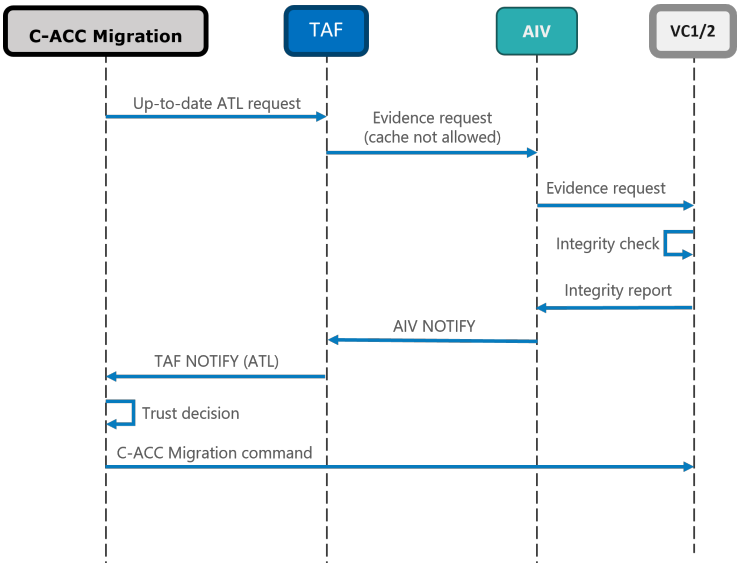


Figure 4.9: Workflow for on-demand upcoming driving situation.

4.2.2 Setup - Topology - Testbed details

The architecture for this scenario is outlined in Section 4.1.2. A key characteristic of this scenario is not to be allowed to use cached data during the trustworthiness assessment. The AIV will consistently be utilised to generate a fresh evidence report. As a consequence, the TAF response time will be longer when compared to Scenario 1.

4.2.3 Trust Model

The trust model outlined in Section 4.1.3 applies to this scenario without any caveat. Both the RTL and the trust evidence remain unchanged as well.

4.2.4 User Story Realisation

The user story realisation was detailed in Section 4.1.6. Scenario 2 assesses the trustworthiness of in-vehicle computers used in upcoming driving situations. This defines which in-vehicle computer a C-ACC instance can operate on, ensuring that it remains trustworthy for that function (CACC.US.1). From this user story realisation, we can observe the latency involved in requesting an assessment and receiving it from TAF. ATL values used in this scenario are based solely on fresh evidence; cached data is not permitted.

To make a trust-based decision, we consider the realisation of CACC.US.2. Both the TAF and the application should be capable of making a decision based on a comparison of the RTL and the ATL. By following the CACC.US.2 realisation, we can measure both TAF and C-ACC time to effectively compare the ATL and RTL, which is necessary for making an informed decision.

Once a decision is made, for CACC.US.3 realisation, it requires that C-ACC selects a trustworthy platform, gets activated on it and then deactivates the old instance on the previous (untrustworthy) host. This realisation is used to measure the time taken, even though it is not significant, as it occurs in the background

and does not contribute to downtime. The new host only takes control once it is ready, running in parallel with the old instance for a period of time to ensure a smooth transition.

4.2.5 KPI & Acceptance Criteria

Table 4.8 presents the KPIs set to evaluate the use case when assessing the in-vehicle trustworthiness by a pull-based trust assessment. The KPI is used to evaluate the assessment of the in-vehicle computer's trustworthiness when cached data is not allowed, and TAF needs to collect fresh evidence.

User story	KPI description	Acceptance criteria	Related benchmark & Result
CACC.US.1	Latency for upcoming driving situation	TAF responds to C-ACC request with the assessment based on freshly collected evidence with latency $<$ **2 sec**. The time concerns the collection of fresh trust evidence, ATL calculation, and trust decision-making by the TAF.	1 (see Section 4.2.6). **Achieved**.

Table 4.8: Evaluated KPI for on-demand trust assessments for Scenario 2.

4.2.6 Evaluation

For pull-based trust assessment in Scenario 2, the benchmark indicates the total time required by the C-ACC Migration Process to request and receive a new ATL from the TAF when it may not use cached data, but instead relies on freshly collected trust evidence. The measurement is conducted through Benchmark 1, similarly to Scenario 1, and it is detailed below and in Figure 4.10.

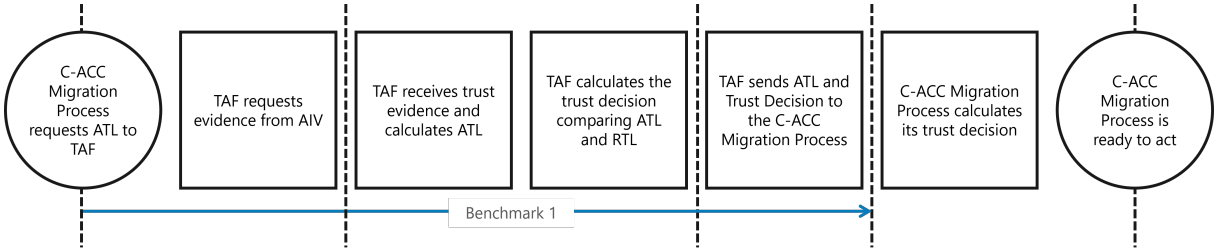


Figure 4.10: Benchmark for on-demand trust assessment for upcoming driving situations.

Benchmark 1 - Measure the time the TAF takes to respond to the C-ACC Migration Process request for a new ATL, including the collection of evidence and ATL calculation.

The benchmark tests the latency for upcoming driving situations to ensure it meets the maximum allowed delay of 2 seconds to receive a trust assessment from TAF. The KPI does not distinguish between a TAF being executed outside or within a CONNECT TEE.

To assess the benchmark, the scenario is executed 200 times with TAF running outside a TEE. For evaluating when TAF is executed within a TEE, we refer to experiments from Deliverables D3.3 ([Con25c]). The results are discussed in Section 4.2.7.

4.2.7 Discussion & Critique - Lessons Learnt

In this deliverable, we focus on assessing the performance of pull-based trust evaluations. To summarise Scenario 2, the C-ACC Migration Application requests a new ATL and does not allow TAF to utilise any cached data. Consequently, for each iteration, TAF sends AIV requests, which require AIV to perform trustworthiness checks on each in-vehicle computer, as specified in the trust model.

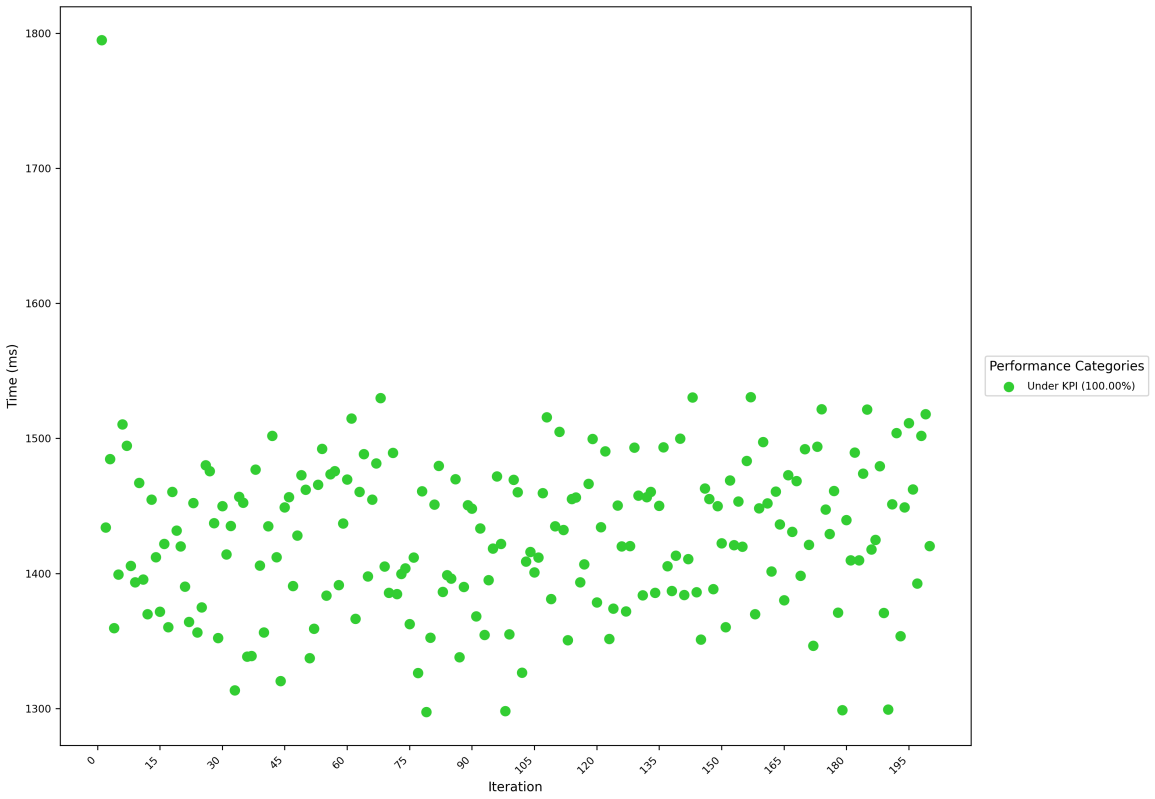
Table 4.9: Summary of the benchmark results for imminent driving situations considering pull-based trust assessment (from sending request until receiving response with current ATL). Times expressed in milliseconds.

Benchmark	Iterations	KPI	Minimum	Maximum	Mean	Median	Std. Dev.
1	200	2000	1294.489	1794.855	1427.422	1428.735	58.897
Software-based TEE (+6.59%)*	-	2000	1379.796	1913.136	1521.490	-	-
Hardware-based TEE (+38.78%)*	-	2000	1796.492	2490.900	1980.977	-	-

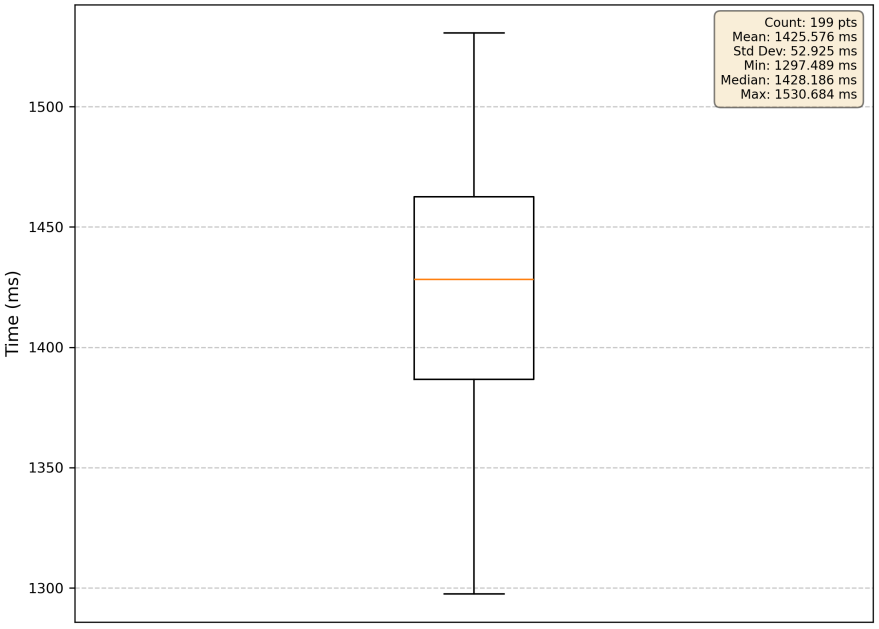
* The results are considered based on previous experimentation with TAF executed within a TEE.

In this scenario, we evaluated a total of 200 requests, measuring the time taken from the initial request to the receipt of a fresh ATL. Figure 4.11 presents a summary of the benchmark results for Scenario 2. Table 4.9 summarises the results of the benchmarks performed in this evaluation phase for Scenario 2.

Based on Benchmark 1, we can conclude that Scenario 2 is successfully meeting the KPIs. Aside from the initial request, which takes longer due to the time needed to establish SSH connectivity, pull-based trustworthiness assessment requests average 1427.422 milliseconds when the TAF is outside a TEE. When the TAF is within a software-based TEE, the average response time is estimated to be approximately 1522 ms, and within a hardware-based TEE, it is estimated to be approximately 22000 ms. In all tests, TAF took less than 2 seconds to perform the trustworthiness assessment, even with the enforcement of fresh evidence collection for every request.



(a) Scatter plot of benchmark 1 when cached data is not permitted. KPI = 2000 ms.



(b) Box plot for statistical distribution of the response time in upcoming driving situations without initial outlier.

Figure 4.11: Benchmark 1 for Scenario 2 - Performance analysis of the benchmark of on-demand trust assessment in upcoming driving situations

4.2.7.1 General Observations from both scenarios

Observing both scenarios 1 and 2, it is clear that the use of cached data has a significant impact on performance. When possible, utilising cached data is more efficient, but it can create a false sense of trustworthiness. This occurs when important changes in trustworthiness levels are overlooked because they have not yet been updated in the cached data.

Based on all the benchmarks conducted, TAF operates effectively for both subscription and pull-based assessments needed for this use case. One crucial factor to consider is the duration of each trust check. If a comprehensive trust check is conducted in the evaluated system, it will directly impact the AIV response time. This effect is cumulative and increases with the number of checks required. Before establishing a performance threshold, it is essential to assess the trustworthiness checks, as they can become a bottleneck in the overall assessment of trustworthiness.

The TAF always subscribes to AIV, even when performing pull-based trust assessments, and it does not differentiate between updates received through AIV subscriptions and those requested on demand. When benchmarking, significant discrepancies can happen if pull-based requests overlap with the TAF receiving updated evidence collections from subscriptions. Although faster assessments may not be a practical problem, and could even be seen as an advantage, since quicker evaluations are generally preferable, this overlap can create challenges for benchmarking pull-based requests, potentially leading to a misleading impression of speed.

4.3 Additional Experiments

In addition to scenarios 1 and 2, we analysed how each trust source interacts with the others and their impact on the final calculated ATL for this use case, in Section 4.3.1. Additionally, the execution of user story CACC.US.4 was benchmarked in Section 4.3.2, as it occurs after a trustworthiness assessment and is not connected to any specific scenario.

4.3.1 Trustworthiness classification

Using the RTL defined in Section 4.1.5 as a threshold, we generated all possible ATLs from all possible trust source combinations, resulting in 4096 cases. The results, depicted in Figure 4.12, reveal a perceptible pattern among ATLs, with identical values sometimes originating from different combinations. The dots on the triangle indicate the ATL locations, their size directly correlating with the frequency of their corresponding combinations. Larger dots represent higher frequencies.

The analysis of the ATLs reveals that the system is predominantly classified as ”Untrustworthy” (3956 combinations, or approximately 96.58% of cases), indicating strict conditions for achieving a trustworthy status, based on the defined cybersecurity-based RTL.

A ”Trustworthy” classification (140 combinations, or approximately 3.42% of cases) is achieved even when up to 41.67% of individual claims are ”imperfect” (failed, not engaged, or evidence not received), highlighting a tolerance for minor faults.

”Trustworthy” status consistently presented a high belief score (ranging from 0.62 to 0.76) with controlled low disbelief (0.10 to 0.24) and uncertainty (0.14 to 0.28). The highest trustworthiness level is specifically achieved when belief is 0.76, disbelief is 0.10, and uncertainty is 0.14.

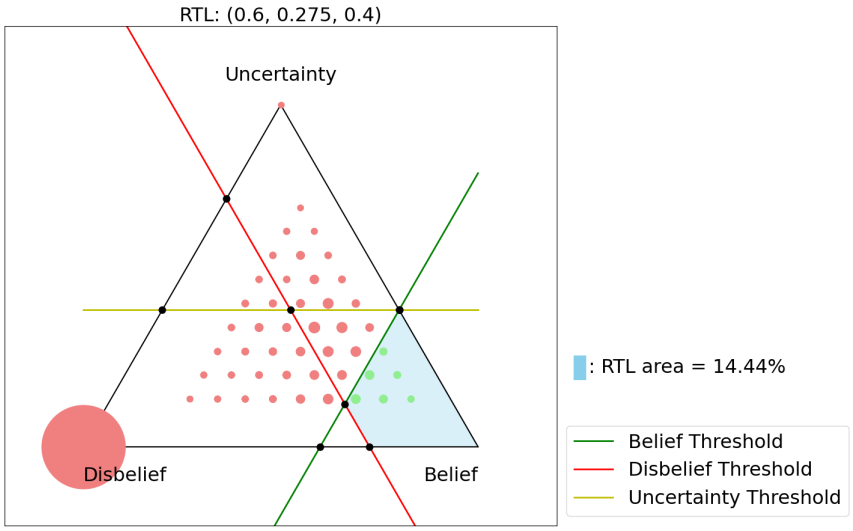


Figure 4.12: Trustworthiness classification of ATLs generated from all possible trust source combinations for the use case

Figure 4.13 illustrates what a trustworthy status looks like and how flexible the system is in allowing certain claims to fail while still being classified as trustworthy overall. We can observe that trustworthiness is consistently associated with the secure boot claim being successfully verified, demonstrating its foundational importance, while allowing for some flexibility in access control and secure OTA claims.

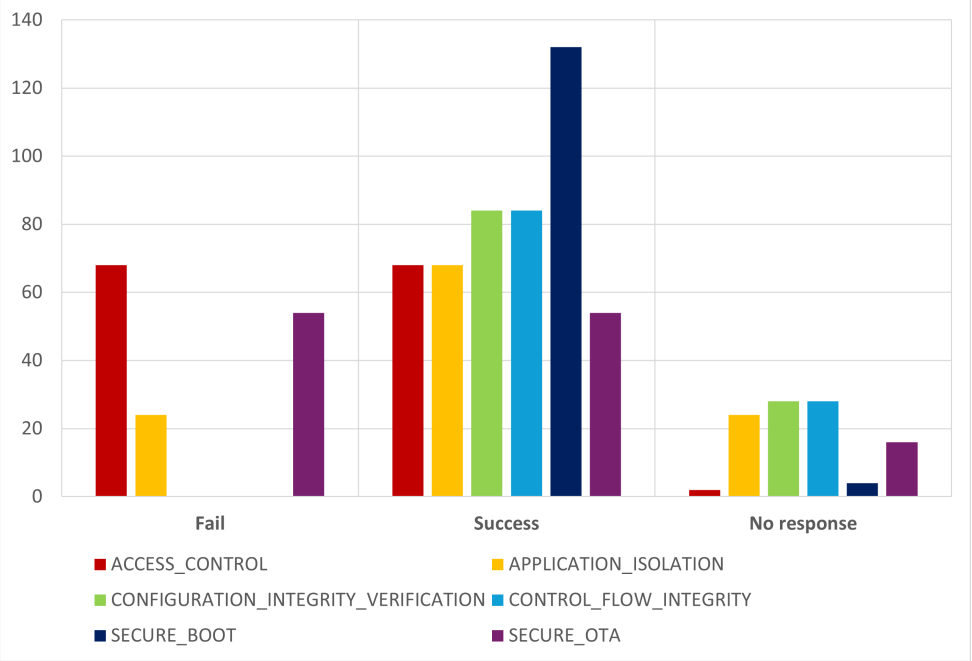


Figure 4.13: Trustworthy status characterisation through trust evidence status

4.3.2 Benchmark 2 - Reaction to Changes in sensor's trustworthiness

The second benchmark aimed to assess the reaction time of the C-ACC main application when it detects a change in the sensor's trustworthiness by comparing ATL to RTL. For that, we considered that it happens after trustworthiness assessment and decision-making. Figure 4.14 represents the system architecture

considered for this benchmark. Note that this was an assumption, as trustworthiness assessment for sensors was not implemented in TAF and, consequently, could not be performed.

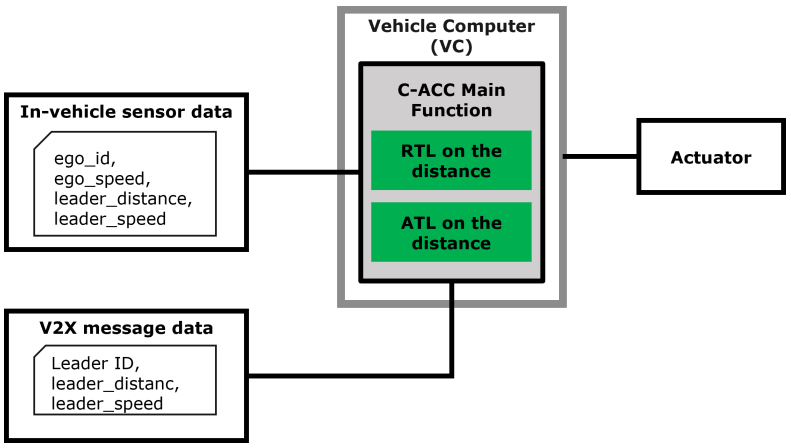


Figure 4.14: Architecture assumed for Benchmark 2.

For benchmark purposes, we considered a fixed RTL and different ATLs to trigger the decision-making. RTL and ATLs are specific to this scenario. Once triggered, the C-ACC Main Function increased or decreased the gap to the vehicle ahead based on the trustworthiness status of the location sensors. Benchmark 2 is summarised in Figure 4.15. This benchmark measures how long this reaction takes to increase the gap between the ego and the leading vehicle. For this Benchmark, 50 ms has been defined as the KPI.

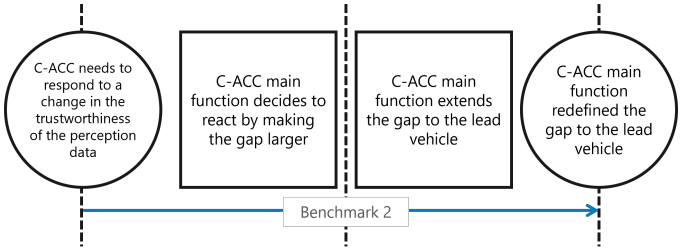


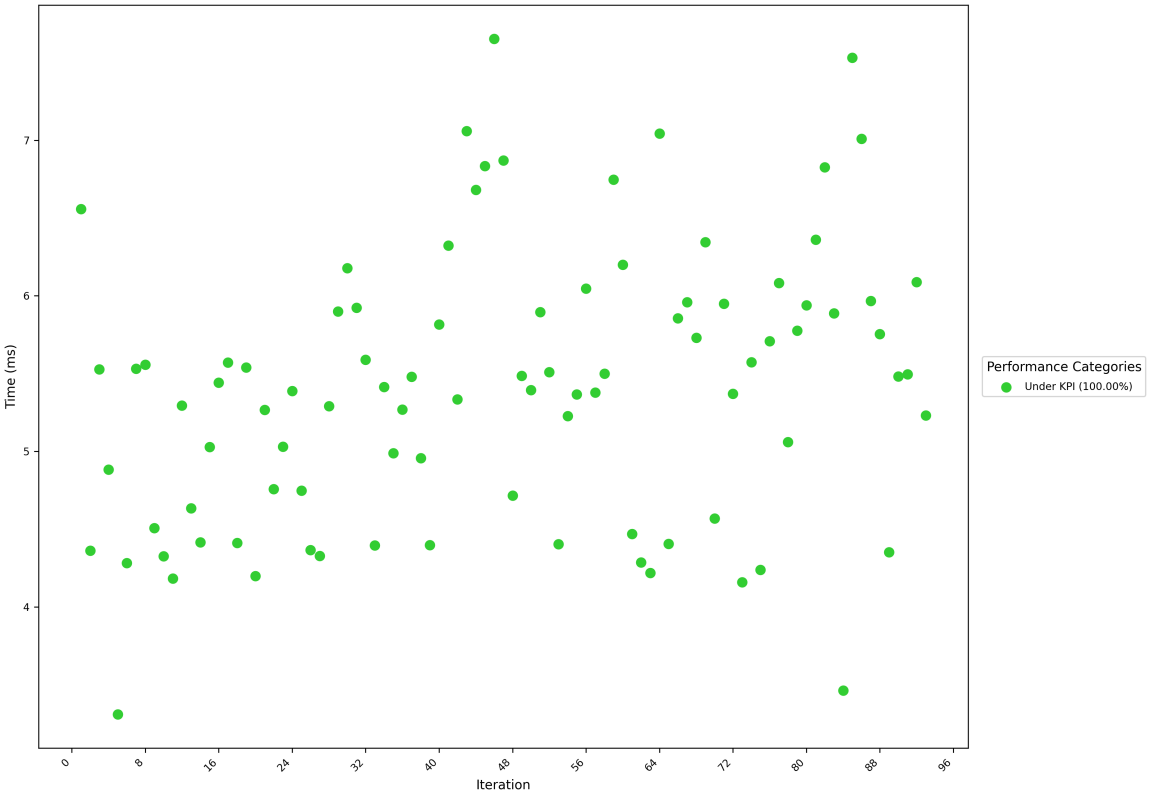
Figure 4.15: Benchmark for C-ACC when responding to changes in the trustworthiness of perception data.

Benchmark 2 - Measure the reaction time for the C-ACC to increase the gap to the lead vehicle in response to a trustworthiness issue in its perception data. Refer to Figure 4.15.

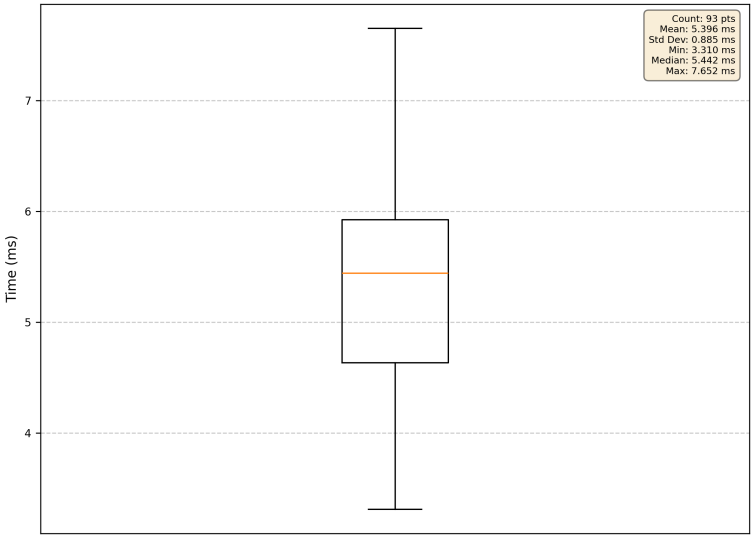
This change in trustworthiness was analysed 94 times, and the results are summarised in Table 4.10 and Figure 4.16.

Table 4.10: Summary of the benchmark results for C-ACC Main Function reaction when trustworthiness level changes. Times expressed in milliseconds.

Benchmark	Iterations	KPI	Minimum	Maximum	Mean	Median	Std. Dev.
2	94	50	3.310	310.287	8.640	5.460	31.459
2 w/o outlier	93	50	3.310	7.652	5.396	5.442	0.885



(a) Scatter plot for the Benchmark 2 without initial outlier. KPI = 50 ms.



(b) Box plot for statistical distribution of the Benchmark 2 without outlier.

Figure 4.16: Benchmark 2 - Performance analysis of the time taken by the application to change behaviour when the sensor's trustworthiness status changes.

In the first iteration, there was a noticeable peak; however, over 98% of the reaction times fell below the 50-ms threshold, with an average reaction time of under 5.5 ms. Based on this experiment, we can conclude that the use case effectively meets the expected KPI.

4.3.2.1 Benchmarks for verifying the actual trustworthiness of each in-vehicle computer.

AIV uses Linux tools to generate integrity checks, considering each trustworthiness claim, which includes Access Control, Secure Over-The-Air (OTA) Updates, Secure Boot, Application Isolation, Control Flow Integrity (CFI), and Configuration and Integrity Verification (CIV). Table 4.11 provides a summary of the security checks that were conducted, along with the average time taken for each check, in milliseconds. For measuring the time, each check is run 100 times.

Table 4.11: Security checks and criteria that are managed by AIV and performed on each in-vehicle computer.

Check	Success Criteria	Failure Examples	Timing (ms)
Access Control	AppArmor logs are verified by the in-vehicle computer, confirming that the mandatory access control policies are correctly enforced and no unauthorised access attempts are recorded.	AppArmor logs show violations of defined policies, or the logs are incomplete/missing, indicating a potential bypass or misconfiguration of access controls.	139 ms
Application Isolation	In-vehicle computer confirms that UFW is active and correctly configured to enforce application isolation rules, and a review of UFW logs reveals no error messages or unauthorised connection attempts.	In-vehicle computer finds UFW is inactive, misconfigured, or its logs contain error messages or evidence of unauthorised network connections bypassing firewall rules.	184 ms
Control Flow Integrity	In-vehicle computer verifies that the Linux kernel is compiled with CFI enabled and that all critical applications are running exclusively from their correct and expected binary paths, preventing unauthorised code execution.	In-vehicle computer discovers the Linux kernel is compiled without CFI, or an application is found executing from an unexpected or unauthorised binary path, indicating a potential CFI bypass or code injection.	140 ms
Secure Boot	In-vehicle computer verifies that it has successfully booted with Secure Boot enabled, that no unsigned kernel modules are loaded, and that `mokutil` confirms proper UEFI and MOK management.	In-vehicle computer detects that Secure Boot is disabled, unsigned kernel modules are loaded, or `mokutil` reports errors indicating compromised UEFI or MOK integrity.	97 ms
Secure OTA	In-vehicle computer confirms that all OTA update mechanisms are securely configured (e.g., cryptographic signatures, secure communication channels) and that the system has successfully applied the latest available security updates.	In-vehicle computer identifies insecure OTA update configurations (e.g., unsigned updates accepted, unencrypted communication), or the system is found to be running outdated software versions with known vulnerabilities.	813.5 ms

Chapter 5

Demonstrator #3: Slow Moving Traffic Detection (SMTD)

5.1 Description

The Car2Car Communication Consortium, in its “Guidance for Day 2 and Beyond Roadmap” [CAR19], identifies Collective Perception Messages (CPMs) as a central component of the V2X Day 2 phase, referred to as *Sensing Driving*. However, the same document emphasizes that consistency and plausibility checks on safety-related messages must ultimately be integrated into a unified security framework to ensure message integrity and authentication of data and origin. One of the main vulnerabilities in the CPM chain lies in the plausibility of the data: since CPMs describe the kinematics of road users, any implausible kinematic information undermines the usefulness of the data. For this reason, the CONNECT project selected, designed, implemented, and evaluated a CPM-based use case titled as Slow-Moving Traffic Detection (SMTD).

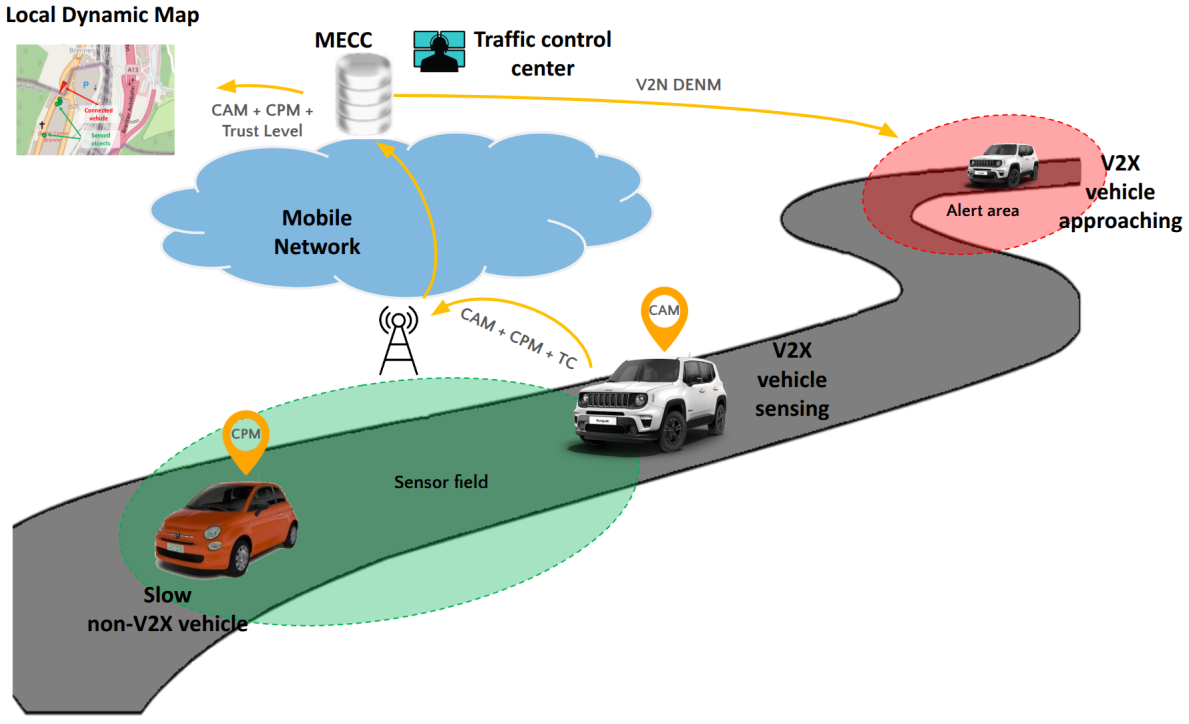


Figure 5.1: SMTD Use Case.

The aim of the Slow-Moving Traffic Detection (SMTD) use case is to signal traffic congestion on the road leveraging Vehicle-to-Everything (V2X) communication technologies. Specifically, a V2X-equipped vehicle (we will refer to it as ego-vehicle) can use its on-board sensors to perceive the presence of slow-moving traffic in front and it can signal it to a central control centre managing the traffic of the whole area.

The SMTD use case is designed to demonstrate how to mitigate traffic congestion and improve road safety by utilizing Vehicle-to-Everything (V2X) communication technologies—specifically Vehicle-to-Network (V2N) communication is used in the SMTD use case. Indeed, in real-world deployments, particularly in urban environments, CPMs should also be transmitted in real time to traffic management centres and associated cloud or edge infrastructures. This use case shows how V2X equipped vehicles can reliably share real-time information about slow-moving traffic participants. This data sharing can contribute to reduced congestion, enhanced traffic flow efficiency, and lower environmental footprint.

At the core of the SMTD system are the Collective Perception Messages (CPMs) and Cooperative Awareness Messages (CAMs), which convey real-time environmental and kinematic information. These messages are securely signed using short-term anonymous credentials, ensuring both data integrity and source authenticity. Figure 5.1 illustrates the overall functionality of the SMTD use case. In this scenario, a slow-moving vehicle that is not equipped with V2X capabilities—represented by the orange car in the figure—is detected by a following V2X-equipped ego-vehicle using its front-facing camera sensor. The sensing vehicle (from now on, the terms sending vehicle and ego-vehicle are going to be used interchangeably) is then able to generate CAMs, which report its own position over time, as well as CPMs, which convey the presence and position of the detected vehicle ahead.

The V2X messages, along with Trustworthiness Claims (TCs) (which are essential for creating a trust opinion of the ego-vehicle), are transmitted via an LTE cellular connection to a Multi-Access Edge Computing (MEC) server. However, in our experimental setup, the server used as the MEC is a standard cloud-based server located behind the Internet cloud and not directly connected to the base station, as would be expected in a typical MEC deployment. To reflect this architectural distinction, we refer to it as Multi-Access Edge Cloud Computing (MECC), highlighting its placement in the cloud rather than at the network edge. The purpose of this use case is to allow the V2X messages to be received by a server operated by a Traffic Control Center (TCC), which monitors message flows and evaluates the reliability and trustworthiness of the perceived road objects.

Ultimately, the TCC operating the MECC server is responsible for analysing message flows to detect potential misbehaviours and for computing a trustworthiness level for the perceptions conveyed in the CPMs thanks to the Trustworthiness Assessment Framework (TAF) and to the Trustworthiness Claims Handler (TCH). The positions of the ego-vehicle—derived from CAMs—and the detected slow-moving vehicle—extracted from CPMs—along with the associated trustworthiness level, are visualized on a web interface known in ETSI standards as the Local Dynamic Map (LDM) [ETS14a]. This information is also made available on a display mounted on the ego-vehicle system. If the perception of the slow-moving vehicle is deemed trustworthy, a specific area will be highlighted on the LDM to indicate the presence of slow-moving traffic to approaching V2X-equipped vehicles, thereby enabling early awareness of the potentially hazardous situation. In this chapter, we will present our setup, define the KPIs for the test sessions, explain the architecture and the communication between its components, and provide a comprehensive overview of both simulated and real-world tests.

Notably, the SMTD use case serves as a **living lab** demonstration scenario, leveraging vehicles equipped with Advanced Driver Assistance Systems (ADAS) and V2X hardware to transmit real V2X messages, made possible through the OScar (Open Stack for Car) framework [RRC24]. In the initial phase of the CONNECT project, a test was conducted on the Stellantis test track involving two vehicles: a Fiat 500 without V2X capabilities and a Jeep Renegade equipped with a frontal camera sensor and a dedicated On-Board Unit (OBU) for V2X communication (acting as ego-vehicle). During the tests, the ego-vehicle successfully detected the presence of vehicle ahead and generated CAMs to report its own position, as well as CPMs to signal the presence and position of the detected vehicle. These V2X messages were transmitted via an LTE cellular connection to a dedicated MECC server using the Advanced Message

Queuing Protocol (AMQP) version 1.0, as specified by the C-Roads standards [C-R23].

Table 5.1 summarizes the updates to the evaluation activities conducted as part of the final framework release. The left column lists all the tasks completed during the first half of the project, which were evaluated in Deliverable D6.1. The right column details the corresponding progress made since Release A, representing the tasks and advances discussed in this document.

***CONNECT* SMTD Evaluations in Release A**	**SMTD Evaluations in Final Release of *CONNECT***
Vehicle architectural setup and living lab environment definition. Identification and integration of all hardware components required for the Stellantis ego-vehicle. Installation of dedicated power supply connectors onboard to support the V2X hardware. Development of the necessary software modules for all onboard components. Finalization and deployment of the complete vehicle architecture within the defined living lab environment.	**Vehicle-to-MECC latency measurements.** Measurement and verification of the propagation delay between V2X messages generated by the vehicle and their reception by the MECC, exploiting the cellular network.
	Task Offloading hardware setup and field testing. Extension of the in-vehicle hardware to support the Task Offloading hardware and execution of field tests to evaluate system performance against defined KPIs.
	Local Dynamic Map development. Backend development enhancements on the MECC for aggregating incoming V2X data and Actual Trust Level (ATL) outputs from the TAF. Refactor and extend the a frontend web-based interface for real-time visualization of vehicle and object position, incorporating the TAF reports (e.g., ATL scores) on a tablet mounted onboard.
	Data monitoring setup and field testing. Implementation of mechanisms to log motion and perception data from the vehicle. Support for both periodic and event-triggered transmission of logged CAMs and CPMs via the cellular connection to the MECC.
Evaluation of the CONNECT AS-IS scenario and collection of traces from field tests. Deployment of the defined living lab environment on the Stellantis test track. Execution of tests validating the transmission of CAMs and CPMs at appropriate frequencies. Transmission of V2X messages to the MECC over a cellular network.	**Misbehaviour Detector and Trustworthiness Assessment Framework deployment.** Integration of the Misbehaviour Detector (MBD) and formatting of received V2X messages and Misbehaviour Reports (MRs) for TAF processing. Deployment of the Trustworthiness Assessment Framework (TAF), initialization of message exchanges, and extraction of Actual Trust Levels (ATLs) from TAF reports as output.
	End-to-end latency measurements. Insertion of generation timestamps into the incoming TCs before being processed by the TAF. Correlation with the timestamps in the TAF reports to compute end-to-end latency across the full processing chain.
	Trust Level evolution validation. Monitoring and analysis of ATL dynamics during both simulated and real-world scenarios to verify correct behaviour in diverse scenarios.
	Trustworthiness Claim Handler deployment and performance analysis. Installation of the TCH on the vehicle's On-Board Unit (OBU). Assessment of the impact of transmitted Trustworthiness Claims (TCs) on the ATL calculations pertaining to the trust propositions of both the ego-vehicle and the perceived objects. Evaluation of end-to-end latency under varying TC transmission frequencies.

Table 5.1: CONNECT Framework SMTD Evaluations between the different Releases

5.2 Setup - Topology - Testbed details

The complete vehicle setup—including the architectural design, the connection between the vehicle's CAN bus and the V2X On-Board Unit (OBU), and the mechanisms for generating, transmitting, and receiving V2X messages—is thoroughly described in Deliverable D6.1 [Con24b].

5.2.1 MECC architecture

In the second half of the CONNECT project, the focus of the SMTD use case shifts toward the MECC architecture and the services deployed within it. This represents the core contribution of the present deliverable, where the vehicle setup and V2X message communication are considered as given. Figure 5.2 shows a block diagram illustrating the structure of the MECC system. The V2X-equipped vehicle, operating in a living lab scenario (test track), transmits its messages via LTE to an AMQP Broker server. Similarly, streams of V2X messages from simulated vehicles can also be directed to the AMQP Broker. These simulated message flows are used to intentionally introduce misbehaviours into the system, which are essential for conducting a comparative analysis with the living lab test scenario.

The AMQP Broker handles the flow of CAMs and CPMs originating from both real and simulated vehicles. These messages are published in real time to all AMQP clients subscribed to the relevant topics. Notably, one of these AMQP clients runs directly on the MECC server, where it processes the incoming messages and forwards them to other relevant components or services.

The first service responsible for handling V2X messages is the Misbehaviour Detector (MBD). The MBD deployed in this context is the same as the one developed for the Intersection Movement Assistance (IMA) use case—see Chapter 3. This service processes every message received by the AMQP client and performs checks to detect potential misbehaviours in the data.

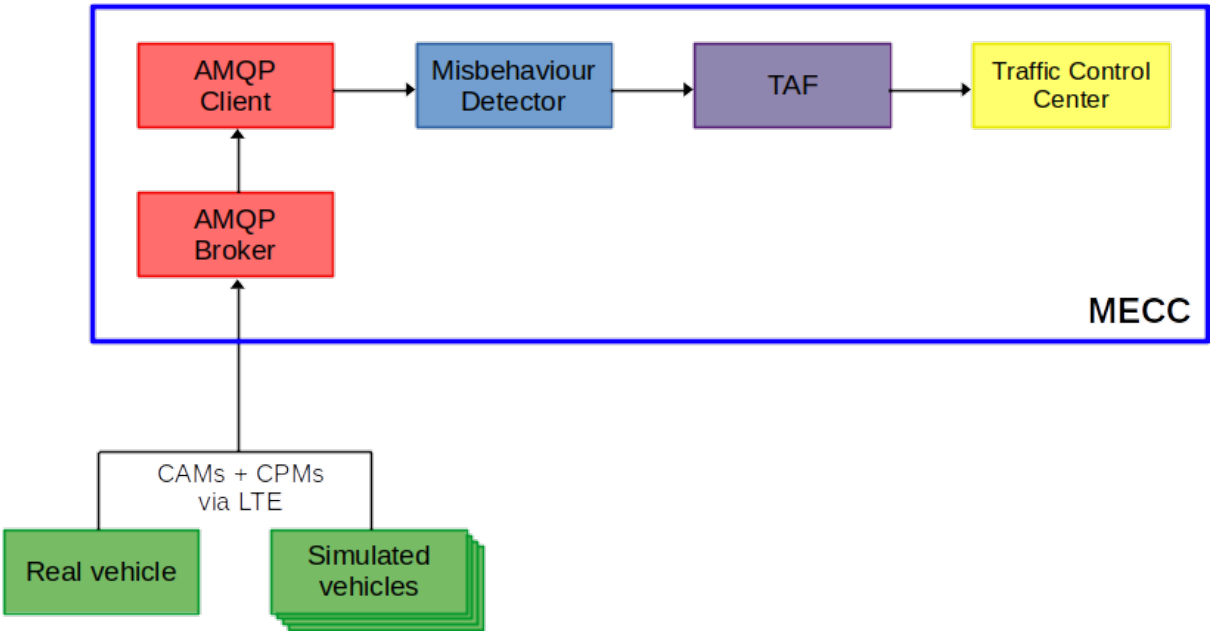


Figure 5.2: MECC architecture of the SMTD use case.

The output of the MBD service is subsequently forwarded to the Trustworthiness Assessment Framework (TAF) service. The primary goal of the TAF is to characterize the trustworthiness of data—through plausibility checks—transmitted by a V2X entity, whether referring to its own state or to the entities it perceives. This evaluation is expressed as a dynamic trust percentage that evolves over time, reflecting the system's confidence in the reliability of the perceived objects.

Finally, the positions of the ego-vehicle and the perceived vehicle, along with the associated trust percentage of the perception, are visualized through a Local Dynamic Map (LDM) service, which emulates the functionality of a Traffic Control Center (TCC).

The following subsections provide a detailed overview of the implementation process for the main services deployed on the MECC.

5.2.2 Simulated vehicle instances

In the SMTD use case, a vehicle is used for test track demonstrations, as detailed in Section 5.6.1. However, to test and evaluate the proper functioning of the Misbehavior Detector (MBD) and the Trust Assessment Framework (TAF), it is necessary to create ad-hoc test cases that include deliberately introduced misbehaviours. For this reason, the SMTD use case also leverages simulated vehicle instances to cover instances that cannot be tested in the test track.

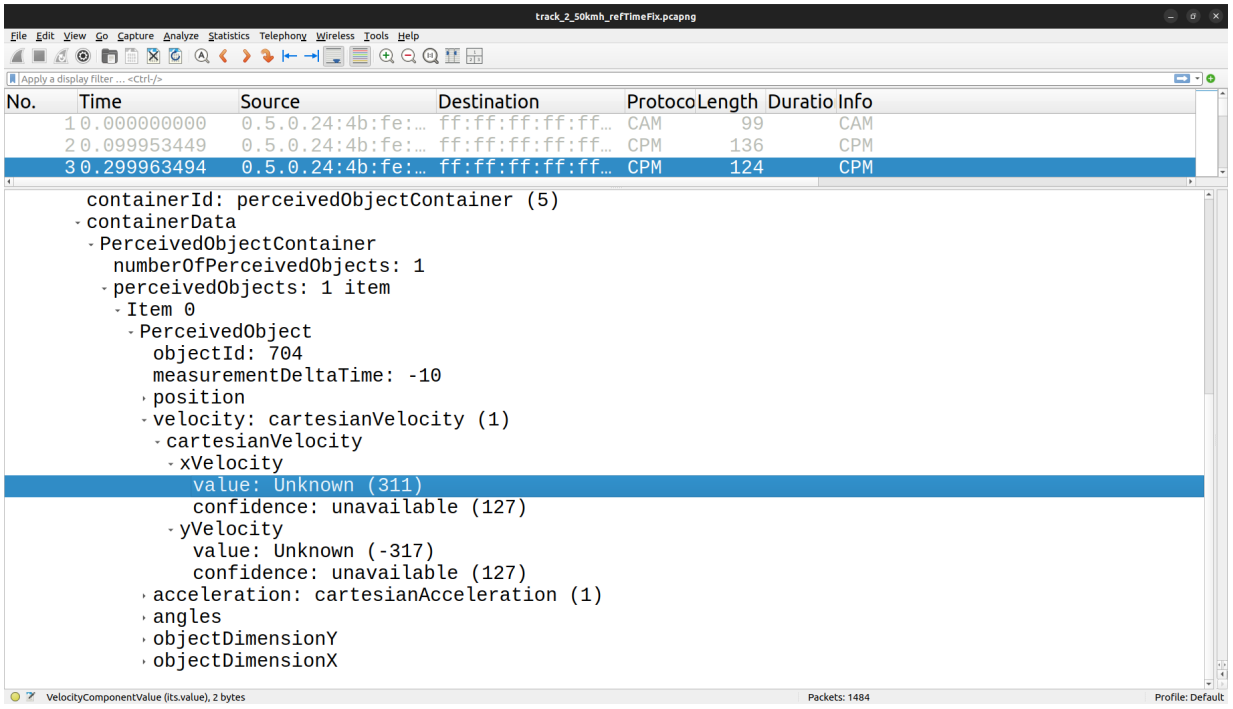
To ensure the simulated instances closely replicate real conditions, we developed a dedicated tool: TRACEN-X (Telemetry Replay and Analysis of CAN Bus and External Navigation Data) [GRCR+25]. TRACEN-X is a software tool specifically designed to support the development and validation phases of vehicular field tests. It provides two main functionalities: Record and Replay. The Record function is used during living lab tests to log key data from the vehicle, including positioning information from the Global Navigation Satellite System (GNSS) and telemetry from the Controller Area Network (CAN) bus, covering all relevant on-board sensor data. This allows us to capture a comprehensive dataset representing the vehicle's behaviour during actual test drives. The Replay function can then reproduce this data in a controlled environment, reading from the log files generated by the Record phase. This makes it possible to recreate the exact conditions of a field test in a lab setting.

From this perspective, TRACEN-X proves essential during the development and validation of vehicular applications. TRACEN-X is especially valuable for the SMTD use case: by replaying a real-world scenario in the lab, we can manually introduce controlled misbehaviours into the data stream, providing a safe and repeatable way to evaluate the MBD and TAF systems under adverse or faulty conditions.

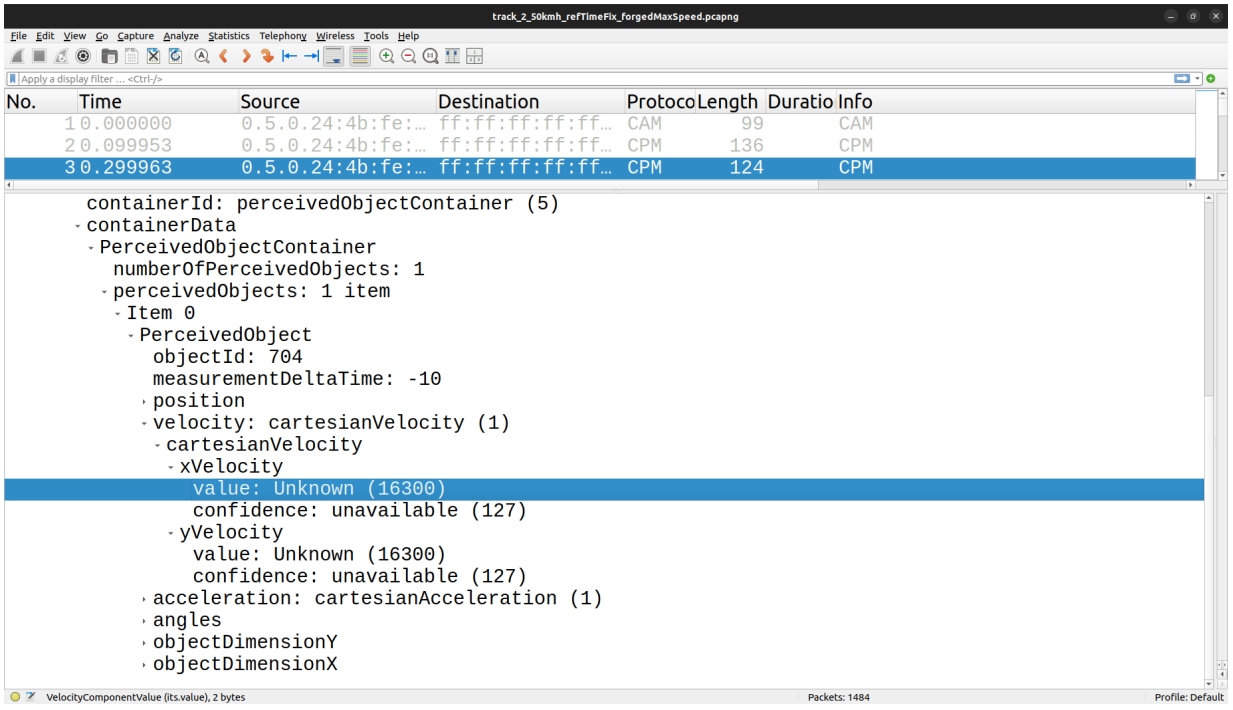
An example of this is shown in Figure 5.3, where we used the Replay function of TRACEN-X to reproduce data previously collected during a field test. This approach allows us to generate V2X messages in a controlled laboratory environment exactly as they were originally produced during the test track experiment. Figure 5.3 displays the V2X messages generated by simulated vehicle instances and captured using the well-known packet analysis tool Wireshark. Specifically, Figure 5.3a shows a genuine trace replayed with TRACEN-X, while Figure 5.3b presents a trace where a misbehaviour (MB) was intentionally introduced.

In particular, the two figures highlight a field from a CPM message containing the velocity of a perceived object. In Figure 5.3a, the Cartesian velocity components are 311 and -317, corresponding to 3.11 m/s and -3.17 m/s respectively (plausible values for a moving vehicle). By contrast, Figure 5.3b shows the same field with altered values: both components are set to 16300, which translates to 163 m/s (or 586.8 km/h), an unrealistically high speed. Despite this, such values are still syntactically valid and accepted by the ETSI standard, underlining the importance of dedicated mechanisms for detecting semantic anomalies in V2X data.

In the remainder of this deliverable, when referring to introduced misbehaviours (MBs), we will specifically mean the alteration of the velocity components of a perceived object in a CPM message, as previously



(a) Genuine CPM message captured from a field test.



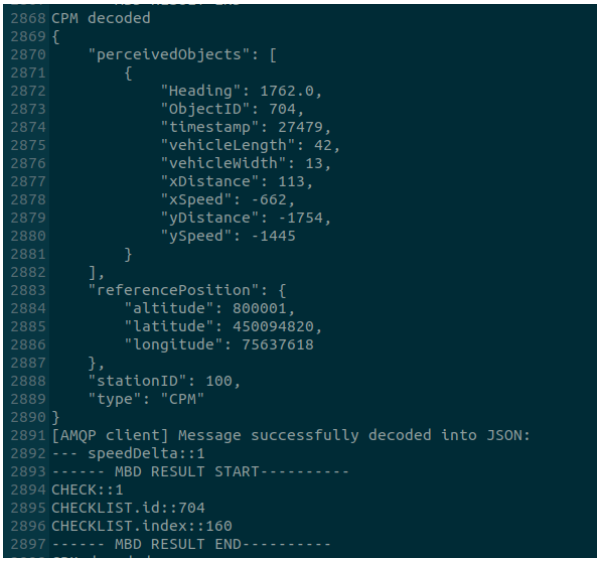
(b) CPM message captured from a field test with an introduced MB on the speed of the perceived object.

Figure 5.3: Message traces from simulated vehicle instances. (a) Trace without introduced MBs. (b) Trace with introduced MBs.

described. Based on this type of MB, we defined three distinct scenarios that will be used throughout the simulated experiments for evaluating both the MBD and the TAF. These scenarios are:

1. **Scenario without introduced MBs:** This scenario contains no manually injected misbehaviours. However, since the data originates from real field tests, it may still include *intrinsic MBs* caused by inaccurate or faulty sensor perceptions. These are naturally occurring anomalies and are not

(a) CPM with a MR. No MBs are detected.



(b) CPM with a MR. MBs are detected.

Figure 5.4: CPMs with Misbehaviour Report (MR) appended by the MBD. (a) MR with no MB detected. (b) MR with MBs detected.

artificially introduced;

2. **Scenario with MBs in all CPMs:** In this case, every CPM message is manually modified to include the same type of MB: the Cartesian velocity components of a perceived object are set to an extremely high value (i.e., 163 m/s);

3. **Scenario with MBs in one third of the CPMs:** This scenario introduces the same type of MB as above, but only in approximately one out of every three CPM messages at random.

These three scenarios will serve as the basis for all evaluations of the MBD and TAF using simulated vehicle instances. Additionally, in Section 5.6.1, we will describe the final set of experiments conducted in living lab with real vehicles. Naturally, only the scenario without introduced MBs can be reproduced in the real-world tests.

5.2.3 MBD implementation

The Misbehaviour Detector (MBD) is the first software component that processes the V2X messages (CAMs and CPMs) generated by the vehicles, whether real or simulated, after they have been received and decoded by the AMQP client. The primary objective of the MBD is to analyse each incoming V2X message and perform various checks (e.g., plausibility checks, consistency checks, topology checks, behavioural checks, etc.) to detect potential misbehaviours (MBs).

It is important to note that the MBD used in this use case is the same component developed by SystemX for the Intersection Movement Assist (IMA) use case. Therefore, the internal mechanisms and logic of the MBD will not be detailed in this section, as they are already described in the documentation of the IMA use case (see Chapter 3). Instead, this section will focus on evaluating how the MBD behaves when applied to the input data generated in our specific context.

Specifically, each V2X message is provided as input to the MBD, which then appends a Misbehaviour Report (MR) to the message as output. Figure 5.4 illustrates two examples of CPMs with their corresponding MRs appended by the MBD. In Figure 5.4a, the CPM is associated with an MR indicating no misbehaviour

detected. In this case, the MR contains only the entry “CHECK::0”, which signals that the message passed all the checks without raising any issues.

Conversely, Figure 5.4b shows a CPM accompanied by a MR, reporting detected misbehaviours. Here, the string “CHECK::1” indicates that one or more misbehaviours have been identified. The entry “CHECK-LIST.id::704” refers to the ID of the perceived object involved in the detected MB, while “CHECK-LIST.index::160” represents the type(s) of misbehaviour(s) identified.

This last value corresponds to a bitmask that encodes the presence of different types of misbehaviours. Each bit in the binary representation of the bitmask indicates a specific misbehaviour type: a bit set to ‘1’ means that the corresponding misbehaviour has been detected. The complete list of misbehaviour types and their corresponding bitmask values is shown in Figure 5.5.

For example, the MR in Figure 5.4b shows a bitmask value of 160, which corresponds to “10100000” in binary. This means that two specific misbehaviours were detected in the CPM: an inconsistency between position and speed based on a Kalman filter (i.e., KalmanPosSpeedConsS), and an issue related to the local perception verification of the perceived object (i.e., LocalPerceptionVerif).

5.2.4 TAF implementation

The Trustworthiness Assessment Framework (TAF) represents the second software component in the SMTD architecture and is, in fact, the core element of the use case. Its main function is to process all incoming V2X messages, along with their associated MRs, and produce as output an Actual Trust Level (ATL), which can be assigned either to a perceived object or to the ego-vehicle.

As the internal mechanisms of the TAF have already been thoroughly described in the D3.x series of deliverables, this section will not repeat those details. Instead, it will focus on the practical implementation aspects within the SMTD use case. Specifically, we will describe how the message exchange required for TAF initialization was handled, how input and output messages were managed through the appropriate TAF communication topics, provide an overview of the Trust Model adopted in the SMTD use case, and finally present the results obtained from the TAF in terms of Trust Level evaluation across the different scenarios, along with an analysis of the end-to-end latency of the full processing pipeline. In general, throughout the evaluations in the context of the SMTD use case, we take into account all available outputs provided by a TAF report. Specifically, in the majority of the evaluations we take into consideration the ATL

Values of `check` Field in `observationSet`

Bitmask	Detector
`00000000000000000000000000000001`	distPlau
`00000000000000000000000000000010`	SpeePlau
`00000000000000000000000000000100`	SpeeCons
`00000000000000000000000000001000`	PosSpeeCons
`00000000000000000000000000010000`	KalmanPosCons
`00000000000000000000000000100000`	KalmanPosSpeedConsS
`00000000000000000000000001000000`	KalmanPosSpeedConsP
`00000000000000000000000010000000`	LocalPerceptionVerif

Figure 5.5: Bitmask of the MBs detected.

output as this allows us to examine the dynamic nature of runtime, evidence-based trust characterizations. For this purpose, we examine the ATL value both in terms of the resulted projected probability (i.e., scalar number from 0-1) as well as the resulted trust opinion and its internal components, namely belief, disbelief and uncertainty. Finally, in the context of the task offloading scenario we leverage the final trust decision derived as a result of the comparison of the ATL value against the predefined RTL values as defined in the corresponding trust model template. This allows the task offloading inference process to continue receiving camera data if and only if there is sufficient evidence about the trustworthiness of the sensing vehicle.

The message exchange during the TAF initialization phase is illustrated in the diagram in Figure 5.6. The TAF relies on Kafka topics to manage both input and output communication with adjacent components in the software pipeline. The main internal topics involved are the *mdb topic* and the *taf topic*. The initialization process is structured as a six-way handshake that leverages these topics for coordination.

The sequence begins with a TAS_INIT_REQUEST message, followed by an MBD_SUBSCRIBE_REQUEST message. This subscription request includes a Request ID, which must be echoed in the subsequent MBD_SUBSCRIBE_RESPONSE message. Once the MBD subscription is acknowledged, the TAF replies with a TAS_INIT_RESPONSE message that contains a Session ID. This Session ID is then used to issue a TAS_SUBSCRIBE_REQUEST, and in response, the system receives a TAS_SUBSCRIBE_RESPONSE message containing a Subscription ID.

The obtained Subscription ID must be included in all V2X messages and their associated MRs submitted to the TAF. The output of the TAF, comprising the computed Actual Trust Levels (ATLs), is encapsulated in TAS_NOTIFY messages, which are published to a dedicated Kafka topic named *application.smtd topic*.

To properly manage the TAF interfaces, a dedicated script was developed to handle the generation, transmission, and reception of all messages involved in the TAF initialization phase. This script is also responsible for formatting the V2X messages along with their corresponding MRs and sending them in input to the TAF. A second, separate script is used to listen for the TAF output, extract the resulting Actual Trust Levels (ATLs), and record latency measurements. Finally, the ATLs are forwarded to the last software component in the pipeline, the Traffic Control Center (TCC), for visualization through the tablet Local Dynamic Map (LDM) interface.

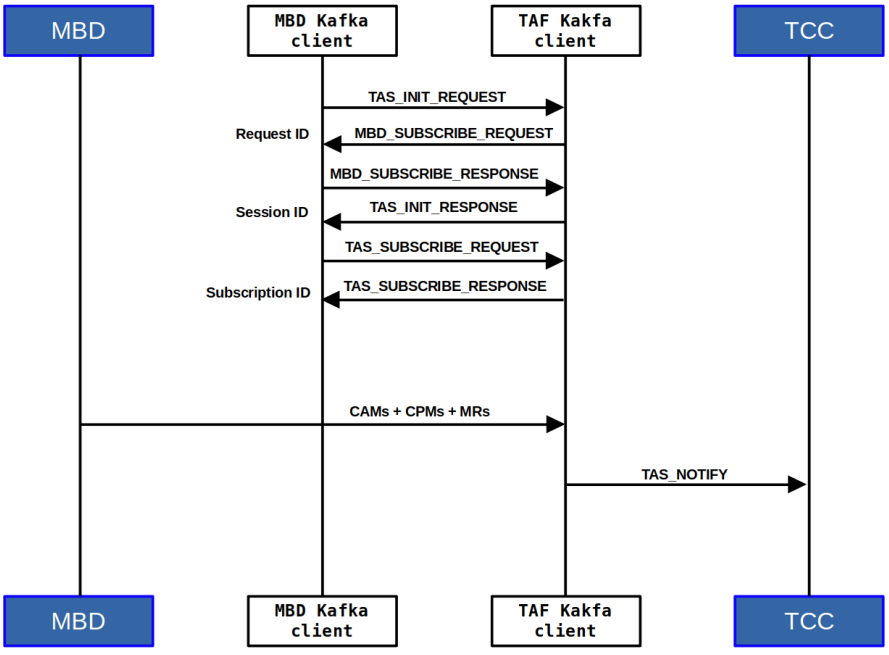


Figure 5.6: Messages being exchanged during TAF initialization.

Figure 5.7: Stellantis test track in Orbassano shown from the LDM.

5.2.5 TCC and LDM implementation

The final component of the MECC architecture is the Traffic Control Center (TCC). The purpose of this module is to emulate a central entity (e.g., an infotainment agency, local authority, or other traffic management organization) responsible for overseeing and controlling vehicular traffic within a specific area. These entities are commonly referred to as Traffic Control Centers (TCCs), as they serve as central hubs for collecting traffic-related data and V2X message flows. The primary objectives of a TCC are to manage critical traffic events, disseminate relevant information to drivers, and deploy intelligent vehicular applications.

In the SMTD use case, the TCC acts as the final component of the processing chain. It receives both the V2X messages generated by the vehicle and the Actual Trust Levels (ATLs) computed by the Trustworthiness Assessment Framework (TAF), which are based on the Misbehaviour Reports (MRs) produced by the Misbehaviour Detector (MBD) and on the Trustworthiness Claims (TCs) produced by the Trustworthiness Claims Handler (TCH).

Beyond simply aggregating all the data generated in the use case, the TCC builds a so-called Local Dynamic Map (LDM). The LDM, standardized by ETSI [ETS14a], is designed to provide a real-time representation of the environment, combining both static and dynamic information relevant to a specific area. In this context, the TCC functions as the backend of a web-based application, which receives and updates all motion and situational data to construct an LDM instance. Simultaneously, a tablet installed in the vehicle accesses the frontend of this web interface, allowing it to display the same real-time information to the driver.

Figure 5.7 provides an example of the static portion of the LDM, centered around the Stellantis test track in Orbassano, Turin.

5.2.6 TCH implementation

The shift of this use case towards the MECC architecture presented in Figure 5.2 places the TAF instance outside of the source of the examined V2X data. This requires the need for trustworthiness evidence that characterize the overall trustworthiness of the V2X messages (i.e., CPM/CAM) that are transmitted from real (and simulated) vehicles. This involves not only the misbehaviour reports that are constructed within the MECC environment, but also concrete trustworthiness evidence pertaining to the security properties of the source V2X entities. The Trustworthiness Claims Handler (TCH) is a component of the *CONNECT* framework and its purpose is to aggregate and convey evidence about the trustworthiness of the

source V2X objects, in the form of Trustworthiness Claims (TCs). For example, in the context of this use case, Trustworthiness Claims are transmitted from the vehicles towards the MECC providing evidence about the configuration integrity of the in-vehicle topology.

The implementation details of the TCH component are presented in Section 2.2.1. In the context of this evaluation, there are two TCH instances that are spawned: i) the first one is deployed in the vehicle side and periodically transmits TCs through dedicated Kafka channels to the MECC server, and ii) the second one is responsible for consuming all incoming TCs, verifying their authenticity and forwarding them to TAF for further trust quantification calculations. Regarding the former TCH instance in the context of the simulated vehicles, we have extended the TCH instance to provide simulated Trustworthiness Claims describing the state of each of the simulated vehicles. However, in the living lab case we have deployed the actual TCH instance as a docker service. Once configured, the TCH instance aggregates and transmits TCs from the CONNECT-equipped vehicle side towards the MECC server. An example of a TC in the context of the living lab, end-to-end evaluation is captured below:


```
{
  "sender": "tch-vec",
  "serviceType": "ECI",
  "messageType": "TCH_NOTIFY",
  "message": {
  "tchReport": {
    "trusteeID": "vehicle_19",
    "trusteeReports": [
      {
        "attestationReport": [
          {
            "appraisal": 1,
            "claim": "SECURE_BOOT",
            "timestamp": "2025-05-02T10:42:00Z"
          },
          {
            "appraisal": 1,
            "claim": "ACCESS_CONTROL",
            "timestamp": "2025-05-02T10:42:00Z"
          },
          {
            "appraisal": 1,
            "claim": "CONTROL_FLOW_INTEGRITY",
            "timestamp": "2025-05-02T10:42:00Z"
          },
          {
            "appraisal": 1,
            "claim": "SECURE_OTA",
            "timestamp": "2025-05-02T10:42:00Z"
          },
          {
            "appraisal": 1,
            "claim": "APPLICATION_ISOLATION",
            "timestamp": "2025-05-02T10:42:00Z"
          },
          {
            "appraisal": 1,
            "claim": "CONFIGURATION_INTEGRITY_VERIFICATION",
```

```
            "timestamp": "2025-05-02T10:42:00Z"
          }
        ]
      }
    ]
  },
  "evidence": {
    "timestamp": "2025-05-02T10:42:00Z",
    "signatureAlgorithmType": "ECDSA-SHA256",
    "signature": "84137ee2f7edf21f2ba...",
    "keyRef": "vehicle_19_public_key"
    }
  }
}
```


Listing 5.1: TC transmitted by the TCH instance deployed on the in-vehicle OBU. basicstyle

The aforementioned snippet captures a TC payload that represents that the in-vehicle topology in V2X object with id ”vehicle_19” is at a correct state; according also to the set of claims and appraisals that are provisioned as part of the overall trust assessment framework - i.e., the trust model in the context of this use case. To enable a thorough evaluation and how the TCs impact the overall end-to-end evaluation, the in-vehicle TCH instance is developed with a configurable periodicity associated with the TC transmission. To provide enhanced flexibility in the simulated evaluations, the TCH functionalities are also able to provide simulated TCs emulating evidence from different V2X entities with varied reported evidence.

5.2.7 Task offloading: testing methodology and SMV algorithm

The CONNECT task offloading (TO) functionality is tested during the SMTD sessions in the CRF test track. The required hardware installed in the CONNECT experimental vehicle is exactly the same with the one used for our lab testing, depicted in Figure 5.8 left and detailed in D5.3 [Con25b]. Namely, it’s the OBU hosting the offloading and the CONNECT trusted computing base applications, a 5G modem with an Italian SIM card and the camera capturing the front area of the vehicle. Appropriate power resources were used to support all the needs of the offloading pipeline, the topology of which was (for confidentiality reasons) not linked to the in-vehicle network (see Figure 5.8 right).

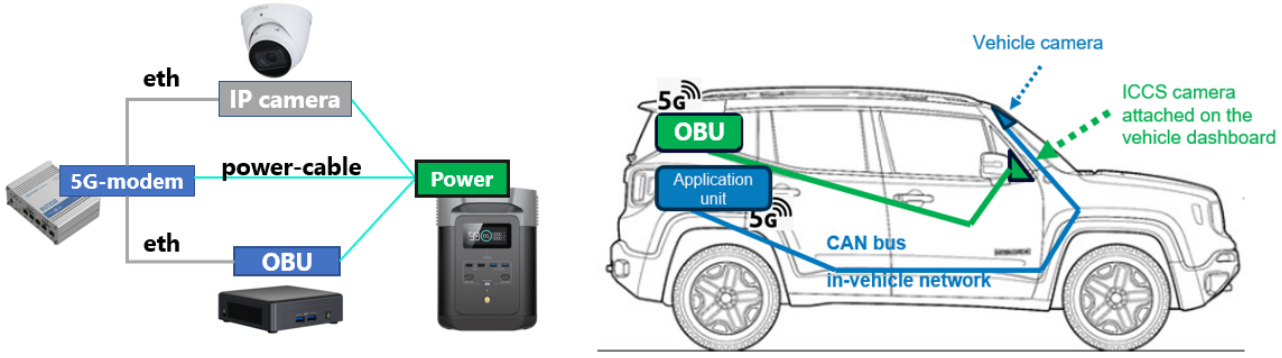


Figure 5.8: Left: The offloading pipeline hardware modules and their topology Right: Offloading pipeline topology installed in the vehicle independently from the typical in-vehicle network bus.

Our inference application located in the MECC, when fed by the CONNECT vehicle video data, identifies the proceeding vehicle and specifies when events of slow moving vehicle(s) take place. For the latter needs, we model the actual test track driving setup (i.e., a 'regular' passenger car moving in front of the CONNECT equipped vehicle, as depicted in Figure 5.11 picture) with the schematic of Figure 5.9. In detail, on time t1 we assume the Ego Vehicle (EgoV) located at the position with coordinates (Long, Lat) moving with speed S_E and a vehicle in front, called Remote Vehicle (RV) moving with speed S_r.

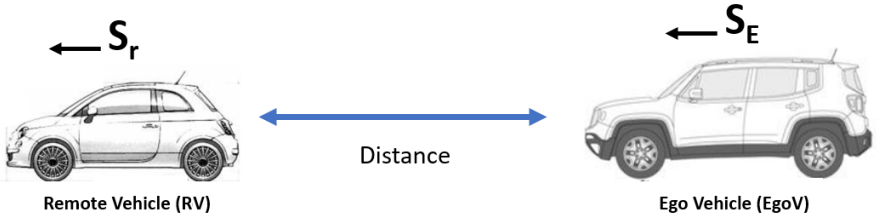


Figure 5.9: The testing vehicles (EgoV being the CONNECT-equipped one) and the involved quantities

In order to experiment on the inference of slow moving traffic at the MECC, based on offloaded video data, we introduce two relevant driving scenarios whereby different instances of slow-moving traffic are realised. Those serve as the basis for our measurements:

TO Scenario 1: Slow Moving Vehicle event due to sharp speed reduction of the leading vehicle. In this case, both RV and EgoV start moving with 40Km/h and gradually create/keep a distance of about 70m. Suddenly RV "breaks" [i.e., heavily reduces speed] and is now moving with 20Km/h while the EgoV continues moving with 40Km/h until reaching a distance of 20m from the RV. At around that distance, both vehicles fully break and stop, marking the end of the scenario.

TO Scenario 2: Slow Moving Vehicle event due to common low speed. In this scenario, both the RV and EgoV start moving with 50Km/h and maintain a distance of about 40m. Both the RV and the EgoV gradually reduce their speed to 15Km/h while keeping the same distance of 40m. After having reached a speed of around 15Km/h, they continue moving (with that speed) for 2 minutes and then they stop, ending this trial scenario.

Before the realisation of the two scenarios, the two vehicles performed some rounds to collect reference measurements and fine-tune the aforementioned velocity/distance values. Furthermore, the parameters of an algorithm (discussed below and) required to recognise slow moving traffic instances, were also tuned.

The introduced Slow Moving Vehicle Algorithm (SMVA) is a simple algorithm, running at the MECC, to identify a slow moving vehicle in front of the ego vehicle utilising the CONNECT trusted offloading mechanism. The algorithm generates an alarm in the following Algorithm cases:

1. The remote vehicle is standing still
2. The remote vehicle is moving below a defined vehicle value (configurable)
3. Both remote and ego vehicle are moving with similar low speeds
4. The relative distance between the ego and remove vehicle is decreasing quickly

SMVA is typically executed every second. During the trials, the algorithm was configured to run every 2 seconds. SMVA communicates with the data distribution process which gathers vehicle information (and is also located on the MECC) through sockets and retrieves the latest values for the following parameters: a) latitude of Ego position; b) longitude of Ego position; c) the X component of distance between Ego and

Remote vehicle; d) the Y component of distance between Ego and Remote vehicle; e) the X component of Remote vehicle speed; f) the Y component of Remote vehicle speed.

The algorithm initially calculates the actual distance between the Ego and the Remote vehicle using the Ego's X and Y speed components, as well as the speed of the Remote vehicle using its X and Y speed components. Then, it calculates the (relative) acceleration, which determines how fast the following vehicle is approaching the front vehicle. Clearly, a positive value for the acceleration suggests that two vehicles are converging, while a negative value means that they are moving away from each other.

SMVA starts by analysing the speed of the remote vehicle and checks if this speed is zero or if this speed is below a defined threshold (sp_{th}). In the first case, Algorithm case (1) is fulfilled, while in the second case Algorithm case (2) is fulfilled. In both cases, an alarm is generated. An alarm is illustrated in the HMI with a red circle around the position of the incident. In addition, if the speed of the remote vehicle is low and the distance is kept steady, then the Algorithm case (3) is fulfilled. Again, a red circle is presented in the vehicle dashboard. If none of the above cases are valid, then SMVA is continuously monitoring the relative acceleration between the two vehicles. If the relative acceleration exceeds a specific threshold (acc_{th}), the algorithm case (4) becomes valid, and an alarm is generated as well.

During our lab experimentation with the algorithm, specific values for the parameters: a) frequency of algorithm execution (dt); b) sp_{th} and; c) acc_{th} were identified. During the actual testing in the Stellantis test-track, these parameters were fine-tuned in order to effectively accomplish the objectives of the trial. The fine tuned values of the parameters are marked below: a) dt = 2 seconds; b) sp_{th} = 4.17m/s = 15Km/h; c) acc_{th} = $8.34m/sec^2$.

5.3 Trust Model

The Trust Model for assessing the trustworthiness of CPMs received by the MEC running the Traffic Control Center (TCC) is shown in Figure 5.10. The TCC uses the CPMs as input to functions which help monitor the traffic and detect jams. As such, CPMs must not have their integrity compromised i.e., they must not have been manipulated with.

- **Location of the TAF**: MEC TCC
- **Root node / agent of the TM**: MEC TCC that runs the traffic detection function, M_{TCC}
- **Propositions of the TM**: Observations inside CPMs, $C_{x|x}$, $C_{x|y}$, etc.
- **Other Trust Objects in the TM**: Sender vehicle, V_x

The TAF and the trust model in this case are hosted on the MEC. The trust model is similar to the trust model for the Intersection Movement Assist Scenario 2, with the exception that the root node is the MEC and not the ego vehicle. A trust model instance is created for each sender vehicle as in the IMA Scenario 2. The propositions are observations representing all the vehicles that have been observed and reported inside a CPM sent by the sender vehicle.

The MEC will build direct opinions on the propositions through the use of a MBD system, as well as through the evidence - i.e., TCs - from the sender vehicle about its own trustworthiness. The MEC uses this evidence to build a secondary set of opinions on the observations.

5.3.1 Trust Opinions

There are the following trust opinions in this TMI:

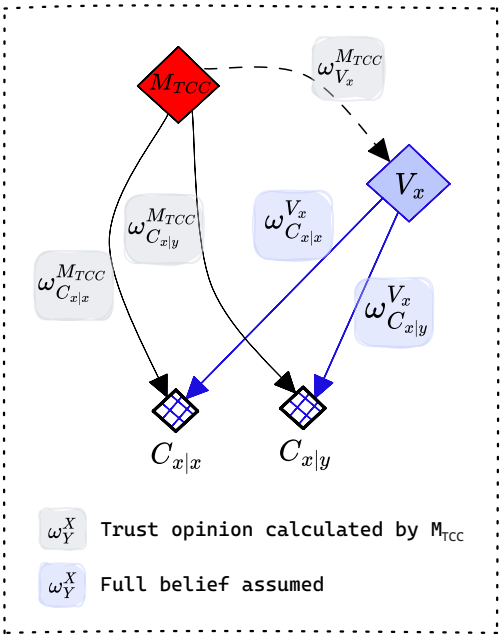


Figure 5.10: SMTD - Trust Model on the MEC TCC

1. $\omega^{M_{TCC}}_{V_x}$ = the trust opinion assessed by the **MEC TCC**, M_{TCC}, on the trustworthiness of the **vehicle** V_x to send messages (CPMs) whose integrity has not been compromised.
2. $\omega^{M_{TCC}}_{C_{x|x}}$ = the trust opinion assessed by the **MEC TCC**, M_{TCC}, on the trustworthiness of the **observation about the vehicle** V_x, contained in the most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.
3. $\omega^{M_{TCC}}_{C_{x|y}}$ = the trust opinion assessed by the **MEC TCC**, M_{TCC}, on the trustworthiness of the **observation about the vehicle** V_y, contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.
4. $\omega^{V_x}_{C_{x|x}}$ = the trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about itself**, contained in the most recent CPM, $C_{x|x}$, with respect to its integrity not being compromised.
5. $\omega^{V_x}_{C_{x|y}}$ = the trust opinion assessed by the **sender vehicle**, V_x, on the trustworthiness of the **observation about the vehicle** V_y, contained in the most recent CPM, $C_{x|y}$, with respect to its integrity not being compromised.

5.3.2 Trust Sources and Evidence

Given that this is a standalone TAF, it does not request trust opinions from external entities. The standalone TAF, however receives evidence from external trust sources, such as another vehicle. What follows is a list of all the different trust sources and evidence to be collected for each trust opinion in the trust model. Moreover, a trust quantification approach is given for each evidence.

1. $\omega^{M_{TCC}}_{V_x}$ $\rightarrow$ the **MEC TCC** assesses this opinion based on the evidence which it receives from the sender vehicle about its own trustworthiness in form of verifiable presentations.
   - **Trust Source** = TCH

- **Evidence** = Verifiable presentations - Verifiable presentations include the output of the single security controls of the vehicle computer of the sending vehicle if the security controls are not implemented (value: -1), are implemented and detected something (value: 0) or are implemented and did not detect anything (value: 1)
- **Trust Quantification Approach** = The approach for calculating the trust opinion based on the verifiable presentations is described in D6.1 [Con24b] and D3.3 [Con25c].

2. $\omega_{C_{x|x}}^{M_{TCC}} \rightarrow$ the **MEC TCC** assesses this opinion based on the results of the Misbehavior Detection system running on the **sender vehicle**
   - **Trust Source** = MBD system on the TCC
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
3. $\omega_{C_{x|y}}^{M_{TCC}} \rightarrow$ the **MEC TCC** assesses this opinion based on the results of the Misbehavior Detection system running on the sender vehicle messages
   - **Trust Source** = MBD system on the TCC
   - **Evidence** = Output of misbehavior detectors
   - **Trust Quantification Approach** = The approach for calculating the trust opinion based on the output of the misbehavior detectors is described in D6.1 [Con24b] and D3.3 [Con25c].
4. $\omega_{C_{x|x}}^{V_x}$ = the sender vehicle assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.
5. $\omega_{C_{x|y}}^{V_x}$ = the sender vehicles assumes this to be full belief = $(1, 0, 0, 0)$ and performs no assessment.

5.3.3 Actual Trustworthiness Level

$ATL_{C_{x|x}}$ is obtained in two steps: 1) first, $\omega_{V_x}^{M_{TCC}}$ and $\omega_{C_{x|x}}^{V_x}$ are discounted to obtain $\omega_{C_{x|x}}^{M_{TCC};V_x}$, 2) second, the resulting opinion $\omega_{C_{x|x}}^{M_{TCC}}$ is fused with the direct opinion $\omega_{C_{x|x}}^{M_{TCC}}$ to obtain the ATL. We are using the opposite-belief trust discounting and the cumulative fusion operators in this example.

$$ATL_{C_{x|x}} = \omega_{C_{x|x}}^{M_{TCC}} \oplus (\omega_{V_x}^{M_{TCC}} \otimes \omega_{C_{x|x}}^{V_x}) \tag{5.1}$$

Similarly, $ATL_{C_{x|y}}$ is also obtained in two steps: 1) first, $\omega_{V_x}^{M_{TCC}}$ and $\omega_{C_{x|y}}^{V_x}$ are discounted to obtain $\omega_{C_{x|y}}^{M_{TCC};V_x}$, 2) second, the resulting opinion $\omega_{C_{x|y}}^{M_{TCC};V_x}$ is fused with the direct opinion $\omega_{C_{x|y}}^{M_{TCC}}$ to obtain the ATL. We are using the opposite-belief trust discounting and the cumulative fusion operators in this example.

$$ATL_{C_{x|y}} = \omega_{C_{x|y}}^{M_{TCC}} \oplus (\omega_{V_x}^{M_{TCC}} \otimes \omega_{C_{x|y}}^{V_x}) \tag{5.2}$$

5.4 User Story Realisation

The SMTD use case counts six user stories (from SMTD.US.1 to SMTD.US.6). In the previous deliverable D6.1 some Key Performance Indicators (KPIs) were assessed. Specifically, among others, all the KPIs of SMTD.US.3 were satisfied, and therefore SMTD.US.3 will be considered now completed and will not be

addressed in the present deliverable. In this deliverable, we are implementing and testing the following user stories:

[SMTD.US.1]: As a Traffic Control Centre, I want to be able to receive events (from the MECC) on the position of both equipped and non-equipped vehicles based on trustworthy data.

[SMTD.US.2]: As a Driver, I want to be notified in real-time of any traffic congestion and/or blocking points that may affect my journey. In particular, I want to be able to receive correct and trustworthy traffic congestion events based on the LDMs calculated at the MECC Level.

[SMTD.US.4]: As a Traffic Control Center, I want to be able to enhance my LDM with trustworthiness levels of all vehicles based on plausibility checks performed by the MD service.

[SMTD.US.5]: As a vehicle, I want to be able to offload resource-demanding tasks to the MECC.

[SMTD.US.6]: As a Traffic Control Centre, I want to receive from the vehicles the CAM and CPM messages generated in the last sixty seconds (CCHP - CAM and CPM History Package) in a trustworthy, conflict-free and resource-efficient manner.

The objective of SMTD.US.1 is to verify the complete process of generating, transmitting, and receiving CAMs and CPMs from the vehicle to the MECC. In particular, this use case focuses on measuring the vehicle-to-MECC latency.

SMTD.US.2 aims to ensure the correct reception of notifications regarding slow-moving traffic ahead. These notifications must be based on perceptions evaluated and validated with an appropriate trust level computed by the TAF. As with SMTD.US.1, the KPI in this case is the end-to-end latency, which must also remain below the average human reaction time of 1.5 seconds.

The goal of SMTD.US.4 is to validate the correct computation and temporal evolution of the Actual Trust Levels (ATLs) generated by the TAF. ATLs can be derived from CPMs and linked to vehicles perceived through sensors, using both Misbehavior Reports (MRs) and Trustworthiness Claims (TCs). This use case also includes an analysis of how different TC generation frequencies impact the overall performance of the OBU. For this purpose, the end-to-end latency will be measured across scenarios with varying TC generation rates.

SMTD.US.5 is intended to assess the correct operation of the Task Offloading mechanism when resource-intensive processes are required. In particular, this use case involves detecting slow-moving traffic using a dedicated dashboard camera. The corresponding KPI concerns the total processing time for the image recognition and task offloading workflow, which must remain below 1 second.

Finally, the objective of SMTD.US.6 is to validate the correct generation and transmission of the CAM and CPM History Package (CCHP). The CCHP aggregates all relevant vehicle information collected over the previous 60 seconds. The vehicle must be able to periodically, or on demand, transmit the CCHP to a central MECC server.

5.5 KPI & Acceptance Criteria

In this section, we introduce the Key Performance Indicators (KPIs) defined to assess the overall performance and effectiveness of the CONNECT functionalities within the SMTD use case. Their evaluation will be presented in Section 5.6.

User story	KPI description	Acceptance criteria	Result
SMTD.US.1	**Vehicle to MECC latency in order to let the driver know in advance the existence of a traffic congestion ahead**	**<= 1.5s**, lower than the average human reaction time	Success
SMTD.US.2	**MECC to vehicle LDM latency in order to allow for the driver's reaction time to a received alert.**	**<= 1.5s**, lower than the average human reaction time	Success
SMTD.US.4	**The ATLs of the active V2X-vehicle evolves correctly:** it increases as more evidence of correct behaviour is gathered; it degrades as malicious behaviour is injected in the scenario.	**TRUE**	Success
	Processing complexity until LDM update: Observation's ATL assessment by the TAF. If V-TCs present, verification of the TCs and attributes extraction and emitter V2X-node's ATL assessment by the TAF.	**TRUE**	Success
SMTD.US.5	**Offloading trust calculations where the result needs to be rapidly provided to the TAF so as to not affect safety**	**<= 1s**, excluding network latency	Success
SMTD.US.6	**Duration that the information saved**	**60s**, to not overload the OBU device with too much useless information	Success
	Completeness of information, all needed signals are present.	**TRUE**, all motion characteristics of the ego-vehicle and perceived vehicle	Success

Table 5.2: Evaluated KPIs by user stories in the present deliverable.

For driver reaction time, the American National Highway Traffic Safety Administration (NHTSA) in Dec. 2016 released the public document: "Human Factors Design Guidance For Driver-Vehicle Interfaces" [Adm] in which it reports that ISO 15623 indicates that there should be a minimum of 0.8 seconds for the driver to respond to the alert, with the implemented value for driver brake reaction time as a design parameter to be determined by the manufacturer. Driver reaction time is highly variable. Laboratory and test track studies have found effects of expectancy, driving experience, age, cognitive and visual load, arousal and fatigue, and urgency on brake reaction time. Naturalistic driving data suggests the presence of significant effects of eyes off road time, age, weather, traffic density, and lighting conditions upon brake reaction time. Taken as a whole, the research suggests that expectancy, experience, arousal, and situational urgency generally reduce brake reaction time, while increasing age, cognitive and visual loading, lower levels of arousal or fatigue, increased eyes off road time, poorer weather conditions, heavy traffic density, and poor lighting conditions are associated with increased brake reaction time. Values from research indicate that a primed (expectant of the cue) driver under optimal conditions may have a brake reaction time of under 1 s, while analysis of naturalistic data suggest that brake reaction time could range from 1.5 s to 2.5 s (and occasionally greater depending on driver state, such as distraction). However, as mentioned above, the lack of consensus within the research precludes the use of a canonical value for driver brake reaction time. Based on this NHTSA study, CONNECT decided to take 1.5 s as minimum reference value for driver reaction time related to braking in the case of a slow moving vehicle in front.

Figure 5.11: Example of a field test on the Stallantis test track.

5.6 Evaluation

In this section, we evaluate all the previously defined Key Performance Indicators (KPIs). The assessment is based on a combination of field test experiments conducted in a living lab environment and simulation-based analysis. The results from the field tests are presented in Section 5.6.1, while those from the simulation experiments are detailed in Section 5.6.2.

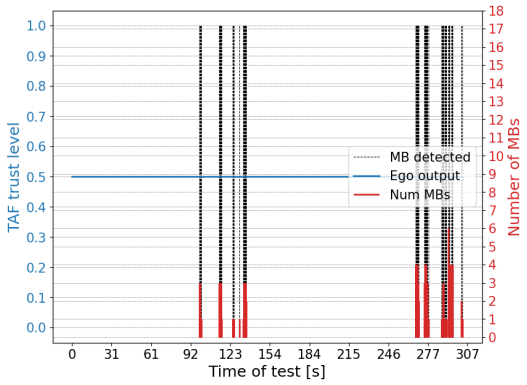
5.6.1 Field tests

As in deliverable D6.1 [Con24b], the tests described in this document were also conducted at the Stellantis CRF facility in Orbassano, near Turin. This facility includes a dedicated test track specifically designed for field testing in a controlled environment. The track consists of a circular road, which enables uninterrupted testing without external traffic interference, thus offering an ideal living lab setting.

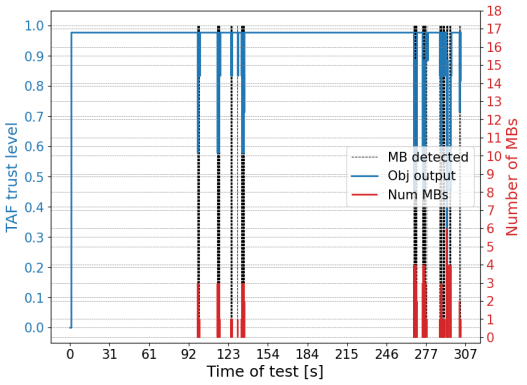
For the SMTD use case, the testing setup involved two vehicles: one equipped with a front camera, a radar sensor, and V2X transmission capabilities, and another vehicle not equipped with such systems. A detailed description of the vehicle configurations is provided in deliverable D6.1 and will not be repeated here. An illustrative example is shown in Figure 5.11. It is important to note that the choice of vehicle brands and models was solely driven by testing requirements; other Stellantis vehicles could have been used equally.

In all tests, the sensor-equipped vehicle followed the non-equipped one. Unless explicitly stated otherwise, both vehicles maintained a constant speed of 50 Km/h and a fixed distance of approximately 10 meters throughout each test session.

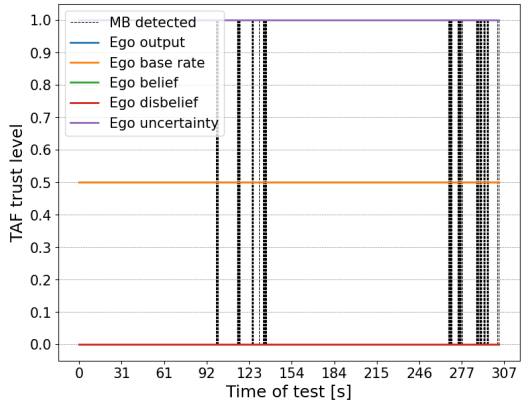
Since the tests described in this subsection were conducted in a real-world living lab environment, it was not feasible to manually inject misbehaviours (MBs) into the trace during runtime. Consequently, only the scenario without artificially introduced MBs is considered in this subsection. The complete set of scenarios, including those with MBs, is instead evaluated in the following subsection 5.6.2, which focuses on simulated results.



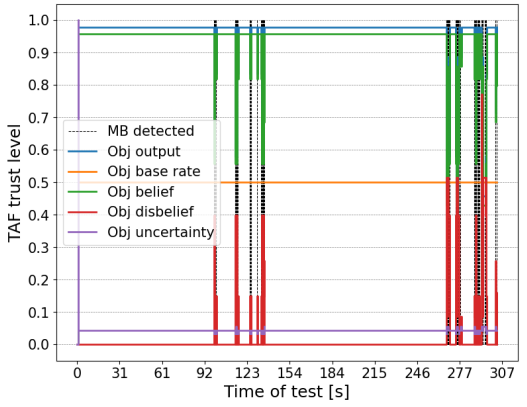
(a) ATL of the ego-vehicle.



(b) ATL of the perceived object.



(c) TAF parameters of the ego-vehicle.



(d) TAF parameters of the perceived object.

Figure 5.12: Evolution of the Actual Trust Level (ATL) in a field test experiment with no TCs sent. (a) ATL of the ego-vehicle (projected probability). (b) ATL of the perceived object (projected probability). (c) TAF parameters of the ego-vehicle (trust opinion: belief, disbelief, uncertainty). (d) TAF parameters of the perceived object (trust opinion: belief, disbelief, uncertainty).

5.6.1.1 Trust Level evolution

The scope of this section is to evaluate the correct evolution over time of the Actual Trust Level (ATL) output by the TAF during field test experiments. Specifically, during the tests, the real vehicle transmits V2X messages to the AMQP Broker running on the MECC. The AMQP client deployed on the MECC side is responsible for receiving, decoding, and forwarding the messages to the Misbehaviour Detector (MBD), which processes each message and produces corresponding Misbehaviour Reports (MRs). These MRs, along with the original V2X messages, are then passed to the Trustworthiness Assessment Framework (TAF) for the computation of the ATL.

It is important to note that the results presented in this section can be directly compared with those in Section 5.6.2, where the same experiments are reproduced using simulated vehicle instances.

All results shown in this use case were obtained using TAF version 0.5.0.

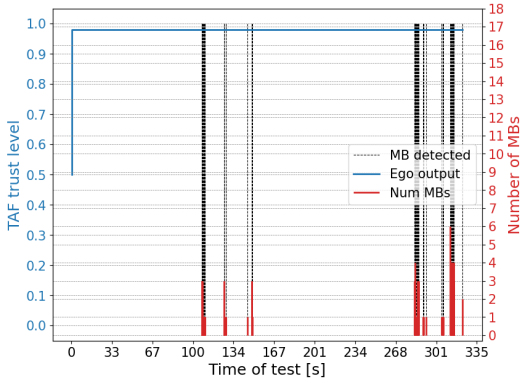
Figure 5.12 illustrates the evolution of the ATL during a field test conducted in a living lab environment, where the TCH was not activated and, consequently, no TCs were transmitted by the vehicle. It is worth noting that, although no MBs were manually introduced in this trace, intrinsic MBs were still detected by

the MBD. Each MB occurrence is marked by a dashed vertical black line in the figure. Additionally, red bars at the bottom of the figure indicate the number of MBs detected at each time step, showing that up to six MBs can be associated with a single CPM. The blue line represents the ATL as it evolves over time. As expected, when MBs are detected by the MBD, the ATL decreases from a fully trusted value (close to 100%) to a lower level. The higher the number of MBs detected within the same CPM, the more significant the corresponding drop in ATL.

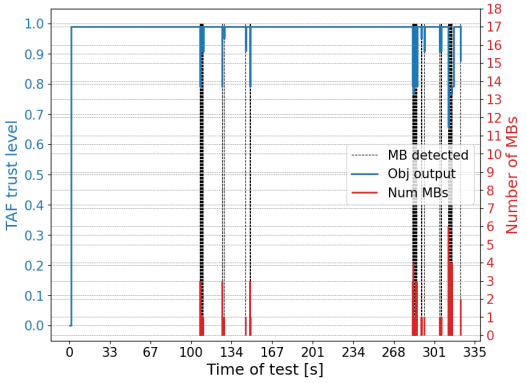
Specifically, three of the plots depict the evolution of the ATL expressed in the form of projected probability: i) Figure 5.12a shows the ATL associated with the ego-vehicle; ii) Figure 5.12c reports the corresponding TAF parameters for the ego-vehicle; iii) Figure 5.12b illustrates the ATL evolution for the object detected ahead of the ego-vehicle. Finally, Figure 5.12d presents the evolution of the ATL as a trust opinion, focusing on how each of the internal components, namely belief, disbelief, uncertainty (and base rate) evolve over the course of the test case.

From Fig. 5.12, the following conclusions can be drawn:

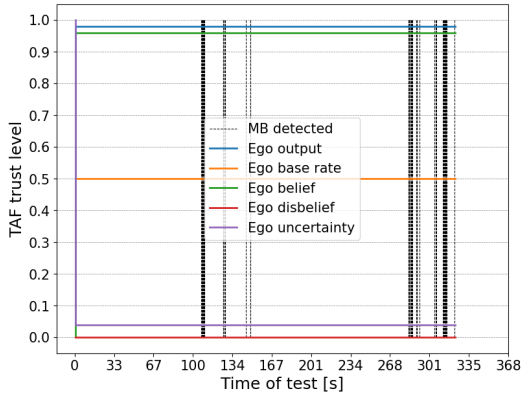
1. The ATL of the ego-vehicle remains fixed at the base rate of 50% throughout the experiment. This is because the TCH is not active, preventing the transmission of TCs that would otherwise allow the



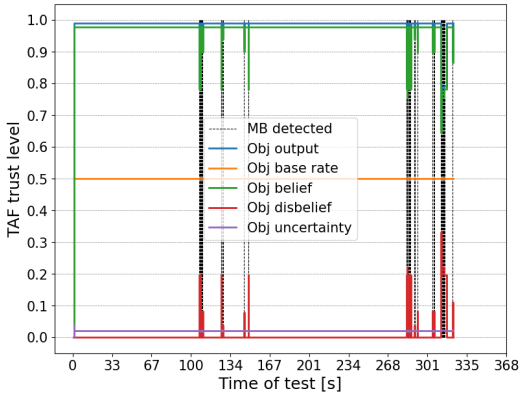
(a) ATL of the ego-vehicle.



(b) ATL of the perceived object.



(c) TAF parameters of the ego-vehicle.



(d) TAF parameters of the perceived object.

Figure 5.13: Evolution of the Actual Trust Level (ATL) in a field test experiment with TCs sent. (a) ATL of the ego-vehicle (projected probability). (b) ATL of the perceived object (projected probability). (c) TAF parameters of the ego-vehicle (trust opinion: belief, disbelief, uncertainty). (d) TAF parameters of the perceived object (trust opinion: belief, disbelief, uncertainty).

TAF to build a trust opinion of the ego-vehicle;

2. The ATL of the detected object evolves as expected: it remains at its maximum value in the absence of MBs, and decreases whenever MBs are identified by the MBD and passed to the TAF. The extent of the drop in ATL depends on the number of MBs associated with a single CPM. Notably, after a MB is detected, the ATL returns immediately to the maximum value upon the first subsequent valid perception. This behaviour stems from the trust model of the SMTD use case, that does not take into consideration the temporal evolution of trust opinions. Therefore, a trust opinion is calculated from scratch when a new event arrives without taking into consideration past events, unlike other use cases where temporal correlation is taken into account thanks to the federated TAF.

Figure 5.13 illustrates the evolution of the ATL in a field test scenario where the TCH is active and TCs are sent to the TAF, capturing the state (e.g., integrity) of the ego-vehicle. Specifically: i) Figure 5.13a shows the ATL evolution of the ego-vehicle in terms of t; ii) Fig. 5.13c reports the corresponding TAF parameters for the ego-vehicle; iii) Fig. 5.13b depicts the ATL evolution for the perceived object; and iv) Fig. 5.13d presents the associated TAF parameters of the perceived object.

Based on Fig. 5.13, the following conclusions can be drawn:

1. The ATL of the ego-vehicle reaches its maximum value as soon as the first TC is received and processed by the TAF. This is because the TAF can now formulate a trust opinion on the ego-vehicle based on the received TCs;

2. The ATL of the perceived object evolves as expected: it increases to the maximum level when a V2X message without any detected MB is processed, and drops when a MB is identified. As soon as the TAF processes a new valid V2X message (i.e., without MBs), the ATL returns to its maximum. This behaviour stems from the design of the SMTD use case, where each event is evaluated independently by the TAF;

3. The trust opinion associated with the ego-vehicle positively influences the ATL of the perceived object. This is evident when comparing this scenario (with TCH enabled) to the previous one (see Fig. 5.12b), where no TCs were transmitted. It is clear that ATL drops are more severe in the absence of TCs, whereas when TCH is active, the ATL of the perceived object is generally higher due to its association with a trusted source;

4. Interestingly, while the trust opinion of the ego-vehicle positively affects the ATL of the perceived object, the reverse does not occur (i.e., the drops in the ATL of the perceived object do not affect the ATL of the ego-vehicle). This asymmetry is intentional and relies on the trust model template designed and implemented for this use case: in the SMTD use case, the TAF processes the trustworthiness of the ego-vehicle and the perceived object as distinct and independent entities.

The results presented here prove that the ATL behaves as expected during the tests conducted with real vehicles in a living lab environment. Furthermore, the correct evolution of the ATL, both in the presence and absence of TCs, will be further validated by the results obtained from simulated vehicle instances, discussed in Section 5.6.2.

5.6.1.2 LDM visualization

This subsection focuses on the Local Dynamic Map (LDM) displayed on the tablet mounted inside the ego-vehicle. By leveraging the real-time reception of CAM and CPM messages, the MECC processes the incoming message flows and computes the real-time Actual Trust Level (ATL) associated with the received information. Both the positions of the ego-vehicle and the object detected in front (along with

(a) LDM without TCs being sent.

(b) LDM with TCs being sent.

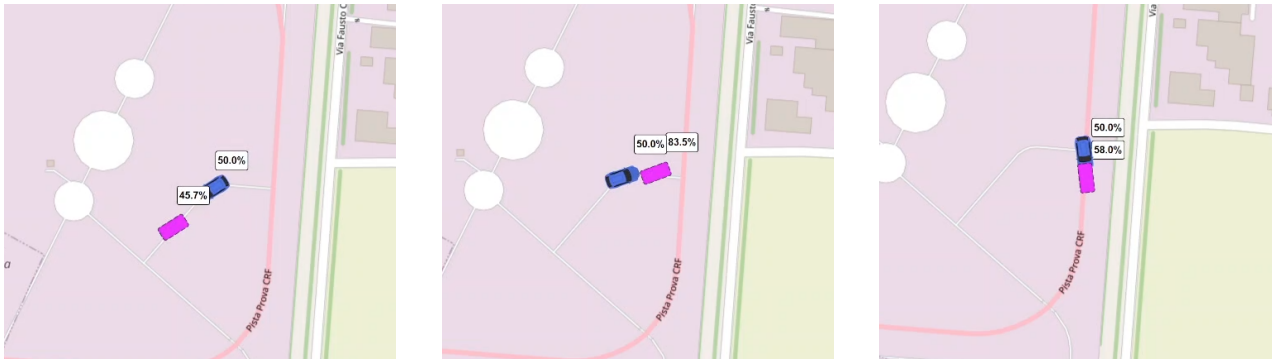
Figure 5.14: Local Dynamic Map (LDM) visualization. (a) Without TCs. (b) With TCs.

their corresponding ATLs) are visualized in real time on a dynamic map interface, referred to as the Local Dynamic Map (LDM). The LDM is rendered on a web page, making it accessible via the in-vehicle tablet.

Figure 5.14 shows two examples of the LDM visualization: the ego-vehicle is depicted with a blue car icon, while the perceived object is shown as a pink rectangular shape. These icons move in sync with the actual positions of the vehicles on the Stellantis test track, providing real-time feedback on their location. The labels adjacent to the icons display the evolution of the ATLs over time. During the test, variations in ATL can be observed directly on the LDM interface.

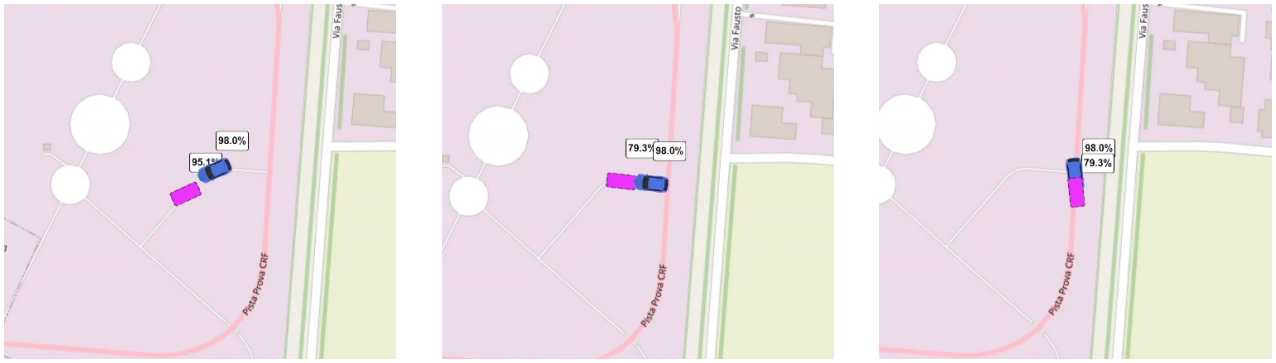
Figure 5.14a shows the ego-vehicle with a ATL of 50%. This result aligns with the findings discussed in the previous subsection, where it was demonstrated that the ego-vehicle's ATL remains fixed at its initial trust opinion (set to full uncertainty) in the trust model. The resulted projected probability of this trust opinion is eventually 0.5 showing full uncertainty on the probability of the ego-vehicle to be considered as trustworthy.

Conversely, Figure 5.14b presents a screenshot from a field test where TCH is active and TCs are transmitted to the TAF. In this scenario, no MBs are present either. Compared to the previous case, the ego-vehicle now has a ATL of 98%, consistent with prior results showing how TAF increases the trust level once TCs are processed. The perceived object also benefits from this: its ATL increases from 97.8% (without TCs)



(a) (b) (c)

Figure 5.15: MBs being detected while perceived vehicle is turning in the scenario with no TCs being sent.



(a) (b) (c)

Figure 5.16: MBs being detected while perceived vehicle is turning in the scenario with TCs being sent.

to 98.9% (with TCs), demonstrating the positive influence of a trusted ego-vehicle.

Monitoring the ATL evolution on the LDM in relation to vehicle positions also enables analysis of MB occurrences during the field tests. In particular, MBs are frequently triggered when the perceived object undergoes abrupt trajectory changes, such as during tight curves. While the Stellantis test track is a large oval with smooth curves (causing no MBs) the service roads used for entering and exiting the track feature tighter turns, which lead to false positive MB detections.

Figure 5.15 presents screenshots from a test without TCs, showing that MBs are triggered during these sharp turns. As a result, the ATL of the perceived object drops significantly (from the maximum value of 97.8% to values such as 45.7%, 83.5%, and 58%). A similar behaviour is observed in Figure 5.16, which refers to a test where TCs are transmitted. Although MBs still occur at tight turns, the ATL reductions are less severe (e.g., 95.1% and 79.3%) due to the trust propagation effect from the ego-vehicle.

This phenomenon is explained by the internal mechanisms of the Misbehaviour Detector (MBD). Among the various checks performed, one involves predicting the future trajectory of the perceived object using Kalman filters. These filters assume that future behaviour will be consistent with current observations. Therefore, if an object is perceived as moving straight, the system expects it to continue in a straight line. A misbehaviour is triggered when this expectation is violated, such as when an object begins to turn or returns to a straight trajectory after a curve, resulting in a false positive MB detection.

5.6.1.3 End-to-end latency

This subsection is dedicated to end-to-end (E2E) latency measurements. The e2e latency is defined as the time elapsed between the generation of a V2X message (or a TC) on the vehicle and the corresponding output produced by the TAF, which is then forwarded to the LDM for visualization.

To compute the e2e latency, TAF version 0.5.0 is used. In this version, a tag string is embedded in each TAF output, allowing the association of the output with its originating message based on the generation timestamp included in the tag. This enables precise computation of the e2e latency. It should be noted that, due to the internal design of the TAF, asynchronous TAF outputs are generated only when there is a variation in the ATL. Consequently, e2e latency could only be measured for a subset of the V2X messages processed by the TAF (i.e., those that result in a change in ATL).

It is also important to consider the impact of clock synchronization errors. The e2e latency computation relies on two timestamps: one from the vehicle's clock at the time of message generation, and one from the MECC's clock at the time of TAF output. Ideally, both systems should be synchronized using a Network Time Protocol (NTP) service. However, in realistic vehicular deployments, where a central server typi-

cally handles V2X traffic from thousands of vehicles, perfect synchronization of all devices is impractical. Therefore, clocks are usually left unsynchronized, and the synchronization error is estimated instead. Crucially, this clock synchronization error affects only the measurement of e2e latency, not the actual latency experienced by the system.

In our scenario, we estimated the clock offset between the vehicle and the MECC by comparing the round-trip time (RTT) and the one-way delay. The RTT was measured by sending a message from the vehicle to the MECC and immediately returning it, with the vehicle recording both the transmission and reception times. Since both timestamps are generated by the vehicle's clock, RTT is not affected by clock skew.

In contrast, the one-way delay was computed by comparing the vehicle's transmission timestamp with the reception timestamp at the MECC, and is therefore affected by clock differences. Assuming symmetric network routing (i.e., identical delay in both directions), any deviation of the one-way delay from half of the RTT is attributable to clock synchronization error.

Using this approach, we estimated the clock synchronization error between the vehicle and the MECC to be consistently below 20 ms. As a result, all e2e latency measurements presented in the remainder of this subsection should be interpreted with an uncertainty margin of ± 20 ms.

While the generation, transmission, and reception rules for CAMs and CPMs are defined by ETSI standards [ETS14b, ETS23a], and the vehicle-to-MECC propagation delay is primarily influenced by the state of the cellular network, the TAF and its auxiliary Trust Sources (e.g., TCH) represent novel contributions developed within the CONNECT project. As such, it is crucial to analyse the impact of these components on the overall e2e latency.

In particular, the TCH is implemented as an additional service running on the OBU, responsible for generating TCs at configurable transmission frequencies. This context motivates an analysis of how the overall e2e latency responds to changes in the transmission frequency of TCs. As part of this e2e evaluation we intentionally exclude the overhead for creating the actual TCs which is extensively measured in D4.3 [Con25d]. Here, we mainly focus on the overhead inflicted due to the transmission of TCs across the network and their processing in order to derive the final ATL output.

Figure 5.17 presents the measured e2e latency for various TC transmission frequencies, ranging from

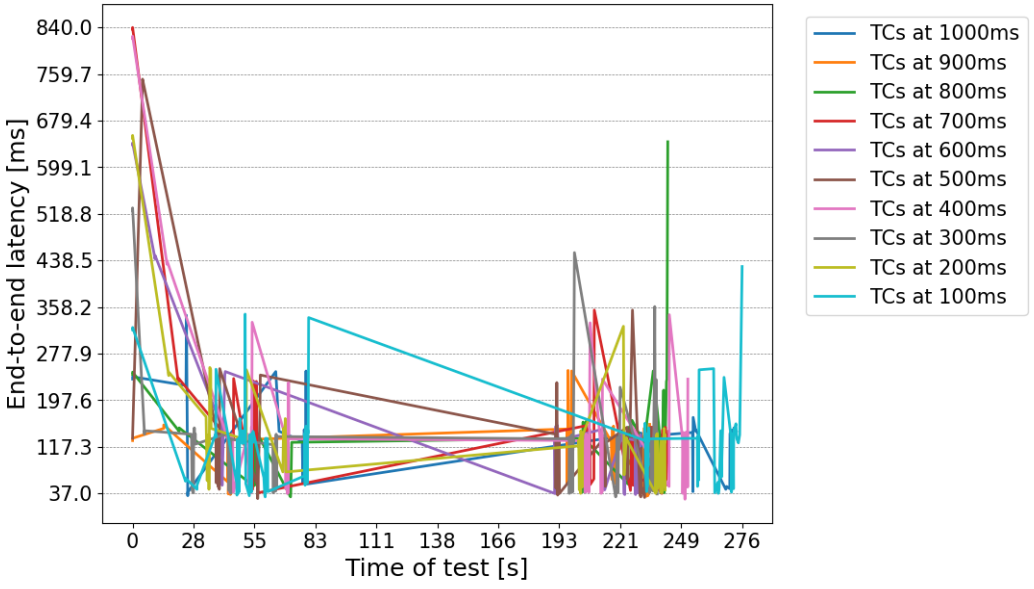


Figure 5.17: End-to-end latencies with TCs sent at different frequencies.

100 ms to 1 s, during a set of field test experiments. The results show that the complete process, from the generation of a V2X message on the vehicle to the output produced by the TAF and sent to the LDM, remains consistently below 840 ms, with average latencies between 120 ms and 130 ms.

Moreover, we can observe from the figure that the TC transmission frequency has only a marginal impact on the overall latency. This is because the OBU is capable of handling multiple V2X services concurrently. The TCH, when activated, simply operates as an additional service, introducing negligible computational overhead. As a result, the transmission of other V2X messages remains unaffected.

Importantly, the observed e2e latencies are well below the 1.5 s latency KPI defined for the CONNECT system, ensuring that both the vehicle-to-MECC and MECC-to-LDM communication paths meet the performance requirements.

5.6.1.4 Task Offloading evaluation

In our field test measurements, we evaluated the trusted offloading performance on the basis of the KPI reported in the [SMTD.US.5] user story presented in the CONNECT D2.1 [Con23c]. There, the indicator of interest has been defined to be the time interval within which the task offloading is feasible (providing insights for each potential CCAM function).

As presented in D5.2 [Con24a] and D5.3 [Con25b], we have selected a demanding task to offload rather than the typically more trivial trust calculations. Our video analytics application requires the video data from the vehicle camera to be sent to the MECC where a trained model can recognise a proceeding vehicle and estimate whether it is slow moving. As such, we are particularly interested in the end-to-end (E2E) latency, from the time the data are offloaded from the vehicle OBU up until the inference is performed in the MECC.

It is important to note that the desired E2E latency computation involves the sum of the enclaves functionality (i.e., launch and isolation operations for all involved containers), the network latency from the vehicle OBU to the MECC server and any application-specific latency. This last quantity needs to be left-out (subtracted) from the final E2E latency to alleviate any implementation details and derive a result capturing only the footprint of the CONNECT mechanisms and the network delays.

Indeed, our video analytics application, as described in D5.3 [Con25b], requires the buffering of adequate data (video frames) to subsequently feed the trained model for inference; this process required for the specific streaming protocol and ML operations induces extra delays that other implementations may be able to minimize.

Table 5.3: E2E latency for CONNECT video analytics trusted offloading

KPI	Average value (ms)	Standard deviation (ms)
E2E latency in the Reference Scenario (Local Streaming)	1381	10
E2E latency during Turin Trial Scenarios	2267	387
E2E Task Offloading Latency	885	390

Along these lines, the “E2E Task Offloading Latency” KPI, shown in the last row of Table 5.3, is defined as the difference between the measured latency of the ”E2E task offloading scenario (during the Turin trials)” shown in the second raw of Table 5.3, which includes all implementation-specific delays, and the Reference scenario. The Reference scenario is defined as the scenario in which the video is streamed locally (at the MECC server) towards a collocated inference application, meaning that the aforementioned subtraction essentially removes the frames buffering delays. The average and standard deviation values of Table 5.3 are calculated for both scenarios on (around) 80 samples. The reported measurements are collected during the defined traffic offloading scenarios realised in our field-testing (see Section 5.2.7). In both scenarios (reference and offloading) the inference process is executed every time when 40 video

frames have been received/buffered in the inference application at the MECC Server. The frame rate of the video is 30fps.

Our average value of 885ms, being within the expected range reported in [Con23c], suggest that even demanding (ML) tasks that require the shift of considerable amounts of data to the MECC can be *trustfully* offloaded to the edge infrastructure without sacrificing the utility for automotive applications; for instance, automotive ML-based predictions exhibit acceptable accuracy up to a 5 sec time horizon [M+20] implying that offloading mechanisms with the CONNECT achievable latency can be particularly useful.

Table 5.4: Network and energy measurements for CONNECT video analytics trusted offloading

KPI	Average value	Standard deviation
Throughput Downlink (Kb/s)	52.27	19.85
Throughput Uplink (Kb/s)	3566.37	1328.87
Vehicle OBU power consumption (Watts)	3.87	1.0
MECC power consumption (Watts)	27.8	5.6

Further to the latency measurements, we have carried-out some more experiments (beyond the D2.1 defined KPIs) to gain more insights on the CONNECT offloading performance. Table 5.4 shows throughput and energy performance indicators. The “Throughput” KPI is defined as the data rate between the Application Server (located in the vehicle OBU) and the Application Client (located in MECC). The uplink direction is denoted as the direction from the OBU to the MECC. The average and standard deviation values are calculated on around 300 samples. The reported quantities reflecting the one-way (uplink) direction of the offloaded data were collected during the execution of our driving scenarios using the `ifstat` tool on the specific (vpn) interface between the vehicle OBU and the MECC server.

Finally, in the same table, the “Vehicle OBU power consumption” KPI is defined as the power consumption in the vehicle OBU, being the starting point of the task offloading process. The “MECC power consumption” KPI is defined as the power consumed in the MECC by all running tasks (i.e., applications running there that relate to task offloading but also to other MECC-hosted tasks). The average value and corresponding standard deviation are calculated on around 900 samples. The metrics were collected in micro Joules and converted to Watts. Interestingly, our OBU consumption results suggest that the *trusted* offloading operations can be supported (from an energy standpoint) also by a handheld device. Even if video offloading from a mobile phone seems non-practical, assuming a modern smartphone’s battery of 2400 mAh/3.8 V [EB-], its capacity amounts to 32,8 KJ which corresponds to around 9 Joules in the unit of time. That seems sufficient to address the CONNECT trusted offloading power needs.

5.6.1.5 Data monitoring

Finally, this subsection presents the results related to the data monitoring functionality. This feature is designed to ensure that the vehicle-to-MECC connection can also serve as a channel for sharing both vehicle motion data and information perceived by the vehicle’s onboard sensors. Such data can be crucial for post-event analysis, particularly in the case of incidents or accidents, where it becomes essential to reconstruct the sequence of events leading up to the crash.

To enable this, a log file mechanism has been implemented to store the CAM and CPM information generated over the past 60 seconds. This log is then encapsulated in a CAM and CPM History Package (CCHP), which can be transmitted to the MECC over the cellular network. CCHPs can be sent either periodically or based on specific events.

In the periodic mode, as illustrated in Fig. 5.18, CCHPs are generated and transmitted at fixed intervals (e.g., every 60 seconds), each containing a complete snapshot of the CAM and CPM data collected during the previous 60 seconds. This allows continuous monitoring of the vehicle’s behaviour and environment.

```
[2025-07-10 10:44:15] {
  "timestamp": 1752137055.7169352,
  "trigger_type": "scheduled",
  "data_count": 60,
  "vehicle_data": [
    {
      "timestamp_monotonic": 1551256.958,
      "timestamp_realtime": 1752136996665.978,
      "lat": 45.009144,
      "lon": 7.563705,
      "speed": 11.85,
      "heading": 184.7,
      "acceleration": 0.0,
      "yawRate": 0.18,
      "received_at_realtime": 1752136996.6665494
    },
    {
      "timestamp_monotonic": 1552257.033,
      "timestamp_realtime": 1752136997666.027,
      "lat": 45.009038,
      "lon": 7.563692,
      "speed": 11.89,
      "heading": 184.9,
      "acceleration": 0.3,
      "yawRate": 0.5,
      "received_at_realtime": 1752136997.6666021
    },
    {
      "timestamp_monotonic": 1553257.267,
      "timestamp_realtime": 1752136998666.26,
      "lat": 45.008929,
      "lon": 7.563682,
      "speed": 12.44,
      "heading": 185.4,
      "acceleration": 0.8,
      "yawRate": 0.65,
      "received_at_realtime": 1752136998.6667364
    },
    {
      "timestamp_monotonic": 1554256.963,
      "timestamp_realtime": 1752136999665.955,
      "lat": 45.008813,
      "lon": 7.563662,
      "speed": 13.09,
      "heading": 187.5,
      "acceleration": 0.3,
      "yawRate": 3.44,
      "received_at_realtime": 1752136999.6665626
    },
```


(a)


```
}[2025-07-10 10:45:15] {
  "timestamp": 1752137115.8742185,
  "trigger_type": "scheduled",
  "data_count": 60,
  "vehicle_data": [
    {
      "timestamp_monotonic": 1611260.679,
      "timestamp_realtime": 1752137056669.672,
      "lat": 45.010019,
      "lon": 7.557032,
      "speed": 15.29,
      "heading": 32.9,
      "acceleration": 0.0,
      "yawRate": 2.59,
      "received_at_realtime": 1752137056.6702278
    },
    {
      "timestamp_monotonic": 1612256.938,
      "timestamp_realtime": 1752137057665.958,
      "lat": 45.010133,
      "lon": 7.55714,
      "speed": 15.27,
      "heading": 35.5,
      "acceleration": 0.0,
      "yawRate": 1.91,
      "received_at_realtime": 1752137057.6664639
    },
    {
      "timestamp_monotonic": 1613256.99,
      "timestamp_realtime": 1752137058666.01,
      "lat": 45.010243,
      "lon": 7.557255,
      "speed": 15.09,
      "heading": 37.0,
      "acceleration": -0.2,
      "yawRate": 1.31,
      "received_at_realtime": 1752137058.6665227
    },
    {
      "timestamp_monotonic": 1614257.02,
      "timestamp_realtime": 1752137059666.013,
      "lat": 45.010349,
      "lon": 7.557369,
      "speed": 14.9,
      "heading": 38.5,
      "acceleration": -0.1,
      "yawRate": 1.33,
      "received_at_realtime": 1752137059.6666439
    },
```


(b)

Figure 5.18: Data monitoring results. Two instances of a periodic transmission of the 60 s data window.

Alternatively, CCHPs can be transmitted in an event-triggered manner. In this case, data is sent only when specific predefined conditions are met, such as the detection of an accident, sudden deceleration, or the vehicle entering a high-risk zone. Fig. 5.19 shows an example of three CCHPs transmitted based on such event triggers.

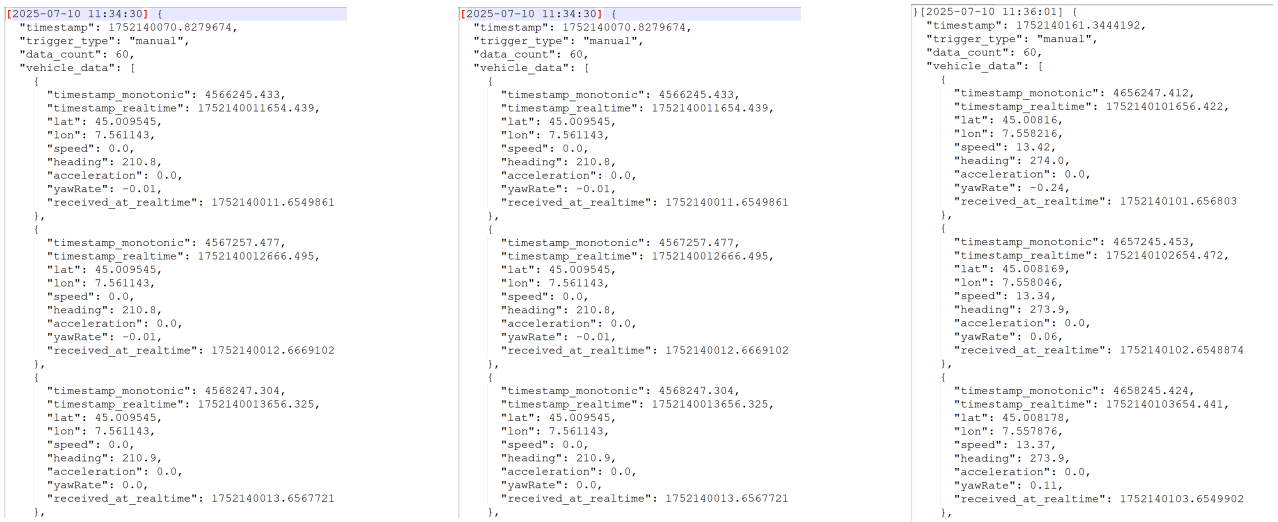
This dual-mode approach ensures flexibility and scalability: the periodic transmission provides a constant data stream for routine monitoring, while the event-triggered transmission enables efficient use of bandwidth and computational resources in safety-critical situations.

5.6.2 Results from simulated vehicle instances

In this section, we present and discuss the results of the complete pipeline process using the simulated vehicle instances.

It is important to note that the results presented here serve as a comparison with those shown in Section 5.6.1, where the same experiments are conducted using real vehicles in a field test environment.

All results presented in this use case were produced using the TAF version 0.5.0.



[2025-07-10 11:34:30] {
"timestamp": 1752140070.8279674,
"trigger_type": "manual",
"data_count": 60,
"vehicle_data": [
{
"timestamp_monotonic": 4566245.433,
"timestamp_realtime": 1752140011654.439,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.8,
"acceleration": 0.0,
"yawRate": -0.01,
"received_at_realtime": 1752140011.6549861
},
{
"timestamp_monotonic": 4567257.477,
"timestamp_realtime": 1752140012666.495,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.8,
"acceleration": 0.0,
"yawRate": -0.01,
"received_at_realtime": 1752140012.6669102
},
{
"timestamp_monotonic": 4568247.304,
"timestamp_realtime": 1752140013656.325,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.9,
"acceleration": 0.0,
"yawRate": 0.0,
"received_at_realtime": 1752140013.6567721
},

(a)

[2025-07-10 11:34:30] {
"timestamp": 1752140070.8279674,
"trigger_type": "manual",
"data_count": 60,
"vehicle_data": [
{
"timestamp_monotonic": 4566245.433,
"timestamp_realtime": 1752140011654.439,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.8,
"acceleration": 0.0,
"yawRate": -0.01,
"received_at_realtime": 1752140011.6549861
},
{
"timestamp_monotonic": 4567257.477,
"timestamp_realtime": 1752140012666.495,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.8,
"acceleration": 0.0,
"yawRate": -0.01,
"received_at_realtime": 1752140012.6669102
},
{
"timestamp_monotonic": 4568247.304,
"timestamp_realtime": 1752140013656.325,
"lat": 45.009545,
"lon": 7.561143,
"speed": 0.0,
"heading": 210.9,
"acceleration": 0.0,
"yawRate": 0.0,
"received_at_realtime": 1752140013.6567721
},

(b)

}[2025-07-10 11:36:01] {
"timestamp": 1752140161.3444192,
"trigger_type": "manual",
"data_count": 60,
"vehicle_data": [
{
"timestamp_monotonic": 4656247.412,
"timestamp_realtime": 1752140101656.422,
"lat": 45.00816,
"lon": 7.558216,
"speed": 13.42,
"heading": 274.0,
"acceleration": 0.0,
"yawRate": -0.24,
"received_at_realtime": 1752140101.656803
},
{
"timestamp_monotonic": 4657245.453,
"timestamp_realtime": 1752140102654.472,
"lat": 45.008169,
"lon": 7.558046,
"speed": 13.34,
"heading": 273.9,
"acceleration": 0.0,
"yawRate": 0.06,
"received_at_realtime": 1752140102.6548874
},
{
"timestamp_monotonic": 4658245.424,
"timestamp_realtime": 1752140103654.441,
"lat": 45.008178,
"lon": 7.557876,
"speed": 13.37,
"heading": 273.9,
"acceleration": 0.0,
"yawRate": 0.11,
"received_at_realtime": 1752140103.6549902
},

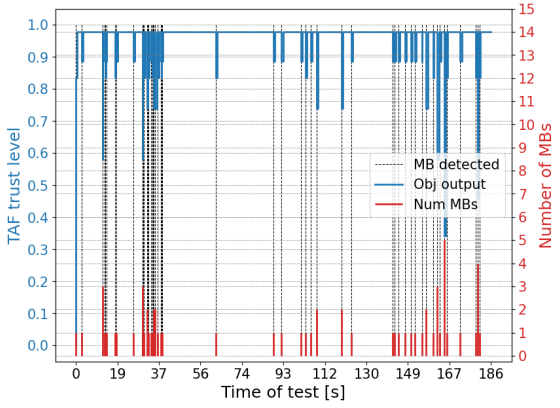
(c)

Figure 5.19: Data monitoring results. Three instances of a manually-triggered transmission of the 60 s data window.

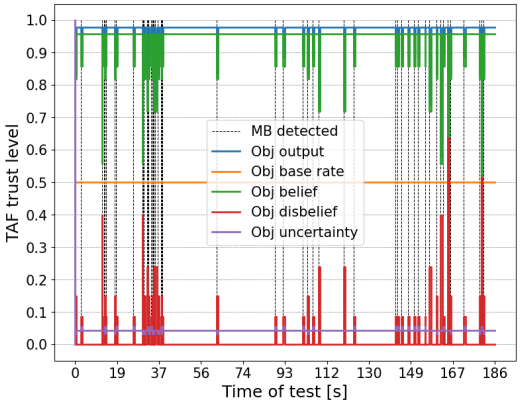
5.6.2.1 Without TCs

First, we evaluate the Actual Trust Level (ATL) results in the absence of Trustworthiness Claims (TCs) transmitted by the Trustworthiness Claim Handler (TCH). Specifically, we examine the three scenarios defined in Section 5.2.2: the baseline scenario with no manually introduced misbehaviours (MBs), a scenario where MBs are present in every CPM, and a scenario where MBs are introduced in one third of the CPMs.

Figure 5.20 shows the ATL evolution for the detected object in the first scenario. It is important to note that, although no MBs were manually introduced in this trace, intrinsic MBs are still detected by the MBD. Indeed, in Figure 5.20a, each MB occurrence during the recorded trace is marked with a dashed vertical black line. At the bottom of the same figure, the red lines indicate the number of MBs detected by the



(a) ATL of the perceived object.



(b) TAF parameters.

Figure 5.20: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with no introduced MBs. (a) ATL of the perceived object. (b) TAF parameters.

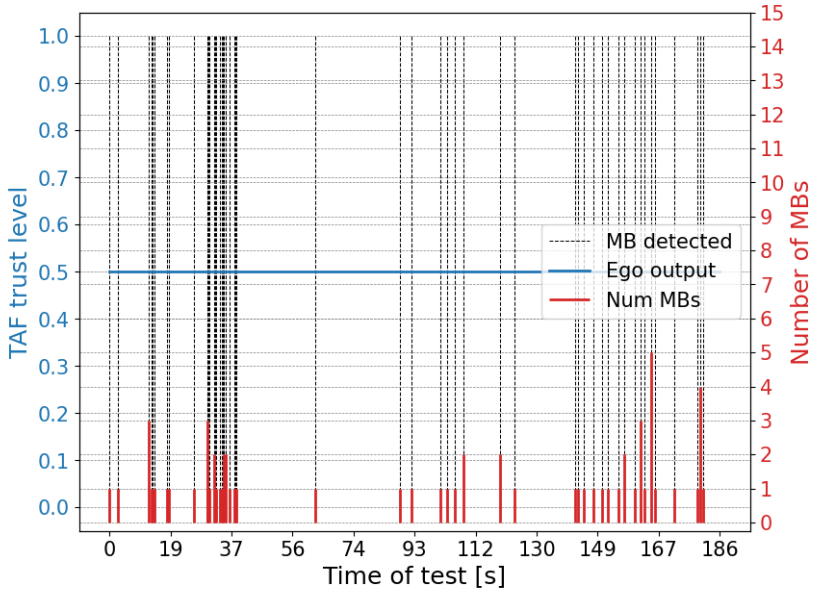


Figure 5.21: Evolution of the Actual Trust Level (ATL) of the ego-vehicle in a simulated scenario with no introduced MBs.

MBD at each time step, showing that up to five MBs can be associated with a single CPM. The blue line represents the ATL over time. As expected, when MBs are detected by the MBD, the ATL drops from a fully trusted value (close to 100%) to a lower level. The greater the number of MBs detected in the same CPM, the more significant the drop in the corresponding ATL.

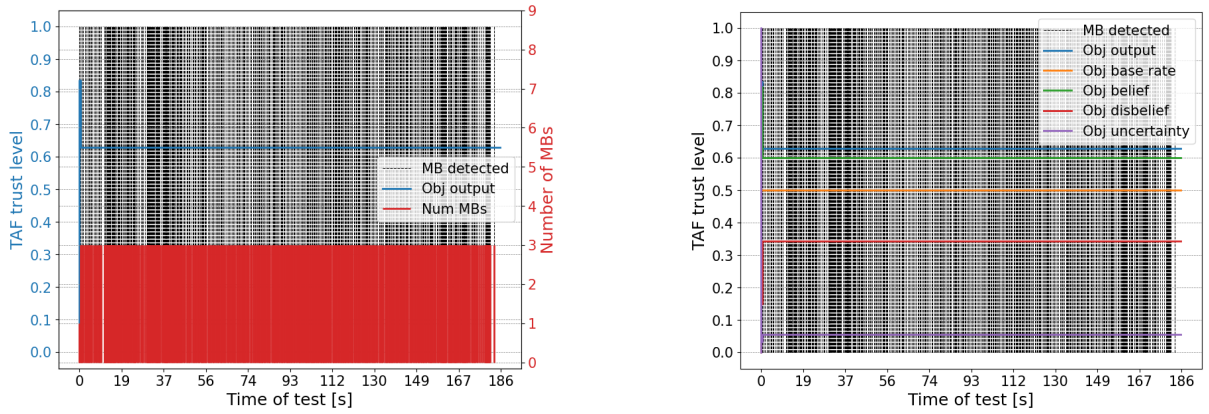
Figure 5.20b illustrates the evolution of the ATL trust opinion for the same scenario. As in the ATL plot, the presence of MBs is highlighted by dashed vertical black lines, and the blue line represents the ATL. The other lines represent the TAF's internal belief state: base rate (orange), belief (green), disbelief (red), and uncertainty (purple).

From this experiment, the following observations can be made:

1. Misbehaviours are detected even in genuine traces. The root causes of these intrinsic MBs was investigated in the field tests described in Section 5.6.1;

2. The ATL evolves as expected: it reaches the maximum value when no MBs are detected for the perceived object ahead. Conversely, when MBs are identified, they are propagated to the TAF, causing a decrease in the ATL. The more MBs are associated with a single CPM, the lower the ATL. It is worth noting that after an MB is detected, the ATL immediately returns to the maximum value upon the first subsequent valid perception. This behaviour is due to the absence of correlation between consecutive trust opinions in the SMTD use case: this is due to the SMTD trust model that does not take into consideration the temporal evolution of trust opinions. Therefore, a trust opinion is calculated from scratch when a new event arrives without taking into consideration past events.

Finally, Figure 5.21 presents the ATL evolution for the ego-vehicle, which is responsible for generating the perceptions through its on-board sensors. In this set of experiments, the Trustworthiness Claim Handler (TCH) was not activated. As a result, the TAF does not receive any Trustworthiness Claims (TCs) on which to base a trust assessment of the ego-vehicle. Consequently, the ATL of the ego-vehicle remains constant and equal to the "full uncertainty" trust opinion, yielding a projected probability of 50%. This behaviour is consistent across all three scenarios presented in this subsection. However, the impact of enabling the



(a) ATL of the perceived object.

(b) TAF parameters.

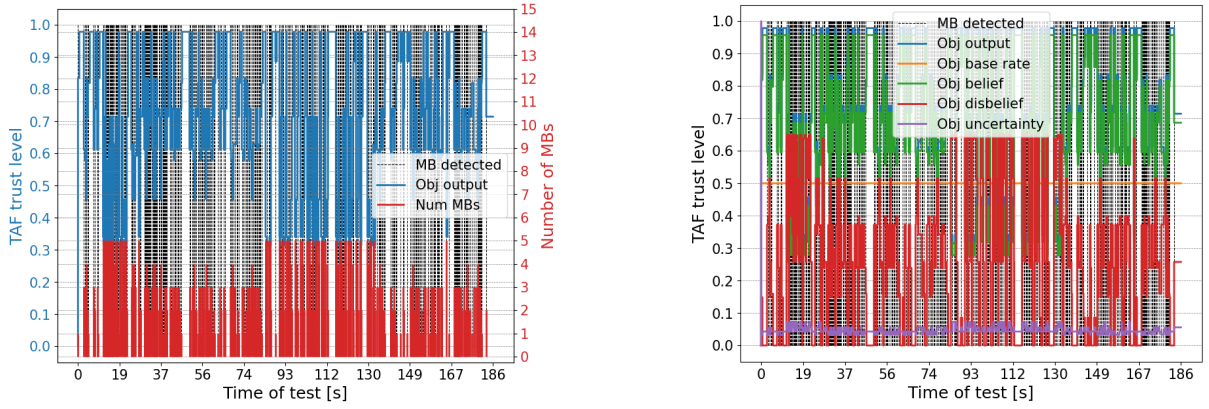
Figure 5.22: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in every CPM. (a) ATL of the perceived object. (b) TAF parameters.

TCH and the availability of TCs on the trust evaluation of the ego-vehicle will be discussed in detail in Section 5.6.2.2.

Figure 5.22 illustrates the evolution of the ATL associated with the perceived object in a scenario where a MB is manually introduced in every CPM. As in the previous scenario, Figure 5.22a focuses on the ATL progression over time, while Figure 5.22b presents the evolution of all the relevant TAF parameters. As shown in both figures, the dashed vertical black lines (i.e., the detection of MBs) are consistently present throughout the entire duration of the test, confirming that a MB was indeed injected in every CPM.

From the plots, the following conclusions can be drawn:

1. The MB introduced corresponds to the one described in Section 5.2.2. However, due to the internal mechanisms of the MBD, each altered CPM results in the detection of three distinct MB types;



(a) ATL of the perceived object.

(b) TAF parameters.

Figure 5.23: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in one third of CPMs. (a) ATL of the perceived object. (b) TAF parameters.

2. Since the same types of MBs are reported in every CPM and the TAF treats trust assessments as independent, the resulting ATL remains constant at 64%.

As in the previous scenario, the TCH is not activated, and thus the ATL for the ego-vehicle remains equal to the TAF base rate of 50%, as shown in Figure 5.21.

Finally, Fig. 5.23 presents the results for the simulated scenario in which MBs were randomly introduced in one third of the CPMs. As in the previous cases, Fig. 5.23a shows the evolution of the ATL associated with the perceived object, while Fig. 5.23b reports the corresponding TAF parameters. As observable in the figures, the dashed vertical black lines, indicating the presence of detected MBs, appear sporadically, in accordance with the randomized injection of MBs.

From the plots, the following conclusions can be drawn:

1. As in the previous scenarios, the MB injected is the one described in Section 5.2.2, and its characteristics remain consistent throughout the trace. However, the MBD incorporates historical information from past messages when performing its checks. As a result and differently from the previous scenario, here the number of MBs detected varies, with values ranging from 0 to 5 MBs for a single message;

2. These variations in the number of MBs lead to corresponding fluctuations in the ATL evolution. Specifically, the trust level computed by the TAF for the object perceived by the on-board sensors ranges from a high of approximately 98% to a low of 34%, reflecting the varying degrees of trustworthiness inferred from the input data.

As in the previous scenarios, the TCH is not active, and no TCs are transmitted. Consequently, the TAF is unable to generate a trust opinion for the ego-vehicle, which remains fixed at the base rate of 50%.

5.6.2.2 With TCs

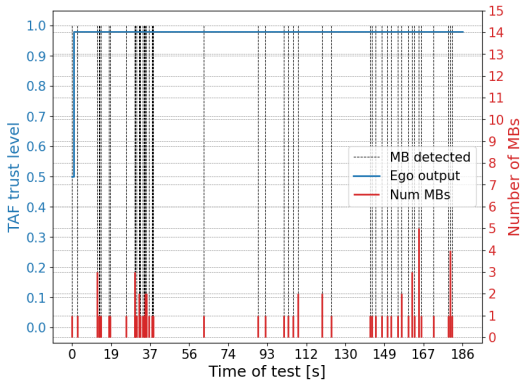
In this subsection, we repeat the three experiments from the previous subsection, this time with the Trustworthiness Claim Handler (TCH) enabled. The TCH is responsible for periodically sending Trustworthiness Claims (TCs) from the vehicle's On-Board Unit (OBU) to the TAF running on the MECC, as described in Section 5.2.6. For the scope of this analysis, TCs are transmitted at a fixed frequency of 1 s. The impact of different TC transmission frequencies on end-to-end latency was evaluated during the field tests presented in Section 5.6.1.

Figure 5.24 presents the ATL evolution in a scenario where no MBs were manually introduced. With the TCH active, the TAF can now compute trust opinions not only on the perceived object but also on the ego-vehicle itself. Specifically, i) Fig. 5.24a shows the ATL evolution related to the ego-vehicle; ii) Fig. 5.24c illustrates the corresponding TAF parameters for ATL of the ego-vehicle; iii) Fig. 5.24b reports the ATL evolution of the perceived object; iv) Fig. 5.24d depicts the TAF parameters associated to the ATL of the perceived object.

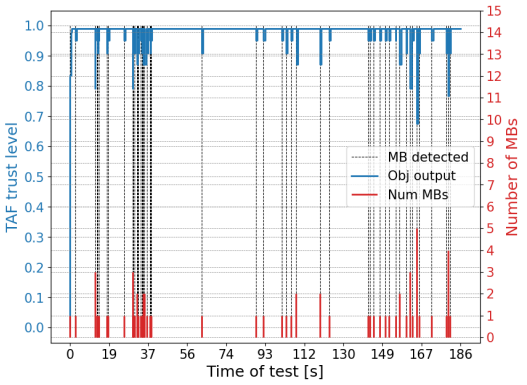
From Fig. 5.24, the following key observations can be drawn:

1. Once the first TC is received and processed by the TAF (i.e., within the first second of the experiment), the ATL associated with the ego-vehicle immediately rises to its maximum value and remains stable for the duration of the scenario, as TCs continue to be sent every second;

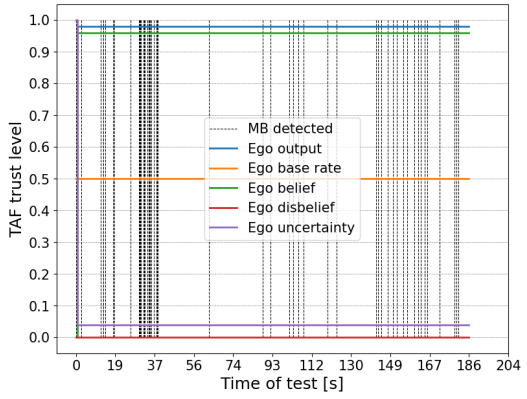
2. The ATL associated with the perceived object confirms that even in the absence of manually injected MBs, intrinsic MBs may still be detected. These result in temporary ATL drops. However, since the TAF evaluates each CPM independently, these drops do not propagate over time. As a result, the ATL quickly returns to maximum upon processing the next CPM without MBs, and the observed ATL fluctuations are minimal;



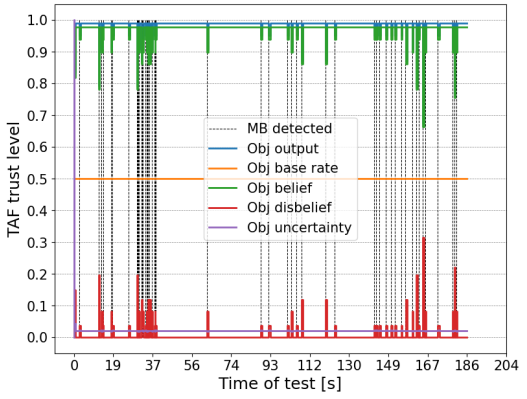
(a) ATL of the ego-vehicle.



(b) ATL of the perceived object.



(c) TAF parameters of the ego-vehicle.



(d) TAF parameters of the perceived object.

Figure 5.24: Evolution of the Actual Trust Level (ATL) in a simulated scenario with no MBs introduced. (a) ATL of the ego-vehicle. (b) ATL of the perceived object. (c) TAF parameters of the ego-vehicle. (d) TAF parameters of the perceived object.

3. The trust opinion of the ego-vehicle positively influences the ATL of the perceived object. This becomes evident when comparing the results of this scenario (with TCH active) to those of the previous subsection (Fig. 5.20a, without TCH). While both experiments use the same trace and thus the same MBs appear at the same timestamps, the ATL drops are more pronounced in the absence of TCs. When TCH is active, the ATL associated with the perceived object is higher, as it benefits from being reported by a trusted source;

4. Interestingly, while the trust opinion of the ego-vehicle influences the ATL of the perceived object, the opposite is not true. This asymmetry is by design: in this use case, the TAF processes the trustworthiness of the ego-vehicle and the perceived object as independent entities.

Figure 5.25 illustrates the evolution of the ATL in the second scenario, where MBs are manually introduced in every CPM, and the TCH is activated. As in the previous case: i) Fig. 5.25a presents the ATL evolution for the ego-vehicle; ii) Fig. 5.25c reports the TAF parameters corresponding to the ego-vehicle; iii) Fig. 5.25b shows the ATL trend for the perceived object; iv) while the corresponding TAF parameters are displayed in Fig. 5.25d.

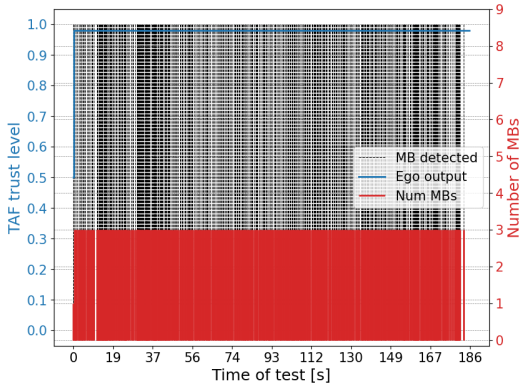
From Fig. 5.25, the following observations can be made:

1. As soon as the first TC is received and processed by the TAF, the ATL associated with the ego-

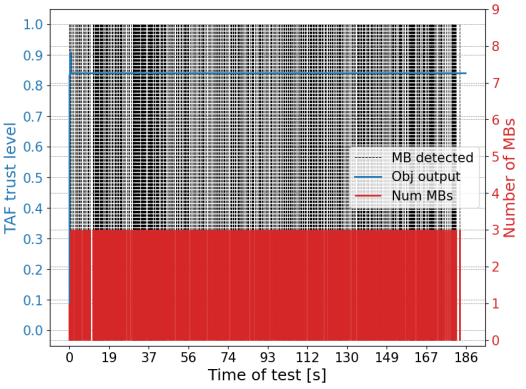
vehicle immediately reaches the maximum value and remains constant for the entire duration of the experiment. This behaviour reflects the intrinsic nature of TCs, which (when periodically received) continuously reinforce a positive trust opinion on the ego-vehicle;

2. The ATL associated with the perceived object stabilizes at 84% throughout the test. This is due to the fact that the same type of MB is introduced in every CPM and, as per the current implementation, each MB is treated as an independent event by the TAF;
3. Compared to the corresponding scenario without TCs (see Fig. 5.22a), the ATL of the perceived object increases significantly: from a constant 64% without TCH to 84% with TCH enabled. This improvement highlights the influence of TCs: the positive trust opinion formed on the ego-vehicle by the TAF indirectly contributes to a higher trust level for the perceptions generated by that vehicle;
4. Conversely, due to the independence of the trust assessments in this use case, the ATL associated with the perceived object does not influence the ATL of the ego-vehicle, which remains unaffected by any drops or fluctuations in object-level trust.

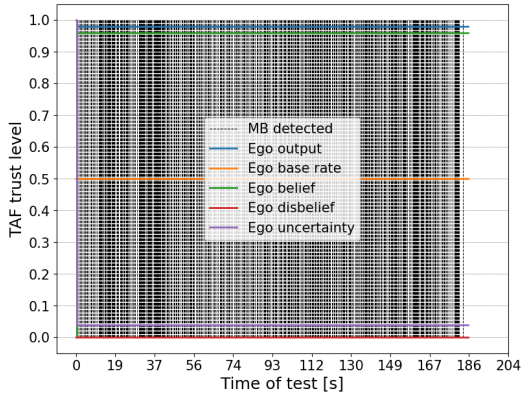
Figure 5.26 reports the evolution of the ATL in the third scenario, where MBs are manually introduced in approximately one third of the CPMs and the TCH is enabled. Specifically: i) Fig. 5.26a shows the ATL



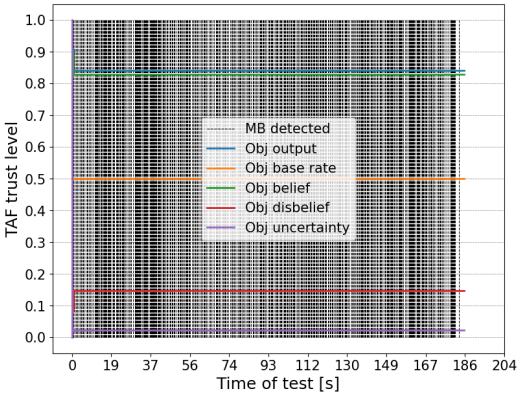
(a) ATL of the ego-vehicle.



(b) ATL of the perceived object.

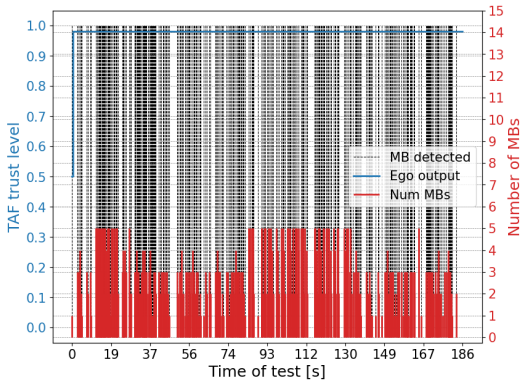


(c) TAF parameters of the ego-vehicle.

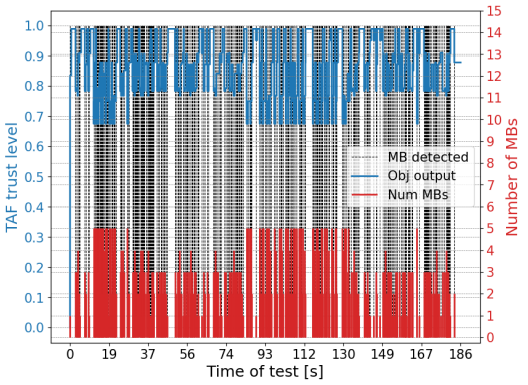


(d) TAF parameters of the perceived object.

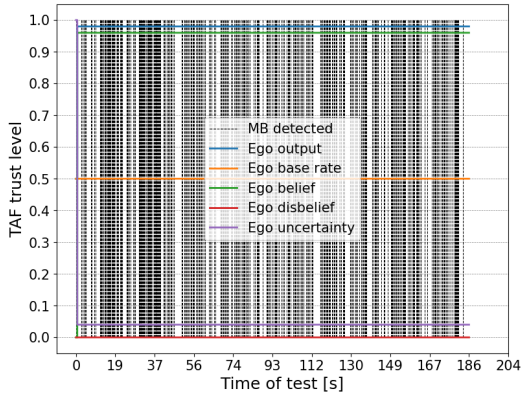
Figure 5.25: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on every CPM. (a) ATL of the ego-vehicle. (b) ATL of the perceived object. (c) TAF parameters of the ego-vehicle. (d) TAF parameters of the perceived object.



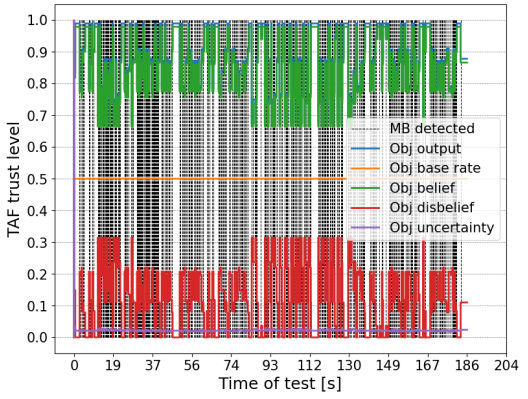
(a) ATL of the ego-vehicle.



(b) ATL of the perceived object.



(c) TAF parameters of the ego-vehicle.



(d) TAF parameters of the perceived object.

Figure 5.26: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on one third of the CPMs. (a) ATL of the ego-vehicle. (b) ATL of the perceived object. (c) TAF parameters of the ego-vehicle. (d) TAF parameters of the perceived object.

evolution of the ego-vehicle; ii) Fig. 5.26c reports the TAF parameters corresponding to the ego-vehicle; iii) Fig. 5.26b illustrates the ATL trend for the perceived object; iv) Fig. 5.26d displays its related TAF parameters.

From the results in Fig. 5.26, the following conclusions can be drawn:

1. The ATL of the ego-vehicle quickly reaches its maximum value as soon as the first TC is received by the TAF and remains stable throughout the test due to the periodic transmission of TCs at a frequency of 1 s. This confirms the correct functioning of the TCH in building and maintaining a trust opinion on the ego-vehicle;

2. The ATL associated with the perceived object shows noticeable fluctuations that are directly linked to the presence or absence of MBs in the CPMs. As expected, when MBs are detected, the ATL drops; when no MBs are present, it recovers. These ATL variations range between approximately 98% and 67%, indicating a less severe degradation compared to the same scenario without TCH (Fig. 5.23a), where ATL values could drop as low as 34%;

3. The mitigation effect of TCH is again evident: although the same MBs are introduced at the same time in both scenarios, the ATL values for the perceived object are consistently higher in this case,

thanks to the positive trust opinion on the ego-vehicle generated by the TCs;

4. As observed in previous scenarios, the trust opinion built for the ego-vehicle does not degrade even when MBs are detected in its perceptions, due to the independent handling of each trust event by the TAF.

Overall, the results presented in this section confirm the correct behaviour of the ATL computation mechanism, both in the presence and absence of TCs. The KPIs associated with SMTD.US.4 have therefore been successfully validated in this simulated environment.

5.7 Discussion & Critique - Lessons Learnt

This section concludes the chapter on the SMTD use case by summarizing the key findings and insights gained throughout its development and evaluation.

The first and most fundamental achievement of this use case was the successful setup and demonstration of a V2X scenario involving real-time message generation and transmission by a physical vehicle operating in a living lab environment. This alone presented a significant technical challenge. Beyond this, V2X messages collected during field tests were processed through the Misbehaviour Detector (MBD) and the Trustworthiness Assessment Framework (TAF), yielding meaningful results. The observed Trust Level (TL) evolution was consistent and appropriate given the context and dynamics of the living lab.

It is important to emphasize that the SMTD use case within CONNECT was validated in the Stellantis test track, a controlled setting with limited unpredictability. This distinction highlights a clear difference between a living lab and an open-road scenario, where real traffic introduces a higher degree of variability and complexity. Therefore, a natural next step beyond CONNECT would be the validation of KPIs and functional behaviour of the SMTD system in real-world conditions, such as on public roads. This would further elevate the Technology Readiness Level (TRL), aligning it more closely with automotive industry standards.

Another notable outcome of the SMTD use case was the evaluation of the Trustworthiness Claim (TC) mechanism. It was demonstrated that even at a high transmission frequency (e.g., every 100 ms), the continuous dispatch of TCs had no significant impact on the end-to-end latency of the overall process. This is a valuable result, as it confirms that the Trustworthiness Claim Handler (TCH) can be considered for standardization as an additional V2X service designed to enhance trust and security in V2X communications without compromising performance.

An important insight from the test, in the test track was the critical dependency of TL evolution on the quality of checks performed by the MBD. Specifically, it was observed that the TL of perceived objects frequently dropped whenever the leading vehicle executed a turn. Upon further investigation, it was determined that the triggering of Misbehaviour Reports (MRs) in these cases was caused by the Kalman filter applied to the perceived object's trajectory. This filter produced inaccurate results during tight curves, leading to false positives. This finding highlights the significant influence of the MBD on the overall trust assessment process and underlines the need for careful design and validation of MBD checks. These checks must be robust and meaningful in real-world conditions, where traffic behaviours are diverse and often unpredictable. Reducing or eliminating false positives is essential to ensure system reliability and driver confidence.

Finally, a key element validated across the entire CONNECT project is the role of the TAF. In light of recent ETSI developments, where collective perception through sensor data sharing is a cornerstone of upcoming V2X Day 2 applications, it is increasingly unrealistic to assume that all shared information can be fully trusted. While extreme cases involving deliberate attacks may be easier to detect and are typically addressed through traditional security measures (e.g., authentication and integrity checks), a more subtle

and prevalent challenge arises from inaccurate, poorly calibrated, or malfunctioning sensors, especially under adverse environmental conditions. In such scenarios, a mechanism like the TAF becomes essential. It provides an subjective, dynamic trust evaluation of received perceptions, enabling receiving entities to make informed decisions, such as whether to ignore unreliable data or use it in safety-critical applications like collision avoidance.

Chapter 6

Ethical Analysis of Trust and Trustworthiness in CCAM

6.1 Trust and Trustworthiness

Trust and trustworthiness are vital in cooperative, connected, and automated mobility (CCAM), as they ensure seamless collaboration among technologies, stakeholders, and users and foster safety, reliability, and societal acceptance of intelligent transportation systems. However, as discussed in D1.3 and D2.2, trust and trustworthiness do not always coincide. CCAM systems may be trustworthy without being trusted. Likewise, users may trust CCAM systems without these systems being worthy of that trust. Trustworthiness without trust can cause social problems, under-use, or even active resistance. Trust without evidence of trustworthiness can cause physical harm, breach of privacy, or exploitation of private individuals. While it is not possible to ensure trust, it is possible to manage and promote the trustworthiness of CCAM systems.

The trustworthiness of CCAM systems does not solely depend on the reliability of the technologies. It also depends on the credibility of manufacturers, the effectiveness of regulatory frameworks, and the robustness of the governance mechanisms surrounding technical systems. In other words, the trustworthiness of CCAM systems is rooted in the broad network of human decisions, institutional oversights, and societal norms and values that guide their development and use. In the following paragraphs, we summarize these approaches derived from the work performed in the context of D1.3 and D2.2.

6.2 Assessment List for Trustworthy CCAM

In 2019, the European Commission's High-Level Expert Group on Artificial Intelligence (AI HLEG) released their Ethics Guidelines for Trustworthy Artificial Intelligence (AI). This provides a framework for the ethical development and use of AI systems. The guidelines propose a set of 7 key requirements that AI systems must meet to be considered trustworthy. These requirements include:

1. human agency and oversight,
2. technical robustness and safety,
3. privacy and data governance,
4. transparency,
5. diversity, non-discrimination and fairness,

6. societal and environmental wellbeing, and
7. accountability.

CONNECT took these requirements, originally set out by the AI HLEG, and calibrated and adapted them in the context of trust management for CCAM systems; see D2.2. In this context, the seven requirements can be defined as follows.

Human agency and oversight: CCAM systems should empower human beings, allowing them to make informed decisions and fostering their fundamental rights. At the same time, proper oversight mechanisms need to be ensured to decide when and how to use the Automated Information Handling (AIH) system. This can include the decision not to use an AIH system in a particular situation.

Investigating the importance of human agency and oversight for building trustworthy CCAM helps us answer questions such as: could the AIH system affect human autonomy by generating over-reliance by end-users? Could the AIH system affect human autonomy by interfering with the end-user's decision-making process in any other unintended and undesirable way? Have the humans (human-in-the-loop, human-on-the-loop, human-in-command) been given specific training on how to exercise oversight? Could the AIH system generate confusion for some or all end-users or subjects on whether they are interacting with a human or AIH system? See D2.2.

Technical Robustness and safety: CCAM systems should rely on technical robustness to both ensure and assure users of their reliability and safety. The systems need to be secure and resilient. Resilient systems are able to recover quickly from challenges. CCAM systems need to be safe, ensuring a fall back plan in case something goes wrong, as well as being accurate, reliable, and reproducible. That is to ensure that unintentional harm can be minimized or completely prevented.

Exploring the role of technical robustness and safety for building trustworthy CCAM helps us answer questions such as: were possible threats to the AIH system (design faults, technical faults, environmental threats) and their possible consequences identified? Were measures put in place to ensure the integrity, robustness, and overall security of the AIH system against potential attacks over its life-cycle? Could the AIH system cause critical, adversarial, or damaging consequences (e.g. pertaining to human safety or overall trust in CCAM systems) in case of low reliability and/or reproducibility? See D2.2.

Privacy and data governance: Prevention of harm to privacy necessitates adequate data governance that covers the quality and confidentiality of the data used, its relevance in light of the domain in which the AIH systems will be deployed, its access protocols, and the capability to process data in a manner that protects privacy. Protection of privacy entails protection of personal data, secure communication with other vehicles, and resilience against cyber-attacks; see D2.2.

Examining the significance of privacy and data governance in the development of trustworthy CCAM allows us to explore questions such as: depending on the use case, are specific mechanisms established that allow flagging issues related to privacy concerning the CCAM system? Were AIH systems trained and developed by using or processing personal data (including special categories of personal data)? Were the AIH system aligned with relevant standards (e.g., ISO25, IEEE26) or widely adopted protocols for data management and governance? See D2.2.

Transparency: A crucial component of achieving Trustworthy CCAM is transparency which encompasses three elements: 1) traceability, 2) explainability and 3) open communication about the limitations of the AIH system. AIH systems and their decisions should be explained in a manner adapted to the stakeholder concerned. Humans also need to be aware that they are interacting with an AI system and must be informed of the system's capabilities and limitations.

Investigating how transparency, in terms of traceability, explainability, and open communication, contribute to the trustworthiness of CCAM enables us to address questions such as: are adequate

logging practices put in place to record the decision(s) or recommendation(s) of the AIH system? Will downstream producers and service providers have the capacity to survey the key stakeholders (such as OEMS) to assess if they understand the decision(s) of the AIH system? Are specific mechanisms in place that will allow users to be informed about the purpose, criteria, and limitations of the decision(s) generated by the AIH system? See D2.2.

Diversity, non-discrimination and fairness: Bias must be avoided, as it could have multiple negative implications, from the marginalization of vulnerable groups to the exacerbation of prejudice and discrimination. Fostering diversity, CCAM systems should be accessible to all, regardless of any disability, and involve relevant stakeholders throughout their entire life circle.

Exploring the role of diversity, non-discrimination, and fairness for building trustworthy CCAM helps us answer questions such as: are strategies or a set of procedures established to avoid creating or reinforcing unfair bias in the AIH/CCAM system, both regarding the use of input data as well as for any algorithm design? Are the CCAM systems' user interfaces usable by those with special needs or disabilities or those at risk of exclusion? Is there a mechanism to include the participation of the widest range of possible stakeholders in the CCAM systems' design and development? See D2.2.

Societal and environmental well-being: Sustainability and ecological responsibility of CCAM systems should be encouraged, and research should be fostered into AI solutions addressing areas of global concern, for instance the sustainable development goals. CCAM systems should benefit all human beings, including future generations. They should also consider the environment, including other living beings, and their social and societal impact should be carefully considered. These considerations should be taken into account in the decision making process through, for example, choosing routes that minimise emissions or the footprint of developing AIH systems.

Examining the significance of societal and environmental well-being in the development of trustworthy CCAM allows us to answer questions such as: where possible, are mechanisms to evaluate the environmental impact of the CCAM system's development, deployment and/or use (for example, the amount of energy used and carbon emissions) established? Do the CCAM systems impact human work and work arrangements? Could the CCAM system have a negative impact on society at large or democracy? See D2.2.

Accountability: Mechanisms should be put in place to ensure responsibility for the development, deployment and use of AIH systems for CCAM, and to ensure responsibility for AI systems and their outcomes. Auditability, which enables the assessment of algorithms, data and design processes plays a key role therein, especially in critical applications. Adequate and accessible redress should also be ensured.

Studying the role of accountability in the development of trustworthy CCAM helps us to respond to questions such as: are there established mechanisms that facilitate the AIH systems' auditability (e.g., traceability of the development process, the sourcing of training data, and the logging of the AIH systems' processes, outcomes, positive and negative impact)? Are there established ethics review boards or similar mechanism to discuss the overall accountability and ethics practices, including potential unclear grey areas? Does this process include identification and documentation of conflicts between the six aforementioned requirements or between different ethical principles and explanation of the 'trade-off' decisions made? See D2.2.

6.3 Survey outline

This framework has been validated through two surveys conducted on the LimeSurvey online platform. They surveys were anonymized and preceded by preliminary questions designed to obtain the informed consent of participants. The purpose of the surveys was to gather responses from experts in trust and

connected vehicles. In particular, the surveys were designed to collect the participant's assessment of how the seven key criteria are being implemented in CONNECT and connected vehicles in general. The results were expected to help us understand how trustworthiness can be operationalized, assessed, and promoted. For this validation exercise, two surveys were conducted aimed at two separate stakeholder groups who are directly or indirectly involved in, or affected by, the CCAM system throughout its life cycle.

The first stakeholders are those individuals who are involved in, or have the necessary technical expertise to understand, the core theoretical and technical features of CCAM components, processes, and systems. For instance, software engineers involved in designing the trust assessment framework, the production of outputs for actual trust levels, cybersecurity experts tasked with assessing risks of malicious intrusion etc. More specifically, in the context of the CONNECT project, the first stakeholders were experts from the consortium.

The second stakeholders are those individuals working for organisations who use the CCAM components, processes, and systems in their production of CCAM. For instance, this is OEMs (original equipment manufacturers), automotive vendors, security service providers and so on. The second set can loosely be considered users but not consumers as end users. For CONNECT this is Tier 1 CCAM researchers such as DENSO, Tier 2 researchers such as CRF, automotive vendors such as Toyota, and security service providers such as Intel. Tier 1 and Tier 2 are defined based on their position in the supply chain. A Tier 1 supplier directly supplies OEMs with components that are ready for installation into the vehicle. A Tier 2 supplier provides components to the Tier 1 suppliers and is the next level in the supply chain.

Following the method outlined in D2.2, these two stakeholder groups were chosen to test whether the questions regarding trustworthiness are applicable and understandable. Given their expertise in the development of CONNECT and CCAM more generally, the two stakeholder groups were chosen to validate the approach taken by CONNECT, and to provide expert feedback on how the 7 key requirements have been adapted to CCAM, as well as how effective CONNECT has been in approaching and understanding trustworthiness by reference to the 7 key requirements. This meets the intention of having an approach to trustworthiness that is dynamic and responsive to stakeholder feedback and advice.

For Stakeholder Set 1, the validation and evaluation steps tracked to the following questions:

- On a scale of 1 to 10 (with 1 being not relevant at all and 10 being extremely relevant), how relevant is "human agency and oversight" to your work on CONNECT?
- On a scale of 1 to 10, how relevant is "human agency and oversight" to CONNECT generally?
- Here you can explain the way in which "human agency and oversight" are relevant for your work on CONNECT or for CONNECT in general.

For Stakeholder Set 2, the validation and evaluation steps tracked to the following questions:

- On a scale of 1 to 10 (with 1 being not relevant at all and 10 being extremely relevant), how relevant is "human agency and oversight" to CONNECT?
- On a scale of 1 to 10, how relevant is "human agency and oversight" to CCAM generally?
- Here you can explain the way in which "human agency and oversight" are relevant to CONNECT or CCAM in general.

Screenshots of a page of each of these surveys are provided in the annex at the end of this document. For both surveys, these questions were repeated for all 7 key requirements, with each set of questions focusing on each one of the requirements, i.e. how relevant is technical robustness and safety, how relevant is privacy and data governance, and so on. At the end of the survey, the participants were given space to add other values and/or design requirements which they believed were important for trustworthiness of CCAM systems but were not covered by the 7 requirements.

Several reasons influenced the formulation of the survey questions. Most importantly, they were chosen to support validity and reliability of the 7 key requirements. Neutral wordings were aimed to ensure that respondents' answers reflected their actual views rather than being influenced by the phrasing of the question. The questions were also formulated using plain and unambiguous language to ensure that all participants, regardless of their technical knowledge or background, understand them consistently. The design of the questions was informed by theoretical frameworks, in this case, the 7 key requirements from the EU's AI HLEG. The questions were structured to operationalise abstract concepts like "transparency" or "accountability" into measurable items.

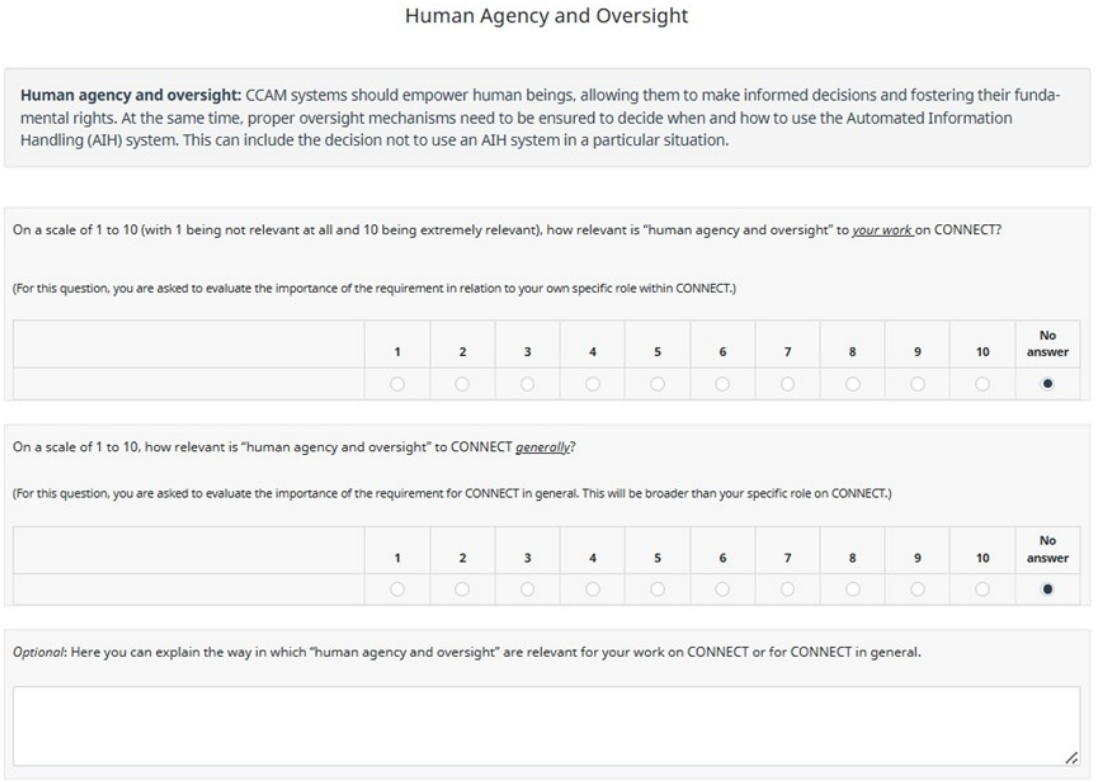


Figure 6.1: Online Questionnaire Layout

6.4 Survey results

In total, 16 experts in CCAM design, within and outside the CONNECT project, participated in the surveys. On average, survey participants rated all seven key requirements for trustworthy CCAM systems as having moderate to high degrees of importance, with average scores ranging from 5.9/10 to 8.5/10 (mean = 7.1; median = 6.9). When asked to assess the relevance of these requirements specifically to their own work within the CONNECT project, participants provided slightly lower ratings, with scores ranging from 4.9/10 to 8.1/10 (mean = 6.2; median = 6.0). The slightly lower ratings can be attributed to the fact that some of the key requirements may be less directly relevant to the survey participants in their specific areas of expertise, though the experts still recognise their importance for CCAM systems overall.

Among the 7 key requirements, "technical robustness and safety", "transparency", and "accountability" received the highest average ratings of, respectively, 8.5/10, 7.6/10, and 8.0/10. On the other hand, "human agency and oversight", "privacy and data governance", "diversity, non-discrimination and fairness", and "societal and environmental wellbeing", respectively, received the average ratings of 6.5/10, 7.2/10, 6.5/10, and 5.9/10. See Chart 1.

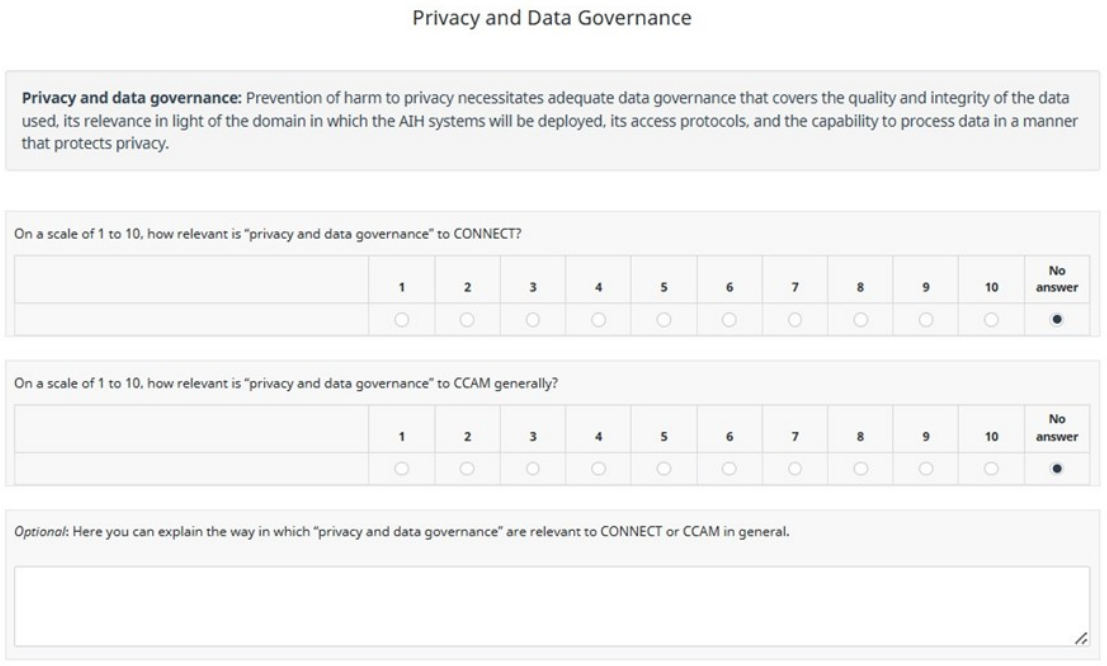


Figure 6.2: Online Questionnaire Layout 2

6.5 Discussions and Interpretations of Survey Results

Due to the limited number of participants, the survey results should be interpreted with caution. Future studies should be conducted to confirm the results' findings. However, given that the participants considered all seven key requirements moderately or highly important, the results indicate that AI HLEG requirements are overall a pertinent starting point to understand the trustworthiness of CCAM systems and break them down into its components. This validates our general approach that all of the 7 key requirements can be used for assessing the trustworthiness of CCAM systems.

These findings also suggest that in the context of CCAM, "technical robustness and safety", "transparency", and "accountability" are particularly important for promoting system trustworthiness. This is not surprising for at least two reasons – given the physical nature of CCAM, the physical risks posed to road users from CCAM, and the legacy of automotive engineering taking physical risks seriously, we would expect that technical robustness and safety would be seen as essential values to be a priority in any CCAM design. What is quite interesting is that transparency and accountability are also considered to be of significant importance. This can be explained by two related features of CONNECT. First, given that CONNECT's focus on building trustworthy CCAM is information based – the ultimate physical safety and cybersecurity of the CCAM systems are not simply dependent upon, but ensured by trustworthy production and communication of information, systems concerned with the production and communication of information – transparency of and accountability for that information - are an essential feature of CONNECT's core concept. Second, we must also bear in mind the relative 'novelty' of the CCAM approach to the connected and driverless vehicles. "Just as the introduction of technologies like genetically modified crops... were heavily impacted by public perceptions of trust, community attitudes will impact the way that [driverless] vehicles ultimately come to be integrated into our existing driving systems and larger sociotechnical infrastructures" [Hen20]. It is unsurprising that methods to both ensure and assure the trustworthiness of these new sociotechnical infrastructures would be considered highly important.

A key point to emphasise here, as argued by Henschke and Arora [HA24], is that the CCAM systems much recognise that a plural set of values are important to trustworthy CCAM. It is not just that the survey respondents saw that technical robustness, or transparency, or accountability are of high importance to trustworthy CCAM, but that each of these three requirements are highly important. "The design of complex

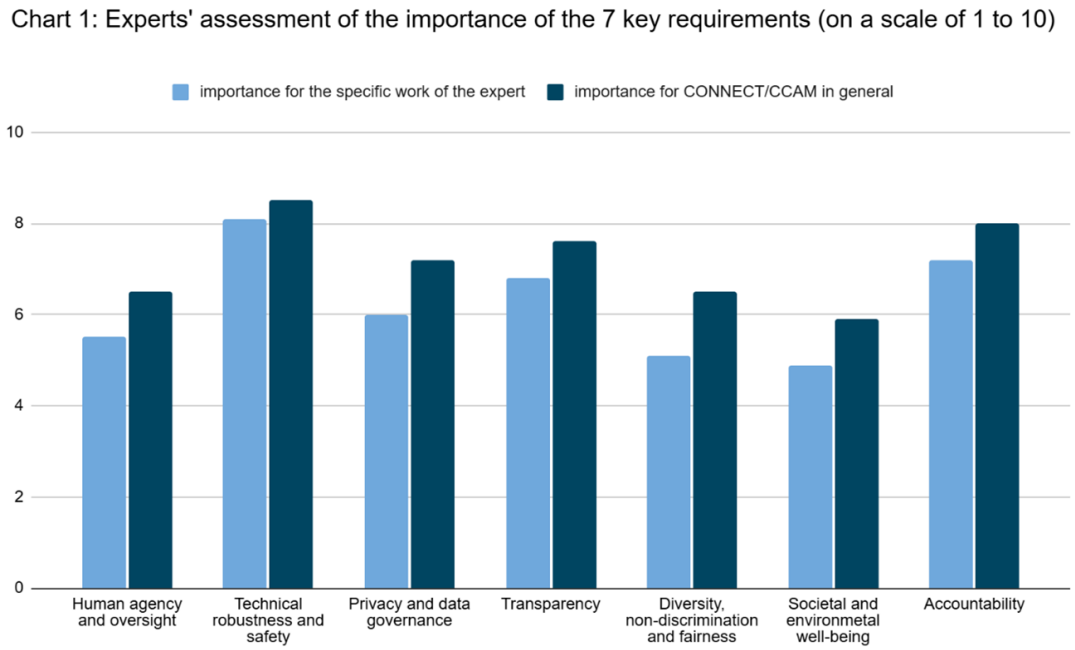


Figure 6.3: Results from Ethics-related questionnaires

socio-technical systems like [connected vehicle systems] are best approached with an understanding that such systems afford a plural set of values" [HA24]. Further to this, as all of the seven requirements were seen to be of importance, CCAM must recognise all of the 7 key requirements in their design.

The fact that the survey participants believed that technical robustness and safety are important factors for trustworthy CCAM means that a key requirement for AIH systems is their ability to operate reliably under changing conditions or adversarial interactions. To be robust and safe, these systems must be designed to prevent, withstand, and quickly recover from risks [Hen20]. AIH systems should also be resilient. Building resilience into CCAM systems is a form of managing potential failures; it is about absorbing and responding to failures by retaining the core characteristics of the system [Sol25]. This ensures security, safety, accuracy, and reliability provide clear fall-back plans and reproducible behaviour. According to one of the survey participants, while the CONNECT project relates mostly to technical robustness, safety is often addressed by the relevant processes and standards, such as Hazard Analysis and Risk Assessment (HARA), ISO 26262, and ISO 21448, also known as Safety of the Intended Functionality (SOTIF).

The high importance of transparency, as rated by the participants, indicates that the data and processes that yield the AIH systems' decisions should be properly documented to strengthen trustworthiness of CCAM. Transparency is defined in terms of traceability, explainability, and open communication about the limitations of AIH systems. The survey results support the claim that AIH driven decisions, to the extent possible, should be explained and understood to those directly and indirectly affected, in order to allow for contesting of such decisions. This tracks to arguments that digital systems more generally must ensure that their systems meet the standards and norms expected of them, and have effective communicative measures to assure key stakeholders that such standards and norms are met [RH17]. Transparency is an essential measure to ensure trustworthiness and assure users of that they can or should trust CCAM.

Accountability is another key requirement that was rated as highly important for building trust in CCAM systems by the survey participants. Accountability is a cornerstone of building trustworthy CCAM systems. It ensures that clear responsibilities are assigned across the development, deployment, and operational phases, enabling traceability of decisions and actions. Establishing robust accountability mechanisms improves public trust, facilitates regulatory compliance, and provides a framework for addressing errors,

failures, or unintended consequences in a transparent and systematic way. As Nyholm has argued, in systems like CCAM, a 'retribution gap' can arise if there is no clear accountability for failure or accidents. 'When someone is injured through somebody else's agency (whether it is a human or a robot), people generally tend to want to find some individual or individuals who can be punished for this. But if the agency in question is a robotic agency... since robots are themselves not fit to be punished and no humans are fully responsible for their robotic agency, there is a potential "retribution gap"...' [Nyh18]. This is particularly important for accountability in CCAM, as 'people will have a strong impulse to want to punish somebody, but nobody will be an ... appropriate target for punishment' [Nyh18]. As such, promoting accountability is also a way of promoting human agency and oversight. Establishing effective mechanisms for accountability is one channel through which humans exert their agency and influence the operation of technical systems [Sol25]. Accountability is a way of ensuring and assuring users that not only is the vehicle itself worthy of trust, but that there are measures in place to hold relevant individuals to account should the CCAM system fail.

In addition to the 7 key requirements, the survey participants highlighted other points that can be considered to evaluate and promote the trustworthiness of CCAM. For example, some survey participants indicated that openness to faults and problems, continuous improvement, and vendor trust can contribute to further strengthening of users' trust into the systems. Vendor trust is built on the vendor's ability to deliver reliable, secure, and transparent technologies and ensuring that CCAM systems operate as expected, safeguard user privacy, provide clear accountability, and address potential risks such as system failures or malicious activities. Similarly, some participants noted that in connected vehicles, users receive data from external sources, and one key aspect would be the trustworthiness of the data and the external sources. However, these requirements can be interpreted as the extension, rather than the expansion, of the existing seven requirements. In particular, they pertain to the three requirements that received highest importance ratings from survey participants, namely, technical robustness and safety, transparency, and accountability. Similarly, another participant noted that technical robustness might need to be extended to validated, tested robustness, meaning that apart from design requirements and usage of robust technologies, it is important to ensure that the final system has been exhaustively tested. It is the result of tests, such as the Euro NCAP (European New Car Assessment Programme) crash tests, that essentially contribute to trustworthiness.

Finally, as highlighted by some survey responses, the relationship between trustworthiness of CCAM systems and some of the key requirements is not always unilateral. This is particularly true for social wellbeing. While social and environmental wellbeing enhances system trustworthiness, the reverse is also true. When the trustworthiness of CCAM systems is ensured, users may adopt the technology with greater peace of mind and reassurance, ultimately contributing positively to their well-being.

6.6 Conclusions

There are three conclusions to draw from the survey results: The survey is a valid approach to drawing information on trustworthiness and trust, that more research is needed, and that all seven key requirements are needed for CCAM to be trustworthy. On the first point, the responses showed that CONNECT's approach to both operationalising trustworthiness in CCAM, and to providing a useful tool to give stakeholder assessments on CCAM, and on informationally mediated trust relationships more generally. The approach developed by CONNECT, in which the HLEG 7 key requirements are adopted and adapted to CCAM is a useful template for future work looking at trustworthiness and trust.

Second, as per D2.2, this survey engaged with Stakeholder sets 1 and 2, meeting the deliverables promised by CONNECT. There is significant work to be done in expanding these assurance methods through research involving stakeholder sets 3 (experts in CCAM oversight and accountability) and 4 (wider non-expert users and community members affected by CCAM). Conducting surveys with stakeholder sets 3 and 4 is beyond the scope of CONNECT's work and require more extensive surveys, which can indicate

areas of fruitful and necessary work. Given the effectiveness of the approach developed by CONNECT, future directions for research can focus on stakeholder sets 3 and 4.

Finally, the survey responses also validate the value-pluralist approach developed by CONNECT. While it is perhaps uncontroversial to say that trustworthy CCAM involve more values than just physical safety, the survey has shown that trustworthiness is more than mere technical reliance. In order for CCAM to be worthy of the trust that people place in these systems, CCAM must include a range of values. The pioneering work by the AI HLEG to give substance to these values, to explicate the range of values, and elucidate what these values consist in, is a highly useful foundation for this value-pluralist design. As discussed, it is unsurprising that stakeholder sets 1 and 2 would consider technical robustness and safety, transparency, and accountability to be of primary importance, but we also saw clear evidence that the other requirements were also considered to be essential for trustworthy CCAM. Following from the second conclusion, future research involving stakeholder sets 3 and 4 may place greater emphasis on wider social values than the stakeholder sets 1 and 2. However, this spread of attitudes to these values validates not just the method developed here, but also the value-pluralism that underpins it.

Chapter 7

Impact Assessment

Within a CCAM ecosystem we encounter applications of mixed criticality, often operating under dynamic and highly distributed environments. The abundance of data exchanged among V2X entities enables informed decision-making, even in safety-critical CCAM applications. In this context, security measures are essential for safeguarding the safety assurances that are offered by such services. Despite the necessary security controls that may be in place - and could potentially increase the perceived trustworthiness on the communication channels - one remaining challenge lies in the assessment of the actual content of the data, going beyond the binary-centred plausibility checks that are currently investigated and enforced as part of a misbehaviour detection lifecycle. In fact, such strict detect-and-react policies are considered obsolete, especially in such dynamically evolving scenarios with potentially conflicting evidence on the trustworthiness of the data. By modelling uncertainty in trustworthiness evaluations, *CONNECT* has made significant contributions towards realizing a generic Trust Assessment Framework within the CCAM ecosystem.

The concept of assessing trust and trustworthiness constitutes an extensively studied area across many application domains. At the same time, in the context of V2X communications, research has already yielded significant results in the detection of misbehaviour patterns or the assessment of error propagation in control algorithms [ETS23b]. Building on top of these concepts, *CONNECT* is one of the pioneering initiatives that provides a unified, overarching Trust Assessment Framework that engrains trust-aware decision making into safety-critical CCAM applications. Through the use of Subjective Logic, the *CONNECT* framework is able to incorporate epistemic uncertainty and trust in information sources, making it useful in decision-making under uncertainty. This is especially useful in environments where we often have incomplete or conflicting evidence, such as when data is missing, delayed, or of varying quality.

Materializing such a framework requires the careful design of an overarching architecture that covers all aspects of a trust assessment workflow. To this end, the overall architecture includes the core trust-related functionalities - the backbone of the *CONNECT* Trust Assessment Framework - but also all auxiliary mechanisms that enable the evaluation of trust propositions on both the V2X entities as well as the data exchanged among them. Throughout the lifespan of the project, the design, development and testing of the overarching TAF in the three *CONNECT* use cases has yielded prototypes that exhibit the following framework characteristics towards enabling trustworthy CCAM:

Implementable: All major concepts are implemented and operational.

Openly accessible: All major components are made available open source.

Generally applicable: The *CONNECT* architecture and its Trust Assessment Methodology has proven to be applicable to a large range of scenarios, including in-vehicle as well as two different cooperative, V2X scenarios.

Effective: Throughout a number of studies, the *CONNECT* approach to trust assessment and particularly

the TAF system have shown their capability to identify and discriminate between trustworthy nodes and data and untrustworthy ones.

Efficient: In our micro-benchmarking (see Chapter 2), all major performance-related KPIs have been met - in fact they have been exceeded - showing that trust assessment introduces no unacceptable system load or delay.

All these achievements position *CONNECT* as a core initiative that may act as a stepping stone for future research towards continuous and dynamic trust assessments, considering environments where trust sources and trust relationships constantly evolve. In fact, this constitutes a timely endeavour, as the need for establishing a generic and harmonized trust assessment methodology has already been acknowledged by the community in an effort to attain the goal of trustworthy CCAM. At the same time, *CONNECT* serves as a foundational research, shedding light to unresolved challenges that are still pending to be tackled in order to facilitate the vision of an overarching trust assessment framework. As further illustrated in this chapter, there are still key challenges and insights that need to be taken into consideration. First, an important open challenge relates to the convergence of trust assessment practices between both security and safety aspects. This constitutes one major milestone towards broad adoption in CCAM. Beyond that, it is also critical to realize the complexity and limitations of transferring the *CONNECT* research outcome to uncontrolled, real-world scenarios (Section 7.1). In addition, the realization and evaluation of the *CONNECT* framework has yielded important insights that could impact future research in the field of dynamic and cooperative trust evaluations (Section 7.2). Finally, these insights are also extended towards the emerging need of incorporating AI systems as part of modern CCAM deployments (Section 7.3).

7.1 Scale of Testing in Living Lab scenario

As aforementioned, *CONNECT* demonstrates the full process of developing a new trust assessment methodology, incorporating the evaluation of the actual content of V2X data: from its conception and prototyping to real-world evaluation. This includes simulations in hardware-in-the-loop environments, but also evaluations in living-lab scenarios. Regarding the latter case, this involves the evaluation of the *CONNECT* TAF based on data transmitted by an actual V2X-enabled vehicle operating on a test track as shown in the context of the SMTD use case (see Chapter 5). Achieving the significant milestone of TRL 4 (primarily due to the conduction of the experiments in a controlled environment), the *CONNECT* methodology demonstrates the potential to be considered as a candidate for higher TRLs, either independently or in conjunction with other concepts proposed by the CCAM community. Hence, one critical question that comes in the forefront is the following: *What is the roadmap that allows the automotive industry to adopt the* CONNECT *framework and in general evidence-based trust assessment methodologies as part of the decision-making control loop in the avenue for higher levels of automation in autonomous driving?*

Overall, there are four pillars that constitute the roadmap to a wider adoption of the *CONNECT* framework by the CCAM community. First, in order to exhibit higher TRLs - i.e., greater than 5 - it is essential to validate the trust engineering process of the *CONNECT* Trust Assessment Methodology. This validation process needs to take place in tandem with industrial working groups - such as C2C-CC, and 5GAA - in order to ensure alignment with timely CCAM requirements. As demonstrated in [Con25a], *CONNECT* has already taken significant steps towards this. An indicative example is the liaison with 5GAA that culminated in the publication of two white papers covering critical aspects of the trust engineering process. In fact, the second white paper covers the specification of a generic trust assessment methodology that revolves around the accurate identification of the two core trust-related aspects that we have examined throughout the project: Actual Trustworthiness Level (ATL) and Required Trustworthiness Level (RTL). As part of these activities, we also summarize in Section 7.2 the important research questions and challenges that have been identified and could greatly impact future research in the field.

The second core pillar that will unlock a high TRL relates to the evaluation of the *CONNECT* methodology in the context of safety-related properties. For this matter, *CONNECT* has already reached out - through 5GAA - and gained consensus with respect to the envisioned technologies that shall be employed. Nevertheless, there is still work to be done in this front and further research needs to be conducted to enable this convergence across security and safety trust characterizations (see Research Question #3 in Section 7.2). In the same direction, new project initiatives such as ConnRAD and PoDIUM - with which *CONNECT* has already established a liaison [Con25a] - are exploring how the paradigm of evidence-based Subjective Logic can be integrated in the overall safety engineering process in CCAM.

Along with the previous pillar, it is equally essential to ensure wide adoption by industrial stakeholders. This involves the continuation of discussions across different levels of the CCAM community as well as the emergence of additional research initiatives with the aim to address the open challenges that *CONNECT* has identified - part of them is highlighted in Section 7.2. Up to this point, the feedback from the community is promising: through ongoing trust-related activities in the context of technical 5GAA working groups, as well as continuous exploration and research through emerging initiatives on the design of trust mechanisms for evaluating trust in safety-critical CCAM functions.

Finally, another important factor towards higher TRLs involves the enhancement of evaluation scenarios incorporating actual ECUs and evaluating the footprint of the *CONNECT* methodology in the operational behaviour of CCAM functions deployed in real in-vehicle topologies. As the evaluation experiments focused on the feasibility and applicability of the *CONNECT* framework across multiple CCAM scenarios, we employed endpoints that emulate the interfacing with in-vehicle ECU components (e.g., endpoints leveraged as part of the CACC use case 4). Consequently, the natural next step would be to deploy the *CONNECT* artifacts into actual ECU hardware. This will provide valuable insights on the actual footprint of the *CONNECT* artefacts in the residual software stack that is available by modern in-vehicle components.

7.2 Desiderata for a Generic Trust Assessment Methodology

In what follows, we critically reflect on the identified core pillars that heavily affect the trust-aware decision-making process in the *CONNECT* methodology and highlight key outstanding issues that need to be further researched. This should allows us to address one of the core questions around the *CONNECT* Trust Assessment Methodology: How uncertainty can be systematically expressed in a way that can have a meaningful impact on the safety-critical decision process? The following trust engineering insights - presented in the form of research questions - will help guide future research that needs to be conducted so that the *CONNECT* framework can tackle this important challenge.

Research Question #1: Can the *CONNECT* Trust Assessment Framework be generalised to other application domains?

While *CONNECT* as a project, is heavily focusing on automotive scenarios and CCAM, the TAF itself is developed as a generic concept and there is nothing that prevents the architecture and even the prototype from being deployed in a great multitude of domains. In principle, the need to provide helper assertions regarding data trustworthiness extends beyond CCAM concepts and constitutes a generic challenge in almost every domain where cyber-physical systems are deployed. It is easily conceivable how a TAF instance can be deployed, for example, in a smart grid system where different sensors like smart meters and actuators (that produce and distribute energy) have to cooperate to maintain grid stability. Trust models would reflect the structural trust dependencies when, for example, an aggregator collects meter readings from a group of smart meters and forwards them in an aggregated way to a central control centre that, then, controls power production. Respective trust sources would be based on knowledge about devices and trusted computing concepts like remote attestation, but would also include plausibility and consistency checks similar to those found in an automotive misbehaviour detection system. A TAF instance could be

placed centrally in the control centre, but also be federated between different components like aggregators or power plants.

Another similar example could be a smart factory where different robots and tooling machines cooperate to solve a joint production tasks. Again, this creates trust relationships where one entity needs to trust that another entity provides correct data - e.g., on numbers of produced parts that would influence production planning and ordering. Taking it a step further, one could even assess the trustworthiness of one entity to produce parts in a good quality.

Going beyond cyber-physical systems, the overall TAF can conceptually be deployed in every distributed system that consists of multiple entities with trust boundaries between them. This can range from a single software system that involves multiple components or libraries[1] to large scale network topologies where trust assessing entities could be used for establishing a Trusted Path Routing paradigm [BVL+25]. For this purpose, having a generalized, and systematic trust assessment methodology will allow the instantiation of core trust characterization across different domains addressing the intrinsic challenges in the derivation of both ATL, and RTL values. Regarding the latter one, the accurate realization of RTL values introduces several intrinsic challenges, as they may stem from the corresponding risk assessment methodology, applicable to a particular domain. In this case, capturing all hardware and software assets in a risk topology can be a demanding task. At the same time, the availability of such granular risk-related information for each subsystem in a complex topology (e.g., ECU from a third-party OEM), such as the one envisioned inside a vehicle system of systems environment, may not be accessible or available at all. Consequently, there needs to be an adequate level of abstraction in the information that is critical for the trust engineering process, without also compromising the accuracy of the derived RTL values.

So, even though this application domain exploration is orthogonal to the objectives of the *CONNECT* project, our aim is the design of a highly generic system that would only be constrained in its application by the need to identify suitable trust sources, quantify their output, and design suitable trust models capturing the trust relationships in the target environment. Future research efforts could focus on evaluating the TAF instantiation in different domains to identify specific challenges and evaluate the effectiveness.

Research Question #2: What is the impact of *CONNECT* in the community?

CONNECT has introduced the fundamental concepts of trust and trustworthiness in CCAM. In fact it is the first initiative that brings forward the paradigm of subjective logic - before even discussed in the standards - as a core mechanism to provide a systematic and generalized approach towards trust characterization in safety-critical applications.

*CONNECT*ś engagement with the CCAM community sheds light on the missing points that would allow the standardization and, thus, wide adoption of trustworthiness enablers for assessing data and systems. To this end, *CONNECT* has contributed to bridge the gap between current standardization activities and trust assessment. This culminated in the identification of concrete standardization activities towards a generic trust assessment methodology [Ass].

These activities are primarily organized into two categories. On the one hand, standardization efforts shall focus on the specification of interoperable procedures pertaining to the consistent generation, evaluation, and evolution of trust assessments across various systems and scopes. This includes aspects such as the specification of trust model templates, decomposition of trust propositions depending on the available trust sources, and trust opinion quantification functions depending on the available trustworthiness evidence. On the other hand, it is crucial to specify concrete profiles for systems to apply the generic trust methodology while also taking into consideration the requirements and priorities of a particular domain or use case. Such standardized profiles shall act as a guidance for implementers of the trust assessment methodology in order to apply e.g., the appropriate subjective logic operators, and RTL methodology that best-captures the characteristics of specific groups of use cases.

[1] although we acknowledge that our current TAF would be too heavyweight for such an application

Research Question #3: How to reach a consensus in the industry about the benefits of this technology by converging security and safety?

CCAM services are driven by an ecosystem of very diverse partners and, given the many stakeholders involved, it is very hard for a single company to drive their adoption alone: an industry-wide consensus is necessary. However, the industry's focus is not predominantly on the topic of cybersecurity but is mostly driven by the aim to assess trustworthiness from a safety perspective.

Safety engineering is generally seen as a major hurdle in the discussion on CCAM adoption that needs to be overcome. At the moment, it is very hard to run CCAM services with automated driving decisions because safety engineers lack the methodology to decide how to treat vehicle-external data in safety-critical decisions.

Therefore, adoption of CCAM services suffer from lack of methodological integration with safety processes. This is why the presentation and discussion of *CONNECT*'s methodology for cybersecurity gained a lot of attention by safety experts in the 5GAA community. A corresponding technical report made these ideas known to a broad community and many there see a high potential that *CONNECT*'s methodology can be transferred to the safety domain to answer the question of how reliable or accurate V2X data is. *CONNECT*'s results could help reaching an industry-wide consensus on how to treat V2X data in CCAM services in general and how to integrate the concept of trustworthiness into cybersecurity and safety engineering processes. While *CONNECT* demonstrated with a high TRL how the methodology can be applied for cybersecurity, our efforts are fuelling and supporting on-going discussions and efforts to evaluate the application of the methodology in the safety domain.

In December 2024, together with sister projects like PoDIUM and the German project ConnRAD, *CONNECT* has organized a joint workshop to tackle the question on how to integrate subjective-logic-based trust assessment for both security and safety in an integrated way which is now paving the way for follow-up research efforts towards this goal. Through this workshop, participants from all projects have contributed to the identification of the core challenges that need to be tackled to derive a generic and harmonized ATL methodology that would allow the expression of both security and safety properties.

Research Question #4: What are the key dimensions that affect trust characterization?

The experience gained from the thorough evaluation of the *CONNECT* Trust Assessment Methodology throughout the project allows us to critically reflect on the identified key dimensions that heavily affect trust characterization, but also highlight outstanding issues that need to be further researched.

Starting from the ATL calculations, one critical aspect that affects the trust outcome lies in the modelling of uncertainty. In this context, uncertainty interpretation hinges on multiple factors, including the relevance of trust sources, the suitability of the selected trust opinion quantification methods, and the nature of the input data used by these methods. Specifically, different levels of uncertainty can greatly impact the accuracy of the ATL calculations and may stem from either the observed evidence (e.g., lack of evidence, or evidence reported under varying confidence levels) or even the mechanisms used to quantify the evidence into trust opinions (e.g., arising from the inherent limitations or imprecision in the process used to quantify atomic trust opinions). Consequently, the systematic quantification and modelling of the different sources of uncertainty in ATL calculations constitute one of the core factors that impact the overall trust engineering process.

Moving from the ATL Methodology towards the generic *CONNECT* Trust Assessment Framework, it is important to express the trust requirements in a way that allows comparison of computed ATL and configured RTL values. In principle, the derivation of a trust decision is intrinsically linked to the accepted risk that the trustor is willing to take, with respect to the fact that a trustee behaves as expected in a given context. In fact, in *CONNECT* we showcase how the risk-related information extracted through TARA constitutes the common denominator for deriving, on one hand, the RTL (e.g., maximum level of risk that can be considered accepted) and, on the other hand, the ATL value based on evidence indicating the enforcement of the specified security controls and/or the residual risk remaining at accepted levels. Therefore, one key factor that shapes trust outcomes is the ability to express both ATL and RTL values under shared semantics,

making comparison possible and enabling well-founded trust decisions.

Finally, from the experience gained through the dynamic and distributed scenarios in the IMA use case (see Chapter 3), we have identified two aspects that need to be taken into consideration when assessing trustworthiness in continuously evolving environments: convolution and evolution of trust. The dynamic nature of real-world scenarios calls for dynamic trust evaluation, where new evidence continuously shapes the assessment of trust levels. While the concepts of trust convolution are discussed in the subsequent research question, capturing the evolution of trust levels over time is largely affected by the semantics of the considered trust sources and their available evidence. Within *CONNECT*, we distinguish two classes of trust sources, depending on their semantics: binary and probabilistic. The former category provides concrete guarantees, acting as proof of ownership with respect to a specific system property (e.g., OBU has secure boot integrity). Whereas, the latter category provides suggestive evidence pertaining to the detection of an abnormal event under a specific (probabilistic) confidence level. This second kind of trust sources is often associated with statistical patterns in the evidence observations - e.g., for the MBD trust source considered in *CONNECT*, this pattern is found in the temporal correlation of negative evidence. Hence, it becomes crucial for accurate trust characterization to capture these features in the trust assessment process. Within the *CONNECT* scope, we have introduced the ”Temporal-ATL” concept, enabling temporal correlation in the observable MBD evidence.

Research Question #5: What are the benefits of federated TAF modality?

The first *CONNECT* TAF prototype is capable of acting as a standalone component: running in tandem with the target environment under evaluation and interacting with the available trust sources so as to form a trust opinion - i.e., an ATL value - for its target propositions. Apart from in-vehicle scenarios, where a single TAF instance is capable of assessing the trustworthiness of in-vehicle CCAM functions, other realistic scenarios leverage the standalone modality as a building block in order to create a trust plane where different V2X entities are able to exchange trust-related information. This unlocks the formation of a wider perception with respect to the assessed trust propositions characterizing their subcomponents and the exchanged data. In the context of safety-critical, multi-agent systems the benefit of hierarchical interaction between TAF instances may unlock aspects of federation and cognitive computing in the overarching *CONNECT* methodology. Within the *CONNECT* project, the TAF federation is evaluated in the context of TAF instances deployed within vehicles and transmitting ATL reports to a TAF instance running on a MEC infrastructure. This allows the latter TAF instance to construct a wider perception on the trustworthiness of the V2X data transmitted from all available vehicles in the context of an IMA federated scenario.

As already mentioned in Research Question #1, the overall challenges of the *CONNECT* TAF can be transferable to multiple domains. Similarly, the same applies for the concept of federated trust evaluations. In fact, this constitutes one of the imminent challenges encountered in different domains ranging from Trusted Path Routing [BVL+25] concepts to 6G service provisioning. In this regard, *CONNECT* showcases how this federation can be realized in terms of TAF instance-to-TAF instance interaction, highlighting the core challenges that need to be addressed towards the realization of a fully federated trust assessment framework (e.g., synchronization of trust model instances, processing of trust opinions derived from different trust models).

Research Question #6: What *CONNECT* did in ensuring chip-to-cloud-assurance and what are the gaps still remaining considering the fragmentation of HW-based SE and TEEs?

Current security solutions for CCAM where focused on securing the vehicle. ECU security ensured that a specific ECU was protected. Security zones in the vehicle protected high-assurance zones from risks that are introduced by lower security zones such as infotainment. As CCAM services are increasingly dependent on cloud services, this piecemeal approach needed to be expanded to provide an integrated end-to-end security framework that provides security from the ECU (the ”chip”) all the way to the cloud.

Towards this direction, *CONNECT* is not limiting its contributions to the design and prototyping of the core trust assessment functionalities. It also provides all the necessary auxiliary mechanisms for evidence-based monitoring in a verifiable and privacy-preserving manner. Through containerized deployments,

we establish a common framework that can be implemented on (larger) ECUs as well as the cloud. To demonstrate the applicability of our framework, we employ hardware security of recent Intel CPUs to showcase confidential containers. These containers are protected on each ECU. In addition, in [Con25d] we described how these containers can be securely migrated from the ECU to the Edge Cloud / MEC.

These innovations by *CONNECT* enable a wide range of impacts:

Cloud Services Augmenting the Vehicle Collaborative services – such as intersection management – can be securely hosted in the cloud/MEC and thus enhancing efficiency and security of collaboration.

Cloud Offloading If the vehicle is too communication of compute constrained for a given service at a given time, it can offload services into the cloud.

Cloud Failover Cloud services can provide hot standby services that represent the vehicle (similar to digital twins). This allows that other collaborators can use this service as a proxy in case the vehicles is currently not reachable.

Vehicular Cloud Services Similarly, if the vehicle expects to be offline, it can migrate certain lighter-weight cloud servics into the vehicle to ensure that the service remains available locally despite the vehicle being offline.

Support for Heterogeneous Roots of Trust Due to the generic Trust Assessment Framework, we can evaluate and assess a wide range of evidence that is generated by divers hardware roots of trust. E.g. evidence by ARM TrustZone, Intel SGX, and today's smaller ECUs can all be evaluated using this unified framework to then derive unified trust scores. While prototyped on Intel CPUs, this approach allows later deployment on a wide range of technologies.

Sufficient Performance for Most In-vehicle ECUs Our benchmarks have demonstrated that the security enforcement is efficient enough to be implemented on large and mid-range ECUs that are increasingly used. Smaller ECUs are usually single-function and can continue to deploy today's assurance mechanisms while the generated evidence can still be evaluated and assessed.

All the aforementioned auxiliary mechanisms have been designed in a RoT agnostic manner, while the designed cryptographic - and attestation - schemas have followed a crypto-agility approach to remove any hard dependency with specific technology stacks. However, this level of abstraction constitutes a double-edged sword, considering also the heterogeneity of modern ECUs. In fact there are still a lot of fragmentation and interoperability barriers due to the different set of capabilities and interfaces exposed by the underling discrete hardware security elements. This introduces the need for the careful specification of a custom Trusted Computing Base (TCB) which is able to unlock this level of abstraction, provided that the fundamental root of trust properties are present: root of trust for measurement, reporting, and storage. To this extent, *CONNECT* has already presented preliminary steps towards the development of such a custom TCB, namely UBILib [2], with the goal to deliver a universal, RoT-agnostic library that empowers any device to evolve from an untrusted host into a HW-rooted "security hardened" token.

Overall, then chip-to-cloud assurance and services framework allows a wide range of services innovation. By the *CONNECT* end-to-end security architecture, security remains guaranteed in this distributed services architecture. In addition to this chip-to-cloud horizontal integration, we have also demonstrated that we can support different sizes of ECUs and diverse roots of trust to accommodate the diverse ecosystem of suppliers that are important for the automotive market.

[2] https://github.com/ubitech/ubitrust

7.3 Towards Trustworthy AI

While *CONNECT* primarily focuses on assessing the trustworthiness of actors and data in V2X, the CCAM Strategic Research and Innovation Agenda (SRIA) makes clear that the end goal is trustworthy automated decision-making, where AI systems become fundamental to CCAM deployments [CCA24]. For that goal to materialise, the *certainty—and uncertainty*—of trust sources must be accounted for not only at run time but also during AI model training and operation. *CONNECT*'s dynamic trust characterisation offers precisely these building blocks: evidence-based trust signals that can be propagated beyond per-message checks to inform how training datasets are curated and how models later adapt and behave. In this sense, *CONNECT* provides a CCAM-specific foundation that is compatible with broader trustworthy-AI practices now emerging in the CCAM community [AI423].

In parallel, CCAM use cases are distributing AI across the vehicle–infrastructure–edge continuum for collective perception, cooperative decision-making, and actuation, work that must reconcile trust with latency and resource limits. This architectural shift also aligns with the Software-Defined Vehicle vision, where data and AI services are orchestrated across vehicle, edge, and cloud. In this context, *CONNECT*'s trust signals become a scheduling and safety primitive, guiding decisions on *what* to use, *where* to compute, and *when* to defer, consistent with emerging work on trustworthy edge intelligence [WWW+23, ap24]. The remainder of this section therefore examines the bridge from trustworthy data to trustworthy AI models as a forward path for *CONNECT* not to prescribe solutions, but to articulate the open challenges that must be addressed and the ways in which *CONNECT*'s building blocks can contribute.

This brings us first of all to the concept of robustness. In conventional machine learning, robustness is narrowly defined as the model's stability under small, adversarial perturbations. However, in *CONNECT* we also approached robustness as a core KPI for evaluating the Trust Assessment Framework (TAF) itself, ensuring that trust inferences remain stable under noisy, incomplete, or even conflicting V2X inputs. Beyond this, an open challenge remains: how to elevate robustness towards helper assertions on the correctness of AI model outputs, particularly in the context of uncertainty. In safety-critical domains like CCAM, this means not only that the trust assessment mechanism is reliable, but also that its outputs can meaningfully inform and constrain the internal classification or decision-making process of downstream AI components. This was discussed extensively in the AI Trustworthiness Workshop that *CONNECT* organized in April 2025[3], where it became obvious that achieving robust and trustworthy AI in practice will require such integration, where trust assessments act as epistemic anchors for AI reasoning, enabling AI systems function in the absence of perfect data, reason under uncertainty, and adapt when their sources of evidence shift or degrade.

In this sense, robustness is a reflection of how well the system handles imperfect or contested knowledge. This perspective aligns with *CONNECT*'s approach: rather than assuming clean, well-curated data, it confronts the problem of epistemic uncertainty directly, treating trust as a dynamic variable inferred from context and source coherence and reliability. By integrating these trust signals before data is consumed by AI components, *CONNECT* effectively redefines robustness not as resistance to manipulation, but as the capacity to act wisely under uncertainty. Availability of such dynamic trust assessment mechanisms, can avail next-generation vehicle and infrastructure-based environment perception technologies for robust, reliable, and trustworthy CCAM operations. This intelligence lies at the heart of sense-think-act systems of CCAM considering the vehicle, the infrastructure, the cloud at-the-edge while guaranteeing security and safety convergence.

This re-framing has the ability to overcome the limitations of traditional adversarial ML techniques as a primary defence strategy. While useful in controlled settings, adversarial training and similar techniques are fundamentally reactive and narrow, tailored to specific threat models. They operate under assumptions, like known perturbation bounds or stable input distributions [SSKE18, OFR+19], that rarely hold in decentralized, evolving systems such as CCAM. More importantly, they ignore a deeper epistemic flaw:

[3] https://horizon-connect.eu/workshop-on-trustworthy-ai-2/

AI systems trained on untrustworthy or unverified data cannot be made trustworthy through model-side adjustments alone. Without knowing which data to believe, no amount of parameter tuning can ensure meaningful or justifiable decisions.

The AI Trustworthiness Workshop, which *CONNECT* organized, explored exactly these aspects, in the context of safety-critical systems, like CCAM, but also extended the discussions to the broader field of Trustworthy AI. The resulting report identified a more extended understanding of robustness as one of the core challenges towards the manifestation of Edge-AI mechanisms as a building block for resilient CCAM; one that includes uncertainty quantification, context awareness, and resilience to epistemic limitations, rather than narrow resistance to adversarial inputs.

Taken together, these considerations reveal a deeper challenge: dataset trustworthiness and model trustworthiness are not independent, but tightly coupled aspects of the same unresolved problem. In CCAM, achieving reliable AI under uncertainty demands the ability to trace, preserve, and reason over trust information from data acquisition to model adaptation. Yet, how to represent and propagate this trust through complex learning pipelines remains an open question. The next two subsections examine this challenge, first at the level of training datasets, and then, at the point where models inherit, transform, or obscure that trust.

7.3.1 From Data Quality to Training Dataset Trustworthiness

Artificial Intelligence (AI) systems increasingly underpin high-stakes decisions in domains such as autonomous mobility. While research has advanced the trustworthiness of AI models through fairness, explainability, and privacy-preserving techniques, less attention has been paid to the trustworthiness of the training datasets that underpin those models. Yet empirical studies show that dataset issues like sampling bias, label noise, and privacy vulnerabilities can undermine model fairness, robustness, and interpretability [SBS+24]. In critical domains, even subtle flaws in data collection or curation can propagate into harmful downstream outcomes.

While data quality refers to intrinsic properties like accuracy, completeness, and consistency [Int22], data trustworthiness is not a static attribute of data but a property that emerges from how data is produced, curated, and shared and how much confidence we can place in that process. A dataset may meet quality benchmarks yet remain untrustworthy, for example, if it originates from opaque or untrustworthy sources. For instance, as *CONNECT* has shown, in multi-agent systems such as connected vehicles, sensor data from different sources may individually appear high quality, but differ in calibration, source reliability, or integrity. Similarly, in federated learning, local datasets may exhibit sampling bias or undetected distribution shifts, making global judgments difficult without modeling uncertainty in the evidence each node contributes. Trustworthiness thus builds upon data quality, adding an epistemic layer particularly relevant in decentralized or uncertain environments where evidence is fragmented or ambiguous.

CONNECT has shown how to assess and quantify the trustworthiness of data based on actual evidence about various trust properties (e.g., integrity, authenticity, accuracy). However, the project has largely focused on properties of individual data points, where evidence can often be tied to a specific source or transaction. What remains unsolved is how to assess dataset-level properties that only emerge when data is considered collectively. Properties like bias, fairness, or representativeness cannot be inferred from isolated records; they require a global view and sufficient coverage to be meaningfully evaluated. What kind of evidence do we need to evaluate such properties, and how do we quantify trust based on it, especially if such evidence is incomplete? Addressing this challenge calls for a new methodological foundation, one that enables reasoning under uncertainty, supports aggregation across evidence sources, and yields interpretable trust assessments for global dataset properties.

7.3.2 From Dataset Trustworthiness to Trust-Integrated Models

From the above discussion, it becomes clear that a model's trustworthiness cannot be evaluated independently of the data it learns from, since trustworthiness propagates from data into models, and any gap in epistemic confidence at the dataset level becomes a potential fault line in the AI system's decision logic. This creates an urgent need to move from per-dataset trust assessments toward a deeper integration of epistemic awareness into the learning pipeline itself. In practice, models are adapted repeatedly (transfer, continual, or federated learning), and every adaptation implicitly assumes that all incoming data are equally credible. The open challenge is therefore to elevate trust from a property of data to a property within the learning and operation process.

Conceptually, this challenge has three interlocking dimensions. **(i) Representation:** trust must be represented as a first-class signal that survives the journey from V2X acquisition to training and adaptation. Formalisms such as Subjective Logic (belief, disbelief, and uncertainty) make this *explicit* and quantifiable across sources and time [CNB20]. **(ii) Propagation:** in transfer or federated learning, representations learned on heterogeneous sources are re-used under distribution shift; here, trust information must *propagate* with the data and features, shaping how knowledge is imported, discounted, or rejected [WH24]. Indeed, data provenance may be partial, sources may disagree, and operating conditions evolve. **(iii) Decision behaviour under uncertainty:** even with careful curation, predictive uncertainty and calibration degrade under shift. Trustworthy systems therefore need new approaches on how to communicate residual uncertainty.

It would be meaningful to explore how *CONNECT*'s message-level trust assessment and trust opinions computed at the V2X layer can be treated as epistemic metadata that flow into model adaptation, so the model's learned representations remain conditioned on evidence quality. This aligns with broader trustworthy-transfer viewpoints [WH24], which argue that transfer pipelines should be evaluated not just for accuracy but for how they manage fairness, robustness, privacy, and transparency during knowledge reuse. We identify this as a promising open research direction for the future.

Chapter 8

Conclusion and Outlook

This deliverable detailed the final validation activities of the *CONNECT* project, marking the successful transition from the component-level analysis of our initial work to a holistic, end-to-end evaluation of the integrated *CONNECT* Trust and Security Framework. By rigorously testing the framework against three diverse and realistic use cases—Intersection Movement Assistance (IMA), Cooperative Adaptive Cruise Control (C-ACC), and Slow Moving Traffic Detection (SMTD)—we have demonstrated its tangible impact on enhancing the resilience, robustness, and trustworthiness of the CCAM ecosystem.

Our empirical results provide compelling evidence of the framework's effectiveness. The IMA evaluation validated the framework's core security promise: its ability to successfully mitigate perception-based attacks by identifying and excluding untrustworthy data. This was enabled by novel mechanisms such as the Temporal-ATL, which proved essential for correctly interpreting time-dependent evidence from sources like Misbehavior Detection (MBD). Furthermore, the federated architecture leveraging the MEC was shown to be a cornerstone of the *CONNECT* design, significantly enhancing system robustness by aggregating evidence from multiple sources, leading to faster and more accurate threat mitigation. The C-ACC evaluation showcased the system's high performance and flexibility, with the pull-based Trust Assessment Framework (TAF) delivering trust verdicts in milliseconds when using cached data, while still meeting KPIs for non-time-critical requests that mandate fresh evidence collection.

Beyond performance metrics, this work has yielded crucial lessons for the future design of trustworthy CCAM systems. A key insight from the SMTD evaluation is the critical dependency of the entire trust system on the quality of its underlying evidence sources. The false positives generated by the MBD's Kalman filter during turns underscore that the TAF, while powerful, cannot compensate for poorly designed detectors. Similarly, the federated IMA evaluation highlighted the architectural challenge of "double counting" evidence—a fundamental risk in any distributed system that we successfully mitigated through careful information flow design. Finally, the stakeholder survey validated our value-pluralist approach, confirming that for CCAM to be truly trustworthy, it must address a broad spectrum of requirements beyond technical safety, including transparency, accountability, and fairness.

The results and lessons learned illuminate clear paths for future work. This includes extending testing from controlled living labs to unpredictable open-road scenarios to further raise the Technology Readiness Level (TRL); engaging with a broader public (stakeholder sets 3 and 4) to deepen our understanding of societal trust; and pursuing the standardization of successful, low-overhead mechanisms like the Trustworthiness Claim (TC), which was shown to have no significant impact on V2X communication performance.

In conclusion, the *CONNECT* project has successfully designed, implemented, and validated a comprehensive, multi-layered framework for dynamic trust assessment in cooperative mobility. By addressing complex challenges in evidence fusion, temporal trust dynamics, and federated security architectures, *CONNECT* has made a tangible contribution towards building a safer, more resilient, and ultimately more trustworthy CCAM ecosystem for the future.

Appendix A

Appendix

A.1 SMTD plots

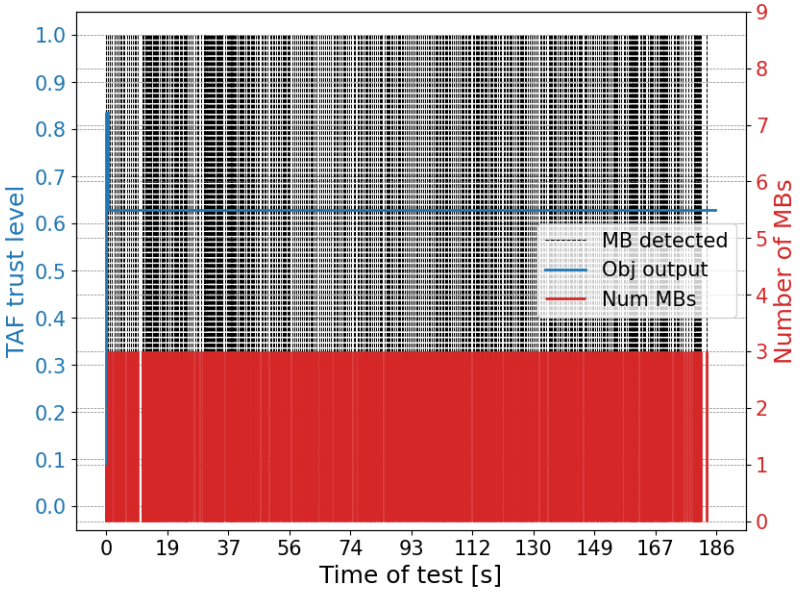


Figure A.1: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in every CPM. ATL of the perceived object.

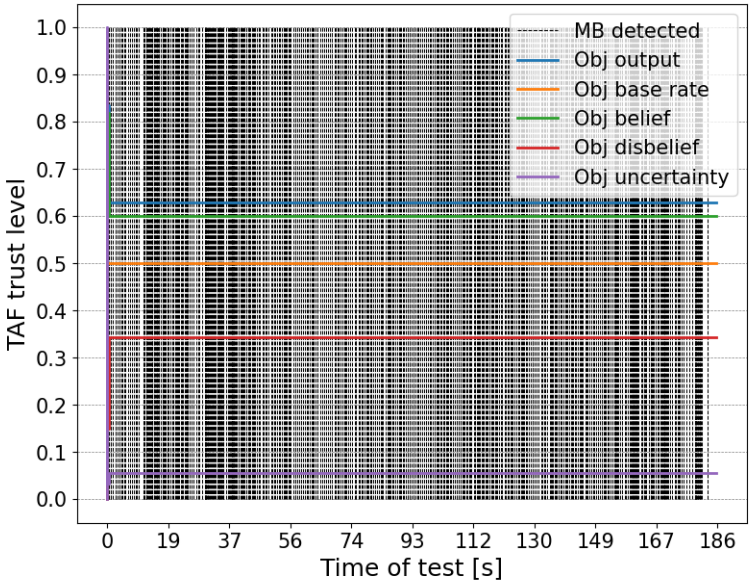


Figure A.2: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in every CPM. TAF parameters.

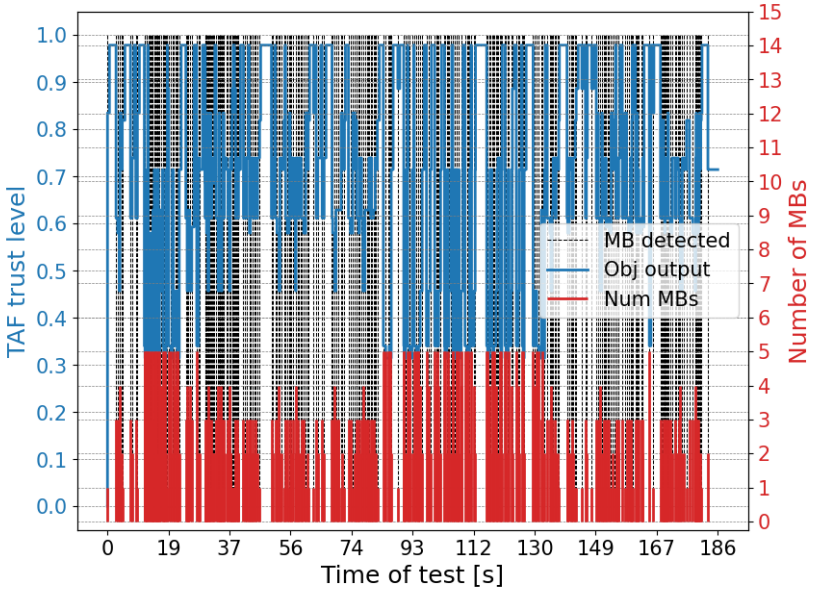


Figure A.3: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in one third of CPMs. ATL of the perceived object.

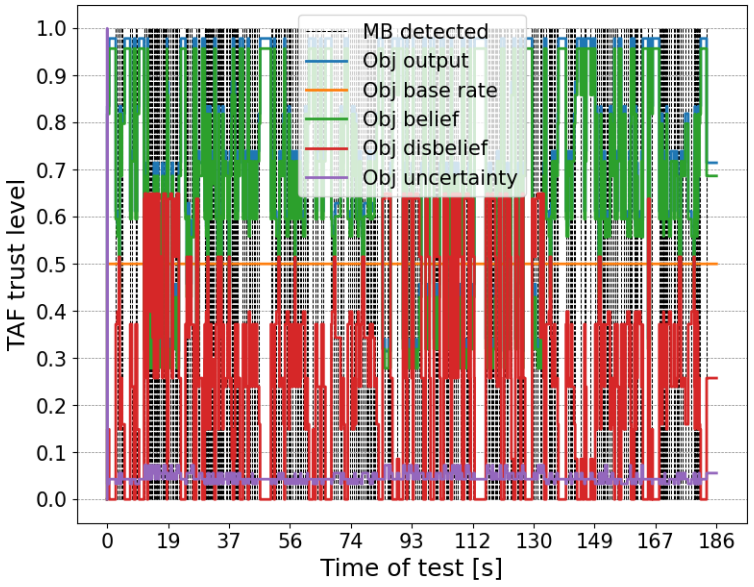


Figure A.4: Evolution of the Actual Trust Level (ATL) of the perceived object in a simulated scenario with MBs introduced in one third of CPMs. TAF parameters.

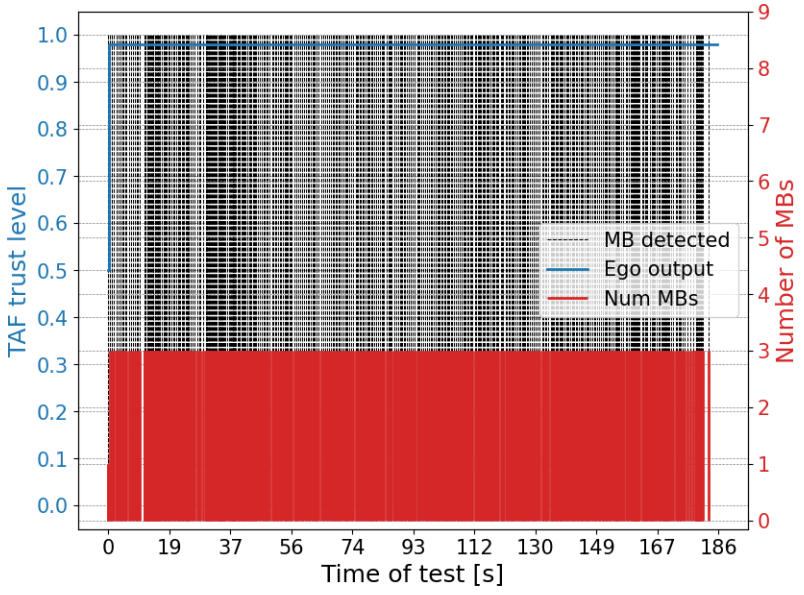


Figure A.5: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on every CPM. ATL of the ego-vehicle.

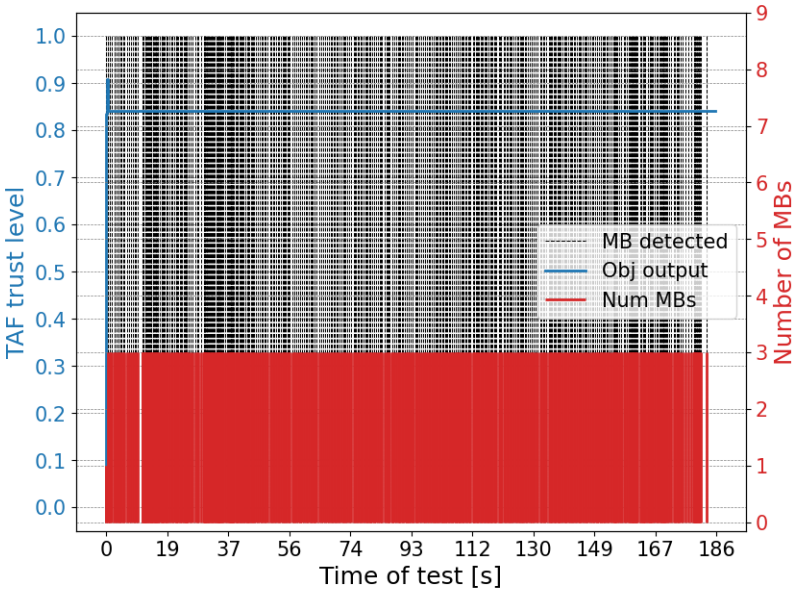


Figure A.6: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on every CPM. ATL of the perceived object.

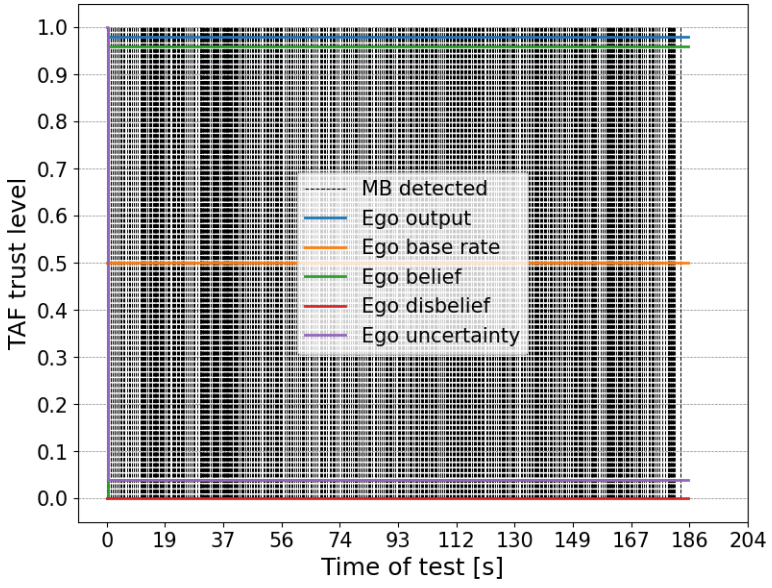


Figure A.7: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on every CPM. TAF parameters of the ego-vehicle.

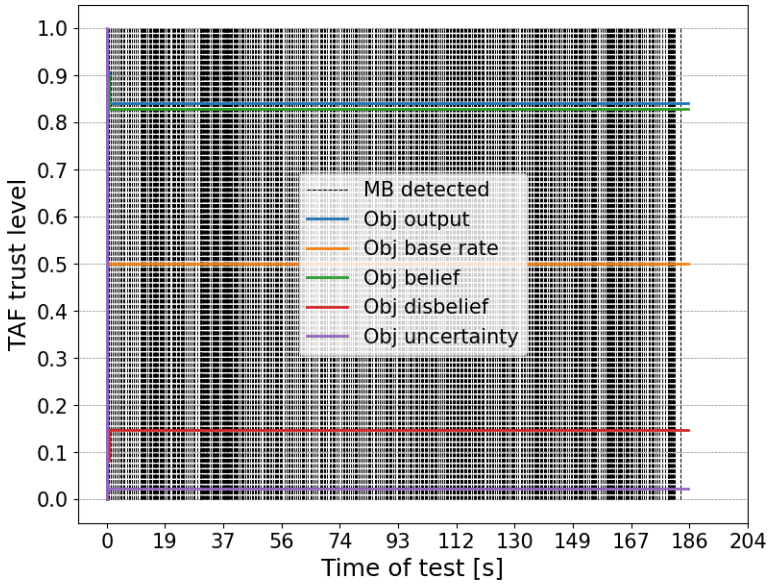


Figure A.8: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on every CPM. TAF parameters of the perceived object.

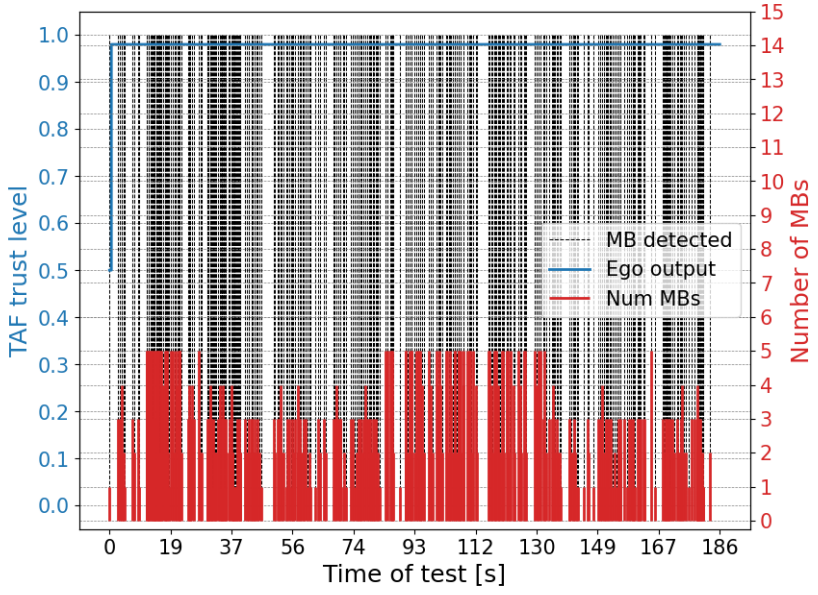


Figure A.9: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on one third of the CPMs. ATL of the ego-vehicle.

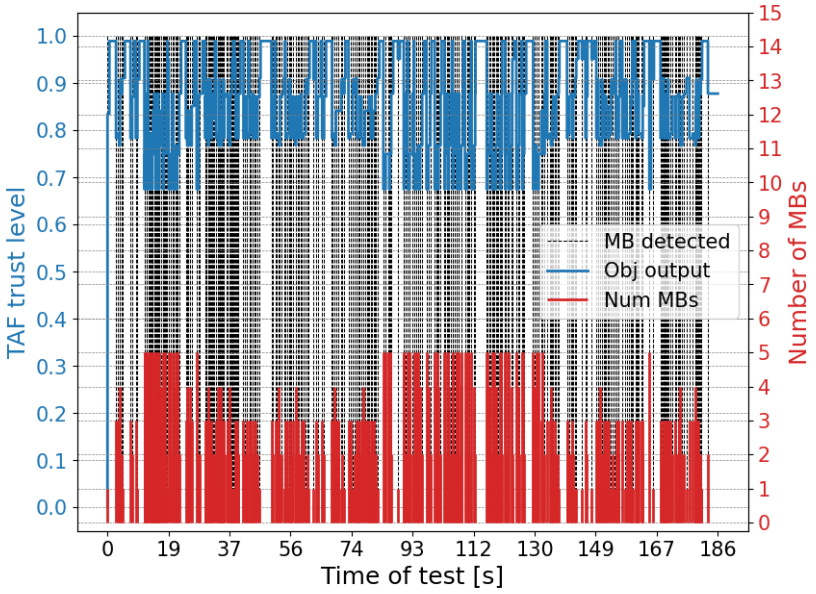


Figure A.10: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on one third of the CPMs. ATL of the perceived object.

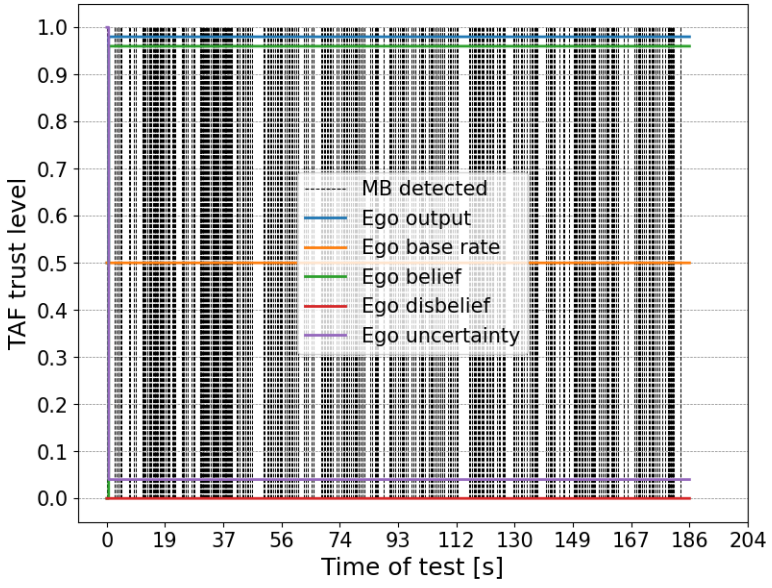


Figure A.11: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on one third of the CPMs. TAF parameters of the ego-vehicle.

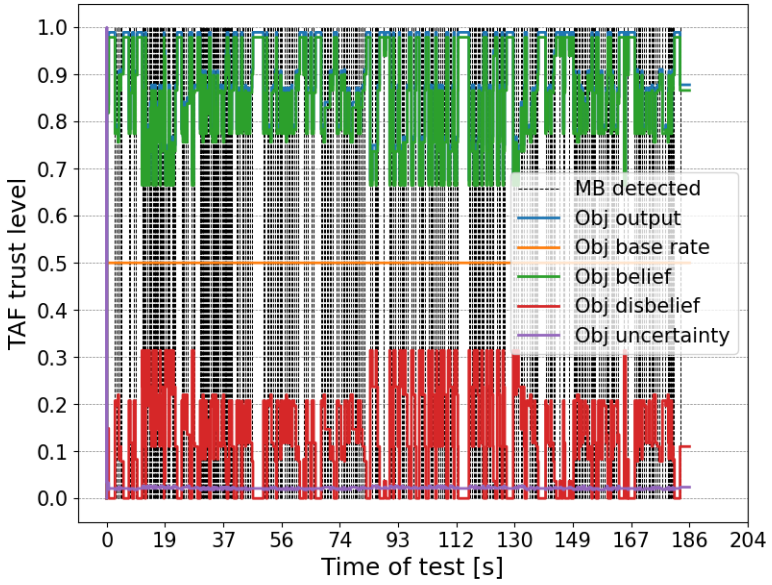


Figure A.12: Evolution of the Actual Trust Level (ATL) in a simulated scenario with MBs introduced on one third of the CPMs. TAF parameters of the perceived object.

Appendix B

Glossary and User Roles

A-ECU An *A-ECU* is an *electronic control unit (ECU)* with a *Trusted Execution Environment (TEE)* providing secure storage for keys and other data. it is able to do asymmetric and symmetric cryptography.

AIV Attestation and Integrity Verification.

CCAM The European Commission has on 30th of November 2016 adopted a European Strategy on Cooperative Intelligent Transport Systems (C-ITS), a milestone initiative towards cooperative, connected and automated mobility. The objective of the C-ITS Strategy is to facilitate the convergence of investments and regulatory frameworks across the EU, in order to see deployment of mature C-ITS services in 2019 and beyond [CCA].

DLT Distributed Ledger Technology.

ECU An electronic control unit (ECU), also known as an electronic control module (ECM). In automotive electronics it is an embedded system that controls one or more of the electrical systems or subsystems in a car or other motor vehicle.

FL The Facility Layer (FL) manages the (kinematic) data stemming from the in-vehicle sensors by relaying them to all components of the Vehicle Manager that have subscribed to receive them. These are essentially the: (i) CAM/CPM Encoder/Decoder component that will start the construction of the respective V2X messages (e.g., CAM, CPM) to be broadcasted either following the currently ETSI specified standards or including also the Verifiable Presentations (VPs) comprising the trust-related information outputed by the TCH (i.e., T-CAM/T-CPM messages), (ii) CCAM Application module for constructing the local view of the vehicle's vicinity towards supporting the decisions making process of the service (e.g., breaking, changing lanes, etc.), (iii) the Misbehavior Detection service for checking the vercity of the measures kinematic data through a series of plausibility checks, and (iv) Trust Assessment Framework (TAF) for associating a Trust Opinion to each data object. It is important to highlight that the FL doesn't perform any processing or checks to the received data. Any verification controls, especially for asserting to the integrity of the data and its safety in the context of been signed and processed by only "*certified*" applications is performed by the IAM.

MBD The Mis-behaviour Detector (MBD) component monitors the data from the vehicle and from elsewhere (from CPM/CAM messages) and looks for anomalies. If these are detected is sends misbehaviour reports to the TAF and outside of the vehicle. Reports for the TAF will be 'normally' signed, while those being sent outside will be anonymously signed.

SGX *Intel SGX* is a hardware feature of Intel CPUs that provides a *TEE* for user-space applications on Intel CPUs. The goal is to protect an application from unauthorized access or modification by any component outside the TEE. I.e. neither the operating system nor other untrusted applications should be able to breach the confidentiality or integrity of the protected application.

TAF The TAF component does the trust assessments and forms trust opinions on the vehicle and data. The trust opinion on the data is sent outside the vehicle and needs to be anonymously signed.

TCB The *Trusted Computing Base (TCB)* of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security, in the sense that bugs or vulnerabilities occurring inside the TCB might jeopardize the security properties of the entire system. By contrast, parts of a computer system that lie outside the TCB must not be able to misbehave in a way that would leak any more privileges than are granted to them in accordance to the system's security policy.

TCH The Trustworthiness Claims Handler (TCH) is the component responsible for sharing all trust-related information outside the Vehicle in a privacy-preserving manner. This data bundle (encoded in the context of a VP) comprises Trustworthiness Claims (TCs), the Trust Opinion (produced by the TAF) and the Misbehavior Report (produced by the MBD). The TC is usually produced (by the Attester) so as to provide trustworthiness evidence ("Trust Source") that can be used for appraising the trustworthiness level of the Attester in a **measurable** and **verifiable** manner. Measurable reflects the ability of the TAF to assess an attribute of the Attester against a pre-defined metric (e.g., RTL) while verifiability highlights the need for all claims to have integrity, freshness and to be provably & non-reputably bound to the identity of the original Attester. Examples sets of TCs might include (among other attributes) evidence on system properties including: (i) integrity in the context that all transited devices (e.g., ECUs) have booted with known hardware and firmware; (ii) safety meaning that all transited devices are from a set of vendors and are running certified software applications containing the latest patches and (iii) communication integrity.

TEE A *Trusted Execution Environment* allows to execute applications while enforcing well-defined security policies for a given application. An example is *Intel Software Guard Extensions (Intel SGX)*.

ZC The *Zonal controller (ZC)* is an *A-ECU* that acts as a gateway between the ECUs and the vehicle computer. As an *A-ECU* they will have a *TEE* providing secure storage for keys and other data and will be able to do asymmetric and symmetric cryptography.

References

[Adm] National Highway Traffic Safety Administration. Human factors design guidance for driver-vehicle interfaces.

[AI423] Methodology for trustworthy ai in ccam. https://www.connectedautomateddriving.eu/blog/methodology-for-trustworthy-ai-in-ccam/, Nov 2023. AI4CCAM blog post.

[ap24] Anonymous (arXiv preprint). Software defined vehicles for development of deterministic services, 2024. arXiv:2407.17287.

[Ass] 5GAA Automotive Association. Creating Trust in Connected and Automated Vehicles. https://5gaa.org/creating-trust-in-connected-and-automated-vehicles/.

[ATG+16] Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Dan O'Keeffe, Mark L. Stillwell, David Goltzsche, Dave Eyers, Rüdiger Kapitza, Peter Pietzuch, and Christof Fetzer. SCONE: Secure linux containers with intel SGX. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 689–703, Savannah, GA, November 2016. USENIX Association.

[BVL+25] Henk Birkholz, Eric Voit, Peter Chunchi Liu, Diego Lopez, and Meiling Chen. Trusted Path Routing. Internet-Draft draft-voit-rats-trustworthy-path-routing-12, Internet Engineering Task Force, July 2025. Work in Progress.

[C-R23] C-Roads Platform, Working Group 2 Technical Aspects, Taskforce 4 Hybrid Communication. C-ITS IP Based Interface Profile Version 2.0.8. Technical report, C-Roads, 2023.

[CAR19] CAR 2 CAR Communication Consortium. Guidance for day 2 and beyond roadmap. Technical report, CAR 2 CAR Communication Consortium, 2019.

[CCA] Cooperative, connected and automated mobility (CCAM). https://transport.ec.europa.eu/transport-themes/intelligent-transport-systems/cooperative-connected-and-automated-mobility-ccam_en.

[CCA24] CCAM Partnership. Strategic Research and Innovation Agenda 2021-2027. Technical report, January 2024.

[CNB20] Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. There is hope after all: Quantifying opinion and trustworthiness in neural networks. *Frontiers in Artificial Intelligence*, 3:54, 2020.

[CON23a] CONNECT. Operational landscape, requirements and reference architecture – final version. Deliverable D2.2, Project 101069688 within HORIZON-CL5-2021-D6-01, August 2023.

[Con23b] The CONNECT Consortium. Distributed processing and CCAM trust functions offloading & data space modelling. Deliverable D5.1, Project 101069688 within HORIZON-CL5-2021-D6-01, Nov. 2023.

[Con23c] The CONNECT Consortium. Operational landscape, requirements and reference architecture - initial version. Deliverable D2.1, Project 101069688 within HORIZON-CL5-2021-D6-01, Nov. 2023.

[Con24a] The CONNECT Consortium. Distributed processing, fast offloading and MEC-enabled orchestrator. Deliverable D5.2, Project 101069688 within HORIZON-CL5-2021-D6-01, Mar. 2024.

[Con24b] The CONNECT Consortium. Integrated framework (first release) and use case analysis. Deliverable D6.1, Project 101069688 within HORIZON-CL5-2021-D6-01, May 2024.

[Con24c] The CONNECT Consortium. Trust & risk assessment and CAD twinning framework (initial version). Deliverable D3.2, Project 101069688 within HORIZON-CL5-2021-D6-01, February 2024.

[Con24d] The CONNECT Consortium. Virtualization- and edge-based security and trust extensions (first release). Deliverable D4.2, Project 101069688 within HORIZON-CL5-2021-D6-01, Jan. 2024.

[Con25a] The CONNECT Consortium. Dissemination, communication, clustering activities including concrete exploitation measures. Deliverable D7.3, Project 101069688 within HORIZON-CL5-2021-D6-01, August 2025.

[Con25b] The CONNECT Consortium. MEC-enabled orchestrator, continuous authorization, trust management and DLT-based secure information exchange. Deliverable D5.3, Project 101069688 within HORIZON-CL5-2021-D6-01, Mar. 2025.

[Con25c] The CONNECT Consortium. Trust & risk assessment and CAD twinning framework (final version). Deliverable D3.3, Project 101069688 within HORIZON-CL5-2021-D6-01, June 2025.

[Con25d] The CONNECT Consortium. Virtualization- and edge-based security and trust extensions (final release). Deliverable D4.3, Project 101069688 within HORIZON-CL5-2021-D6-01, Mar. 2025.

[cTPV17] Chia che Tsai, Donald E. Porter, and Mona Vij. Graphene-SGX: A practical library OS for unmodified applications on SGX. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 645–658, Santa Clara, CA, July 2017. USENIX Association.

[EB-] Samsung specifications: Eb-bg531bbe battery - 2400mah li-ion.

[ETS14a] ETSI. ETSI EN 302 895 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM), 2014.

[ETS14b] ETSI EN 302 637-2 V1.3.1. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical report, ETSI, 2014.

[ETS23a] ETSI TS 103 324 V2.1.1. Collective Perception Service. Standard, European Telecommunications Standards Institute (ETSI), June 2023.

[ETS23b] ETSI TS 103 759, v2.1.1. Intelligent Transport Systems (ITS); Security; Misbehaviour Reporting service; Release 2. Technical report, ETSI, 2023.

[GRCR+25] Diego Gasco, Carlos Mateo Risma Carletti, Francesco Raviglione, Marco Rapelli, Claudio Casetti, et al. TRACEN-X: Telemetry Replay and Analysis of CAN Bus and External Navigation Data. In *2025 IEEE 102nd Vehicular Technology Conference (VTC2025-Fall)*, pages 1–5. IEEE, 2025.

[HA24] Adam Henschke and Chirag Arora. Pluralism and the Design of Autonomous Vehicles. *Philosophy & Technology*, 37(3):115, 2024.

[Hen20] Adam Henschke. Trust and Resilient Autonomous Driving Systems. *Ethics and Information Technology*, 22:81–92, 2020.

[Int22] International Organization for Standardization. ISO/IEC 22989:2022 - Information technology — Artificial intelligence — Artificial intelligence concepts and terminology. https://www.iso.org/standard/74296.html, 2022.

[M+20] Sajjad Mozaffari et al. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):33–47, 2020.

[MSS+13] Vicente Milanés, Steven E Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on intelligent transportation systems*, 15(1):296–305, 2013.

[Nyh18] Sven Nyholm. Attributing Agency to Automated Systems: Reflections on Human–Robot Collaborations and Responsibility-Loci. *Science and Engineering Ethics*, 24(4):1201–1219, 2018.

[OFR+19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[RH17] Scott Robbins and Adam Henschke. Designing for Democracy: Bulk Data and Authoritarianism. *Surveillance and Society*, 15(3):582–589, 2017.

[RRC24] Marco Rapelli, Francesco Raviglione, and Claudio Casetti. OScar: An ETSI-Compliant C-ITS Stack for Field-Testing with Embedded Hardware Devices. In *2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet)*, pages 1–4, 2024.

[SBS+24] Daniel Schwabe, Katinka Becker, Martin Seyferth, Andreas Klaß, and Tobias Schaeffter. The METRIC-framework for assessing data quality for trustworthy AI in medicine: a systematic review. *NPJ Digital Medicine*, 7(1):203, 2024.

[Sol25] Sadjad Soltanzadeh. A Metaphysical Account of Agency for Technology Governance. *AI & Society*, 40:1723–1734, 2025.

[SSKE18] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[WH24] Jun Wu and Jingrui He. Trustworthy transfer learning: A survey. *arXiv preprint arXiv:2412.14116*, 2024.

[WWW+23] Xiaojie Wang, Beibei Wang, Yu Wu, Zhaolong Ning, Song Guo, and Fei Richard Yu. A survey on trustworthy edge intelligence: From security and reliability to transparency and sustainability. *arXiv preprint arXiv:2310.17944*, 2023.

[ZJN24] Jiahao Zhang, Ines Ben Jemaa, and Fawzi Nashashibi. Simulation framework of misbehavior detection and mitigation for collective perception services. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2437–2442. IEEE, 2024.