CONNECT
CCAM TRUST & RESILIENCE

D5.1: Distributed Processing and CCAM Trust Functions Offloading & Data Space Modelling

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:	R — Document, report
Reference Number:	D6-01-04 / D5.1 / V1.1
Workpackage:	WP 5
Due Date:	31st August, 2023
Actual Submission Date:	24st November, 2023
Responsible Organisation:	ICCS
Editor:	Panagiotis Pantazopoulos
Dissemination Level:	PU - Public
Revision:	V1.1
Abstract:	D5.1 paves the way to the CONNECT orchestration implementation by first defining the task migration/offloading problem and then presenting a comprehensive analysis of the state-of-the-art. Derived insights suggest candidate approaches for the CONNECT implementation. Furthermore, the document explores identity management and proposes verifiable and presentation claims to verify a CONNECT entity's attributes or identity. The involved data model to hold the needed information is finally introduced.
Keywords:	Task migration & offloading, Solutions Survey, Verifiable claims & presentations, Trustworthiness evidence data model

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

Panagiotis Pantazopoulos (ICCS)

Contributors (ordered according to beneficiary numbers)

Stefanos Vasileiadis, Dimitris Karras, Thanassis Giannetsos (UBITECH)
Panagiotis Pantazopoulos, Pavlos Basaras, Manos Vasilopoulos (ICCS)
Anderson Ramon Ferraz de Lucena, Alexander Kiening (DENSO)
Christopher Newton (SURREY)

Disclaimer

Executive Summary

The document at hand gathers work along two axes i.e., the research background of the task migration/offloading problem as well as the concepts and data models needed for the secure and authenticated exchange of trust-related information (structured as security claims) based on which trust in data and/or entities is going to be assessed. On a conceptual note that correlates the two threads: The CONNECT migration/ offloading task requires for a trust relation to have been established to the involved migration/offloading target (node); trust (between any actors) is based on the secure and privacy-preserving modelling of trustworthiness evidence which, in turn, necessitates the appropriate definition of the *security structures* that can hold this information.

The first part of the document introduces the problem and discusses potential solutions. It identifies the involved parameters and surveys the corresponding state-of-the-art. Then, it classifies the identified approaches capturing their relevance to the CONNECT software entities orchestrator. Based on a brief description of an instance of the problem applied to the automotive setting and the involved requirements, the (general) criteria for the selection of the CONNECT offloading solution (to be developed in the context of WP5) are identified.

In the second part of the document, addressing trustworthiness evaluation challenges, the notions of Verifiable Credentials are described; the way they are constructed and used to confirm the identities or actor-centric attributes (e.g., configuration and operational integrity of a vehicle producing some kinematic data) of the CONNECT system modules, is highlighted. In CONNECT, we follow the *zero-trust paradigm* for establishing trust, based on which there are no inherent assumptions on the baseline trust of any actor which must be bootstrapped through the secure communication and verification of appropriate trustworthiness evidence that depict the state of the actors throughout the service lifecycle. Thus, relevant extensions to the so-far standardised structures defining the aforementioned notions are introduced to capture the dynamicity of the automotive setting. Subsequently, a comprehensive definition of the data model that CONNECT will use to capture the required trustworthiness evidence is introduced. Design principles accounting for privacy requirements shape the CONNECT choice of the appropriate data model to meet the requirements posed by the diverse data sharing cases of CONNECT.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The document at hand constitutes the first outcome of CONNECT WP5. It essentially gathers the work of two WP5 currently-active tasks; 5.1 which focuses on the trust-computations offloading decision-making and (to some extent) the work carried-out in task 5.4 which addresses the way that verifiable credentials (VC)s protect secure communications among CCAM actors.

The document adopts a systematic approach to a) the discussion of the relevant State-of-the-Art (SotA) and b) the presentation of the initial CONNECT work on the authentication/authorisation of CCAM services in a verifiable way. Those two dimensions represent the two core parts in this deliverable.

In the former case, the task offloading problem is stated (Section 2) and the relevance to the overall CONNECT concept is highlighted (with pointers to the project use-cases). The main body of the SotA analysis (Section 3) evolves around a careful identification of the so-far approaches, the introduction of a relevant classification scheme and its relevance to the corresponding software tool (i.e., the resources orchestrator) to implement any of those approaches. Subsequently, the focus is directed (Section 4) to the automotive setting, where we highlight an example and identify the specific needs of the considered environment. Based on them, we preliminary identify characteristics of the relevant CONNECT solution.

The second part (Sections 5 and 6) of the deliverable covers the concept of verifiable credentials (to shape the quantification of trustworthiness in CONNECT, see D3.1). The way distributed identifiers collectively make up VCs for the CONNECT actors, is detailed. Subsequently, the use of harmonised attributes and the selection of the CONNECT VC data model is discussed (and justified) on the basis of the CCAM needs. The relevance of the CONNECT data model to standardised approaches is also explained.

On a forward-looking note, the work presented herein will pave the way for the realisation of two fundamental CONNECT concepts:

- the task offloading decision-making and the development of the software tools (for its realisation). Central to that direction is the design and implementation of the CONNECT virtualized resources orchestrator that will enable all (container) interactions in the CONNECT automotive setting.
- the establishment of the continuous trust-aware authentication and authorisation for a vehicle and the derivation of information about their trustworthiness

Along these lines, D5.1 constitutes an important deliverable marking the first step of the WP5

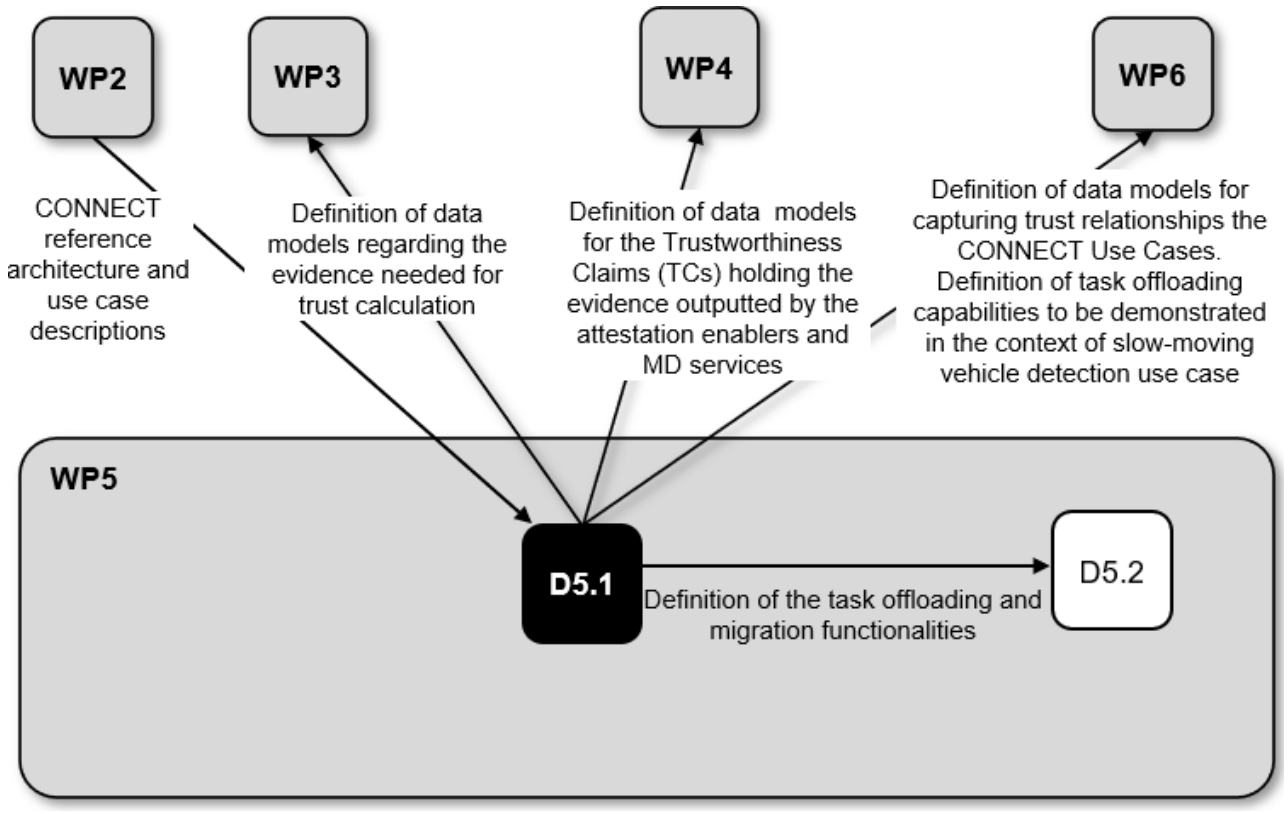

Figure 1.1: WPs relation

technical work which will offer the basic software enabler to realise the CONNECT concept and subsequently support the showcase of its effectiveness in the project use cases.

1.1 Scope and Purpose of the Document

The document covers the background of the task migration/offloading solutions laying the ground for the final identification of the CONNECT relevant solution that will be implemented in WP5. Furthermore, the document serves as a (modeling) basis for the forthcoming implementation of verifiable credentials and presentations needed for the CONNECT continuous authorisation, authentication and secure communication between CCAM actors

1.2 Relationship with other CONNECT Deliverables

The deliverable presents work that will act as the basis for the orchestration of the CONNECT task offloading process as well as the establishment of secure and authenticated communication among CCAM entities. As such, it draws on CONNECT concepts (i.e., architecture, use-cases introduction) firstly presented in D2.1 and will influence the rest of WP5 deliverables. Subsequently, links might be identified to the CONNECT integration (D6.1) and use cases analysis (D6.2). In terms of relation with other WPs, this deliverable receives input from WP2 regarding the architecture, the functional requirements as well as the descriptions of the use cases, while it provides input to WP3, WP4 and WP5. More specifically the data models provided in WP3 are focused on

the evidence needed for the trust calculation, while the data models provided in WP4 are focused on the Trustworthiness Claims (TCs) holding evidence outputted by the attestation enablers and the misbehaviour detection (MD) services. Lastly, the data models provided in terms of the WP6 refer to the trust relationships of the CONNECT use cases, while the task offloading capabilities to be demonstrated in the context of the slow-moving vehicle detection further leverage this input.

Chapter 2

The Task-Migration and Task-Offloading Problem

2.1 Problems Statement

The ever-increasing (wireless and most notably, mobile) communications technology together with pervasive computation resources (available for instance, in each point of interest -such as vehicle platform, RSUs or the edge- in an automotive setting) provides unprecedented capabilities for agile and efficient computing solutions, regardless the environment's characteristics. At the same time, emerging applications of any vertical, running in end (user) devices (e.g., from a handheld device to a vehicle-platform components) are becoming increasingly intelligent and requiring 'high throughput/low latency'. On top of that, applications that are becoming complex tend to expose broad attack surfaces and therefore, need to be capable of increased mitigation capabilities; they also increasingly rely on multiple data sources that need to be (assessed as) trustworthy, typically at a high computational cost.

The net result is that stringent requirements are posed mainly for the computational power/ resources of the involved end (user) devices over which the applications are running, and to a lesser extent to the underlying network resources. Clearly, the emerging technical *challenge* is to identify locations (whether a nearby device or more distant infrastructure) where increased availability of rich computational resources together with the necessary reliability, offer extended opportunities for efficient computations taking away the otherwise unbearable/heavy load from the user devices.

Along these lines we identify two relevant distinct instances of the above challenge and shape the corresponding problem statements that has/will become relevant in the CONNECT trust execution environments (WP4), implementation (WP5) and use cases (WP6). We use the brackets to highlight the different dimensions that are involved in the corresponding problem instances:

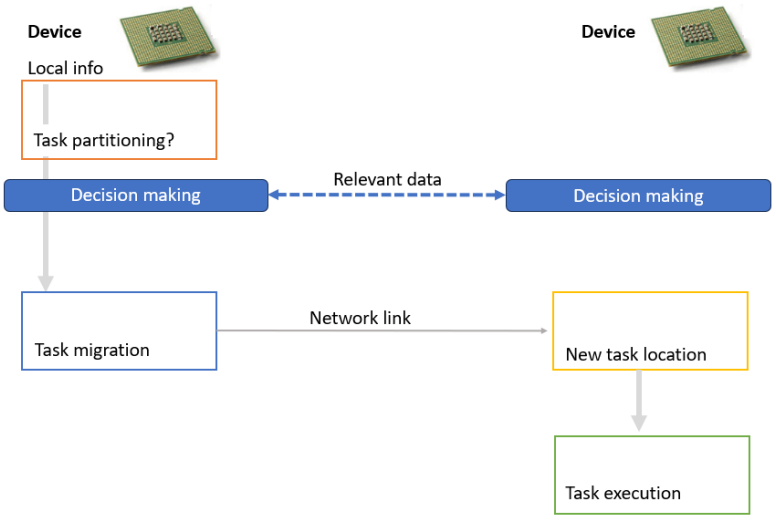

Figure 2.1: The CONNECT task migration concept

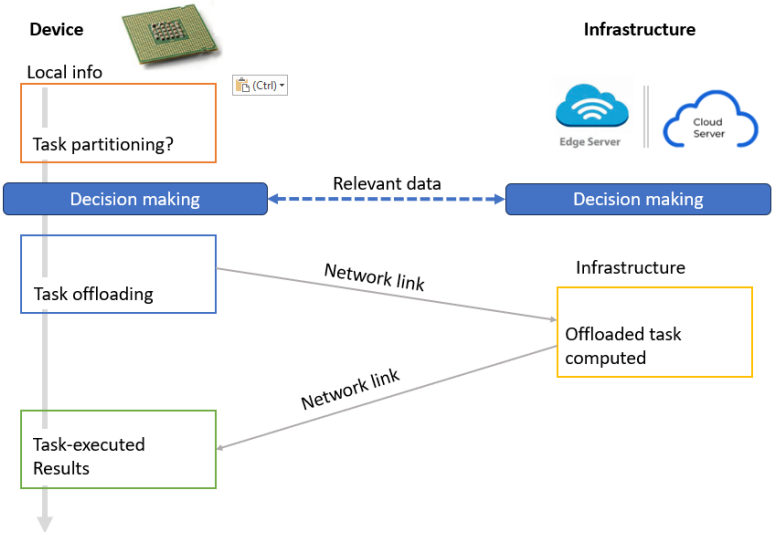

Figure 2.2: The CONNECT task offloading concept

(a) CONNECT *task migration* (see Fig. 2.1): [efficiently] transfer within a [deadline], a [part of a] resource intensive computational [task] from one hosting device/platform to another one, considered more appropriate in line with [identified] [device features]

(b) CONNECT *task offloading* (see Fig. 2.2): [efficiently] transfer within a [deadline], a [part of a] resource intensive computational [task] from a limited capability end-device to an [appropriate location] in the resource-rich infrastructure, under given network [conditions]

In both Fig. 2.1 and 2.2 we illustrate a concept phase (entitled ”decision making”) whereby relevant state information is exchanged between the involved locations. This state information may include computation, memory, energy resources or network condition information that will be used to drive/shape the corresponding task migration or offloading operation.

The highlighted dimensions (captured in brackets) will essentially be reflected as different problem formulations and corresponding solutions in Section 3. In general, both problems can lend

themselves to scheduling, resource allocation and task placement problems and be solved by a variety of approaches.

2.2 Relevance of the stated problems to the CONNECT concept

The stated CONNECT problems (a) and (b) find direct application to the realisation of the project use cases. We briefly discuss the way this instantiates in the CONNECT real-world experimentation use cases, pointing at the detailed descriptions of the D2.1. The first CONNECT use case (i.e., the Intersection Moving Assistance) is to be developed and evaluated through simulation means. The vehicle relies on the infrastructure (i.e., MEC) to be promptly notified about potential collisions with other vehicles in an intersection. MEC essentially is used to provide services that assist the vehicle to assess the trustworthiness levels associated with the observations reflected on the involved C-ITS messages. In that sense, despite the MEC involvement, the notion of task migration or offloading as captured respectively in (a) and (b) statements are not directly employed in this use case.

2.2.1 Cooperative adaptive cruise control

The second CONNECT use case involves the application of the CONNECT trust assessment framework in the cooperative adaptive cruise control (C-ACC) systems, introduced to enhance traffic flow efficiency and vehicle safety. As reported in CONNECT D2.1, C-ACC, relies on sensory data and communications to extend typical cruise control capabilities by sharing vehicle steering information with other vehicles and roadside infrastructure. Consequently, this enables coordinated traffic flows with smaller intervals in-between vehicles but at the same time poses important trust and safety challenges.

Implementation-wise, a service-oriented zonal architecture is considered in the vehicle (see D2.1) under which the main C-ACC Component is executed on an Electronic Control Unit (where information from sensors or neighbouring vehicles/infrastructure can be available). While the C-ACC operation requires the enforcement of cryptographic capabilities, (further) trustworthiness challenges arise due to the diverse data sources involved and relevant data crossing the (in-vehicle) sub-network boundaries.

The application of the CONNECT trust assessment framework is expected to benefit the C-ACC function providing a concrete way for the quantification of trust. One identified C-ACC need is to be capable of realising appropriate response policies (see CACC.US.3 in D2.1) when the CONNECT framework (which keeps monitoring the involved trust sources and verifying relevant attestation and integrity properties) suggests that the required trust level for the C-ACC-hosting ECU cannot be (further) met. This fact raises the need -in line with such predetermined policies at design phase- to securely relocate parts of the C-ACC software components (e.g., excluding parts of the key management part) from the untrusted to other trustworthy execution platforms (i.e., another vehicle ECU platform) that will allow the C-ACC to achieve increased levels of service continuity. When the platform that fulfils C-ACC's trustworthiness requirements is identified, the migration process can take place; that consists of the preparation of the new instance at the new location, the 'activation' of the original one and subsequently, the immediate switching to that

new instance. Certain verification processes are needed to ensure the whole process has been successfully concluded.

Such a migration process corresponds to the CONNECT task migration problem (a) expression in Section 2.1 (also see Fig 2.1). Clearly, the relevant computation task is the C-ACC function, the identified “more appropriate” hosting device is the new, trustworthy-proved ECU platform. Finally, the involved migration delay (see the C-ACC function downtime during migration KPI in CACC.US.3) corresponds to the deadline for the task migration in the 2.1 section stated problem.

2.2.2 Slow moving traffic detection

The third CONNECT use case relates to a collaborative automotive scenario whereby sensor readings (from equipped vehicles) and V2X communication technologies are employed to offer a Slow Moving Traffic Detection (SMTD) capability (see D2.1). The timely dissemination of information about a slow moving vehicle to the rest of the neighbouring vehicles is facilitated by a Traffic Control Center which gathers various types of data from the vehicle fleet (of equipped vehicles). Their process essentially results in sending standardised notification messages to targeted road areas, leading to minimisation of congestion, improvement of traffic flows and contribution to road safety and energy-efficiency.

The involved messages that accommodate real-time awareness, sensor data and are supplemented by the CONNECT trust metrics, are received and processed in terms of their correctness/trustworthiness/misbehaviors by the CONNECT edge services and subsequently, the trustworthy information is forwarded to a traffic control centre where it serves the needs of a high-definition map to represent (in real-time) the automotive environment. Along this line, SMTD services that are deployed in the CONNECT edge servers are capable of detecting a slow moving vehicle and accordingly (through the traffic centre) generate dedicated notification messages sent through cellular communications to all approaching vehicles (in the affected area).

In this scenario, one important (vehicle) requirement amounts to the capability of offloading resource-demanding tasks (such as a sensor-data processing task) to the edge infrastructure where appropriate digital twin instances can be deployed. Thus, the vehicle will be able to direct its resources to processing of other more (safety) critical tasks. The process maps directly to the CONNECT offloading instances (expression b in subsection 2.1). An initialisation phase (see Fig 2.2) includes security checks and potential network/state information of the involved end-points. Then, such information would act as input to the problem of identifying the edge (i.e., digital twin) location and efficiently offload the task to be computed there. Subsequently, the result of the execution should be safely returned to the corresponding vehicle application (Fig 2.2).

Chapter 3

Surveying the Task Offloading Approaches and the Implementation Dimension

This Section gathers the SotA analysis and discusses its relevance to the identified CONNECT dimensions (see Section 2.1) of the task migration/offloading problem. It introduces a mapping of the above dimensions to characteristics of the environment as well as related technologies. It also provides a brief taxonomy of solutions (that fall into the identified problem formulation choices) and comments on the way those solutions compare to the CONNECT vision. Finally, it describes (in-high level) the expected role of the CONNECT orchestrator in supporting the task offloading needs.

3.1 Solutions SotA for the CONNECT problem statement

Both stated problems in the previous section have received numerous twists (i.e., expression variants) and accordingly a vast space of solutions has been devised. An indication of the involved extremely-high interest is the survey-paper efforts. Already more than five or six recent survey-papers (and a couple of older ones) seek to systematically breakdown the so far work in the considered problem[1]. We hereafter briefly discuss the research directions that the survey works cover; the most relevant of the works presented in the considered surveys will feed the taxonomy of Section 3.2.

The work in [39] places emphasis on the MEC highlighting a number of edge computing architectures. Then, it identifies four categories for the existing offloading schemes based on the locations of the offloading end-point. A large number of different task-offloading modeling methods (capturing resources and communication types) are presented and relevant open challenges are highlighted. In [54] a careful categorisation of approaches that consider both edge and cloud infrastructure is presented, focusing on the rigorous mathematical formulation of the corresponding optimisation problems. A diverse set of approaches is classified under (combinatorial) optimisation, artificial intelligence and control theory. The survey work in [8] presents a collection of the lately-employed machine learning (ML) approaches to the offloading problem. It identifies works that fall under the reinforcement learning, the supervised/unsupervised as well as deep learning.

[1] From this point on, for the sake of convenience and clarity, we will use only the task-offloading term as an umbrella term for both (a) and (b) statements of subsection 2.1. Clearly, any concept or solution that solves the one, can be shaped to address the other.

Comparing ML-based approaches to convex optimisation solutions shows that the former may appear more efficient in terms of running time. In the automotive case, however, the required training data are hard to find due to severe confidentiality constraints.

The above survey papers gather numerous (i.e., hundreds) relevant works and provide a general view of offloading potentially in any domain. The works that place more focus on the automotive setting are rare [4] (i.e. that is yet another survey which also includes vehicle to vehicle communications being outside of our scope) but still do not account for trustworthiness properties (and continuous attestation) like the CONNECT practical implementations.

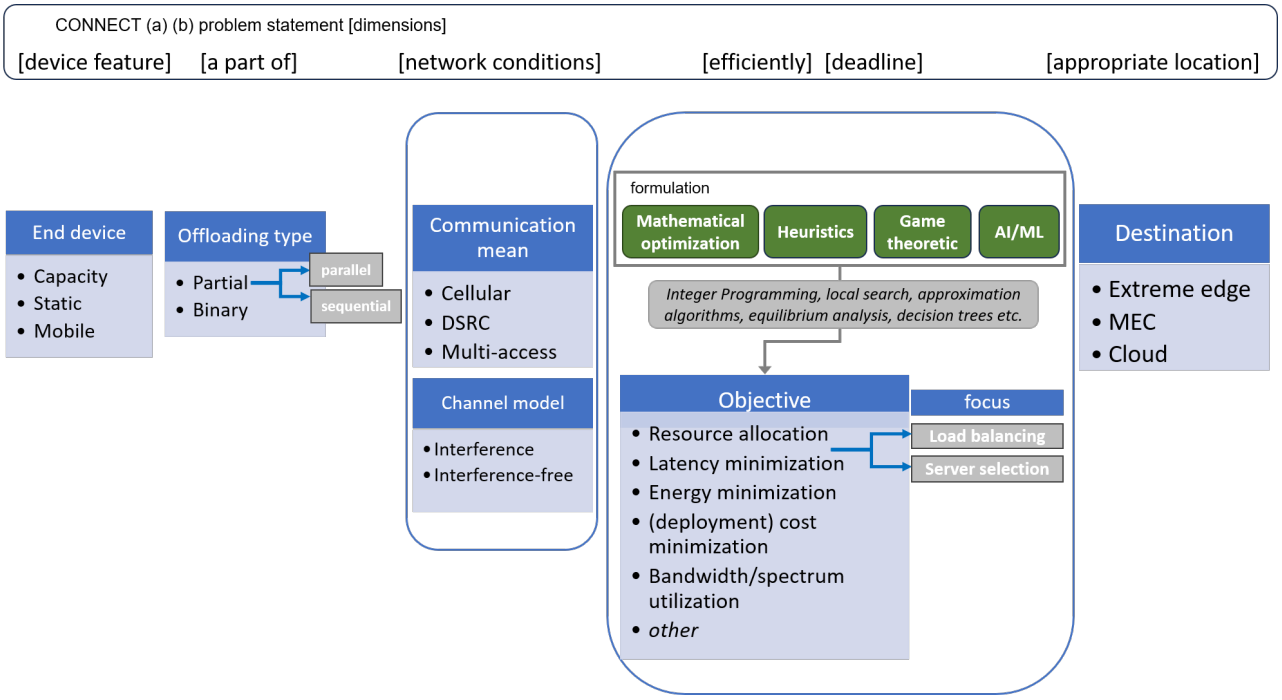

Figure 3.1: Mapping of the identified dimensions (in the CONNECT problem statement) to technology and modelling approaches

In view of such detailed collections of offloading works, we choose to avoid exhaustively go-through 'generally relevant' to task-offloading work items. On the contrary, we provide a detailed mapping (Fig. 3.1) of the identified dimensions shaping the CONNECT problem statement, to a) the characteristics of the considered automotive setting; b) the technology artefacts and tools; c)the modeling approaches employed so-far. All of them are comprehensively covered in the above literature with varying emphasis and a plethora of considered characteristics, giving shape to this large body of research/technology which is relevant to CONNECT. Finally, it is to be noted that the (four) identified problem formulations (shown in green boxes) considered as main task-offloading problem formulations, are used as drivers for the proposed taxonomy (of the selected most relevant works) in the next Section.

3.2 A task offloading solutions taxonomy

In this section we present a brief solution taxonomy of the so-far task offloading approaches. A couple of remarks are relevant here: a) the collection of works is by no means exhaustive. As

mentioned in the previous paragraph, the background in the task offloading problem is huge and its detailed collection is out of scope for this deliverable b) The focus is mainly on the objectives that are mostly relevant to CONNECT (e.g., vehicle to vehicle solutions are not included) while the categorisation of the approaches is done in line with the identified formulations in the mapping of Fig 3.1. Finally, it is to be noted that while CONNECT does not include in its original concept the AI/ML toolbox for task offloading needs, we add them in the taxonomy for the sake of completeness. Clearly, the AI/ML tools that enjoy broad usage in numerous ICT problems, have been also used for solving the considered problem.

Table 3.1: Taxonomy of (selected) task offloading approaches

Objective	Adopted approach			
	Mathematical Optimisation	Heuristics	Game Theoretic	AI/ML
Delay/RTT latency	[52], [42]	[41]	[63], [43]	[45]
Energy	[36], [18]	[41], [23]		[28], [49]
Load balancing	[15]		[20]	
Resource allocation		[26], [10]	[40], [6], [60]	[30]
(Deployment) costs	[19], [17]	[59]		[25]
Security/trust metric		[64]	[9]	

Individual details on the potential problem twists or the exact type of solution for each presented work in Table 3.1 would call for extended details that go beyond the purpose of this document. Some interesting high-level remarks are -however- gathered in the following Section.

3.2.1 Remarks on the described background and pointers to the CONNECT vision

A couple of important remarks can be put forward triggered by the characteristics of the existing solutions reflected in the proposed taxonomy. Those remarks essentially relate directly to the CONNECT vision, pointing to those directions that offer significant added value to the project.

A very limited part of the presented literature includes offloading schemes that consider security [62] or trust [65] as an optimisation parameter or constraint in the involved decision making. Likewise, employing offloading (migration) to address security incidents is limited [38]. The vast majority of the state-of-the-art employs offloading to serve the purposes of energy consumption minimization subject to execution delays. As such, a first identified point is the 'confirmation' of the CONNECT motivation to introduce a distributed trust framework that will be also used to facilitate migration/task-offloading needs.

Another remark relates to the employed evaluation approaches. A large part of the so-far solutions to the migration/offloading problem are typically evaluated through means that remain rather distant to reality; the solutions assessment may be carried-out either through analytical models [46], numerical evaluation [24] or (in numerous cases) by simulation tools [29] [33]. Testbed

experiments are scarce [55]. That means that the current state-of-the-art does lack extensive experimental results derived over real-world systems/implementations. CONNECT fills this gap with two (i.e., C-ACC and SMTD described in D2.1) of its use case implementation (and demonstration) being essentially the outcome of the CONNECT solution applied over real-world systems.

Along this same line, the literature lacks contributions that present implementation approaches (at system level) discussing the way virtualization artefacts are orchestrated [47] to manage and practically facilitate [50] the task off-loading concept; that is what will be realised by the CONNECT resource orchestrator, as highlighted in the following paragraph.

3.3 The role of the virtual resources orchestrator in offloading realisation

In line with the generic task offloading statement (in 2.1), certain demanding computational tasks or services in the the context of cloud/edge computing, can be transferred from one computation resource (often local, e.g., an IoT node, a mobile device, or, a connected vehicle) to another resource (often remote, e.g., a cloud or edge server). The latter usually exhibits more (in number) or dedicated resources capable of handling the workload (and meeting the involved requirements) of a particular task or a set of tasks.

From this point of view, the orchestration engine, is a crucial part of the system for the offloading process, as it handles the management, coordination and optimisation of virtual resources (CPU, GPU, Storage, RAM, networking etc.) across a distributed cluster of resources composed of physical and virtual worker nodes. Particularly, the CONNECT orchestrator will exploit kubernetes (k8s) [2], which is an open-source system for automating deployment, scaling, and management of containerized applications, including also the toolkit (i.e., the algorithmic approach and necessary telemetry) that will provide the offloading decisions, and implement them on the cluster nodes, e.g., at the edge/cloud.

In more detail the resource orchestrator (RO) keeps track of the cluster's *resource usage*. The functionality of resource orchestration is distributed across various components in k8s including etcd, various controllers (ReplicaSet, Node, Service), the scheduler, kubelet, proxy (Figure 3.2). Particularly, the various k8s agents (e.g., kubelet or other native/custom telemetry modules), installed on each worker node of the cluster provide continuous monitoring and reporting of the status, health, and performance of virtual resources and services (e.g., via k8s Metrics Server [3]). Based on the needs of the offloaded service (e.g., network/compute latency), and the current status of the available resources, the RO decides where and how a task should be executed. This includes for instance selecting a cluster node with GPU capabilities, or adequate free CPU, RAM and storage capacity, specific security related constraints (e.g., attested software at the target location), while also maintaining the service requirements in terms of e.g., application layer latency. Another task handled by the RO is *resource scaling*. For example, when more vehicles request concurrently the same (edge) service the RO can provision additional virtual resources (e.g., add CPUs) to the respective container/service to efficiently handle the higher service demand and de-provision them when they are no longer needed. This facilitates a dynamic and scalable environment for seamless service execution. *Load balancing* is also part of the RO functionalities ensuring that no single resource (e.g., CPU) of a host node is overwhelmed with too many tasks, which could degrade the performance of all hosted services. This entails scheduling and migrating services to different host in the cluster to avoid overloading the resources of a single

worker node. Finally, RO manages also *automated recovery* of failed services. In other words, if a service/container fails during run-time (e.g., memory error/crash) the RO can re-establish the task on another resource or take corrective/proactive actions to ensure the service continuity.

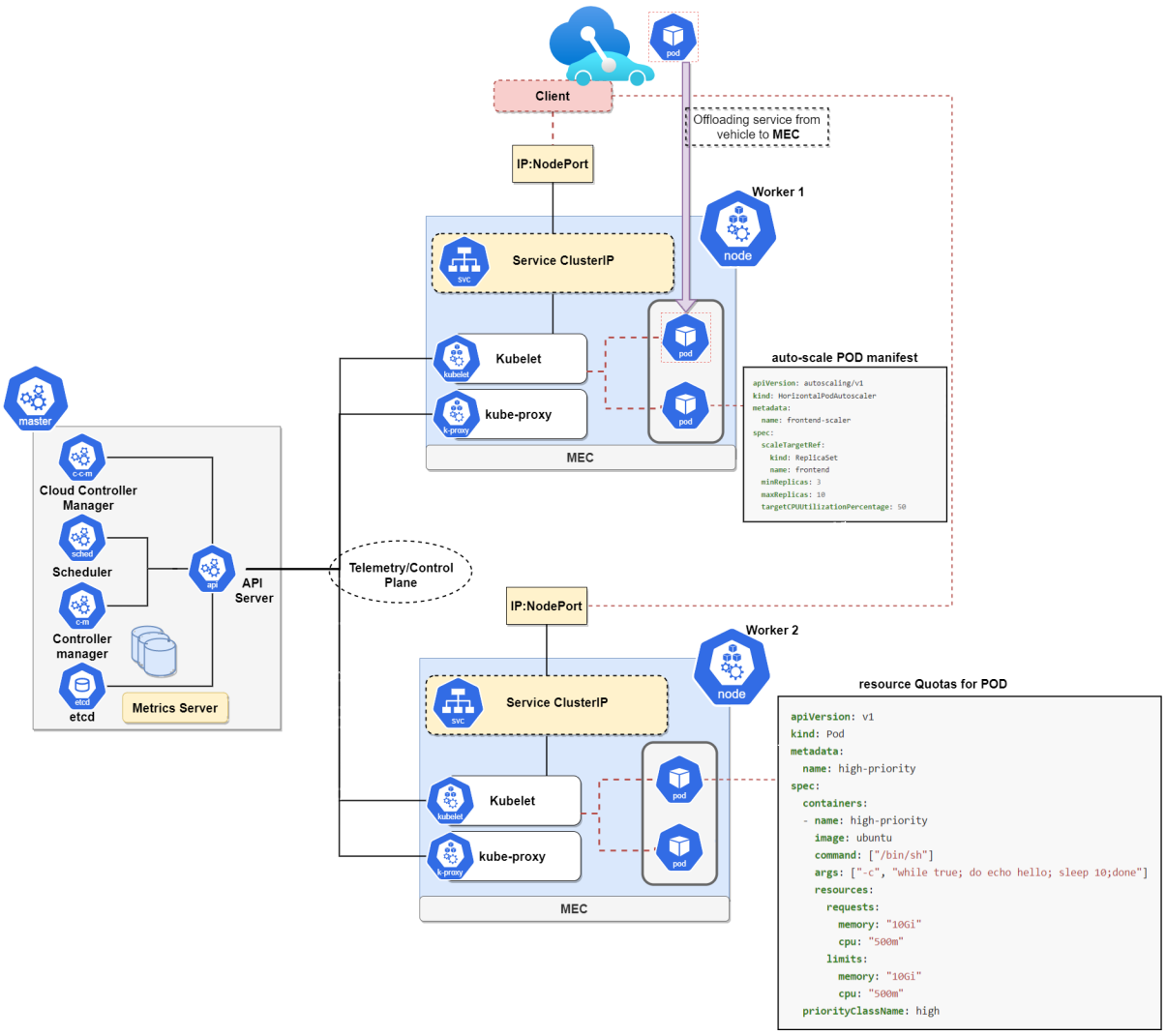

Figure 3.2: Kubernetes (baseline) architecture for facilitating POD control, lifecycle management operations and resource control. Offloading services to the MEC from client/vehicles will be facilitated through custom telemetry and monitoring to ensure that the POD and service requirements are met.

Note that the above mentioned services are basic functionalities that are facilitated via the RO. Given the requirements of a specific candidate service for offloading, additional functionalities may be needed. Figure 3.2 illustrates a toy image for the RO (with example POD manifests for auto-scaling and resource quotas configurations), including k8s agents for control and data plane exchanges. For more details please refer the official k8s repository [2].

In summary, the discussed orchestration engine and basic functionalities mentioned above, present a virtualization technology suite with the ability to efficiently handle containerized applications at scale, manage configurations and deployment options, perform load balancing and auto-scaling operations in a dynamic manner (e.g., move applications across hosts or offload applications from a vehicle to the MEC), exploit cluster native telemetry (including vehicle and MEC data) for monitoring the end-to-end services and associated lifecycle management operations triggered

via telemetry monitoring, ensuring service continuity (e.g., ”make before break”) tailored to the CCAM ecosystem and services. Details of the architecture for the CONNECT system (detailing the interconnection points and functionalities between vehicles and the MEC) and the CONNECT orchestrator can be found in D2.1, whereas the focus of this Section is on the way that the CONNECT orchestration engine will adhere to the offloading principles.

Chapter 4

A closer Look to the Automotive Setting

4.1 A use-case example

We now turn our focus on CONNECT setting and seek through a short yet indicative result to shed light on the important details (specificities) of the stated problems (see Section 2.1) when addressed in the automotive environment.

From the automotive perspective, task offloading can be a good strategy, e.g., for optimising the task needed to be carried-out inside the vehicle [48, 35]. Theses tasks can be migrated within the vehicle, by shifting a task from one ECU to another ECU with similar processing power, or occasionally to the infrastructure outside the vehicle, such as a multi-access edge computing (MEC) [51]. Another interesting strategy would be to mitigate cyberattacks on a vehicle by task offloading, providing a faster response to some threats at run-time, avoiding the long wait for updates from the manufacturer. In this Section -without loss of generality- the threat mitigation strategy (i.e., task migration) is discussed as an indicative use case.

To exemplify the use of task migration for attack mitigation, is illustrated in Figure 4.1-I a simplified Cooperative Adaptive Cruise Control (C-ACC) architecture, where the ECU A is responsible for running the C-ACC main component, as well as receiving data from the sensors and V2X messages. Consider that the ECU A is under attack and its actual trustworthiness level is lower enough to consider that it can not run this application anymore (Figure 4.1-I). From this, we can consider two options: stop running, disable the C-ACC function and wait for some update from the manufacture, which can take intolerable time, or to migrate this task to another available ECU and keep the function running on the new ECU B (Figure 4.1-II).

The task migration requirement and characteristics are going to be addressed in more details in the section 4.2, but, briefly observing this use case, the C-ACC function can be migrated from ECU A to ECU B, which is capable to run this task (Figure 4.1-II). Regarding the migration process, it is necessary to identify which components are migratable or not, e.g., TLS and its keys, C-ACC main function binary and SOME/IP client are migratable, on the other hand, the MACsec may not be offloaded, so it will need to be re-established after migrating to ECU B.

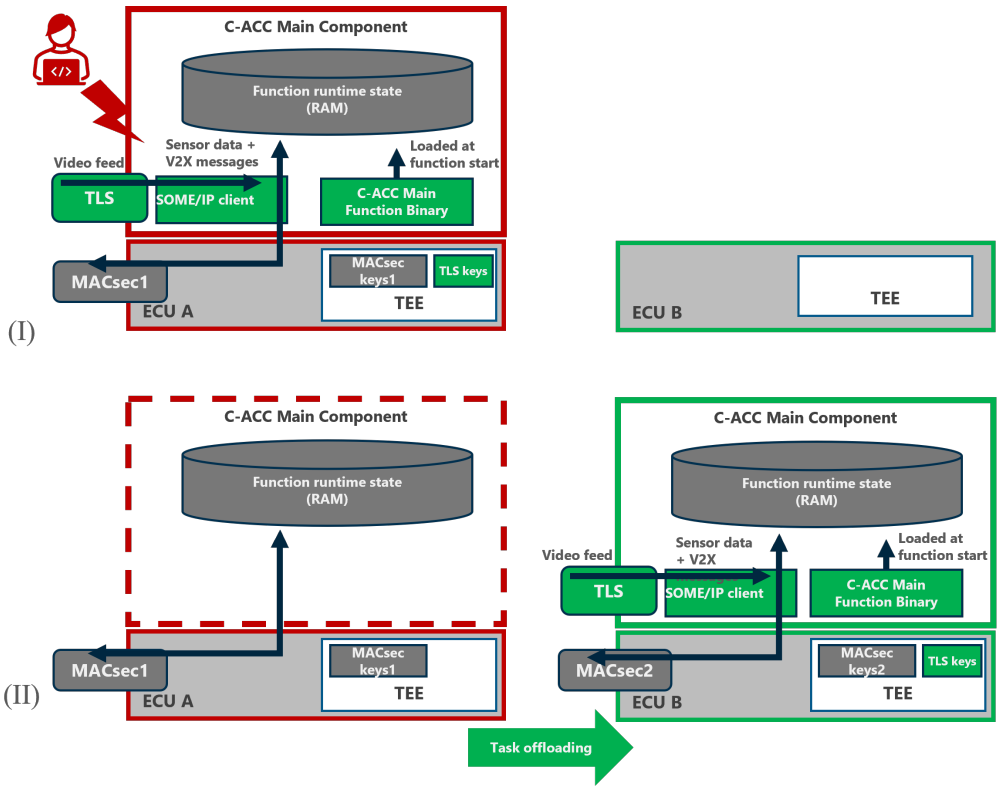

Figure 4.1: (I) Attacker compromises C-ACC function in the ECU A, causing malfunction affecting the vehicle's security. ECU A is no longer secure to run the C-ACC application. (II) The C-ACC function and its migratable parts (in green) migrated to ECU B.

After the task migration process, ECU A can be isolated, keeping the overall system running safely and with less impact than stopping the function. If only the C-ACC is compromised on the ECU A, only this component can be isolated, keeping the ECU A still running, considering that the rest of its functionalities are still trustworthy.

4.2 Requirements and characteristics

Task offloading is an effective strategy in terms of the provided flexibility in task processing; it copes with the dynamism of transferring responsibility for executing a task to another component (when necessary). Before the migration to another component and the uptake of the relevant responsibility, some remarks need to be taken into account:

- **Where to migrate the task:** When deciding to migrate a task to another component, the capability of this host component to perform the given task must be taken into account. For this, it is necessary to analyse hardware compatibility, processing power, cybersecurity aspects and the trustworthiness level for processing this task. The task may be offloaded to an in-vehicle component, such as another ECU, or to an external component, such as a MEC. In all scenarios, factors such as access to sensors and actuators required for the successful completion of the task must also be considered. Observing the example previously introduced in Figure 4.1, the choice of ECU B is of fundamental importance for the success of the task carried out. Considering the C-ACC application, this new ECU must have hardware compatible with ECU A, possible access to the sensors used for C-ACC function, such

as camera, GNSS and radar. Another important aspect is defining how the integration will take place, especially when there are some technologies that require previously established data, such as cryptographic keys or certain protocols.

- **Which parts of task are migratable or not:** in a complex system, a simple task might make use of several components and data. Integrating these components into the host is of fundamental importance for the success of offloading process. However it must be observed and classified in advance which of these components may or not be migrated together with the migrated task. Some characteristics shall be previously studied, especially in cases of attack mitigation. Then, challenges such as the state of the stored data (i.e., still intact) and the(ir) respective traustwothiness are of importance. Furthermore, the extent to which some data are exclusively tied to the hardware and may not be migrated (e.g., cryptographic keys used for communication protocols) needs to be determined. Looking at the Figure 4.1 again, the green components represents theses migratable parts, e.g. TLS and its keys, SOME/IP client, and of course, the C-ACC main function binary, which is the application itself. On the other hand, a new MACsec security mechanism must be set up, including new keys. The current status of the task can also be migrated, but it depends on the application; in the considered example the current status was not relevant.
- **Integration into the new architecture:** Still linked to the previous topic, it is necessary that the new host to be able to perform the offloaded task in an equivalent way to the old host; without affecting its current work demand or enforce any kind of task prioritisation (if applicable). Access to old sensors and actuators shall be maintained or replaced while established protocols shall be reconfigured and adjusted for the new host component. For example, for C-ACC (Figure 4.1), it is necessary to use TLS to transfer videos from the camera, as well as MACsec to transfer and receive sensory data or V2X data. The new host component shall have the same or equivalent characteristics, as well as access to the necessary data. That may require the creation of new interfaces or give access to sensors attached to the host component, for instance.

4.3 Characteristics and selection criteria for the CONNECT task-offloading approach

To identify the criteria that are to be applied in order to have an informed decision of the CONNECT solution, we draw on the above automotive result somewhat generalising the raised points to characteristics (i.e., 'requirements') any task migration/offloading task should meet:

- the new host to have similar computation capabilities and available resources as the original host
- the new host to avail similar access to information (i.e., needed input data for the task) as the original host
- capability to transfer data to the new host before the migration/offloading, if needed
- the new host to be able to execute the task under the same computing requirements (typically, to be at least of equal efficiency as the original host) and a given deadline

- minimise any impact on task execution policies (i.e., enforce priorities) of the new host, upon arrival of the task
- minimise any reconfiguration of involved protocols to the task execution, at the new host

Some of the above points when applied to the CONNECT case, may become trivial. That can be justified since a) the task offloading to the resources-rich infrastructure (i.e., from vehicle to MEC) is the mostly relevant case and thus the concerns for adequate resources at the destination (location) are typically overcome. MEC resources are numerous times more than the ones in the vehicle); b) the virtualisation/orchestration technologies allow for efficient access to distributed data sources and coordination of network management decisions (which relate and essentially ease the offloading task). As such, monitoring needs, data acquisition challenges or interoperability concerns are expected to be minimal.

Focusing on the realisation (and essentially orchestration) part of the expected CONNECT solution, we further identify some characteristics to be combined with the above ones. Orchestration decisions need to be taken based on a consistent snapshot of all available network resources. Another point relates to availability: Occasionally, the single-point-of-failure of the control plane, a multi-master orchestration deployment may be needed. Ideally, the control plane of such clusters is distributed across a set of dedicated nodes. This setup is often referred to as a ”High Availability (HA)” Kubernetes cluster, were multiple master nodes enhance the resilience and reliability of the cluster (in case of master node failures), making it well-suited for critical production workloads, and stringent CCAM service requirements. Finally, the time requirements need to be carefully considered. As most applications pose hard/soft real-time requirements (e.g., related to involved encryption functions), the container run-time [2] needs to support these; and the orchestrator needs to be able to configure the scheduling parameters based on the relevant deployment requirements.

With the above points to be kept and checked against the CONNECT orchestrator, at the development phase, we proceed with a preliminary characterisation of the candidate solutions. The CONNECT task offloading mechanisms developed in a real-world system to be applied in the slow moving traffic detection use-case (see paragraph 2.2.2) will be shaped by the corresponding environment and the relevant scope. For instance, any radio network conditions (affecting any reliability metric) and their impact on the task offloading are left out of our scope. As such, our attention will lie on three dimensions which will shape the selection of our approach. a) delay/RTT requirements for the offloading and sending back the results; b) the resource allocation at the MEC location where the task will be offloaded-to; c) any security mechanism that may require the exchange of data before/along the offloading process (e.g., handshakes, authorisation/authentication data, encryption keys etc.). One final dimension is the mobility support which is (for the needs/characteristics of our use case) essentially reflected in the first one.

Considering the above points as criteria for the selection of the CONNECT approach, the straightforward choice for a CONNECT agile task offloading solution is to resort to the heuristics toolbox (e.g., [34]) instead of any rigorous optimisation approach [54]. The main motivation towards that decision is the welcome characteristics of the heuristics class. They typically pose the least of requirements both for input data and their computation needs (CPU/memory), they are characterised by simplicity of development and can easily become compliant with the checks of the -previously identified- host needs.

Chapter 5

CONNECT Verifiable Credentials (VCs) and Verifiable Presentations (VPs)

An additional dimension behind the consideration of task offloading capabilities in CONNECT, is to support the offloading of possibly resource-intensive trust calculations that need to be performed for constructing the local trust opinions on the vehicles' side. This is part of the overall *federated trust assessment* architecture (detailed in D3.1 [11]) where reasoning about trust in data or entities is based on the fusion of both *direct evidence* (trustworthiness evidence extracted from the deployed security controls; e.g., attestation enabled, misbehavior detection services, etc.) and *indirect evidence* obtained via referral paths of the ecosystem on the reputation of the source entities. These essentially capture relationships that have already been established and can yield additional information on the behaviour of the system. However, this requires appropriate mechanisms for the trust-aware continuous authentication of the entities and the secure exchange of all this trustworthiness evidence - either between the vehicles themselves or the vehicles and the MEC-instantiated service.

In this context, the evidence based on which the trust assessment process will be performed need to be expressed in a formal verifiable manner. Which, in turn, necessitates the design of appropriate cryptographic protocols that are capable to encode them as *attributes* to be exchanged as part of Verifiable Credentials. Some attributes can be public, but others should be treated in a privacy-respective manner, as they can contain sensitive information. Also important is to have some flexibility in who can generate these statements, to have issued claims (i.e., credentials that are provided by a trusted issuer) or self-made claims (also called attestation evidence that the entity creates itself and that typical have a short life span). In what follows, we start by presenting the concept of such Verifiable Credentials and their use in CONNECT for supporting the secure and authenticated exchange of trust-related data between participating entities and detail their structure. These credentials are aligned to the well-established Self-Sovereign Identity (SSI) standards so as different trust levels can be supported representing different trustworthiness and assurance levels for the participating CCAM entities.

It serves as a predecessor to the definition of the exact type of attributes that need to be modelled as part of the trustworthiness evidence, that need to be embodied in such VCs, that will be presented in Chapter 6.

5.1 An Introduction to Verifiable Credentials

Verifiable Credentials (VCs) are statements that an entity ”the holder” has the specified attributes, or properties. In general the VC will contain an issuing authority’s signature that can later be verified to confirm its validity (hence verifiable). The holder may have many VCs and can use these, or a selection of them, to create a verifiable presentation (VP). Which VCs are included in a VP is under the control of the holder.

VPs and VCs are being standardised by the W3C [61]. As defined, they provide a convenient structure for reporting on system attributes that are assigned as the different components of the vehicle (or MEC) are built, installed and configured. While using the same basic structure we will be extending it to allow the inclusion of self-issued credentials used to report on dynamic assessments of their trustworthiness. Trust in an entity will not only be derived from the fact it was correctly assembled and configured (static attributes) but also from enhanced reporting on dynamic assessments of its different components.

We begin by describing the W3C model and how Verifiable Credentials (VC) are currently used and defined. We then go on to describe how CONNECT plans not just to use the current definitions for VCs, but to extend these to allow them to be used to support the trustworthiness claims of vehicles providing information to other vehicles and the MEC.

5.2 W3C Verifiable Credentials and Verifiable Presentations

In CONNECT we will be considering different entities, such as vehicles and MECs that wish to confirm their identities, or that they have particular trustworthiness attributes. However, in this W3C description we use the more familiar case of human subjects acquiring credentials that allow them to confirm, for example, that they have a driving licence. One of the building blocks

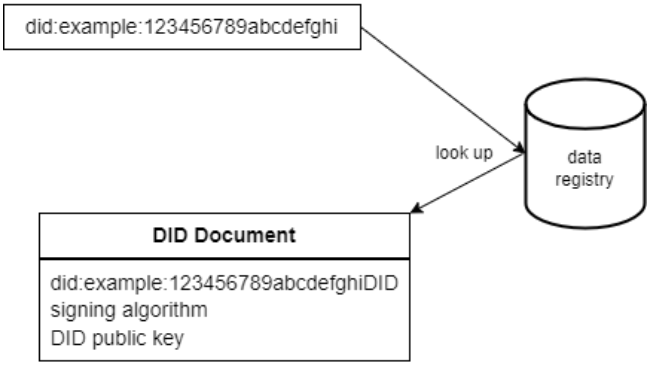

Figure 5.1: DID resolution

for the VC are decentralised identifiers (DID), these can be used to uniquely identify the different actors involved in generating a VC.

DIDs are unique identifiers that in a similar way to URLs resolve to a DID document that contains the data necessary for the holder to prove that they own that identity (see Figure 5.1). The DID document is stored in a data registry that should be universally available and tamper resistant, it is often implemented as a distributed ledger.

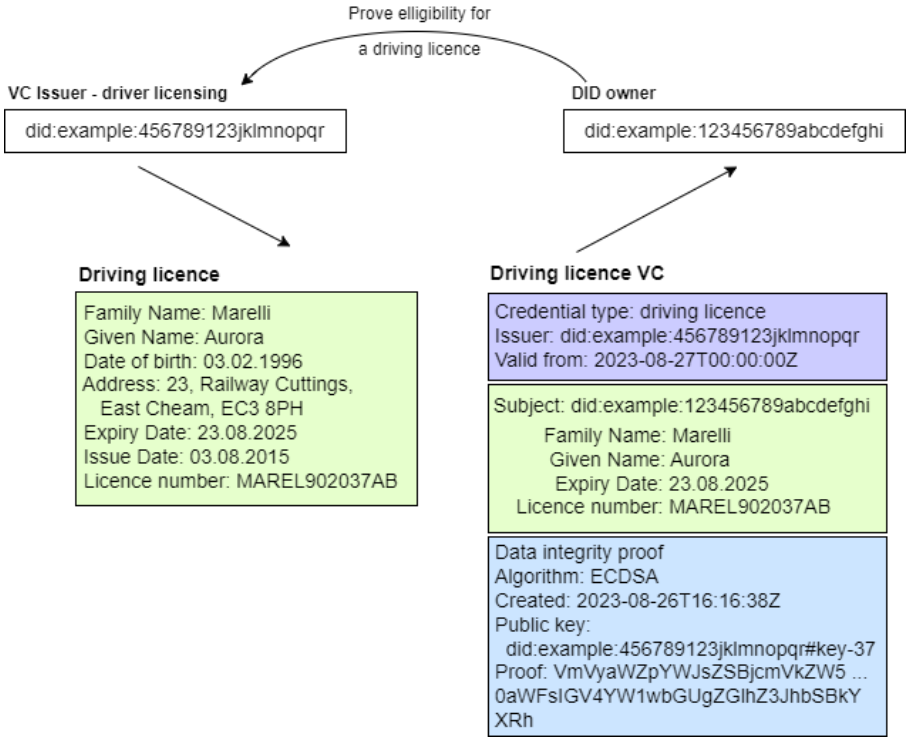

Figure 5.2: A VC used to confirm that the holder has a driving licence

At its simplest, the holder of the DID will generate an asymmetric key pair and the public key will be included in the DID document together with details of the algorithm being used. When challenged to confirm ownership the holder can use their private key to sign a challenge and this can be verified by the challenger using the given public key. DIDs are being standardised by the World Wide Web Consortium (W3C) [53].

A VC will be provided by an issuer who confirms the identity of the holder and checks that they also have the necessary credential (for example, that they actually do have a driving licence – this is done out-of-band and is not considered here).

The VC is then generated and to be verified it must contain details of the holder, the the issuer, information about the credential being confirmed and details of how the issuer signed the information in the VC and which key was used (see Figure 5.2).

Typically the VC would be stored in a digital wallet. To confirm that they have a driving licence to a challenger the holder will use a verifiable presentation (VP). The challenger will provide a nonce (to ensure freshness) and the VC holder will use their digital wallet to generate the VP including the credential and nonce and signed with the key associated with their DID.

In general a VP can contain a set of different VCs, bound together and signed by their owner. Which VCs are included in a particular VP is under the control of the owner and so, unlike showing someone my actual driving licence and giving them other information about me (for example my address or date of birth) I can use a VP to confirm that I have a driving licence and nothing else (see Figure 5.3).

In this simple example the VC and VP contain limited information, just enough to confirm that the holder has a driving licence. However, the holder's name and DID are also included and these will be received by the verifier. To provide more flexibilty and allow the holder to remain anonymous (if they want to) the W3C specification also allows for the use of zero knowledge

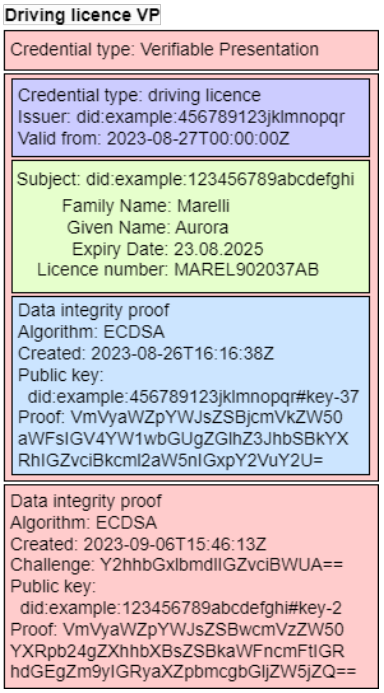

Figure 5.3: A VP used to confirm that the holder has a driving licence

a_1	Subject: did:example:123456789abcdefghi
a_2	Family Name: Marelli Given Name: Aurora
a_3	Date of birth: 03.02.1996
a_4	Address: 23, Railway Cuttings, East Cheam, EC3 8PH
a_5	Expiry Date: 23.08.2025 Issue Date: 03.08.2015 Licence number: MAREL902037AB

Figure 5.4: Data for a more flexible driving licence VC

proofs of knowledge (ZKPoK), for example using BBS+ or CL signatures [5, 7] (for VCs using BBS+ there is a separate W3C document being developed [44]).

Each data item in the VC is associated with a key (an attribute key) and this enables the holder to use a ZKPoK to prove that they have the information and to just reveal which items that they want to show. This is much more flexible, but it does require attribute keys to be defined, possibly by the VC Issuer who then securely provides them to the holder for storage in their digital wallet.

5.3 CONNECT Verifiable Credentials

Figure 5.5 shows the entities involved in the CONNECT system that may be described by the issued VCs. A more detailed architecture description is given in D2.1, this diagram just highlights that there are many entities involved. All of these entities are systems in their own right and need their own security and trustworthiness controls. They are also inter-connected and need to manage what access on what resource is provided to whom and when. Verifiable credentials (VC) and verifiable presentations (VP) allow these varying requirements to be met in a consistent

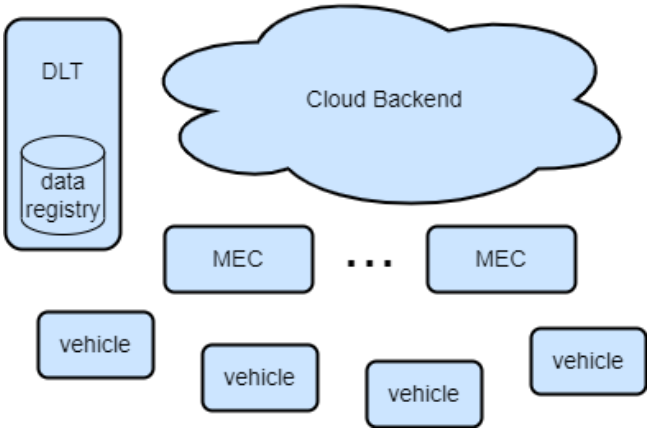

Figure 5.5: CONNECT System Entities

manner. Their use of a distributed data registry for their construction and verification is particularly important for CONNECT where vehicles will be moving around connecting to different MECs and each other.

In CONNECT we will use VCs:

1. to provide reliable evidence about an entities' hardware and software configuration. For example, VCs for the vehicle computer provided by:
   - the processor manufacturer:
     - Identity
     - Hardware security - TEE-guard HSM, ...
   - the OEM regarding
     - Trusted computing base
       - Software versions
       - Reference values
     - Secure containers and the applications running in them

   Note:

   (a) Other devices in the vehicle will have their own set of VCs.

   (b) The Identity and Authentication Management (IAM) component of the vehicle computer will store (and update as necessary) the VCs for the vehicle computer and for other devices in the vehicle.

   (c) These configuration VCs will be used to provide information about the vehicle's different components to the TAF where, for example, different software versions may have different trust profiles, and also to the AIV for generating the harmonised attributes.

2. when ECUs provide their attestation reports to the AIV. These reports will be then used by the AIV to generate VPs for the TAF and for the TCH.

3. for continuous trust-aware authentication and authorisation for access to the DLT. In this case a vehicle can be provided with a set of VCs relating to the properties of the vehicle

and these can be selectively used as required to generate a VP that can be used for authentication and authorisation. Rather than using the VP directly for continuous access it may be used to obtain an access token similar to an OAuth token [27] or, JSON web token [56]. There is continuing interest in this area [1, 66, 37]. These VPs will again need to be signed anonymously, but to avoid the overheads often associated with this type of signature (from the use of cryptographic pairings) a VP could be used to set up a session key (pseudonym). Use of this session key could be constrained by the TEE-guard only allowing it to be used if the vehicle is in a trustworthy state. *Note:*

(a) Access to the MEC and digital twin will use the 'normal' vehicle pseudonyms and PKI already defined for use with CAM/CPM messages.

(b) The DLT access mechanism will be separate from the ABE used to protect data stored on the DLT, while making it available for analysis by the OEM, or by the relevant authorities after an issue is identified.

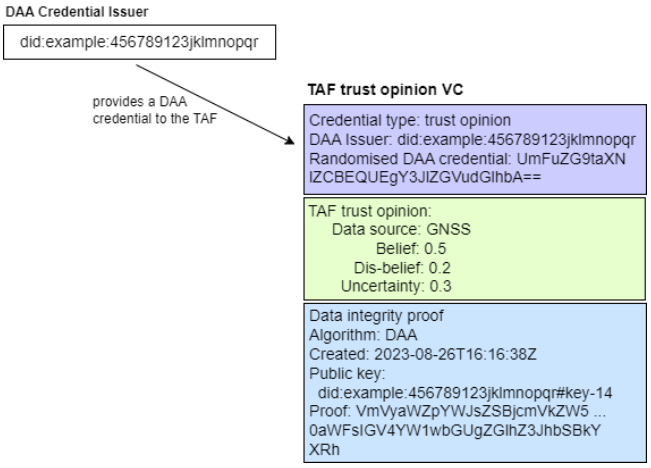

Figure 5.6: A CONNECT VC for a TAF's trust opinion on data from the GNSS.

4. to enable a vehicle to provide information about their trustworthiness. In this case the components in the vehicle providing trustworthiness claims will be configured with a key that they can use to sign anonymously. To do this an issuer will provide them with a randomisable credential for this key and it is this credential together with information about the public key of the issuer that will be included in the VC when it is signed and sent for verification (see Figure 5.6 for an example based on the TAF using DAA to sign a trust opinion on data from the GNSS). The VCs for the different trust assessment components in the vehicle (harmonised attributes, trust assessments from the TAF and misbehaviour reports) will be signed separately and combined into a VP when sent outside of the vehicle. The VP will be signed using the 'normal' pseudonyms and PKI mechanism used for CAM/CPM messages.

   Note: The anonymous signatures used to sign the different trust assessments will be extended to include traceability, so that, in the case of failure or compromise, the OEM can discover which vehicle was involved and use this information to recover data for analysis from the DLT.

5.4 Dictionary of Trust and Trust Data Models

Considering all the above, in Table 5.1 we provide a comprehensive glossary of terms regarding the definition of trust and trust data models. These will be used extensively throughout the remainder of this deliverable, which is dedicated to the description of the data model to be used for the expression of trustworthiness evidence in the context of CONNECT.

Term	Description
Trust	Trust is defined as the assured (i.e., characterised by certainty) belief in some aspect or property of an entity that can be depended on, from the perspective of a different entity.
Trust Assessment Framework (TAF)	A software framework which, given a trust model for a specific function running inside a CCAM system, is able to evalaute trust sources for trustworthiness evidence and evaluate propositions within the trust model in order to obtain their Actual Trustworthiness Levels (ATL). Optionally, a Required Trustworthiness Level (RTL) can be evaluated and trust decisions can be taken and communicated to the application. Note that the TAF in the context of CONNECT has been described in D3.1 [11], where all related vocabulary has also been defined (Chapter 2).
Characteristics/Properties supporting Trust	Aspects of a device or entity, based on which trust in the entity can be evaluated. Such properties may include *integrity*, *security*, *availability*, *robustness*, etc.
Trustworthiness Evidence	Data used in order to assess the trustworthiness of an entity or data item, provided by the entity or devices managing and communicating the data. Such evidence is used in order to assess properties or attributes of the entity. In the context of CONNECT, such data originates from the in-vehicle components (ECUs) and are sent to the Attestation Integrity Verification (AIV) component, based on which an attestation report is created. This is afterwards sent to the TAF so that it can make a trustworthiness assessment of the data item.
Trustworthiness Claim (TC)	A Trustworthiness Claim (TC) is a data structure created by the TC Handler, for holding the harmonised version of the trustworthiness evidence (originating from an entity of CONNECT and produced by a security control). The TC is part of the presentation that is transmitted from the vehicle to external parties. Note that harmonisation of the evidence entails the obfuscation or removal of any personally identifiable information, and is meant to ensure that a TC vector reported outside of the vehicle will not lead to vehicle fingerprinting or implementation disclosure attacks.
Security Controls	Mechanisms used for the generation and extraction of Trustworthiness Evidence, to be used for the assessment of an entity or data item. In the context of CONNECT, these may be **remote attestation**, **misbehaviour detection**, or **intrusion detection**.
Trust Relationship	A relationship between assets or entities that is governed by criteria for secure interaction, behaviour, and outcomes relative to the establishment of trust between entities. Trust relationships can be categorised into **isolation relationships**, **interaction relationships**, and **representation relationships**.
Trustworthiness Evidence Vector	A data structure containing Trustworthiness Evidence originating from all possible data sources (entities and security controls), which will be used in order to make a trust decision.
Data Model	A data structure that is able to represent all types of trustworthiness evidence, regardless of the security control used for its extraction or the entity it originates from. The data model should also capture the mapping of security controls with the type of trust evidence that they provide as part of the overall part assessment, as an inherent part of the data structure.

Capabilities of Assets	The operations of an asset that enable the execution of security controls, in order to enable the collection of runtime trust evidence.
Zero-trust	Based on the zero-trust principle, no device or entity is assumed trustworthy by default. Trust between two or more entities needs to be established based on analysis of trustworthiness evidence, in a manner that provides the required trustworthiness guarantees. In the context of CONNECT, the zero-trust principle is not only applicable to the vehicle components, but also to the trust assessment components (TAF, AIV and TC Handler). Inside the vehicle, trustworthiness evidence takes a number of different forms: attestation evidence from components in the vehicle and from the secure containers running the trust assessment components; together with evidence from mis-behaviour detection monitoring data being generated and intrusion detection running on the different components. Outside of the vehicle, the trustworthiness evidence for data being provided by the vhicle will consist of harmonised attributes, the TAF's trust opinion and a mis-behaviour report. For the MEC, the trustworthiness evidence will consist of attestation evidence from the different secure containers running there.
Node-based trust	Node-based trust refers to the trust between devices based which are responsible for producing, processing, or relaying data to be exchanged between nodes or entities, as part of an application in a service graph chain.
Data-based trust	Trust assessment of a data flow towards achieving the establishment of a trust relationship in the context of a service. Recall that in the context of CONNECT, the various ECUs (sensors, cameras) inside the vehicle are connected to Zonal Controllers, who aggregate the received data and forward it to the in-Vehicle Computer.
Vehicle fingerprinting	The identification of aspects of a vehicle that may compromise the underlying privacy requirements by revealing information such as the brand of the vehicle and its internal architecture, thus enabling a malicious party to perform implementation disclosure attacks.
Trust Objectives	Trust Objectives are statements that define the actions that need to be taken towards the establishment of trust, and are directly dependent on the type of data that is created and exchanged to support a target application.
Direct Objectives	Objectives which have been initially defined by the owner or the manufacturer of a system, and pertain to the main goals of the system as a standalone component.
Indirect Objectives	Objectives which are indirectly derived from the security relationships and interactions that need to be established between system components, and lead to the creation of trust chains to produce a holistic trustworthiness solution for the entire system.
Trust Source	A device, entity, or data object that is able to provide trustworthiness evidence to be used as part of a Trustworthiness Claim.
Trustworthiness Appraisal	An asserted value designed to enable a common understanding of a trustworthiness appraisal by an entity.
Verifiable Credential	As described above, verifiable credentials allow information to be certified and then provided in a verifiable manner and are used throughout the CONNECT system.
Verifiable Presentation	As described above, these allow information from one, or more, VCs to provided to an entity who can then verify their validity and take action on the results.

Table 5.1: Dictionary of Trust and Trust Data Models

Chapter 6

CONNECT Trustworthiness Claims

For complex CCAM “Systems-of-Systems” (SoS) that are the focus of CONNECT, most entities of this ecosystem depend on the correct provisioning of input data from other entities for the execution of safety-critical services. In this context, as highlighted in the previous chapter and further justified in the context of D3.1 [11], the establishment of trustworthiness and trust relationships between participating entities is of paramount importance for the correct execution of safety-critical services - considering various aspects of trust-related dimensions such as integrity, resilience, availability, etc.. In CONNECT, we follow the *zero-trust paradigm* for establishing trust, based on which there are no inherent assumptions on the baseline trust of any actor which must be bootstrapped through the secure communication and verification of appropriate trustworthiness evidence that depict the state of the actors throughout the service lifecycle.

In what follows, we provide details on the formalisation of the data models, adopted in CONNECT, for capturing the characteristics of such trustworthiness evidence to be continuously monitored from all CCAM actors. This accounts for all layers of the CCAM ecosystem, defining models for mounting both in-vehicle system topologies but also the relationships between the vehicles themselves as well as the requirements of the dynamically changing landscape of communicating vehicles with the backend MEC and cloud-based services. These structures unlock the operation of the Trust Assessment framework [11] as they provide the necessary trust sources based on which the evaluation of the Actual Trust Level of each node is calculated and reasoned against the already defined Required Trust Level per service. To this end, the data structure that we will utilise is based on the Trustworthiness Claims (TCs) data structure, which has been defined as part of the Yet Another Next Generation (YANG) data model [31] and is able to capture trust relationships as part of the construction of secure path configurations in next-generation smart-connectivity systems, thus, exhibiting overlapping requirements with the CONNECT-target environments.

6.1 Towards Defining Trustworthiness Claims & Policies

Trust is a relationship between two entities, the trustee and the trustor, and reflects the belief by the trustee that the trustor is behaving, or will behave, correctly. One example in the context of CONNECT may be that a vehicle that receives position data in a CAM message wishes to know whether it can be relied upon. In CONNECT, in addition to the position data, we also provide evidence on the trustworthiness of the data being provided. Note that the notion of trust is not absolute it will also depend upon the subject of the trust relationship. So, for our previous example, while the position data may be trusted, other data from the same vehicle may be much

less reliable. In the context of CONNECT, we do not only refer to trust between devices or hardware-based entities, but we also consider node-based trust and data-based trust. In other words, trust between devices is always related to the type of data exchanged or shared between nodes or entities as part of an application in a service graph chain, and we always include the trust assessment of a data flow as part of establishing a trust relationship. CONNECT D3.1 [11] provides a more detailed discussion of trust and describes, in outline, one of the methods of trust assessment that will be used.

Considering all the above, CONNECT aims to capture the establishment of all types of trust relationships and the execution of trust assessment between entities in complex CCAM ecosystems, specifically (i) in-vehicle trust assessment between systems in the vehicle topology which extract and process the data, (ii) vehicle-to-vehicle relationships, and (iii) relationships between the vehicle and the backend (MEC or cloud). In the latter case, the MEC assesses the level of trust in the vehicle providing the data, and conversely the vehicle assesses the level of trust of the MEC infrastructure where a service is running.

Trustworthiness of an entity, or data item, is assessed by collecting trustworthiness evidence, provided the entity, or devices providing and communicating the data. Such evidence may include *integrity*, *security*, *availability*, *robustness*, of devices and their communication channels. In the context of CCAM, one such example may be the provision of evidence that a vehicle (for instance in a collision avoidance scenario) is equipped with ECUs with adequate level of integrity, capable of providing location data extracted and monitored through a non-compromised software stack. In order to do this we need a data structure that is able to depict such trustworthiness evidence in an interpretable and verifiable manner. The data structure that we will use is the Trustworthiness Claim (TC), which has been defined as part of the Yet Another Next Generation (YANG) data model [31] and is able to capture trust-related information needed in order to perform trust path routing. Therefore, we will adopt the notion of TCs for the representation of trustworthiness evidence and we will expand upon them as part of the use of trust in the context of CONNECT.

For example, consider the case of Cooperative Adaptive Cruise Control (C-ACC), which will be used as a reference point throughout this chapter. C-ACC is essentially a driving assistance system that allows the vehicle to automatically keep a safe distance from other vehicles based on both its own sensor data, as well as sensor data originating from other vehicles on the road and information from the road-side infrastructure. In this case, we need to be able to verify that data collected from low-level sensors such as the camera or LIDAR has been provided by ECUs which satisfy the trustworthiness requirements for their data. This assurance is provided by the ECU's Trustworthiness Claims. The in-vehicle system provides input to the main C-ACC application, which is executed on the Vehicle Computer. Therefore, a Prover device, which may be an ECU belonging to the overall service graph chain, needs to be able to provide trustworthiness evidence on specific sets of properties, in order to enable the assessment of the trust relationship between all devices and systems where this data flow is processed.

In order to collect such trustworthiness evidence, runtime execution and instantiation of runtime security controls. As outlined in D2.1 [13], these may include remote attestation capabilities, misbehaviour detection, and intrusion detection, which provide evidence that can be used as trust sources to the Trust Assessment Framework (TAF). In remote attestation, a Prover device aims to assert its trustworthiness to a Verifier device by providing an appropriate set of trustworthiness evidence on a specific aspect or attribute to be attested to. Other types of security controls include misbehaviour detection, which is able to identify semantic inconsistencies which may be symptoms of data manipulation attacks, and intrusion detection, which aims to detect malicious activity within the Prover device. As it was outlined in D3.1 [11], the CONNECT environment re-

quires the establishment of trust relationships between various entities, both on the vehicle (e.g., on-vehicle ECUs) and cloud entities (e.g., MEC servers). Therefore, it is essential to provide a methodology for the establishment of such Trustworthiness Claims in a manner that enables a common understanding between a broad array of devices, while also ensuring the verifiability of the collected evidence. As aforementioned, trust assessment refers both to trust relationships between nodes, but also to data exchanged in the context of a service or application. Furthermore, trust can be focused on different characteristics for each relationship, which can be assessed from different type of evidence collected from a multitude of security controls. Thus, the data model to be adopted needs to be able to cope with all these intricacies and dependencies, and capture the different types of evidence involved.

The main goal of CONNECT is to enable the dynamic trust assessment in complex CCAM ecosystems, which in turn allows the convergence of security and safety. Specifically, enabling trust assessment in CCAM in the context of the operation and decision-making process of safety-critical applications (for example, to determine whether a vehicle is permitted to change driving lane or not) is based on the use of trustworthy data, originating from devices whose level of trust has been assessed. Specifically, the convergence of security and safety in CCAM is performed through both (i) the establishment of cyber-secure data sharing between data sources in the CCAM ecosystem with no or insufficient pre-existing trust relationship, and (ii) outsourcing tasks from the vehicles to the MEC and the cloud in a trustworthy manner. Therefore, in order to assess these types of dynamic trust relationships, a trust model and trust reasoning framework based on which all involved entities can establish trust for cooperatively executing safety-critical functions has been put forth in D3.1 [11] along with the Trust Assessment Framework (TAF) variants (standalone, federated, and DT-based), which will be documented in detail in D3.2 and D3.3. Therefore, in this deliverable, having defined the high-level reference architecture of the TAF in D2.1 [13] and D3.1 [11], we design the necessary data model abstractions that are capable to handle all the different type of evidence. By instantiating the CONNECT TAF as part of the conceptual model of trustworthiness, considered in the standards, we identify the set of questions that the TAF needs to answer for making a trust decision based on the collected data. This, in combination with the privacy needs (to be elaborated later on), sets the scene of the data structures that need to be defined. Note that, in this deliverable, we provide a high-level description of the CONNECT Data Model that will be further elaborated and instantiated in the context of the use cases in D4.2 and D5.2.

In this context, we need to capture the types of trust policies used within CONNECT, calculated together with the trust models (during the design-time phase, but can also be dynamically configured during runtime). These policies dictate the mode of operation of the TAF and need to capture the required information flow, while also specifying what is the necessary information that should be included within a policy and how it can be conveyed. For example, consider the aforementioned C-ACC use case, where we need to ensure various properties (security, integrity, robustness) of the LIDAR or camera data exchanged between vehicles, as well as within components of the in-vehicle architecture. Therefore, a trust policy should contain the properties of trust that must be attested for the evaluation of one trust relationship, and which are the types of evidence that the TAF needs to have access to in order to perform trust assessment, considering the heterogeneity of trust evidence, which may originate from attestation, misbehaviour detection, or other security controls. In this regard, towards creating privacy- and trust-aware service graph chains through the enforcement of strong security controls, the most prominent method to achieve this is the Common Information Model (CIM) [57], which is the main DMTF (Distributed Management Task Force) standard that provides a common definition of trustworthiness data management functions, independent of the type of security controls used for enforcing

the collection of trustworthiness evidence. The CIM model will allow us to define a data model that can capture the mapping of security controls with the type of trust evidence that they provide as part of the overall part assessment, as an inherent part of the data structure. Within the CIM, the concepts of authorisation, authentication, and filtering, obligation and delegation policies are defined. However, CIM models are not suitable on their own as a policy model, due to the large number of classes they contain. Therefore, in CONNECT, we need to be able to consider the required level of abstraction when formulating trust policies, in order to be able to capture the heterogeneous nature of the trust relationships to be assessed, whether at a node- or data-level, associated with the different types of evidence to be captured.

Considering all the above, as a first step in identifying the abstractions to be modelled, we need to identify the aspects tat impact the level of trust, for which we need to extract the required trustworthiness evidence. Therefore, we need to build upon the trust definition of D3.1 [11] in order to identify the trust objectives that need to be answered through the collected evidence/data captured by the Trustworthiness Claims. Thus, we start by defining what are the trust objectives, as well as the trust relationships and interactions within the various entities and components by answering the following questions:

What needs to be done about trust? This means that we need to identify the trust properties that need to be evaluated through the collected trustworthiness evidence, such as confidentiality, integrity, availability, non-repudiation, authentication, authorisation, and auditability. This also entails the identification of the security controls required in order to collect and extract the trustworthiness evidence based on which the TAF will evaluate the aforementioned properties, such as the aforementioned attestation and misbehaviour detection mechanisms. In addition, the verifiability, integrity, and correctness of the trustworthiness evidence collected by these mechanisms should be ensured. One way to achieve this is through the use of the appropriate crypto primitives (Section 6.1.5).

Where is the security control required? We need to identify the entities where a trust decision needs to be made, as well to express as the type of security controls to be deployed in these entities and their positioning in the overall ecosystem. Recall that, in the context of CONNECT, such entities may be located in the in-vehicle architecture or the backend (MEC or cloud), and the types of security relationships that need to be defined may be within the vehicle itself, from vehicle to vehicle, or between a vehicle and the backend. For instance, components of the in-vehicle architecture may contain the various ECU classes that may be present on a vehicle, such as the Vehicle Computer, the Zonal Controllers, smart sensors/actuators, or any other type of ECU that may be used. At the backend, such entities include the MEC, which may need to provide trustworthiness claims for the supplied services, through evidence about the trustworthiness of V2X messages it has received. For the aforementioned C-ACC service, the appropriate security controls need to be applied in the respective entities both in-vehicle and in the cloud, considering also their positioning in the overall ecosystem, in order to provide trustworthiness guarantees for maintaining the required distance from other vehicles.

Between which nodes in the chain does the security control need to be executed? As aforementioned, security controls refer to any mechanism that is able to output trust evidence, such as attestation, misbehaviour detection, intrusion detection, etc. As an extension of the previous point, in the context of CONNECT, the endmost goal is to establish and manage *trust between entities*, starting from bilateral interactions between two single components and continuing as such systems get connected to even larger entities in order to reach the required level of trustworthiness for the entire service graph chain. Recall that, the context of CONNECT, three types of trust relationships are defined: in-vehicle, vehicle-to-vehicle, and vehicle-to-cloud. For each type

of trust relationship, we may need not only a different security control, but we may also require different types of trustworthiness evidence from a specific security control. Therefore, as part of the trust relationship, when we have different trust relationships of the same nature, we may have different evidence that we want to monitor, supported by the security controls. For instance, in the case of in-vehicle relationships, a Prover device may need to be attested, and a Verifier that is responsible for verifying the correctness of the attestation evidence provided by the Prover. As outlined in D2.1 [13], the Prover device belongs to the hierarchy in the vehicle (i.e., ECUs which control the sensors and actuators connected to a set of zonal controllers, which are connected to the Vehicle Computer) and provides attestation data to the Attestation and Integrity Verification (AIV) component of the Vehicle Computer. In turn, the AIV acts as a Verifier for the attestation evidence. In general, in order to accommodate the pressing need for establishing federated trust between services and devices, we need to identify which devices need to fulfil these roles in the context of a trust-aware service graph chain. Conversely, in the vehicle-to-cloud case, as outlined in D2.1 [13], an *Enclave-CC* environment is used in order to deploy a safety-critical application and all its dependencies in a dockerized container. In this case, the MEC is able to assess the integrity of not only the software running on the vehicle, but also on the entire container.

When is the security control required? This refers to the temporal relation of the service with the occurrence of the corresponding security event, i.e. before, during, or after. In CONNECT, a security event may occur according to the threat landscape defined in D3.1 [11] and pertains to the entire life cycle of the device. This question is equivalent to the identification of how often trust evidence needs to be captured. Specifically, the TAF must be notified of any change in a trust property or an entity, meaning that any event that may impact the integrity of a device should be able to be captured by a security control and sent back to the TAF. Therefore, we must be capable of capturing any event that has an impact on trust. In this context, we need to support the continuous execution and operation of the security control, which should be able to support multiple modes of operation: (i) push operation, where a trust-compromising event is monitored and pushed to the TAF, and (ii) pull operation, to be able to respond to requests from the TAF to pull trust evidence from security controls. Also, the security controls should be capable of running in asynchronous mode, meaning that they should be notified in case a change is detected.

What additional elements to complement the specification of a security objective are needed? Equivalently, beyond the trust evidence, what additional information should be contained within the data model to enable trust assessment? This refers to any additional information that may be required in order to properly execute a security control (e.g., attestation or misbehaviour detection task) for achieving a security objective. In this context, there are two layers to consider: (i) Evidence based on which a trust decision is made, and (ii) referral trust decisions considering the fusion of trust decisions that have already been calculated from other nodes. Specifically, as part of in-vehicle trust assessment, the only information needed is the evidence extracted from the security controls. However, in the case of trust assessment between vehicles or between a vehicle and the cloud, additional information needs to be shared related to the subjective output of the local trust assessment, as well as the additional security controls (misbehaviour reports). In addition, towards providing privacy-preserving properties to the Trustworthiness Claims, the data model should be able to obfuscate the contained trustworthiness evidence.

Based on all the above, we need to define a model that can be used to exchange and share the information required in order to assess the level of trust, originated from the available trust sources (such as attestation and misbehaviour detection). Therefore, answering the aforementioned questions will enable us to identify the elements to be included in the data models required for serving the entire lifecycle of the trust assessment process: (i) data model for capturing

the trust policy, (ii) data model for capturing the triggering of the trust assessment through the Trust Assessment Request, and (iii) data models for the secure and privacy-preserving exchange of trustworthiness evidence, depending on nature of the information exchanged so as to avoid breaching the privacy of the users (e.g., vehicle fingerprinting).

6.1.1 Alignment with International Standards

In order to ensure the compliance of the Trustworthiness Claims modelling approach followed in CONNECT with the relevant international standards, we aim to position this problem within the framing of an architectural approach in accordance with the ISO/IEC 30141:
2018 standard [58], whose scope includes the definition of a universal IoT Reference Architecture. Through this approach, we aim to provide a framework which can convey the abstractions to be captured by the employed trust modelling approach, including the connections and interactions between the information and properties per asset, as well as how these interactions translate to the level of trustworthiness that we want to achieve through the execution of the available security controls.

In this context, a graphical representation of the conceptual model of trustworthiness is provided in Figure 6.1, aligned with the ISO/IEC 30141:2018 standard. Based on this model, user Trust on the Behaviour of a system is a measure of Confidence in the overall system. In other words, the degree to which a system fulfils a set of required characteristics (e.g., Safety, Security, Privacy, Resilience, and Reliability) can be considered as a measure of confidence of the users in the entire system. Conversely, these Characteristics are the aspects of a system that enable the expression of these Trust-enabling Behaviours.

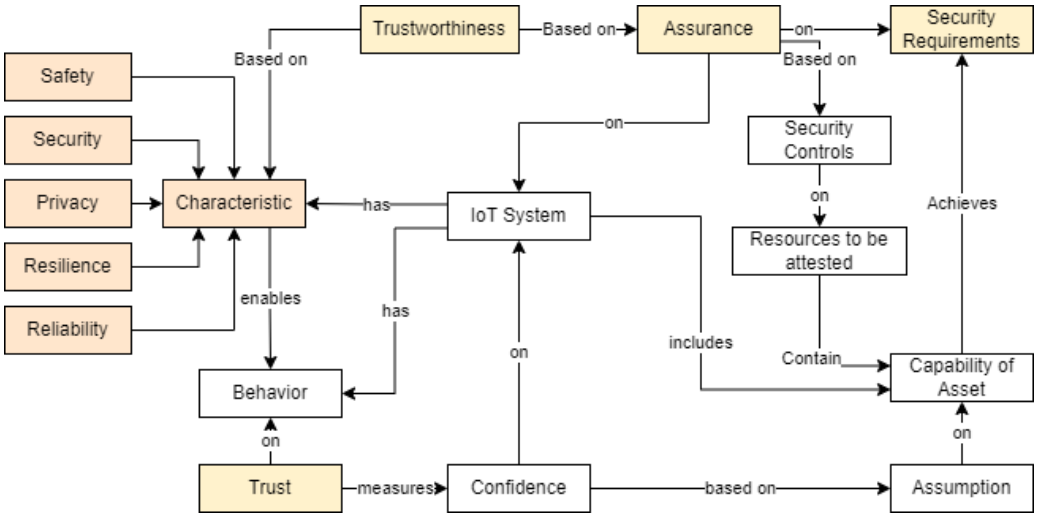

Figure 6.1: Conceptual Model of Trustworthiness within CONNECT

Trustworthiness is based on the concept of Assurance on the overall system, which includes its asset cartography and the interdependencies between the assets. Assurance is determined based on the available Security Controls (such as the aforementioned attestation and misbehaviour detection mechanisms), which are performed on the Resources to be attested, as defined in the conceptual CONNECT architecture in D2.1 [13]. In order to apply and execute these Security Controls, we need to consider our Assumptions on the Capabilities of the Assets. These capabilities essentially refer to the operations of the asset that enable the execution of security controls (such as runtime attestation, misbehaviour detection, and intrusion detection) in order to

collect runtime trust evidence. In the context of CONNECT, these may include the secure bood module, the Trusted Computing Base, (TCB), and the Runtime Tracer. Recall that trustworthiness evidence can be split into design-time and runtime evidence, based on which trust assessment can be calculated. The Trustworthiness Claims modelling methodology will be presented throughout the remainder of this Chapter, where the construction of TCs is performed considering the asset Capabilities based on which a trust assessment can be made.

In order to further ensure compliance with the relevant international standards, we aim to provide a Trustworthiness Claim modelling methodology aligned with the conceptual model of cybersecurity defined in ISO15408-1:2009. Based on this standard, the security of a system is established through the definition of security objectives and security relationships for the system components, i.e., the assets of the system.

Trust Objectives: These define the actions that should be taken towards the establishment of trust, and are directly dependent on the type of data that is created and exchanged for the support of the envisioned application. The trust objectives of a system are directly dependent on the type of data created, managed, and processed within the system as part of the application. However, in the context of CONNECT, the concept of security goes beyond the notion of securing an individual asset, and extends to the security of a hierarchical composition of assets, representing the structure of all types of components present within a vehicle or automotive system. Recall that, in CONNECT, the hierarchical composition of assets is essentially the set of devices (e.g., ECUs) from which a data workload will have to go through in order to reach the CCAM application. Whether we refer to the in-vehicle topology comprising multiple ECUs and components, or multiple vehicle topologies that need to be assessed as part of an application, trust objectives need to capture information for the entire topology. Therefore, in order to achieve the desired level of trustworthiness, the interpretation of the above questions yields two types of trust objectives:

- *Direct Objectives*: These are initially defined by the owner or manufacturer of a system, as the main goals of the system as a standalone component. For example, a target ECU may need to be attested based on properties defined by a manufacturer (such as integrity, security or robustness). An ECU may contain a Hardware Security Module (HSM), which is able to manage securely storing keys and the result of the secure boot process, which enables the fulfillment of such objectives.

- *Indirect Objectives*: These are indirectly derived from the trust relationships and interactions that need to be established between system components, thus creating trustworthiness chains to produce a holistic trust solution of the entire ecosystem. These refer to the aforementioned types of trust relations, i.e., between in-vehicle components, between multiple vehicles, or between the vehicle and the cloud.

Trust Relationships: Trust relationships between the system components complement trust in a comprehensive and systemic manner, by uncovering additional trust objectives between related components to produce an internal trust solution. In general, trust relationships between system components stem from the nature of trust, which is defined by three concepts: isolation, interaction, and representation. Based on this, the three types of trust relationships are outlined in Table 6.1, including the application context for each type of relationship in CONNECT.

Trust Relationship	Description	Application Context

Isolation	Exists when the component or asset is totally separated from other components located inside or outside a system. This separation can be performed by another component, which acts as a partial or total isolator of the first component with respect to the other components. An isolation relationship requires one or both components to be physical, while one (but not both) of the components can be digital. In the context of CONNECT, direct ECU-to-ECU communication is not possible, and ECUs can exchange information only through a a Zonal Controller. Therefore, this creates an isolation relationship with regard to the ECUs.	The ECUs belonging to an in-vehicle architecture are directly related to the extraction and management of kinematic data based on which all CCAM applications operate. Therefore, we should be able to extract evidence on the correctness of the configuration and operational state of the ECUs, both in the case of resource-capable ECUs with capabilities for asymmetric cryptographic capabilities, or resource-limited ECUs with capabilities for only symmetric crypto. In the former case, as analysed in D2.1 [13], in order to store data and keys or execute binaries in a hardware-protected environment, computationally capable devices may leverage **Trusted Execution Environments (TEEs)**. These essentially creates an isolation relationship for the protected assets, since software-only attacks outside of this protected environment cannot affect the protected keys, data, and software. In the latter case, resource-limited devices that do not support the installation of TEEs may not have capabilities to create such isolation relationships.
		Therefore, since the level of isolation provides different weight on each type of trust evidence, we should be able to extract such information regardless of the achievable level of isolation.
Interaction	Exists when two components interact or communicate in any way. In the context of trust, the communication interfaces are not of relevance, but only the location of the interactions for the purpose of identifying trustworthiness requirements of the interacting components. In the context of CONNECT, interactions may take place within the vehicle, between vehicles, or between the vehicle and the MEC or cloud. This requires different types of trust evidence to be captured for assessing these types of relationships.	In the case of in-vehicle relationships, this type of trust relationship is captured through the envisioned interconnectivity structure between the assets of the vehicle. Specifically, as defined in D2.1 [13], various types of ECUs within the vehicle are connected to the Zonal Controller, which in turn is controlled by the in-vehicle manager. In the context of vehicle-to-vehicle or vehicle-to cloud commmunication, the in-Vehicle Computer is responsible for the construction of CAM/CPM messages. Such messages can be either transmitted to other cars (signed with pseudonymous certificates), or may contain misbehavior reports and Trustworthiness Claims to e transmitted to the Cloud or MEC. Therefore, we need to be able to capture all types of trust evidence required in the context of the aforementioned types of interactions.

Representation	Exists between a system component acting on behalf of another component. Components that participate in this type of relationship can be of any type, physical or digital, and enable the gateway between different types of components. Specifically, when performing a trust assessment within a vehicle, the central TAF or application running on the Vehicle Computer to represent the trust level of all internal ECUs comprising this vehicle. This also enables trust assessment at a higher layer (vehicle-to-vehicle and vehicle-to-MEC), but also to capture the required privacy-preserving properties.	In the context of CONNECT, this type of relationship is captured through the use of **Verifiable Credentials (VCs)**, which are constructed in order to represent the attributes of a device in a trustworthy manner, so that they can be used to report on the trustworthiness of the vehicle. The types of defined VCs include *Harmonised attributes*, *trust opinions of the TAF*, and *misbehaviour reporting*, and will be analysed in detail in Section 5.1. Specifically, in the case of Vehicle-to-Vehicle and Vehicle-to-MEC communication, the notion of harmonised attributes is leveraged in order to capture the underlying privacy requirements with regard to data sharing, so as to not compromise the privacy of the vehicle or the user. In addition, CONNECT employs the notion of **Digital Twins (DTs)**, where if an ECU needs to perform a specific operation but lacks the necessary resources or capabilities, it can delegate the task to a trusted DT that possesses the required expertise or resources.

Table 6.1: Types of Security Relationships and Application Context in CONNECT

6.1.2 Design Principles & Requirements

Based on all the above, as well as the reference architecture on conceptual trustworthiness presented in D3.1 [11], in Table 6.2 we provide the set of requirements that needs to be fulfilled by the approach to be followed for the modelling of the Trustworthiness Claims, in a manner that considers both the heterogeneity of trustworthiness evidence, and the types of employed security controls.

Requirement	Description & Need
Simplicity	The data model must enable the expression of the trustworthiness evidence in a manner that is simple and understandable, and can be interpreted by the TAF, which is responsible for evaluating the evidence. It should also be able to provide the level of granularity required in order to express the available sources of trustworthiness, regardless of the trust source they originate from (attestation, misbehaviour detection, intrusion detection). In addition, the trustworthiness claims must be simple enough to enable the fast verification of the sources of trustworthiness by the TAF, in order to enable the fast and efficient trustworthiness evaluation of a system.

Expressiveness	The data model should be expressive enough to capture different types of trustworthiness evidence, which is managed by different types of security controls and for different entities. For instance, we should be able to express integrity in the context of the in-vehicle components, but also for the infrastructure where the MEC is running. Therefore, we should be able to express evidence that is able to provide correctness proof on both the vehicle and the infrastructure, and the data model of CONNECT should be able to capture the heterogeneous types of entities present both at the cloud (e.g., MEC) and at the edge (e.g., the various types of ECUs and sensors of the hierarchical composition of devices in a vehicle). It should also be able to capture all the types of properties to be attested (e.g., design time integrity, boot-up integrity, runtime integrity, and communication integrity). Thus, the data model should be able to express the security controls to be performed, considering the types of devices and properties, in order to extract a trustworthiness decision.
Extensibility	The data model needs to capture all the types of trustworthiness evidence that can be included as part of the data model which were defined in the threat landscape defined in D3.1 [11], the types of threats included in the threat model to be defined in D3.2, as well as the types of heterogeneous devices comprising a vehicular environment. However, it should also provide the capability to consider new types of threats and new types of devices, as well as any new security controls that may be required in order to mitigate the newly identified threats.
Abstraction & Encoding	Encoding should follow a universal approach, based on which the same type of evidence can be extracted and used in order to evaluate particular properties of trust and perform trust assessment following the same methodology. In other words, the evidence that expresses the types of attributes to be attested (e.g., security, integrity, correctness) should be interpretable as part of a trust assessment process in a universally adopted manner. The trustworthiness evidence which is needed in order to determine the Actual Trust Level (ATL) based on the outcomes of the aforementioned types of security controls, which needs to be compared to the Required Trust Level (RTL) defined for the target system. Therefore, a common semantic expression is required in order to compare the RTL and ATL.
Diversity and Trust Evidence Definition	The employed data model should be able to support the configuration of all defined security controls (such as attestation enablers and misbehaviour detection systems) in a manner that is able to assess the trust level of all assets of the target system. In addition, it should be able to support all types of trustworthiness evidence to be utilised, as outcomes of the aforementioned security controls.
Continuity	The policies dictating the security controls for the collection of trustworthiness evidence should take into consideration the hierarchical structure of the defined architecture. For instance, in the in-vehicle case, a number of ECUs control the sensors and actuators connected to a set of zonal controllers, which are connected to the in-Vehicle Computer. Thus, the data model should ensure the continuity of the policy chain, meaning the tracking of policy enforcement and the devices a policy is assigned to.
Periodicity & Freshness	There need to be guarantees that a deprecated Trustworthiness Claim cannot be used in order to assess the trustworthiness level of an asset (e.g., in the context of the avoidance of replay attacks). In addition, the confidence of an entity in the correctness of a Trustworthiness Claim may be time-limited, meaning that such a claim may expire after some time has elapsed. In addition, in cases where communication is disrupted (e.g., a system reboot), subsequent trust re-establishment is required. The evidence for this process is provided by the secure boot process, which provides an attribute dedicated to the state of the device at boot-up time.

Verifiability	Since the trustworthiness evidence may originate from various different sources (such as attestation enablers or the misbehaviour detection module), the collection and use of this evidence should be supported through the introduction of the appropriate crypto primitives that ensure the correctness and integrity of the data used as part of the Trustworthiness Claims.

Table 6.2: Design principles and requirements for CONNECT trustworthiness model

As it will be outlined throughout the remainder of this Chapter, in order to fulfil all the requirements presented in Table 6.2, we select Yet Another Next Generation (YANG) data modelling language in order to enable the expression of Trustworthiness Claims in CONNECT. Specifically, the YANG data model has been defined in order to accommodate the sharing of trust-related information by the relevant IETF standards [31] in the context of a problem with overlapping aspects with regard to trust. It has been defined so that it is able to capture the trust evidence that needs to be exchanged as part of the Trustworthiness Claims, in order to enable the establishment of trusted paths in a network topology. Therefore, the YANG data model has been defined based on the same nature of problems as the ones we are trying to solve in CONNECT, which acted as the cornerstone of our choice behind the adoption of the YANG data model.

6.1.3 The Role of Privacy in TC Construction

As it has been outlined in D2.1 [13], the issue of dynamic trust establishment in CCAM ecosystems spans over three dimensions: (i) technical, (ii) policy-making, and (iii) societal. Regarding trust assessment, we have so far focused on the first dimension, through the definition, from a technical perspective, of measures of trustworthiness and the formation of trust policies containing the appropriate security controls for the collection of the necessary trustworthiness evidence. The second dimension refers to the trust that users have in policymakers (private or public) to properly regulate the production and operation of vehicles. For the third dimension, we need to consider the intricate interplay between trust and perceived risk. One of the key aspects of this dimension is the perceived risk to one's privacy (privacy risk), defined as *the perception of misuse and loss of control of personal data that happens when using the vehicle and which will typically arises because of the transmission of CCAM messages*. This is of paramount importance in CCAM systems, since the perception of privacy risks for the user stemming from the incorrect handling of personally identifiable data may lead to a decrease in the level of trust in a vehicle and by extension the OEMs and the vehicle manufacturer. This is important, since a perceived privacy risk is likely to have a negative effect on the intention of consumers to purchase and use a specific vehicle.

Work so far has culminated in the standardisation of PKI-based solutions and the use of crypto primitives such as pseudonyms, in order to be able to safeguard the privacy of vehicles. The introduction of trustworthiness evidence to facilitate trust assessment leads to the introduction of a new dimension to the data being exchanged between vehicles, this and its privacy implications, have not been yet considered in the literature. Specifically, as outlined in Chapter 3 of D2.1 [13], it is possible that the trust-related evidence can contain sensitive information that may lead to vehicle fingerprinting, meaning the inference of various personally identifiable aspects of the vehicle, such as the brand of the vehicle. This may lead to breaches in the unlinkability and untraceability of the vehicle and enable implementation disclosure attacks, since a malicious party may deduce information related to the internal architecture of the vehicle. Therefore, in the context of privacy

preservation, it is imperative to consider the types of additional privacy concerns introduced when combining trust evidence with the exchanged standardised CCAM messages which may lead to the fingerprinting of the vehicle, in order to not compromise the perceived level of trust in the vehicle by the users.

Privacy risk	Description
Breach of location data	Navigation and ”infotainment” systems, that are present in many modern vehicles, have access to location data of the vehicle, as well as to the destinations they have driven to. These systems may also have access to sensor data, that might be used in order to perform **Vehicle Fingerprinting**, i.e., the identification of the location of the vehicle.by an illicit third party. We expand on this notion in Section 6.1.3.1.
Breach of data related to entertainment systems	Entertainment systems that work via a connection to a driver's smartphone may be able to obtain access to mobile device information, such as their contact lists. This may be exploited by a malicious party to compromise the identity privacy of the driver or the vehicle.
Breach of telematics data	This refers to data related to driving habits, which is recorded through various sensors by modern cars and may be used by insurance companies to offer safe driver discounts to their customers. This type of data is intended to be used in investigations in the event of an accident, but may be collected without the knowledge or the consent of the driver.
Misuse of behavioural or travel data	Refers to the possibility that behavioural data or travel data originating from the normal use of a vehicle may be obtained by third parties such as automotive developers without the consent of users, to be used to extract analytics for development purposes or monetary gain.

Table 6.3: Privacy risks in automotive domain

Considering the discussion above and the risks identified in Table 6.3 the following question arises: *Beyond the standardised privacy solutions available, what else needs to be done with regard to the expression of trustworthiness evidence, in order to not breach the privacy profile of the vehicle?*. We need to define what type of processing should be performed in the data model in terms of encoding, in order to avoid privacy breaches. One part of the CONNECT solution (analysed in D4.1 [12]) is to use harmonised attributes for the vehicles, their use prevents the identification of the type of vehicle providing the data. These claims are constructed by the CONNECT Trustworthiness Claims Handler based on the output of the attestation enablers and the device environment providing the data. The definition of harmonised attributes, and their role in the preservation of privacy in the context of the exchange of trustworthiness evidence, will be described in detail in Section 6.1.4.

6.1.3.1 Vehicle Fingerprinting

As outlined in the previous sections, it is crucial that the data collected by the security controls (attestation enablers, misbehaviour detection, intrusion detection) to be used as trustworthiness evidence does not reveal any type of information that may have adverse effects on the privacy of the vehicle and/or user. One of the most predominant threats in this regard is vehicle fingerprinting, which refers to the identification of aspects of the vehicle that may compromise the underlying privacy requirements, and to enable a malicious party to perform implementation disclosure attacks.

For example, consider the C-ACC use case which has been outlined throughout this Chapter. Recall that, in this case, data collected from cameras in a vehicle, or low-level sensors such as

a LIDAR, needs to be sent to the Zonal Controllers of the vehicle, and afterwards to the C-ACC application on the in-Vehicle Computer. However, in addition to the information originating from the vehicle itself, the C-ACC application also requires information originating from the roadside infrastructure, information from the Global Navigation Satellite Service, as well as positioning and kinematic data from other vehicles. In the latter case, a vehicle may need to send information, such as steering data and location data, to other vehicles for the execution of a C-ACC service. When providing trustworthiness information such messages could contain detailed information regarding the correct configuration of an ECU within a vehicle and this may lead to the inadvertent disclosure of data such as the vehicle's brand or internal architecture. Therefore, we need to define a method for sharing such types of messages, without requiring the disclosure of such sensitive information.

It follows that it is not enough for the employed data model to fulfil the design principles and requirements outlined in Table 6.2, but it is also important to consider the issue of privacy. Specifically, the data model should not only have the required level of expressiveness that enables the representation of all trustworthiness evidence in a manner that fulfils the given design principles and requirements, it should also ensure that it is not possible for a malicious party to perform vehicle fingerprinting and, by extension, exploit the extracted information (for example, in the context of an implementation disclosure attack). In this regard, it is important to consider that increasing the level of expressiveness may lead to revealing information regarding the brand or internal architecture of the vehicle while if trust assessment knowledge is obfuscated in order to enhance privacy, this may lead to a reduction of the level of expressiveness in the data model, thus having adverse effects on the precision of the trust assessment process.

There is a trade off: *How can we disclose the information required in order to perform a trust assessment, without significantly compromising either the accuracy of the assessment, or the privacy of the vehicle or user?* In order to address this issue, we introduce the use of harmonised attributes, whose construction and role towards the achievement of privacy preservation will be described in detail in the following section.

6.1.4 Beyond Zero Trust: Harmonising TCs for Privacy-Preserving Trust Management

As previously outlined, a core consideration in CONNECT is the preservation of the privacy of the vehicles participating in the network, by ensuring that they cannot be identified based on the trustworthiness claims that they provide. Extensive research has been performed in the literature regarding the preservation of privacy in CCAM (for location privacy, anonymity, etc.). Recall that, the approach followed in CONNECT, is to leverage trustworthiness evidence originating from various trust sources these are: evidence collected from each device in the system (from the execution of a security control, such as attestation), misbehaviour reports from the MBD and trust opinions from the TAF. In this section we focus on evidence from the devices in the system and how it can be reported outside of the vehicle without allowing the vehicle to be identified in any way. Evidence from the ECUs is aggregated by the corresponding Zonal Controller, and forwarded to the in-Vehicle Computer. Afterwards, the Trustworthiness Claims Handler is responsible for the creation of the Trustworthiness Claims to be sent outside of the vehicle. The main challenge in this process is to ensure that no privacy issues arise, as the trustworthiness claims represent the assets of the attested system.

To this end, we introduce the concept of harmonised attributes. The motivation behind the con-

struction of the harmonised attributes is that all participating vehicles should use the same set of attributes when reporting on the devices in the vehicle, so that a trustworthiness claim vector will not leak any information on the assets of the vehicle providing the claims. Thus, the trustworthiness claims do not report on the assets of the vehicle themselves, but on how these assets contribute to the trust level of the vehicle. For example, while a harmonised attribute may disclose that an ECU possesses an underlying Root-of-Trust (RoT), it will not disclose the type of RoT (e.g., TPM). Similarly, it may be disclosed that an ECU has a secure boot mechanism, but no further information regarding the version or the numbering of the operational system, as this could lead to the fingerprinting of the brand of the vehicle, thus compromising the unlinkability of the vehicle. These harmonised attributes will be anonymously signed, so that a receiving entity knows that it originates from an authentic vehicle, without revealing its identity. This is in accordance with the notions of identity and location privacy, as standardised by ETSI, and used to report on the state of the devices involved in providing data to external entities, either other vehicles or the MEC. These signed harmonised attributes will help the TAF of another vehicle, or the MEC, to form a trust opinion on the vehicle providing the data.

As described in D2.1 [13], in order to be aligned with the latest standards in the vehicular industry, the devices inside a vehicle, will follow a tree-like topology that enables the better provisioning and execution of services (service oriented topology). In this structure (depicted in Figure 6.2), a Vehicle Computer with high computational power manages the vehicle's Zonal Controllers, each one of which manages a subset of the vehicle's ECUs. In the employed tree-like architecture, the lowest level in the hierarchy consists of the ECUs which are categorised into two types depending on their computational power: (i) S-ECUs which represent low-end control units that are dedicated to the execution of safety critical applications, thus S-ECUs only supports symmetric cryptographic functionalities, and (ii) A-ECUs which support both symmetric and asymmetric cryptographic functionalities. Therefore, a communication and processing trustworthiness chain is defined starting from an ECU, going through the respective zonal controller, and leading to the Vehicle Computer.

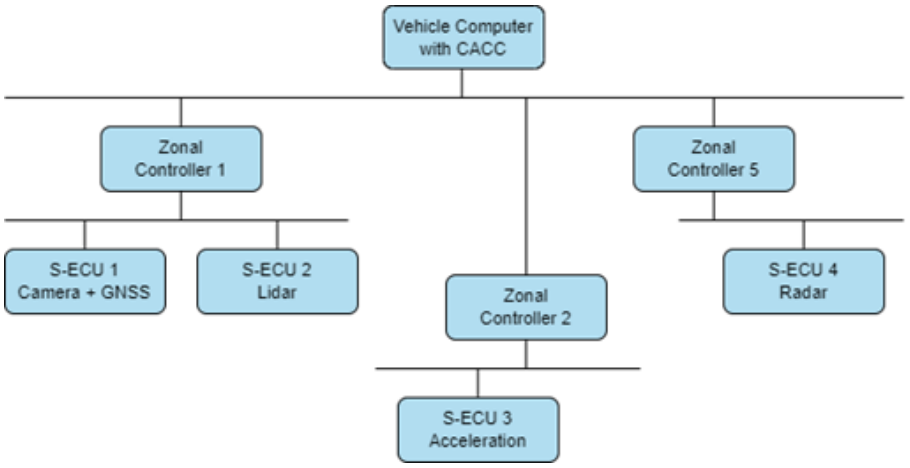

Figure 6.2: Tree-based structure of vehicle in CONNECT

During the execution of a security control mechanism (such as attestation) the ECUs involved produce their Trustworthiness Evidence based on the security control performed on them. the Trustworthiness Evidence produced is sent from the ECUs to the Zonal Controllers, which contribute their own trustworthiness evidence to the Trustworthiness Evidence Vector. Afterwards, the Trustworthiness Evidence vector is sent the AIV, which encodes the vector to an attestation

report in the YANG data model, which is going to be described in Section 6.2. After this process is completed, the report containing the necessary level of harmonisation is sent to the TAF so a trust opinion on the vehicle can be formed and to the TCH to generate the harmonised attributes. Note that, as previously outlined, it is of paramount importance to maintain the freshness of the TC Vector, in order to ensure that the TAF's trust opinion and the harmonised attributes from the TCH are based on on the current state of a vehicle (and not an outdated or deprecated state).

The harmonised attributes depend on the status of all the components that are participating in the processing chain, regarding all the trust properties defined in D3.1 [11]. As described above the harmonised attributes summarise information provided by the processing chain with regard to trustworthiness attributes such as the safety, resilience, robustness and integrity of the system, without disclosing information on the specific assets they possess. Such harmonisation will remove or obfuscate any information that may lead to the identification/fingerprinting of the vehicle. The detailed definition of the abstract data model for all types of trust properties will be described in D4.2, based on the YANG data model, which will be described in Section 6.2. For this purpose, we provide an example on the concept of the integrity trust property, as the integrity attribute is the common denominator to all CCAM ecosystems, where trustworthiness evidence needs to be exchanged for the establishment of trust. In the context of this example, the harmonised attributes can be placed into the following categories:

1. *Design time integrity:* We define an attribute dedicated to the capabilities each entity can support. As defined in the previous section, all devices in a vehicle are capable to perform at some level of cryptographic operation, either symmetric cryptographic operations such as encryption and HMAC, or asymmetric cryptographic functionalities where more elaborate schemes can be exploited. Another important aspect regarding this attribute is the underlying Root-of-Trust (RoT), in which case numerous solutions are avaliable, such as the Hardware Security Modules (HSMs) that offer secure key management and protected memory (e.g., in TPMs), or Trusted Execution Environments (TEEs) such as Intel SGX that acts as an extension of the CPU and isolates the memory pages occupied by a specific application from the rest of the system. Therefore, all capable devices are configured and possibly certified to perform attestation processes. The only component that cannot, to our knowledge, support any cryptographic operations, is the Global Navigation Satellite System (GNSS). *The design time integrity attribute will be represented as a three values: 1 if the device has cryptographic capabilities and a hardware root of trust, 0 if the device has cryptograhic capabilities but doesn't have the support of a hardware root of trust, or -1 if the device does not have cryptographic capabilities.*

2. *Boot-up integrity:* We define an attribute dedicated to the state of the device at boot-up time. Specifically, as it is well-documented in the literature, we want to offer a secure boot type of operation, directly linked to the underlying root of trust, that is going to provide a fingerprint of the read-only memory of the device containing the BIOS and the firmware of the device. Thus, it can be checked whether or not the code loaded in the device is verified before it is allowed to be executed. *The Boot-up integrity attribute will be represented as a binary value, 1 if this devices has some kind of secure boot mechanism, 0 if this device does not have any kind of secure boot mechanism.*

3. *Runtime integrity:* In the types of complex systems considered in CONNECT, we need runtime proofs that each component is in a correct state. To this end for example, we aim to leverage tracing mechanisms, in which case numerous options are available, categorised into software-based solutions (such as eBPFs) and pseudo-hardware-based solu-

tions (such as Intel PT). For this purpose, we need an attribute dedicated to the device's capability to perform introspection on the running processes. Depending on the integrity assurance that is needed, different tracing modes will be employed, such as Configuration Integrity Verification (CIV) and Control Flow Integrity (CFI). In addition, apart from the introspection aspect, run-time integrity is directly related to the protection of the respective cryptographic keys. Specifically, this attribute also refers to the underlying Root-of-Trust and the enforced key restriction usage policies. Assuming the existence of a hardware-based RoT, CONNECT keys are going to be bound with a correct configuration of the respective functionality of the attested unit. *This harmonised attribute will be represented by three values, 0 if introspection and a RoT is not supported at all, 1 if there is runtime integrity through the RoT, and -1 if the integrity check fails.*

4. *Communication integrity:* In the context of CONNECT, proof of correctness is required in the transmission of sensitive data over a specific communication session. Therefore, it is crucial to prevent tampering of the data during transmission. As aforementioned, the components comprising the trust processing chain have established cryptographic keys between them for encryption and authentication purposes (ECUs $\rightarrow$ Zonal Controller $\rightarrow$ Vehicle Computer). The communication integrity attribute will dictate whether or not the cryptographic keys are correctly installed and used. Note that the installation of keys may also be considered as a design-time integrity attribute, while deciding that they are correctly used may be challenging. *This harmonised attribute will be represented as a binary value, 1 if the device has correctly established its respective cryptographic keys, or 0 if the devices does not have cryptographic capabilities for communicating with other devices inside the system.* It should be noted that those keys need to be protected by the underlying RoT, to prevent key leakage and key corruption.

5. *Certified application attribute:* Following our concerns regarding the sensitivity of the data, in the context of safety in CONNECT, we consider that only certified services and functions can be part of safety-critical applications. Specifically, in order to achieve this, the Identity and Authentication Management (IAM) module establishes with all ECUs separate keys dedicated to the processing of their respective data. In this context, in CONNECT we propose the definition of an attribute for data safety, so that the IAM can check whether or not the data protection keys were used during the data processing. It has to be noted, that each data source is associated with a unique key known by the IAM.*This harmonised attribute will be represented as three values, 0 if this attribute is not relevant (this application is not safety critical), 1 if all data processing in a safety critical application is done by certified services and functions only, and -1 if an unauthorised application has processed the data.*

In general, harmonised attributes can be split in two categories, namely static and dynamic:

Static attributes, specifically design-time integrity and boot-up integrity, depict the static state properties of a device, and can be extracted from trust extensions that are associated with the underlying Root-of-Trust (RoT). These attributes could be stored inside a Trusted Component, in order to be protected in an isolated environment during runtime. Note that these attributes should not be changed once the system is up and and running, except when the system is updated. However, even in this case, if the updates are executed correctly, these attributes should not be altered.

Conversely, dynamic attributes, such as runtime attributes and communication attributes might change throughout the life cycle of a device. For example, in case a cryptographic key, involved

in the communication chain from an ECU to the Vehicle Computer gets revoked for detecting misbehaviour, this may lead to the modification of the communication attribute. Another example in the same context could be the modification of the run-time attribute. In this case, either the certificate of the Tracer was deemed invalid (expired/revoked) or the application that was monitored is not longer relevant for the creation of the Trustworthiness Claims.

The role of each attribute depends on the trust properties that need to be assessed. The goal is to obfuscate the trustworthiness claims vector as much as possible, without adversely affecting the accuracy of the trust assessment. In addition, as aforementioned, there is a tradeoff between privacy and how it impacts safety considerations of a specific operational domain. For instance, in the context of the envisioned *Collaborative Cruise Control* use case [13], obfuscating the evidence depicting the state of the in-vehicle sensors that produced the kinematic data, based on which lane-changing decisions are been made, might affect the accuracy of the vehicle manoeuvring. However, conveying all the details of the sensors' data collection process/software might, on the other hand, lead on the fingerprinting of the vehicle (i.e., manufacturer brand) which, in turn, can allow for implementation disclosure attacks [16]. Specifically, *the higher the level of harmonisation, the bigger the impact on the accuracy of the trust decision*, which in turn can affect the safety profile of the application. Therefore, this tradeoff needs to be considered in order to achieve the required level of privacy, without compromising road safety. In Table 6.4, we present an example of such a harmonisation.

Computer	Assets	Description	Harmonised Attribute
Vehicle Computer	1. Intel SGX 2. SW based Tracer (CFI) 3. Secure Boot mechanism 4. Comm. crypto keys	The Vehicle Computer possesses a strong computational unit and supports both symmetric and asymmetric cryptography. It has established keys with all the respective zonal controllers and ECUs to ensure communication integrity. It supports a Trusted Execution Environment, but this information is not to be disclosed. Finally it possesses a SW-based Tracer on CFI mode in order to perform introspection, but this information is also not going to be disclosed, as it can open the path to possible attacks and privacy issues.	1. Design-time Integrity = 1 (Cryptographic Capabilities and RoT) 2. Run-time Integrity = 1 (SW Based Tracer) 3. Communication Integrity = 1 (Established Shared Secret Keys with all communicating devices) 4. Bootup Integrity = 1 (Secure Boot Mechanism providing a fingerprint of the Read Only memory of the computer)

Zonal Controller 1	1. Intel SGX 2. SW-based Tracer 3. Secure Boot mechanism 4. Comm. crypto. keys 5. Certified application running	Zonal Controller 1 possesses a strong computational unit and supports both symmetric and asymmetric cryptography. It has established keys with all the respective Zonal Controllers to ensure communication integrity. It supports a Trusted Execution Environment, but this information is not to be disclosed. Finally, it possesses a SW-based Tracer on CIV mode to perform introspection, but this is also not going to be disclosed, as this information can open the path for possible attacks and privacy issues. Finally, in this Zonal Controller, a certified sensitive data processing application is installed.	1. Design-time Integrity = 1 (Cryptographic Capabilities and RoT) 2. Run-time Integrity = 1 (SW Based Tracer) 3. Communication Integrity = 1 (Established Shared Secret Keys with all communicating devices) 4. Bootup Integrity = 1 (Secure Boot Mechanism providing a fingerprint of the Read Only memory of the computer) 5. Certified application attribute = 1 (Certified application is processing the sensitive data)
A-ECU	1. Intel SGX 2. Secure Boot mechanism 3. Comm. crypto keys	The A-ECU possesses a strong enough computational unit to perform asymmetric cryptographic operations. It has established a secret key with all communicating devices for secure communications and supports a Trusted Execution Environment leveraging the capabilities of Intel SGX.	1. Design-time Integrity = 1 (Cryptographic Capabilities and RoT) 2. Communication Integrity = 1 (Established shared Secret Keys with all communicating devices) 3. Bootup Integrity = 1 (Secure Boot Mechanism providing a fingerprint of the Read Only memory of the computer)
S-ECU	Communication cryptographic keys	The S-ECU possesses a relatively weak computational unit and can only perform symmetric cryptographic functionalities. It should be noted that the S-ECU might have access to a built-in RoT with only root of storage capabilities (e.g., Hardware Storage Module) for securely storing key hierarchies, and thus, pre-shared keys need to be established for communicating with the corresponding communicating devices.	1. Design-time Integrity = 1 (Cryptographic Capabilities based on the use of limited RoT capabilities) 2. Communication Integrity = 0 (Established Shared Secret Keys with all communicating devices, but with limited authentication capabilities)

N-ECU	Communication cryptographic keys	The N-ECU also has access to pre-established keys with the core difference that such devices are not equipped with any secure elements that can expose functionalities for *secure storage, verifiable measurements, and runtime monitoring*	1. Design-time Integrity = 0 (Cryptographic capabilities and no RoT) 2. Communication Integrity = 0 (Established Shared Secret Keys with all communicating devices, but there is no protection by a RoT)

Table 6.4: Example of an Attribute Harmonisation

In the example presented in Table 6.4, the attributes that are not specified in the final column are set to the default values, thus dictating non-relevance in the creation of the Trust model. It has to be mentioned that the harmonisation happens in the trustworthiness claims handler (TCH) and the harmonised attributes depicted in the last column of the table 6.4 are how the actual attributes translate to the harmonisation process. The harmonisation eventually is compiled through a logical operation through the actual attributes.We assume that the TAF possesses a strong safety element and can act like an authorisation entity. The Trustworthiness Claims, apart from the harmonised attributes, contain certificates created by the crypto schemes, each one supporting verifiable evidence of the committed harmonised attributes.

Note that the goal of CONNECT with regard to the materialisation of the harmonised attributes is for all OEMs and CCAM service providers to adopt the same type of abstraction to evaluate the trust level on all aspects of trust defined in D3.1 [11] based on the same types of attributes. One approach that can be followed in this regard is to leverage the International Data Space (IDS) of harmonised attributes to be able to create a generalised/generic trustworthiness profile capturing these abstraction models, as IDS aims to create a universal secure and trustworthy data ecosystem. By adopting this approach, CONNECT aims to align with the Gaia-X standard to push it to all OEMs, not only to ensure the integrity of trust assessments, but also to reinforce the principle that the identification of individual vehicles or breaches of privacy should not be feasible within a V2X environment.

6.1.5 Models and Crypto Primitives for Trustworthiness Evidence Extraction

As mentioned previously, in order to ensure the verifiability, correctness, and integrity of the trustworthiness evidence utilised as part of the Trustworthiness Claims in the context of CONNECT, as well as safeguarding the privacy of the vehicle and the user, it is possible to use a wide variety of crypto primitives, including both encryption schemes and signature schemes. Specifically, for V2X communication, CONNECT envisions the build on top of the Public Key Infrastructure (PKI) and pseudonyms, as standardised by ETSI and 5GAA, for safeguarding the identity and location privacy of communicating vehicles.

With regard to the employed signature schemes, when a component in a vehicle provides a Trustworthiness Claim, it will need to be independently signed in order to confirm the authenticity of the Claim. In addition, the employed signature scheme needs to provide privacy-preserving properties for the identity of the vehicle. It is important to ensure the authenticity of the Trustwor-

thiness claims vector, but with no direct link with any crucial/sensitive information that regarding the vehicle's fingerprint.Apart from high privacy preserving schemes CONNECT needs to support traceability as well, the ability of a Verifier to trace the source of the signed. More specifically, in case of a failed Trust source detection an authorised Trusted entity, for example this entity could be the O.E.M., will be able to trace back to the identity of the failed ECU, in other words controlled linkability.

In addition to signature schemes, in the context of CONNECT it is possible to use symmetric cryptography schemes leveraging symmetric encryption and HMACs in order to ensure the correctness of the integrity of the data shared as part of a Trustworthiness Claim. As it has been outlined in Section 6.1.4 and in D2.1 [13], different types of ECUs may have different cryptographic capabilities:

- S-ECUs are only capable of performing symmetric cryptography, using keys managed by a Hardware Security Module (HSM), which provides a high level of trustworthiness with regard to key management. Note that S-ECU devices which do not possess an HSM are also able to perform symmetric crypto operations. However, in this case the cryptographic keys will not have the hardware-based guarantees provided by the HSM, which will be reflected in the trust calculations performed by the TAF. During the manufacturing phase, the S-ECU is provided with a unique identity key and a symmetric key, but throughout the lifecycle of the system it will be possible to install additional keys to ensure the integrity of communications, to perform access control on the transmitted data, or to prove the provenance of received data.
- A-ECUs are capable of performing both symmetric and asymmetric cryptography, and possess a TEE-guard for securely storing the required cryptographic keys. In addition, these devices utilize key restriction usage policies to ensure that their attestation data can only be signed if the device is in a state that is known to be correct and trustworthy, thus ensuring the integrity of the attestation operation. During the manufacturing phase, A-ECUs are provided with asymmetric signing keys and certificates, which are used when integrating the devices into the vehicle. Note that, as in the case of the S-ECUs, it is possible to install additional keys if needed in the context of a security operation or communication.

One core aspect that needs to be considered in the transmission of trustworthiness evidence is Evidence Freshness, in order to ensure that the evidence is not deprecated, and in order to avoid replay attacks. Towards this direction, we need to consider that (i) the evidence should include a freshness indicator that can be understood by a Verifier, and (ii) the employed Trustworthiness Claim model should support the conveyance of freshness proof in a manner that is useful to Verifiers and their appraisal procedures. In general, the information element contained in a Trustworthiness Claim that is intended to uniquely distinguish evidence and/or determine the freshness of the Evidence is referred to as a Handle. Such a Handle can be used by a Verifier as an indicator of authenticity or attestation provenance, as only Provers and Verifiers who are intended to exchange evidence should have knowledge of the corresponding handles. Examples of such elements include nonces or signed timestamps.

Considering all the above, the trustworthiness evidence extraction and appraisal model can be compounded at a high level in accordance with the outcomes of the RATS Working Group of the Internet Engineering Task Force (IETF) [22], as depicted in Figure 6.3. Specifically, this process consists of the following steps:

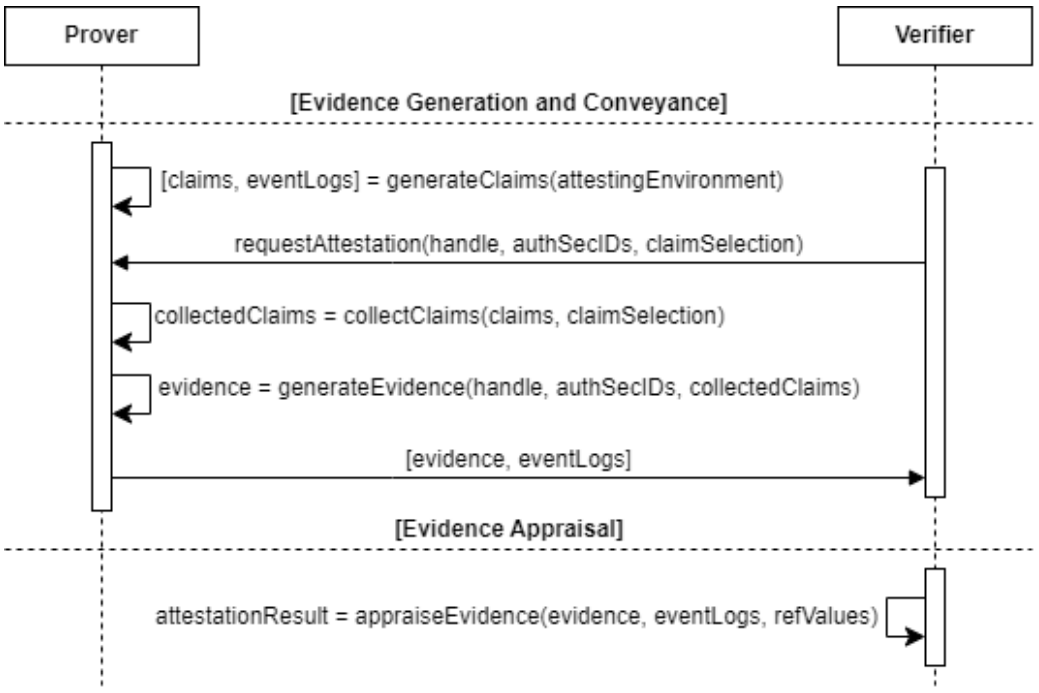

Figure 6.3: High-level trustworthiness evidence extraction and appraisal model.

1. The Prover device boots up, and may collect claims about its boot state or its operational state.
2. The Verifier initiates the attestation procedure, by issuing a request to the Prover to provide one or more Trustworthiness Claims (referred to as Claim Selection), as well as a Handle containing a fresh nonce and the list of Authentication Secret IDs specifying the keys with which the Prover is expected to sign the Trustworthiness Evidence with.
3. The Prover collects the requested Claims based on the Claim Selection.
4. Based on these Claims, the Handle, and the Authentication Secret IDs, the Prover collects the required Trustworthiness Evidence and signs it with the appropriate keys.
5. The Prover transmits the Evidence back to the Verifier.
6. Upon reception of the Evidence, the Verifier performs an Evidence Appraisal process in order to evaluate the correctness of the Evidence. Note that this process is application-specific and depends on the security control used (e.g., attestation or misbehaviour detection).

Considering the requirements and the applied crypto techniques mentioned above, we are proposing some signature scheme that are suitable for CONNECT through simple Alice/Bob examples.

1. Direct Anonymous Attestation (DAA): DAA may be a suitable candidate for CONNECT due to its cryptographic characteristics.Apart from Privacy-Preserving authentication and Zero-Knowledge Proofs that are core aspects of DAA, in CONNECT we want to leverage DAA's controlled linkability, where only authorised entities can trace back to the signer and perform if needed revocation actions.

(a) Alice creates an attestation key pair and in order to acquire a credential for this key she communicates with a Trusted Third party that issues DAA credentials.

(b) When Alice wants to create a DAA signature, she randomises the DAA credential and uses this together with the DAA key pair to sign her message. The randomised credential and the freshly created signature are then sent to be verified.

(c) When Bob receives the DAA signature he verifies it and checks whether the randomised credential was issued by the respective Trusted Third Party (using cryptogaphic pairings). Because the credential is randomised and Bob does not receive the public part of the DAA key, Bob cannot extract any information regarding the Alice's identity.

(d) If Alice is caught lying by Bob, the Trusted Third Party that issued her DAA credential using the basename technique can trace back to the identity of Alice and perform the necessary actions.

2. Direct Anonymous Attestation with Attributes (DAA-A):DAA-A has similar cryptographic characteristics with the DAA, although in this case we focus on the selective disclosure capabilities of DAA-A. In this case a Holder can prove to a verifier the possession of a set of attributes without disclosig them.

   (a) Alice creates an attestation key pair and communicates first with a Trusted Third Party to get her DAA credentials, and then with another Trusted Third Party to acquire her verifiable credential bound to her attestation key and the attributes she possesses.

   (b) When Alice wants to create a Verifiable presentation on a subset of the attributes she claims to have, she first randomises the verifiable credentials and then discloses the desired subset of attributes. Finally, she uses the attestation key to sign a message and finalise the verifiable presentation.

   (c) When Bob receives the Verifiable presentation, he first verifies that the randomised verifiable credential was issued by the correct Trusted Third Party through pairings. He then verifies the verifiable presentation for the disclosed attributes. Because of the randomised credential and the absence of the public part of the DAA key, Bob cannot extract any information regarding the identity of Alice.

   (d) If Alice is caught lying by Bob, the Trusted Third Party that issued her DAA verifiable credential using the basename technique can trace back to the identity of Alice and perform the necessary actions.

3. Threshold Signatures:Another suitable candidate for CONNECT regarding its cryptographic characteristics is threshold signature.In threshold signatures, Trust is distributed to all entities that have a share of the collective secret key.This translates to a ZERO Trust model, in which we assume that any of the participants could be missbehaving.

   (a) Alice, Bob, Carol, and all other participants choose a threshold value T.

   (b) Alice, Bob, and the rest of the participants create their own attestation key pairs, and share their public keys with each other.

   (c) The Group manager (Trusted Third Party) generates a collective public key that represents the entire group, and divides a private key into multiple shares, one for each participant, in respect to the Shamir secret sharing. Thus, no single share can create a valid signature.

(d) To produce a valid signature at least T members of the group collaborate, contributing their own secret share.

(e) Evelyn by possessing the collective/group public key can verify the validity of the Threshold signature, thus she cannot extract any information on who contributed to the creation of the signature, thus protecting the privacy of its signer.

4. Aggregated Verifiably Encrypted Signatures (AVES): Same as threshold signatures, apart from the privacy-preserving authentication AVES offers, CONNECT aims to leverage the distributed Trust modelling that it can achieve.More specifically, we assume that every entity could be misbehaving, so we don't allow any of the other entities to possess an external signature as plaintext data.

   (a) Alice and Bob generate their own attestation key pair and a Trusted Third Party generates an encryption key pair. The public part of the encryption key will be shared with Alice and Bob.

   (b) Alice and Bob both want to sign a message with their individual attestation keys and then encrypt the freshly generated signatures with the shared encryption key. This encryption ensures that the signature remains confidential. The signed message is also included in the encryption, so that the Verifier is able to link the signature to its respective message without revealing its content.

   (c) The two encrypted signatures are then aggregated with the respective homomorphic encryption scheme, dictated by the encryption key.

   (d) When Evelyn wants to verify the signature, she uses the encryption public key and checks with pairing cryptography whether or not the cipher-text/signature is properly generated. Therefore, Evelyn cannot extract any information on the identity of Alice and Bob, as the the committed signatures are wrapped with the encryption key and cannot be verified individually, even if Evelyn possesses the public keys of Alice and Bob.

The cryptographic schemes described above through simple examples can be easily translated to the context of CONNECT, as Alice and Bob could represent any type of communication defined in CONNECT, such as in-vehicle communication between ECUs, Zonal Controllers, and the Vehicle Computer, in vehicle-to vehicle-communications, and vehicle-to-MEC communications. We will further elaborate on the different types of crypto schemes to be leveraged in CONNECT in D4.1.

Throughout the remainder of this Chapter, we will expand on the aforementioned data model, by providing details on the approaches followed in CONNECT with regard to Trustworthiness Evidence collection, as well as the generation and evaluation of TCs.

6.2 Trustworthiness Claims Data Structures & Encoding

To support the expression of harmonised attributes so that an external Verifier (another CCAM entity; e.g., Vehicle) can assert the trustworthiness level of a communicated CAM/CPM message (and all involved software and hardware assets that were part of the in-vehicle service graph chain towards extracting and processing the specific type of kinematic data) in a privacy-preserving manner, as described before, the YANG (Yet Another Next Generation) data modelling language

will be adopted and expanded. YANG is maintained by the NETMOD working group in the Internet Engineering Task Force (IETF) and enables a common understanding of how to interpret the encoded trustworthiness evidence (as part of the generated Trustwrothiness Claim) and asses the existence/ownership of the defined trust properties (cf. Section 6.1.4), by the attesting entity, enabling efficient and accurate trust-aware decision making decisions. It has been designed explicitly for facilitating a Verifier's ability to interpret and assess security reports and claims on the trustworthiness state of a node towards the construction of Trusted Network Topologies [32]. Thus, it already provides appropriate data structures for encoding different types of trustworthiness evidence; in the current specification, focus have been given on evidence stemming from instantiating attestation enablers but CONNECT will extend these models with the necessary abstractions for also composing claims comprising trustworthiness evidence as outputted by other security controls such as Misbehavior Detection, been one of CONNECT's main trust sources. YANG models the hierarchical organization of (trust-related) data as a tree in which each node has a name, and either a value or a set of child nodes. The definition of modules allows for the hierarchical definition of data structures and the segmentation into modules and submodules. It defines a set of built-in types but also supports the definition of new ones. Groupings of nodes permit the definition and reusability of information among different modules, even specified by different entities (e.g., standardisation bodies, institutions, organisations, individuals). An important feature of this data modelling language is the ability to convert YANG modules to XML syntax and vice versa without any loss of granularity and accuracy, thus, leveraging all the XML-related functionalities (XML parsers, XSL transformations) and supporting interoperability with legacy systems that have already been equipped with XML-based data connectors for communicating also with the backend infrastructure (e.g., OEMs) towards the provision of services such as SW/FW update and Quality-of-Service monitoring.

These functional capabilities and the design principles of the YANG data modelling make this language a perfect candidate to accommodate the needs of CONNECT Trust Assessment Framework (TAF) [11] as it pertains to the execution of trust-related data sharing agreements with different levels of privacy: covering both extremes of *full disclosure*, as part of in-vehicle trust assessment, where all details and evidence associated with the output of runtime security controls are communicated to the TAF, as well as *selective disclosure*, as part of V2E communication, where only harmonized/obfuscated attributes are shared with external entities to perform their local trust assessment. In fact, the YANG data model covers all of the design principles and requirements presented in Section 6.1.2. Specifically, the YANG modelling language provides the means for defining a simple data model grouping only those specific sets of primitives (e.g., identity, feature, typedef, grouping) needed to expose the necessary data (i.e., modules).

In parallel, IETF's Trusted Path Routing specification [32], provides a set of templates with respect to encoding attestation evidence for network devices. Such templates set the baseline for been able to express, in a simple (and verifiable) manner, the assurance (attestation) evidence reported by the ECU devices or even the MEC virtualisation infrastructure where the various services have been instantiated. Apart from such trustworthiness evidence, these templates can be extended to encode the harmonized trustworthiness claims in case they have to be communicated outside the vehicle in a privacy-preserving manner. Subsequently, the expressiveness of the YANG data modelling language allows for granular description of various trust sources coming from different security controls. For instance, it is possible to encode attestation evidence related to the integrity measurements, reported by a device equipped with a secure element (such as a Trusted Execution Environment), while also facilitate the modelling of the output of runtime Misbehaviour Detection controls and plausability checks on the veracity of (kinematic) data produced by the underlying sensors and actuating ECUs. This diversity is a core enabler and feature supporting

rich and heterogeneous trust-related data communication guiding the overall trust assessment process. Additional YANG templates showcase how to associate freshness with the exchanged trustworthiness evidence or how to include cryptographic primitives for ensuring their authenticity and integrity.

Finally, it is crucial that the designed data model can support all the phases of a trust assessment process - starting from triggering trust-related operations (initiated by the TAF based on pre-defined trust policies deployed by authenticated CCAM stakeholders through the Blockchain infrastructure) to the runtime operation of the Trust Level Expression Engine for calculating data- and/or node-centric trust opinions. In particular, the data model should encode the Trust Assessment Request (TAR) which is sent by the Trust Assessment Framework towards the Attestation Integrity Verification (AIV) component. Through the TAR data structure, the TAF component is able to specify the IDs of the in-vehicle components that need to be assessed, as part of the required trust relationship assertion; the attestation evidence to be collected based on the trust property of interest (e.g., *integrity*, *security*, *availability*,*robustness*); as well as parameters regarding the type (i.e., synchronous/asynchronous) and periodicity of the collection process. Subsequently, the data model should be able to cope with devices of different (trust) capabilities that directly affects the type of evidence to be provided and are associated to a different Level of Assurance. Recall, for instance, that in-vehicle topologies comprise both asymmetric-capable ECUs and symmetric-capable ECUs which dictate the type of crypto primitives and authentication mechanisms to be employed for the secure communication of such trust-related evidence. In the same context, devices might be equipped with different Root-of-Trust (RoT) variants supporting different functionalities ("*RoT for Storage, Measurement and Reporting*") while exhibiting various levels of assurance; i.e., a HW-based RoT can enable a higher level of isolation when executing safety-critical applications than a SW-based secure element where cryptographic primitives (i.e., secret keys) are also stored at the untrusted operational layer of the device [12]. An asymmetric ECU, with a capable RoT, will be able to report attestation results as signatures bounded to the key restriction usage policies, enforced in the device during on-boarding, thus, guiding the usage of the underlying (HW-based) Attestation Key to be allowed by the RoT *if and only if* the device is at an expected state. On the other hand, symmetric ECUs - with solely secure storage capabilities - can only report state quotes with no guarantees on the integrity of the extracted device configuration and behavioural measurements.

All of these results are available to the AIV component which verifies the collected integrity measurements, for each device, aggregates them as part of the Trustworthiness Vector and forms the overall attestation evidence for the entire service graph chain. This attestation evidence is sent back to the TAF for performing the trust assessment analysis in the context of the predefined trust model. Eventually, the attestation evidence, the trust opinion produced by the TAF and other trustworthiness evidence (i.e., misbehaviour detection module) are forwarded to the Trustworthiness Claims Handler (TCH) which produces the Trustworthiness Claims: This data model includes the harmonized version of the attestation evidence (trustworthiness claims), as reported by the AIV component, aggregated with the other type of trust assertions. The entire Trustworthiness Claim payload is signed by the Identity and Access Management (IAM) component using PKI-provided pseudonyms. These actions ensure the privacy-preserving knowledge sharing with entities external to a vehicle (e.g., Digital Twin TAF running on a MEC infrastructure), thus mitigating the device fingerprinting issue highlighted in Section 6.1.3.1. Based on the aforementioned functionalities an initial YANG module is drafted in the following subsections in order to support all the various required data models. The concrete and final data model for the CONNECT YANG module is presented in D4.2 [14].

Name	Description/Relevance with CONNECT
Hardware	The AIV component has appraised an ECU's hardware and firmware which are able to expose fingerprints of their identity and running code.
Executables	The AIV component has appraised and evaluated relevant runtime files, scripts, and/or other objects which have been loaded into the ECU's (Target) environment's memory
Configuration	The AIV component has appraised an ECU's configuration, and is able to make conclusions regarding the exposure of known vulnerabilities.
Instance Identity	The AIV component has appraised an ECU's unique identity based upon verifiable evidence which can be correlated to a unique instantiated instance of the ECU. This is also related to the harmonized attribute that signals whether an Attesting device is a certified device according to the information stored in the IAM component.

Table 6.6: Common attestation attributes as a trust source in the context of CONNECT

6.2.1 Attestation Attributes as Trust Source

As described in D2.1 [13], one of the core functionalities of the Trust Assessment in CONNECT is its ability to take into consideration different type of trust sources (i.e. runtime attestation security controls, output from misbehaviour detection, output from intrusion detection system). In the scope of this deliverable, the primary goal is to demonstrate the harmonization capabilities which ensure the sharing of trustworthiness evidence in a privacy preserving manner. Thus, when it comes to the trust properties discussed, we focus on the integrity as one of the core trust pillars considered in all safety-critical applications (and in the envisioned use cases): *CAM/CPM messages need to encode kinematic data that have been extracted from in-vehicle sensors with high level of integrity while at the same time been verified for their correctness as part of the Misbehavior Detection process*. A complete analysis of the trust sources, associated with each one of the trust properties defined in D3.1 [11], will be listed in D4.2 [12] and D4.3 [14].

Most of the Trust Sources collected in scope of a Trust Assessment process, refer to attestation attributes which are used to monitor the correct status of a device. Regardless of the attesting environment adopted, it is possible to define a common set of attestation attributes that capture that a device is in a correct state. Independently of whether the trusted execution environment is process-based (e.g., Intel SGX), VM-based (e.g., Intel TDX) or use a hardware security module (e.g., use of a TPM), it is possible to express common attestation attributes while maintaining an adequate level of granularity required by the TAF to perform the trust assessment analysis. Finally, the attestation attributes should be defined in a way that allows the obfuscation of information in case the attestation attributes have to be sent outside of the vehicle. Table 6.6 provides a list of attestation attributes as specified in the respective IETF YANG data models concerning the attestation results reported by a device. These data integrity attestation attributes are fully covering the CONNECT ecosystem.

6.2.2 Example Description on C-ACC Use Case

To identify the necessary data models for the Trustworthiness Claims within the CONNECT framework, we consider a simplified example from the Cooperative Adaptive Cruise Control (C-ACC) use case, depicted in Figure 6.2: This scenario assumes the existence of a Vehicle Computer (with a C-ACC functional component instantiated) which collects data from ECUs with symmetric

cryptographic capabilities. For example, a symmetric ECU (S-ECU 1) collects information from a Camera and a Global Navigation Satellite System (GNSS), while a second symmetric ECU (S-ECU 2) collects information from a lidar sensor. To ensure that the information from the sensors is computed and transmitted correctly it is important to build a trust opinion for each one of the participating links among the devices. This is achieved through the TAF which requires a set of trustworthiness evidence to be collected from all the participating devices.

In particular, the Trust Assessment Framework (TAF) component sends a request to the Attestation Integrity Verification (AIV) component which initiates the collection and verification of attestation evidence. In the request, TAF specifies the trust sources that should be collected by the AIV from the devices forming a specific link (i.e., trust relationship). An initial proposal for the YANG data models for the TAF request as well as the attestation results reported by AIV is presented in section 6.2.5.

The collected attestation results mainly enable the form of a trust opinion by the TAF for a trust relationship within a vehicle. However, there are cases where Trustworthiness Claims need to be shared with other entities outside of a vehicle. Such a case is discussed in section 4.1 where a task offloading operation may take place between a vehicle and a MEC infrastructure. In this case, it is important to provide a Trustworthiness Claims data model that preserves privacy and addresses the risk of the device fingerprinting. This is achieved through the harmonization of the attestation evidence through the Trustworthiness Claims Handler (TCH). A proposed example for such a Trustworthiness Claims data model is specified in 6.2.6. Figure 6.4 describes the associations of the data models presented in this section. Each edge label signals the subsection where each data model is discussed.

6.2.3 Generic Information Elements

Plenty of work has already been achieved by numerous IETF specification with respect to formally defining data models for expressing attestation results and Trustworthiness Claims. In particular, the IETF specification on ”Trusted Path Routing” [32] has provided a minimum set of Trustworthiness Claims that attesting devices shall satisfy in order to be part of a trusted network topology. The scope of this subsection is to introduce important YANG data structure that are reused in the CONNECT data models defined in the following sections. The elements defined in this subsection are either reusing or extending some of the concepts introduced in the IETF specifications.

6.2.3.1 Device Capabilities

First and foremost, in the context of CONNECT it is essential to have the ability to annotate the various capabilities of an asset. This is achieved through the definition of YANG features. Features are attributes that contribute to the definition of elements within the data model that are applicable provided that the respective attribute is satisfied. That being said, the following features are identified:

```
feature asymmetric-crypto {
    description
       "Signals that a device (e.g., ECU) supports asymmetric cryptographic
         operations";
}
```

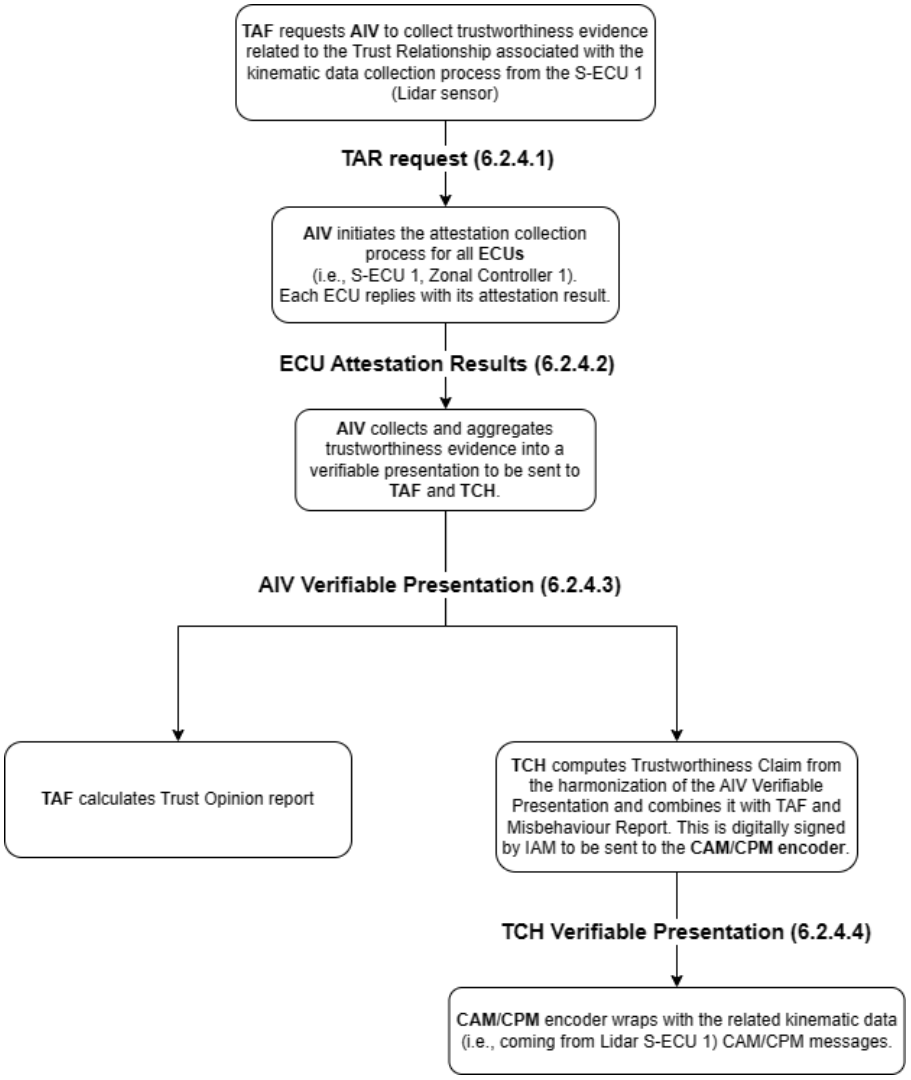

Figure 6.4: Diagram showcasing the different Yang Data Models

```
feature symmetric-crypto-only {
    description
       "Signals that a device (e.g., ECU) supports only symmetric crypto-
        graphic operations";
}

feature capable-rot {
    description
       "The ECU has a capable Root of Trust (RoT) in the sense that it
        provides the means for performing secure storage, secure measu-
        rement and secure reporting of its state.";
}

feature secure-storage-only {
    description
       "The ECU has a Hardware Secure Module (HSM) which provides secure
```

```
       storage capabilities";
}
```

6.2.3.2 Trust Properties

Secondly, it is important to define a set of identities signalling the trust property that a data model refers to. For instance, as thoroughly stated in the next subsection, there are cases where the attestation evidence from specific trust sources need to be collected for a given trust property. As shown below, the trust properties are identities that inherit from a base identity, namely ”trust-property”:

```
identity trust-property {
    description
       "Base identity signalling a trust property in the context of CONNECT";
}

identity integrity-property {
    base trust-property;
    description
       "Indicates the Integrity trust property";
}

identity security-property {
    base trust-property;
    description
       "Indicates the Security trust property";
}

identity availability-property {
    base trust-property;
    description
       "Indicates the Availability trust property";
}

identity robustness-property {
    base trust-property;
    description
       "Indicates the Robustness trust property";
}
```

Theses YANG identities can be used for the data model of the Trust Assessment Request sent by the TAF component. As discussed in subsection 6.2.4, the ”trust-property” base identity can be used in order to signal what types of attestation evidence needs to be collected per trust property. An example on how we could structure the attestation request per trust property is depicted below:

```
container attestation-per-trust-property {
```

```
    leaf trust-property {
        type identityref {
           base trust-property;
         }
    }

    list attestation-evidence-items {
      key "attestation-evidence-id";
      description "List of results";

      leaf attestation-evidence-id {
        type string;
        description "Unique name of the attestation result to be reported
        by an attester";
      }

      leaf report {
        type boolean;
        description "True if this attestation result should be reported
        by the attester.";
      }
    }
}
```

To provide an even more simplified data model tailored to the integrity trust property, the proposed data model could look like the following:

```
container attestation-per-trust-property {
    leaf trust-property {       \\ Based on the simplified data model
                                \\this should be
                                \\ set to "integrity-property"
        type identityref {
           base trust-property;
         }
    }

    leaf report-sw-version {
        type boolean;
    }
    leaf report-secure-boot {
        type boolean;
    }
    leaf report-secure-comms {
        type boolean;
    }
    leaf report-control-flow-integrity {
        type boolean;
    }
}
```

Similarly, the trust-property identity can be used to encode the harmonized attribute in the context of the Trustworthiness Claim data. This data structure is explained in more detail in the following subsection on ”Trustworthiness Appraisal”. Specifically, the TCH component collects the verifiable presentation sent by the AIV component and structures the Trustworthiness Claim. Each of the harmonized attributes within a Trustworthiness Claim can be conceived as a piece of information related to a trustworthiness appraisal of a specific characteristic in the context of a particular trust property. This information is constructed in a privacy-preserving manner in order to avoid disclosing the identity of a vehicle.

6.2.3.3 Trustworthiness Appraisal

As stated in subsection 6.2.5, different types of trustworthiness evidence is reported from various trust sources in the Trustworthiness Claim data structure. To provide a common way of encoding the results, a single generic data type is required. This is achieved through the definition a new YANG typedef, namely the ”trustworthiness-appraisal” typedef which exposes specific encoded values, thus simplifying the processing of the enumerations by the the consumer of a Trustworthiness Claim. Following the IETF's enumeration encoding defined in [21], the associated value is encoded as a single signed 8 bit integer. For the value there are four different Trustworthiness Tiers:

1. Affirming, values (2)-(31), (-2)-(-32): The Verifier (e.g., AIV) affirms the Attester support for this aspect of trustworthiness,
2. Warning, values (32)-(95), (-33)-(-96): The Verifier (e.g., AIV) warns about this aspect of trustworthiness,
3. Contraindicated, values (96)-(127), (-97)-(-128): The Verifier (e.g., AIV) asserts the Attester is explicitly untrustworthy in regard to this aspect,
4. None, values 0, -1, 1: The Verifier (e.g., AIV) makes no assertions about this Trustworthiness Claim. For example, the Verifier may return status '-1' to signal due to an error in its process.

```
typedef trustworthiness-appraisal {
   type int8;
   description
       "A Verifier asserted value designed to enable a common
         understanding of a Verifier trustworthiness appraisal.  The
         value assignments for this 8 bit signed integer will follow
         these guidelines:

         None:
           - Value 0: The evidence received is insufficient to make a
           conclusion. Note: this should always be always treated
           equivalently by the Relying Party as no claim being made.
           I.e., the RP's Appraisal Policy for Attestation Results
           SHOULD NOT make any distinction between a Trustworthiness
           Claim with enumeration '0', and no Trustworthiness Claim
```

```
            being provided.
            - Value 1: The Evidence received contains unknown elements
            which the Verifier is unable to evaluate. An example might
            be that wrong type of evidence has been delivered. Another
            case is that of Evidence coming from a composite Attester:
            Verifier may partially understand and leave as "unknown"
            the claims related to features it can't appraise.
            - Value -1: A verifier malfunction occurred during the
            Verifier's appraisal processing.

          Affirming: The Verifier affirms the Attester support for this
          aspect of trustworthiness
            - Values 2 to 31: A standards enumerated reason for affirming.
            - Values -2 to -32: A non-standard reason for affirming.

          Warning: The Verifier warns about this aspect of trustworthiness.
            - Values 32 to 95: A standards enumerated reason for the
              warning.
            - Values -33 to -96: A non-standard reason for the warning.

          Contraindicated: The Verifier asserts the Attester is explicitly
          untrustworthy in regard to this aspect.
            - Values 96 to 127: A standards enumerated reason for the
              contraindication.
            - Values -97 to -128: A non-standard reason for the
              contraindication.";
}
```

This type of encoding allows different consumer applications of a Trustworthiness Claim to easily understand and process the results of harmonized attributes originated from various trust sources. In some cases, such as the harmonized attributes associated with the attestation evidence, it is straightforward to define such an encoding to represent the different states of a attestation verification procedure. However, this cannot apply in all cases. For instance, the trust opinion computed by the TAF component consists of values ranging from 0-1. Thus, the goal is to provide a mapping to associate the numerical values of the belief, disbelief, uncertainty and Actual Trust Level (ATL) provided by the TAF component with a set of concrete states signalling the decision made by the component and reported in the Verifiable Presentations. Detailed definition of this mapping is provided in D4.2 [14].

```
container harmonized-attribute {
    leaf trust-property {
        type identityref {
          base trust-property;
        }
    }
    leaf characteristic {
        type string;
    }
```

```
    leaf value {
        type trustworthiness-appraisal;
    }
}
```

An example of how the TCH component could serialize in JSON one harmonized attribute before including it in a Trustworthiness Claim is presented below:

```
harmonized-attribute: {
    trust-property: "integrity-property",
    characteristic: "boot-up state",
    value: 1
}
```

The above example shows the association of a harmonized attribute with a particular trust property. Based on the harmonized attribute categories mentioned in subsection 6.1.4, it is straightforward to encode one of the presented attributes according to the harmonized-attribute data model. For example, for the ”Boot-up integrity” attribute the identity of the structure should be defined to ”integrity-property”, the characteristic attribute should be set to ”boot-up state” and the value attribute should be mapped to one of the defined trustworthiness appraisal encoding values mentioned in 6.1.4.

6.2.3.4 Modes of Trust Assessment Request

As discussed in 6.2.4, concerning the Trust Assessment Request sent by the TAF component, it is possible to specify whether the request shall be treated by the AIV component in a periodical manner (e.g., provide attestation evidence every 5 seconds) or asynchronously (e.g., provide attestation evidence when a change in the Trust Relationship has been detected). To support the signalling of the TAR type the following identities are defined:

```
identity tar-type {
    description
       "Base identity signalling the type of a trust assessment request";
}

identity synchronous-tar {
    base trust-property;
    description
       "Indicates that the AIV forwards attestation reports from the ECUs
       in a periodic fashion. Frequency should be specified in a
       separate attribute.";
}

identity asynchronous-tar {
    base trust-property;
    description
        "Indicates that the AIV forwards attestation reports from the ECUs
        once a change in an ECU state is observed.";
}
```

6.2.3.5 Verifiable Evidence

Finally, two important YANG grouping definitions are related to the verifiable evidence that a device can provide depending on its cryptographic capabilities. Specifically, in the cases where a device supports asymmetric cryptography capabilities (e.g., A-ECU), we can define an enriched data structure that acts as a verifiable evidence ensuring the authenticity and integrity of a payload. Thus, a new asymmetric-evidence grouping is defined for this purpose. It consists of an appraisal-timestamp attribute which ensures the freshness of the reported payload, and the digital signature of the payload including the appraisal-timestamp. The signature algorithm and the certificate associated with the key pair used for the digital signature is also included in the evidence structure. Finally, it is also worth stating that this grouping can be used under the condition that the feature ”asymmetric-crypto” is satisfied (i.e., an asymmetric evidence can be generated by a device that supports asymmetric cryptographic capabilities).

```
grouping asymmetric-evidence {
    if-feature "asymmetric-crypto";
    description
        "Evidence generated by the Signer of the payload.";
    leaf appraisal-timestamp {
        type yang:date-and-time;
        mandatory true;
        description
           "The timestamp reported by the Signer entity. This can be used
            by a Relying Party to determine the freshness of the reported
            content.";
    }
    leaf signature-algorithm-type {
        type string;
        mandatory true;
        description
            "Platform asymmetric algorithm used by the Signer.";
    }
    leaf signature {
        type binary;
        mandatory true;
        description
            "Signature from the related content";
    }
    leaf verifier-certificate-keystore-ref {
        type string;
        mandatory true;
        description
            "A reference to a specific certificate to an asymmetric key
            of the Signer which can be used to validate the 'signature'
            attribute.";
    }
}
```

Similarly, in the case where a device supports only symmetric cryptographic operations a new

grouping shall be specified. This grouping, namely symmetric-evidence is applicable only when the ”symmetric-crypto-only” feature is satisfied for an asset. It contains the appraisal-timestamp reported by the device and the symmetric cryptographic algorithm and the ciphertext produced by the encrypting the related content and the appraisal timestamp.

```
grouping symmetric-evidence {
    if-feature "symmetric-crypto-only";
    description
        "Evidence generated by a device supporting only symmetric crypto
        functionalities. Symmetric encryption of the Verifier across all
        the current objects in trustworthiness evidence including the
        appraisal-timestamp";
    leaf appraisal-timestamp {
        type yang:date-and-time;
        mandatory true;
        description
           "The timestamp reported by the device. This can be used by the
           consumer of the evidence to determine the freshness of the
           related content.";
    }
    leaf algorithm-type {
        type string;
        mandatory true;
        description
            "Symmetric cryptographic algorithm used by the device.";
    }
    leaf encryption-value {
        type binary;
        mandatory true;
        description
            "The ciphertext produced from the encryption of the payload
            consisting of the related content and the appraisal timestamp
            value.";
    }
}
```

6.2.4 Verifiable Credentials for In-Vehicle Trust Assessment

This section presents how the YANG data modelling language can be used in order to address the data models defined in the introduction of section 6.2:

1. Trust Assessment Request from TAF to AIV,
2. Attestation Results from ECU to AIV,
3. Attestation Evidence from AIV to TAF and TCH,
4. Verifiable Presentation from TCH to CAM/CPM encoder (i.e., to be wraped alon with the CAM/CPM payload in order to be transmitted to other vehicles or any back-end system).

This is an initial proposal on how a YANG module - ”connect-verifiable-credentials” - can be defined in order to cover all the aforementioned cases. The example of subsection 6.2.3 is also used to illustrate the specifics of each data structure. The concrete definition of the data models is provided in D4.1 [12].

6.2.4.1 Trust Assessment Request

The Trust Assessment Request (TAR) is the payload sent by the TAF to the AIV component in order to collect attestation evidence for a set of devices (i.e., ECUs) comprising a Trust Relationship. Different trustworthiness evidence can be collected for a given trust property. Specifically, in the TAR payload it is possible to specify a unique identifier for the request (i.e., uuid attribute) and a list of the devices that comprise the Trust Relationship (i.e., attester attribute). For each device, we may have zero, one or more requests (i.e., attester-request attribute) depending on the number of trust properties that we want to assess. For each trust property, the TAF component specifies the type of attestation evidence that needs to be collected by the AIV component (see subsections 6.2.3.1, 6.2.3.2). The entire tar payload is digitally signed by the TAF component (see subsections 6.2.3.5). A YANG Tree Diagram that could describe the TAF request data model is presented below.

```
module: connect-verifiable-presentations
+--rw trust-assessment-request
| +--rw tar-payload
| | +--rw uuid string
| | +--rw attesters-information
| | | +--rw attester* [id]
| | | | +--rw id string
| | | | +--rw name? string
| | +--rw attester-requests
| | | +--rw attester-request* [uuid]
| | | | +--rw uuid string
| | | | +--rw attester_id string
| | | | +--rw attestation-for-trust-properties
| | | | | +--rw attestation-per-trust-property* [trust-property]
| | | | | | +--rw trust-property identityref
| | | | | | +--rw report-sw-version? boolean
| | | | | | +--rw report-secure-boot? boolean
| | | | | | +--rw report-secure-comms? boolean
| | | | | | +--rw report-control-flow-integrity? boolean
| | +--rw tar-configuration
| | | +--rw tar-type identityref
| | | +--rw interval-format? string
| | | +--rw interval? string
| +-- taf-signature asymmetric-evidence
```

The final part of the TAR payload data structure is related to the specification of the way that this request is handled by the TAF component (see subsection 6.2.3.4). In particular, it is possible for the TAF component to request attestation evidence periodically - within a time interval - or

in an asynchronous manner provided that the AIV detects any change for the Trust Relationship specified in the TAR payload. Regarding the former case, the periodicity of the reports by the AIV can be determined using a numerical value signalling a time unit (e.g., every 10 seconds, every 10 minutes) or even using a cron expression to allow the definition of more complex statements (e.g., signal that the collection of evidence shall be executed every 30 minutes from Monday to Friday: ”0,30 * * * 1-5”). For this purpose, the interval-format and interval attributes are defined.

As a conclusion, it is worth mentioning that in the case of the periodic collection of attestation evidence for a Trust Relationship it is important to take into consideration the resource consumption that such a process might have for the devices involved (i.e., the AIV component which collects attestation results and the devices comprising the Trust Relationship). As the attestation evidence collection process gets more frequent, it is possible to collect near to real-time results regarding the state of the attested device. Striking a balance between maintaining a continuous detection mechanism and conserving resources involves a complex trade-off. On one hand, a continuous system can provide real-time insights and timely responses to potential issues, which can be critical in scenarios where immediate action is necessary. On the other hand, running such a system can lead to a significant allocation of computational power and energy, potentially resulting in unnecessary costs and environmental impact.

6.2.4.2 ECU Attestation Results

Once the AIV receives a Trust Assessment Request from the TAF component it initiates the attestation collection process between the requested ECUs. Each ECU is able to provide attestation results depending to its capabilities. The variety of the ECU devices require an expressive data model to capture the different cryptographic capabilities as well as the attestation results.

From the example defined in subsection 6.2.3 it is evident that some ECUs have a Trusted Execution Environment (TEE) while other ECUs posess a hardware security module (HSM). If an ECU has a capable root of trust (i.e., supports secure storage, measures and reports its state) then it is possible to provide as an attestation result a signature signed by the TEE. Since this ECU has local attestation capabilities, it is able to provide a signature that is bound to the key restriction usage policy enforced in the TEE. This allows the verifier - i.e., the AIV component in our case - to validate the signature without requiring any additional information. On the other hand, when an ECU supports only secure storage capabilities the provided attestation result can be a TPM quote which enables the verification of the contents of the TPM’s Platform Configuration Registers. When the AIV receives such an attestation result, it needs to access the Identity Access Management (IAM) component in order to verify the TPM quote against a set of golden hashes which act as reference measurements and ensure the correctness of the ECU’s state.

In the snippet below, we showcase how the features ”capable-rot” and ”secure-storage-only” can be used in order to distinguish the two different categories of ECUs and the type of attestation evidence that they can provide (see subsection 6.2.3.1). Regarding the ”capable-rot” feature, the result attribute is called ”tee-attestation-result” and refers to the signature provided by the TEE while the ”secure-storage-only” feature results in the ”hsm-attestation-result” attribute referring to the TPM quote provided by the HSM.

```
module: connect-verifiable-presentations
+--rw ecu-attestation-result-verifiable-credential
| +--rw (attestation-result)?
| | +--:(tee-attestation-result) {capable-rot}?
```

```
| | |  +--rw tee-attestation-result? binary
| | +--:(hsm-attestation-result) {secure-storage-only}?
| | |  +--rw hsm-attestation-result? binary
```

Finally, it is worth mentioning that the ecu-attestation-result-verifiable-credential data structure is intended to be stored in a blockchain system for auditing reasons. Specifically, this evidence allows any application to analyze, reproduce and assess the correctness of the appraisals performed by the AIV component. This enhances the capabilities of both auditing and explaining the AIV decisions.

6.2.4.3 AIV Verifiable Presentation (Converting attestation evidence to a trust source)

One of the key functionalities of the AIV component is its ability to aggregate the attestation results collected from the devices (i.e., ECUs) as requested in the TAR payload. Then the AIV component - acting as a verifier - is able to assess a set of attestation attributes for each ECU and for each trust property. For this data model of the attestation evidence by the AIV, it is possible to report the software version running on an ECU (i.e., ”sw-version” attribute). Secondly, the AIV can report whether an ECU offers secure boot type of operation (i.e., ”secure-boot” attribute). AIV is also able to report on the integrity of communications of an ECU (”secure-comms” attribute). Finally, the ”control-flow-integriy” attribute allows the AIV to signal that the software stack running on an ECU (e.g., software for extracting/managing kinematic data) exhibits the expected behaviour. Once the AIV collects all the information for all the ECUs requested by a TAR payload, the AIV provides its own trustworthiness evidence (i.e., signature from its TEE). The entire payload is eventually signed by the AIV (”aiv-evidence” attribute, see subsection 6.2.3.5) and it is sent back to the TAF component in order to initiate the calculation of the trust opinion for the respective Trust Relationship.

```
module: connect-verifiable-presentations
+--rw attestation-evidence-verifiable-presentation
| +--rw uuid string
| +--rw trust-source-payload
| | +--rw ecu-attestation-report* [ecu-id]
| | | +--rw ecu-id string
| | | +--rw attestation-evidence-for-trust-properties
| | | | +--rw attestation-evicence-per-trust-property* [trust-property]
| | | | | +--rw trust-property identityref
| | | | | +--rw sw-version? string
| | | | | +--rw secure-boot? boolean
| | | | | +--rw secure-comms? boolean
| | | | | +--rw control-flow-integrity? boolean
| +--rw aiv-trustworthiness-evidence
| | | +--rw nonce? string
| | | +--rw attestation-hash? String
| +--rw aiv-evidence? asymmetric-evidence
```

The attestation evidence reported by the AIV component for a Trust Relationship is also forwarded to the Trustworthiness Claim Handler as we discuss in the following subsection.

6.2.5 TCH Verifiable Presentation (TC Encoding & Abstraction for Trust Assessment vs. Privacy Interplay)

The endmost goal is to provide the computed trustworthiness evidence along with the data items that are being transmitted from a vehicle. In fact, in the C-ACC example presented in 6.2.3, the aim is to include the verifiable presentations as part (i.e., extension) of the CAM/CPM serialization specified by ETSI. The added information can be then consumed by applications (e.g., collision avoidance detection, traffic analysis) that can evaluate the level of trust referring to the process of extracting and communicating the kinematic data from the actuator (e.g., Lidar S-ECU) to the C-ACC in-Vehicle Computer. Given that CAM/CPM messages are intended for cross vehicle and vehicle-to-backend communication, the trustworthiness information for a data item (e.g., extraction and processing of kinematic data within a vehicle) shall be encoded in a privacy preserving way in order to avoid any type of vehicle fingerprinting.

The final data model described in this section refers to the encoding of the Verifiable Presentation created by the Trustworthiness Claim Handler (TCH). In scope of this deliverable we focus on the attestation attributes as a trust source as well as how they could be harmonized in order to enable privacy preserving knowledge sharing. However, in the context of CONNECT, this Verifiable Presentation consists of the Trustworthiness Claim the TAF trust opinion and Misbehaviour Detection report. To ensure the protection of the identity of the vehicle the Verifiable Presentation is digitally signed by the IAM component through the use of PKI pseudonyms before being sent to the CAM/CPM encoder.

In the snippet presented below, we showcase an example of a TCH Verifiable Presentation data model reported for the Trust Relationships related to the entire process of extracting and processing kinematic data from a Lidar ECU. This Verifiable Presentation consists of the Trustworthiness Claim (i.e., ”trustworthiness-claim” attribute, subsection 6.2.3.3) that signals the harmonized attributes per trust property as computed and digitally signed by the TCH component. Additionally, it contains the aggregated TAF trust opinion (i.e., ”taf-trust-opinion-report” attribute) for the participating Trust Relationships. Similarly, the Misbehaviour Detection report is also appended (i.e., ”misbehaviour-VC” attribute). The ”verifiable-presentation-items” and ”data-item” attributes are digitally signed by the IAM component with PKI pseudonyms in order to ensure the protection of the identity of the vehicle. Eventually, the produced tch-verifiable-presentation is forwarded to the CAM/CPM encoder which wraps it along with the related kinematic data to be shared with the intended applications.

```
module: connect-verifiable-presentations
+--rw tch-verifiable-presentation
| +--rw data-item? string
| +--rw verifiable-presentation-items
| | +--rw trustworthiness-claims
| | | +--rw harmonized-attributes-VC
| | | | +--rw harmonized-attribute* [trust-property characteristic]
| | | | | +--rw trust-property identityref
| | | | | +--rw characteristic string
| | | | | +--rw value trustworthiness-appraisal
| | | +--rw tch-evidence asymmetric-evidence
| | +--rw taf-trust-opinion-VC
| | | +--rw taf-trust-opinion-report
```

```
| | | | +--rw belief? uint32
| | | | +--rw disbelief? uint32
| | | | +--rw uncertainty? uint32
| | | | +--rw atl? uint32
| | | +--rw taf-evidence asymmetric-evidence
| | +--rw misbehaviour-VC
| | | +--rw misbehaviour-report
| | | | +--rw report? uint32
| | | +--rw misbehaviour-evidence asymmetric-evidence
| +--rw iam-evidence asymmetric-evidence
```

Chapter 7

Conclusions & Future Work

The document at hand constitutes the first report on the CONNECT WP5 work. It gathers and presents the progress made along two CONNECT research fronts. Both are of significance for the coming WP5 orchestration development, the introduction and usage of claims (also in WP4) as a basis for the trust assessment mechanism (developed in the context of WP3) and notably, the CONNECT use cases implementation to follow.

The first front is a detailed state-of-the-art analysis for the algorithms and solutions to address the efficient transfer (i.e., migration/offloading) of a demanding task from one point (i.e., vehicle device) to another location, whether the latter is a more powerful device or the resource-rich infrastructure. The deliverable stated two problems (i.e., task -migration and -offloading) and having identified the key-parameters that shape them, proceeded with a comprehensive presentation of previous work accounting for those parameters. A careful taxonomy of background solutions was introduced based on the aforementioned state-of-the-art presentation to serve the purposes of a relevant overview. Subsequently, a more focused description of the automotive needs (through a relevant in-vehicle example) assisted the identification of those characteristics that the CONNECT solution should exhibit.

The second front amounts, firstly, to the definition of the data models adopted for embodying the trustworthiness evidence to convey device state information based on which the dynamic trust assessment can occur. One important aspect in this front is the capability to provide such information in a verifiable manner that, however, does not breach the privacy of the users. To this end, CONNECT employs the concept of Verifiable Credentials and Verifiable Presentations that can be self-issued (by the instantiated CONNECT TEE Guard) and can allow for the runtime appraisal of the trustworthiness level of each actor/component that is part of the overall service graph chain. Such data models allow a Holder to provide the necessary assurance on the ownership of specific attributes. The way the (under standardisation) notion of decentralised identifiers is used in those statements and their (digital wallet) storage, has been clarified.

Further elaborating on this front, Chapter 6 puts forth the detailed definition of the data structures and models that will be employed for capturing such trustworthiness evidence. By starting with an analysis of the type of trust relationships to be considered, fleshing out all intrinsic characteristics of in-vehicle topologies, Vehicle-to-Vehicle communication patterns and Vehicle-to-(MEC and/or Backend Cloud Services) requirements, we converged on the set of attributes that best depict the "*trusted state*" of a device as part of the overall trust assessment process. These attributes will be extracted with the support of built-in trust anchors and will be further obfuscated so as to avoid privacy implications. They will be communicated in a verifiable manner by been encoded through specially crafted Verifiable Credentials and Presentations allowing for the self-issuance of

such security claims with the necessary level of authentication guarantees i.e., been constructed by devices (vehicles and/or ECUs) equipped with a valid secure element (see the CONNECT Trusted Computing Base in D4.1.). Based on all the above, we concluded that the YANG data model is the most appropriate to be used in the context of CONNECT, which has been expanded and enriched in order to fulfil all aforementioned requirements so that it can be applicable to all types of data sharing communications in CONNECT, namely in-vehicle, vehicle-to-vehicle, and vehicle-to-MEC.

Equipped with the outcomes of the two aforementioned fronts, the project may take the first steps to the design and development of the algorithm/function that will enable the CONNECT orchestrator to drive the offloading process. In parallel, the usage of verifiable statements/presentations (realised through decentralized identifiers) enables CONNECT entities (whether vehicles or edge-residing software instances) to confirm their identities and establish trustworthy communications.

Chapter 8

List of Abbreviations

Abbreviation	Translation
5GAA	5G Automotive Association
AIV	Attestation and Integrity Verification
CPU	Central Processing Unit
C-ACC	Cooperative Adaptive Cruise Control
C-ITS	Cooperative ITS
DAA	Direct Anonymous Attestation
EC	European Commission
ECU	Electronic Control Unit
ETSI	European Telecommunications Standards Institute
GNSS	Global Navigation Satellite System
GPU	Graphics Processing Unit
HMAC	Hash-Based Message Authentication Codes
HSM	Hardaware Security Module
ICT	Information and Communications Technology
ITS	Intelligent Transport Systems
KPI	Key Performarce Indcator
MACsec	Media Access Control security
MANO	Management and Orchestration
MEC	Multi-Access Edge Computing
MECsec	Media Access Control Security
ML	Machine Learning
PKI	Public Key Infrastructure
RAM	Random-Access Memory
RO	Resoure Orchestrator
RSU	Road Side Unit
RTT	Round Trip Time
SMTD	Slow Moving Traffic Detection

Abbreviation	Translation
SOME/IP	Scalable service-Oriented MiddlewarE over IP
TAF	Trust Assessment Framework
TC	Trustworthiness Claim
TCG	Trusted Computing Group
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TPM	Trusted Platform Module
VC	Verifiable Credentials
VP	Verifiable Presentation
V2X	Vehicle To Everything

Bibliography

[1] Verifiable Credential Authentication via OpenID Connect, October 2022.

[2] Kubernetes architecture componets. https://kubernetes.io/docs/concepts/overview, 2023.

[3] Kubernetes metrics server. https://github.com/kubernetes-sigs/metrics-server, 2023.

[4] Manzoor Ahmed, Salman Raza, Muhammad Ayzed Mirza, Abdul Aziz, Manzoor Ahmed Khan, Wali Ullah Khan, Jianbo Li, and Zhu Han. A survey on vehicular task offloading: Classification, issues, and challenges. *Journal of King Saud University - Computer and Information Sciences*, 34(7):4135–4162, 2022.

[5] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, pages 111–125, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[6] Gagangeet Singh Aujla, Rajat Chaudhary, Neeraj Kumar, Joel J. P. C. Rodrigues, and Alexey Vinel. Data offloading in 5g-enabled software-defined vehicular networks: A stackelberg-game-based approach. *IEEE Communications Magazine*, 55(8):100–108, 2017.

[7] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 56–72, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[8] Bin Cao, Long Zhang, Yun Li, Daquan Feng, and Wei Cao. Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework. *IEEE Communications Magazine*, 57(3):56–62, 2019.

[9] Xiangshen Chen, Hongzhi Guo, and Jiajia Liu. Efficient and trusted task offloading in vehicular edge computing networks. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 5201–5206, 2022.

[10] Sukjin Choo, Joonwoo Kim, and Sangheon Pack. Optimal task offloading and resource allocation in software-defined vehicular edge computing. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 251–256, 2018.

[11] The CONNECT Consortium. Architectural specification of connect trust assessment framework, operation and interaction. Deliverable D3.1, 2023.

[12] The CONNECT Consortium. Conceptual architecture & customizable tee and attestation models specification. Deliverable D4.1, 2023.

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

[14] The CONNECT Consortium. Virtualization- and edge-based security and trust extensions (first release). Deliverable D4.2, 2024.

[15] Yueyue Dai, Du Xu, Sabita Maharjan, and Yan Zhang. Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3):4377–4387, 2019.

[16] 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.

[17] Jianbo Du, F. Richard Yu, Xiaoli Chu, Jie Feng, and Guangyue Lu. Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization. *IEEE Transactions on Vehicular Technology*, 68(2):1079–1092, 2019.

[18] Jianbo Du, Liqiang Zhao, Jie Feng, and Xiaoli Chu. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Transactions on Communications*, 66(4):1594–1608, 2018.

[19] Jianbo Du, Liqiang Zhao, Jie Feng, and Xiaoli Chu. Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Transactions on Communications*, 66(4):1594–1608, 2018.

[20] Wenhao Fan, Mingyu Hua, Yaoyin Zhang, Yi Su, Xuewei Li, Bihua Tang, Fan Wu, and Yuan'an Liu. Game-based task offloading and resource allocation for vehicular edge computing with edge-edge cooperation. *IEEE Transactions on Vehicular Technology*, 72(6):7857–7870, 2023.

[21] Internet Engineering Task Force (IETF) RATS Working Group. Attestation Results for Secure Interactions, September 2021.

[22] Internet Engineering Task Force (IETF) RATS Working Group. Reference Interaction Models for Remote Attestation Procedures, October 2023.

[23] Bo Gu and Zhenyu Zhou. Task offloading in vehicular mobile edge computing: A matching-theoretic framework. *IEEE Vehicular Technology Magazine*, 14(3):100–106, 2019.

[24] Hongzhi Guo, Jiajia Liu, and Jianfeng Lv. Toward intelligent task offloading at the edge. *IEEE Network*, 34(2):128–134, 2020.

[25] Hongzhi Guo, Jiajia Liu, Ju Ren, and Yanning Zhang. Intelligent task offloading in vehicular edge computing networks. *IEEE Wireless Communications*, 27(4):126–132, 2020.

[26] Hongzhi Guo, Jiajia Liu, and Jie Zhang. Computation offloading for multi-access mobile edge computing in ultra-dense networks. *IEEE Communications Magazine*, 56(8):14–19, 2018.

[27] D. Hardt, A. Parecki, and T. Lodderstedt. The OAuth 2.0 Authorization Framework, July 2023.

[28] Xiaoming He, Haodong Lu, Miao Du, Yingchi Mao, and Kun Wang. Qoe-based task offloading with deep reinforcement learning in edge-enabled internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(4):2252–2261, 2021.

[29] Vu Huy Hoang, Tai Manh Ho, and Long Bao Le. Mobility-aware computation offloading in mec-based vehicular wireless networks. *IEEE Communications Letters*, 24(2):466–469, 2020.

[30] Muhammad Ibrar, Aamir Akbar, Syed Rooh Ullah Jan, Mian Ahmad Jan, Lei Wang, Houbing Song, and Nadir Shah. Artnet: Ai-based resource allocation and task offloading in a reconfigurable internet of vehicular networks. *IEEE Transactions on Network Science and Engineering*, 9(1):67–77, 2022.

[31] Internet Engineering Task Force (IETF). The YANG 1.1 Data Modeling Language, October 2023.

[32] Internet Engineering Task Force (IETF). Trusted Path Routing, August 2023.

[33] Akhirul Islam, Arindam Debnath, Manojit Ghose, and Suchetana Chakraborty. A survey on task offloading in multi-access edge computing. *Journal of Systems Architecture*, 118:102225, 2021.

[34] Mike Jia, Jiannong Cao, and Lei Yang. Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 352–357, 2014.

[35] Yuhan Kang, Haoxin Wang, BaekGyu Kim, Jiang Xie, Xiao-Ping Zhang, and Zhu Han. Time efficient offloading optimization in automotive multi-access edge computing networks using mean-field games. *IEEE Transactions on Vehicular Technology*, 2023.

[36] Wali Ullah Khan, Furqan Jameel, Guftaar Ahmad Sardar Sidhu, Manzoor Ahmed, Xingwang Li, and Riku Jäntti. Multiobjective optimization of uplink noma-enabled vehicle-to-infrastructure communication. *IEEE Access*, 8:84467–84478, 2020.

[37] Michael Kuperberg and Robin Klemens. Integration of self-sovereign identity into conventional software using established IAM protocols: A survey. In *Open Identity Summit 2022, Copenhagen, Denmark, July 7-8, 2022*, volume P-325 of *LNI*, pages 51–62, 2022.

[38] Haijun Liao, Yansong Mu, Zhenyu Zhou, Meng Sun, Zhao Wang, and Chao Pan. Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4051–4063, 2021.

[39] Hai Lin, Sherali Zeadally, Zhihong Chen, Houda Labiod, and Lusheng Wang. A survey on computation offloading modeling for edge computing. *Journal of Network and Computer Applications*, 169:102781, 2020.

[40] Xi Lin, Jianhua Li, Wu Yang, Jun Wu, Zhifeng Zong, and Xiaodong Wang. Vehicle-to-cloudlet: Game-based computation demand response for mobile edge computing through vehicles. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–6, 2019.

[41] Jianhui Liu and Qi Zhang. Offloading schemes in mobile edge computing for ultra-reliable low latency communications. *IEEE Access*, 6:12825–12837, 2018.

[42] Pengju Liu, Junluo Li, and Zhongwei Sun. Matching-based task offloading for vehicular edge computing. *IEEE Access*, 7:27628–27640, 2019.

[43] Yujiong Liu, Shangguang Wang, Jie Huang, and Fangchun Yang. A computation offloading algorithm based on game theory for vehicular edge networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[44] Tobias Locker and Orie Steele. BBS cryptosuite v2023. W3C working draft, W3C, May 2023. https://www.w3.org/TR/vc-di-bbs/#bbs-signature-2023-0.

[45] Homa Maleki, Mehmet Başaran, and Lütfiye Durak-Ata. Reinforcement learning-based decision-making for vehicular edge computing. In *2021 29th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, 2021.

[46] Mohammad Masdari and Hemn Khezri. Efficient offloading schemes using markovian models: a literature review. *Computing*, 102(7):1673–1716, 2020.

[47] Naresh Nayak, Dennis Grewe, and Sebastian Schildt. Automotive container orchestration: Requirements, challenges and open directions. In *2023 IEEE Vehicular Networking Conference (VNC)*, pages 61–64, 2023.

[48] Khoa Nguyen, Steve Drew, Changcheng Huang, and Jiayu Zhou. Parked vehicles task offloading in edge computing. *IEEE Access*, 10:41592–41606, 2022.

[49] Zhaolong Ning, Peiran Dong, Xiaojie Wang, Lei Guo, Joel J. P. C. Rodrigues, Xiangjie Kong, Jun Huang, and Ricky Y. K. Kwok. Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme. *IEEE Transactions on Cognitive Communications and Networking*, 5(4):1060–1072, 2019.

[50] George Papathanail, Ioakeim Fotoglou, Christos Demertzis, Angelos Pentelas, Kyriakos Sgouromitis, Panagiotis Papadimitriou, Dimitrios Spatharakis, Ioannis Dimolitsas, Dimitrios Dechouniotis, and Symeon Papavassiliou. Cosmos: An orchestration framework for smart computation offloading in edge clouds. In *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, 2020.

[51] Guanhua Qiao, Supeng Leng, Ke Zhang, and Yejun He. Collaborative task offloading in vehicular edge multi-access networks. *IEEE Communications Magazine*, 56(8):48–54, 2018.

[52] Salman Raza, Shangguang Wang, Manzoor Ahmed, Muhammad Rizwan Anwar, Muhammad Ayzed Mirza, and Wali Ullah Khan. Task offloading and resource allocation for iov using 5g nr-v2x communication. *IEEE Internet of Things Journal*, 9(13):10397–10410, 2022.

[53] Markus Sabadello, Amy Guy, Drummond Reed, and Manu Sporny. Decentralized identifiers (DIDs) v1.0. W3C recommendation, W3C, July 2022. https://www.w3.org/TR/2022/REC-did-core-20220719/.

[54] Firdose Saeik, Marios Avgeris, Dimitrios Spatharakis, Nina Santi, Dimitrios Dechouniotis, John Violos, Aris Leivadeas, Nikolaos Athanasopoulos, Nathalie Mitton, and Symeon Papavassiliou. Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions. *Computer Networks*, 195:108177, 2021.

[55] Stefano Secci, Patrick Raad, and Pascal Gallard. Linking virtual machine mobility to user mobility. *IEEE Transactions on Network and Service Management*, 13(4):927–940, 2016.

[56] Yaron Sheffer, Dick Hardt, and Michael B. Jones. JSON Web Token Best Current Practices, February 2020.

[57] DMTF Standard. Common Information Model, October 2023.

[58] International Standard. ISO/IEC 30141:2018 Internet of Things (IoT) 0 Reference Architecture, October 2023.

[59] Chuan Sun, Hui Li, Xiuhua Li, Junhao Wen, Qingyu Xiong, Xiaofei Wang, and Victor C. M. Leung. Task offloading for end-edge-cloud orchestrated computing in mobile networks. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2020.

[60] Zemin Sun, Geng Sun, Yanheng Liu, Jian Wang, and Dongpu Cao. Bargain-match: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks. *IEEE Transactions on Mobile Computing*, pages 1–18, 2023.

[61] Oliver Terbu, Orie Steele, Gabe Cohen, Michael Jones, and Manu Sporny. Verifiable credentials data model v2.0. W3C working draft, W3C, July 2023. https://www.w3.org/TR/2023/WD-vc-data-model-2.0-20230730/.

[62] Jun-Bo Wang, Hui Yang, Ming Cheng, Jin-Yuan Wang, Min Lin, and Jiangzhou Wang. Joint optimization of offloading and resources allocation in secure mobile edge computing systems. *IEEE Transactions on Vehicular Technology*, 69(8):8843–8854, 2020.

[63] Yunpeng Wang, Ping Lang, Daxin Tian, Jianshan Zhou, Xuting Duan, Yue Cao, and Dezong Zhao. A game-based computation offloading method in vehicular multiaccess edge computing networks. *IEEE Internet of Things Journal*, 7(6):4987–4996, 2020.

[64] Dexiang Wu, Guohua Shen, Zhiqiu Huang, Yan Cao, and Tianbao Du. A trust-aware task offloading framework in mobile edge computing. *IEEE Access*, 7:150105–150119, 2019.

[65] Dexiang Wu, Guohua Shen, Zhiqiu Huang, Yan Cao, and Tianbao Du. A trust-aware task offloading framework in mobile edge computing. *IEEE Access*, 7:150105–150119, 2019.

[66] K. Yasuda, M. Jones, and T. Lodderstedt. Self-Issued OpenID Provider v2, January 2023.