CONNECT
CCAM TRUST & RESILIENCE

D3.1

Architectural Specification of CONNECT Trust Assessment Framework, Operation and Interaction

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:	Report
Reference Number:	D6-01-04 / D3.1 / 1.2.1 - Revised on April 04, 2024
Workpackage:	WP 3
Due Date:	30st June, 2023
Actual Submission Date:	10th July, 2023
Responsible Organisation:	Ulm University
Editor:	Artur Hermann
Dissemination Level:	PU
Revision:	1.2.1 - Revised on April 04, 2024
Abstract:	Deliverable 3.1 presents initial work towards the Trust Asssessment Framework (TAF). It includes definition of core terminology for trust assessment, a state of the art analysis and requirements engineering which motivates our decision for a TAF based on Subjective Logic. Subsequently, an approach to trust modeling is described including a methodology for deriving trust relationships and examples for such trust relationships and trust sources are given based on the four CONNECT use case. Finally, a high-level architecture of the TAF is presented and first considerations for the evaluation of the TAF are described.
Keywords:	Trust Assessment Framework, Subjective Logic, State of the Art, Trust Assessment Framework Architecture

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

Artur Hermann (UULM)

Contributors (ordered according to beneficiary numbers)

Anna Angelogianni, Thanassis Giannetsos (UBITECH)
Ana Petrovska, Ioannis Krontiris, Theo Dimitrakos (HUAWEI)
Chris Newton, Liqun Chen (SURREY)
Panagiotis Pantazopoulos, Pavlos Basaras (ICCS)
Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
Anderson Ramon Ferraz de Lucena, Alexander Kiening (DENSO)
Dmitrii Kuvaiskii, Matthias Schunter (INTEL)
Chirag Arora, Adam Henschke (UTWENTE)
Marco Zanzola (CRF)
Claudio Casetti, Marco Rapelli (POLITO)
Francesca Bassi, Ines Ben Jemaa (IRTSX)

Disclaimer

Executive Summary

This deliverable provides a first specification of the Trust Assessment Framework (TAF), which is a main building block of the CONNECT project. To this end, various aspects ranging from the definition of terminology to the evaluation of the TAF will be described.

In the first part of this deliverable, a roadmap will be described how the TAF will be developed in three distinct phases where functionality will iteratively increased. In step one, a standalone TAF is provided that models and evaluates trust relationships within a vehicle or MEC based on a selected set of trust sources. In the second step, a federation of TAFs is added. This federation allows TAFs from different entities to work together in a cooperative system. In step three, a TAF-DT is added to the architecture. The TAF-DT is a digital twin of a TAF that can run in a MEC to which a vehicle's trust computations can be offloaded.

Following this roadmap, terms and definitions for trust and trust assessment will be introduced, which will serve as common language for WP 3 and the whole project. Specifically, terms such as trust, trust relationships, and trust networks will be specified, providing the basis for a clear description of the TAF and its architecture.

After the specification of the terminology, a state of the art analysis of trust frameworks will be performed to give an overview of already existing works in the automotive domain, but also beyond to identify gaps in the context of trust assessment. In a next step, the requirements for trust assessment in the automotive domain will be described. Based on an overview of possible decision logics and the identified requirements, it will be explained why a TAF based on subjective logic provides superior trust assessment capabilities compared to other approaches.

Next, the deliverable will focus on the trust modeling part of the TAF. For this purpose, a methodology for identifying and describing trust relationships is defined and examples of trust relationships in all CONNECT use cases are identified. In addition, trust sources are enumerated and categorized in order to assess trust in the identified trust relationships.

Based on this, a first high-level architecture of the TAF will be described along with its functional specifications and its individual components.

Finally, we will introduce considerations and properties of interests describing how later evaluation of the different TAF phases are planned to be conducted.

All the aspects mentioned above form the basis for the next deliverable, in which a fine-grained architecture for the standalone TAF and an implementation of a first prototype will be presented.

Contents

List of Figures

List of Tables

Document History

Version	Date	Summary of changes	List of Contributors
v0.1	30.03.2023	Table of Contents & Allocation of tasks to the partners	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v0.2	13.04.2023	Description of the SOTA in decision logics and motivation behind adopting Subjective Logic as the foundation of the CONNECT Trust Assessment Framework (Chapter 5)	Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v0.3	28.04.2023	Definition of the possible types of Trust Sources that can be used as evidence for basing the calculation of a trust opinion (Chapter 6)	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Chris Newton, Liqun Chen (SURREY), Anna Angelogianni, Thanassis Giannetsos (UBITECH), Anderson Ramon Ferraz de Lucena, Alexander Kiening (DENSO)
v0.4	12.05.2023	Definition of the Evaluation Methodology of the TAF framework based on the functional specifications defined in D2.1 (first draft) (Chapter 8)	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Anna Angelogianni, Thanassis Giannetsos (UBITECH)
v0.5	22.05.2023	Approach and time plan for the design and implementation of the envisioned TAF variants (Chapter 3)	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Anna Angelogianni, Thanassis Giannetsos (UBITECH)
v0.6	30.05.2023	Vocabulary of all trust-related terminology that will be adopted for all CONNECT activities (Chapter 2)	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Anna Angelogianni, Thanassis Giannetsos (UBITECH), Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Chris Newton, Liqun Chen (SURREY)
v0.7	07.06.2023	Definition of methodology for identifying the trust relationships, comprising a trust model, as well as documentation of the trust relationships for the envisioned use cases (Chapter 6)	Anderson Ramon Ferraz de Lucena, Alexander Kiening (DENSO), Marco Zanzola (CRF), Francesca Bassi, Ines Ben Jemaa (IRTSX), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI)
v0.8	08.06.2023	State of the art analysis in trust assessment mechanisms and solutions in the automotive domain (Chapter 5)	Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v0.9	09.06.2023	Methodology for trust relationships and high level TAF architecture (Chapter 7)	Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)

v1.0	12.06.2023	Trustworthiness and trust definition (Chapter 2)	Chirag Arora, Adam Henschke (UTWENTE), Anna Angelogianni, Thanassis Giannetsos (UBITECH), Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM), Francesca Bassi, Ines Ben Jemaa (IRTSX)
v1.01	16.06.2023	Final version of the definition of the Evaluation Methodology with a detailed list of all properties of interest based on which the assessment of the TAF will be done in the context of the envisioned use cases (Chapter 8)	Anna Angelogianni, Thanassis Giannetsos (UBITECH), Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v1.1	20.06.2023	Functional specification of TAF (Chapter 7)	Ana Petrovska, Ioannis Krontiris, Teo Dimitrakos (HUAWEI), Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v1.18	26.06.2023	Internal review of the deliverable	Claudio Casetti (POLITO), Thanassis Giannetsos (UBITECH)
v1.2	03.07.2023	Integration of review comments and final version	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v1.2	04.07.2023	Release of the deliverable to the consortium for final approval	Artur Hermann, Frank Kargl, Nataša Trkulja (UULM)
v1.2	10.07.2023	Submission of the deliverable to the EC	Nicole Mitsche (TEC)
v1.2.1	04.04.2024	Update of EU disclaimer and EU logo	Artur Hermann (UULM)

Chapter 1

Introduction

1.1 Towards Dynamic Trust Assessment in Future CCAM Services

Trustworthiness and trust relationships are central to security and safety of CCAM systems. Being highly distributed and cooperative systems, the different entities in a CCAM System-of-Systems (SoS) depend on the correct provisioning of services or input data from other system entities. This dependency constitute a trust relationship, where a trustor needs to rely on a trustee in order to provide its own service. Following the zero trust paradigm, such trust should not be blindly assumed, but rather be based on trustworthiness evidence and analysis. Enabling systems to model and assess such trustworthiness and take informed trust decisions is at the core of CONNECT and fundamental to reliable and resilient CCAM systems. Failure to recognize where trust assumptions fail will make CCAM systems vulnerable to failures or malicious attacks, and therefore undermine the public acceptance of CCAM.

In CONNECT, we approach this problem by investigating the concept of trust assessment and sources of trustworthiness evidence leading to a trust assessment framework (TAF), which allow us to make CCAM systems more trustworthy, more resilient, and more safe. Our design is such that the TAF is generic and can operate using a broad choice of trust sources based, among others, on observation-based mechanisms, like misbehaviour detection, or the use of evidence on integrity and correctness of the device design, configuration, and execution, provided through either software-based or hardware-based solutions. Nonetheless, the TAF does not have any restrictions on the type of Root-of-Trust that needs to be present to support such specific sources.

WP3 is the core work-package that investigates how trust is assessed and managed within the CONNECT architecture. Its main result will be a trust assessment framework (TAF) that will enable the applications in the use-case to model trust relationships, analyze trustworthiness during design and run-time, and take trust decisions based on provided trustworthiness evidence.

As we argue in Chapter 5, Subjective Logic will be the main formalism that our TAF builds on. This is due to its inherent capability to model trust networks and to also consider uncertainty and reason under incomplete evidence, which makes it the perfect tool for our effort. This deliverable, therefore, also introduces major concepts of Subjective Logic and describes their applicability to CCAM trust models and trust assessment.

As we detail in our overall roadmap for WP3 in Chapter 3, this Deliverable provides an overall introduction into the topic of trust modeling in the automotive domain and its state of the art. We also describe and define all high-level concepts and terminology used within WP3 and CONNECT

to develop and define our TAF. In addition, we specify the roadmap along which the TAF will be specified and implemented. Finally, we specify the high-level architecture of the standalone TAF that serves as the basis for later implementation and refinement of the standalone TAF and its extension towards steps 2 (federated TAF) and 3 (TAF-Digital Twin). Please refer to Chapter 3 for details.

1.2 Relation to other WPs and Deliverables

With the documentation of the research road-map towards the design of a holistic Trust Assessment Framework (TAF), capable of quantifying the trustworthiness of individual CCAM entities, this deliverable (D3.1) guides the development of the components that must be delivered as part of the CONNECT TAF. It provides the **definition of those fundamental building blocks that need to work in concert for enabling the trusted interactions between all actors in the CCAM ecosystem**; from the vehicles to the MEC infrastructure and to the cloud in order to guarantee a safe and secure operation of vehicles. Starting from in-vehicle components, interacting for collaboratively executing safety-critical functions, to vehicles needing to engage into a trustful data exchange with the infrastructure for endowing greater safety and efficiency, in the remaining of this deliverable we put forth all the **requirements and functional specifications** that such a trust assessment mechanism needs to achieve for allowing the establishment of a high level of trust into received data and the functions that rely on this data to take critical driving decisions.

In this context, Figure 1.1 depicts the direct and indirect relationships of D3.1 to the other Tasks and Work Packages (WPs). First, it distils all important properties that can impact the trustworthiness level of an entity and need to be taken into consideration when making such trust decisions. For instance, being trustworthy can encapsulate different things: (i) a vehicle reacting to monitored (kinematic) data with a high degree of precision, (ii) a vehicle reporting its state (e.g., position, velocity) with consistent accuracy, (iii) a vehicle (or a comprising ECU) not being malicious, i.e., not purposefully engaging in behaviour that can endanger system security and safety goals. All these behaviours correspond to different properties that need to be exhibited by an entity in order to attest to its level of trustworthiness. **D3.1 documents a first list of trust properties and elaborates on how they can be instantiated in the context of the envisioned use cases**; i.e., "*Intersection Movement Assistance*", "*Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (C-ACC)*", and "*Slow Traffic Movement Detection*". This classification of trust relationships, together with the detailed definition of the use cases functionalities (as documented in D2.1 [10]), will drive the technical work of the project and steer the evaluation and validation process to be conducted in WP6.

Within WP3, this deliverable also puts forth a detailed planning of the different trust assessment variants that will be investigated: Starting from the foundational view of a **Standalone Trust Assessment Framework (TAF)**, capable of analysing the trustworthiness level of individual entities in isolation; to its extension to a **Federated Mode of Operation** for accounting additional sources of trust that may stem from communicating entities; to the construction of a universal **TAF, comprising agents deployed throughout the entire CCAM ecosystem (including other vehicles as well as the MEC) and the instantiated Digital Twins, where trust calculations can be offloaded for supporting the processing of the abundance of data towards creating more accurate atomic opinions on the level of trust of consumed data, services and neighbouring vehicles**. Therefore, D3.1 provides details on the early version of the Standalone TAF and serves as the core of how the internal building blocks will be operating so that extended versions can be released in the subsequent deliverables, D3.2 and D3.3, respectively (in M18 and M30).

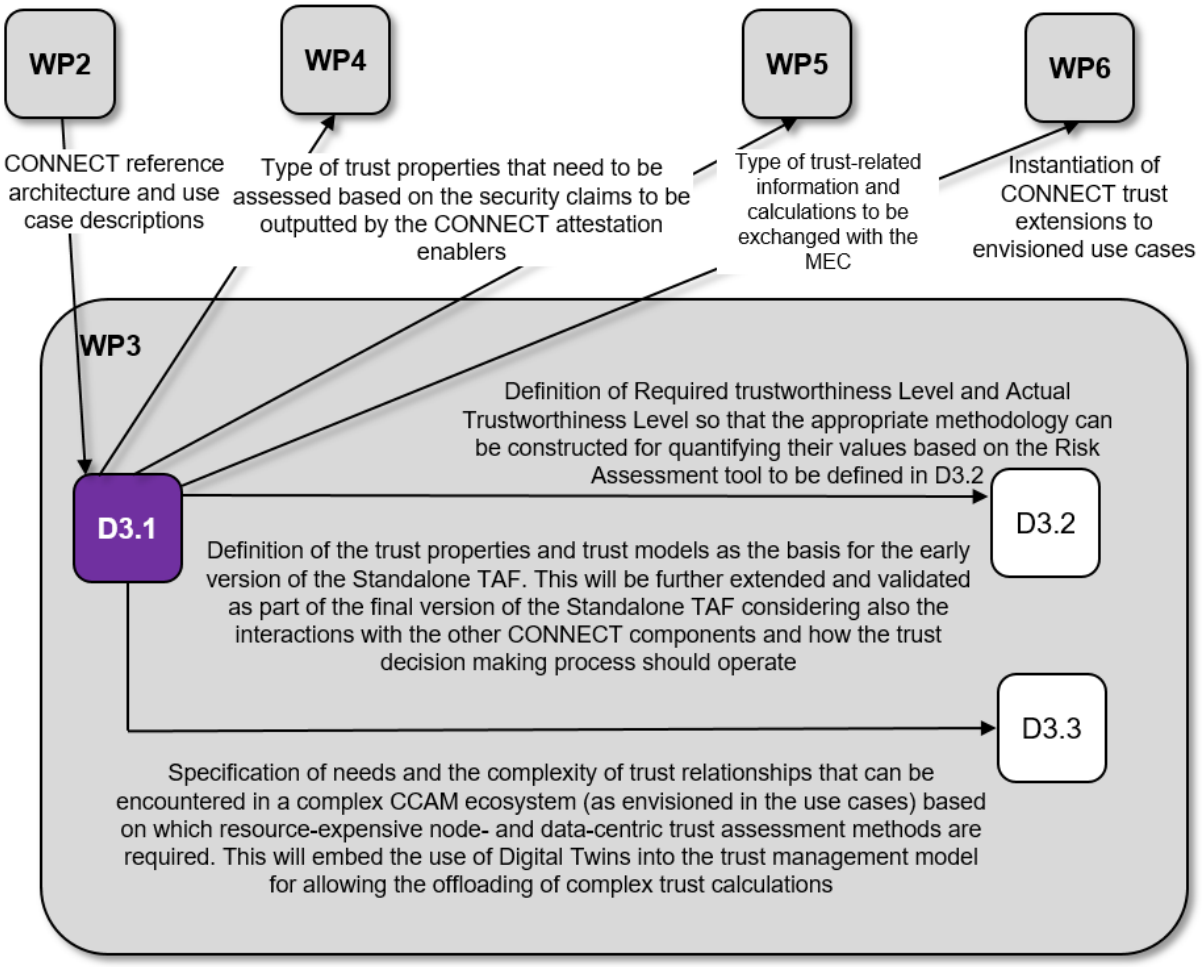


Figure 1.1: Relation of D3.1 with other WPs and Deliverables

In addition, given the CONNECT architectural components (defined in D2.1 [10]), D3.1 also exposes the interaction flows among the internal TAF building blocks and the interfaces needed for enabling the secure communication with the other CONNECT components in order to drive the technical work of WP3 – WP5 and provide a solid basis for the establishment of a trust-aware decision-making framework that uses the quantified trust values towards enabling C-ITS stations to assess the trust level of their neighbouring stations and the received data with the desired trade-off between safety and efficiency.

Overall, the outcome of this deliverable serves as the first milestone of CONNECT's efforts towards the design of an overarching trust assessment framework by consolidating the core architecture of how such a mechanism should be modelled to be able to cope with the complex set of trust relationships that need to be assessed in the CCAM ecosystem. Two main building blocks for this trust architecture are the principles of **Zero Trust**, where no initial trust between entities is assumed but all trust has to be based on collected trustworthiness evidence, and the use of **Subjective Logic** as a powerful reasoning framework that allows to reason about trust relationships under uncertainty.

1.3 Deliverable Structure

This deliverable is structured as follows: Chapter 2 describes definitions of terms related to the components of the Trust Assessment Framework (TAF) and its associated functionalities. Chapter 3 describes the road-map to be followed for the development of CONNECT's holistic Trust Assessment Framework (TAF) and the functions that each TAF variant will be focusing on. In Chapter 4 working definitions on Trust, Trustworthiness and related terms are specified to be able to scope those properties of interest based on which trust decisions will be made. This listing, although not exhaustive, captures all requirements of those trust relationships that comprise the envisioned use cases (cf. Chapter 6). In Chapter 5, a detailed State-of-the-Art analysis of trust management, in the context of autonomous vehicles and C-ITS, is provided. For this purpose, related work on trust assessment in the automotive domain, but also beyond, is described. Furthermore, requirements for trust assessment are provided, based on which the CONNECT TAF will be designed. Following these requirements and possible decision logic mechanisms that could be used in the context of trust management, it is explained why Subjective Logic was adopted (in CONNECT) as the core pillar behind the overarching trust reasoning framework that can deal with high uncertainty and contradicting evidence.

Continuing in this line of trust modelling, Chapter 6 proceeds with the description of a methodology to derive data- and node-centric trust relationships that need to be continuously assessed in the context of the envisioned use cases towards enhancing safety and improving the efficiency of the evaluated CCAM services. For the three evaluated use cases (i.e., "*Intersection Movement Assistance*", "*Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (C-ACC)*", and "*Slow Traffic Movement Detection*") exemplary data- and node-centric trust relationships are described which need to be evaluated. This is also accompanied by a detailed listing of those evidence that need to be provided (in a verifiable manner) by the deployed CONNECT security mechanisms (e.g., Misbehavior Detection, Trusted Computing & Attestation) so that trust can be assessed in the context of these trust relationships.

Chapter 7 then constitutes the heart of this work and details on the reference architecture and foundational view of the Standalone TAF and all its internal building blocks. More specifically, the functional specification of the TAF is described, showing how the assessed service interacts with the TAF and the other CONNECT components. Chapter 8 proceeds with the compilation of an evaluation methodology, putting forth those properties of interest that will drive the benchmarking of the TAF itself when investigated in the context of the envisioned use cases. The endmost goal is to present a detailed benchmarking in an attempt to consolidate whether the deployment of such complex trust assessment methodologies can benefit the decision-making process of the system towards greater safety for the users. Finally, Chapter 9 provides a summary and concludes the deliverable.

Chapter 2

Trust Management Terms and Definitions

This chapter summarizes terms and definitions of those terms related to trust management, the Trust Assessment Framework (TAF) and its associated functionalities along with a selected set of its characteristics capturing those important specifications that enable the **characterization and quantification of trust in complex environments such as the ones envisioned in CCAM ecosystems with potentially high level of uncertainty in the trustworthiness of exchanged data**. This vocabulary will act as the reference resource, throughout all project activities, so that all stakeholders can have a common understanding of the characteristics that could be used to describe the trustworthiness of a data item or node as well as the methodology and concepts to be followed for allowing stakeholders to make a judgment if a service, function or entity can meet the required expectations (based on the trustworthiness definition as documented in Chapter 5).

While this vocabulary primarily targets documentation of terms capturing the mode of operation of a TAF in the context of Autonomous Vehicles (AVs), it is intended for use horizontally in the information technology domain and all domains where Subjective Logic is used as the foundation of trust reasoning.

ATL (Actual Trustworthiness Level) The ATL reflects the result of an evaluation of a specific (atomic or complex) proposition for a specific scope provided by the TLEE. It quantifies the extent to which a certain node or data can be considered trustworthy based on the available evidence.

ATO (Atomic Trust Opinion) An ATO is a subjective logic opinion created by the TSM when quantifying one specific type of a Trust Source based on trustworthiness evidence. An ATO is formed by a Trust Source in the context of a single trust relationship.

Component Diagram A component diagram is a graph whose vertices represent components necessary to realize a concrete system function, and whose edges represent communication links between these components. A component diagram is used to design function-specific trust models.

Data-centric Trust Data-centric trust is evaluated in the context of a data-centric trust relationship. In this trust relationship, trust is evaluated from one node to data, where the trustor (the one who trusts) is a node and the trustee (the one who is trusted) is data.

Functional (direct) Trust Functional (also called direct) trust is evaluated in the context of a functional trust relationship. In this trust relationship, trust is evaluated between two trust objects have a direct relation, i. e., the trustor has a direct observation of the trustee and

where the trustor engages in a functional relationship with the trustee within the system model like receiving a data item from the trustee.

Node-centric Trust Node-centric trust is evaluated in the context of a node-centric trust relationship. In this trust relationship, trust is evaluated from one node to another node so that the trustor (the one who trusts) and the trustee (the one who is trusted) are both nodes.

PROP (Proposition) A *proposition* is a logic statement about some phenomenon of interest whose level of trustworthiness we are interested in assessing. A proposition could be 1) atomic—a proposition whose truth or trustworthiness can be directly assessed, or 2) composite, comprising of multiple atomic propositions. The proposition describes the fulfillment of the properties in relation to *data* or *nodes*.

Referral Trust Referral trust is evaluated in the context of a referral trust relationship. In this trust relationship, trust is evaluated between two trust objects that do not have a direct relation but the trustor has a direct relationship with another intermediate node(s) that have a direct observation of the trustee.

RTL (Required Trustworthiness Level) The RTL reflects the amount of trustworthiness of a node or data that an application considers required in order to characterize this object as trusted and rely on its output during its execution.

TA (Trust Assessment) The TA is a component inside the TAF which orchestrates the overall process of trustworthiness level evaluation and trust decision taking.

TAF (Trust Assessment Framework) A software framework which, given a trust model for a specific function running inside a CCAM system, is able to evaluate trust sources for trustworthiness evidence and evaluate propositions within the trust model to obtain their ATLs. Optionally, also an RTL can be evaluated and trust decisions can be taken and communicated to the application.

TAF-API (TAF - Application Programming Interface) Application Programming Interface by which the TAF and its functionality can be accessed from an application.

TAF-DT (TAF - Digital Twin) The digital twin allows a vehicle to replicate its TAF including among others its TMs and TSs within a MEC. This allows a vehicle to outsource trust assessment to a MEC where the TAF-DT is expected to run inside a TEE so that confidentiality and integrity of its data and state can be protected from the MEC.

TDE (Trust Decision Engine) The TDE is a component inside the TAF which performs the last step before an output is provided to the application that requested trustworthiness assessment. The TDE either forwards the Actual Trustworthiness Level (ATL) calculated by the TLEE along to the application or outputs a Trust Decision (TD). A TD is created after comparing the ATL to the Required Trust Level (RTL) in a predetermined manner. Whether the output of the TAF is an ATL or a TD depends on the needs of the application requesting trustworthiness assessment.

TLEE (Trustworthiness Level Expression Engine) The TLEE is a component inside the TAF that calculates the level of trustworthiness for a concrete trust model and the proposition that needs to be evaluated. The TLEE uses the numerical values of the Atomic Trust Opinions

computed based on the trust sources collected in the TSM. Based on these inputs, the TLEE calculates an ATL and provides it to the TA. The TLEE encapsulates most of the Subjective Logic formalism.

TM (Trust Model) Trust model is a graph-based model which is built on top of a system model which represents all components and data needed to perform a certain function. Components represented either create, transmit, process, relay, and receive the data used as input to a function. The vertices in a trust model correspond to an abstraction called trust objects, and the edges in a trust model correspond to trust relationships between a pair of trust objects. The trust model also encompasses a list of trust sources used to build up / quantify trust relationships by providing atomic trust opinions. The trust model is a main input to the TMM and the TLEE. Since trust is a directional relationship between two trust objects and it is always in relation to a concrete property or scope, then as part of the trust model, there can be multiple trust relationships between the same two trust objects, depending on different properties of the trust relationship, or the scope of the trust relationship.

TMM (Trust Model Manager) The TMM is a component inside the TAF responsible for storing trust models and making them accessible for TLEE and other purposes. In particular, it is able to provide TMs for specific functions running in a CCAM system, also considering different scopes that TMs may cover.

Trust Objects Trust objects are core building blocks of a trust model. They represent entities that assess trust or for which trust is assessed. The trust objects are identified 1) based on the *components* from the component diagram and 2) the *atomic propositions* (i.e., the properties about data or nodes for which trust assessment is conducted).

Trust Relationships A trust relationship is a directional relationship between two trust objects that are called trustor (i. e., the “thinking entity”, the assessor) and a trustee (one who is trusted). The trust relationship is always in relation to a concrete property and a certain scope.

Trustworthiness Tier Trustworthiness Tier is a categorization of the levels of trustworthiness which may be assigned by the TAF (or another Verifier that appraises the attestation results of a TC and communicates them to the TAF) to a specific Trustworthiness Claim.

Trustworthiness Claims A Trustworthiness Claim (TC) is a form of node-centric ATO provided by a TS and contains a specific data quote used for conveying the information needed by the TAF to make a decision on the trust level of an object. The TC is usually produced (by the Attester) so as to provide trustworthiness evidence (cf. “Trust Source”) that can be used for appraising the trustworthiness level of the Attester in a *measurable* and *verifiable* manner [38]. Measurable reflects the ability of the TAF to assess an attribute of the Attester against a pre-defined metric 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**. For a more detailed list of possible system (behavioural) evidence that can be appraised as part of TCs, please refer to Section 6.3.

TS (Trust Source) A TS manages one or multiple trustworthiness evidence inside the TAF. On request of the TA, it quantifies the trustworthiness of a trustee based on a specific type of evidence in form of an atomic trust opinion.

TSM (Trust Sources Manager) The TSM is a component inside the TAF responsible for handling all available TS inside a TAF and to establish and integrate new TSs dynamically through a plugin interface.

Chapter 3

Approach and Road-Map for CONNECT Overarching Trust Assessment Framework

The CONNECT Trust Assessment Framework is a highly complex and feature-rich system that will be the core artefact of the WP3 activities towards the design of a reasoning framework that allows an entity to appraise the Actual Trust Level of a (data and/or node) object even in environments with high uncertainty. Considering the highly complex ecosystems envisioned in CONNECT, consisting not only of multiple vehicles comprising thousands of ECUs and sensors but also infrastructure entities from Road Side Units (RSUs), Multi-Access Edge Computing (MEC[1]) Service Providers and the Cloud, it is of paramount importance to include **trustworthiness management as a functional component of any safety-critical application**: *Assess dynamic trust relationships and define adequate trust models based on which involved entities can establish trust for cooperatively executing safety-critical functions*.

Focusing on the next generation of Intelligent Transport Systems (ITS) enabling Connected Cooperative Automated Mobility (CCAM), CONNECT establishes such a trust reasoning framework rooted in the zero-trust concept for bootstrapping vertical trust from the vehicle up to the MEC and cloud environment. This trust management framework (Chapter 7) can provide the formalism to describe the functional trust relationships between entities involved in a cooperative task (different vehicles on the road; different on-board components on the same vehicle); and to dynamically assess the strength of those trust relationships, based on the exchanged data. Such an approach makes it, for instance, possible to safely combine the vehicle's systems with information available in the MEC, thus expanding the knowledge on the environment required for decision-making, allowing outsourcing of the calculations in a trustworthy way to the backend, and ultimately helping to make faster decisions, cooperatively and without delay, increasing the safety of CCAM.

In this context, to design its architecture and implement and evaluate prototypes, as aforementioned, we take a **three-step approach** that will allow a continuous enhancement and also limits the complexity in each step. This follows the idea of agile development instead of classical waterfall models. However, specifying and designing the full architecture in one step is, from our perspective, too error-prone and should therefore be avoided.

Still, the finally envisioned outcome needs to be clear from the start, to avoid possible later incompatibilities. In what follows, we therefore detail each of the three steps and the required functionality in each of them. Those steps are as follows:

[1] Throughout this document we use the (well-known) term MEC for the sake of a clear -to the reader- reference. What we mean is in-general any edge computing infrastructure in the user vicinity hosting trust/security/CCAM applications, even tolerating reasonable technical diversion from the typical MEC specification proposed by ETSI in (https://www.etsi.org/committee/1425-mec)

Step 1: Standalone TAF A standalone TAF can operate, for example, within a vehicle or a MEC. While it is able to evaluate trust also on remote entities and data, it will do so solely from its own perspective and based on internal assessment and will not cooperate with other TAFs.

Step 2: Federated TAFs In this step, we add the capability to federate and exchange information among TAFs. This includes exchange of trust models and opinions and establishing an overall understanding of a combined trust model within a large CCAM "Systems-of-Systems".

Step 3: Digital Twin for Trust Assessment In the final step, we will enable vehicles (or other nodes) to outsource its trust model and TAF state to an external party (typically a MEC) where reasoning on trust models can be executed on behalf of the original TAF.

Results of these three steps will be designed and reported in the WP3 deliverables as outlined in Sections 3.1 to 3.3.

We should emphasize here that all TAF variants are able to cope with a variety of trust sources in a manner that remains agnostic to the underlying element that provides this evidence.

3.1 Step 1: Standalone Trust Assessment Framework (TAF)

This is what this document is concerned with, so the full high-level architecture of a Standalone TAF is reported here in D3.1 and is due on M10. A proof-of-concept implementation of a Standalone TAF on which use-cases can be build and evaluated upon (cf. Section 3.4) is foreseen due to M18 as part of D3.2 [11]. In that deliverable, details on the fine-grained architecture and description of the prototype will be put forth showcasing how all internal building blocks, catalogued in Chapter 7, will be working in tandem towards enabling CCAM entities to assess the trust level of its communicating nodes and received data.

Towards this direction, the core functionality of a Standalone TAF is summarized below:

- ✓ to establish and manage a trust model of **varying complexity based on the types of trust relationships** that need to be captured - considering both **data- and node-centric** relationships where trust opinions need to be calculated in relation to a property (of a piece of) data or a node, respectively. A detailed list of those core trust relationships that need to be assessed per use case is described in Section 6.3;
- ✓ to determine atomic **trust opinions** on data and nodes via a variable and dynamic number of trust sources including trustworthiness evidence on system and behavioural properties such as *integrity, safety, authenticity, etc.* For a rather complete list of such relevant properties based on which trust evaluation will be conducted by the TAF please refer to Section 4.5;
- ✓ to evaluate **propositions** within a trust model; and
- ✓ to calculate an ATL and compare it to a static RTL towards accommodating trust-aware decisions making strategies and achieve greater safety and better performance.

The high-level architecture for the Standalone TAF described herein includes:

- ✓ the description of the inner component structure and mechanisms of the TAF, including the use of **subjective logic for trustworthiness level evaluation** capable of coping with complex environments trust opinions might need to be constructed with high degree of *uncertainty* and based on *contradicting evidence*;
- ✓ a list of trust sources that the TAF should support (Section 6.3).

What is left out (as this will be specified in the fine-grained architecture) are details of the inner-workings of the TAF, including

- the internal APIs of the TAF;
- the external APIs that need to be exposed for enabling the necessary interactions with the triggering application (wishing to calculate a trust opinion of a data or node object) and the other CONNECT components based on the Reference Architecture, as defined in D2.1 [10]. Especially, as it pertains to the communication with the CONNECT Attestation Integrity Verification Component that is responsible for providing the verifiable trustworthiness evidence conveying information system properties including integrity, safety, communication integrity, etc. (cf. definition of "Trustworthiness Claims" in Chapter 2);
- the inner workings of the components, and
- the details how atomic opinions in specific trust sources are derived.

While these are already being worked on, their details might change for the prototype, as they are also developed and refined alongside our analysis of the use-cases. Such details will then be included in the fine-grained architecture in D3.2 [11].

3.2 Step 2: Federated Trust Assessment Framework (TAF)

In a next step, the **Standalone TAF will be extended to be able to establish links to other TAFs** (instantiated in other ECUs within the same vehicle; other vehicles or road-side infrastructure; or even the Digital Twin of the vehicle executed in a secure container on the MEC) and evaluate propositions that span trust models from different TAFs (or, as an alternative, exchange of partial trust models that can combined and evaluated within a single TAF).

This will be reported in D3.2 [11] which is due on M18. A prototypical implementation of a federated TAF on which use-cases will be evaluated upon, based on the detailed planning put forth in Section 3.4 and further elaborated in Chapter 8, is scheduled for the second experimentation round of the envisioned use cases to be finalized by M33.

The core functionality of a federated TAF is:

- ✓ to **establish and communicate with other entities in the evaluation of propositions that span multiple TAFs**. *We have to highlight here that the TAF, instantiated in various places of the entire CCAM ecosystem, share the same design principles while they will be loaded with different trust models based on the type of function, data or node that they want to create a proposition for*;

- ✓ to **protect this communication from external manipulation**. One important design property of the overarching TAF is its resilience against attacks that both try to affect its **operational assurance** (hence, the wrapping and execution of each TAF as part of a secure enclave [9]) but also its **accuracy** - by altering the (input) trustworthiness evidence based on which the trust assessment will take place;
- ✓ to access **remote trust sources**; and
- ✓ to **assess and consider the trustworthiness of a remote TAF in such a federation**. The endmost goal in such an architecture is the creation of a "*Chain of Trust*" [1], comprising all trust assessment components, to make sure that a provided trust proposition (by one TAF) is trustworthy prior to been fused with all other trust proposition values. This will require to establish and maintain trust relationships between all TAFs. These should be bidirectional trust relationships. These trust relationships should also be checked as required, as trust should generally be expected to decay over time unless further checks are made. Among other properties, one key assurance that we need to establish is this of the authenticity of all TAF instances to avoid attacks on the TAF and/or the created trust models (Chapter 8).

The Federated TAF architecture that will be described in D3.2 [11] will include:

- the definition of the APIs and protocols by which two TAFs can interact;
- the description of appropriate security safeguards to protect federated TAFs from external manipulation (as offered through the CONNECT TEE Guard and the support of the secure containers that will be leveraged for safely executing the TAF instances); and
- a discussion on the impact of federation on trustworthiness level evaluation.

3.3 Step 3: Trust Assessment Framework - Digital Twin (TAF-DT)

In a final step, the trust management will be extended to support also the **consideration of a virtual representation of an entity (Vehicle), classified as Digital Twin,** as a supporting factor where **resource-expensive trust calculations (that need to be performed by the Vehicle TAF) might be offloaded** to formulate collective trust calculations and trust-aware decision-making strategies for achieving better performance. To this end, a standalone TAF should be able to replicate and share its current structure and state between the two TAFs, running inside the deployed Trusted Execution Environments (as part of the secure containers). This process will be facilitated through the CONNECT TEE Guard and more specifically through the *secure migration* functionality offering the capability for *securely exporting, migrating (machine-to-machine) and importing* application-level states from one secure enclave to another.

A first concept with a high-level architecture for a TAF-DT will be also described in D3.2 [11], due on M18, and a fine-grained architecture will be specified in D3.3 [12] due on M30. A prototypical implementation of a TAF-DT on which use-cases can be build is again foreseen to be finalized and evaluated in the context of the second experimentation round of the envisioned use cases scheduled for M33.

The core functionality of a TAF-DT is summarized below:

- ✓ to replicate and share the structure and state between TAF instances deployed in a physical entity (Vehicle) and its Digital Twin (as part of the MEC service landscape). As a prerequisite, this process requires the **establishment and management of an inherent trust relationship between the entity (Vehicle loaded with the original TAF instance) and its Digital Twin**. This trust relationship is one of the key relationships that the entity (with the original TAF) has, as the TAF on the Digital Twin is generally expected to provide *transitive trust services* for the original entity with other objects. If trust in this relationship fails, the assurance that the original entity can hold in other trust relationships cannot be easily renewed, and should be expected to decay, as there is no easy way to maintain it;
- ✓ to keep the (MEC-deployed) **TAF-DT synchronized to the state of the original TAF** as events occur and atomic opinions or structures of trust models change and evolve over time;
- ✓ to authorize a MEC to query a TAF-DT on behalf of the original vehicle as part of an evaluation of a proposition (*Delegation of Trust*); and
- ✓ to manage the **hand-over of an original vehicle's data from one (MEC) Service Provider (SP) to another** whenever the connection quality starts to deteriorate due to physical distance or overloading [23]. This will also necessitate the hand-over of the Digital Twin (of the Vehicle) which, in turn, will also trigger the migration of the TAF-DT. CONNECT will investigate the definition of the optimal strategy for capturing all the requirements of such a hand-over scenario: both as it pertains to *performance* so that the hand-over strategy causes the least frequent violations of round-trip measurements (as this is a vital aspect of safety standards for automotive applications) but also *trust concerns* when trying to establish trust between different MEC SPs.

MEC applications offered by different Application Servers (ASs) are usually grouped together into different trust domains, adhering to varying security requirements. This translates to different sets of security assurances that need to be provided by a vehicle that wants to consume such MEC applications offered by different providers; or different types of assurance when migrating one object (TAF) between two different ASs. Indeed, in principle, *a TAF instantiated in AS1, for example, needs to be considered (by AS2) as possibly misconfigured (running in a "hostile" environment offered by AS1), so that it needs to be carefully assessed before being migrated and leveraged in AS2*. In CONNECT, such security assurances will be offered through the integration of trusted computing mechanisms (i.e., remote attestation). These security assurances (based on the attestation of a computing platform's integrity) can directly translate into trust in a vehicle's capability to protect its information and functional assets (MEC applications) against varying sets of threats.

The TAF-DT architecture that will be described in D3.2 and D3.3 will include

- the definition of the API and protocol by which an original TAF and its TAF-DT can interact;
- the description of appropriate security safeguards (in particular a TEE) to protect the TAF-DT from external manipulation; and
- a model and mechanism how the TAF-DT remains synchronized to the original TAF.

3.4 Demonstration & Evaluation Plan

In each of the aforementioned steps, we will design and describe a high-level architecture documenting all internal building blocks in the subsequent WP3 deliverables, as described in Section 1.2. This high-level reference architecture will then be refined and implemented into a fine-grained components palette reflecting the prototype that will be integrated and evaluated in the context of the envisioned use cases; i.e., "*Intersection Movement Assistance*" (IMA), "*Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (C-ACC)*" (VRU-CACC), and "*Slow Traffic Movement Detection*" (SMTD). The endmost goal is not only to demonstrate the **feasibility and applicability** of such complex trust frameworks, by applying it in three heterogeneous use cases with different requirements on the type of trust relationships to be established, but also to conduct a detailed **benchmarking of all features so as to evaluate both the technical merits and constraints but also to assess whether such methodologies can benefit trust-aware decision-making strategies for achieving greater safety while minimizing the performance overhead**. We note that the actual evaluation within the use-cases is not part of WP3, but will be investigated as part of WP6.

- **Technical Performance Evaluation:** As detailed in Chapter 8, one core property of interest is the *run-time performance* of the trust framework to validate that it can support the timing constraints of CCAM safety-critical functions like the ones met in IMA and VRU CACC (*strict time constraints*) and in a less degree in the context of SMTD. This goes beyond the calculation of atomic trust opinions, based only on the assessment of the (local) Standalone TAF, but should also consider the case where TAF agents are deployed over multiple entities - covering the entire spectrum from resource-capable ECUs (in-vehicle components), to the Vehicle Computer (In-Vehicle Onboard Unit) and to the MEC. The motivation would be to appraise whether this mode of operation enabling collective trust calculations and trust-aware decision-making strategies can achieve better (or equivalent) performance to the Standalone TAF but for more complex trust models comprising a multitude of data- and node-centric trust relationships to be assessed.

- **Safety Impact Evaluation:** Another core hypothesis that we want to evaluate is that the integration of such complex trust assessment mechanisms can benefit the overall safety profile of CCAM ecosystems and enable the transition to Day 3 phase of automated driving leveraging trusted exchange of data for making safety-critical decisions. An example scenario can be considered in the context of the VRU-CACC use case where (data-centric) trust decisions on the *integrity* of the extracted kinematic data can solicit a vehicle's distance calculation (to other proceeding vehicles) and acceleration which, in turn, dictates its decision whether it needs to stop or decelerate to avoid a crash. *Having such trust propositions on the trustworthiness of the data, do they help the system to make more accurate decisions? How this translates to CCAM functionalities that leverage different types of data originating from different sources?*

- **Robustness & Resilience to Attacks:** Finally, it is also important to evaluate the resilience of the trust management in coping with attacks against its operation but also with threat models that try to provide wrong input so as to manipulate the type of fusing operator used. Even in such extreme scenarions, the TAF should be able to output meaningful information that can help the trust decision-making process (Chapter 8).

In the following table (Table 3.1), we summarize the main vision on the instantiation and evaluation of the previously described TAF variants in the context of the envisioned use cases. As

aforementioned, to better evaluate the various features of the designed TAF, we have selected the deployment of only those modes of operation that best fit the complexity and timing constraints and needs of each use case environment. This will not only allow a more complete benchmarking of the TAF, but will provide the necessary evidence on the applicability of such a solution in heterogeneous ecosystems.

The evaluation will expand over two experimentation periods as part of the overall CONNECT Framework [10] to be able to identify and resolve any issues and glitches that may arise during the evaluation (over the first experimentation round) of the Standalone TAF that provides the basis of also the Federated TAF and the TAF-DT.

Table 3.1: Instantiating Trust Assessment Framework Variants in the CONNECT Use Cases

Use Case	Instantiation of TAF Variants
Intersection Movement Assistance (IMA)	**Standalone TAF:** In the context of the IMA UC (where the focus is on timely alerting the driver of potential collisions with other vehicles), the Standalone TAF will be deployed as part of the main In-Vehicle Computer (VC) to assess the trustworthiness of both the (locally) extracted kinematic data but also those data from the CAM/CPM messages received from neighbouring vehicles. This information (mainly focusing on data-centric trust relationships) will enable the IMA application to access trustworthy information from its local dynamic map on any surrounding objects' dynamics in the intersection. Another interesting feature here is the richness of the trust sources to be considered and evaluated since besides evidence on the *integrity* of the kinematic data, the Standalone TAF will also base its trust decision on the output of the Misbehaviour Detection service for the *correctness* of the data. **Federated TAF:** In the context of evaluating the Federated TAF, additional TAF agents will be deployed (especially on the MEC) for enabling the (data-centric) trust assessment based on additional information on the trustworthiness level (node-centric) of the vehicle transmitting CAM/CPM messages that are used by neighbouring vehicles. Such trust opinions will be calculated by a TAF, that is instantiated on the MEC, and will be conveyed to the TAF agents of requesting vehicles which will then fuse their local observations with the trust recommendations stemming from the MEC, based also on the trust level of the MEC infrastructure itself. **TAF-DT:** Not considered in this use case
Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (VRU-CACC)	**Standalone TAF:** The Cooperative Adaptive Cruise Control (C-ACC) is a driving assistance system that allows the vehicle to automatically keep a safe distance to other vehicles based on its own sensors' data and steering data coming from other vehicles on the road and road-side infrastructure information. In the context of this scenario, the focus is mainly on the in-vehicle trust model, and the Standalone TAF will again be instantiated as part of the main In-Vehicle Computer for been able to assess the trustworthiness of both the (locally) extracted kinematic data but also those data from the CAM/CPM messages received from neighbouring vehicles. As a first step, the Standalone TAF will construct trust opinions on the trust level of the monitored data without considering the trustworthiness of the environments where such data were extracted. **Federated TAF:** As a next step, TAF agents will be deployed in resource-capable Electronic Control Units (ECUs) for additionally calculating trust opinions on the *integrity* and *operational assurance* of the sensor that was responsible for extracting, collecting and parsing the C-ACC related data in a trustworthy manner. All these trust opinions will then be sent to the main TAF, as part of the In-Vehicle Computing, for performing the final fusion operation. **TAF-DT:** Not considered in this use case

Slow Traffic Movement Detection (SMTD)	**Standalone TAF:** This use case mainly relies on the establishment of data-centric trust relationships for assessing the trustworthiness of real-time kinematic data received by active vehicles (through CAM/CPM messages) towards the detection and identification of a slow-moving traffic condition of the road. In this scenario, the Standalone TAF will again be instantiated as part of the main In-Vehicle Computer for been able to assess the trustworthiness of the local environments where the kinematic data were extracted. The focus here is on constructing trust opinions on the kinematic data of the transmitting vehicle to share this trust level with the Traffic Control Center in the backend. **Federated TAF**: In this scenario, the (local) TAF agents of the transmitting vehicles share their trust opinions on their kinematic data with the (main) TAF agent been instantiated on the MEC. This way, the main TAF will be able to construct a global map of slow-moving vehicles based on data with a high enough trust level. This process will again be based on *trust discounting* as well as *fusing various trust opinions*. **TAF-DT:** As a final step, in the context of this use case, each vehicle will also be represented as a Digital Twin been instantiated on the MEC. Considering the scarcity of resources of many in-vehicle ECUs where a trust assessment process might need to run, resource-expensive security and trust calculations will be offloaded on the respective Digital Twin. Examples of such calculations will mainly target the provision of trustworthiness evidence on the operational assurance of the vehicle that provided the data; i.e., offload the verification and signing of control-flow attestation evidence [9] that can provide assurances that the device's software (that extracted the data) has not been compromised (system integrity). This is a rather resource-intensive process, and the goal is to showcase how collective trust-aware decisions can be enabled through such TAF agents.

Chapter 4

General Principles of Trust and Trustworthiness

As part of this chapter, we first summarize our definitions on Trustworthiness and Trust, which we need as a basis for the later descriptions of the Trust Assessment Framework (TAF). Here, we provide detailed definitions of terminology beyond the rather brief glossary of Chapter 2.
Concretely, in Section 4.1, we first introduce and define other related terms relevant to the definitions of Trustworthiness and Trust, followed up with a taxonomy of trust relationships in Section 4.2. We define Trustworthiness and Trust in Sections 4.3 and 4.4, respectively. Finally, in Section 4.5, we discuss trustworthiness properties in CCAM systems.

4.1 Definitions of other Related Terms

Propositions. A proposition is a logical statement about some phenomenon of interest (i.e., a variable) whose level of trustworthiness we are interested in assessing. The proposition describes the fulfilment of a certain property of data or a node. A proposition could be 1) atomic—a proposition whose truth or trustworthiness can be directly assessed or verified through some evidence (from one of more trust sources), or 2) composite, consisting of multiple atomic propositions.

Trust objects. Trust objects are entities that assess trust or for which trust is assessed. The trust objects are identified

1) based on entities or components from the concrete system under consideration. For example, in different use cases from an automotive domain, these entities could be vehicles, ECUs, Zonal Controllers (ZC), MEC, etc. Throughout our definitions we refer to them as **nodes**; and,

2) atomic propositions, related to the fulfilment of the certain properties of data (e.g., camera data has not been compromised) or nodes (e.g., vehicle A reports its position accurately).

The trust objects are the main building blocks for trust relationships.

Trust relationship. Trust relationship is a directional relationship between two trust objects that can be called trustor and a trustee (one who is trusted), see Fig. 4.1. The trust relationship is always in relation to a concrete property.
The trustor is the “source” trust object as part of a trust relationship for which trust is assessed (one who trusts, the “thinking entity”, the assessor). Trustor can only be a node. The trustee is a

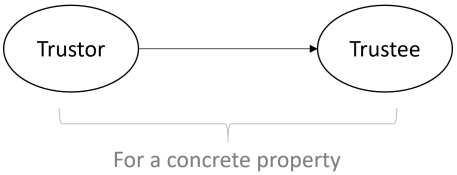


Figure 4.1: Trust relationship.

"sink" trust object as part of a trust relationship for which trust is assessed (one who is trusted). Trustees are propositions or statements about nodes or data.

Trust network. The trust network combines various trust relationships among different trust objects. Figure 4.2 shows an example of a trust network, where the same trust object (for example, trust object B), can be both a trustor (in B→D trust relationship), and a trustee (in A→B trust relationship). With red boxes we label the (atomic) propositions as trust objects as part of the trust network. The (atomic) propositions are always in the leaves of the trust networks, and are always trustees.

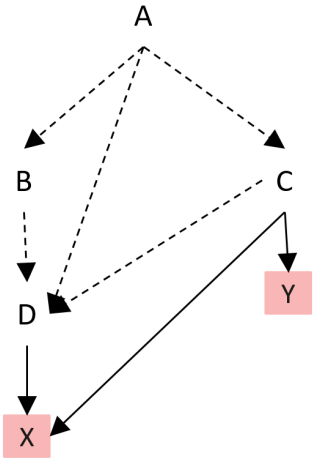


Figure 4.2: Trust network.

4.2 Taxonomy of Trust Relationships

In the following, we present a taxonomy for trust relationships in trust assessment, based

1) if the trustor has a direct observation on the trustee (e.g., A has a direct relationship with B), or if the trustor has a direct relationship with another intermediate node that has a direct observation on the trustee. For example, let's assume B is the trustor and X is the trustee. In this case, B does not have a direct observation of X; however, B has a (referral) relationship with D that has a direct relationship with X; and,

2) on the type of propositions (node-centric and data-centric).

Concretely, if we refer to the trust network example above, there are two types of trust relationships as part of the trust network that we mark with different arrows:

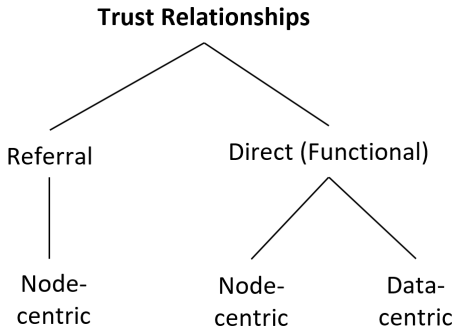


Figure 4.3: Trust relationships taxonomy.

1) dashed arrows that represent *referral trust relationships*, e.g., A→B, A→C, B→D. These relationships are always related to trustworthiness assessment on nodes (from a node to a node), and

2) solid arrows that represent *direct (also called functional) trust relationships*, e.g., D→X, C→X, C→Y, related to trustworthiness assessment of a proposition (from a node to a proposition). In this case, the nodes have a direct observation on a concrete proposition.

Additionally, as previously explained, the propositions are related to the fulfilment of the certain properties of data or nodes. As a result, based on the type of propositions, we differentiate between two types of direct trust relationships: *data-centric* and *node-centric*. As part of the data-centric trust relationship, the trustee is a proposition on (a piece of) data; whereas, in a node-centric trust relationship, the trustee is a proposition on a concrete node. Please note that the referral trust relationships are always node-centric.

4.3 Trustworthiness

Trustworthiness can be broadly conceived as a measure of the trustee's ability to achieve a specific task, as part of a trust relationship. In other words, the entrusted task here can be understood in terms of the expectations a trustor has from the trustee. Given this, we define Trustworthiness as **the likelihood of the trustee to fulfil trustor's expectations in a given context.**

Here, expectations could relate to the correctness of data, as well as the assessor's ability to assess the correctness of the data. However, expectations can also relate to the behaviour of the trustee (e.g., if the trustee is a node then in relation to the functionality of that node), the process through which the entrusted task was carried out by the trustee, and the purpose for which the task was chosen. In that case, we need to bridge the trustworthiness of data sources with expected behaviour, since there is nothing in the trustworthiness of data sources and data that would entail consistent behaviour. Different assessors (i.e., trustors) might have different rules on how to translate this to expected behaviour. So, we can do this bridging based decision-making, for example rules or policies that could support making decisions or calculations about expected behaviour related to the data that we have collected by different data sources. For example, one of the ways in which an autonomous vehicle can be trustworthy to a user or another vehicle is by fulfilling the certain expectations regarding safety (by driving safely and not causing accidents) and providing evidence to the user or the other vehicle regarding how such safety expectations will be met.

Additionally, let us suppose the principle of no initial or implicit trust, i.e., Zero Trust, is followed. This implies that no initial implicit trust between two nodes (e.g., vehicle A and vehicle B) is assumed and therefore, verifiability is a necessary part of establishing trustworthiness. Verifiability depends on the ability of the trustee (e.g., vehicle B) to provide evidence of meeting trustor's (e.g., vehicle A) expectations in a measurable or demonstrable way (see [5]). The process through which trustworthiness can be evaluated is based on the ability of the trustee to provide evidence of meeting those expectations in a verifiable manner.

Formally, given a trustor A and a trustee B, one can denote Trustworthiness of B for A's reasonable expectations regarding B's behaviour $R(x)$ (where x is A's goal-related expectation and B's behaviour is a function of that) in a context C as:

$$Tw_{B,A} \text{ – The likelihood that B will exhibit behaviour } R(x) \text{ in Context C} \tag{4.1}$$

Further, Trustworthiness can be a matter of degree or levels. That is, a trustee B may have the likelihood to fulfil the trustor's expectations to some degree or level L between 0 to 1, where 0 denotes no such likelihood and 1 denotes a maximal likelihood to fulfil trustor's expectations.

(4.1) can then be re-written as:

$$Tw_{B,A}(C, L) \text{ – The likelihood that B will exhibit behaviour } R(x) \text{ in Context C to a level L} \tag{4.2}$$

Finally, as stated, trustworthiness needs to be verifiable in the sense that the trustor should have access to evidence regarding B's likelihood to fulfil the relevant expectations. For the ideal/maximal evidence E, which would warrant appropriate trust in the trustee, we can write:

$$\begin{aligned} Tw_{B,A}(C, L, E) \text{ - The likelihood that B will exhibit behaviour } R(x) \text{ in Context C} \\ \text{to a level L established by evidence E} \end{aligned} \tag{4.3}$$

We should note here that the evidence is objective, but verification is subjective, meaning that the interpretation of the same evidence by different trustors might be based on different procedures, resulting in different verification results on the same evidence. For example, according to the verification procedure of trustor A, an evidence given in relation to a proposition might not be sufficient to justify the proposition, while in a different verification procedure, for example of trust B, it could be sufficient.

4.4 Trust

While trustworthiness is related to the trustee, the trust is in relation to the trustor. As previously explained, trustworthiness is a measure of a trustee's ability to meet the trustor's expectations. On the other hand, trust is a decision made (or an attitude held) by the trustor to trust or not to trust a concrete trustee.

A trusts B implies that A has expectations that B will have the property of being Trustworthy.

The trustor makes this decision based on an evaluation or assessment of the level of trustworthiness of the trustee, where such level exceeds a minimum required level (which may depend on the risks associated with the trustor trusting the trustee). When trustor A trusts trustee B, A deems that the likelihood that B will meet A's expectation is very high, or higher than what may be required given A's expectations and risks considered acceptable by A.

4.5 Properties of Trustworthiness

As mentioned, trustworthiness can be defined as the measure of the likelihood of the trustee to fulfil the expectations of the trustor in a given context, and the trustworthiness needs to be measurable and verifiable. One way to evaluate this likelihood is by assessing whether the trustee exhibits the right and relevant set of properties that enable it to meet the trustor's expectations in a given trust relationship. For example, consider a trust relationship between a zonal controller within a vehicle and a camera ECU during a The Cooperative Adaptive Cruise Control (CACC) function where the zonal controller is a trustor that relies on the camera ECU, the trustee, to deliver non-compromised camera data to it. Here, the camera ECU needs to exhibit, among others, the property of reliability. So, assessing whether the camera ECU is reliable in passing on its data to the ECU can give positive evidence of its trustworthiness.

The properties to evaluate trustworthiness of such trustees can in general be categorized into three broad categories:

1. Performance-based – These properties are linked to the performance of the trustee. For example, reliability and robustness.
2. Based on Ethical aspects – These properties are linked to the ethical aspects and implications of the trustee's expected behaviour in the given context. For example, privacy protection and safety.
3. Based on User acceptance – These properties are linked to properties that have implications for acceptance of the trustee (and the overall system) by the users. For example, transparency and usability.

These properties can be overlapping in the sense certain properties may belong to more than one category, or even in all three categories. For example, safety is a property that is linked to the performance of the CCAM system, but safety is also a required ethical value for the system to exhibit. Further, safety is also a property that potentially leads to higher acceptance of the system by the users. Similarly, integrity is an important property, which relates to the communication and data exchanged between different sensors and vehicle software remaining unaltered without proper authorization. The integrity of such communication also has critical significance both in terms of performance as well as for protection of key ethical values such as safety.

Here, we describe an indicative set of properties that are relevant for evaluation of trustworthiness of CCAM systems and their components. This list has been extracted from sources such as documentation on standards (such as [5], [4] and [3], existing literature on autonomous vehicle systems and trustworthiness (such as [19], and existing documentation on CCAM systems (Cooperative, Connected and Automated Mobility ([8]). This indicative list of properties would be further refined based on relevance for and purposes of CONNECT.

The descriptions of properties here are meant to indicate how a trustee can exhibit these properties, or how such properties can be verified, in the context of CCAM. As stated earlier, Verifiability itself is an important aspect of establishing trustworthiness in this context. Verifiability also involves the trustee's ability to provide evidence for justification for the decision by the trustor to trust the trustee. In future work, more precise conceptions of what properties are applicable and relevant to the evaluation of a particular trustee (for example, a particular zonal controller), how this particular trustee can exhibit these relevant properties, and what criteria it should fulfil for a positive evaluation or verification of trustworthiness will be formulated.

Table 4.1: An indicative list of relevant properties for evaluating trustworthiness in the context of CCAM systems

Property	Description
Accountability	Being responsible and answerable for the actions and decisions made by the autonomous vehicle system or its components. For example, in order to be trustworthy, in case of a technical malfunction of a specific component, the manufacturer who implemented the component must be accountable for the malfunction. (See [5] and [19])
Functional Safety	ISO 26262 [2] defines Functional Safety as the absence of unreasonable risk due to hazards caused by malfunctioning behaviour of Electrical/Electronic systems. According to 5GAA, trust in the context of received V2X data from the point of view of functional safety and safety of the intended function implies a) knowledge of the intended function, b) information about the required quality and accuracy, and c) knowledge of how the data generating subsystem is designed, developed, implemented, maintained and operated. (See [2])
Privacy Protection	Safeguarding personal information and ensuring that it is appropriately collected, used, secured, removed when not needed, and accessible only to authorized parties. Aspects of such appropriate collection include, for example, proper consent mechanisms or other similar measures which may be enshrined in the local data regulations (See [19]). In the context of CCAM, where data is collected through various sensors and cameras, privacy protection may also include data sanitization procedures such as removal of personally identifiable information.
Security	(Availability) Availability is the property of a system, service, or data to be accessible and operational when requested by authorized users. It ensures that the necessary resources are reliably and consistently available, without interruptions, failures, or deliberate attacks, thereby enabling the execution of intended tasks effectively. (See [5])
	(Integrity) Regarding data, integrity is a property whereby data have not been altered in an unauthorized manner since they were created, transmitted or stored [ISO 29167]. Regarding a system, integrity is a property of accuracy and completeness [ISO 27000]. Integrity measures the confidence an entity can place in the fact that information supplied has not been maliciously manipulated, which can be achieved through evidence provided to the entity. Possible trust sources that provide evidence for the trustee to ensure integrity are listed in Chapter 6.3. (See [5] and [4])
	(Authenticity) Ensuring the identity of an entity is as it is claimed by it. (See [5])
	(Confidentiality) Ensuring the protection of sensitive information from unauthorized access or disclosure. Confidentiality could be relevant in the context of CCAM, for example, when camera pictures are processed in the vehicle or in a MEC server. (See [3])
Accuracy	The ability to provide outputs within the expected range of closeness between the measured or computed value and the true value (or the value accepted as being true) (See [5]). Typically, a system, e.g. a positioning system, demonstrates such ability by reporting the distribution of the errors under the form of an error percentile, which represents the accuracy of its output.
Reliability	The ability of a system to demonstrate dependable behaviour and performance under varying conditions. As an example, for CACC to run properly, it must receive reliable data from various in vehicle sensors and cameras. (See [5] and [4])
Robustness	Demonstrating the ability to operate with a sufficient level of performance (and also a high level of consistency) in a variety of circumstances, i.e., under various challenging conditions and scenarios. In the context of CCAM, the system is robust if it can still provide results even if, for example, available data is not that accurate or the conditions on the road do not allow for high accuracy. Further, in the same service ecosystem, the property of robustness also relates to the ability of the system to react in case the conditions have reached a critical point of uncertainty such that safe functioning is impossible given the quality and accuracy of the data available. (See [5]).

Table 4.1: An indicative list of relevant properties for evaluating trustworthiness in the context of CCAM systems

Property	Description
Resilience	Being able to adapt, recover, and continue functioning effectively in the face of disruptions, failures, or unexpected events. Such failures or unexpected events in CCAM systems, for example, could be message loss or invalid input provided by a sub-system. (See [5] and [19]). Resilience comes also in the form of certificates that appropriate and up-to-date security mechanisms have been deployed that are capable to efficiently respond in cases of disruptions, failures, or unexpected events. For instance, as in the envisioned VRU-CASS use case, where security policies triggering the migration of part of the ACC functionality to an ECU with a high Level of Assurance constitutes such a (certified) response mechanism.
Transparency	Being open, clear, and understandable about the functioning, algorithms, and data usage of the autonomous vehicle system. (See [5] and [19])
Stability	Not changing easily or maintaining consistency over time without fluctuations. (See [4]). Further, the system should be able to produce the output of the system should not fluctuate if the input remains the same.
Completeness	Ensuring all necessary and relevant information is available without omission. (See [4])
Relevance	The ability to match the expected extent to which the information and outputs provided by the autonomous vehicle system are applicable and useful to the current context or situation. (See [4])
Consistency	The ability to deliver coherent and reliable performance, outputs, and decision-making over time and across different scenarios. A vehicle would have the property of consistency if, for example, the positions provided by the vehicle are compatible with each other. So the position provided by the vehicle is compatible with the prediction of the position of the vehicle, based on the previous send position and the kinematic data of the vehicle. (See [4])
Recency	Ensuring that the most up-to-date and recent information is used in decision-making and operation. (See [4])
Explainability	Providing understandable explanations or justifications for the decisions and actions taken by the autonomous vehicle system. (See [19])
Usability	The ease of use and user-friendliness of the autonomous vehicle system, ensuring that it is accessible, understandable, and navigable for users, promoting effective interaction and user acceptance. (See [5])
User-Centric	The ability to match the expected extent to which the autonomous vehicle system's decisions, actions, and behaviours align with and take into account the intended goals and objectives of the users or stakeholders. (See [19])
Equitable Access	Ensuring fair and unbiased access to the market for autonomous vehicle systems, without undue advantage or discrimination towards any particular system or provider. (See [8])

Chapter 5

Analysis of the State-of-the-Art in Trust Assessment

As part of this chapter, we provide an in-depth analysis of the state-of-the-art in Trust Assessment. The chapter is structured as follows. In Section 5.1, we discuss previous work on trust modelling and trust assessment, followed up with a discussion on the requirements for trust assessment in the automotive domain in Section 5.2. In Section 5.3, we discuss, analyse and compare existing decision logics that have been proposed in the literature to ascertain information in uncertain and unpredictable conditions, which supports and justifies our decision for choosing the Subjective Logic as calculus and theoretical framework for our TAF. Lastly, we explain the relevant parts of the formalization of Subjective Logic in Section 5.4.

5.1 Trust Modelling and Trust Assessment

There exists already some work related to Trust Modelling and Trust Assessment in the automotive domain, but also beyond in the context of other applications. In what follows, a more generic representation of the various methodologies will be documented on how trustworthiness assessment can be conducted for services, technologies and the systems that are providing these. Then, this will transition to how such mechanisms have converged to also allow the definition of trust models and trust reasoning frameworks based on which entities involved in the automotive domain can establish trust for cooperatively executing safety-critical functions.

Having a common understanding of the characteristics that could be used to describe trustworthiness and a common way to define the vocabulary and characteristics to allow stakeholders to make a judgement if a product, service or technology meets the stakeholder expectations, is crucial for any type of trust assessment.

An example of a work outside the automotive domain is the work of Kurdi et al. [29] who used Trust Assessment in the context of Cloud Service Providers (CSPs). CSPs collaborate and share services with each other when there is a shortage of resources. Thus, in case of a resource shortage, requests can be delegated to another CSP. However, if an untrustworthy CSP is requested that responds with a high delay or not at all, this can affect the quality of service. Therefore, a trust management framework was created to assess the trustworthiness of other CSPs. Based on this, it is decided to which CSP the request will be sent. For this purpose, subjective trust opinions are created based on a reputation based system and service level agreements (SLAs) [29].

In another work, trust assessment is used in the context of continuous authorization in Internet of Things (IoT) systems. Here, the zero trust approach is used (as it is also the case in the mod-

els created in the context of CONNECT), so that nothing is trusted by default. Therefore, each operation must not only be initially approved, but continuously verified throughout its lifecycle. To decide whether a device is allowed to access a resource, the trust level of the device is first assessed. For this purpose, information from many sources are used and discounted by operations inspired by subjective logic. Based on the calculated trust value, it is decided whether an entity is authorized to access resources or not [17].

A further work also addresses trust assessment in the context of IoT. More specifically, this work focuses on crowd-sensing environments where sensory sources provide streaming information to a processing node. Since in an IoT network the trustworthiness of nodes can be questionable, it is assessed how trustworthy the information provided by the node sources is. This quality of information is then used to determine the quality of inferences of the inferences drawn based on the information received by the node sources. This quality of inference is a level of trust expressed by subjective logic. The determined level of trust is then taken into account in the decisions that are made based on the derived output of the processing node [18, 22].

As already mentioned, in addition to cloud and IoT systems, trust assessment is also used on the automotive domain. For example, Sohail et al. used subjective logic in the context of VANETs. Here, the vehicles determine a subjective trust opinion on other vehicles based on either direct or indirect trust assessment. In direct trust assessment, a vehicle i determines its opinion about another vehicle j based on collected evidence and a one-to-one interaction. For this purpose, vehicle i sends a packet to the vehicle j and observes the forwarding behaviour of the vehicle j. Based on an enhanced ad-hoc on-demand distance vector protocol, the trust opinions can be distributed to other vehicles so that the other vehicles can calculate trust opinions based on direct or indirect trust assessment [36].

Garlichs et al. [20] have proposed a trust assessment framework used in the context of vehicle platooning. In vehicle platooning, vehicles have to cooperate with unknown and potentially malicious vehicles. If a malicious vehicle misbehaves, it can easily lead to safety problems, since vehicles in platooning are usually driving in short distance from each other. In the mentioned work, the host vehicle regulates the distance to its predecessor depending on how much the host vehicle trusts its predecessor in the platoon. The host vehicle determines this trust opinion by comparing the actual behaviour of the sender of a V2X messages it received with the actual behaviour of the sender over an extended period of time. Based on the calculated trust opinion and predefined values, it is decided which safety distance to use. The analysis in this paper has shown that it is very effective in the platoon context. However, the mechanism is highly application specific and provides a rather static trust model.

Müller et al. [31] used subjective logic to generate trust opinions about nodes to detect faulty and malicious vehicles. Each node runs a misbehaviour detection system and analyses the received messages. When a node receives inconsistent messages, it reports this to a broker. The broker uses a court-case like procedure to decide which vehicle is misbehaving based on the available messages from the vehicles. The trust in the misbehaving vehicle is then reduced. Moreover, the trust in the nodes that send the reports and thus behaved correctly is increased. In this way, misbehaving nodes can efficiently be detected and isolated.

As the aforementioned works show, trust assessment mostly focuses on peer-to-peer relationships between two entities which communicate directly with each other. Here, often only misbehaviour and reputation are used to assess trust, However, in a CCAM ecosystem, there are often many nodes involved that interact with each other to provide the service. This leads to a complex trust network where trust has to be assessed. This is where the CONNECT project comes in to provide the fundamental building block for trust assessment between all actors in

the CCAM ecosystem, from vehicles over MEC servers to the cloud, where a wide variety of trust sources can be considered for trust assessment. In this way, safe and secure operations of CCAM applications in the vehicle should be ensured.

5.2 Requirements for Trust Assessment in Automotive Domain

By focusing on trust assessment in the automotive domain, we can elicit a couple of requirements from the concrete type of use cases, systems, and applications, which can also be leveraged for designing the solution and conceptualizing some of the properties that a Trust Assessment Framework shall provide more concretely. Recall that the goal is to build upon and expand the Zero Trust concept to tackle the issue of how to bootstrap vertical trust from the application, the execution environment, and device hardware from the vehicle up to MEC and cloud environments in a CCAM ecosystem.

Assessing Trust in Complex Networks and Collaborative Trust: Modern systems are interconnected, forming even more complex systems-of-systems. First, the systems themselves are consisting of a growing number of components or sub-systems. For example, today's vehicles contain hundreds of Electronic Control Units. Second, the systems become more connected and collaborative with other systems, which enables them to attain goals that one system in isolation cannot, for example, Cooperative Intelligent Transport Systems (C-ITS), where intelligent transport systems (e.g., vehicles, infrastructure equipment, traffic control centres) communicate, collaborate and share information among each other.
As a result, there is the emerging need in the automotive domain, to assess trust on a complex networks, instead on a single trust relationship (i. e., assessing trust among two nodes). As a result, the Trust Assessment shall be able to establish trust in an entity based on input from multiple cooperative nodes, i. e., vehicles or MEC that work in collaboration. Additionally, there should be higher confidence in the trustworthiness of a node, when the trust is built based on opinions or inputs from multiple nodes (e. g., multiple vehicles), versus only one node (e. g., a single vehicles).

Trust Transitivity, Direct and Referral Trust Relationships: From our definitions on trustworthiness and trust from chapter 4, we know that trust is conceived between a trustor and a trustee. However, there might be cases where evidence is not available in a direct (or functional) trust relationship, or in general, the direct trust relationship between two entities (between two vehicles or between vehicle and MEC) might not exist. As a result, the Trust Assessment shall allow transitive trust, obtained through a single or a chain of referral trust relationships. In transitive trust relationships, entity A might not have a direct relationship with entity X, for which it needs to build trust. Instead, entity A has a relationship with entity B, and entity B has a relationship with entity X. Therefore, entity B can give a recommendation to entity A about entity X, which enables entity A to build an indirect trust relationship with entity X, through B's recommendation about entity X, and A's trust on B. To summarize, the problem of lack of direct trust between entities can be easily tackled by utilizing the notion of transitive trust relationships.
For example, let us assume that in one of our use cases, a vehicle is entity A and the MEC is entity B, which sends a CPM to the vehicle. This CPM is entity X. The vehicle has a (referral) trust relationship with the MEC and therefore assesses its trust in the MEC. The MEC has a (direct or functional) trust relationship with the CPM and therefore assesses trust in the CPM. In this way, the MEC can give a recommendation to the vehicle about the CPM, which enables the vehicle to

derive functional trust in the CPM. The process of combining referral and direct trust relationships is the essence of trust transitivity.

Node- vs Data-Centric Trust: As a third requirement, the Trust Assessment shall allow combining trustworthiness in data and trust in nodes. Namely, since trustworthiness in a data item depends not only on the initial trustworthiness in the data item but also on the node which provided or processed that data item, the Trust Assessment shall be able to combine both types of trustworthiness and trust, respectively.
If evidence of trust on a node cannot be provided, it is possible to assess trustworthiness and trust only on the data item received from the corresponding node. In this way, trust on data can be determined without having to assess trust on the node providing the data.

Trust based on Varying Sources of Trust: Since the complex trust networks contain different heterogeneous nodes, the trustworthiness and trust are assessed based on different and varying sources of trust for each node in the trust network. First, there are many possible trust sources, and second, what sources of trust are used also depends on the properties of the trust relationship. Additionally, the available trust sources might change over time. These trust sources provide evidence based on which trust is evaluated. Concretely, the trustee provides evidence that the trust sources are available, for example, that secure boot is implemented and can be used as a trust source. The trustor then receives and uses this evidence in the trust assessment process.

Probabilistic Trust Propositions: In the fourth requirement, we focus on required properties for the opinions based on which trustworthiness and trust are calculated as part of our Trust Assessment. In CCAM scenarios we focus on systems with increased dynamicity and ubiquity, including systems operating in unpredictable, uncertain and dynamically changing environment or context. Therefore, in the Trust Assessment we cannot use, for e. g., binary logic and assert propositions to only true or false, since we cannot determine the trust value proposition with high certainty. Instead, there emerges the need to use probabilistic opinions in the trust assessment process that also express degrees of uncertainty about the concrete proposition. Additionally, there are different nodes and types of nodes in the collaboration, e. g., vehicles or MEC, vehicles produced by different OEMs, all of them using different technological stack for their realization. Therefore, all the nodes have only a subjective perception of their environment, which is 1) only partial due to the limitations of the sensing capabilities of the system and 2) could be potentially contradicting with other nodes in the collaboration, especially in the presence of adversary nodes. In response, the Trust Assessment shall include belief ownership in order to reflect the subjective nature of beliefs that once merged they (ideally) reflect the objective world more precisely than all the opinions in independence (cf. second requirement).

Safety under critical time constraints: Due to the dynamic aspects of the systems, the Trust Assessment that dynamically assesses and calculates opinions shall operate in real-time under strict time requirements. This is further emphasized and put in the focus in CCAM scenarios, like applications from the automotive domain. Namely, here we focus on real-time applications with the highest requirements for safety and security, especially because any faults or failures can directly endanger users and even their lives. To summarize, as a fifth requirement, the process of calculating trust opinions and the entire trust assessment process should be fast and robust to be used at run-time for real-time applications.

5.3 Existing Decision-Logics

There are various approaches and methods that have been proposed in the literature to ascertain information in uncertain and unpredictable conditions that could be potentially use for assessing trust. In this section we discuss, analyse and compare some of them, concretely Probabilistic Logic, Fuzzy Logic, Bayesian Probability, Dempster-Shafer Theory and Subjective Logic. Based on this analysis, towards the end of this section, we make an argumentation why Subjective Logic was the logic of choice for the TAF in our project.

5.3.1 Probabilistic Logic

When we assume an objective world, we can use binary logic to assert propositions about a state of the world to be either false or true. However, the world is unpredictable, and in many situations, one cannot determine the nature of a proposition with certainty. In other words, it is practically impossible to determine with absolute certainty whether a given proposition is true or false. Through probability calculus, which takes argument probabilities in the range [0,1], we allow propositions to be partially true.

5.3.2 Fuzzy Logic

Fuzzy logic is a form of many-valued logic, employed to handle the concept of ”degrees of truth” in which the truth value of variables may be any real number between 0 and 1, compared to the Boolean logic where the truth values of variables may only be the integer values 0 or 1 [32]. For example, person’s height is a proposition with variable answers, the possible values in fuzzy logic could be “short,” “average,” or “tall.” Concretely, a person measuring 180 cm could be considered 0.5 tall and 0.5 average [7].

In the literature, fuzzy logic has been considered as another formal trust metric where the domains for variables incorporate uncertainty and vagueness. For instance, in [6], a fuzzy logic trust model has been proposed to detect untrusted nodes in smart grid networks.

One of the well-known fuzzy systems is Mamdani’s rule-based system [30], which uses the following rules: 1) fuzzify all input values into fuzzy membership functions, 2) execute the IF-THEN rules that map input or computed truth values to desired output functions, and 3) defuzzify the fuzzy output functions to get a continuous variable from fuzzy truth values.

In the process of fuzzification numerical inputs of a system are assigned to fuzzy sets with some degree of membership between 0 and 1. The values 0 and 1 indicate that the value does not belong, respectively belongs, within the fuzzy set. Any value between 0 and 1 represents the degree of uncertainty that the value belongs in the set.

5.3.3 Bayesian Probability

Bayesian Probability is an interpretation of the concept of probability that can be seen as an extension of propositional logic that enables reasoning with hypotheses. In the Bayesian view, a probability is assigned to a hypothesis, unlike a frequentist inference where a hypothesis is typically tested without being assigned a probability [21]. To summarize, in Bayesian statistics, probability is used as the fundamental measure of uncertainty; therefore, allowing for reasoning with propositions whose truth values are uncertain. Being an evidential probability, the prior probability of a proposition (i. e., the probability of the proposition being true prior to any evidence

being accounted for) is assigned, and as new evidence becomes available and accounted for, the probability of the proposition is updated through a mechanism called *Bayesian inference* [33]. Bayesian inference is realized in three steps: 1) represent all sources of uncertainty as statistical random variables, 2) determine and assign a prior probability distribution to the random variables and 3) as more evidence is made available, update the probability distributions by applying the Bayes' formula:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}, \tag{5.1}$$

where $P(A)$ represents the prior probability of the proposition A being true, and $P(A|B)$ is the conditional probability of A being true given B is true. As new evidence becomes available, the probability distributions describing the propositions are updated, and these updated probabilities are then used as priors for further calculations with new evidence.

5.3.4 Dempster-Shafer Theory

The Dempster-Shafer theory (DST) introduced by Dempster [15] and Shafer [34] is a flexible theoretical framework to represent uncertain data, providing a method for merging independent belief pieces of evidence collected from different agents. In the literature, DST is also referred to as evidence theory or theory of belief functions, and the theory is a generalization of the Bayesian probability. In the DST formalism, *the degree of belief* is called mass, and it is represented as a belief function rather than a Bayesian probability distribution. Belief functions map degrees of belief (or confidence, or trust) for one question based on the subjective probabilities for related questions. The belief functions are the first main idea behind the DST. The second idea, is using the Dempster's rule [15] for combining such degrees of belief stemming from independent sources of evidence. DST has been applied in many fields, including decision making, information fusion, pattern recognition, and sensor fusion [35].

In the following, we present in more detail the main notions of DST: mass, belief and plausibility. Let $\Theta = \{\theta_i, i = 1, ..., n\}$ be the *frame of discernment* of a problem under consideration. The set Θ consist of n exhaustive and exclusive possible values for a state variable v of interest. The powerset of Θ, denoted as 2^{Θ}, represents the set of all subsets of Θ. A *basic belief assignment* (BBA) m on Θ [34, 16, 26], also called a belief mass function, is a mapping of the powerset 2^{Θ} to $[0, 1]$ that satisfies the following conditions:

$$m(\emptyset) = 0 \; and \sum_{X \subseteq \Theta} m(X) = 1, \tag{5.2}$$

where the values $m(X)$of a BBA are called *basic belief masses* and represent how strongly the evidence supports X. If $m(\emptyset) = 0$, m is said to be normal and if and only if $m(X) > 0$, each subset $X \subseteq 0$ is called a focal element of m. A BBA m can also be represented by its associate belief function *Bel* and plausibility function *Pl* respectively, defined as follows:

$$Bel(X) = \sum_{Y \subseteq X \neq \emptyset} m(Y) \; and \; Pl(X) = \sum_{Y \cap X \neq \emptyset} m(Y). \tag{5.3}$$

In contrast to probability theoretic approaches, the DST facilitates the expression of the belief in every element of 2^{Θ} and not only the elementary hypotheses. Thus, even information sources can be used that can only provide their knowledge about subsets of Θ. Thus, it is feasible to fuse two BBAs received from sources S_1 and S_2 with different reliability using Dempster's rule of combination:

$$m_{S_1 \oplus S_2}(X) = m_{S_1}(X) \oplus m_{S_2}(X) = \frac{1}{1-k} \sum_{A \cap B = X} m_{S_1}(A) m_{S_2}(B), \tag{5.4}$$

where k is a normalization constant representing the *degree of conflict* between m_{S_1} and m_{S_2} defined as:

$$k = \sum_{A \cap B = \emptyset} m_{S_1}(A) m_{S_2}(B). \tag{5.5}$$

The use of Dempster's rule is mathematically possible only if m_{S_1} and m_{S_2} are not totally conflicting, and the normalization constant k reflects the degree of conflict between the two sources. This normalization effectively redistributes the conflicting belief masses to the non-conflicting ones, hereby eliminating the conflict between the sources.

5.3.5 Subjective Logic

Subjective Logic (SL) [25, 24] is a framework for artificial reasoning based on probabilistic logic and Dempster-Shafer theory of evidence [15, 34]. The idea of explicit representation of ignorance and fusing various sources of evidence are inherited from the Dempster-Shafer belief theory [24, 34], and the interpretation of an opinion in Bayesian perspective is possible by mapping opinions into probability distributions [24]. In Section 5.4, we give more details on the foundations of SL.

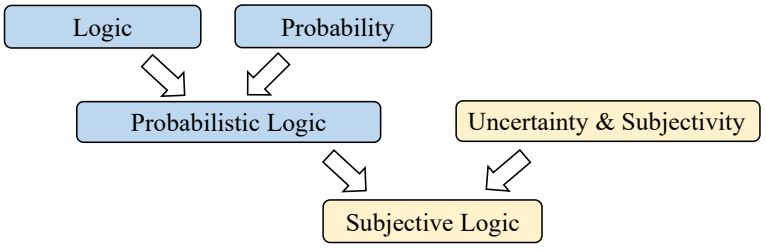


Figure 5.1: Subjective Logic Framework from [25].

5.3.6 Comparison of Different Decision-Logics

As discussed earlier, in probabilistic logic the truth values of propositions are probabilities. However, due to the lack of sufficient evidence, we are often unable to estimate probabilities with confidence. Furthermore, whenever the truth of a proposition is assessed, it is always done by an individual, and it cannot be considered to represent a general and objective belief. In order to reflect as faithfully as possible the perceived world in which we are immersed, a formalism to express degrees of uncertainty about beliefs is needed; said formalism also shall include belief ownership to reflect the subjective nature of beliefs [25, 24].

To reason with propositions whose truth-values are uncertain, Bayesian probability and statistics can also be employed [21]. As previously explained in Section 5.3.3, Bayesian probability is an interpretation of the concept of probability that can be seen as an extension of propositional logic. In Bayesian statistics, probability values are used as the fundamental measure of uncertainty. However, this type of probabilistic logic does not allow to seamlessly model situations where different agents express their beliefs about the same proposition. Dempster-Shafer Theory and

Subjective Logic explicitly integrate the subjective nature and ownership of beliefs in its formalism, allowing the combination of different beliefs about the same proposition.

In DST, probability values are assigned to sets of possibilities rather than single events. Furthermore, Dempster-Shafer's rule of combination is associative, commutative and non-idempotent [26] and the functions are simple to implement and compute. These properties of Dempster-Shafer's rule of combination are beneficial for real-time applications as sources can be combined sequentially, and at a random order [16]. However, these results should be considered with caution, as a later study [16] showed that the order of at which sources get aggregated may impact the results. Another downside of the DST is that it can lead to counter-intuitive results when combining conflicting sources [40], meaning that the rule cannot be applied if the two sources are in complete opposition. Additionally, the original formalization of DST does not model the aspect of trust transitivity, which is essential for the use cases and the types of systems that we focus as part of CONNECT. These limitations of the Dempster-Shafer theory are addressed in the Subjective Logic theory [25, 24].

Fuzzy logic is closely related to probability-based theories; however, there are essential differences is their meaning and how these theories have been conceptualized. First and foremost, fuzzy logic and probabilistic logic address different forms of uncertainty. Probability is usually associated with events instead of facts, that can either occur or not, with a certain probability of occurrence. Whereas, fuzzy logic tries to capture the essential concept of vagueness, focusing on degree of truth. In a nutshell, we can say that probability logic works with probabilities that are precisely assigned, and fuzzy logic works with imprecise possibility. To exemplify this, we gain refer to the example from Section 5.3.2. In probabilistic logic, we can say that there is a probability of 0.2 for a person's height to be 180 cm. On the other hand, in fuzzy logic, a person measuring 180 cm could be considered 0.5 tall and 0.5, if the possible values in fuzzy logic could be "short," "average," or "tall." Note that the height of a person can be measured in an exact and crisp way, whereas variables consist of terms that are fuzzy/vague in nature. However, in SL, the domains consist of terms that are considered crisp in nature, whereas subjective opinions contain belief mass and uncertainty mass that express uncertainty and vagueness. This difference between fuzzy logic and SL is depicted in Fig. 5.2. Therefore, if we consider SL (or DST) and fuzzy logic, then we can agree that they handle different aspects of uncertainty and vagueness. Potential combination of these reasoning frameworks needs to be further explored in the future [7].

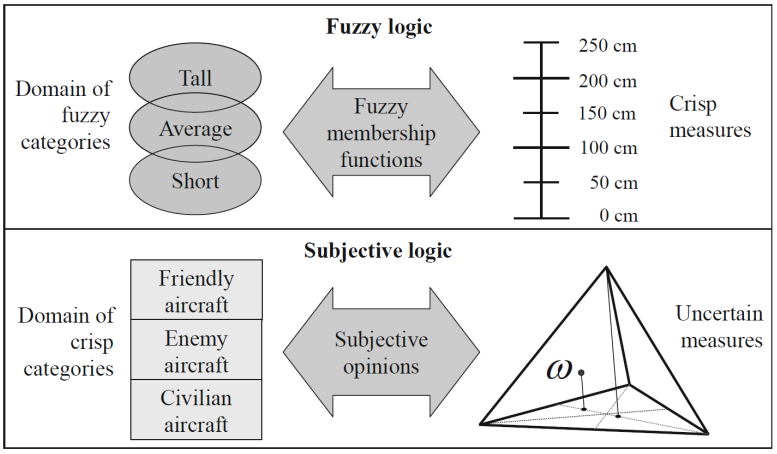


Figure 5.2: Difference between fuzzy membership functions and subjective opinions from [25]

In Tab. 5.1, we compare different decision logics based on different properties, and in the following, we briefly explain the differences between different logics. Namely, in Tab. 5.1, we can see that apart from the binary logic, on high-level, all the other logic in the table are dealing with uncertainties and uncertain situations; however, they all tackle this issue differently and with

different expressiveness. Except the binary logic, where the propositions can be either true or false, and the fuzzy logic that is many-valued logic, all the other logics are expressing probabilistic truth values. Additionally, as previously explained, in comparison to the probabilistic logic, Bayesian probability incorporates past evidence, through the consideration of the prior probability of a proposition. Lastly, although both DST and SL consider subjective beliefs from multiple agents on the same proposition, only SL enables merging conflicting sources of evidence. As previously explained, at the end of Section 5.3.4, the use of Dempster's rule is mathematically possible only when the belief masses of the two sources are not in total conflict. This was further supported by the results of a separate study by Zadeh and Ralescut [39], in which the authors state that although it is relatively straightforward to calculate, Dempster's Rule generates counter-intuitive results when there is a high degree of conflict between the two belief masses. Additionally, DST does not support and model the notion trust transitivity, which as explained in the requirements that we elicited in Section 5.2, is one of the core requirements for the type of systems that we focus in CONNECT. Namely, instead of assessing trust on trust relationships in this project we focus on assessing trust in more complex trust networks, where the notion of referral and direct trust relationships—therefore, trust transitivity, are put at the forefront. In response, as part of our project, we have decided to use the formalism of SL theory for trust assessment under uncertainties. We explain some of the relevant concepts from the SL theory for our project, in more detail, in the following section.

Table 5.1: Comparison of different decision logics

Decision Logics	Dealing with uncertainties	Probabilistic truth values	Incorporating past evidence	Subjective beliefs	Merging conflicting sources	Trust Transitivity
Binary Logic	✗	✗	✗	✗	✗	✗
Probabilistic Logic	✓	✓	✗	✗	✗	✗
Fuzzy Logic	✓	✗	✗	✗	✗	✗
Bayesian Probability	✓	✓	✓	✗	✗	✗
Dempster-Shafer Theory	✓	✓	✓	✓	✗	✗
Subjective Logic	✓	✓	✓	✓	✓	✓

5.4 Foundations of Subjective Logic

In recent years, SL has gained prominence because of its capability to deal with the degree of (un)certainty of propositions, i. e., it explicitly represents the amount of "uncertainty on the degree of truth about a proposition" [24]. Concretely, SL inherently allows:

1) uncertainties representation as part of the fundamental building block of SL, called *subjective opinion* (see Section 5.4.1), and

2) reasoning about the uncertainties through a process of *belief fusion* in which multiple subjective opinions are aggregated based on the selected fusion operator (see Section 5.4.2).

Additionally, in Section 5.4.3, we explain the concepts of *subjective trust networks*, *trust discounting* and *DSPG*, since they are relevant for the engineering of our Trust Assessment Framework (TAF), in particular the Trustwortiness Level Expression Engine (TLEE) that we explain in Section 7.2.4. Namely, the propositions that the TLEE receives as input are *rewritten* as expressions inside the TLEE. In other words, these expressions are then continuously expanded in different modules of the TLEE, based on the trust network, and using the SL-related concepts like trust discounting, trust fusion and the DSPGs, based on which the final, aggregated subjective opinion is being derived.

5.4.1 Subjective Opinions

The fundamental building block of SL is a *subjective opinion* that represents the amount of uncertainty on the degree of truth about a proposition. The representation of a subjective opinion is a composite function consisting of belief masses, uncertainty mass and base rate. Formally, a subjective opinion expresses a belief about a state of a variable X which takes its values from a domain $\mathbb{X}$ (i. e., a state space). The domain is the set of all different possible states for X. The different values of a domain are exclusive, i. e., only one state value is possible at any moment in time-, and exhaustive, i. e., all possible state values are included in the domain. Domains can be binary (exactly two values) or n-ary (n values) where $n > 2$. A binary domain can be denoted $\mathbb{X} = \{x, \overline{x}\}$, where $\overline{x}$ is the complement (negation) of x.

Binomial Opinions

In general, the notation ω_X^A is used to denote opinions in subjective logic, where the subscript X indicates the target variable or proposition to which the opinion applies, and the superscript A indicates the subject agent/source who holds the opinion. Superscripts can be omitted when it is implicit or irrelevant who the belief owner is. A binary domain consists of only two values and the variable is typically fixed to one of the two values. Formally, let a binary domain be specified as $\mathbb{X} = \{x, \overline{x}\}$, then a binomial random variable $X \in \mathbb{X}$ can be fixed to $X = x$. Opinions on a binomial variable are called binomial opinions.

Definition 1 (Binomial Opinion [25])**.** Let $\mathbb{X} = \{x, \overline{x}\}$ be a binary domain with binomial random variable $X \in \mathbb{X}$. A binomial opinion about the truth/presence of value x is the ordered quadruplet $\omega_x = (b_x, d_x, u_x, a_x)$, where the additivity requirement

$$b_x + d_x + u_x = 1$$

is satisfied, and where the respective parameters are defined as:

b_x : belief mass in support of x being TRUE (i.e. $X = x$),
d_x : disbelief mass in support of x being FALSE (i.e. $X = \overline{x}$),
u_x: uncertainty mass representing the vacuity of evidence,
a_x : base rate, i.e., prior probability of x without any evidence.

The projected probability of a binomial opinion about value x is defined by:

$$P(x) = b_x + a_x u_x. \tag{5.6}$$

5.4.2 Belief Fusion

One of the central features of subjective logic is the process of belief fusion (also known as trust fusion) that allows multiple opinions regarding the same proposition to be merged or aggregated into a single, collective opinion. The resulting opinion shall represent the truth better than each opinion in independence. The process of belief fusion is illustrated in Fig. 5.3. There could be different sources of evidence based on which opinions are formed, for e. g., different agents in a CCAM scenarios, or sensors, ECUs, etc. Depending on the specific application, a suitable operator of belief fusion with different properties is required.

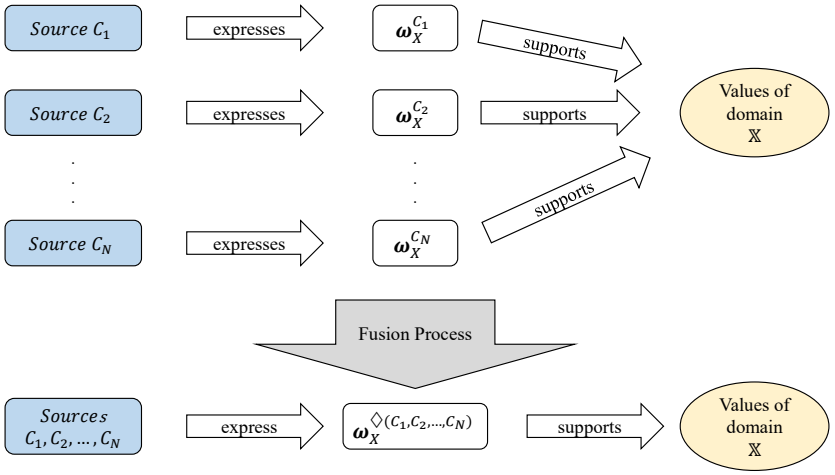


Figure 5.3: Fusion process, updated from [27].

Subjective Logic Operators

Since SL is a generalization of binary logic and probability calculus, some of these operators originate from binary logic (e. g., AND and OR) and set theory (e. g., union, difference and Cartesian product). However, there are other fusion operators that are unique to subjective logic: *constraint fusion, cumulative fusion, averaging fusion, weighted fusion, consensus & compromise fusion, unfusion*, as well as *trust discounting*. Each of these fusion operators are useful in different scenarios, hence the choice on the operator depends on the application and goal of the fusion process. For a more detailed explanation and mathematical definition of different fusion operators, please refer to [27, 25].

5.4.3 Subjective Trust Networks, Trust Discounting and DPSG

A subjective trust network (STN) represents trust and belief relationships from agents, via other agents and sensors to target entities/variables, where each trust and belief relationship is expressed as a subjective opinion [25]. The trust network is typically represented as a graph.
In Figure Fig. 5.4, we show a simple example of an STN graph. The main SL operators used in computational trust and STN graph processing are the fusion operators and trust discounting. As previously explained in Section 5.4.2, there are various types of fusion operators, and the decision on which operator should be used is application- and use case-dependent. On the other hand, trust discounting is the operator for deriving trust from transitive trust paths.
Namely, trust discounting is closely related to the transitive trust principle. Referring again to Fig. 5.4, according to the transitive trust principle, A trusts B, since A knows that B has a

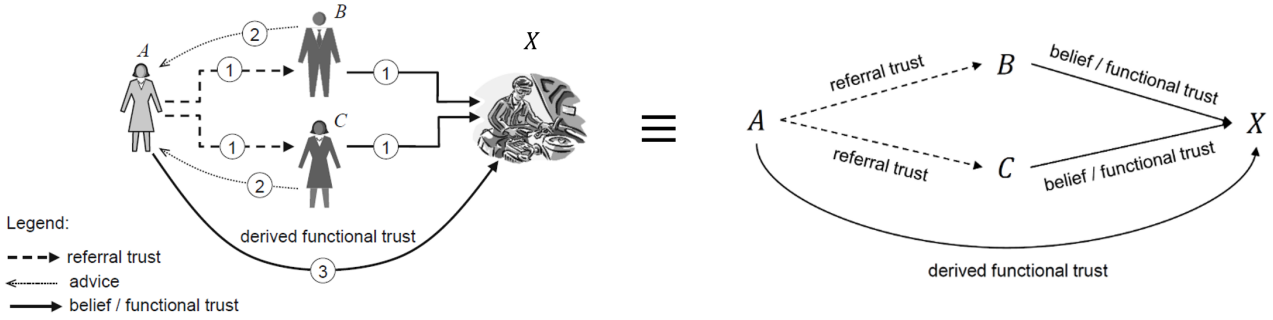


Figure 5.4: STN and example of trust fusion, from [25].

direct experience with X (i. e., referral trust), and B trusts X, since B has the direct observation or experience with X (i. e., functional trust). Therefore, A could derive trust on X (i. e., derived functional trust). In other words, based on the combination of A's trust in B, as well as on B's opinion about X received by A, it is possible for A to derive an opinion about X. The process of trust discounting is illustrated in Fig. 5.5.

Figure 5.5: Trust discounting of opinions, updated from [25].

Lastly, the opinion ω_X^A in Fig. 5.4, is calculated as follows:

$$\omega_X^A = (\omega_B^A \otimes \omega_X^B) \odot (\omega_C^A \otimes \omega_X^C),$$

where the discounted opinions from $\omega_B^A \otimes \omega_X^B$ and $\omega_C^A \otimes \omega_X^C$ are fused together. They symbol $\odot$ depicts a general fusion operator, which will be replaced with the chosen fusion operator in the later stages of the TAF development.
For the operators for fusion and trust discounting to be applied to referral and functional trust relationships, trust networks need to be represented as a Directed Series-Parallel Graph (DSPG). DSPGs have a fundamental role in the TLEE implementation.

Definition 2 (Directed Series-Parallel Graph (DSPG) [25])**.** A graph is called a Directed Series-Parallel Graph (DSPG) if it can be decomposed as a combination of Series and Parallel graphs and it only consists of directed edges that form paths without loops from the source to the target.

By employing trust discounting and fusion operations, the DSPG trust network can be effectively transformed into an equivalent single edge. This consolidated edge encapsulates the comprehensive trust flow from an agent to a proposition while considering the various trust relationships and their corresponding subjective opinions. The described fusion of opinions and the derivation of the final opinion is the foundation for trust assessment as part of our work.

Chapter 6

Trust Modeling, Trust Relationships, and Trust Sources in CCAM Ecosystems

In this Chapter, we exemplify different types of trust relationships that frequently occur and need to be assessed in the context of CCAM ecosystems together with introduction of a well-defined methodology for extracting such relationships based on the nature of the application of interest as well as the security and timing constraints of its operation. Finally, we leverage this methodology to list in detail different types of trust relationships that we identified in the envisioned use cases that will become important when later specifying trust models for these use-cases. We start with a discussion of how trust models are derived and modelled.

6.1 Trust Models and Modelling Methodology

Let us assume that while assessing trust we always focus on a concrete function of a system, e.g., the ACC as part of the *VRU Protection through Cooperative Adaptive Cruise Control* use case. Therefore, it is necessary to include all the components of the system and components' relationships that are needed to realize that concrete system function. Therefore, as part of our TAF we refer to it as a *Component Diagram*. The Component Diagram is an undirected graph and serves as a foundation for the derivation of our trust model.

A trust model is a graph-based model which represents all components and data needed to perform a certain function. It consists of trust objects and directional trust relationships between trust objects (i. e., the trust network). It also stores trust sources used to quantify trust relationships. A correct and complete trust model (TM) is vital for the functioning of the TAF, as it is the main input to the TMM and the TLEE. Therefore, a structured and validated design methodology for extracting and building trust models is of high importance. For in-vehicle TAF, we envision that the trust models are built by an engineer at vehicle design-time. The engineer decides which vehicle safety-critical functions the application will need to assess the trustworthiness of, and has the task of building trust models for each of those functions. The following information is needed to build a trust model for any specific function: i) *Component Diagram*, ii) *Function's Data Flow Diagram*, and iii) *additional knowledge about the function*.

A simplified Component Diagram is shown in Figure 6.1. As can be seen from the figure, the vertices represent in-vehicle components such as the Vehicle Computer, Zonal Controller, Radar ECU, etc., and links represent the physical connections between components through which the data flow. An example Data Flow Diagram for a ACC function is given in Figure 6.2. It shows which components the data passes on its way from the data sources to the destination. This

diagram also only contains the data sources which provide the input necessary for the function to run. Additional knowledge about the function typically includes knowing which component the function is executed on.

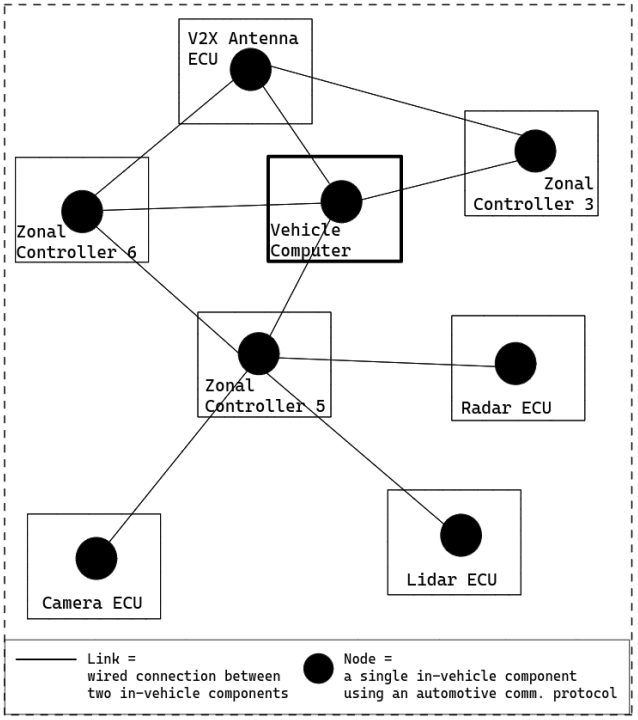


Figure 6.1: Example Component Diagram

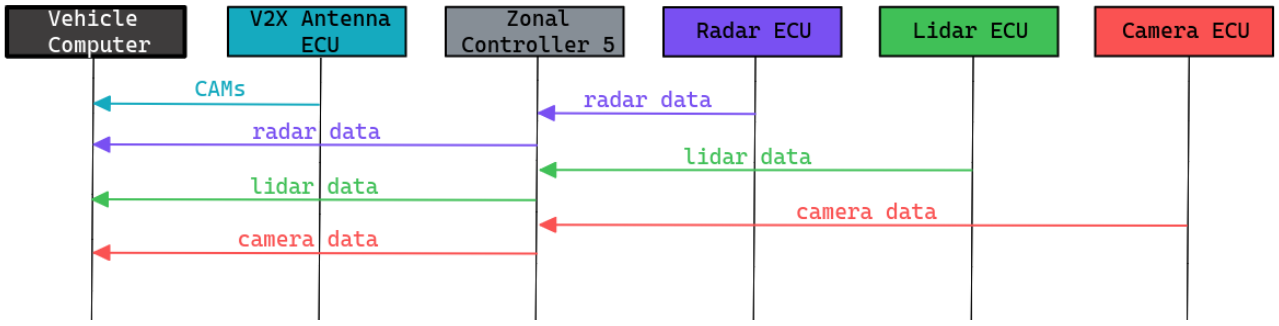


Figure 6.2: Example Data Flow of a Function

Equipped with these tools, the engineer can follow the subsequent methodology to build a function-specific trust model (note that there are two sub-steps within each step, one for the component / data flow diagram, the other for the trust model):

1. *Component Diagram*: Identify the component where the function will be run, i.e., the component that the TAR will come from. **Trust Model**: Instantiate a node trust object which represents this component.

2. *Component Diagram*: Identify source components whose data is provided as input to the function. **Trust Model**: Instantiate node trust objects for every one of those components.

3. *Component Diagram & Data Flow Diagram*: Identify the components through which the data flow to get from the source components to the destination component running the function. **Trust Model**: Instantiate node trust objects for every one of those components.

4. *Component Diagram & Data Flow Diagram*: Identify the exact routes and the direction of the data flow between sources and destination. **Trust Model**: Create directed links between node trust objects that match the data flow but point in the opposite direction. Each link represents a trust relationship and has a scope.

5. *Component Diagram*: - **Trust Model**: Instantiate a node trust object for the TAF and create directed links between the TAF and the component running the function pointing from the TAF to the component. The amount of links between the two should equal to the number of data sources.

6. *Component Diagram*: - **Trust Model**: If the data source components are deemed capable of hosting a TAF, instantiate data trust objects representing the data created by each of those sources. Create directed links between the node trust objects representing the data sources and the data trust objects representing the data.

In Step 6, the engineer will instantiate a data trust object representing the data only if the data source is capable of hosting a TAF to assess the trustworthiness of its data. If it is not, as is often the case in simpler sensors, the data's trustworthiness will be equated to the trustworthiness of the data source itself and the node trust object will be the leaf in the trust model.

Using this methodology and the example component and data-flow diagrams in Figures 6.1 and 6.2 respectively, we created a sample trust model given in Figure 6.3. As can be seen from the figure, there is a trust object for every component that either creates (*camera ECU* C, *lidar ECU* L, *radar ECU* R), relays (*antenna ECU* A, *zonal controller 5* $ZC5$), or processes the data (*vehicle computer* VC); as well as for every data whose source component has the computing capability to host a local TAF (*camera data* $Cdata$, *lidar data* $Ldata$, *radar data* $Rdata$).

Note that trust relationships are embedded in trust networks, precisely in trust models. As previously explained in the taxonomy on trust relationships that we propose in Section 4.2, we have referral and functional (direct) trust relationships. For example, in Fig. 6.3, the trust relationship from the vehicle computer VC to the zonal controller 5 $ZC5$, is a referral trust relationship; whereas, the trust relationship from the camera ECU C and its data $Cdata$ is a direct or a functional trust relationship. Additionally, in the direct trust relationships, we differentiate between data-centric and node-centric trust relationships, based on the type of propositions (see Section 4.2). Namely, data-centric trust relationships are formed between 1) nodes as trust objects that represent data source components and 2) trust objects that are related to propositions about the created data (e.g. a data-centric trust relationship between camera ECU C and its data $Cdata$). Node-centric trust relationships are formed between nodes and propositions related to properties of nodes (not depicted in Fig. 6.1). Moreover, depending on the model, some trust objects will be only trustors (e.g. VC), others will be only trustees (e.g. $Ldata$), and some will be both (e.g. $ZC5$). Note that the underlying assumption in this example trust model is that the V2X antenna ECU does not have the capability to assess the trustworthiness of the data it receives on its own. A detailed description of node-centric, data-centric, referral and functional trust relationships is provided in Section 4.1.

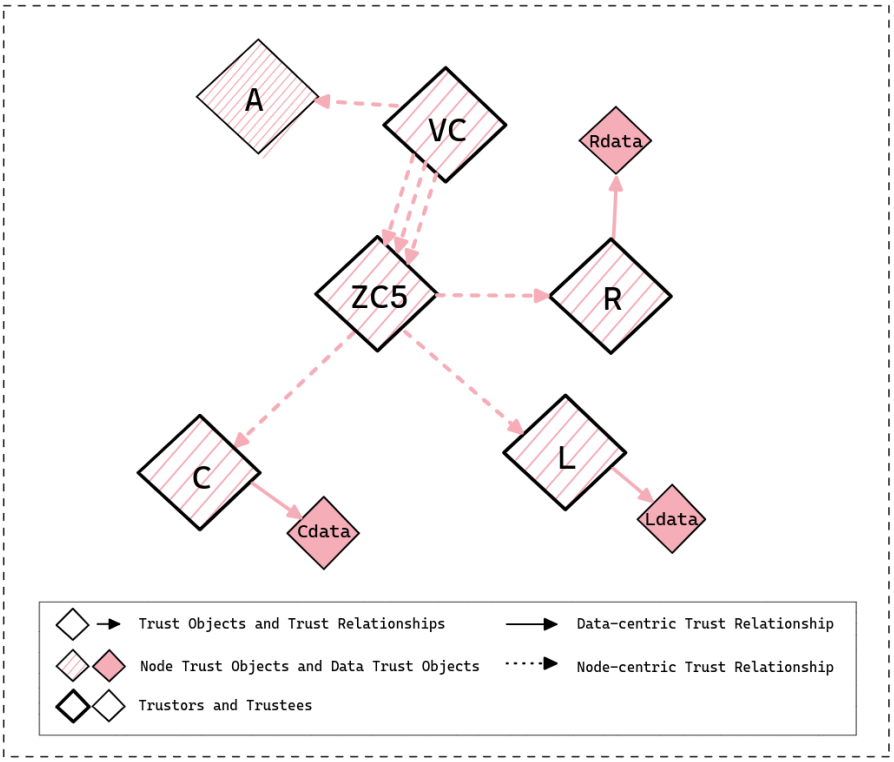


Figure 6.3: Example Trust Model

The objects and the trust relationships in the in-vehicle trust models can be mostly static and do not necessarily need to be updated during run-time. This is because they are based on the structure of the in-vehicle network which, unlike in ad-hoc networks such as V2X, stays mostly static during the life-time of a vehicle. If any changes are made on a need-basis, then the trust models also need to be updated. For use-cases which are of an ad-hoc nature, trust models will need to be built and updated dynamically during run-time. How this will be done is part of our future work.

6.2 Trust Relationships

Following the aforementioned trust modelling methodology, in the remaining part of this chapter, we identify the types of trust relationships that comprise the trust model for each one of the envisioned use cases. Recall that as described in D2.1 [10], and summarized in Chapter 3, each use case focuses on different CCAM functionalities with varying requirements as it pertains to data- and node-centric trust. This also entails that trust level calculations will be based on the aggregation of evidence that depict different trust properties (cf. Section 4.5), e.g., safety, security, resilience, robustness, reliability, etc. However, the common denominator, in all these safety-critical functions, the has the highest impact in their trust level is **integrity** - *be it integrity of the devices comprising the service graph chain or integrity of data communication based on which critical driving decisions are made*.

As previously defined in chapter 4, trust relationships describe a directed relationship between a trustor and a trustee for a specific property. These potential relationships are described, in the context of the envisioned use cases (i.e., "*Intersection Movement Assistance*", "*Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (C-ACC)*", and "*Slow Traffic*

Movement Detection") to create a trust chain between the trustor and the trustee. Based on this chain-of-trust, the trust opinion the trustor has on the trustee can be determined and calculated by aggregating all collected evidence that depict the behaviour of the trustee. For instance, if an ECU wants to evaluate the trustworthiness of the data coming from the camera, there would be one trust relationship between the ECU and the camera, and another between the camera and its own data, since there is no direct link between the ECU and the camera data. In addition, for each trust relationship, a list of possible trust sources that can be used as evidence is provided.

6.2.1 VRU Protection through Cooperative Adaptive Cruise Control

The Cooperative Adaptive Cruise Control (C-ACC) is a driving assistance system that allows the vehicle to automatically keep a safe distance to other vehicles based on its own sensors' data and data received from other vehicles via V2X communication. In this use case, we focus on the in-vehicle point part and investigate the data flows and computation steps within a vehicle to generate the information exchanged in the context of C-ACC and use the information to derive in-vehicle driving commands.

- **The C-ACC vehicle system:** The in-vehicle E/E architecture is composed by a main vehicle computer (VC), where the C-ACC function runs, as well as a main TAF. The VC receives the data coming from the sensors through the zonal controllers (ZC). There are three ZCs: ZC1, responsible for forwarding data from both Lidar ECU and Camera ECU, as well as GNSS; ZC2, responsible for forwarding the Radar data and another one, ZC5, for forwarding steering and acceleration commands. As sensors, there is a Radar ECU, a Lidar ECU and a Camera ECU, which are responsible for perceiving the surroundings (e.g., pedestrians, another vehicles, obstacles). Furthermore, there is a camera which is equipped with a GNSS sensor for GPS data. For receiving external data and cooperating with the surrounding infrastructure and vehicles, the system has a smart antenna.

Considering the high safety impact of the C-ACC data and commands, it is important to establish the trustworthiness of the environments where such data was extracted; i.e., be able to verify the operational assurance of the sensor that was responsible for extracting, collecting and parsing the C-ACC related data in a trustworthy manner. The following list of trust relationships and interactions between components provide examples of trust relationships that occur in this use-case and that need to be reflected in our model.

ID	TR.1
Type	Unidirectional node-centric trust relationship
Trustor	Camera ECU
Trustee	GNSS sensor
Description	The camera ECU requests GPS data from the GNSS sensor. The camera ECU needs a trust opinion on the GNSS sensor to determine the trustworthiness of the GNSS sensor. Based on the trustworthiness, it should be determined whether the integrity of the GNSS sensor and its data is provided. The camera ECU is also responsible for creating an opinion on the GNSS data, since the GNSS sensor is not able to create an opinion on its own data due to resource constraints.
Trust Sources	TS.7, TS.10, TS.13

ID	TR.2
Type	Unidirectional node-centric trust relationship
Trustor	Zonal Controller 1
Trustee	Camera ECU
Description	The Camera ECU sends the produced pictures, as well as the GPS data from the GNSS sensor to the Zonal Controller 1. Therefore, the Zonal Controller needs to have a trust opinion on the *integrity* and *correctness* of the Camera ECU in order to attest the integrity of the extracted data.
Trust Sources	TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.3
Type	Data-centric trust relationship
Trustor	Camera ECU
Trustee	Camera data
Description	The Camera ECU sends the taken pictures, as well as the GPS data from the GNSS sensor to the Zonal Controller 1. Therefore, the Zonal Controller needs a trust opinion on the Camera ECU to determine the trust of the data sent by the camera. This trust is assessed in the context of the integrity of the data.
Trust Sources	TS.7, TS.10, TS.12, TS.13, TS.14

ID	TR.4
Type	Unidirectional node-centric trust relationship
Trustor	Zonal Controller 1
Trustee	Lidar ECU
Description	The Lidar ECU sends Lidar data to the Zonal Controller 1. Therefore, the Zonal Controller 1 needs to calculate a trust opinion on *integrity* and *correctness* of the Lidar ECU to determine the trust of the extracted data.
Trust Sources	TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.5
Type	Data-centric trust relationship
Trustor	Lidar ECU
Trustee	Lidar data
Description	The Lidar ECU sends the Lidar data to the Zonal Controller 1. Therefore, the Lidar ECU needs to calculate a trust opinion on the Lidar data to determine the trustworthiness of the created data.
Trust Sources	TS.2, TS.7, TS.8, TS.10, TS.12, TS13, TS.14

ID	TR.6
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer
Trustee	Zonal Controller 1
Description	The Zonal controller 1 is responsible for receiving and forwarding requests and data from/to the vehicle computer. The vehicle computer needs to calculate a trust opinion on the *integrity* and *correctness* of the Zonal Controller 1 to determine the trustworthiness of the forwarded information. If the Zonal Controller 1 were untrustworthy, it could falsify the forwarded data and commands.
Trust Sources	TS.1, TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.11, TS.15

ID	TR.7
Type	Unidirectional node-centric trust relationship
Trustor	Zonal Controller 2
Trustee	Acceleration actor
Description	The Acceleration actor is responsible for speeding up and down the vehicle, according to the C-ACC component commands. This is a crucial safety-critical action, as part of this application. Therefore, the Zonal Controller 2 needs to calculate a trust opinion on the *integrity* and *correctness* of the acceleration actor to determine the trust of the actions made by this component, which is part of the operational environment.
Trust Sources	TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.8
Type	Unidirectional node-centric trust relationship
Trustor	Zonal Controller 2
Trustee	Steering actor
Description	The Steering actor is responsible for steering the vehicle, according to the CACC component needs. This is a crucial action in this application. Therefore, the Zonal Controller 2 needs a trust opinion on the Steering actor to determine the trustworthiness of the actions made by this component.
Trust Sources	TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.9
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer
Trustee	Zonal Controller 2
Description	The Zonal controller 2 is responsible for receiving and forwarding requests from/to the vehicle computer and send commands to the actuators for steering and acceleration. The vehicle computer needs a trust opinion on the Zonal Controller 2 to determine the trustworthiness of the forwarded data. If the Zonal Controller 2 were untrustworthy, it could falsify the forwarded data and commands.
Trust Sources	TS.1, TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.11

ID	TR.10
Type	Unidirectional node-centric trust relationship
Trustor	Zonal Controller 5
Trustee	Radar ECU
Description	The Radar ECU sends radar data to the Zonal Controller 5. Therefore, the Zonal Controller 5 needs a trust opinion on the Radar ECU to determine the trustworthiness of the Radar ECU. If the Radar ECU were untrustworthy, it could falsify the created data.
Trust Sources	TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.11
Type	Data-centric trust relationship
Trustor	Radar ECU
Trustee	Radar data
Description	The Radar ECU sends radar data to the Zonal Controller 5. Therefore, the Radar ECU assesses the trust on the Radar data it provides so that it can send the trust of the Radar data to the STDM.
Trust Sources	TS.2, TS.7, TS.8, TS.10, TS.12, TS13, TS.14

ID	TR.12
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer
Trustee	Zonal Controller 5
Description	The Zonal controller 5 is responsible for receiving and forwarding requests from/to the vehicle computer and sends commands to the radar. The vehicle computer needs a trust opinion on the Zonal Controller 5 to determine the trustworthiness of the data forwarded. If the Zonal Controller 5 were untrustworthy, it could falsify the forwarded data and commands.
Trust Sources	S.1, TS.2, T.S3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9

ID	TR.13
Type	Data-centric trust relationship
Trustor	Vehicle Computer
Trustee	CACC data
Description	The Vehicle Computer needs a trust opinion on the CACC data to determine the trustworthiness of its requests and processing data. More precisely, the Vehicle Computer wants to know whether the commands sent to it regarding speed and steering movements are trustworthy.
Trust Sources	TS.2, TS.7, TS.8, TS.10, TS.12, TS13, TS.14

ID	TR.14
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer
Trustee	Smart antenna
Description	The Vehicle Computer needs a trust opinion for data coming from the smart antenna and received over the V2X connection. For this purpose, the vehicle computer needs a trust opinion on the smart antenna.
Trust Sources	TS.2, TS.7, TS.8, TS.10, TS.12, TS13, TS.14

For this use case, 14 trust relationships were identified, where 10 are node-centric relationships and 4 are data-centric. The components with more computational power can have a TAF instantiated to assess a trust opinion on their own data that they create. In our use case, this could for example be the case for the Camera ECU. If a node is not able to assess the trust in its own data, the opinion must be created by the trustee, e.g,, the Camera ECU needs to create a trust opinion on the GNSS data because the GNSS ECU can not create an opinion on its own due to absence of a TAF and computational power.

It should be mentioned that no trust relationships were described that are created based on trust discounting of already existing trust relationships. Here, only trust relationships that are evaluated on the basis of trust sources were described. Therefore, the total number of trust relationships in the trust model will be higher.

6.2.2 Intersection Movement Assist

The goal of the Intersection Movement Assist (IMA) application is to timely alert the driver of potential collisions with other vehicles. The IMA, running on the ego vehicle, estimates the possible trajectories of the surrounding vehicles based on its consolidated view of the scene, which is built using the local perception and the data contained in the received V2X messages. This data is stored in the Local Dynamic Map (LDM), a database where each item is the kinematic description of a dynamic object, or observation.

The IMA application needs a reliable consolidated view of the scene. A key to achieve this is assessing the trustworthiness of the elements in the LDM, with respect to its correctness (i.e., capacity of faithfully describing the reality). The vehicle sub-system can be assisted in this by services provided by the MEC. We consider here two scenarios. In Scenario 1 (S1) the MEC provides the V2X Node Trustworthiness Assessment Service (NTS). The MEC, which receives and is able to process evidence on the capacity of V2X-nodes of producing correct V2X messages, provides the vehicle with node-centric trust assessment on active V2X-nodes. This can be exploited by the in-vehicle system when it performs data-centrc trust assessment on the data received from a known V2X-node. In Scenario 2 (S2) the MEC provides an extended version of the Collective Perception Service called the geo-Collective Perception Service. The MEC receives V2X messages from the vehicles and elaborates a local (at the MEC) consolidated view of the intersection scene, which includes trust levels on each description of dynamic object. The MEC encodes geo-CPM messages containing observations and the relative trust levels, and diffuses them to the vehicles. The MEC provides data and data-centric trust assessment to the vehicle. The in-vehcile system uses the data from geo-CPMs to build the local consolidated view of the scene. Notice that if the geo-CPM service from the MEC is not available, the in-vehicle system exploits the contents of incoming CAMs and CPMs to build the consolidated view of the scene, as done in S1.

In the following paragraph, we describe the system components involved in our use case.

- **In-vehicle system:** It runs on the Vehicle Computer, and it is composed of the V2X communication module, which is in charge of encoding the V2X messages (i.e. the CAM and the CPM messages), their transmission and their reception. In S1, the Misbehaviour Reporting generation and transmission functions are handled by the V2X communications module, as well. The in-vehicle system hosts the Local Misbehaviour Detection module, which processes incoming CAMs and CPMs. In S1 the output of this module is used by the Misbehaviour Reporting module. Moreover, in S1 and in S2, the output of the Local Misbehaviour Detection module is transmitted to the TAF. The TAF is used to assess data-centric trustworthiness in the observations contained in CAMs and CPMs (S1 and S2 if the geo-CPM service from the MEC is not available); node-centric trustworthiness in other V2X-nodes (S2 if the geo-CPM service is not available); node-centric trustworthiness levels in services provided by the MEC (S1 and S2). The in-vehicle system maintains the Active MEC Service Directory (ASD), which is a database of the identities of known and active services delivered by the MEC, with the current associated trustworthiness level. The in-vehicle system also maintains a local Active V2X Node Directory (AND) which contains the trustworthiness level of the active V2X-nodes. The in-vehicle system maintains a LDM which stores the received observations of dynamic objects.
- **The MEC system:** To provide the NTS and geo-CPS services, the MEC uses the TAF to assess data-centric trustworthiness in the observations contained in the CAMs and CPMs it receives from the V2X-nodes, with respect to their correctness; to assess node-centric trustworthiness in V2X-nodes, with respect to their ability to include correct data in CAMs and CPMs. The MEC maintains the Active V2X Node Directory (AND) which is a database of the list of active known V2X nodes with their trustworthiness level. It also has a V2X message reception module, a misbehaviour report reception module, a Local Misbehaviour Detection module and a geo-CPM encoding and transmission module. It also maintains an LDM which is populated with the observations contained in the V2X messages received by the V2X-nodes.

The following list of trust relationships are examples of trust relationships that occur in this use-case and that need to be reflected in our trust model.

ID	TR.15
Scenario	S1 and S2
Type	Unidirectional node-centric trust relationship
Trustor	V2X-node
Trustee	MEC service
Description	The V2X-node wants to use a service provided from the MEC (in S1 it is the V2X Node Trustworthiness Assessment Service (NTS); in S2 it is the geo-CPM Service). To do so, it needs to assess the trustworthiness level of the service provided by the MEC, with respect to its ability of providing the correct functionality.
Trust Sources	TS.1, TS.2, TS.9, TS.10, TS.11

ID	TR.16
Scenario	S1
Type	Unidirectional node-centric trust relationship
Trustor	MEC service
Trustee	V2X-node
Description	The goal of the V2X Node Trustworthiness Assessment Service (NTS) at the MEC is to maintain and update a database (AND) of known V2X-nodes (identified by their respective vehicular PKI certificates) and their respective trustworthiness levels with respect to their capacity of including correct kinematic data in CAMs and CPMs, so that they can be disseminated to the road users.
Trust Sources	TS.12

ID	TR.17
Scenario	S1 & S2
Type	Data-centric trust relationship
Trustor	V2X-node
Trustee	Data in CPM or CAM from another V2X node
Description	The in-vehicle system receives a description of a dynamic object contained in a CAM or a CPM sent by a V2X-node. To appropriately include the observation in the consolidated view of the scene, the in-vehicle system needs to attribute it a trustworthiness level with respect to its correctness. The trustworthiness level of the sending V2X-node may be available at the in-vehicle system, if a connection to the MEC and its V2X Node Trustworthiness Assessment Service (NTS) exists (S1); or evaluated by the in-vehicle system (S2).
Trust Sources	TS.12, TS.13

ID	TR.18
Scenario	S2
Type	Unidirectional node-centric trust relationship
Trustor	V2X-node
Trustee	V2X-node
Description	The on-board system maintains and updates a database (AND) of known V2X-nodes, identified by their respective vehicular PKI certificates. The system wants to determine their respective trustworthiness levels with respect to their ability to send correct data in CAMs and CPMs.
Trust Sources	TS.12

ID	TR.19
Scenario	S2
Type	Data-centric trust relationship
Trustor	MEC
Trustee	Data in CPM or CAM from a V2X node
Description	The MEC receives a description of a dynamic object contained in a CAM or a CPM sent by a V2X-node, and it needs to attribute it a trustworthiness level, to be able to use it to deliver the geo-CPM service.
Trust Sources	TS.12, TS.13

For this use case, a total number of five trust relationships were identified. Of these relationships, three are node-centric and two are data-centric trust relationships. It should be emphasized that here no trust relationships were considered that are created by trust discounting of already existing trust relationships and trust opinions, but only trust relationships which are assessed based on trust sources. So the total number of trust relationships in the trust model will be higher.

6.2.3 Slow Moving Traffic Detection (SMTD)

This use case focuses on data-centric trust, specifically trustworthiness of the real-time kinematic data received by vehicles, through V2X messages (CAMs, CPMs), that enable the detection and identification of a slow-moving traffic condition on the road in order to properly reconstruct the scene in the SMTD service located in the MEC been able to send the appropriate DENM message to the upcoming vehicle. The SMTD use case stems from the “Detection of Non-Connected Road Users” use case proposed in the standard ETSI TR 103 562 V2.1.1 (2019-12). This use case is particularly important in the context of road safety and traffic management. Indeed, in a highway scenario, a slow-moving traffic situation may occur due to roadworks or other reasons. When such a situation happens, vehicles approaching have to abruptly decelerate from high travelling speed. In addition, if the approaching vehicle has the Automated Cruise Control (ACC) activated, the driver may not be immediately ready to take over the control of the vehicle and avoid a collision. In this context, having all the vehicles in an approaching area informed of the slow-moving traffic ahead, it is relevant from a safety and traffic management point of view.

The trust relationship that must be established for the use case’s application are pointed out in Figure 6.4.

- **In-Vehicle Sub-system:** The V2X Module inside the Vehicle Computer (V2X OBU) is in charge to build the CAM messages collecting the source data of position and 3D acceleration from the GNSS receiver with IMU and to build the CPM messages in conjunction with the radar detection. The V2X module uses the (local, standalone) TAF to construct a trust opinion of the level of trust of the constructed CAM and CPM messages before send them to the Traffic Control Center (TCC instantiated on the MEC) through the 4G/5G network. We assume that the in-vehicle network is secure and provides integrity guarantees through the integration of state-of-the-art protocols such as SECMac so the V2X module needs a trust opinion on the GNSS receiver and on the TAF. The GNSS receiver is very precise (more than the one usually installed in the car). Instead, the radar is the commercial one used for ACC (Assisted Cruise Control) so it’s very poor and the precision can be affected by an error around 10%. For these reasons, we can consider the CAM messages very precise. On the contrary, the CPM messages could be affected by a significant error in positioning.

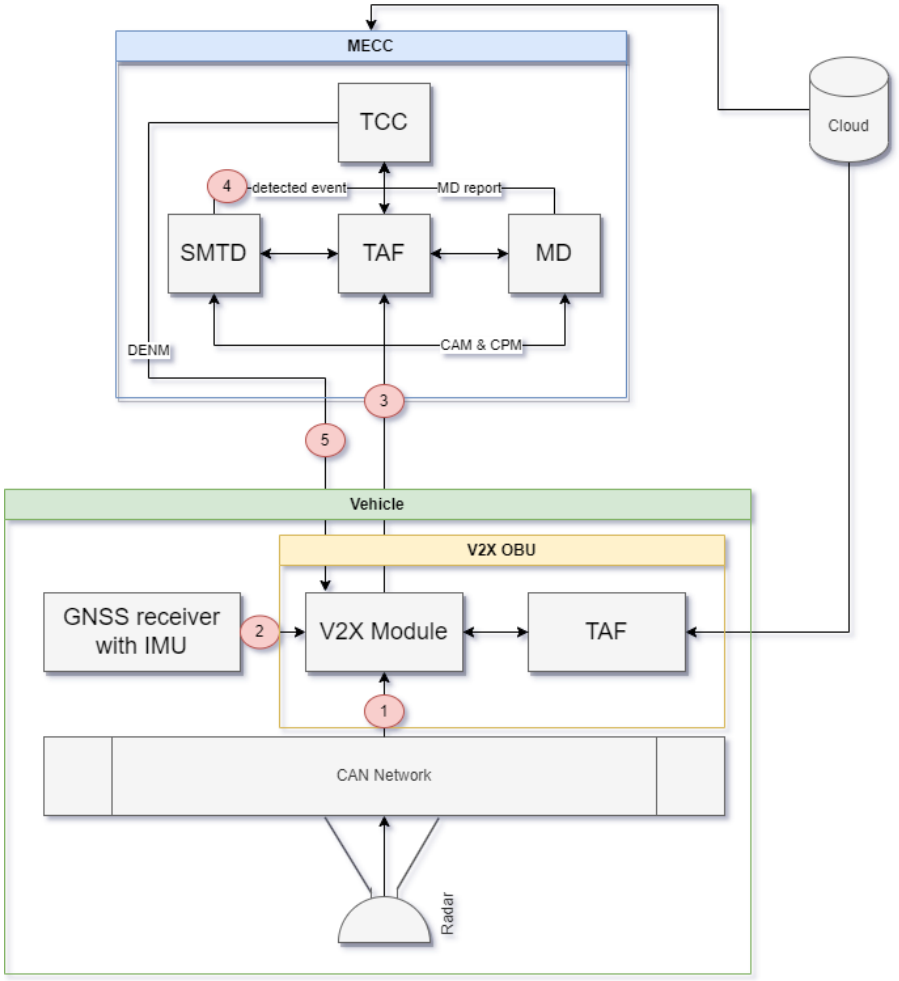


Figure 6.4: Functional relationships in the SMTD use case that lead to trust relationships.

- **MEC Sub-system:** The SMTD analyses the CAM and CPM messages in order to find a slow moving vehicle. If it detects this vehicle, it raises an event to the TCC. The TCC can then check the data integrity and authentication of the received messages thanks to the TAF. With the misbehaviour report generated by the MD is able to filter the messages with a level of uncertainty above a predefined value and ignore them. So the TCC need a trust opinion on SMTD and MD.
- **Central Sub-system:** CONNECT is based on a Kubernetes cluster, namely a set of physical or virtual machines and other infrastructure resources that are needed to run the containerized applications. Each machine in a Kubernetes cluster is called a node. There are two types of node in each Kubernetes cluster:

  ✓ Master node(s): this node hosts the Kubernetes control plane and manages the cluster

  ✓ Worker node(s): runs the containerized applications

  In the CONNECT implementation the Master Node, present in the Cloud, is in charge of the deployment of the services running in the MEC (the centralized worker node) and used by the TCC: the SMTD, the TAF and the MD. The MEC is connected to the distributed worker nodes present in each vehicle: the Vehicle Computer (V2X OBU).

The following list are examples of trust relationships that occur in this use-case and that need to be reflected in our trust models.

ID	TR.23
Type	Data-centric trust relationship
Trustor	Radar ECU
Trustee	Radar data
Description	The radar writes the radar data (2 angles and one distance) in the vehicle CAN network and these data are read by the Vehicle Computer (V2X OBU). The radar ECU needs to calculate a trust opinion on the accuracy level of the data it provides so that it can send the trustworthiness of the data to the Vehicle Computer.
Trust Sources	TS.12, TS.14, TS.15

ID	TR.24
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer (V2X OBU)
Trustee	Radar
Description	The Radar sends the position of the detected object. Therefore, the Vehicle Computer (V2X OBU) needs a trust opinion on the Radar to determine if the Radar has not changed the data produced.
Trust Sources	TS.1, TS.2, TS.3, TS.6, TS.7, TS.14

ID	TR.25
Type	Data-centric trust relationship
Trustor	GNSS Receiver
Trustee	GNSS and IMU data
Description	The Vehicle Computer (V2X OBU) acquires data from the GNSS receiver with IMU the position and the speed on the three axes of the vehicle. The GNSS Receiver needs to calculate a trust opinion on the accuracy level of the data it provides so that it can send the trustworthiness of the data to the Vehicle Computer.
Trust Sources	TS.12, TS.14, TS.15

ID	TR.26
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer (V2X OBU)
Trustee	GNSS receiver with IMU
Description	The GNSS receiver sends the position of the vehicle, while the IMU provides the speed (acceleration) on the three axes relative to the vehicle. Therefore, the Vehicle Computer (V2X OBU) needs a trust opinion on the GNSS receiver and IMU to determine if the data produced was not changed.
Trust Sources	TS.1, TS.2, TS.3, TS.6, TS.7, TS.14

ID	TR.27
Type	Data-centric trust relationship
Trustor	Vehicle Computer (V2X OBU)
Trustee	CAM
Description	The Vehicle Computer (V2X OBU) sends the CAM messages to the STDM on the MEC. The STDM needs a trust opinion on the messages sent by the Vehicle Computer (V2X OBU). Therefore, the Vehicle Computer assesses the trustworthiness on the CAMs it provides so that it can send the trustworthiness of the CAMs to the STDM.
Trust Sources	TS.1, TS.2, TS.3, TS.6, TS.7, TS.14

ID	TR.28
Type	Data-centric trust relationship
Trustor	Vehicle Computer (V2X OBU)
Trustee	CPM
Description	The Vehicle Computer (V2X OBU) sends the CPM messages to the STDM on the MEC. The STDM needs a trust opinion on the messages sent by the Vehicle Computer (V2X OBU). Therefore, the Vehicle Computer assesses the trustworthiness on the CPMs it provides so that it can send the trustworthiness of the CPMs to the STDM.
Trust Sources	TS.2, TS.3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.10, TS.12, TS.13, TS.14, TS.15

ID	TR.29
Type	Unidirectional node-centric trust relationship
Trustor	SMTD on MEC
Trustee	Vehicle Computer (V2X OBU)
Description	The Vehicle Computer (V2X OBU) sends the CAM and CPM messages. Therefore, the SMTD needs a trust opinion on the Vehicle Computer (V2X OBU) to determine if the data produced was not changed.
Trust Sources	TS.1, TS.2, TS.3, TS.4, TS.6, TS.7, TS.8, TS.9, TS.10, TS.13, TS.14, TS.15

ID	TR.30
Type	Data-centric trust relationship
Trustor	SMTD
Trustee	Event of a slow moving vehicle
Description	The TCC receives the event of a slow moving vehicle from the SMTD module. Therefore, the SMTD assesses the trustworthiness on the event of a slow moving vehilce it provides so that it can send the trustworthiness of this event to the TCC.
Trust Sources	TS.1, TS.2, TS.3, TS.4, TS.6, TS.7, TS.8, TS.9, TS.10, TS.13, TS.14, TS.15

ID	TR.31
Type	Unidirectional node-centric trust relationship
Trustor	TCC on MEC
Trustee	SMTD module on MEC
Description	The TCC receives the event of a slow moving vehicle from the SMTD module. Therefore, the TCC needs a trust opinion on the SMTD to determine if the event produced was not changed.
Trust Sources	TS.2, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.10, TS.12, TS.13, TS.15

ID	TR.32
Type	Data-centric trust relationship
Trustor	TCC on MEC
Trustee	DENM
Description	The Vehicle Computer (V2X OBU) receives the DENM message from the TCC. Therefore, the Vehicle Computer assesses the trustworthiness on the DENM it provides so that it can send the trustworthiness of the DENM to the Vehicle Computer.
Trust Sources	TS.2, TS.3, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.10, TS.12, TS.13, TS.14, TS.15

ID	TR.33
Type	Unidirectional node-centric trust relationship
Trustor	Vehicle Computer (V2X OBU)
Trustee	Traffic Control Center (TCC) on MEC
Description	The TCC generates the DENM message. Therefore, the Vehicle Computer (V2X OBU) needs a trust opinion on the TCC to determine if the DENM produced was not changed.
Trust Sources	TS.2, TS.4, TS.5, TS.6, TS.7, TS.8, TS.9, TS.10, TS.12, TS.13, TS.15

In this use case, a total of 11 trust relationships were identified. From these trust relationships, five are node-centric and six are data-centric trust relationships. As in the use cases described above, not all trust relationships are described here. Only trust relationships assessed based on trust sources are described. Trust relationships derived based on trust discounting of already existing trust relationships are not described here. So the total number of trust relationships in the trust model will be higher.

6.3 Trust Sources

As mentioned above, trust relationships describe the relationship between a trustor and a trustee with respect to one or more properties of interest. Possible properties are defined by ISO/IEC 5723 or ITU-T Y.3057 and are extended in Tab. 4.1. Examples of such properties are **accuracy, integrity, or resilience** based on which the Trust Assessment Framework can construct a proposition on the trust level of the trustee leveraging various evidence that can enable the evaluation and validation of the actual trust level of the target entity.

Evaluating and validating trust is the exercise of going through the various measures of trust applicable for a trust relationship, evaluating the levels of assurance, and if they meet the criteria set, validating the trust relationship. This exercise is carried out by the Trust Assessment Framework based on evidence, received from the trustee as sources of trust, conveying information on the status of those properties of interest that can be validated in a verifiable manner. For instance, consider the case of the envisioned "*Vulnerable User Protection through Cooperative Adaptive Cruise Control*" where such systems allow the vehicle to automatically keep a safe distance from other vehicles in front of it, in order to avoid any accidents that can compromise the safety of nearby pedestrians. In this case, C-ACC relies on data that originate from both a vehicle's internal sensors such as the lidar, radar, GNSS, and cameras as well as from kinematic data that are being transmitted from other neighbouring vehicles through CAM and CPM messages. Thus, it is evident that there is an increased need for integrity and high accuracy of the data used on making a decision about the distance between two vehicles. To prevent such incidents from happening, vehicles already have reactive security mechanisms such as Misbehaviour Detection (MD) in place. Reactive security mechanisms are tasked with detecting and correcting malicious activities and MD, specifically, focuses on detecting whether the data integrity has been compromised. However, there are ways for an attacker to trick the MD and pass the data consistency and plausibility checks which it uses. This renders the MD unable on its own to cope with ensuring that data whose integrity has been compromised is not used by a safety-critical function.

Compounding this issue, depending on the property, appropriate trust sources are considered to provide evidence for the fulfilment of the corresponding property. Decisions on trust are rarely made on a single parameter, and trust is always contextual. Thus, depending on the trust properties of interest, different trust sources are selected to create the atomic trust opinion of the trust relationship. In the example scenario above, evidence could be provided by the in-vehicle sensors on their correct configuration and operational state to have guarantees that the software responsible for extracting and processing the data was the intended one (certified application from the OEM) and was not compromised. Such evidence can be provided by the CONNECT attestation primitives been equipped in the resource-capable ECUs acting as the Verifiers of the sensors they govern.

An atomic trust opinion is a trust opinion based only on evidence in the form of trust sources and without fusing or discounting multiple trust opinions. Since the number of possible trust sources is very large and depends on the properties of the trust relationship, in the following examples a tentative list of trust sources is provided. *This list is not intended to be exhaustive and should only provide an overview of possible trust sources*. The trust sources are provided mainly for the integrity property, which is probably the most important property in the CONNECT project besides accuracy and safety properties.

The trust sources were divided into four categories. But depending on the trustee, not all categories may be relevant. The first three categories were derived based on the categorization of relevant components in ITU-T Y.3057. These are essentially the components of a node that have access to the data and could therefore compromise it. The trust sources in these categories are predominantly security mechanisms. In addition to security mechanisms, a fourth category was added that considers the behaviour of the node to get further evidence about the trustworthiness of the node.

Some of the trust sources need to be evaluated regularly because the output of these trust sources might change over time. For other trust sources, a one-time assessment at system startup is sufficient because the output does not change.

6.3.1 Trust Sources related to Communication

In this subsection, trust sources are described that provide evidence of communication. Communication refers to the communication medium used to transfer data between two nodes, for example, between two ECUs or between a vehicle and a MEC server. This communication medium should be secured to protect properties such as integrity or authenticity.

ID	TS.1
Title	Communication technology
Description	Depending on the communication technology, the difficulty of compromising the data sent over this network varies greatly. For example, a communication channel that uses a direct communication link between nodes provides much less attack vectors than a communication channel that uses a bus system. In V2V or V2I communication, where communication is mediated by routing protocols, again different attack vectors are possible. Therefore, the trustworthiness of the communication channel should vary depending on the communication technology used. Examples of direct communication links could be a sensor connected to an ECU. An example for a bus system could be CAN bus, which is common for communication between ECUs. For V2V or V2I communication, 5G or Dedicated Short Range Communication (DSRC) could be used.

ID	TS.2
Title	Protection mechanisms of communication
Description	The communication between two nodes of the network can be protected with different mechanisms that provide different security properties such as confidentiality, integrity or authenticity and thus provide protection against various attacks. Examples of such attacks could be the modification or retransmission of messages that lead to undesired behaviour of the application. In addition, depending on the protection mechanisms, the Level of Assurance against attacks varies. For example, the integrity of transmitted messages is protected to varying degrees depending on the type of signatures and key length used. Since different attacks can be realized with different efforts depending on the protection mechanisms of the communication, the trustworthiness of the communication should be adjusted depending on the used protection mechanisms used. An example of integrity protection in AUTOSAR are Message Authentication Codes (MAC). Here, a short string is generated based on a secret key and the message to be transmitted. This MAC is checked by the receiver to prevent tampering of the message during transmission.

ID	TS.3
Title	Hardware security mechanisms
Description	Hardware secure elements are physical computing chips, that are attached to the host device, and can (among other things) manage and store cryptographic keys and perform encryption and decryption functionality for cryptographic functions. These physical computing devices can be used to conduct cryptographic functions, such as creating signatures, with only authenticated and certified applications having access to the key. Since the cryptographic keys are managed by the hardware security mechanisms, it is more difficult for an attacker to get access to the keys, since the HSM provides secure storage capabilities with strict trust boundaries. Therefore, when hardware security mechanisms are used, it becomes more difficult to impersonate another node in a network. Therefore, these components are inherently trustworthy, so they can be used as the Root-Of-Trust in a system. Examples of hardware security mechanisms are Hardware Security Modules (HSM) or Trusted Platform Modules (TPM). Both techniques provide the functionalities of hardware security mechanisms described above and can be used in in-vehicular networks or on a MEC-server.

6.3.2 Trust Sources related to System Integrity

In this subsection, trust sources are described that provide evidence of the integrity of the system, such as the operating system or the firmware running on the device. In addition, trust sources are described that prevent damage and assure the integrity of the system.

ID	TS.4
Title	Secure boot
Description	Secure boot ensures that a device uses in its boot process only trustworthy software by verifying its integrity and authenticity. Thus, during the boot process, the signature of each software component relevant for the boot process, such as boot loader and operating system, is analysed. This allows the system to determine if this software has been altered or tampered with by a malicious actor. However, it should be emphasized that secure boot can only verify the integrity of the software components at boot time and not during run time. Secure boot can be used, for example, in an ECU in in-vehicle architectures. Whether it should be deployed in the corresponding ECU must be considered at design time. Depending on the existence or non-existence of secure boot, the trustworthiness of the system should be adjusted. If secure boot is implemented, this would allow the ECU to verify the integrity and authenticity of all software components relevant for the boot process. Thus, no software with a lack of integrity or authenticity would be used by this ECU. Thus, the trustworthiness of the system would increase.

ID	TS.5
Title	Run-time integrity check
Description	Based on run-time integrity checks, the integrity and authenticity of the OS or parts of the operating system, and the software stack deployed in the target device, are checked during run-time. In this way, attacks on the software stack that occur after the boot process, where integrity and authenticity checks might have been performed by secure boot, are still detected. In this way, it could be determined if the OS was compromised during run-time. Such run-time integrity checks in the automotive domain are proposed, for example, by the SAE, where the kernel code of the system is periodically checked at run time.

ID	TS.6
Title	Known OS-vulnerabilities
Description	Based on the OS version of the node and a vulnerability database, it can be determined whether there are any known vulnerabilities in the corresponding OS. The vulnerabilities usually also contain a risk value, which can be used to determine how critical the vulnerability is for the node. Based on that, the trustworthiness of the node can be adjusted. This trust source should ensure that the OS is up-to-date, so that all vulnerabilities are patched and thus the trust on the system is higher compared to an outdated OS version. There are several vulnerability databases that contain vulnerabilities of nodes in the automotive domain, such as vehicle. An example is the Common Vulnerabilities and Exposures (CVE) database, which describes vulnerabilities and their risk. Based on this risk value, the trustworthiness of the node can be adjusted.

ID	TS.7
Title	Up-to-date OS/firmware
Description	If there is a new OS or firmware version, this could indicate that there are vulnerabilities or errors in the old version, so that the data in the corresponding node is not processed correctly. Therefore, a non-up-to-date OS or firmware should reduce the trustworthiness of the entity.

ID	TS.8
Title	Authentic OS update
Description	Especially in the context of vehicles, update mechanisms of the ECUs are necessary so that a new version of the OS can be installed when vulnerabilities or bugs are discovered. For vehicles, Over The Air (OTA) updates are becoming more and more common. To prevent a compromised OS version being installed in a vehicle, a secure update mechanism could be used to verify that the provided OS update was really provided by an OEM. This mechanism ensures that no compromised OS version is installed. Such a secure update mechanism protects the system from various attack vectors, so that this mechanism should increase the trustworthiness of the node. In the context of secure OTA updates, several mechanisms can be used to secure the update mechanism. One example are cryptographic signatures, which are created by the OEM of the updates created by the OEM and can be verified within the vehicle to ensure that the updates really come from the OEM.

6.3.3 Trust Sources related to Applications

In the systems running on the vehicle or on the MEC, there are usually also applications executed that process the data and thus could also compromise the data. Therefore, in this subsection, trust sources are described that provide evidence of the application running on the system.

ID	TS.9
Title	Run-time operational assurance
Description	Based on run-time operational assurance, the modification of operations of an application by an attacker can be detected. In this way, it can be determined whether the application was compromised during run-time. This makes the realization of a broad range of attacks more difficult, which can be reflected in the trustworthiness level of the application that uses the input data. An example of run-time operational assurance is Control-Flow Attestation (CFA). This is a set of mechanisms that detect altering the flow of executions of an application and attests to another party that the control flow was not altered. Such a flow of execution could, for example, be altered by control flow attacks that capture and modify the flow of execution of a program by changing the Link Register of the program.

ID	TS.10
Title	Known application-vulnerabilities
Description	The application version and a vulnerability database can be used to determine whether there are any known vulnerabilities in the corresponding application. The vulnerabilities usually also contain a risk value, which can be used to determine how critical the vulnerability is for the overall system. Based on that, the trustworthiness of the system can be adjusted. With the vulnerability database not only the application itself, but also libraries used in the application can be checked for vulnerabilities, since they can also contain vulnerabilities which would affect the trustworthiness of the entire application. An example of such a vulnerability database is the Common Vulnerabilities and Exposures (CVE) database, which describes vulnerabilities and their risk. Based on this risk value, the trustworthiness of the node can be adjusted.

ID	TS.11
Title	Trusted Execution Environment (TEE)
Description	The TEE provides a trusted environment in which data and assets can be stored and code can be executed. The code is protected in that it cannot be viewed or modified by entities outside the TEE. In addition, a TEE allows verification that the code running in the TEE is valid. Also, access to the data and assets in the TEE can be controlled to protect the data and assets from attacks outside the TEE. Thus, integrity and confidentiality of program code and data are provided by the TEE. Such TEEs can be used, for example, inside the vehicle on ECUs so that applications can run on the ECUs in a protected environment. But TEEs could also be deployed outside in-vehicular networks, for example in a MEC server, to protect the applications running there.

6.3.4 Trust Sources related to Entity Behavior

In this subsection, trust sources are provided that analyse the behaviour of a node, e.g., in terms of the data it provides. Based on the results of the analysis, it is decided whether the node behaves suspiciously or not, so that a trust opinion about this node can be derived.

ID	TS.12
Title	Misbehaviour detection
Description	Based on misbehaviour detection, different types of misbehaving nodes can be detected. These nodes can be vehicles, but also MEC servers. To determine whether a node is misbehaving, various detectors can be used to analyse the behaviour of the node or the data it sends. Misbehaviour in this context refers to a node sending incorrect data, such as incorrect position data, so we focus on the veracity of the data. Based on the results of these detectors, the trustworthiness of the node can be increased or decreased. In the following, possible detectors are described. Each detector can either be used as a separate trust source, or all detectors can be used together to determine if the corresponding node is misbehaving, resulting in one trust source. TS.12.1 Plausibility check Depending on the type of data provided by a node, different approaches are possible to check the plausibility of the data. For example, a position value could be compared with other inputs, such as a map, to check whether the position is within a road. TS.12.2 Consistency check Depending on the type of data provided by a node, different approaches are possible to check the consistency of the data. For example, the data could be compared with other inputs from the past. For example, a position value could be compared to position values received a few milliseconds ago to verify that the provided position is consistent with the positions provided in the past. TS.12.3 Redundancy check Redundancy checks can be used when information about another vehicle is received from several nodes. Depending on the type of data, the inputs provided by the nodes can be compared to determine if one of the nodes is providing wrong information and thus is misbehaving. TS.12.4 Misbehaviour Reports Misbehaviour Reports (MR) are used in the context of V2X communication. When a vehicle receives a message from another vehicle, the misbehaviour detection system checks if the data in this message is valid. If the data is not valid, a MR is created by the vehicle running the misbehaviour detection system. The MR contains the message received from another vehicle that has activated a misbehaviour detector of the misbehaviour detection system. The MR can be sent to another node to provide evidence about the misbehaving vehicle. Based on this report, the trustworthiness of the misbehaving node can be adjusted.

ID	TS.13
Title	Misbehaviour Reports
Description	Misbehaviour Reports (MR) are used in the context of V2X communication. When a vehicle receives a message from another vehicle, the misbehaviour detection system checks if the data in this message is valid. If the data is not valid, a MR is created by the vehicle running the misbehaviour detection system. The MR contains the message received from another vehicle that has activated a misbehaviour detector of the misbehaviour detection system. The MR can be sent to another node to provide evidence about the misbehaving vehicle. Based on this report, the trustworthiness of the misbehaving node can be adjusted.

ID	TS.14
Title	Reputation based system
Description	Based on observations of a node's behaviour, a reputation of this node is established. This reputation can be build based on referrals or ratings of other nodes in the network, which are created, for example, based on the results of a misbehaviour detection system. In addition to that, the reputation can be build based on personal experience with that node. In this way, a rating of the past behaviour of a node is generated. Based on this reputation, the trustworthiness of this node is adjusted.

ID	TS.15
Title	Spoofing detection
Description	Depending on the sensors used in the nodes, various spoofing attacks are possible that can cause the sensor to produce a false sensor output. For these spoofing attacks, detection mechanisms exist that can determine if a spoofing attack is being performed. Based on the result of the spoofing detection, the trustworthiness of the values provided by the sensor should be adjusted. For example, in the context of GNSS, there are several works that use machine learning algorithms to detect spoofing attacks on GNSS sensors.

ID	TS.16
Title	Intrusion detection system (IDS)
Description	A network based IDS monitors a network of systems for malicious activities or suspicious behaviour. All malicious activity or behaviour is collected and combined to determine if a malicious activity has truly occurred or if it is a false alarm. In this way, the IDS can detect malicious entities within the network, which would make the corresponding node or system less trustworthy. Such entities could be, for example, ECUs in a in-vehicular network.

There are several approaches how trust sources can be used to derive atomic trust opinions. One approach is that an atomic trust opinion is created for each property. Another approach is that

an atomic trust opinion is created for each property and each component within a node having access to the data. In both cases, multiple trust opinions would exist between two nodes, which then need to be fused into a final trust opinion between those two nodes. However, this can be done with one of numerous fusion operators provided by subjective logic. Which approach is finally be used to calculate the atomic trust opinion out of scope of this deliverable. Also, how the atomic trust opinion is quantified is out of scope of this deliverable and will be described in D3.2.

Another important aspect to highlight is the trade-off between the richness of such trustworthiness evidence vs. the privacy implications that might have on the target object. For instance, providing all the information related to system integrity and/or system behaviour, might allow *vehicle fingerprinting*: By receiving information on the type of ECUs and sensors been integrated into a vehicle this can allow the Verifier to identify the brand of the vehicle. Assuming that additional information on the behavioural correctness of the vehicle might include evidence on the run-time execution path(s) of intellectually-protected software then this aggravates the concern revolving around privacy breaches and beyond. For instance, sharing information on behavioural analysis can lead to implementation disclosure attacks, since the receiving part will have knowledge of the exact control-flow paths following at each point in time of the execution of the software in the target ECU [13, 14].

Compounding this issue, CONNECT formulates the concept of **harmonized attributes**: Harmonized attributes are used to report on the status of the devices involved in providing a data item (or data stream) to entities outside the vehicle (other vehicles, or the MEC). They are designed to do this without compromising the privacy of the vehicle. The same set of harmonized attributes are used by all vehicles, so their use does not identify the type of the vehicle providing the data. Such harmonized attributes will be constructed by the CONNECT Trustworthiness Claims Generator based on the output of the attestation enablers executed for attesting to the trustworthiness (integrity, correct configuration, etc.) of the device environment providing the data. Details on the conversion and mapping of the aforementioned sources of trust to harmonized attributes and the crypto primitives used to enhance their integrity and privacy level are documented in D4.1 [9]

Chapter 7

Trust Assessment Framework

7.1 Functional Specification

CONNECT's Trustworthiness Assessment Framework (TAF) is envisioned as a system component whose main function is to assess the level of trustworthiness of a certain entity with respect to a specific scope. The entity can either be a **node** within the system architecture (which can be a physical node like an ECU connected to a network or a logical node like a library within a firmware) or **data** that is being exchanged between nodes (like a position within a CAM message sent from one to another vehicle).

If the respective entity is data, then the scope of trustworthiness assessment is *data integrity*. In this case, the TAF is assessing how trustworthy it is that the data has not been compromised. If the entity whose level of trustworthiness is being assessed is a node, then the scope of trustworthiness assessment is *node integrity*, i. e., how trustworthy it is that the node integrity has not been compromised. Note that a node whose integrity has not been compromised is here defined as a node whose functionalities and handling of data does not affect the integrity of that same data. The higher the trustworthiness of either data or node is, the more likely it is that the TAF will decide to trust that the integrity of that data or node has not been compromised.

The request for the TAF to assess trustworthiness, referred to as *Trustworthiness Assessment Request (TAR)*, comes from an application able to run many different functions. The TAF only reacts upon a request, and is currently envisioned not to have the ability to trigger the trust assessment autonomously. TAR is sent at a time when an application is seeking to run a specific safety-critical function, such as Adaptive Cruise Control (ACC) or Cooperative Intersection Management (CIM). The TAR is the application's main input to the TAF, and it expects the TAF to output either an *Actual Trustworthiness Level (ATL)* or a *Trust Decision (TD)*. The ATL, ω_X^{TAF}, represents the current level of trustworthiness of an entity X as assessed by the TAF. The TD is a binary decision whether or not an entity is trustworthy and is obtained after the *Actual Trustworthiness Level (ATL)* is compared to a *Required Trustworthiness Level (RTL)*. The RTL, ω_X^R, represents the required level of trustworthiness in an entity and is calculated at design-time. Both ATL and RTL consist of several numeric values indicating the amount of actual or required belief, disbelief, and uncertainty w.r.t. to the entity's trustworthiness. Each of these values ranges from 0 to 1.0, and the sum of all three is equal to 1.0, as per the principles of Subjective Logic covered in Section 5.4.

In order for the TAF to calculate the ATL or make a TD, it requires vast knowledge of the system

architecture, including which nodes are involved in which data flow. This knowledge is embedded in trust models that are stored onto the TAF at design-time and updated, if needed, during run-time. Each trust model represents a single function that an application could run. The application's TAR needs to specify which safety-critical function is queued to be run by providing an ID of the matching trust model. This lets the TAF analyse an appropriate trust model to decide which data's trustworthiness needs to be assessed for the function to be run. Moreover, a TAR needs to include a set of RTL values for each data whose trustworthiness is being assessed.

7.2 High-level Architecture

The architecture of the TAF is given in Figure 7.1. As can be seen from the architecture, the components that the TAF consists of are:

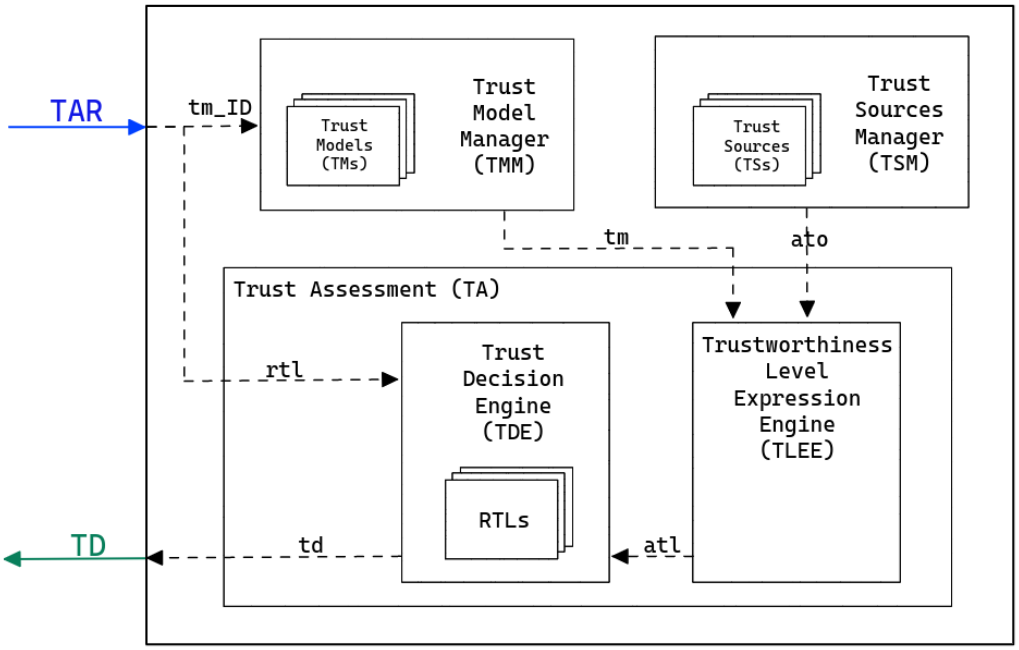


Figure 7.1: Trust Assessment Framework (TAF) Architecture

7.2.1 Trust Model Manager

Trust Model Manager (TMM) is responsible for i) selecting an appropriate, pre-stored trust model (TM) based on the TAR received from the application, and for ii) creating and updating trust models on-the-go. A TMM which is part of an in-vehicle TAF selects an appropriate trust model from a set of TMs that have been created and stored during design-time. A TMM which is part of a TAF running on a Mobile Edge Computer (MEC) will need to dynamically create and update trust models to be representative of all of the nodes present in the system at any point in time. A TM is always matched to a specific function that is being run and informs the TAF about which data the application relies upon to run that function, i.e., which data the TAF needs to output the ATL for. TMM also informs the TAF about which trust sources should be used to assess the ATL.

7.2.2 Trust Source Manager

Trust Sources Manager (TSM) is responsible for storing a pre-defined list of **all** possible trust sources (TSs) that could be used as evidence for assessing an entity's trustworthiness. Based on the input from the TLEE, the TSM will collect evidence for a subset of its trust sources during run-time. Which trust sources evidence is collected for is defined at design-time and stored inside the trust model. The TSM communicates with external components to collect evidence. Once all evidence is collected, the TSM proceeds onto quantifying this evidence into *atomic trust opinions (ATOs)* following an appropriate quantification method. There is one atomic trust opinion per one type of trust sources (e.g. one atomic trust opinion for communication trust sources, another atomic trust opinion for application trust sources). When all atomic trust opinions are successfully calculated, they are then fused together into a single opinion using a pre-defined fusion operator. This fused opinion corresponds to a single trust relationship, and how many fused opinions there are depends on the number of trust relationships inside a trust model. All fused trust opinions are then forwarded to the TLEE for further operation.

7.2.3 Trust Assessment

Trust Assessment (TA) consists of two main parts: the *Trustworthiness Level Expression Engine (TLEE)* and the *Trust Decision Engine (TDE)*. The TLEE is responsible for processing the trust model as well as the input from the Trust Sources Manager to produce an Actual Trustworthiness Level (ATL) by using appropriate Subjective Logic operators. The TDE receives the ATL from the TLEE and it compares the ATL with the RTL to obtain a Trust Decision on a certain entity. Note that the TDE also stores RTLs when it receives them from the application. We explain both the TLEE and the TDE in more detail in the following.

7.2.4 Trustworthiness Level Expression Engine

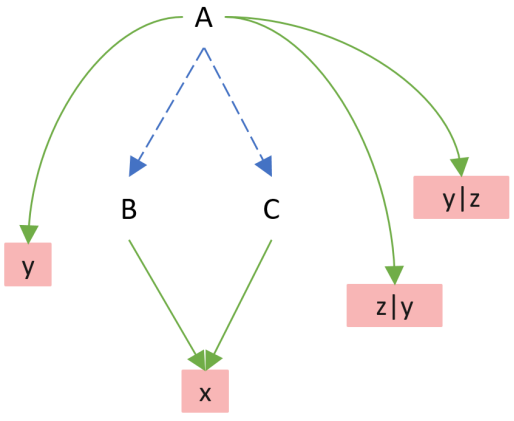


Figure 7.2: A simple trust network.

In this section, we explain the proposed architecture of our TLEE with a reference to the SL, suitable for a workflow where the level of trust of a proposition is assessed relying on a highly dynamic trust model as inputs. Our architecture fulfils the following functional requirements and supports: (i) assessing the trust of both atomic and composite *propositions*; (ii) accessing an arbitrary large *trust model*s among agents; (iii) coping with *conditional relations* between atomic propositions; (iv) allowing future extensions to work with other trust computational theories beyond SL. Please note that dealing with conditional relationships is beyond the scope of CONNECT, however we include that feature of the TLEE as part of this report since the TLEE still supports that.

Terminology. An *expression* is a sentence in a *formal language*, and it is a syntactic representation of a proposition inside the TLEE.
An *agent* is an entity whose viewpoint on the trust of a proposition is relevant and to be assessed. In essence, the agent is a node, which at the same time is the root of the trust model, also referred to as the analyst, or assessor. Depending on the use case, an agent could be for instance

a sensor, an ECU, a router or a dynamic object such as a vehicle or a robot.
A trust network consists of trust relationships between 1) two nodes and 2) nodes' direct observations of atomic propositions. Precisely, a trust network shows whether there is a *referral trust relationship* between two nodes or whether a node has *functional trust relationship* with an atomic proposition x (i. e., A has by direct experience, knowledge, or observation on x's truth in a given scope), as previously explained in Section 5.4 and Chapter 4.
Figure 7.2 shows an example of a simple trust model, where we want to calculate $\omega^A_{x \wedge y \vee (\neg z))}$—the opinion (i. e., the level of trust) that agent A has on the composite proposition x AND y OR (NOT z). In the figure, A is the agent for which the trust assessment is done, x, y and z are the atomic propositions. The benefit of our solution is that even when an agent cannot directly assess its trust on certain atomic components, it is still enabled to derive it by aggregating opinions from other nodes in its subjective trust network, i. e., trust model.

Overview of the TLEE. The TLEE receives as inputs the proposition, the agent's name, the agent's trust model (including other inputs that we explain further), and calculates an agent's opinion on a proposition.
If the TLEE receives a composite proposition as input, then as a first step the TLEE decomposes the proposition into its atomic propositions and starts building the expression, to later evaluate the agents' direct or indirect opinion on each atomic component. We elaborate this process later in this section. Also, as previously explained, during this evaluation process, when the agent does not have a direct observation on some atomic propositions, our TLEE derives the opinion through the aggregation of opinions from the other agents from the trust model. Furthermore, if the agent (e. g., A in Fig. 7.2), cannot derive an opinion on an atomic proposition (e. g., z) through its trust model, then the TLEE looks at whether there are other proposition that relate to z by a dependency relation (e. g., with y in Fig. 7.3). Since the architecture is designed to be agnostic to any trust computational theory, its modules will work at a symbolic level i. e., by rewriting the expression and by referring to *meta-operators* that will be later instantiated in actual trust algebraic operators (in our case SL operators) only when the (rewritten) expression is evaluated.

Figure 7.3: Example of representation of a conditional relation between z and y.

Inputs of the TLEE. The inputs of the TLEE are the following (see also Figure 7.4):

Proposition: The proposition whose trust the TLEE assess.

Agent Name: The name of the agent from whose perspective the TLEE does the trust assessment and evaluates the Proposition. The agents can also be fetched from the root node of the trust model instantiation per agent.

Trust Model: It is a graph, e. g., direct acyclic graph that represents the trust model. The internal nodes of the graph are agent's names and the leaves (i. e., nodes with no outbound edges) are atomic components.

List of Conditional Relationships: A list that contains information whether an atomic proposition logically implies or is implied by another (see Figure 7.3). For instance, it can be

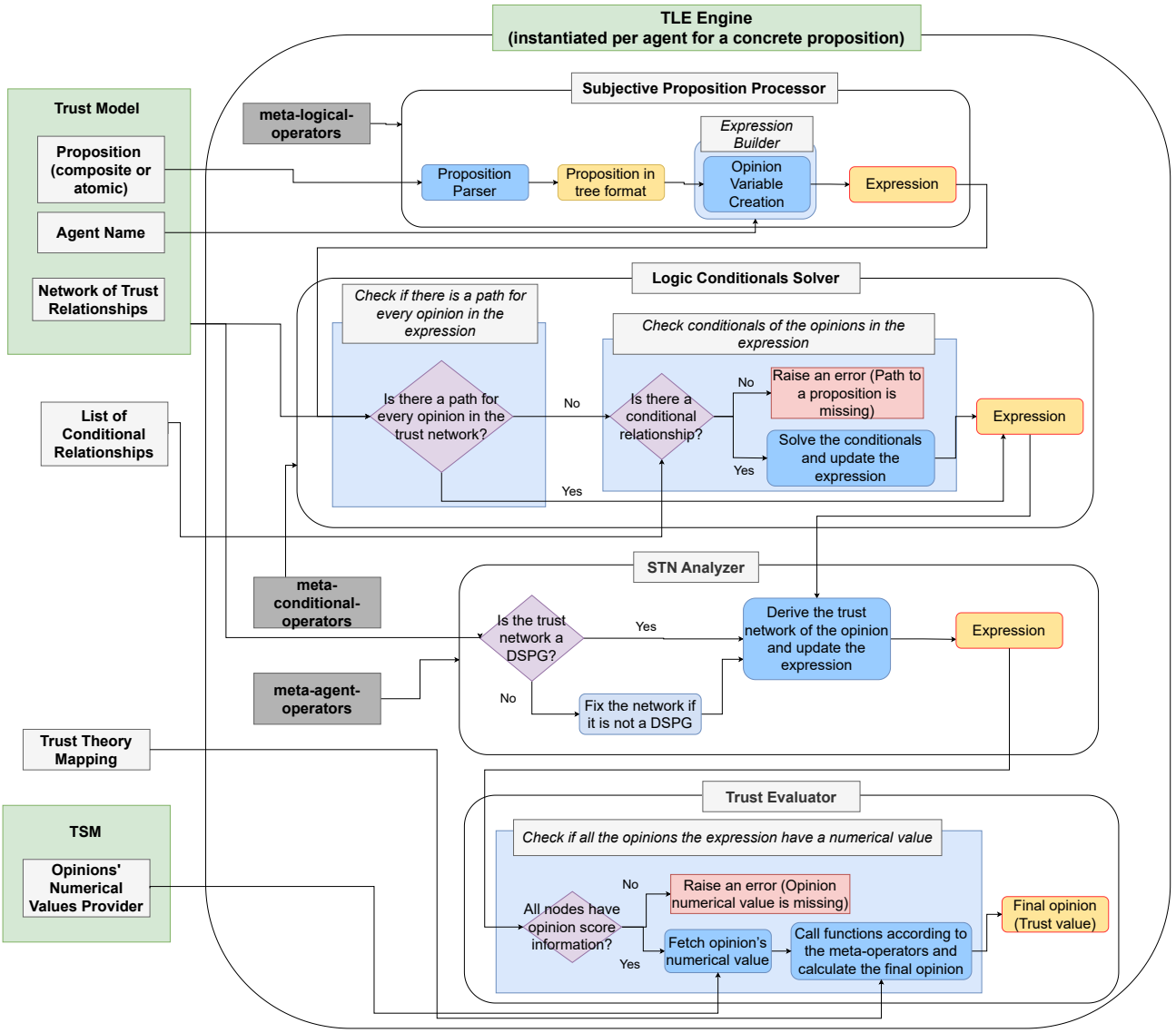


Figure 7.4: The Overall Architecture of the Trustworthiness Level Expression Engine.

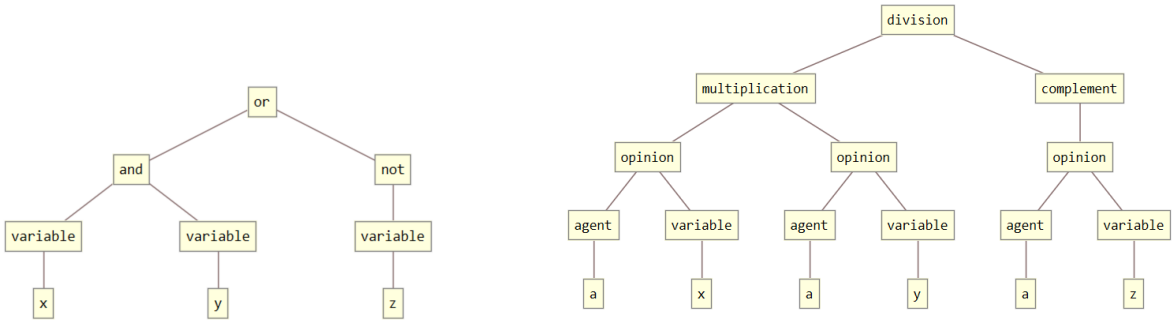
partially ordered on the set of all atomic components [1].

Meta Operators: Names of (logical, conditional, and agent) operators that the modules refer to when rewriting the expression. Those placeholders stand for operations transversal to several trust computational theories: agent operators are about composing an agent's trust by processing the opinions of an agent's peers; conditional operators relate to computing trust by derivation from expressions that have a logic dependence; logical operators are about composing an expression's trust from the trust of its atomic components. Each meta-operator has a signature that specifies the number and type of the parameters it takes.

Trust Theory Mapping: A table that links each meta operator's name to the function that implements the operators in a specific trust computational theory.

Opinions' Numerical Values: A table that stores the known numerical values of the opinions of all the atomic components that are fetched in the final step of the evaluation based on which

[1] In reference to Bayesian networks, as well as, to Markov chains, the network represents the dependencies between atomic components



(a) A proposition.

(b) Expression of agent's opinions.

Figure 7.5: Formation of SL opinion variable and transformation from proposition to expression

the final trust opinion is calculated.

TLEE Modules. The TLEE is organized by four modules, as shown in Fig. 7.4. (a) Subjective Proposition Processor, (b) Logic Conditionals Solver, (c) Subjective Trust Network Analyzer, and (d) Trust Evaluator. Modules (a)-(c) operates symbolically: they rewrite expressions. The last module calculates the final subjective opinion (also called a trust value).

Subjective Proposition Processor.

It inputs a proposition and an agent's name. The module transforms the proposition, e. g., $\omega^A_{x\wedge y\vee(\neg z))}$ (Fig. 7.5a) and starts building the expression into an *expression* of agent's opinions e. g., ω^A_x AND ω^A_y OR (NOT ω^A_z) (Fig. 7.5b). In a nutshell, in this module, the formation of the SL opinion variables takes place.

Logic Conditionals Solver.

If an opinion from the expression from the previous module (e. g., ω^A_z) cannot be derived from the agent's direct observation on proposition z nor from A's trust model, then it must be derived otherwise. In response, this module first checks if there is a path between the agent and the proposition. If no path is found, then this module tries to derive opinions from relationships that A has with other atomic components that have a relation of dependence with z (e. g., $z \mid y$ Fig. 7.2). This is possible if the dependence relation is known, and if the agent has or can have an opinion on those other atomic components, including all other elements that are required to resolve the dependence relation. As shown in Fig. 7.4, this module inputs the Trust Model, List of Conditional Relationships and the expression (from the previous module), and operates on all opinions in the expression.

Subjective Trust Network Analyzer.

This module is in charge to resolve, symbolically, an agent opinion on a concrete proposition by using the agent's trust model. It does for all the opinions in the expression that gets as input from the previous module.
Generally speaking, an agent's trust network for resolving ω^A_x is an acyclic directed graph with source A and target x (see section 7.2.4). The goal of Subjective Trust Network Analyzer is to reduce the network to a single-edged graph (A, x) and to build up an expression that embeds

the history of all the steps performed for the reduction. The reduction of a trust model requires dedicated algorithms. For instance, a trust model that is a DSPG can be reduced by using known algorithms [37]. However, the reduction should not be done independently from the trust computational theory. This module is strongly bound to the underlying trust computational theory: in fact, even if in a pure graph rewriting setting, the order of the steps to reduce the graph may not matter (due to the Church-Rosser property, according to [37]). For a specific trust computational theory that order matters when reduction steps are linked to agent trust operators that are not associative [2]. How trust model is reduced is dictated by trust computational theory of reference, in our case SL. We assume that, despite the specific graph reduction algorithm used, the analysis consists of a combination of *trust discount* (along edges that represent chains of referral trust) and *trust fusion* (across edges that represent opinion from different agents).

Trust Evaluator.

After previous steps of solving conditionals and reducing the trust model, the expression should contain only opinions about atomic components unsolved. Please note that until this step, the TLEE operates on the level of *opinion variables*. As a result, this module evaluates the expression into a trust value (i. e., the final, fused opinion with numerical values for the belief, disbelief and uncertainties) using the numerical values of the opinions given as input to the TLEE. These values are given in the input Opinions' Numerical Values Provider, which is a component that will be probably placed inside the TSM. Therefore, how these values are produced depends on the trust sources and is out of the scope of the TLEE.
To summarize, the output of the TLEE is also an (aggregated, assessed, fused) subjective opinion of the concrete trustor for a concrete proposition, and this resulting subjective opinion is the ATL that is passed to the TDE component.

7.2.5 TDE

The Trust Decision Engine (TDE) can be considered as a component where trustworthiness and trust are formed. It receives a set of ATLs in form of subjective opinions from the TLEE, and either returns those ATLs back to the application as an output, or it compares then with a set of RTLs to produce a set of Trust Decisions (TD). In our prior work [28], we have proposed comparing the ATL and the RTL by calculating the projected probabilities, P_{ATL} and P_{RTL}, of both ATL and RTL respectively. The projected probabilities were calculated using the formula (5.6) from Section 5.4.1. Once the projected probabilities are calculated, they could be compared in the following manner:

$$P_{ATL} > P_{RTL}$$

If the P_{ATL} is greater than the P_{RTL}, then the Trust Decision (TD) is positive and trust is awarded. If the P_{ATL} is smaller than the P_{RTL}, then the Trust Decision (TD) is negative and trust is not awarded. This method of comparison of ATL and RTL is only the first method we have investigated. More comparison methods are to be investigated as part of the future work.
In response to an act mistrust, there could be different actions triggered (also referred to as reaction strategies).

[2] This is in fact the case for SL.

7.3 TAF Sequence Diagram

Figure 7.6 shows the in-vehicle Trust Assessment Framework (TAF) sequence diagram for the scenario when the desired output is a Trust Decision (TD). The main actors in the diagram are: the application, the Request Manager, the Trust Model Manager (TMM), the Trust Sources Manager (TSM), the Trustworthiness Level Expression Engine (TLEE), and the Trust Decision Engine (TDE).

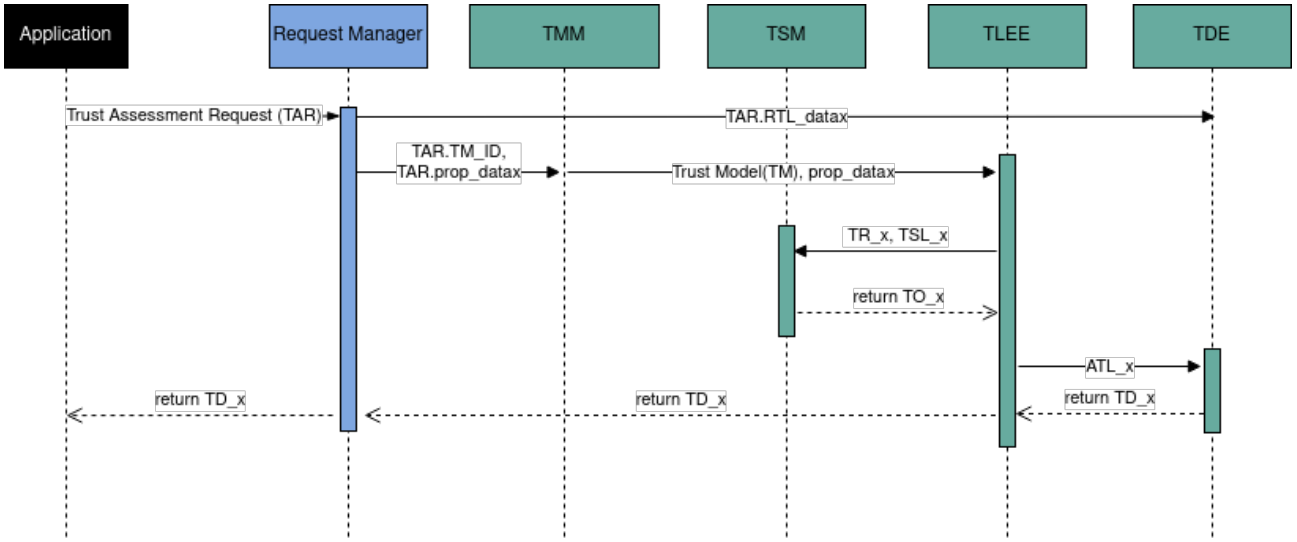


Figure 7.6: TAF Sequence Diagram

As can be seen from the figure, an application sends a Trust Assessment Request (TAR) which is received by the Request Manager. The Request Manager processes the TAR, and extracts from it the following information:

1. TM_ID: the ID of the trust model which corresponds to the function the application wants to run,
2. prop_datax: the proposition which points to the data whose trustworthiness should be assessed, and
3. RTL_datax: the required trustworthiness level (RTL) for each ATL to be assessed (needed because the trust decision (TD) is the required output).

TM_ID and prop_datax are forwarded to the TMM, while the RTL_datax is forwarded to the TDE, which then awaits input from the TLEE. Meanwhile, the TMM uses the TM_ID to locate the matching trust model, which is then forwards to the TLEE along with the prop_datax. Based on these two inputs, the TLEE identifies all the trust relationships (TR_x) for which a trust opinion needs to be build based on a trust sources list (TSL_x) which was also extract from the trust model. TR_x and TSL_x are then forwarded to the TSM which proceeds to collect all the necessary trustworthiness evidence. Once the TSM has completed collecting all of the evidence, it uses an pre-defined method to quantify the evidence into a set of atomic trust opinions. These atomic trust opinions are then fused to produce a trust opinion (TO_x) for every trust relationship that the TLEE requested. These trust opinions are then forwarded to the TLEE which uses them to produce the Actual Trustworthiness Levels (ATL_x) for each data dictated by the propositions received. The ATLs are forwarded to the TDE which then compares them to the RTLs received earlier to finally produce the Trust Decisions (TD_x) which are then returned to the application.

Chapter 8

Evaluation of Trust Assessment Framework

This chapter discusses first considerations regarding the evaluation methodology to be followed for both demonstrating the **feasibility and applicability of all variants of the described trust management framework** (by applying it in three use cases with varying levels of complexity as it pertains to the trust models to be assessed) as well as **benchmarking its performance towards enabling efficient and accurate trust-aware decision-making strategies**. As described in Section 3.4, one of the core hypotheses that CONNECT wants to evaluate is that the integration of a trust assessment mechanisms can benefit the overall safety of CCAM ecosystems and enable the transition to Day 3 phase of automated driving leveraging trusted exchange of data and sharing of vehicle control decisions. Thus, it is imperative to be able to demonstrate that the presented TAF can provide realistic trust assessment, that sufficient trustworthiness evidence can be collected and that the TAF can cope with complex trust relationships and various attacker models with a minimum footprint both on the performance and the modelling of the safety-critical function; i.e., "*Intersection Movement Assistance*" (IMA), "*Vulnerable Road-User (VRU) Protection through Cooperative Adaptive Cruise Control (C-ACC)*" (VRU-CACC), and "*Slow Traffic Movement Detection*" (SMTD). Evaluation in this context will be performed considering the specific functional specifications that the TAF needs to adhere to as listed in D2.1 [10].

Table 8.1: Evaluation Properties based on the Trust Assessment Framework Appraisal will be Conducted

ID	Evaluation Property	Description & Focus
UT1	**Generality**	The TAF should be generic enough to be applicable in different scenarios, e.g., in in-vehicle networks such as in the "*VRU Protection through Cooperative Adaptive Cruise Control*" use case or in inter-vehicle networks such as in the "*Intersection Movement Assistance*" use case. Such a generic TAF would be much more widely applicable, would reduce or eliminate customization effort for each use case, and could even be updated for use in a completely new use case. The TAF is incorporated in three heterogeneous use cases with in-vehicle and inter-vehicle networks. Furthermore, it will be instantiated in a vehicle and a MEC server. In this way, we can evaluate the universality and applicability of the TAF in use cases with varying requirements and different level of trust complexities to be considered.

UT2	**Run-time Performance**	The TAF is intended for CCAM systems that are **time and safety critical**. Therefore, the TAF should be able to assess the trustworthiness of an entity, which can be a node or data item, within the specific time constraints set by the application. The reason for this is that the decisions of the CCAM system should be made based on the output of the TAF. Therefore, the TAF must be fast enough to not cause unacceptable delays in the decision-making process which could lead to safety issues. An example for a use case, with such a time-critical decision process, is the "*VRU Protection through Cooperative Adaptive Cruise Control*" scenario where data-centric trust decisions on the *integrity* of the extracted kinematic data solicit a vehicle's distance calculation (to other proceeding vehicles) and acceleration which, in turn, dictates its decision whether it needs to stop or decelerate to avoid a crash. A "*late*" decision on the low trustworthiness of such data could lead to the vehicle calculating an erroneous distance and accelerating to the point of causing a crash with the vehicle in front of it. Nevertheless, in all use cases, **the standalone TAF should be able to assess the trust level of a proposition within a delay of at most 200 ms, whereas evidence collection and verification may take longer for complex processes**. This latter operation is calculated separately, as part of CONNECT's security mechanisms for providing trustworthiness evidence through verifiable trustworthiness claims, as will be documented in D4.1 [9]. In such cases, the application should be able to specify a maximum latency. The TAF will keep this latency by ignoring delayed evidence in the evaluation. The process of evidence generation and its duration are analysed and evaluated as part of the trustworthiness claim construction process. To analyse the run-time performance, a detailed benchmarking of the TAF will be performed considering the varying time constraints of the different CCAM functions in the context of the use-cases. For this purpose, we will also scale the complexity of the traffic scenarios (like number of cars) to investigate limits of our implementation. However, we anticipate that our prototypical implementations will leave ample room for performance enhancements. Therefore, the Proof-of-Concept evaluation will be conducted in two experimentation rounds in order to make further optimizations and adjustments.
UT3	**Scalability**	The time requirements for CCAM systems must also be fulfilled when the number of participants in the system increases exponentially. An example for this would be the "*Intersection Movement Assistance*" use case. The more vehicles that are at the intersection and want to cross it, the larger the footprint would be on the trust model, which can lead to higher computation times and a higher number of propositions to be calculated. Therefore, it is necessary to check whether the TAF can meet the time requirements for the corresponding application even if there are many participants and thus large trust models. In order to analyse the scalability of the TAF, focus will be given on the "*Intersection Movement Assistance*" use case which will be examined in a heavily loaded traffic situation, emulating a rush hour scenario. For this scenario, it is analysed how long the computation of all required propositions takes so that the IMA system can calculate its output and whether this still meets the time requirements. Furthermore, the TAF-DT is also taken into account, where the calculation of the trust propositions is offloaded to the Digital Twin of the vehicle instantiated on the MEC so that we can also evaluate the benefits that the resource richness of the MEC brings to such trust-related operations.
UT4	**Utility**	The utility describes whether the TAF provides the correct result of a proposition that an entity (data item or node) is trustworthy or not. Thus, this is the key output of the TAF. This output is provided to the application. Based on this output and defined reaction strategies, the application has to decide how to react in case of an untrustworthy entity.

		To evaluate whether the TAF provides the correct result that an entity is trustworthy or not, a scenario-based evaluation is conducted for all envisioned use cases. In the scenario-based evaluation, one scenario is created where all vehicles are trustworthy. In this case, the result of the TAF for all propositions should be that the corresponding entities are trustworthy. In the second case, one or even several vehicles (or their sensors) are not trustworthy. In this case, the results of the TAF for the propositions that include these untrustworthy vehicles should be untrustworthy. In this way, it is possible to assess whether the output of the TAF is correct, depending on the scenario is correct, and thus whether the TAF is working appropriately.
UT5	**Resilience**	There are many ways to attack the TAF itself, resulting in the TAF providing incorrect output or no output at all. Such attacks could be aimed at changing the trust relationships, the trust sources considered, or the trust opinions between two entities. Therefore, the TAF should include mechanisms to be resilient against possible attacks on its operations. Towards this direction, various attack scenarios will be exploited evaluating both the **operational assurance of the TAF itself** (as described in the overall CONNECT Conceptual Architecture, the TAF is instantiated as part of a secure container to be able to execute in a high-level of isolation so as to be safeguarded against tampering attacks that try to alter its configuration or behaviour) as well as its resilience in coping with (input) trustworthiness evidence for which their integrity cannot be verified (e.g., altered evidence on the integrity of the device by manipulating the attestation process). For each attack, countermeasures or explanations are provided why these attacks cannot be realized or are unlikely to be realized in the TAF. To analyse the resilience of the TAF, a Threat Analysis and Risk Assessment (TARA) is performed for the TAF to determine the threats and corresponding risks of the TAF. The risk values of the TARA are used to analyse whether the TAF is resilient.
UT6	**Robustness of the trust model**	Here we will focus more on the TLEE and evaluate it against attacks where an intermediary tries to change the opinions of others or the evaluation expression, resulting in a different graph. For example, if recommendations from others are malicious, an intermediary fusing its own observations with such recommendations will affect the result and pass the fake information through. Even if the intermediary is honest and does not modify the input received from others, it can still provide fake information as a result of the fusion. The right choice of the fusion operator could play a role in improving robustness against such attacks. For example, there are cumulative operators, which try to reduce uncertainty by taking under consideration the referral trust or there are consensus-based operators that are striving for consensus. This might also be dependent on the use case. So overall, the evaluation should include a) the robustness validation of the trust model itself and b) given the description of a specific CCAM scenario, recommending the most appropriate operator to choose. That is, given a design of a trust model, how can we evaluate it against a specific use case where it is applied: for example, which operator is the recommended one for this scenario and which one would produce the optimal output? This will result in some generic design patterns for the trust model in CCAM scenario.
UT7	**Flexibility in Trust Sources**	The TAF should be able to take into account a broad range of trust sources to derive the trust opinions. Depending on the use cases, trust relationships between different trust objects are possible. Depending on the trust object, different trust sources are possible, so the TAF must be flexible in which trust sources are considered.

		Whether the TAF provides flexibility in Trust Sources is evaluated in each use case by showing that at least five different trust sources are included in the use cases. These trust sources are collected so that they are verifiable through trustworthiness claims. In terms of verifiability, different approaches are evaluated where the verification of the trust sources is done. Either this is done within the TAF or outside the TAF in the attestation and integrity verification component.
UT8	**Privacy**	The inclusion of a TAF in a system could have privacy implications in that the processing of trust models and trustworthiness information and the interactions of federated TAFs and TAFs with a TAF-DT in a MEC could lead to additional privacy risks. For example, based on the trustworthiness evidence provided by a node, that node could possibly later be re-identified if that evidence is sufficiently specific to a node. Therefore, **an analysis of whether the TAF creates additional and significant privacy risks and – if so – how they should be mitigated, also needs to be investigated**. A first step towards this goal is a Data Protection Impact Analysis (DPIA) conducted via an established methodology like LINDDUN. Furthermore, another dimension to evaluate is the trade-off between **data (trustworthiness evidence) anonymization and data utility**; i.e., *how much the filtering of trustworthiness evidence into composite "harmonized attributes" - for safeguarding the privacy of the vehicle against fingerprinting - affects the accuracy of the trust assessment process.*

Chapter 9

Conclusions

This deliverable presents our first steps towards the Trust Assessment Framework (TAF) that constitutes a main artefact to be contributed by CONNECT.

To this end, we define a roadmap how to achieve different TAF functionality in a three-step process. Step 1 constitutes a standalone TAF that can model and evaluate trust relationships within a single vehicle or MEC and evaluate a selected set of trust sources for this purpose. Step 2 will provide the added functionality of federation of TAFs, enabling TAFs from different entities to interact in a cooperative system. Finally, step 3 will then introduce the notion of a TAF-DT where a digital twin of a TAF can run within a TEE of a MEC and act on behalf of the original vehicle.

Further contributions in this deliverable are the precise definition of terms and concepts that further work in WP3 and the whole project will rely on. We also survey the state-of-the-art, identify specific requirements for trust assessment in the automotive domain, and identify how a TAF based on Subjective Logic can provide superior capabilities for trust assessment compared to earlier approaches.

Next, we defined a methodology to identify and describe trust relationships and then examples of trust relationships in our use-cases are identified, and trust sources are enumerated and categorized. Based on this, a first high-level architecture for a TAF is presented and first considerations of how this TAF will later be evaluated. This provides a basis for the now starting fine-grained architecture and implementation of a first prototype. Our next deliverable D3.2 will provide this detailed refinement of the standalone TAF architecture, present the prototype, and take a first step towards a federated TAF.

Chapter 10

List of Abbreviations

Abbreviation	Translation
ACC	Assisted Cruise Control
AIC	Attestation Integrity Verification
AMS	Active MEC Service Directory
AND	Active V2X Node Directory
AP	Application Server
API	Application Programming Interface
AS	Application Server
ATL	Actual Trustworthiness Level
ATO	Atomic Trust Opinion
AV	Autonomous Vehicle
C-ACC	Collaborative Automated Cruise Control
CAM	Cooperative Awareness Message
CAN	Controller Area Network
CCAM	Cooperative, Connected and Automated Mobility
CFA	Control-Flow Attestation
C-ITS	Cooperative Intelligent Transport System and Services
CPM	Cooperative Perception Message
CSP	Cloud Service Provider
CVE	Common Vulnerabilities and Exposures
DENM	Decentralized Environmental Notification Message
DSRC	Dedicated Short Range Communication
DST	Dempster-Shafer Theory
DSPG	Directed Series-Parallel Graph
EC	European Commission
ECU	Electronic Control Unit
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
HMI	Human-Machine Interface
HSM	Hardware Security Module
IDS	Intrusion Detection System
IMA	Intersection Movement Assist
IMU	Inertial Measurement Unit
IoT	Internet of Things

LDM	Local Dynamic Map
MAC	Message Authentication Code
MEC	Multi-Access Edge Computing
MD	Misbehaviour Detection
MR	Misbehaviour Report
NTS	Node Trustworthiness Assessment Service
OBU	On-Board Unit
OTA-Update	Over-the-Air-Update
RTL	Required Trustworthiness Level
SL	Subjective Logic
SLA	Service Level Agreement
SoS	Systems of Systems
SMTD	Slow Movement Traffic Detection
SP	Service Provider
STN	Subjective Trust Network
TA	Trust Assessment
TAF	Trust Assessment Framework
TAF-API	Trust Assessment Framework - Application Programming Interface
TAF-DT	Trust Assessment Framework - Digital Twin
TAR	Trustworthiness Assessment Request
TARA	Threat Analysis and Risk Assessment
TBD	to be determined
TC	Trustworthiness Claims
TCC	Traffic Control Center
TCG	Trustworthiness Claims Generator
TD	Trust Decision
TDE	Trust Decision Engine
TEE	Trusted Execution Environment
TLEE	Trustworthiness Level Expression Engine
TM	Trust Model
TMM	Trust Model Manager
TPM	Trusted Platform Module
TS	Trust Source
TSM	Trust Sources Manager
VC	In-Vehicle Computer
VRU	Vulnerable Road User
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
WP	Work package
ZC	Zonal Controller

Bibliography

[1] Network functions virtualisation (nfv); nfv security; security and trust guidance. https://www.etsi.org/deliver/etsi_gr/NFV-SEC/001_099/003/01.02.01_60/gr_NFV-SEC003v010201p.pdf, 2016.

[2] Iso 26262-1:2018 - road vehicles - functional safety - part 1 : Vocabulary. Retrieved April 20, 2023 from https://www.iso.org/standard/43464.html, 2018.

[3] ISO/IEC 22624:2020—Information Technology—Cloud Computing - Taxonomy based data handling for cloud services. Retrieved June 7, 2023 from https://www.iso.org/standard/73614.html, 2020.

[4] Y.3057: A trust index model for information and communication technology infrastructures and services. Retrieved May 26, 2023 from https://www.itu.int/rec/T-REC-Y.3057-202112-I/en, 2021.

[5] ISO/IEC TS 5723:2022—Trustworthiness—Vocabulary. Retrieved May 26, 2023 from https://www.iso.org/standard/81608.html, 2022.

[6] Aljawharah Alnasser and Hongjian Sun. A fuzzy logic trust model for secure routing in smart grid networks. *IEEE access*, 5:17896–17903, 2017.

[7] 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.

[8] Euopean Commission Report. Cooperative, connected and automated mobility (ccam). final report of the single platform for open road testing and pre-deployment of cooperative, connected and automated and autonomous mobility platform (ccam). Retrieved June 7, 2023 from https://transport.ec.europa.eu/system/files/2021-11/Final20Report-CCAM20Platform.pdf, 2021.

[9] CONNECT. Conceptual architecture & customizable tee and attestation models specifications. Deliverable D4.1, The CONNECT Consortium, 15 2023.

[10] CONNECT. Operational landscape, requirements and reference architecture - initial version. Deliverable D2.1, The CONNECT Consortium, 8 2023.

[11] CONNECT. Connect trust & risk assessment and cad twinning framework (initial version). Deliverable 3.2, The CONNECT Consortium, 18 2024.

[12] CONNECT. Connect trust & risk assessment and cad twinning framework (final version). Deliverable 3.3, The CONNECT Consortium, 30 2026.

[13] heini Bergsson Debes, Edlira Dushku, Thanassis Giannetsos, and Ali Marandi. Zekra: Zero-knowledge control-flow attestation. In *2023 Asia CCS*. ACM, 2023.

[14] Heini Bergsson Debes and Thanassis Giannetsos. Segregating keys from noncense: Timely exfil of ephemeral keys from embedded systems. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 92–101, 2021.

[15] Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232, 1968.

[16] Jean Dezert and Albena Tchamova. On the Validity of Dempster's Fusion Rule and its Interpretation as a Generalization of Bayesian Fusion Rule. *International Journal of Intelligent Systems*, 29(3):223–252, 2014.

[17] Theo Dimitrakos, Tezcan Dilshener, Alexander Kravtsov, Antonio La Marra, Fabio Martinelli, Athanasios Rizos, Alessandro Rosetti, and Andrea Saracino. Trust aware continuous authorization for zero trust in consumer internet of things. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1801–1812, 2020.

[18] Anthony Etuk, Timothy J. Norman, Chatschik Bisdikian, and Mudhakar Srivatsa. Taf: A trust assessment framework for inferencing with uncertain streaming information. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 475–480, 2013.

[19] David Fernández Llorca and Emilia Gómez. Trustworthy autonomous vehicles. *Publications Office of the European Union, Luxembourg,, EUR*, 30942, 2021.

[20] Keno Garlichs, Alexander Willecke, Martin Wegner, and Lars C. Wolf. Trip: Misbehavior detection for dynamic platoons using trust. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 455–460, 2019.

[21] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. Part I Fundamentals of Bayesian Inference. In *Bayesian Data Analysis*, chapter 1, pages 4–29. CRC Press, 3. ed. edition, 2014.

[22] Stylianos Gisdakis, Thanassis Giannetsos, and Panos Papadimitratos. Shield: A data verification framework for participatory sensing systems. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, New York, NY, USA, 2015. Association for Computing Machinery.

[23] 5G Automotive Association; Cross Working Group Work Item; gMEC4AUTO;. Technical report on cybersecurity for edge computing, December 2022.

[24] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(03):279–311, 2001.

[25] Audun Jøsang. *Subjective logic*. Springer, 2016.

[26] Audun Jøsang and Simon Pope. Dempster's rule as seen by little colored balls. *Computational Intelligence*, 28(4):453–474, 2012.

[27] Audun Jøsang, Dongxia Wang, and Jie Zhang. Multi-source fusion in subjective logic. In *20th International Conference on Information Fusion, Fusion 2017 - Proceedings*, 2017.

[28] Frank Kargl, Nataša Trkulja, Artur Hermann, Florian Sommer, Anderson Ramon Ferraz de Lucena, Alexander Kiening, and Sergej Japs. Securing cooperative intersection management through subjective trust networks. In *2023 IEEE 97th Vehicular Technology Conference (VTC-Spring)*, 2023.

[29] Heba Kurdi, Bushra Alshayban, Lina Altoaimy, and Shada Alsalamah. Trustyfeer: A subjective logic trust model for smart city peer-to-peer federated clouds. *Wireless Communications and Mobile Computing*, 2018:1–13, 02 2018.

[30] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

[31] Johannes Müller, Tobias Meuser, Ralf Steinmetz, and Michael Buchholz. A trust management and misbehaviour detection mechanism for multi-agent systems and its application to intelligent transportation systems. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 325–331, 2019.

[32] Vilém Novák, Irina Perfilieva, and Jiri Mockor. *Mathematical principles of fuzzy logic*, volume 517. Springer Science & Business Media, 2012.

[33] John Allen Paulos. The mathematics of changing your mind. *New York Times (US)*, 2011.

[34] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[35] Ph. Smets. Practical Uses of Belief Functions. *Uncertainty in Artificial Intelligence 15. UAI99*, pages 612–621, 1999.

[36] Muhammad Sohail, Liangmin Wang, Shunrong Jiang, Samar Zaineldeen, and Rana Umair Ashraf. Multi-hop interpersonal trust assessment in vehicular ad hoc networks using threevalued subjective logic. *IET Information Security*, 13, 05 2019.

[37] Jacobo Valdes, Robert E. Tarjan, and Eugene L. Lawler. The recognition of series parallel digraphs. *SIAM Journal on Computing*, 11(2):298–313, 1982.

[38] E. Voit, H. Birkholz, T. Hardjono, T. Fossati, and V. Scarlata. Attestation results for secure interactions. Standards Track draft-ietf-rats-ar4si-04, IETF RATS Working Group, 2021.

[39] Lotfi A Zadeh and Anca Ralescu. On the combinability of evidence in the dempster-shafer theory. In *Proceedings of the Second Conference on Uncertainty in Artificial Intelligence*, pages 347–349, 1986.

[40] Lotfi A. Zadeh and Anca Ralescut. On the Combinability of Evidence in the Dempster-Shafer Theory. 2013.