DIAS

Smart Adaptive Remote Diagnostic Antitampering Systems

EUROPEAN COMMISSION

HORIZON 2020

LC-MG-1-4-2018

Grant agreement ID: 814951

Deliverable No.	D4.3
Deliverable Title	Distributed ledger technology (DLT) and cloud-based methods for the provisioning of certified data
Issue Date	31/12/2021
Dissemination level	Public
Main Authors	Holger Johann, Robert Bosch GmbH Marianne Klein, Robert Bosch GmbH Alexander Rieger, Robert Bosch GmbH Charalampos Savvaidis, CERTH/ITI Athanasios Sersemis, CERTH/ITI Konstantinos Votis, CERTH/ITI
Version	v1.0

DIAS Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814951.

Document log

Version	Description	Distributed for	Assigned to	Date
V0.1	Draft structure of deliverable	Structure review	Core Group	04/11/2021
V0.2	Draft content of deliverable	Content review	Reviewer 1: Gianmarco Baldini (JRC) Reviewer 2: Dermot O'Brien (JRC) Reviewer 3: Barbara Graziano (FEV) Reviewer 4: Pierre-Louis Ragon (ICCT)	09/12/2021 (phase 1) 14/12/2021 (phase 2)
V0.3	Final content of deliverable	GA check	GA members	22/12/2021
V1.0	First final version	-	-	-

Verification and approval of final version

Description	Name	Date
Verification of the "Final content of deliverable (v0.3)" by WP leader	Barbara Graziano	31/12/2021
Check of the "First final version (v1.0)" before uploading by coordinator	Zissis Samaras	31/12/2021

Executive summary

Traditional integrity measures like message authentication codes and signatures rely on secrets (e.g. secret keys or private keys). While this provides adequate security against outside attacks, the integrity guarantees depend on the preservation of secrets, and hence the processes and integrity of the manufacturer or another trusted third party.

In this report, the application of an additional integrity preservation layer based on Distributed Ledger Technology (DLT) is discussed.

The most prominent application of a certain type of DLT, the Blockchains, is the preservation of integrity of digital currency systems. Blockchains are tamper-evident transaction logs synchronized among a globally distributed network of computers. It combines cryptography, distributed computing, and economic incentives to provide novel decentralized public utilities. What Blockchain and DLT have in common is the aim to avoid third parties and safeguard against a single point of failure.

Therefore, this report proposes to have publicly verifiable distributed notary service (later visible as the Periodical Technical Inspection authority (PTI) and the Emission Certificate Authority (ECA) containers) running on cloud infrastructure. These containers will - first of all - be able to manage their own identity by employing self-sovereign identity (SSI) techniques, and to create trust between one another, as well as to be publicly visible (i.e. discoverable by technical means) for vehicles, who then can initiate a contract with them - also based of self-sovereign identity (SSI) protocols.

The “end result” of the proposed solution in this part of the DIAS project is a sharable data-driven digital certificate**.** The issuer will be the “Emission Certificate Authority (ECA)” and its decisions will be based on the “notarized” data provided in the “Periodical Technical Inspection (PTI)” container.

In the course of this work package, the ECA system was enabled to analyse data of the Periodical Technical Inspection authority (PTI) - in order to prove certain characteristics (e.g. staying within NOx thresholds) to an auditor (e.g. authorities) or another third party.

A further requirement was to make the data analysis possible without disclosing raw data (i.e. the history of each individual measurements will not be revealed, especially in order to keep compliant with data privacy regulations).

The current DIAS solution provides:

1. the vehicle registration (ownership) - this is the contract between vehicle and Vehicle Licensing Authority,
2. the consent on data safeguarding - this is the contract between vehicle and PTI (allowing the vehicle to send data to the pre-defined endpoint provided by the PTI),
3. the notary registration - this is the contract between PTI and ECA (allowing the ECA to work with well-defined and restricted lists of entries out of the PTI database),
4. the consent for diagnostics and certification - this is the contract between vehicle and ECA, including the permission to open a peer-to-peer connection for the transport of the certificate to the vehicle.

The vehicle internal security measures are discussed in the DIAS deliverable “D4.2 In-vehicular antitampering security techniques and integration”.

The solution proposed in this report can respond to the needs of end-users (i.e. vehicle owners) for privacy and conformity, as well as the needs for specification from the device manufacturer's side (as OEM would need to enable the proposed stack on the vehicle's CCU). Further, it discusses the requirements of governmental authorities, i.e. to eliminate tampering attempts or at least to discover them as soon as possible.

Contents

List of Abbreviations

Abbreviation	Full term
CCU	Connectivity Control Unit
CO	Carbon monoxide
CoC	Certificate of Conformity
DID	Decentralized Identifier , see [1], [2]
DLT	Distributed Ledger Technology
DoW	Description of Work
ECA	Emission Certificate Authority
ECE	Emission Certification Engine
ECU	Engine Control Unit
EEE	Emission Evaluation Engine
ESSIF	European Self Sovereign Identity Framework
EU ETS	European Emissions Trading System
IoT	Internet of Things
IoV	Internet of Vehicles
ISO	International Organization for Standardization ISO timestamp, see [3]
JSON	JavaScript Object Notation, see [4]
KUKSA	Short name for the Eclipse kuksa open source project [5]
NMHC	Non-methane hydrocarbons
NOx	Nitric oxide / Oxides of nitrogen
PKC	Public Key Certificates
PKI	Public-Key Infrastructure

Abbreviation	Full term
PM	Particulate matter
PTI	Periodical Technical Inspection authority
PUC	Pollution Control Certificate
PUC	Pollution Control Certificate, see [6]
SCR	Selective Catalytic Reduction
SecOC	Secure Onboard Communication
SSI	Self-sovereign identities, see [7]
THC	Total hydrocarbons
TLS	Transport Layer Security
TSCR	Temperature SCR
URI	Uniform Resource Identifier, see [8]
V2X	Vehicle to any X application communication
VDR	Verifiable Data Registry, see [9]
VLA	Vehicle Licensing Authority

Table 1 - List of abbreviations

Definitions

Emission Certificate Authority (ECA)

The ECA is a fictional name for an authority, which is allowed to issue certificates of conformity of NOx emissions in the context of the solution proposed in this DIAS report.

Periodical Technical Inspection authority (PTI)

The PTI is a fictional name for an authority, which is allowed to collect vehicle data (e.g. related to NOx emissions) in the context of the solution proposed in this DIAS report.

Vehicle Licensing Authority (VLA)

The VLA is a fictional name for an authority, which is allowed to register vehicles in the context of the solution proposed in this DIAS report.

V2X

V2X (also known as vehicle-to-everything) refers to the communication of a vehicle to any entity that may be affected by the vehicle (such as an application in the cloud).

TSCR_Good

TSCR is an abbreviation and refers to the temperature of a Selective Catalytic Reduction (SCR) system within a vehicle. TSCR_Good describes a lead operation condition for the SCR system based on certain attributes (such as the TSCR>=220°C), under which emissions data can be reliably compared.

See details in DIAS deliverable “D5.2 Advanced detection system against known tampering”, chapter 3, “Remote emission reporting system (NOx map)“.

List of Figures

List of Tables

1 Introduction

1.1 Background

The G-20 countries account for 90 percent of global vehicle sales, and 17 out of the 20 members have chosen to follow the European regulatory pathway for vehicle emissions control [10]. With EU regulation becoming more stringent, exhaust aftertreatment systems have been improved resulting in significantly reduced emissions.

However, various manipulations of these systems for different motivations have been reported. Funded by the EU Research and Innovation program Horizon 2020, the project DIAS: “Smart Adaptive Remote Diagnostic Anti-tampering Systems” is tackling this problem with a multi-level approach of complementary security measures and diagnostic methods.

Recent DIAS reports are documenting the market assessment and first countermeasures against tampering, developed within the scope of the project [11].

While nowadays the vehicles are required to be presented for technical inspections in certified workshops on a regular basis, in the course of this work package the attention was directed to the possibility to periodically transmit emission data, that is continuously collected by the vehicle and have a permanent record of said data. It describes the potential actors and procedures from harvesting the data, then analysing, and attesting the conformity to predefined thresholds - resulting in digital certificates which also need to be tamper-proof.

Additionally, it shows how employing self-sovereign identities (SSI) based on distributed ledger technology can provide a mechanism to verify a subject's identity (such as a vehicle) and to exchange data in a secure and integrity-preserving way, without the need for a trusted intermediary. This helps to overcome the need for central authentication systems and allows for a tamper-proof vehicle-to-cloud data exchange.

1.2 Purpose of the document

This document describes diagnostic methods to transmit certified data from the vehicle and prevent tampering during the data transmission and downstream data provisioning. It documents the developments of Task 4.3 (Blockchain and cloud-based methods for provision of certified data).

Within Task 4.3, potential attack vectors for the vehicle-to-cloud data transfer have been analysed. Weaknesses have been carved out and countermeasures have been derived in a prototypical way.

In the vehicle market, current emission tests are often periodic and based on spot-checks. For a more holistic inspection of emissions and detection of tampering, it is important to enable cloud-based methods to analyse emission data and monitor tampering safeguards. This can only be achieved by developing a tamper-proof integrity-preserving method to provide emission data to a cloud backend and securely provision the data for further analysis by a third party.

The methods presented in this deliverable are solutions for the specific DIAS requirements. Different solutions are possible and might be implemented depending on the overall operational requirements and system architecture.

1.3 Document structure

Chapter 1 introduces the DIAS project and sets the context for the work packages discussed in this report.

In c*hapter 2* insights on the methodology used to define the concept and propose the architecture of an exemplary DIAS emission certificate solution are provided.

The complete process flow is depicted in c*hapter 3*. The need to employ conventional channels for the transport of vehicle data and additional (on-top) usage of peer-to-peer communication for specific, sensitive parts are discussed.

Chapter 4 describes how the digital emission certificate - as a verifiable credential by means of SSI - would facilitate a vehicle to access a pre-defined Geo-zone, which would require such certification from all vehicles. Further, it discusses the expectation about the maturity of the technology employed and how the exemplary work could be re-utilized for further use cases, where trusted data exchange is required.

Chapter 5 concludes the achievements of the proposed solution.

1.4 Deviations from original Description of Work (DoW)

1.4.1 Description of work related to deliverable as given in DoW

The original title was “Blockchain and cloud-based methods for provision of certified data”.

The title was changed to “Distributed Ledger Technology (DLT) and cloud-based methods for provisioning of certified data” to align with the more generic definition of a Verifiable Data Registry (VDRs) in a Decentralized Identity (DID) context in regards to the underlying technology used for such registry [12].

1.4.2 Time deviations from original DoW

There were no time deviations from the original Description of Work.

1.4.3 Content deviations from original DoW

The development team decided in July 2021 to change the description of the deliverable to Distributed Ledger Technology (DLT) and Self-sovereign identities (SSI) instead of Blockchain technology as described within section 1.4.1 “Description of work related to deliverable as given in DoW“.

This is because blockchains are a certain type of DLT whereas within a DID context, the Verifiable Data Registry (VDR) is defined more broadly. It most commonly uses a blockchain as a public-readable DLT but may as well employ other decentralized registry methods. Using a DLT as a VDR offers tamper-evident transaction logs synchronized among a globally distributed network of computers. It combines cryptography, distributed computing, and economic incentives to provide novel decentralized public utilities. This helps to avoid centralised trusted third parties and safeguards against a single point of failure.

At the same time, the idea of self-sovereign identities (SSI) for the vehicles as well as for the authorities involved in the process was embraced. By employing SSI with the public keys shared on a distributed ledger, the characteristics can be leveraged to keep the knowledge about identities distributed and hard to tamper with.

2 Methodology

2.1 Introduction

This chapter focuses on the methods used at the conceptional level. It gives a rough introduction to the requirements and challenges of securely delivering data from a vehicle to specific cloud services. Furthermore, it also introduces the basics of a Self-Sovereign Identity (SSI) technology, Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), how different SSI counterparts interact as well as how cloud data may be certified.

The implementation is described in chapter 3 “DIAS process flow“, whereas an outlook is outlined in chapter 4 “Outlook“.

2.2 General requirements for provisioning of authentic data from the vehicle

Advancements in the Internet of Things (IoT) space have increased the need for intelligent connected vehicles. This poses new requirements towards the transmission, storage, and provisioning of sensor data from vehicle to cloud, in a way in which the origin and validity of the data/identity can be trusted, without trusted third party intermediaries.

The five Vs of big data provide a framework upon which one may analyse the requirements of V2X data delivery.

Table 2 - The five Vs, see [13]

Dimension	**Relevant DIAS Trustful Data Exchange Concepts**
Volume	Variable selection Edge pre-processing
Variety	Standardization of input signals
Velocity	Edge data aggregation and delivery in chunks
Value	Accessibility needs by third parties (state authorities, technical inspection authorities)
Veracity	Vehicle-to-cloud preservation of data integrity

An ever-increasing number of sensors within the vehicle as well as the progress in analysis of this sensor data increases the data volume. For this, it is crucial to undergo a variable selection process and decide on ways to reduce the volume of data.

To limit the load on the transport and storage layer, it is necessary to have a look at the data pre-processing on the edge (on the vehicle itself) and at the transport channel to the cloud.

In our case, data pre-processing on the vehicle side serves several objectives: categorization, transformation, and aggregation.

- In our case, *categorization* is a means to reduce the volume of the data transmitted (due to using multiple variables and condensing them to an operating condition) as well as filtering out operating conditions, which are not relevant to the NOx emission certification.

- Data *transformation* touches the topic of data variety. In the automotive environment, there is a need for conformity in terms of input signal standardization from competing OEMs. Therefore, a generalized approach is outlined that uses the GENIVI VSS framework in a KuksaVAL server. This approach may be adapted to several other comparable projects which require digital V2X notarization without requiring big efforts.
- Another element of edge data pre-processing is data *aggregation*, which is affected by the data velocity that is required. The ex-post reporting of emissions does not directly affect the vehicular functions in a control-loop manner. Therefore, the aggregation of data into chunks is applicable. The aggregation decreases the amount of packages sent between vehicle and cloud, and thus also the total volume. Furthermore, it provides a framework to overcome periods of temporary offline status of a vehicle.

For vehicle data to provide value in a data-driven automotive ecosystem, it is important that the data may be made accessible to third parties effectively. It is also important, that the data reported by the vehicle is completed with any vehicle meta data required for the subsequent analysis and evaluation.

Different third parties have different requirements towards vehicular data, which may lead to conflicts of interest that our solution aims to address.

Emissions data is only valuable to an inspection authority if its veracity can be verified. Any approach towards vehicle-to-cloud data transmission must therefore be able to attest non-tampering of the vehicle data provided.

2.3 Challenges of data exchange in an automotive use case

2.3.1 Enabling smart contracts between partners using SSI

Definition of smart contract in the context of DIAS

In the context of DIAS, a smart contract will be closed if two parties establish a triple handshake based on SSI credentials. The reason to have a triple handshake is to establish a trusted consensus between the contract partners.

1. As an information provider, the first handshake is to accept a public invitation from a potential partner who provides services based on the data.
2. As an information provider, the second handshake is to provide a credential with all necessary information including identity to the potential partner.
3. As a partner, the third handshake is to initiate a credential which contains all the necessary information needed. This step can be the final consensus step for the established smart contract.

In the DIAS context, the following elements of the above-mentioned smart contracts are identified (Refer to section 3.2 “Steps for initial trust creation“ for the detailed workflow description.):

Table 3 - Smart contract elements

Smart Contract Elements	**To Vehicle**	**To PTI**	**To ECA**
From Vehicle	-	Presents *Licensing Credential*	Presents *Vehicle Data Endpoint Access Credential* (selective disclosure of PTI endpoint and collection)
From PTI	Offers a public invitation Presents PTI credential Issues *Vehicle Data Endpoint Access*	-	Offers a public invitation Presents PTI credential Issues ECA Data Endpoint Access
From ECA	Offers a public invitation Presents *ECA credential* Issues *Emission Certificate*	Offers a public invitation	-

In summary, SSI technology allows the authentication & authorization of third party services (such as the PTI proving its certification towards the vehicle) as well as the proof of identity via Verifiable Credentials (such as the vehicle's license towards the ECA). In our use case, there is no financial dimension in scope. However, due to the adopted generic architecture, future applications could include additional processes like negotiation and payment.

2.3.2 Guaranteeing identity of all participants (initial setup)

All participants strive for independence from the other components. However, at the same time, it is necessary to prove the authenticity and integrity of data for all actors participating in the scenario.

The following Figure 1 illustrates the participants of the ecosystem.

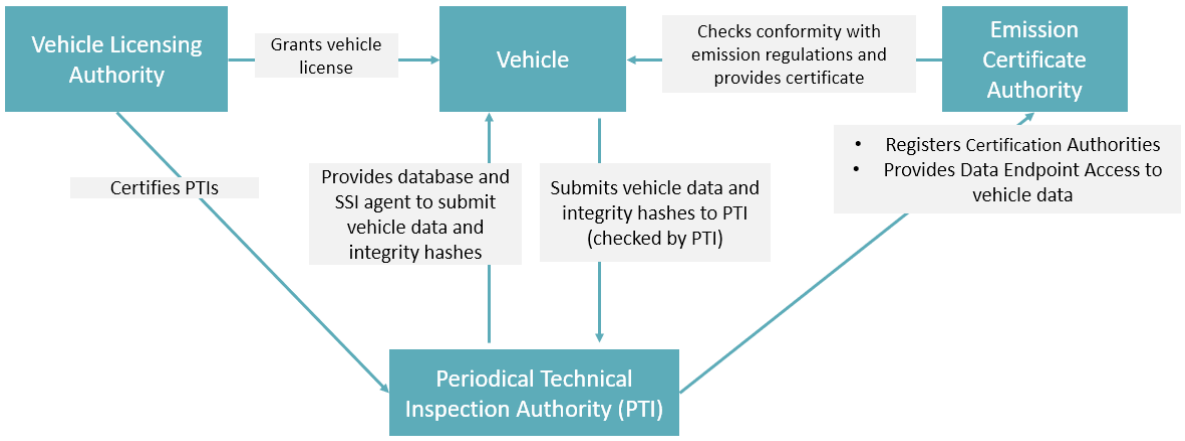


Figure 1 - Identities of the participants

The parties which will need to trust one another are:

- **Periodical technical inspection (PTI) authority**

The vehicle needs to trust that the data it sends is stored safely and not disclosed to In this report, the application of an additional integrity preservation layer based on Distributed Ledger Technology (DLT) is discussed.

The most prominent application of a certain type of DLT, the Blockchains, is the preservation of integrity of digital currency systems. Blockchains are tamper-evident transaction logs synchronized among a globally distributed network of computers. It combines cryptography, distributed computing, and economic incentives to provide novel decentralized public utilities. What Blockchain and DLT have in common is the aim to avoid third parties and safeguard against a single point of failure.

    - without the consent of the vehicle owner.
- **Vehicle license authority (VLA)**
    - The vehicle itself needs to have a trusted identity. As the VLA is the authority which registers the vehicle, it could also be responsible for issuing the digital registration as verifiable credentials.
- **Vehicle**
    - The vehicle needs to identify itself with credentials issued by an authority (VLA). The credential proposal presented to VLA needs to be created by the manufacturer.
- **Emission certificate authority (ECA)**
    - The ECA need to trust the PTI (in the current setup, where the certification business logic runs in a separate container).
    - The ECA needs permission to work on the data stored by the PTI, thus it needs a trustful relation with the other authority.
    - The certificate issued by this instance needs to be provided in a tamper-safe manner to the vehicle. Therefore, SIS credentials are used to issue a certificate to the vehicle.

Self-sovereign identity (SSI) technology will be utilized to eliminate the need for a central authority. Find details on SSI in section 2.4 “SSI technology“ A feasible solution is to employ SSI agents (open-source developed in Hyperledger Indy in conjunction with Hyperledger Aries, see [14] for all participants in the ecosystem, which will take care of managing the proof of their own identity, which is “backed” by another agent in the chain (tree).

To establish trusted data communication between all of these participants, it is needed to once, initially set up the system. The detailed process will be discussed in section 3.1 “General setup of containers and SSI agents“.

These prerequisite steps are necessary to technically inter-link all of the actors: make them known and establish a trusted identity. This is done by creating a credential proposal by the holder and providing this credential proposal to an appropriate issuer. If the issuer accepts the credential proposal, he will issue a signed proposal for the holder. In parallel, the SSI framework will store a signature of the credential into a distributed ledger. A verifier (anyone who will verify the identity of a holder) is able to test the provided credential from the holder against the distributed ledger. Therefore, the issuer will never need to be informed where and when the holder will use the credential.

Holders can share verifiable presentations, which can then be verified without revealing the identity of the verifier to the issuer, see [15].

Besides establishing identity, use case-specific meta information like database links etc. are exchanged as part of the exchanged credentials.

The actors from Figure 1 can act in different roles depicted in Figure 2. The role will change depending on the type of interaction. Example: The PTI is the holder of its own credentials issued by the VLA, is the issuer of credentials to the ECA, and can be a verifier when the vehicle sends a presentation of its vehicle credentials which have been issued by the VLA.

The detailed process will be discussed in section 3.1 “General setup of containers and SSI agents”.

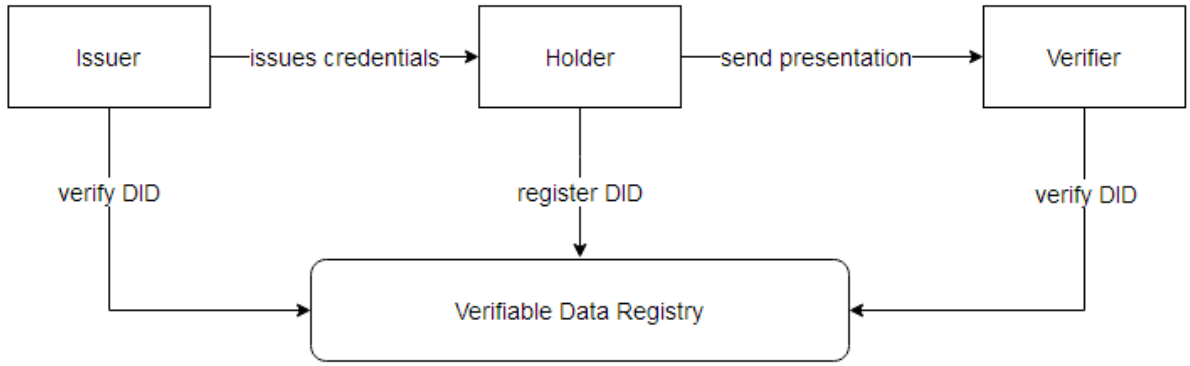


Figure 2 - SSI roles: issuer - holder -verifier

2.3.3 Ensuring data integrity (continuous data exchange)

After establishing an identity for each participant and exchanging meta information as part of the identity process, the same SSI methodology can be utilized for securing data integrity when it comes to data communication.

2.3.3.1 Technical solution

One part of SSI is to provide an identity for any kind of thing (holder) as described above. But there is another side effect establishing an identity exchange. Behind the scenes, the SSI agents establish peer-to-peer communication, based on private/public keys cryptography for further credential exchange using a DID document (see section 2.4.2“Decentralized Identifier (DID)”. This document can be utilized to exchange any kind of information. At this point, this channel is used to exchange data integrity information.

2.3.3.1.1 Payload

The payload is harvested from CAN messages which can appear every 10ms. Based on a dynamic parameter all the received CAN messages are collected into a so-called chunk (e.g. a chunk of 1.000 messages). After the chunk has collected its defined count of CAN messages, a hash is generated based on all CAN messages in the chunk. The hash itself is added to the chunk.

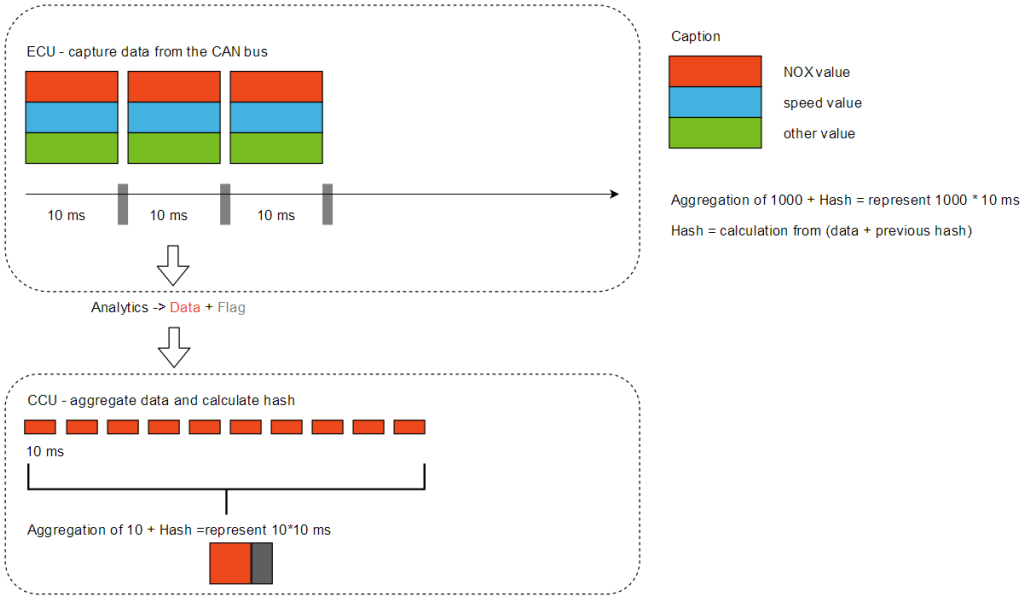


Figure 3 - Aggregation and hashes

2.3.3.1.2 Integrity Hash

At the CCU level, the "raw data snippets" are aggregated into bigger chunks, in order to save northbound traffic. Sending data in 10ms intervals is not feasible for a vehicle.

Before sending a data chunk to a cloud instance a hash is calculated and appended to the data.

Each hash is calculated based on the previously calculated hash, thus a hash chain is used as the most accurate term. This procedure helps to easily identify "altered" data samples.

Find more details at 3.4 "Pre-processing - chunks and hashes".

2.3.3.2 Additionally fulfilled requirements

- **Interoperability**
  - A separate independent SSI channel ensures seamless integration into existing or new data communication protocols. To utilize the solution, a parallel process for data hashing is established before the existing transport protocol come into place. The calculated hash is exchanged by the SSI channel which is technically the established peer-to-peer communication.
- **Independence**
  - This means, the existing payload channel is not touched and is completely separated from the SSI channel.
- **Immutability**
  - Using credentials for the data hashes and in addition, a hash-chain algorithm for creating the hashes ensures immutability.

2.3.4 Certifying emissions data

Air pollution can be considered the concentration of compounds in the atmosphere which will change the composition of the air and may cause disastrous effects on life upon Earth [16] The extensive use of vehicle engines leads to several disadvantages. The most critical is the effect on the environment which could rise to diseases and environmental issues caused by the exhaust emissions. The most

harmful and important emissions a vehicle produces are Carbon Monoxide (CO), Oxides of Nitrogen (NOx), Hydrocarbons (HC) and particulate matter [17]. The limits of the emission values according to the current legislation document EURO 6 emission standard in European Union for passenger vehicles and light commercial vehicles of category N1 class I, are as presented in

Table 4 [18].

Table 4 - Euro 6 emission limits (mg/km) standards for passenger vehicles

	Date	**Mass of carbon monoxide (CO)**	**Mass of non-methane hydrocarbons (NMHC)** mg/km	**Mass of oxides of nitrogen (NOx)** mg/km	**Combined mass of total hydrocarbons and oxides of nitrogen (THC + NOx)** mg/km	**Mass of particulate matter (PM)** mg/km
Diesel	09/2014	500 mg/km	-	80 mg/km	170 mg/km	5 mg/km
Petrol	09/2014	1000 mg/km	68 mg/km	60 mg/km	-	4.5 mg/km

Multiple projects and relative research programs are working on the emission validation problem. For example, the European Emissions Trading System (EU ETS) is an organization to combat climate change and was set up in 2005 and concerns the reduction of greenhouse gas emissions. The authors of [19] propose a digitalized solution based on distributed ledger technology and verifiable credentials in order to enable the validation of authenticity and lifecycle management of emissions.

The current solution described in this document was carried out to certify the vehicles which are compliant with the accepted boundaries in emission values related to the Nitrogen Oxides (NOx) pollutants. Cars, urban transport and taxis are responsible for 72% of NOx emissions in modern cities [20]. From 2019 on, NOx emissions are chargeable to the vehicle owner (this means that the owner of the vehicle has to pay a fee). A “Certificate of Conformity” (CoC) is a document which is provided to the vehicle owner from the vehicle manufacturer or the vehicle seller and includes the NOx emissions for new vehicles [21]. In India - because of the density of population and vehicles - another certificate called “Pollution Control Certificate” (PUC) is issued through authorized testing centers and includes all the information regarding the emission level of a vehicle [6]. This document needs to be updated every 3 months [22]. The current research introduces a well-trusted architecture in order to certify a vehicle with valid emission NOx values. This architecture enables real-time updates to the vehicle owner with notifications inside the application User Interface (UI) and stores the results in a database that can be used by the competent authority to access the emission values and generate a certificate if the NOx values are within acceptable bounds.

2.4 SSI technology

2.4.1 Self-sovereign Identity

Self-sovereign identity (SSI) is a concept of digital identities in which only the user itself should own its identity data without third party involvement. SSI allows the individual identity holders to control their verifiable credentials, without an intermediary or centralized authority while presenting attributes about their personal identity. This is in stark contrast to the classic centralised Public Key Infrastructure

(PKI), in which the intermediary has control over all certificates and the issuer itself must be considered trustworthy by all parties involved in the authentication or authorization process.

The core of SSI is the ability for an individual to create Decentralized Identifiers (DIDs) and functionality to store identity data in the form of Verified Credentials (VCs). SSI allows the attestation of data identity because it encompasses cryptographic proofs to verify a subject's identity.

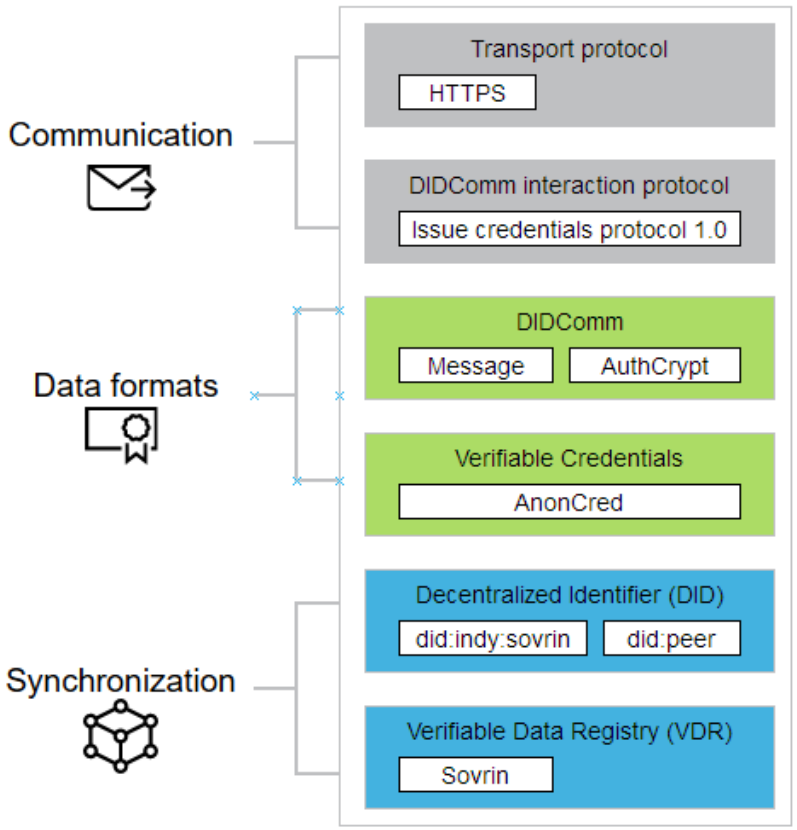


Figure 4 - DID layers

2.4.2 Decentralized Identifier (DID)

Decentralized Identifiers [23] enable a verifiable, decentralized digital identity. In contrast to federated identifiers, DIDs have been designed so that they can be decoupled from centralised registries, identity providers, and certificate authorities. Specifically, while other parties might be used to help enable the discovery of information related to a DID, the design enables the controller of a DID (also known as the holder) to prove control over it, without requiring permission from any other party. DIDs are URIs that identify a subject and that themselves are resolvable to a DID document.

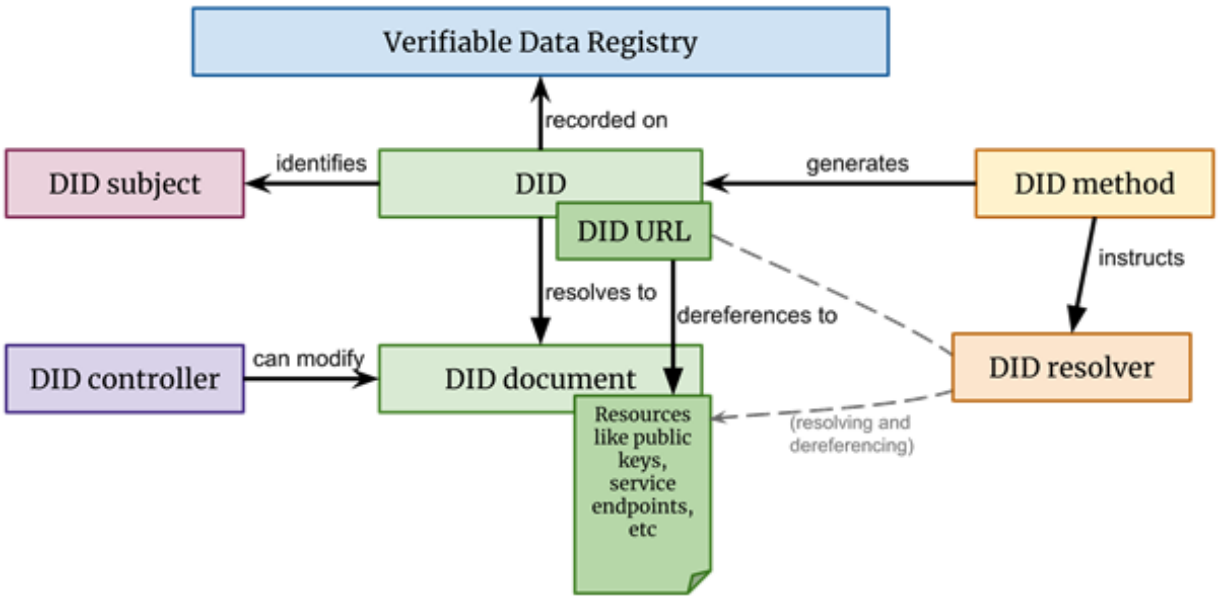


Figure 5 - Verifiable Data Registry

All DIDs have four core properties:

1. Persistent: A DID is bound exclusively and permanently to one and only one subject
2. Resolvable: A DID can be looked up to discover metadata (DID document)
3. Cryptographically verifiable: Claims about a subject can be proven using cryptography
4. Decentralized: No centralized registration authority is required

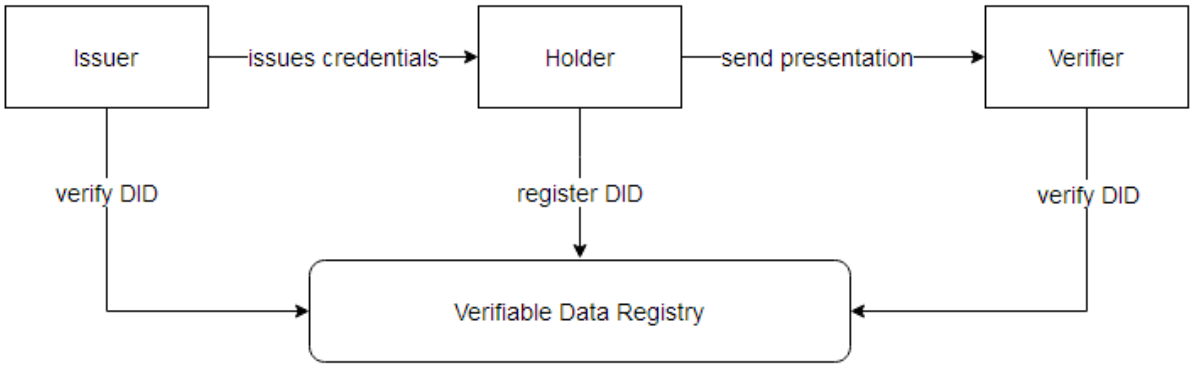
A DID itself is composed of three parts: 1) the scheme "did:", 2) a DID method identifier, 3) unique method-specific identifier generated by the DID method. An example DID as defined by the Sovrin DID method may therefore look like the following: "did:sov:WRfXPg8dantKVubE3HX8pw".

Figure 6 - DID components

A DID method (such as Sovrin, ION or Element DID) defines the associated DID documents and the type of a particular system that facilitates the creation, verification, update, and deactivation of DIDs. This system is called Verifiable Data Registry (VDR) and is in most cases based on a distributed ledger technology, however, it could also be based on a different kind of system.

2.4.3 Verifiable Credentials

Similar to physical credentials, Verifiable Credentials (VCs) can validate information about people, organizations, and things. However, VCs employ cryptography such as digital signatures, which in turn makes them more tamper-evident and more trustworthy than their physical counterparts. The Verifiable Credential standard [24] defines an open ecosystem of credentials that can be easily verified by any interested party.



Repeated Figure 2 - SSI roles: issuer - holder -verifier

Every Verifiable Credential can be specified in a VC schema and created by an issuer. An issuer is an identity-providing entity, as it is able to attest information about a subject to which a credential is issued (the holder**).** Verifiable Credentials are stored in a cryptographically secured wallet (e.g. an application on a mobile phone or on a vehicle). Every VC contains a set of tamper-evident claims and metadata that cryptographically prove who issued it. Holders of VCs can generate verifiable presentations and then share these with verifiers to prove they possess verifiable credentials.

A verifiable presentation is a tamper-evident presentation encoded in such a way that authorship of the data can be trusted after a process of cryptographic verification. Certain types of verifiable presentations might contain data that is synthesized from but do not contain, the original verifiable credentials (for example, zero-knowledge proofs). [25] .

The W3C Verifiable Credential Data Model provides an extensive description of the VC data structure as well as its mapping to JWT and possible proof formats. However, many details are not clearly specified. At the current point of time, most existing implementations use the Indy related “Anoncreds” format.

2.4.4 Data formats and interaction protocols

DIDComm is a standard [26] that defines the secure and authenticated communication channel between DID-controlling entities. By design, DIDComm allows the mutual authentication between any two parties without the use of an intermediary, and if there is a need for an intermediary such as a mediator, this intermediary is unable to access the contents of the DIDComm message.

An example of a DID-controlling entity is an SSI agent. Formally, an agent is software that enables an entity to assume the role of an issuer, holder, or verifier, and to interact with other agents through peer-to-peer communications. Due to the use of pairwise DIDs, the communication between agents is end-to-end encrypted and is able to provide a proof of identity.

The most common framework for an SSI agent is the Hyperledger Aries project, in which DIDComm v1 was born. With DIDComm v2, DIDComm specifications are developed by Decentralized Identity Foundation (DIF) and implemented into the Aries Agents through the Aries Interop Profile specification. The Aries Interop Profile 2.0 includes most of the DIDComm v2 transition. Most frameworks, such as the most established Aries Framework (Aries Cloudagent Python - ACA-Py), currently still have a dependency on Hyperledger Indy and do not offer support for other ledgers.

2.5 Distribution of processing steps between edge and cloud infrastructure

2.5.1 Processing of analytics results on the edge

The advantages and disadvantages of ECU and cloud processing, the incurring architecture decisions regarding the split of edge and cloud processing, as well as emission parameters to be computed on the ECU side are discussed in detail in DIAS deliverable D5.2.

2.5.2 Analysis of processed data within cloud infrastructure

In a modern vehicle, there are more than 200 sensors which can generate around 4000 GB of data per day [27]. The rise of Internet of Things (IoT) and more specifically the Internet of Vehicles (IoV) expand the need for scalable, fast and trustworthy systems that are able to handle this large amount of data coming from different streams and devices. Cloud infrastructure, offers the capability to manage, store and compute large amounts of data through hardware virtualization. Leading enterprises such as Google (GCP), IBM (IBM CLOUD), Microsoft (Azure) and Amazon (AWS) offer pay-as-you-go services that can be utilized to satisfy the requirements of a vast number of projects regarding flexibility, parallel processing and scalability. For example, as is presented in Figure 7 there is a suggested solution in GCP for connected vehicles. This architecture in the depicted image covers the most important requirements for a platform which manage connected vehicles and even though there are alternatives in the market to the GCP solution, it is presented in this document for a better understanding of how it works a typically connected vehicle architecture. The most important requirements that have to be satisfied in a platform which aims to manage vehicle data are the following [28]:

- Device Management (sensors, CCU, ECUs, servers, etc.)
- Continuous processing, continuous data transmission
- Storage
- Analytics and Visualization
- System needs to be extensible and accessible to third parties
- Overall Security

In DIAS project there are different cloud environments which have been configured to satisfy the project needs. For example, one major cloud service that is used is Bosch IoT Insights [29], which receives the transmitted data and make this data accessible to third parties via API requests. A different Cloud Infrastructure can be used to access the data and process it. In DIAS project, a use case would be the certification engine service which has been developed to confirm that the NOx emission values which are created from a vehicle are compliant with the official boundaries for the type of the vehicle. The emission boundaries for passenger vehicles in the EU can be found in

Table 4 in section 2.5.2 “Analysis of processed data within cloud infrastructure”. The certification engine checks specific fields from the data structure and performs threshold checks to examine if the emission values are between acceptable limits. The emission boundaries inside the EU for passenger vehicles are presented in section 2.3.4 “Certifying emissions data” of the current document. If the emission values are acceptable then a certificate can be issued to the vehicle for a specific amount of time that depends on the system characteristics for storage and accessibility for the historical data values. The typical time period in which a certificate can be considered to be valid and not being outdated is three months [20].

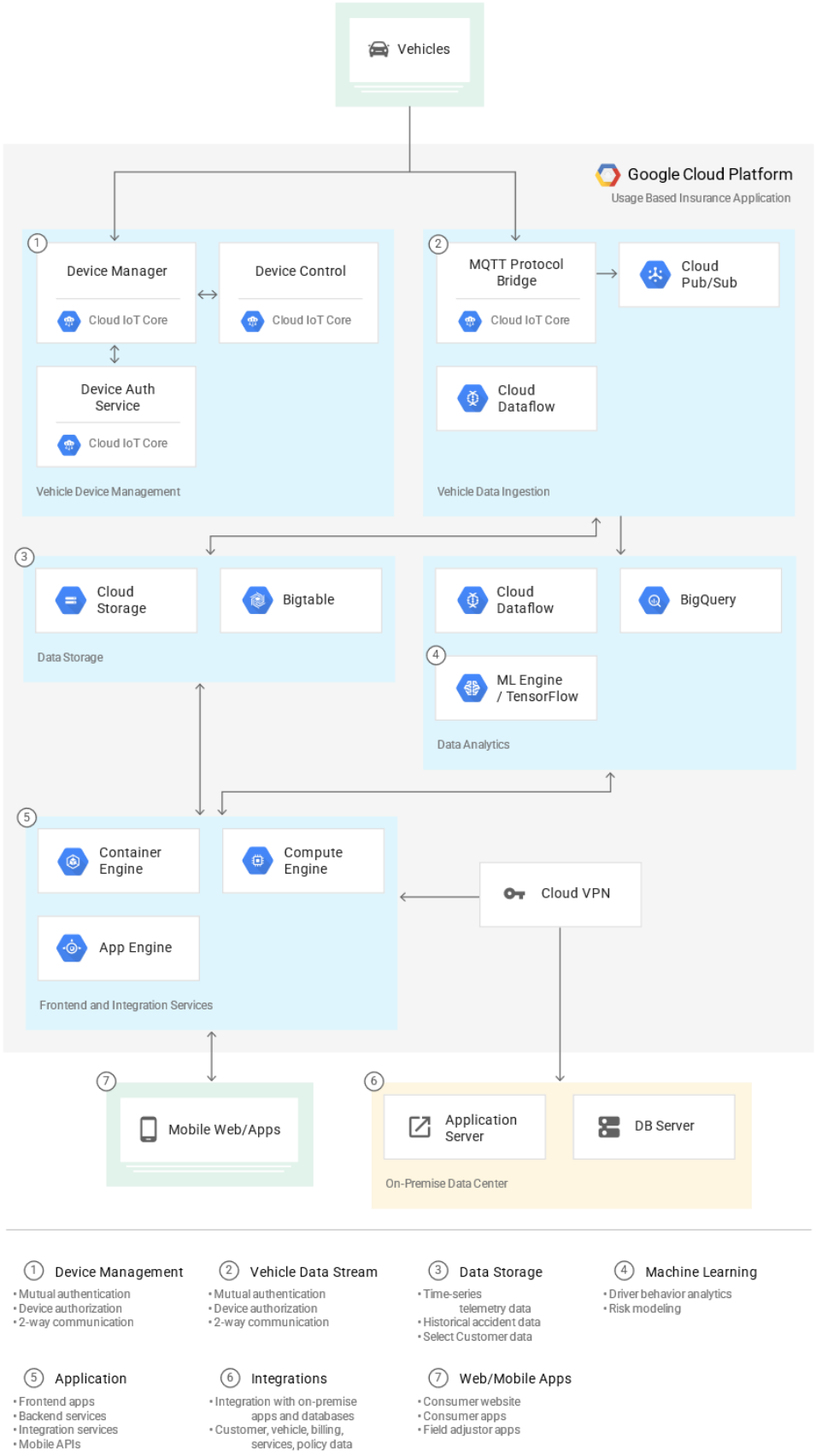


Figure 7 - Designing a Connected Vehicle Platform on Cloud IoT Core [28]

3 DIAS process flow

3.1 General setup of containers and SSI agents

To enable all actors to participate in SSI communication an SSI agent is necessary for all containers. An open-source framework developed by Aries is used.

The following diagram shows how to set up the containers running SSI agents.

1. Vehicle (and Mediator – optional)
2. Vehicle Licensing Authority (VLA)
3. Periodical Technical Inspection (PTI) - who is receiving and storing the raw data as batches and the corresponding hashes
4. Emission Certificate Authority (ECA)

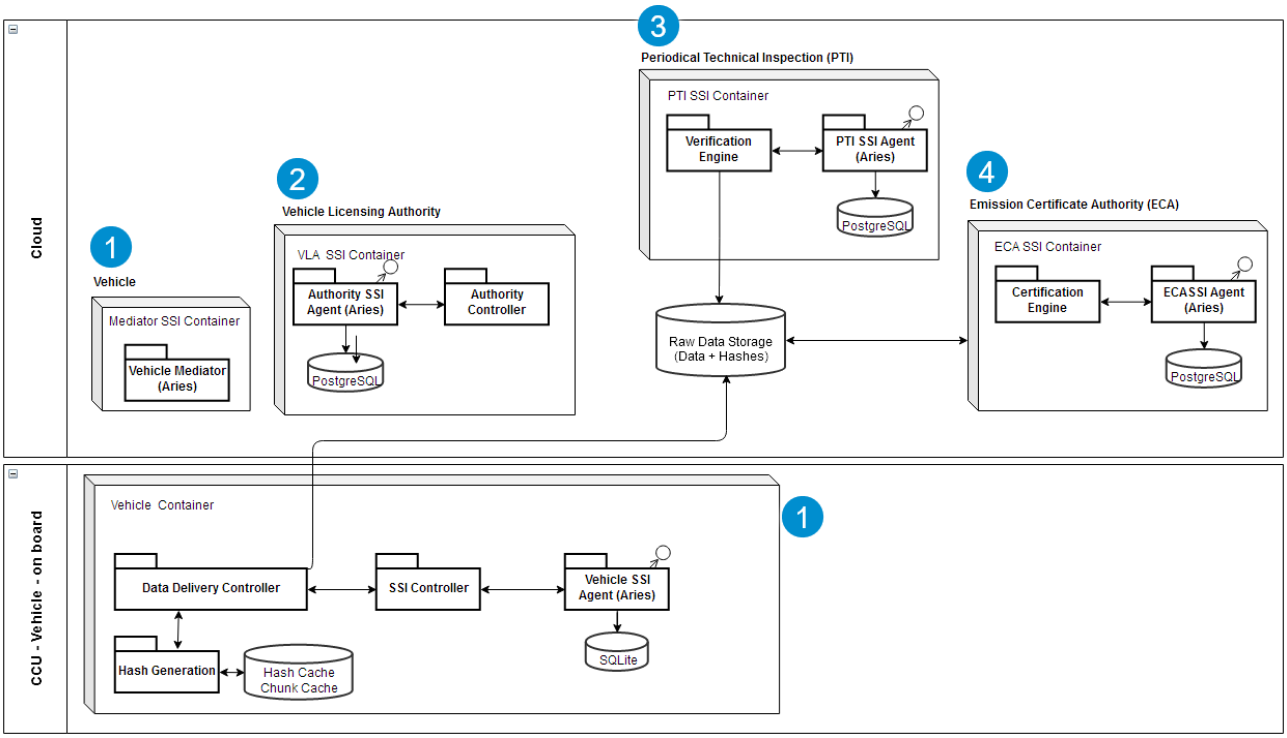


Figure 8 - SSI agents for all container

3.1.1 Vehicle dimension

The vehicle container holds the software which will run in the vehicle itself.

Table 5 - Vehicle container

Name	Vehicle container
Deployment	On the CCU as a Docker container
Components	Data Delivery Controller • Creates and caches hash chain • Delivers the data to the cloud data storage • Instructs SSI Controller to send verification hashes SSI Controller • Interacts with the Vehicle SSI Agent Vehicle SSI Agent (Aries) • Stores credentials (e.g. from licensing authority) and private key in local wallet • Interacts with other SSI agents via DIDComm with the Mediator SSI Container

The mediator container holds the software which will run outside the vehicle in the cloud. Such a mediator is only necessary as a proxy, when the vehicle cannot communicate.

Table 6 - Mediator container

Name	Mediator SSI Container
Deployment	On the cloud as Microsoft Azure container instance
Components	Mediator • Interacts with all other SSI agents to enable always-online-capability

3.1.2 Vehicle Licensing Authority (VLA) dimension

The SSI Container for the Vehicle Licensing Authority can run in the cloud.

Table 7 - VLA container

Name	Vehicle Licensing Authority (VLA)
Deployment	On the cloud as Microsoft Azure container instance
Components	Authority SSI Controller • Interacts with the Authority SSI Agent • Exposes a public invitation to vehicles, to PTI, and to ECA • Controls the issuing of Registration Credentials to vehicles, PTI, and ECA Authority SSI Agent (Aries) Registration • Issues licensing credentials to all respective wallets • Stores all the private keys in local wallet • Creates public invitation for vehicles, PTI and ECA • Interacts with other SSI agents via DIDComm

3.1.3 Periodical Technical Inspection (PTI) dimension

The SSI Container for the PTI can run in the cloud.

Table 8 - PTI container

Name	Periodical Technical Inspection (PTI)
Deployment	On the cloud as Microsoft Azure container instance
Components	Verification Engine • Interacts with the PTI SSI Agent • Receives the integrity hashes from the PTI SSI agent • Verifies the integrity of the raw data storage hash chain using the verification hashes from the PTI SSI agent PTI SSI Agent (Aries) • Stores private key in local wallet • Receives hashes from vehicle SSI Agent (optional via its mediator) • Interacts with other SSI agents via DIDComm
DB	Raw Data Storage • Deployment: Azure Data Storage • (Conventionally) receives and stores the raw data as batches and the corresponding hashes • Is used by the verification engine as a data source

3.1.4 Emission Certificate Authority (ECA) dimension

The Emission Certificate Authority (ECA) is a trusted independent component that is able to perform emission checks on the data that retrieves from Emission data sources and issue Emission Certificates based on them according to Emission Standards.

ECA can be integrated with any trusted source of emission data and can be deployed as an additional service in order to provide:

- Emission data evaluation against the Emission Standard regulation.
- Emission Certificate issuance following Verifiable Credential format.
- Emission Certificate revocation.

Furthermore, ECA is based on trusted emission data sources, thus it is not deal with the integrity and verification of emission data. It can be hosted by PTIs or any other trusted government institution in practical deployments.

ECA main components can be classified into two categories, namely, Core and Agents. The minimum configuration of the ECA components is depicted in Figure 9.

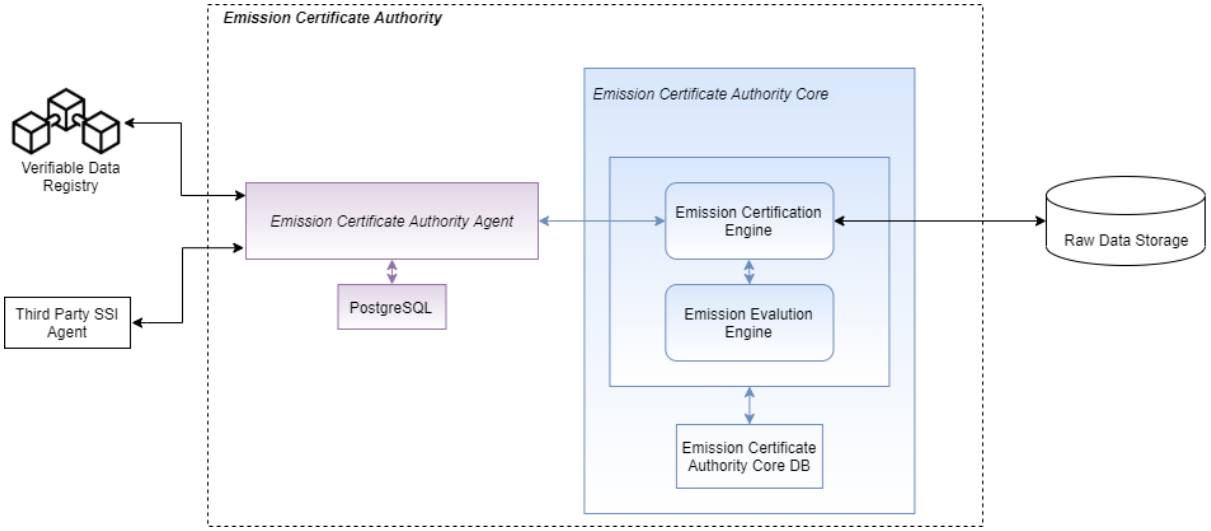


Figure 9 - Simplest configuration of ECA internal architecture

Having Figure 9 as a reference, ECA Core is comprised of two main engines, the Emission Evaluation Engine (EEE) and the Emission Certification Engine (ECE) and a common DB to store data that are shared and used in various operations.

The Emission Evaluation Engine is responsible to perform the business logic of the ECA. More specifically, it interacts with the ECA SSI agent for all related SSI operations, including the issuance of Emission Certificate. It retrieves the emission data from the Raw Data Storage and verifies that the data that are transmitted by the vehicle have been not tampered with. Lastly, it instructs Emission Evaluation Engine to perform emission evaluation checks.

The Emission Evaluation Engine handles the emission data analysis and evaluation.

- Verifies the NOX values for a vehicle for a specific time frame (e.g. 3 months)
- Set of relevant logged diagnostic messages
- Time-series of aggregated emissions data (in chunks)
- Verification flags for each time-series chunk
- Timeframe-based certificate after analysing diagnostic messages and emission thresholds
- business logic could check integrity independently but this is not required

The other main category of the ECA components is its own SSI Agents. These SSI Agents constitutes instances of Hyperledger Aries framework that are linked to a specific ledger and its genesis transactions. ECA SSI Agents handles the SSI Agent logic and are able to manage connections, credentials, presentations, communication and all the crypto material, as described in the Aries Interop Profile 2.0. Thus, they are responsible for the issuance of the Emission Certificate and the interactions with the other third party agents as well as Verifiable Data Registries.

Last but not least, administrators are mandatory for the ECA functionalities as they handle all the configuration and consents regarding the ECA components.

The architecture of the ECA is very extendable, it is built in a way to offer flexibility and scalability and support asynchronous communication. It can be customized to meet individual requirements and specific needs. The ECA can handle many instances of SSI Agents as well as Emission Evaluation and Emission Certification Engines, depending on the use case scenario. It supports one-to-one, many-to-one, one-to-many, and many-to-many relationships between the SSI Agents and its Engines. Therefore, ECA is a verifiable data registry-agnostic system that can connect to different and diverse

verifiable data registry instances. Figure 10 and Figure 11 depict alternative ways of ECA architecture configuration. The communication between ECA Core and ECA SSI Agent is performed via HTTPS and Basic Auth with API Keys.

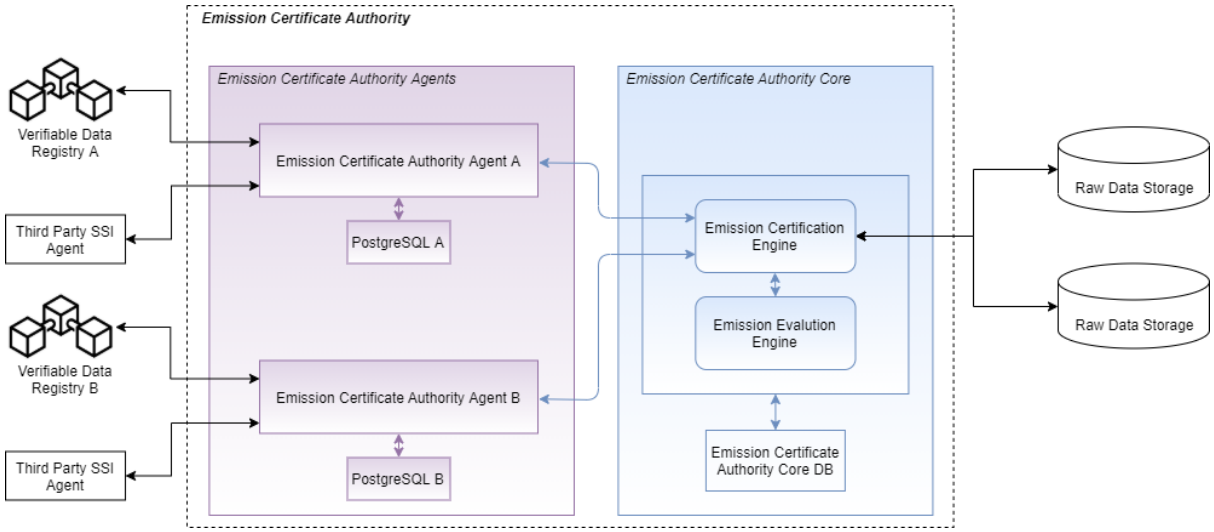


Figure 10 - Alternative configuration option A of ECA internal architecture.

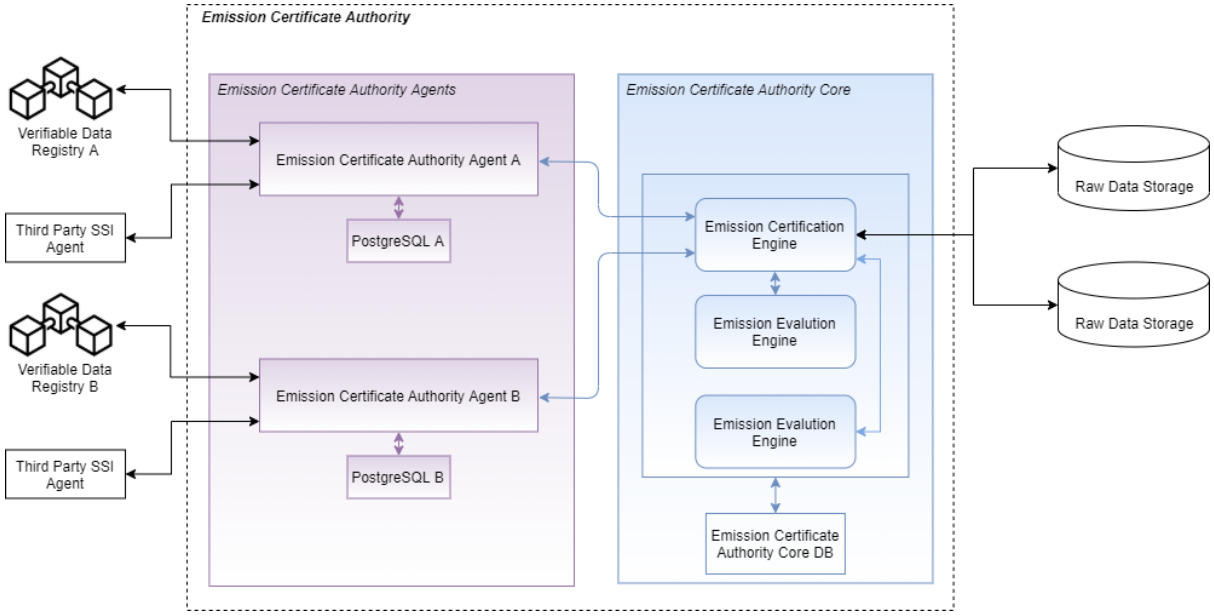


Figure 11 - Alternative configuration option A of ECA internal architecture.

3.1.4.1 Emission Certificate Authority Interfaces

The Emission Certificate Authority provides interfaces that allow users or applications to interact with its services. The interfaces include a user-friendly administrator interface and REST APIs. These APIs have been built with FLASK framework [30] and Python libraries. The API is protected with KEYCLOAK server. KEYCLOAK is an identity and access management platform that offers a variety of security mechanisms and tools such as two-factor authentication, Single Sign On (SSO), OAuth2.0 and Open ID Connect (OIDC). A user has to authenticate his/her credentials with KEYCLOAK server to be able to access the resource API. If the user credentials are valid then a token is generated from KEYCLOAK

server that gives access to the user proper according to the role in which he is assigned to. CLOUDFLARE has been configured to apply additional security measures to the specific API. CLOUDFLARE is a private network that offers security, speed and reliability. CLOUDFLARE includes a variety of security features for example analytics and network data visualization, SSL/TLS protection, DDOS mitigation and SQL injection mitigation.

3.1.4.1.1 Emission Certificate Authority Admin Interface

Emission Certificate Authority Core contains a Graphical User Interface in order to allow administrators to interact with the service. It constitutes a powerful tool that provides the following functionalities:

- Allow administrators to manage connections and credentials.
- Allow administrators to manage emission certificates.
- Allow administrators to manage emission data evaluation.

The Admin Interface contains all the basic features used by the Administrators of the ECA in order to provide valid emission certificates. All these features are mapped with a specific tab in the main menu, as depicted in Figure 12.

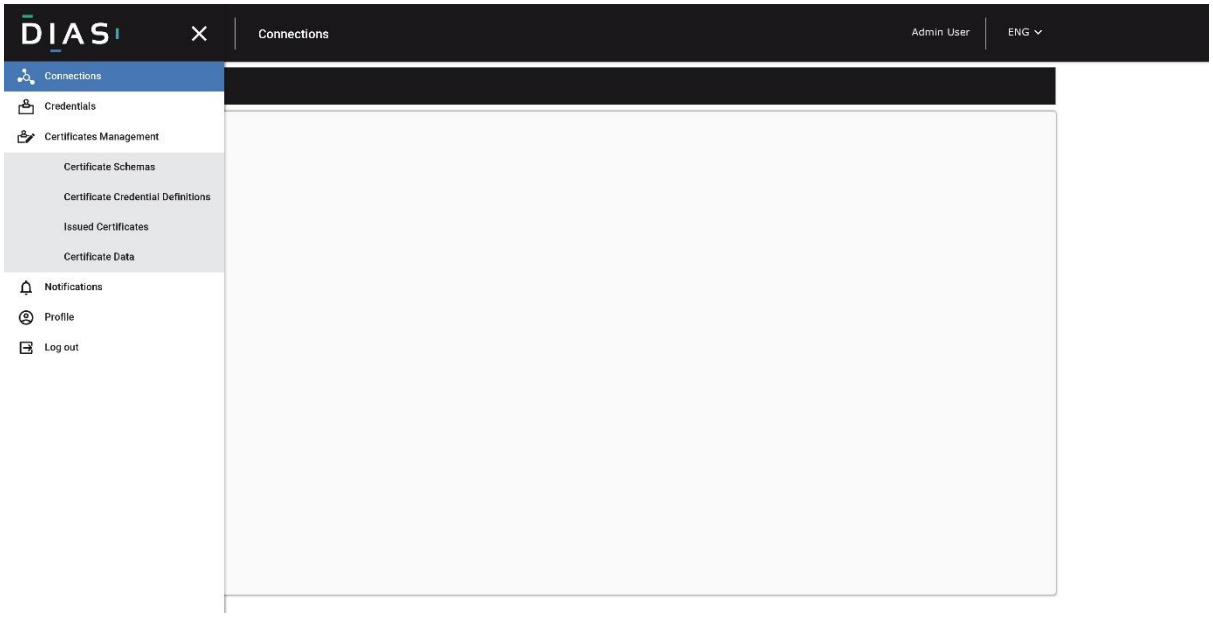


Figure 12 - Main Menu of the ECA Admin Interface

The connection tab focuses on creating private connections to other entities using DIDs. The emission certificates are transferred using these connections.

As shown in Figure 13, for each connection, the interface displays a label for the specific connection along with its state, in accordance with the connection protocol definition of the corresponding Aries RFC. For each connection, the administrators can view additional details by clicking on the appropriate one. On click (Figure 14), a more detailed view will be presented for the corresponding connection. The additional items contain the DIDs of both entities, the timestamp of connection creation and a few additional information that may be used for more advanced operations, i.e., it may contain the presentation of the credentials that have been presented by the connection, as depicted in the "proofs" part. In Figure 15, the paradigm in which the vehicle presents its Data Access Endpoint Credential, as described in section 3.2 "Steps for initial trust creation" is followed. The "Invite Connection" button can be used to create a new private connection. On click, a pop-up with a QR Code or the invitation payload text may appear.

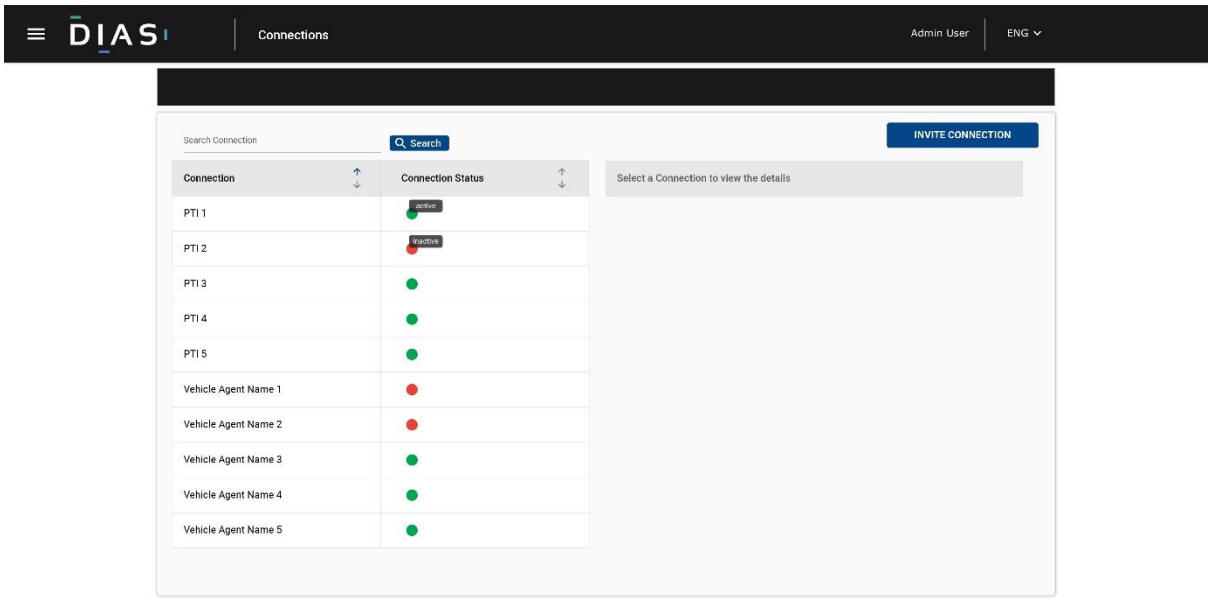


Figure 13 - Connection tab of the ECA Admin Interface

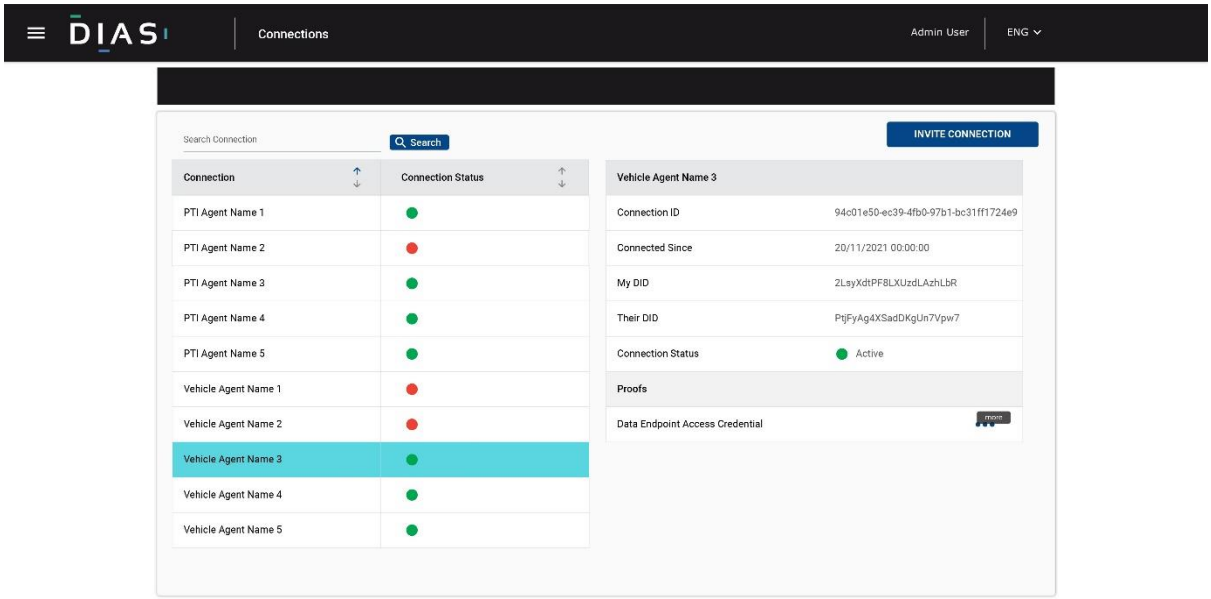


Figure 14 - Connection preview of the ECA Admin Interface

As the main purpose of this ECA Admin interface is to allow ECA to issue credentials to vehicles, the first step needs the definition of a credential schema on the VDR. This can be performed by clicking on the “Create Schema” button on the top right corner of the “Certificate Schemas” tab, as illustrated in Figure 15. On click, a pop-up is presented to the administrators in order to add all the necessary data for the creation of the schema, i.e., name, version number, and a list of attributes. The schema is written on the VDR by clicking the “Create Schema” button, as illustrated in Figure 16. As the schema is created and written in the VDR, the new schema is placed in the list as shown in Figure 15.

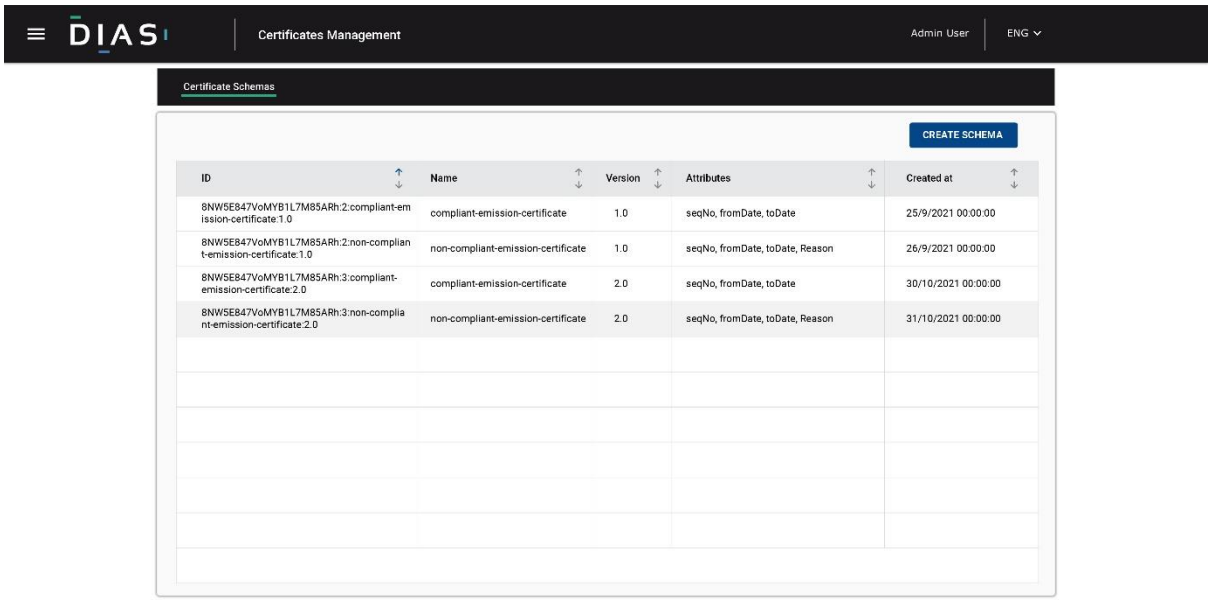


Figure 15 - Certificate Schemas tab of the ECA Admin Interface

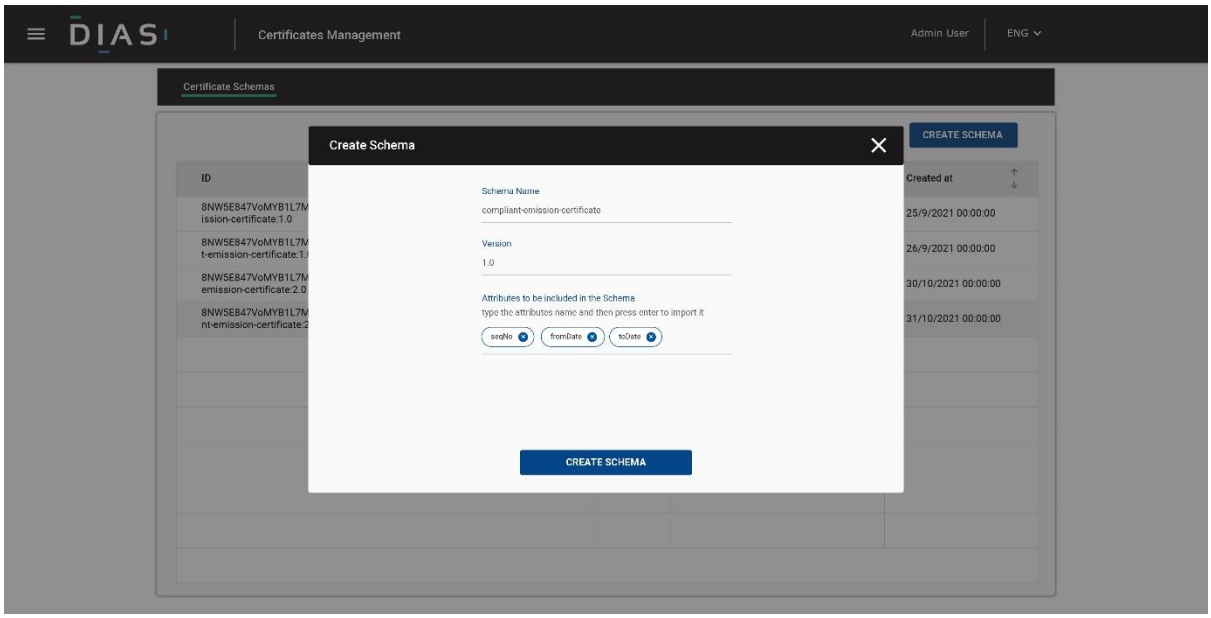


Figure 16 - The form required to create a new credential schema in the ECA Admin Interface

Next, the administrators will write a credential definition in the VDR. Similar to schemas, all the credential definitions are included in the “Certificate Credential Definitions” tab, as depicted in Figure 17. In addition, with the click of the “Create Credential Definition” button, a new pop-up with all the required data is shown (Figure 18). The emission credential schemas and definitions will be used to issue emission certificates.

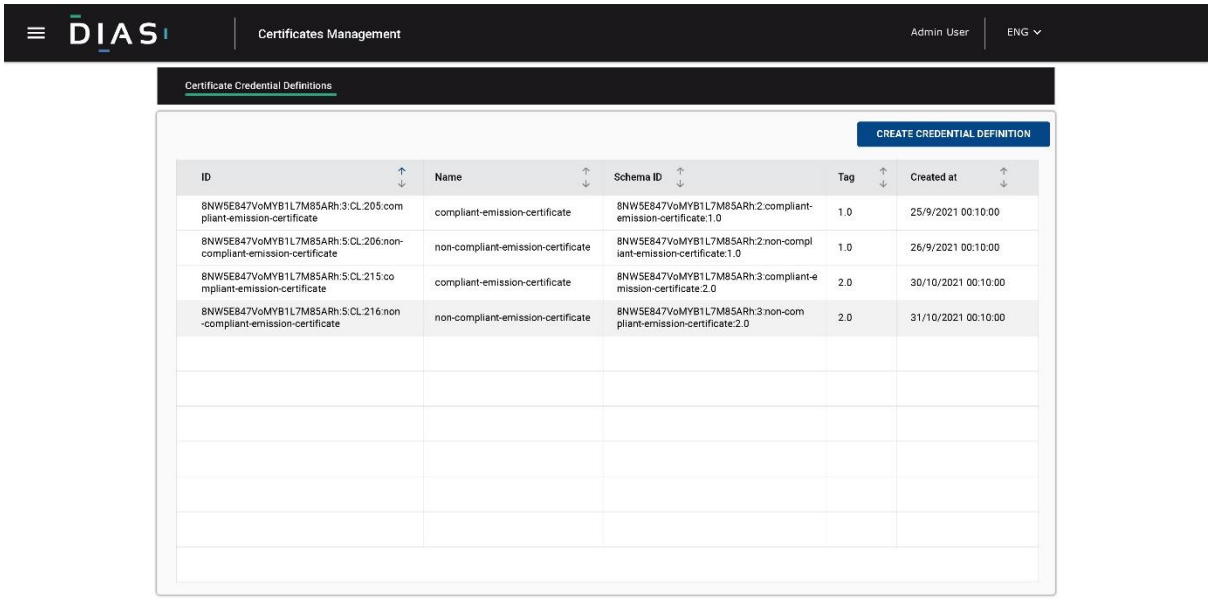


Figure 17 - Certificate Credential Definition tab of the ECA Admin Interface

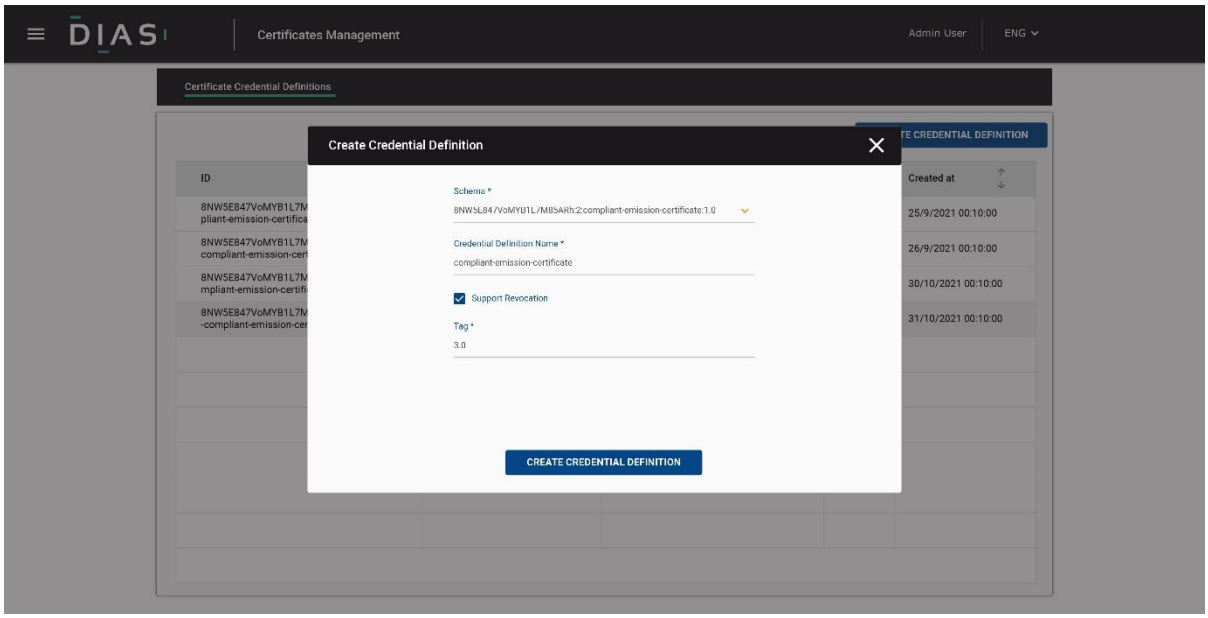


Figure 18 - The form required to create a new credential definition in the ECA Admin Interface

The issued emission certificates from the ECA can be found in the “Issued Certificates” tab, as illustrated in Figure 19. Newly issued certificates are placed to the corresponding table view, which provides preview, search, and filtering functionality as in the previously discussed tabs. Moreover, this view contains a revoke button that allows the revocation of specific certificates. Sample preview of an issued certificate is depicted in Figure 20, which contains full details regarding the chosen certificate.

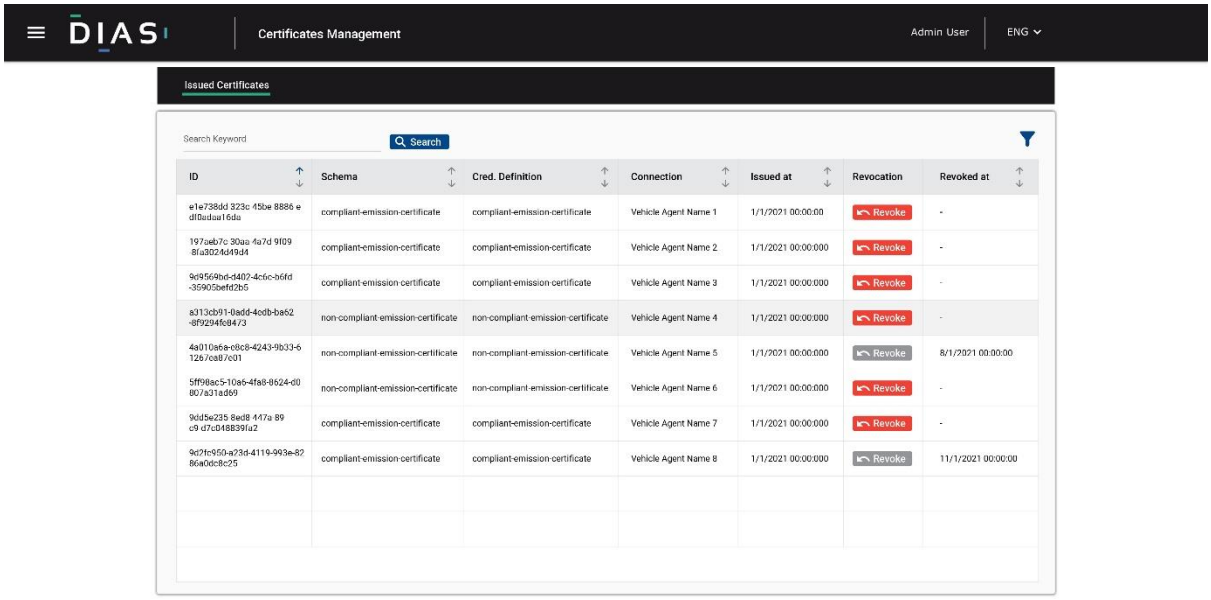


Figure 19 - Issued Certificates tab of the ECA Admin Interface

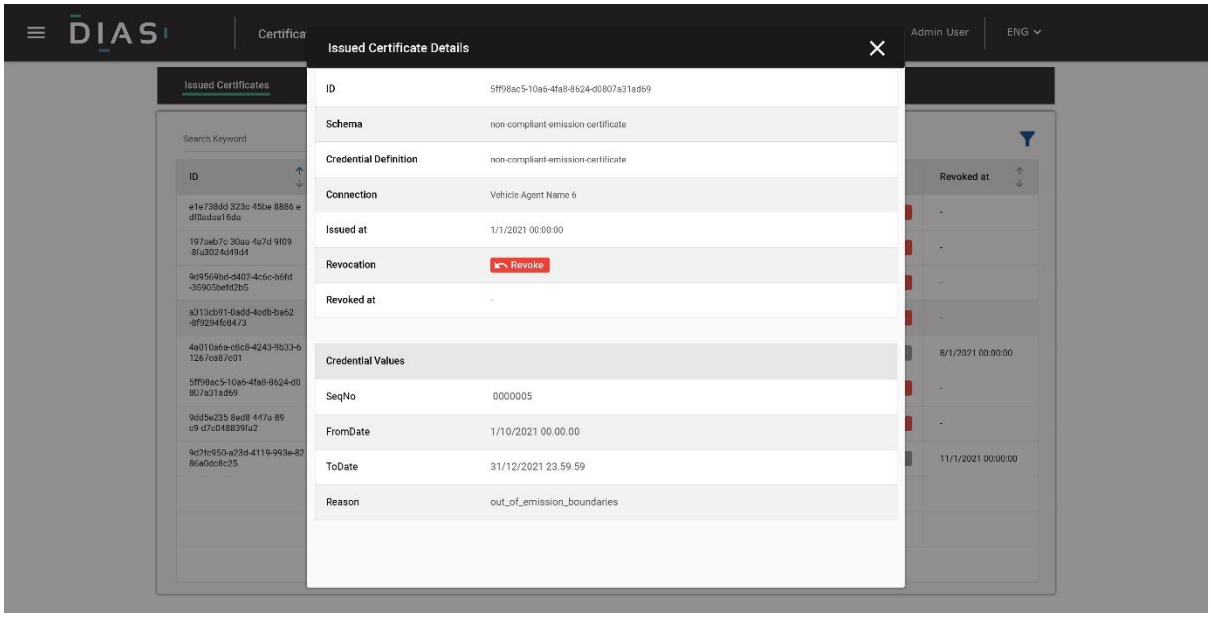


Figure 20 - Detailed preview of issued certificates in the ECA Admin Interface

3.1.4.1.2 Emission Certificate Authority REST API

Table 9 - ECA Core API endpoint

Endpoint	Description	Method	Positive result
/results	The result of the threshold checks	GET	Return the result valid/invalid
/mean_values	Endpoint with critical mean values calculated from the data in the chunks	GET	Return the mean values of all samples
/connectioninvitation	Endpoint to get a new invitation from ECA	GET	Return the invitation payload

Table 10 - Keycloak API endpoints

Endpoint	Description	Method	Positive result
/auth/realms/dias	Issuer	GET	Public Key of the realm, Endpoints for token generation
/auth/realms/dias/protocol/openid-connect/auth	auth_uri	POST	Redirects the user to token URI if authentication is valid
/auth/realms/show_kpis/protocol/openid-connect/token	token_uri	GET	Returns the access token

3.2 Steps for initial trust creation

As already depicted on the schematic view of all containers, the SSI Agents provide a ”public invitation” depicted with the arrow and circle (→O) symbols

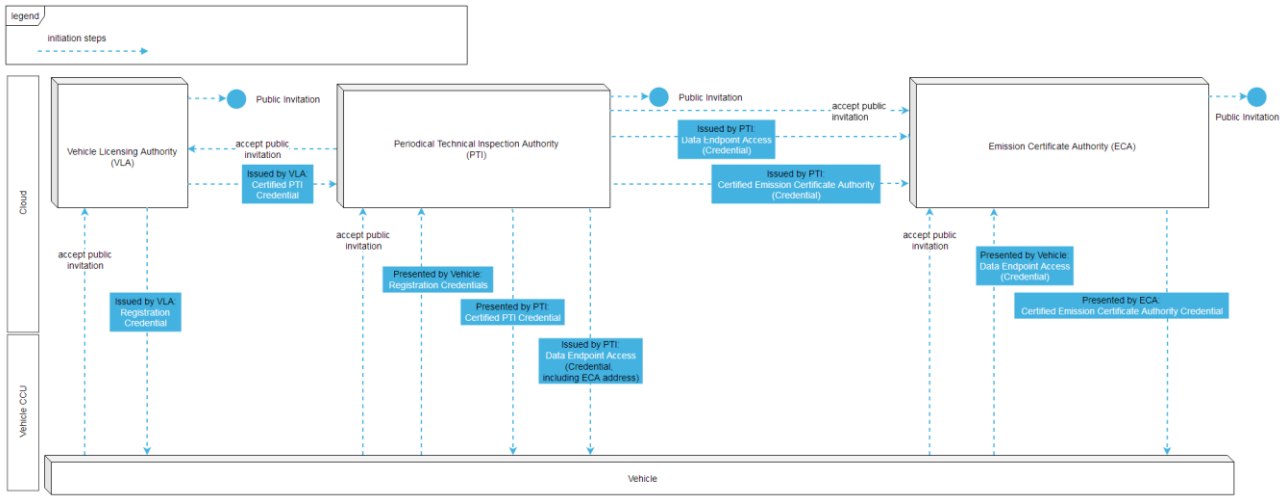


Figure 21 - SSI initial steps

3.2.1 Prerequisites - one-time steps

The following steps need to be proceeded **one-time**, to make all actors know each other.

1. The Vehicle Licensing Authority (VLA) creates a public invitation.
2. The Periodical Technical Inspection (PTI) initiates the relation with VLA.
    a. PTI accepts the public invitation from the VLA, and sends a “connection request”.
    b. VLA accepts the request.
    c. After registering the inspection institution, the ”Certified PTI Credentials” are issued from the VLA to the PTI.
3. The (PTI) initiates the relation with the Emission Certificate Authority (ECA).
    a. Emission Certificate Authority (ECA) creates a public invitation.
    b. PTI accepts the invitation and sends a “connection request”.
    c. After registering, the “Certified Emission Certificate Authority Credentials” are issued towards the ECA.
    d. PTI issues a “Data Endpoint Access Credential” to the ECA, which contains information from where the ECA can receive the Vehicle data.
4. The Vehicle initiates the relation with VLA
    a. The vehicle accepts the public invitation from the VLA, and sends a “connection request”.
    b. VLA accepts the request.
    c. After registering the vehicle, the “Registration Credentials” are issued from the VLA to the Vehicle.
5. The PTI initiates the relation with the Vehicle.
    a. PTI creates a public invitation.
    b. The vehicle accepts the invitation and sends a “connection request”.
    c. PTI accepts the request.

d. As a result, the “SSI connection” is established and used to build trust between both parties.
    i. PTI presents its “Certified PTI Credentials”. Due to the presentation of verifiable credentials, the Vehicle now trusts the PTI.
    ii. The vehicle presents its “Registration Credentials”. Due to the presentation of verifiable credentials, the PTI trusts the Vehicle as the true data emitter.
    iii. PTI issues a “Data Endpoint Access Credential” including the ECA address to the Vehicle.
        1. The credentials contain information where the Vehicle can send its data.
        2. The credential contains information on how to contact the ECA for connection information.
        3. The credential also contains information that the Vehicle can present to ECA.

6. The ECA initiates the relation with the Vehicle.
    a. ECA provides a public invitation.
    b. The vehicle accepts the public invitation from the VLA, and sends a “connection request”.
    c. ECA accepts the request.
    d. As a result, the “SSI connection” is established and used to build trust between both parties.
        i. The vehicle presents only relevant parts of its “Data Endpoint Access Credential” via selective disclosure. The ECA trusts the Vehicle and knows where (in which PTI) to find the data of this Vehicle.
        ii. ECA presents its “Certified Emission Certificate Authority Credentials”. The Vehicle now trusts the ECA and will accept later a certificate issued by this ECA.

3.3 Vehicle internal data flow, processing units & responsibilities (ECU / CCU)

The vehicle internal security measures are discussed in the report of Work Package 4.2 “In-vehicular anti-tampering security techniques and integration”. As the focus in this report is on a generic approach of vehicle-to-cloud transmission of data, attributes that were computed in a pre-processor, such as the tampering indicator value as well as a software and calibration identifier - as proposed in DIAS D5.2 “Advanced detection system against known tampering (Level 2)” are used. A few of these attributes are generated on the ECU side and transmitted on the CAN bus and others are computed on the CCU side in a separate script. DIAS solution then collects these attributes in chunks, computes hashes and transmits both payload and hashes in an independent container on the CCU.

3.4 Pre-processing - chunks and hashes

Sending data in 10ms intervals is not feasible for a vehicle. Therefore, the “raw data snippets” are aggregated into bigger chunks to save northbound traffic.

This happens at the vehicle level - on the CCU.

Before sending a data chunk to a cloud instance (e.g. PTI provider) a hash is calculated and appended to the data.

Each hash is calculated based on the previously calculated hash, thus a hash chain is the most appropriate term. This procedure helps to easily identify “altered” data samples.

3.4.1 Signed integrity hash

Additionally, a *signed integrity hash* will be sent via Basic message (secured SSI-communication) directly to the PTI's SSI agent.

- As this traffic is quite expensive, the trade-off for now is to not send such a message for each data chunk, but recurring, with another frequency, e.g. for each 10th chunk. However, this parameter will be configurable.
- The PTI will be able to verify the signed integrity hash to prove that no third party could interfere in a malicious way of tampering with the data.
    - The verification is done based on the Registration Credential for the authenticity, and by comparing the hash (with was produced as a hash chain) as a second factor.
- The verification result is also persisted in the *Cloud Storage.* (Most probably it will simply flag the respective data chunks as “true” assuring that the source is indeed the *Vehicle* identified in the DID.)

3.5 Transfer of payload to the periodical technical inspection (PTI)

3.5.1 Conventional channels for the exchange of main payload

Vehicle to PTI communication

The main payload is sent in data chunks, which aggregate data extracted from the vehicle's CAN bus. The data chunks will additionally contain a hash (generated at vehicle site - where in each latest hash the chain is calculated based in the previous hash - see details in section 3.4 “Pre-processing - chunks and hashes“. Each data chunk will cover a specific period of time. The vehicle DID will be sent with each chunk, as it will help later in time to filter the data related to the vehicle.

The Vehicle DID and the timestamp will build the *primary key* for accessing the data. Further optimization could be achieved, e.g., by using the DID in the URL/authorization.

The PTI will be able to verify the signed integrity hash to prove that no 3rd party could have interfered in a malicious way of tampering with the data. The verification is done based on the *Registration Credential* for the authenticity, and by comparing the hash (which was produced as a hash chain) as a second factor.

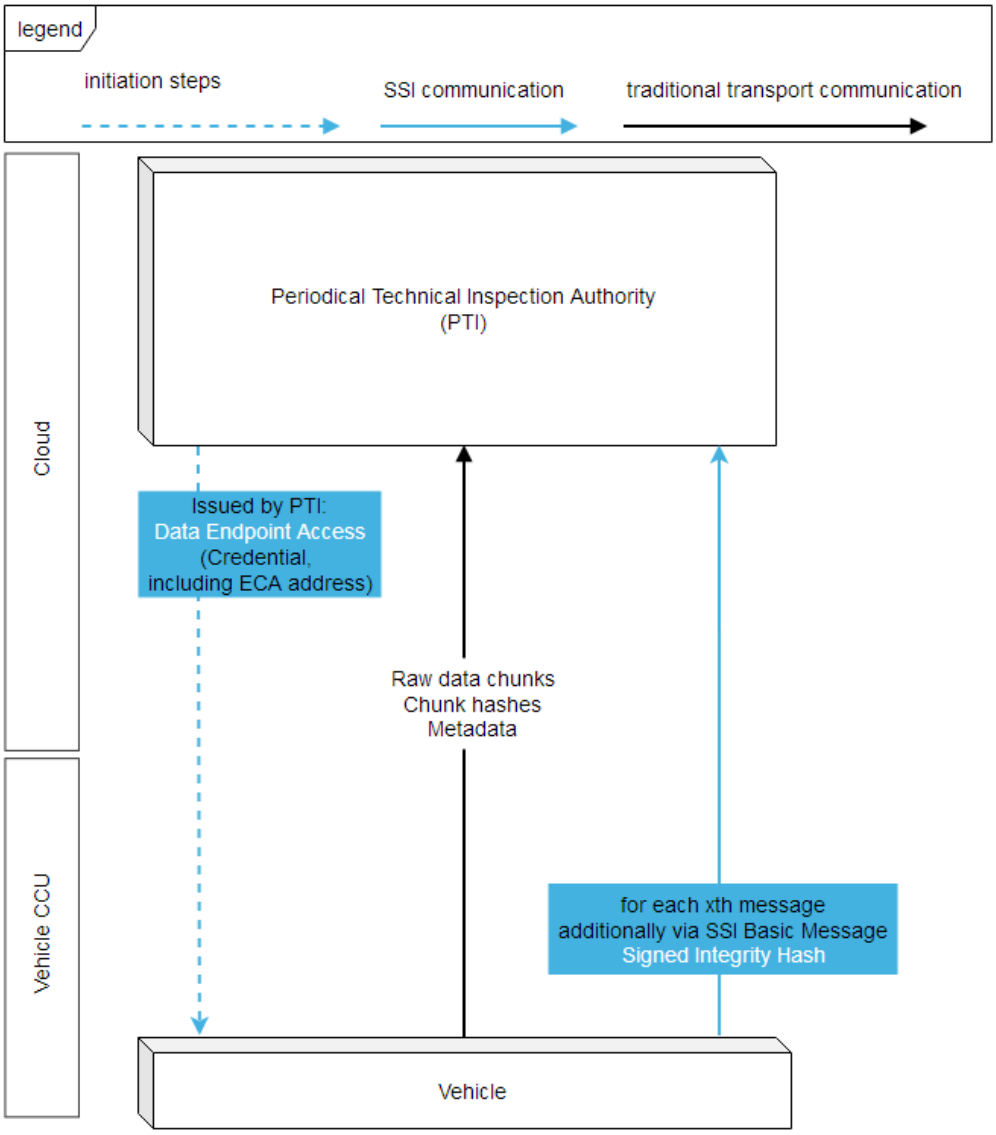


Figure 22 - Vehicle to PTI communication

The verification result is also persisted in the cloud storage, at the data endpoint owned by the PTI. Most probably, it will simply flag the respective data chunks as “true” assuring that the source is indeed the Vehicle identified in the DID.

Note: The implicit signature from the DIDComm Basic message cannot be used by the third party, because it is only verifiable in the peer-to-peer connection.

3.5.2 Safe storage of payload and hash chain

Upon transferring the preprocessed data with integrity hashes from the vehicle to the cloud service, the data is preprocessed and persisted internally in a document-based database.

An example of an entry within this database can be seen in the following code snippet:

Table 11 - Payload example

JSON	**Comments**
`{ did: 'did:sov:123', hash: { value: 'foo', timestamp: '2021-10-08T7:00Z', algorithm: 'SHA-512' }, previousHash: { value: 'bar', timestamp: '2021-10-08T6:00Z', algorithm: 'SHA-512' }, chunk: { hashedData: '[{\"foo\":\"bar\"}]' } }`	`did` - holds as value the public DID of the vehicle `hash` - contains the hash over the previous hash + the new chunks `timestamp` - ISO timestamp of when the message was generated `previousHash` - should be included to close the gap in case there were a few problems when sending chunks to the cloud

It was decided to include the previous hash value as a means of restoring the original hash chain order through easy operations as a fail-safe.

The chunk itself includes the hashed data object.

The hashed data object (which includes the actual attributes) has the following (JSON) structure:

```
{
    "snapshot_metadata":
        {
            "total_sampling": 100,
            "tampering_indicator_value": 77,
            "software_identification_hash": "dummy_hash"
        },
    "tscr_good": {
        "1": {
            "cumulativeNOxDS_g": 0.15,
            "cumulativePower_J": 1200000,
            "samplingTime": 20
        },
        "2": {
            "cumulativeNOxDS_g": 0.04,
            "cumulativePower_J": 295000,
            "samplingTime": 5
        },

        // […]

        "12": {
            "cumulativeNOxDS_g": 0.02,
            "cumulativePower_J": 125000,
            "samplingTime": 2
        }
    }
}
```

Figure 23 - Hashed data object

Note: The hash over the raw data needs to be the same, even when computed in different environments. This is problematic, as the raw JSON data needs to be serialized the same way on each environment, e.g. by using Node.js' JSON.stringify(raw). Depending on the used Node.js version, the outcome will or will not contain whitespace. The same problems will occur when switching to other programming languages. As a straightforward solution, the stringified version of the raw data is sent. This is the exact string on which the hash was computed on the vehicle side. Other users will always get the same hash value when hashing this stringified version.

Each snapshot (which covers a certain time frame) includes metadata relating to the full snapshot, as well as bin data, which relates to one of the twelve bins within the TSCR_Good mapped data:

Table 12 - Hashed data object – caption

Category	Attribute	Type	Description	Usage
Snapshot metadata	Total samples	Integer	Total amount of samples within recorded snapshot	May be used to calculate %-share of TSCR_Good samples compared to total samples
Snapshot metadata	Tampering indicator value	Integer	An integer value between 0 and 255 provided by the ECU / Tampering reporting Module	Threshold check (e.g. value below 100 would pass the check)
Snapshot metadata	Software and calibration identifier	String	A tamper-resilient software identifier provided by the Tampering reporting module	Match check (must match the value provided by the tampering reporting module developer)
Bin-wise TSCR_Good map data	Cumulative NOx Downstream in grams	Float	Total NOx emitted in TSCR_Good state within the specified bin	Used to calculate NOx emissions per unit of power
Bin-wise TSCR_Good map data	Cumulative power in Joule	Float	Total power generated in TSCR_Good state within the specified bin	Used to calculate NOx emissions per unit of power
Bin-wise TSCR_Good map data	Samples within bin	Integer	Total amount of samples recorded within the specified TSCR_Good bin	May be used to calculate %-share of TSCR_Good samples compared to total samples

Alignment with the backend security goals as laid out in section 6.5 of the DIAS deliverable “D4.1 Security analysis, requirements identification and applicability of security solutions for tamper protection” should be ensured. The communication between vehicle and cloud is secured by TLS 1.2 from vehicle to cloud. The database itself is a MongoDB Bosch IT service using a highly secure, cloud-based Lambda infrastructure. Third parties do not directly access MongoDB but use APIs via a secure Bosch load balancer. The ECA side is routed via a REST API call, which is secured through TLS with the cipher suites ECDHE-ECDSA-CHACHA20-POLY1305-SHA256 and AES256-GCM-SHA384.DHE-ECDSA-CHACHA20-POLY1305-SHA256.

3.5.3 DID communication: peer-to-peer transfer of selected integrity hashes

3.5.3.1 Implicit authenticity and re-validation of integrity

Within the workflow of hashing the sensor data, the data delivery controller also employs its in-vehicle Aries agent to sign and submit hashes via DIDComm. This means that each hash sent via the Aries agent is signed with the unique vehicle private key and the resulting message is encrypted. Because the computation of a cryptographic signature (such as a Camenisch-Lysyanskaya) is computationally complex, the amount of generated signatures is limited by only signing and sending periodical "integrity hashes" from the complete hash chain. However, the adoption of the upcoming BBS+ proofs [31], should further decrease the computational load on the vehicle computer.

A trust relationship between vehicle and PTI has already been established upon presentation of the PTI certification credential and issuance of the Data Endpoint Access credential. Therefore, the interaction overhead between vehicle and PTI can be reduced by transmitting the integrity hashes via basic message. An example of the attributes sent from the vehicle to PTI can be seen in the following snippet in Table 13.

```
{
  did: 'did:sov:123', // public DID of the vehicle, already implicit part of basic message
  hash: {
    value: 'foo',
    timestamp: '2021-10-08T7:00Z',
    algorithm: 'SHA-512'
  },
  previousHash: {
    value: 'bar',
    timestamp: '2021-10-08T6:00Z',
    algorithm: 'SHA-512'
  }
}
```

Table 13 - SSI Basic Message from Vehicle to PTI

3.5.3.2 Long term data flags for authentic and valid data

After collection of the payload, its hashes, and the integrity hashes, the PTI is able to perform a completeness and integrity check. Based on the payload data alone, the PTI can sort and recalculate the hash chain through the *previousHash* attribute. The snippet in Table 14 shows a pseudo-code of the approach used in our PTI business logic:

```
entries := [ { /* entry1 */}, { /* entry 2 */ }, {/* ... */} ]
// The entries might not be in order. To find the correct order:
//   Start of a chain is always an entry with entry.payload.previousHash === null.
//   Subsequent entries can be identified by entry.payload.hash.value === subsequentEntry.payload.previousHash.value
// Note entry.payload.previousHash might be null for multiple entries in the response. This will happen every time the car
// is restarted.
// pseudo code for verifying the hash value of an entry
entry := entries[n]
hashAlgorithm := entry.payload.hash.algorithm
hashEncoding := 'hex'
previousHash := entry.payload.previousHash.value if present, otherwise an
empty string ''
hashedData := entry.payload.chunks.hashedData +
entry.payload.previousHash.value
calculatedHash := hash( data: hashedData, algorithm: hashAlgorithm,
encoding: hashEncoding )
isValid := entry.payload.hash.value == calculatedHash
```

Table 14 - PTI business logic - pseudo code

This operation guarantees that the data transfer is complete, but does not include the vehicle's signature. For this, the PTI matches the periodical integrity hashes sent to its SSI Aries Agent with the actual hashes stored in its backend. Upon successful integrity check, the PTI can add another field before storing this JSON object in the database (such as integrity: 'VALID', INVALID, CHUNK_MISSING).

3.6 Provisioning of data to third parties

Looking at the data flow so far, several vehicle-data-based services can be delineated with the following distribution of responsibility:

1. Payload ingress and data storage (PTI)
2. Integrity hash ingress via SSI (PTI)
3. Payload integrity check (PTI)
4. Data usage (ECA)
5. Data validity check and certificate creation (ECA)

It should be noted that this approach is generalistic– therefore, the distribution of responsibilities among these parties could be different for different use cases. One could imagine all services out of one hand with the PTI processing the data, verifying its integrity, and also conducting the emission checks.

Furthermore, the integrity hashes themselves may also be provided to a (semi-)public or private ledger. In such a case any third party who uses the payload data would be able to verify its integrity. However, this opens data privacy questions – the hashes may not relate to the content of the payload, however, they must in a way be linked to a vehicle DID.

For example, if an integrity hash is periodically sent for every 10 chunks of data, storing a bigger amount of integrity hashes on the ledger may indicate higher usage times than vehicles with lower amounts of hashes. Therefore, exposing the integrity hashes to more users would require that at creation time they don't link to a usage pattern. In DIAS proposed case, however, only the PTI has access to the integrity hashes and thus only the PTI can attest the integrity in the stored payload data via an additional attribute. Since the hashes are not exposed publicly, the above-mentioned privacy issues are of no concern for this setup but should be noted for potential extensions or adaptations.

In our case, the ECA requires access to the verified vehicle payload data from the PTI to undertake several checks as part of the emissions certificate. As mentioned in section 3.5.2 “Safe storage of payload and hash chain“, the ECA does not have direct access to the underlying MongoDB database but is granted access to an MongoDB Query Service API [32] using Basic Auth via API key. For this reason, the ECA must first undergo a certification by the PTI, resulting in the exchange of two verifiable credentials: 1) an attestation of the ECA certification (Certified Emission Certification Authority Credential) and 2) a verifiable credential containing the Bosch IoT Insights API credentials to access the data (Data Endpoint Access Credential).

This process allows a third party such as the ECA to query a certification by the data holder, verify its identity and be granted access to the vehicle data in a secure way.

3.7 Data analysis and certification procedures

Although IoV technologies are evolving rapidly, security issues are difficult to address due to resource constraints. These constraints affect the identification, authentication, secure communication and data attestation/certification. In DIAS, the emphasis is on providing an application-agnostic, secure, scalable as well as interoperable solution to resolve these issues. IoV requires cryptography-based

security solutions while accounting for hardware-related constraints such as power usage, storage capacity, computational capacity in order to provide authenticity, confidentiality and integrity of data that are transmitted over digital communication channels. In addition, as IoV inherit the internet protocols, the involved entities have only the knowledge of the addresses of the machines that are communicated with. That implies that there is no information regarding the machine itself or the user or organization that is responsible for that machine. That leads to another main topic of discussion; it is difficult to prove what kind of entity is a machine and what kind of entity is behind the machine and how difficult it is to prove certain characteristics such as sensor and actuators data or diplomas and ages of the entities behind them.

Multiple related models have been proposed and used in production in the last decade. However, due to the differentiation of requirements regarding important domains, such as security, performance, risk management and the involved entities (e.g., the horizontal management policies across all these layers), there is no proper solution that can address all the underlying issues. IDC refers to its research that this is underachievement as at first, is not possible to establish the return on investment (ROI) that is needed to make the transition and second, there is a continuous concern regarding the security issues [33]. In addition, in 2019 cyber-attacks in IoT and IoV systems are raised by 300% [34].

Most of the approaches employ application-level using Public-Key Infrastructure (PKI). PKI enables well-established security protocols, such as Transport Layer Security (TLS) that constitutes the most used standard for securing communications. In addition, most of these implantations are based on the X.509, and other formats of Public Key Certificates (PKC) standard and trust anchors that are utilized by Certificate Authorities (CA). Assuming that the reader is comfortable within a high-level knowledge of PKI and the TLS hierarch, it can be assumed that the context of the PKI, as is employed, introduces many costs and performance issues. In the PKI approaches, a special digital certificate namely the root certificate represents the root of trust. The root certificate must be held by each entity for any certificate verification. Thus, most devices and application comes with a set of built-in root certificates from trusted Certificate Authorities to perform the verification check. These certificates are used by each device in order to enable the root of trust. In addition, the overall maintenance of them is delegated to the device. This could cause problems to million certified entities when the root, e.g. the CA, has been revoked due to data breaches or business reasons. The European Union C-ITS Security Credential Management System (EU CCMS) [35] addresses these challenges.

Furthermore, the trust oversteps cryptographic material, e.g. the proper keys and the ways under which they are exchanged. The trust is based on specific certified information that one knows about the other entity, which is based on putting information in X.509/PKC. This implies full disclosure of any sets of attributes that are included in the certificate without any zero-knowledge proof functionalities.

Also, most of the semantic schemas and web identifiers are vulnerable and limited regarding to their resolvability [36].

On the other side, various approaches are based on the transport layer for authentication and encryption using pre-shared keys or certificates. This comes with several drawbacks regarding security and flexibility, especially in dynamic network conditions.

Today's PKI is built with centralized approaches that contain a single point of failure. Furthermore, most of the private information is stored in centralized databases that constitute honeypots with huge data breaches. Based on the SSI as detailed described in this document, the SSI can provide secure and interoperable identification and authentication, secure communication and privacy-preserving data attestation in the IoV ecosystem. Thus, SSI allows entities to utilize their own root of trust decoupling dependencies, costs, and indirect vulnerabilities of third parties. Moreover, DID

communication and VC exchange implies that data provenance is guaranteed, as Decentralized Identifiers provide globally unique identifiers and verification mechanisms for enabling trusted relationships with secure and private communication channels and Verifiable Credentials provide data models, with the ability of selective disclosure, that are cryptographically verifiable. In addition, these modes are extensible through semantic annotation**.** In addition, DID and VCs are able to enhance operations such as data streams and firmware updates [37].

Concluding, to address the IoV challenges, the DIAS architecture is based on DIDs and VCs to certify emissions data. Issuance of the emission certificate is a multi-step procedure that guarantees proper data analysis, secured transfer, receiving and storing. The term issuance is referring to this whole procedure, from the data evaluation to the creation, transfer and storing of the certificate. This section describes the means under which the components of the ECA employs DIDs and VCs to perform emission data analysis and checks in order to provide a trustful emission certificate. In addition, the advantages of DIDs and DID Communication, as well as VCs, is provided in sections 3.7.3 “Emissions data certification“, 3.7.4 “DID communication: peer-to-peer transfer of certificate to the vehicle“, and 4.1 “Controlling Authority“.

3.7.1 Secure access to emissions data

The NOx Sensors which are inside the vehicle capture the NOx emission values and send it through CAN BUS utilizing Secure Onboard Communication (SecOC) [38] standard to the Engine Control Unit (ECU). The ECU after processing this data, transmits the message to the CCU with SecOC. This means that the data is not only encrypted when transferred in CAN BUS but also supporting integrity since Message Authentication Codes are utilized in the SecOC protocol. In DIAS deliverable “D4.2 In-vehicular antitampering security techniques and integration” they have been demonstrated two different use cases of SecOC. One with symmetric encryption to generate the shared secret key that is required for the SecOC mechanism and a second one in which the shared secret key is generated by utilizing asymmetric encryption. In both algorithms, Elliptic Curve Diffie Helman algorithm (ECDH) [39] was used to generate the shared secret key. Symmetric encryption is faster, easier to implement than the other one with asymmetric, since the devices which are transmitting to the network share the same secret key and it is a reliable solution compliant with the SecOC standard. Asymmetric encryption main advantage is that a pair of public and secret keys can be applied to generate the shared secret key which is required for SecOC. That could merge the structure of DIDs which as it has already been mentioned, consists of public and private parts, or with x.509 certificates which also can be divided into public and private key pairs. That means that a message that utilizes asymmetric encryption is also trusted because each entity has its own private key and the receiver is sure of the sender’s identity. This kind of SecOC may need extra hardware because cryptographic functions demand a high amount of computational power. Additional measures such as Intrusion Detection System, Firewall and Secure logging have been proposed to enhance security even more.

CCU can utilize KUKSA environment to send the messages from the CAN-BUS to DIAS cloud infrastructure. KUKSA maps the CAN messages to json formatted messages and utilizes telemetry protocols such as MQTT to send the data. The MQTT broker needs to fill the requirements for Confidentiality, Availability and Integrity in order for the message to be considered as a trusted message. It is worth to be mentioned that an integrity check is proposed in the current document by including the hash value of the data in the message payload before it would be sent. This hash value can be checked on the cloud side to ensure the message integrity.

During the initialization phase, PTI issues a Data Endpoint Access VC to both vehicle and ECA, which includes all the information to access the endpoint that contains all the necessary data for emission certificate issuance. Moreover, as ECA and vehicle create connection using DIDs, they are exchanged

with them through DIDComm. Thus, ECA is able to ensure that the data that will be evaluated are provided by trusted sources only.

The data which is generated from vehicle sensors and packed in chunks to be sent over the internet is hosted in a cloud platform and can be accessed through an API with HTTP requests. A competent authority could get the data to inspect if the emission values and generate a certificate if the values are compliant with the legislation rules. This means that an independent authority could be authorized to have access to this kind of data.

In addition, at the time that the Data Endpoint Access VC is exchanged between ECA and vehicle, the ECA maps and stores in its internal database the connection between the vehicle and the data endpoint in order to reduce computation time and increase performance each time when an emission certification is going to be issued.

ECA retrieves the emission data endpoint from its internal database according to the vehicle identity and performs an HTTP over TLS request. The response data that contains the emission values of the vehicle will be evaluated and certified. The API request to access the data which is stored in the API BOSCH IOT platform is presented in Figure 24.

```
curl -X POST -k -H 'Content-Type: application/json' -H 'Authorization: Basic
eno2NTI1MDU4MTQ2NTc2LXdhdXBqZC1hcGk6SmctNkdSYlZCZ2RPcGFxVA==' -i 'https://bosch-iot-
insights.com/mongodb-query-service/v2/zz6525058146576/execute-aggregation-query' --
data '{
  "query": [
    {
      "$match": {}
    },
    {
      "$limit": 10
    }
  ],
  "collection": "zz6525058146576_processed_data",
  "tag": "data-analyzer"
}'
```

Figure 24 - API Request curl

This request has been translated to Python programming language:

```python
import requests
import json

url = "https://bosch-iot-insights.com/mongodb-query-service/v2/zz6525058146576/
    execute-aggregation-query"

payload = json.dumps({
  "query": [
    {
      "$match": {}
    },
    {
      "$limit": 10
    }
  ],
  "collection": "zz6525058146576_processed_data",
  "tag": "data-analyzer"
})
headers = {
  'Content-Type': 'application/json',
  'Authorization': 'Basic
    eno2NTI1MDU4MTQ2NTc2LXdhdXBqZC1hcGk6SmctNkdSYlZCZ2RPcGFxVA==',
  'Cookie': 'JSESSIONID=c7d24568-3084-4601-8613-30305359f534;
    __VCAP_ID__=c5a99fc6-94f0-4283-4c5a-966d'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

Figure 25 - API Python Request

The response to this request is described in section 3.5.2 “Safe storage of payload and hash chain“ of the current document. For better understanding, a real case example that includes the response using the API request inside POSTMAN is shown once again:


```
[
    {
        "_id": "61a4cf4177617b2571026bba",
        "payload": {
            "type": "bosch-fake-data",
            "did": "dummy",
            "hash": {
                "value": "3e12135b1343cb8291b486828bf3a048b9ec6bc37c69609d6e527bdcd6e4a9bc1cac95f0577567c162c36fe1f1a522fee96db2e9a18bdea679cd8f9b848f24b4",
                "timestamp": "2021-11-29T13:01:52.995Z",
                "algorithm": "SHA512"
            },
            "previousHash": null,
            "chunks": {
                "hashedData": "[{\"snapshot_metadata\":{\"total_sampling\":1,\"tampering_indicator_value\":28,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.
                    0018267059555555566,\"cumulativePower_J\":0,\"samplingTime\":1}}},{\"snapshot_metadata\":{\"total_sampling\":2,\"tampering_indicator_value\":86,
                    \"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.0053193059555555566,\"cumulativePower_J\":0,\"samplingTime\":2}}},{\"snapshot_metadata\":
                    {\"total_sampling\":3,\"tampering_indicator_value\":81,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.0086268075555555568,
                    \"cumulativePower_J\":0,\"samplingTime\":3}}},{\"snapshot_metadata\":{\"total_sampling\":4,\"tampering_indicator_value\":102,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":
                    {\"1\":{\"cumulativeNOxDS_g\":0.014347482955555566,\"cumulativePower_J\":0,\"samplingTime\":4}}},{\"snapshot_metadata\":{\"total_sampling\":5,\"tampering_indicator_value\":76,
                    \"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.020152390488888897,\"cumulativePower_J\":0,\"samplingTime\":5}}},{\"snapshot_metadata\":
                    {\"total_sampling\":6,\"tampering_indicator_value\":23,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.0257417976000001,
                    \"cumulativePower_J\":0,\"samplingTime\":6}}},{\"snapshot_metadata\":{\"total_sampling\":7,\"tampering_indicator_value\":79,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":
                    {\"1\":{\"cumulativeNOxDS_g\":0.03577570793333345,\"cumulativePower_J\":0,\"samplingTime\":7}}},{\"snapshot_metadata\":{\"total_sampling\":8,\"tampering_indicator_value\":98,
                    \"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.0532286219333333345,\"cumulativePower_J\":0,\"samplingTime\":8}}},{\"snapshot_metadata\":
                    {\"total_sampling\":9,\"tampering_indicator_value\":45,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":{\"1\":{\"cumulativeNOxDS_g\":0.09013374193333334,
                    \"cumulativePower_J\":0,\"samplingTime\":9}}},{\"snapshot_metadata\":{\"total_sampling\":10,\"tampering_indicator_value\":99,\"software_identification_hash\":\"dummy_hash\"},\"tscr_good\":
                    {\"1\":{\"cumulativeNOxDS_g\":0.14886266665333335,\"cumulativePower_J\":0,\"samplingTime\":10}}}]"
            }
        },
        "metaData": {
            "inputDataId": "61a4cf41652055446c944c5e",
            "receivedAt": "2021-11-29T13:01:53.386Z",
            "processedAt": "2021-11-29T13:01:53.396Z",
            "file": "generic"
        }
    },
```


Figure 26 - Response API

3.7.2 Data analysis and conformity check

The data structure, that is retrieved from the Raw Data Storage includes the current hash, the previous hash and the chunks as it is shown in

Table 11 in section 3.5.2 “Safe storage of payload and hash chain“.

Chunk is a group of measured data that comes from the vehicle. Chunks are used for two reasons. First, chunks are hashed with SHA-512 algorithm to generate the current hash value. This use of chunks is to verify the integrity of the data. The idea is that the current hash value which was generated in the one side of the connection and sent together with the message payload has to match the hash value which is generated on the other side of the connection using the chunks. Assuming that the same hashing algorithm, SHA-512 in the current case, and the same message format have been used the two hashes have to match with precision. Second, the raw data which is included in the chunks can be used in order for calculations and analysis to be performed.

Each chunk will contain one or more snapshots with data regarding bins for the TSCR_Good NOx map. TSCR_Good that is based on certain attributes such that the SCR unit in the vehicle is in a normal working condition (e.g., intake gas temperature of the SCR catalyst is warm enough). TSCR_Good has presented in section 3.5.2 “Safe storage of payload and hash chain” of the current document.

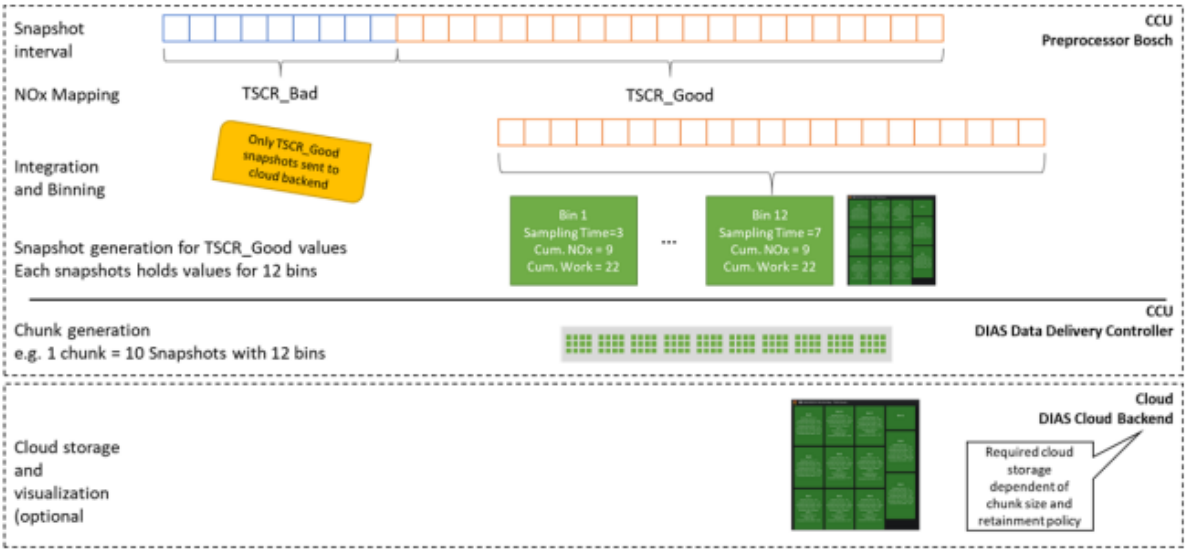


Figure 27 - Visualisation of in-vehicle data generation and delivery flow

Snapshots hold the data from the NOx map TSCR_Good which are necessary for the emission certification. The data structure of snapshots it is presented in Figure 23 in section 3.5.2 “Safe storage of payload and hash chain”.

Snapshot data includes:

1. Total amount of samples covered by the snapshots (e.g., 100)
2. The current tampering indicator value
3. Software and calibration identifier (from the ECU tampering reporting module)
4. Twelve bins of the TSCR_Good NOx map, including the samples per bin, cumulative NOx downstream in grams and the cumulative power in Joule.

During emission validation phase several checks are performed on the incoming data. Most of the checks are simple threshold checks to inspect if the emission values are between acceptable limits according to the legal framework.

Threshold checks are performed during the emission validation phase of the emission values:

- A threshold check of how many total samples fall within the TSCR_Good NOx map
- A threshold check of the tampering indicator value (e.g., a value above 100 should fail an emissions check)
- A threshold check of the NOx emitted per unit of power generated

Besides threshold checks, a match check of the software integration hash should be performed to check the integrity. Also, a match check between the hash that is included in the data that has been received from the vehicle and the newly generated hash of this specific data to the cloud.

The results of the threshold checks are published to an API which was built with FLASK and Python general Libraries. The result can be valid or invalid and it is included in the message alongside useful information from the original message which is useful for the certificate issuance. The response from the specific API endpoint /results is presented in Figure 28.


```
[
    {
        "Result": "valid",
        "SamplingTime": 1,
        "Timestamp": "2021-12-07T12:13:31Z",
        "Did": "dummy",
        "Hash": {
            "value": "3e12135b1343cb8291b486828bf3a048b9ec6bc37c69609d6e527bdcd6e4a9bc1cac95f0577567c162c36fe1f1a522fee96db2e9a18bdea679cd8f9b848f24b4",
            "timestamp": "2021-11-29T13:01:52.995Z",
            "algorithm": "SHA512"
        },
        "PreviousHash": null,
        "CumulativeNOxDS_g": 0.0018257059555555565,
        "cumulativePower_J": 0,
        "Tampering_indicator_value": 28
    },
    {
        "Result": "valid",
        "SamplingTime": 2,
        "Timestamp": "2021-12-07T12:13:31Z",
        "Did": "dummy",
        "Hash": {
            "value": "3e12135b1343cb8291b486828bf3a048b9ec6bc37c69609d6e527bdcd6e4a9bc1cac95f0577567c162c36fe1f1a522fee96db2e9a18bdea679cd8f9b848f24b4",
            "timestamp": "2021-11-29T13:01:52.995Z",
            "algorithm": "SHA512"
        },
        "PreviousHash": null,
        "CumulativeNOxDS_g": 0.005319305955555556,
        "cumulativePower_J": 0,
        "Tampering_indicator_value": 86
    },
```


Figure 28 - API result response

Table 15 - API result

Variable Name	Description	Type	Possible/Example Value
Result	The result of the threshold check for the specific sample	String	Valid/invalid
Sampling Time	The total amount of samples	Integer	25
Did	The DID of the vehicle	String	“dummy_string”
Hash	The hash of the chunk	String	3e12135b1343cb8291…
PreviousHash	The hash of the previous chunk	String	3e12135b1343cb8291…
CumulativeNOxDS_g	A representation of the total emitted NOx emissions within the samples	Float	0.005319305955555556
cumulativePower_J	A representation of the total power generated by the engine within the samples	Float	0.01352532
Tampering_indicator_value	An integer value between 0 and 255 computed from the ECU tampering detection system	Integer	28

Table 16 - Mean Values

Variable Name	Description	Type	Possible/Example Value
CumulativeNOxDS_g_mean	Mean value of CumulativeNOx of all samples	Float	0.0052413405955555556
Did	The DID of the vehicle	String	“dummy_string”
Tampering_indicator_value_mean	Mean value of Tampering_indicator_value of all samples	Float	53.266666666666666
Total Samples	The hash of the previous sample	Integer	25
cumulativePower_J_mean	Mean value of cumulativePower of all samples	Float	0.0132442

3.7.3 Emissions data certification

One of the goals of ECA is to provide the digital equivalent of physical credentials. These credentials are considered tamper-proof and difficult to falsify utilizing portability. They utilize portability and can be verifiable as they provide an authority's attestation in a snip of information about an entity. This entity can provide this information to other entities that are able to verify that the information in the credential is valid, assuming they trust the authority.

The prominent example for IoV is unlimited. For instance, an OEM could be issue credentials for IoT sensor devices, such as its firmware version, serial number, creation date, and many more. Each credential contains a collection of claims about the entity that belongs to. In addition, all these claims must be verifiable in order to be viewed as a proper credential. However, manual verification may be difficult and consume too much time, which is why there is a worldwide black market for counterfeit credentials. It is necessary for a verifier to ensure that the data attestations it receives have not been altered and revoked.

One of the most appealing properties of VCs is they can be digitally validated in milliseconds using powerful cryptography, resulting in higher security guarantees overall than physical credentials. They are also encoded in a machine-readable, standardized format, which eliminates the possibility of human error during the verification process. Summarizing the advantages of VCs are the following:

- They cannot be copied

- They support selective disclosure of zero-knowledge proofs enabling privacy-preservation
- They are prone to attacks
- They are not relying on a centralized system

Hence, as the data evaluation and conformity checks are performed, an emission certificate in a VC format will be issued to the vehicle in order to provide the state of its emission data for a specific time period according to the standards. The result of the evaluation will be encapsulated in the emission certificate and indicates whether the emission values are compliant or non-compliant.

The emission certificate is issued and signed by the ECA using its purpose-specific Public DID and securely transferred to the vehicle. The vehicle from its initialization and appearance on the market, it will obtain a set of sequential emission certificate VCs from every quarter of the year, which can be used throughout its lifecycle. The vehicle can present them to any third party that wants to verify its emission values in order to proceed with further actions.

3.7.3.1 Emission certificate format

The emission certificate format follows the Anonymous Credential format of VC W3C standard. The ECA, and more specifically, its Agents issue two types of emission certificates, namely, the compliant and non-compliant emission certificates, due to the technical characteristics of Anonymous Credentials.

For this reason, the ECA Agents will create and write two different credential schemas and definitions to the Verifiable Data Registry. Credential schemas and definitions contain the attributes of the certificate as well as the required crypto material, e.g., public DID to verify the issuer of the respective certificate. The one represents a compliant certificate and the other the non-compliant certificate. Both certificates have the attributes ‘sequenceNo` which contain the sequence number of the certificate that vehicle receives from its initialization as well as the `fromDate` and `toDate` that contain the specific point of times in which the certificate, and subsequently the emission data, refers to. However, the non-compliant emission certificate schema contains an extra attribute, namely, `reason` that specifies the reasons of non-compliant certificate issuance, e.g., out of boundaries. The contents of the schemas are defined in

Table 17 and Table 18, respectively.

Table 17 - Compliant emission certificate VC schema

Schema Name	Schema Version	Schema Attributes
compliant_emission_certificate	1.0	sequenceNo, fromDate toDate,

Table 18 - Non-compliant emission certificate VC schema

Schema Name	Schema Version	Schema Attributes
non_compliant_emission_certificate	1.0	sequenceNo, fromDate toDate, reason

The selection of the schema only depends on the emission values that are transmitted by the vehicle. If the emission values that are retrieved from the Raw Data Storage contains valid hashes and are between the acceptable levels of emission standards, then the compliant emission certificate schema is selected, otherwise, the non-compliant emission certificate schema will be used for the emission certificate. Table 19 depicts the selection cases of the proper emission certificate schema.

Table 19 - Emission data to emission certificate VC schema

Emission Data	Selected Emission Certificate Schema
Valid Emission Data (valid hashed and acceptable level values)	compliant_emission_certificate
Invalid Emission Data (either invalid hashed either not acceptable level values)	non_compliant_emission_certificate

Furthermore, the values of possible reasons for which the non-compliant schema will be selected are depicted in Table 20.

Table 20 - Reasons of a non-compliant emission certificate VC schema

Reason Name	Description
out_of_emission_boundaries	Valid hash of submitted emission data and not acceptable level
invalid_hashed_data	Invalid hash of submitted emission data, the values will not be evaluated

3.7.3.2 Emission certificate issuance

The time period in which the certificate considers valid can differ and depends on the system requirements and the legal framework. In section 2.3.4 “Certifying emissions data”, it is mentioned that a legal document that proves the emission values and is certified through an independent authority that performs the emission check lasts for three months. The need for real-time emission certification can be achieved with DIAS architecture but the requirements in terms of storage and computational power have to be considered. Therefore, the certification time intervals are set to three months, which means that the emission data will be evaluated and certified every 90 days approximately. Considering the above, as the year is divided into quarters, emission certificates will be provided for a specific quarter of the year.

Subsequently, the emission certificate VC issuance is triggered by ECA every quarter of the year through an automated procedure, e.g., job scheduler. The ECA finds in its internal database the Vehicle DID along with the data endpoint that vehicle writes its emission values. Having this, the ECA performs a request and retrieves the emission data from the specific data endpoint.

The first check of the data that ECA performs is to verify that the submitted values and hashes are valid and have not been manipulated and tampered with. This is done via the checking the ‘isValid’ flag of the retrieved data. This flag defines the next step.

If the flag indicates that the emission values have been manipulated, the non-compliant emission certificate schema is selected. The ECA gets the appropriate timestamps that correspond with the specific quarter of the year and issues a non-compliant emission certificate to the vehicle.

Otherwise, if the flag is valid, the data will be analysed and evaluated as described in section 3.7.2 ”Data analysis and conformity check”. Based on the result of the evaluation, ECA selects the appropriate schema. In both cases, the ECA gets the appropriate timestamps and if the data are between the levels, it issues a compliant emission certificate to the vehicle. In case the data is not between the levels, the ECA issue a non-compliant emission certificate adding the proper reason.

3.7.4 DID communication: peer-to-peer transfer of certificate to the vehicle

The previous sections describe how emission data will be retrieved and evaluated as well as how they specify the content of the emission certificate. Issuance of emission certificate implies at first the creation of it by ECA as well as the secure transfer and storage to vehicles. This section provides details on how this can be achieved using VCs and DIDComm.

Continuing with the analogy of the physical world, the credential is stored in "secured" physical wallet/bag that the holder is able to carry with. The same happens with the DIDs and VCs, they are stored in a secure digital wallet that provides confidentiality, portability and availability.

The digital wallets can be viewed as a software that is managed by SSI Agents. The design principles of digital wallets and agents are:

- Portable by default
- Privacy by design
- Security by design
- Driven by consent

Figure 29 shows the relationship between agents and wallets in the SSI ecosystem.

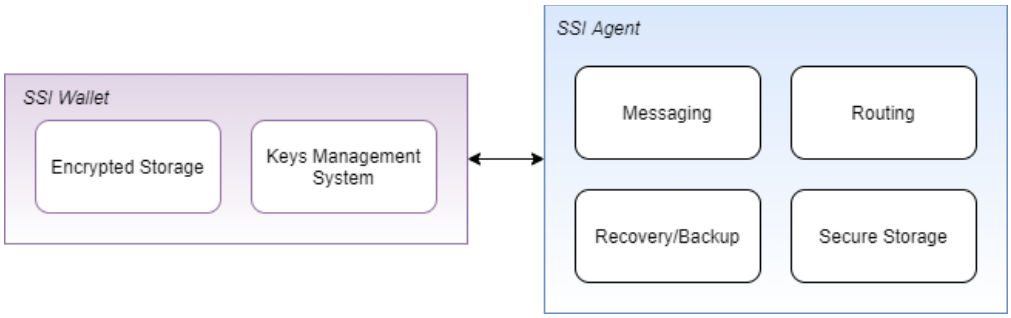


Figure 29 - Relationship between SSI Agents and Wallets

Emission certificate issuance procedure utilizes the Aries Agents and the corresponding PostgreSQL wallet implementation of ECA and Vehicle. It demands a series of messaging exchanges between these entities. The ECA and the vehicle communicate through a peer-to-peer connection that is managed by their SSI agents. This peer-to-peer connection is completely decentralized and utilizes end-to-end encryption. As they establish a trust relationship and a secured, isolated and private communication channel using DIDs, all the messages will be exchanged using this channel. Thus, the emission certificate VC is transferred using DIDComm. In addition, the messaging exchange between the ECA and the vehicle could rely entirely upon DIDComm, i.e., there is no reliance on, e.g., TLS.

The DIDComm or DID Communication is a transport-agnostic, oriented by messages and interaction-based (among peers, e.g. agents,) communication protocol using DIDs. As described in section 2.4 "SSI technology", DIDs are unique global identifiers bound to a DID Document. DID Documents are machine-readable documents that include only the necessary data to enable a secured and trusted connection through DIDComm. This data includes:

- Public keys that will be used to authenticate the entity that the DID belongs to, i.e. the DID subject.
- Service endpoints that the DID subject supports.
- Metadata such as timestamps and cryptographic proofs.

The largest problem of PKI and essentially in IoV PKI is the secure mapping of the entity that controls the public key. For example, when an entity gets the public key of entity X and verifies it

cryptographically, it will never know that the entity Y is the proper entity that wants to communicate with. Thus, it is crucial for the entities to be aware of the proper public keys. The most used solution for this challenge is the TLS PKI with X.509 certificates signed by Trusted Third parties. However, this approach comes with many limitations as stated previously. DIDs are generated based on a set of public/private keys using asymmetric cryptography and are placed in the DID Documents. The DID subject signs the DID Document using its own private key, hence, an entity is able to cryptographically verify the proper owner of the DID Document. Thus, DIDs separate identity verification from public-key verification. The DIDs are able to enable both self-certifying roots of trust as their generation manage by the entity that belongs to, and root of trust with a Verifiable Data Registry. All these steps can be done automatically by SSI Agents. Furthermore, DIDs supports discovery utilities as the DID Documents contains service endpoint URLs, i.e. the information to communicate with the entity that owns the DID. In addition, as the verification is based on the DIDs and their corresponding DID Documents, VCs focus to provide certified data of the DID subject from other trusted entities. This introduces cost minimization and increases in scalability.

DIDs allow you to establish secure DID to DID connections. These connections are consist of pairwise unique DIDs that enables communication channels. The DID to DID communication channels are:

- Permanent until entities decide to
- Encrypted and digitally signed by the private key of DID subjects
- End-to-end
- Trusted for any other information exchange
- Application-agnostic

DIDComm protocol describes the utilization of DID to create DID communication channels and supports a set of protocols for the connection creation, provision of credentials as well as proofs and so on.

The ECA and the Vehicle SSI Agents have established a DID connection in the initialization phase. Using the DID Documents that contains all the appropriate information they create a private peer-to-peer communication channel. Having this channel, there are able to share signed private data that want to disclose. Both of them presents the VCs that have been issued from the PTI, hence, both of them trust each other and are able to exchange sensitive data.

The issuance requires the ECA Agent, that acts as an issuer, to write the corresponding credential schemas and definition in the VDR, as described in section 3.7.3.1 “Emission certificate format”. Then, when the job scheduler triggers the issuance and the ECA Core has obtained the emission evaluation result, it instructs the ECA Agent to issue an emission certificate VC.

The ECA Agent creates a credential offer with all the predefined values according to the appropriate schema. The offer refers to a specific connection (e.g. vehicle) and contains a nonce as well as a cryptographic commitment, thus it is unique and bound to a specific issuer and holder, e.g. the ECA and the vehicle respectively.

The credential offer is sent directly to the vehicle via the SSI Agents and DIDComm. The vehicle in its turn creates a credential request that corresponds and bounds with the received credential offer. The credential request also contains a cryptographic commitment that maps to unique secret, which only the vehicle controls. In addition, the request is sent to the ECA, as the offer, through the SSI Agents and DIDComm.

The ECA accepts the credential request and is ready to issue the credential, namely, the emission certificate. ECA creates and transmits it to the vehicle. Finally, it is stored by Vehicle Agent in its local

digital wallet and is ready to be presented when needed. The emission certificate VC is linked to a specific vehicle and can be presented only by it.

4 Outlook

4.1 Controlling Authority

The overall architecture of the emission certification ecosystem, as inherits the main characteristic of VCs is depicted in Figure 30, where the Vehicle represents the Holder, Issuer represents the ECA and Verifier could be any third party. This third party can be assumed by anyone, i.e., a person, an institution/organization, or even a thing that wants to perform emission data verifications for any legal purpose, hence, providing adaptability in various use cases. Following the VC format, the emission certificates are stored in the Vehicle wallet and can be managed by its SSI Agent.

In the first step of the issuance, as described in section 3.7.3 “Emissions data certification”, the ECA writes its public keys in the VDR in order to be able to provide emission certificates. Thus, it is responsible for the issuance of them. However, when a certificate is issued, it can be provided and proved cryptographically to third parties only be the corresponding vehicle. The vehicle, e.g. the holder, must consent in order to provide its emission certificates to anyone.

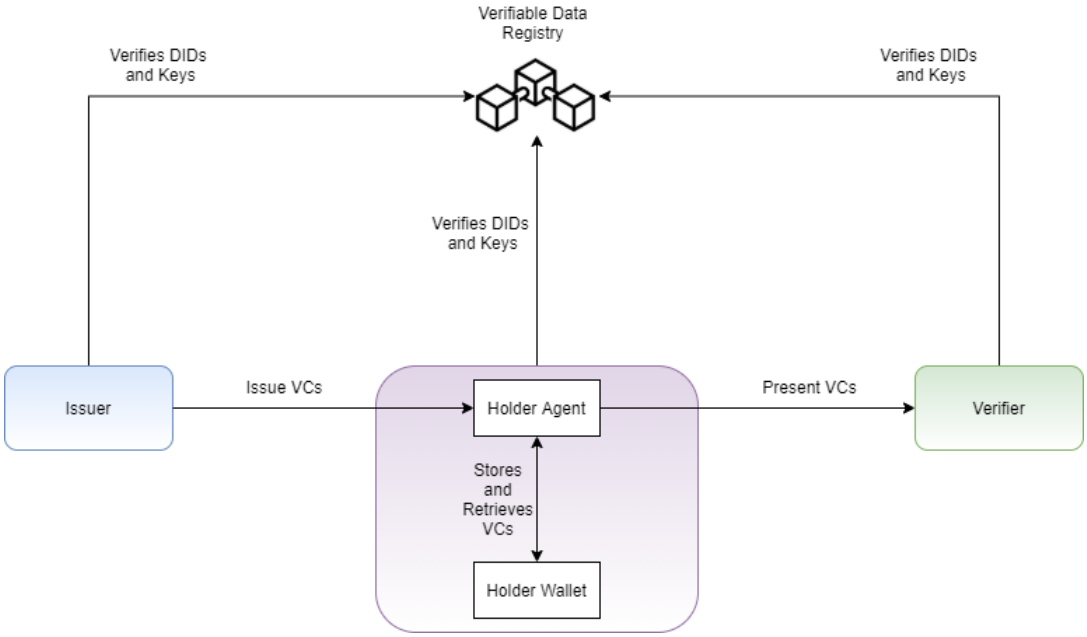


Figure 30 - DIAS VC trust ecosystem

The trust model in the VC ecosystem places the holders in the first place, the verifies needs to trust the issuers as well as the VDR. More specifically the trust model includes:

- Holders trust their agents and wallets
- Holder and Verifiers trusts the Issuers
- All entities trust the VDR

4.1.1 Procedure of certificate presentation

The emission certificate VC contains the certification result for a specific vehicle in a time period. This certificate can be presented by the vehicle to any third party. This third-party verifier should have an SSI Agent associated with it. Its SSI Agent is responsible to request the emission certificate VCs from the vehicle SSI Agent. If the vehicle accepts, the vehicle’s SSI Agent responds with proof that the third party can validate. This procedure of emission certificate presentation and verification relies also on

messages that are exchanged between the corresponding SSI Agents, hence, all the information exchange between the vehicle SSI Agent and the third party SSI Agent, i.e., presentation and presentation request, is performed via DIDs and DIDComm.

To start the procedure, the third party verifier creates, what is called, presentation request, which defines the requested attributes as well as which issuers are trusted to certify their validity. Presentation request contains the accepted VC schemas and definitions, as described in section 2.4.3 These are written in the VDR by trusted ECA issuer instances that have been initialized and verified to provide emission certificate VCs. The presentation request is sent to vehicle to be consumed by its SSI Agent.

The response to a presentation request is referred as verifiable presentation, or simple presentation. The vehicle consent to offer its presentation that is generated using the local wallet and encapsulates the attributes from one or multiple VCs allowing the signature verification of the corresponding issuer, e.g. ECA. Moreover, the vehicle signs the presentation, to prove that is the owner of presentation.

As the third-party verifier gets the presentation, is able to verify it. In this context, the verification of the ECA digital signature, which is achieved using a DID, is a vital stage in this procedure. The validity of the VC is based on the VDR, that contains the data as well as metadata that enables the emission certificate validation, such as Public DIDs of the ECA Issuers, VC schemas and definitions. The presentation contains a signature of the ECA that can be verified through the VDR. Thus, this cryptographic proof is used by a third-party verifier to validate it against the information coming from the VDR. If the VC is valid, the verifier is assured that the emission values are certified by the ECA and is able to perform further actions.

4.1.2 Handling of certification expiry

The third-party that wants to verify emission certificate VC must have the ability to ensure that they have not been revoked, nor have they expired. In the DIAS context, the emission certificate does not include any expiration date since its content can be used for all the lifecycle of the vehicle. Meaning that the vehicle will always be able to provide emission certificate VCs that are referred to a specific period of time and it is valid only for this specific interval. For example, the vehicle could provide its emission certificate VC for the year 2022 in the year 2025.

The validity of emission certificate VC may contain a revocation service that defines whether an emission certificate has been revoked from the issuer, .e.g. the ECA and is no longer cryptographically valid. The emission certificate VC may be revoked when a vehicle is withdrawn or completely damaged. The emission certificate validity can be handled by using accumulators and purpose-built servers or VDRs. ECA is able to define a revocation registry in the VDR that will be used by the verifier to check whether an emission certificate VC has been revoked.

4.2 Adaptation and re-using the technology stack for other use cases

4.2.1 SSI Degree of maturity

As part of this project, the current status of SSI technology is assessed and its underlying frameworks are used and employed to add an integrity and identity dimension towards vehicle-to-cloud data exchange. Although the general SSI concepts remain, it should be noted, that the field of decentralized identity is subject to dynamic changes and this document reflects the status as of December 2021.

Currently, there seems to be a convergence of standards towards W3C Verifiable Credentials and BBS+ Signature Suites for Linked Data Proofs. Most agent frameworks have adopted DIDComm v2 as of now but still have a strong dependency on Hyperledger Indy. However, support for other ledgers is

improving and the Indy DID [40] should increase technical and semantic interoperability between different solutions.

The push for interoperability can also be observed from a public side, such as the joint workshop series between Innovation, Science and Economic Development Canada (ISED) and Directorate General for Communications Networks, Content and Technology from the EU, which aims to enable baseline compatibility between Canada and EU regarding digital credentials and wallets such as adherence to international regulatory standards and best practices [41]. Additionally, international bodies are already involved in the standardization process of SSI, Blockchain and DLT technologies, such as the "ISO/TC 307 Blockchain and distributed ledger technologies" see [42], on an international scale, "CEN-CLC/JTC 19" on EU scale (see [43]), and "DIN 043 02 04 AA" on a German national scale (see [44]).

The most notable projects visible/adopted in Germany are the public-private partnerships of IDunion [45] and ONCE [46], which focus on use cases such as e-government, mobility and tourism.

One important keynote project for self-sovereign identity technology is the European Self Sovereign Identity Framework (ESSIF) [47] by the European Blockchain Services Infrastructure (EBSI) [48]. Its goal is to develop a generic and interoperable Self-Sovereign Identity (SSI) framework for the realization of an ecosystem and enablement of cross-border decentralized identity use cases. One important use case is the eIDAS regulatory framework [49], which for example is expected to publish an EU-wide development toolbox in September 2021.

The government of British Columbia is already using SSI technology in production for government use cases. It launched OrgBook BC, a searchable directory of verifiable data about business organizations in British Columbia, uses the Verifiable Organizations Network (VON) and Sovrin framework, and will be deployed to other Canadian jurisdictions. More input on the progress of current SSI projects can be found in Pöhn et al [50].

4.2.2 Accountability for governmental authorities

As part of this work package, subjects who may act as trusted certifiers/validators of data, as well as identity-providing parties, have been introduced. In a future, where the identity of a vehicle itself is not just defined by a public DID but a multitude of attributes that are supplied via verifiable credentials by these identity-providing parties, it is important that public policy outlines which governmental authority is responsible for providing each attribute.

Translating the DIAS use case into the real-world application, the PTI and ECA are envisioned as a (potentially) joint entity like the periodical technical providers in Germany (e.g. TÜV or Dekra). Currently, these are granted the authority to undertake bi-annual technical inspections and grant (or revoke) a vehicle's roadworthiness in the form of a sticker. DIAS VLA subject certifies the PTI as well as grants a vehicle's license. This means that in a real-world application, the VLA may represent a multitude of different (governmental) authorities, which in turn may again use a trust chain which has to be part of their digital representation.

For example, in Germany technical inspections are done by an "Amtlich anerkannter Sachverständiger" (officially approved technical inspector), which is certified by a "Prüfgesellschaft" (vehicle inspection company) which is assigned by the "Landesregierung" (federal state's government).

Modeling a trust chain using SSI can help in resolving these interdependencies between fragmented authorities in a way that is transparent, secure, and intermediary-independent. Most importantly, it provides a framework to digitize and automate these governmental functions while reducing obstacles such as manual steps. While modeling these trust chains between governmental and private

authorities, it is important that there is a joint approach between public policy expertise and technical expertise, which is why it could make sense to push for this in initiatives such as Gaia-X or Catena-X.

4.2.3 Re-utilization of trustful data exchange for other use cases

During the design phase of the trustful vehicle-to-cloud data exchange, great emphasis was put on re-usability and extensibility. As mentioned, DIAS approach does not represent a finalized architecture for the transfer of data from a vehicle to one verifying entity and the use of vehicle data from another entity.

It should be noted that due to computational workload of signature creation as well as the current agent design, SSI will most likely act as a layer of integrity and traceability rather than a direct payload carrier. SSI is considered as a toolbox, which can be employed to find a solution for an anti-tampering use case. Combining the proposed hash chain with self-sovereign identity technology can provide data integrity and data provenance guarantees to other use cases which are susceptible to tampering.

With the generic approach in mind, a simple container architecture that will be demonstrated on a CCU was implemented. The approach is easily adaptable, because it is payload-agnostic, ledger-agnostic, and cloud provider agnostic. Further, it is built on open-source frameworks such as Aries Hyperledger and KUKSA.val.

Future adaptations may be employed for other use cases with low tooling efforts, given that the connected processing unit (in our case the CCU) can run host/operate a container architecture.

Potential future use cases may include other types of certificates for conformity with similar regulations, e.g. related to commercial vehicles, transport of dangerous goods etc., the internal-combustion-electric usage split of a hybrid engine, or similar scenarios.

5 Conclusions

As described in DIAS deliverable D5.2, a cloud reporting system forms an important part of an overall diagnostic system. It supplies additional functionalities that ECU-based diagnostic functions alone cannot deliver. These include:

1. accessing relevant generalizable parameters by authorized entities
2. documenting the usage of tampering detection software among vehicles
3. delivering easily-verifiable certificates of cloud emissions verifications back to the vehicle.

Therefore, in-vehicle non-tampering detection goes hand in hand with securing the transmission of vehicle data from CCU to cloud to offer more continuous emission testing and verification in contrast to (bi-)annual spot checks.

Within this deliverable, self-sovereign identity technology based on distributed ledgers to provide a trusted and integrity-safeguarding transmission of vehicle data (such as NOx emission parameters or software and calibration identifiers) have been examined and leveraged. The solution is able to maintain integrity along the data value chain on the cloud side. The data is then made available to a third party for an emission certifications procedure, which generates a tamper-proof and easily verifiable emission certificate that is securely sent directly to the vehicle.

This generic approach of a digital certification based on continuously collected data samples sent from vehicles in motion is considered applicable for other use cases where integrity and accountability are paramount, and where regulators or public authorities require its attestation.

6 List of references

[1] Worldwide Web Consortion (W3C), "W3C DID Working Group," [Online]. Available: https://www.w3.org/2019/did-wg/. [Accessed 16 12 2021].

[2] Worldwide Web Consortium (W3C), "Decentralized Identifiers (DIDs) v1.0," 16 12 2021. [Online]. Available: https://w3c.github.io/did-core/.

[3] ISO - International Organization for Standardization, "ISO 8601 - Date and Time Format," [Online]. Available: https://www.iso.org/iso-8601-date-and-time-format.html. [Accessed 16 12 2021].

[4] JSON.org, "Introducing JSON," [Online]. Available: https://www.json.org/json-en.html. [Accessed 16 12 2021].

[5] The Eclipse Foundation, "Eclipse Kuksa," [Online]. Available: https://www.eclipse.org/kuksa/. [Accessed 16 12 2021].

[6] Reliance General Insurance, "What is Pollution Control Certificate (PUC)," [Online]. Available: https://www.reliancegeneral.co.in/Insurance/Knowledge-Center/Blogs/What-is-Pollution-Control-Certificate-PUC.aspx. [Accessed 16 12 2021].

[7] Sovrin Foundation, "What is self-sovereign Identity?," [Online]. Available: https://sovrin.org/faq/what-is-self-sovereign-identity/. [Accessed 16 12 2021].

[8] T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," [Online]. Available: https://www.rfc-editor.org/rfc/rfc3986. [Accessed 16 12 2021].

[9] Worldwide Web Consortium (W3C), "Verifiable Credentials Data Model - Verifiable Data Registries," [Online]. Available: https://www.w3.org/TR/vc-data-model/#dfn-verifiable-data-registries. [Accessed 16 12 2021].

[10] International Council on Clean Transportation, "A technical summary of Euro 6/VI vehicle emission standards," [Online]. Available: https://theicct.org/sites/default/files/publications/ICCT_Euro6-VI_briefing_jun2016.pdf.. [Accessed 12 12 2021].

[11] DIAS Project, "Official Deliverables of DIAS project," [Online]. Available: https://www.dias-project.com/Deliverables/All_WPs. [Accessed 16 12 2021].

[12] Decentralized Identity Foundation, "DIF FAQ: What is a Verifiable Data Registry?," [Online]. Available: https://identity.foundation/faq/#what-is-a-verifiable-data-registry. [Accessed 21 12 2021].

[13] Ishwarappaa and J. Anuradhab, "A Brief Introduction on Big Data 5Vs - Characteristics and Hadoop Technology," 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915006973. [Accessed 16 12 2021].

[14] Hyperledger Foundation, "Hyperledger Foundation," [Online]. Available: https://www.hyperledger.org/. [Accessed 16 12 2021].

[15] World Wide Web Consortium (W3C), “Verifiable Credentials Data Model - Verifiable Presentations,” [Online]. Available: https://www.w3.org/TR/vc-data-model/#dfn-verifiable-presentations. [Accessed 16 12 2021].

[16] P. Brijesh and S. Sreedhara, “Exhaust emissions and its control methods in compression ignition engines: a review,” in *International Journal of Automotive Technology, 14(2)*, 2013, pp. 195-206.

[17] S. U. Essa, “PARTICULATES EMISSION CONTROL USING EXHAUST AFTER-TREATMENT TECHNOLOGY: A REVIEW.,” in *IJRAR-International Journal of Research and Analytical Reviews (IJRAR) 8(1)*, 2021, pp. 769-772.

[18] Publications Office of the European Union ., “Regulation (EC) No 715/2007 of the European Parliament and of the Council of 20 June 2007,” [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32007R0715. [Accessed 16 12 2021].

[19] R. D. C. &. L. G. Mandaroux, “A European Emissions Trading System Powered by Distributed Ledger Technology: An Evaluation Framework.,” in *Sustainability, 13(4)*, 2021, p. 2106.

[20] P. A. M. Devan, F. A. Hussin, R. Ibrahim, K. Bingi and M. Nagarajapandian, “In 2019 IEEE Student Conference on Research and Development (SCOReD),” in *IoT Based Vehicle Emission Monitoring and Alerting System.*, IEEE, 2019 October, pp. 161-165.

[21] Irish Tax and Customs, “Calculating Vehicle Registration Tax,” [Online]. Available: https://www.revenue.ie/en/importing-vehicles-duty-free-allowances/guide-to-vrt/calculating-vrt/nitrogen-oxide-emissions.aspx. [Accessed 16 12 2021].

[22] J. Sharma, G. S. Sindhu, S. Sejwal, J. Solanki and R. Majumdar, “Intelligent vehicle registration certificate,” in *Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, 2019, February, pp. 418-423.

[23] Worldwide Web Consortium (W3C), “Decentralized Identifiers,” [Online]. Available: https://www.w3.org/TR/did-core/. [Accessed 16 12 2021].

[24] Worldwide Web Consortium (W3C), “Verifiable Credentials Data Model,” [Online]. Available: https://www.w3.org/TR/vc-data-model/. [Accessed 16 12 2021].

[25] Worldwide Web Consortium (W3C), “Verifiable Credentials Data Model - Verifiable Credentials,” [Online]. Available: https://www.w3.org/TR/vc-data-model/#dfn-verifiable-credentials. [Accessed 16 12 2021].

[26] Decentralized Identity Foundation, “DIDcomm,” [Online]. Available: https://identity.foundation/didcomm-messaging/spec/. [Accessed 16 12 2021].

[27] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen and X. Shen, “Internet of vehicles in big data era,” in *IEEE/CAA Journal of Automatica Sinica, 5(1)*, 2017, pp. 19-35.

[28] Google Cloud, “Designing a connected vehicle platform on cloud IOT core,” [Online]. Available: https://cloud.google.com/architecture/designing-connected-vehicle-platform#architecture_diagram. [Accessed 16 12 2021].

[29] Bosch IO, “Bosch IoT Insights,” [Online]. Available: https://bosch-iot-suite.com/service/insights/. [Accessed 21 12 2021].

[30] Pallets, “Welcome to Flask,” [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/. [Accessed 16 12 2021].

[31] Worldwide Web Consortium (W3C), “BBS+ Signatures 2020,” [Online]. Available: https://w3c-ccg.github.io/ldp-bbs2020. [Accessed 16 12 2021].

[32] Bosch IO, “Bosch IoT Insights - MongoDB Query Service,” [Online]. Available: https://bosch-iot-insights.com/static-contents/docu/html/MongoDB-Query-Service.html. [Accessed 16 12 2021].

[33] i-SCOOP, “IoT 2019: Spending, Trends and Hindrances Across Industries,” [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/iot-2019-spending-trends.

[34] Z. Doffman, “Cyberattacks On IOT Devices Surge 300% In 2019, ‘Measured In Billions’, Report Claims,” Forbes, 2019. [Online]. Available: https://www.forbes.com/sites/zakdoffman/2019/09/14/dangerous-cyberattacks-on-iot-devices-up-300-in-2019-now-rampant-report-claims/#461686625892. [Accessed 20 12 2021].

[35] European Commission, “European Union C-ITS Security Credential Management System,” [Online]. Available: https://cpoc.jrc.ec.europa.eu/Documentation.html. [Accessed 20 12 2021].

[36] H. Halpin, “Semantic Insecurity: Security and the Semantic Web,” in *PrivOn 2017 - Workshop Society, Privacy and the Semantic Web - Policy and Technology*, Vienna, Austria, pp. 1-10.

[37] G. Fedrecheski, J. M. Rabaey, L. C. P. Costa, P. C. Calcina Ccori, W. T. Pereira and M. K. Zuffo, “Self-Sovereign Identity for IoT environments: A Perspective,” *Global Internet of Things Summit (GIoTS),* pp. 1-6, 2020.

[38] AUTOSAR , “Specification of Communication AUTOSAR CP Release 4.3.1,” [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_COM.pdf. [Accessed 12 16 2021].

[39] R. Haakegaard and J. Lang, “The Elliptic Curve Diffie-Hellman (ECDH). The Elliptic Curve Diffie-Hellman (ECDH),” December 2015. [Online]. Available: http://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf. [Accessed 16 12 2021].

[40] Hyperledger Indy, “Indy DID,” [Online]. Available: https://github.com/hyperledger/indy-did-method. [Accessed 16 12 2021].

[41] Government of Canada, “Canada and the European Union Joint Workshop Series for Enabling Interoperability and Mutual Support for Digital Credentials,” [Online]. Available: https://www.ic.gc.ca/eic/site/153.nsf/eng/h_00006.html. [Accessed 16 12 2021].

[42] ISO - International Organization for Standardization, “ISO/TC 307 Blockchain and distributed ledger technologies,” [Online]. Available: https://www.iso.org/committee/6266604.html. [Accessed 16 12 2021].

[43] European Committee for Standardization, “CEN/CLC/JTC 19 Blockchain and Distributed Ledger Technologies,” [Online]. Available: https://standards.cencenelec.eu/dyn/www/f?p=205:7:0::::FSP_ORG_ID:2702172&cs=148F2B917E4B67BCFD6FE36CE0EA923AC. [Accessed 16 12 2021].

[44] DIN Deutsches Institut für Normung e. V., “Projekte von NA 043-02-04 AA,” [Online]. Available: https://www.din.de/de/mitwirken/normenausschuesse/nia/nationale-gremien/74630/wdc-grem:din21:268432395!search-grem-details?masking=true. [Accessed 16 12 2021].

[45] IDunion, “IDunion,” [Online]. Available: https://idunion.org/. [Accessed 16 12 2021].

[46] ONCE Identity, “ONCE - Sichere digitale Identitäten,” [Online]. Available: https://once-identity.de/. [Accessed 22 12 2021].

[47] Decentralized Identity Web Directory, “ESSIF - European Self Sovereign Identity Framework,” [Online]. Available: https://decentralized-id.com/government/europe/eSSIF/. [Accessed 16 12 2021].

[48] E. Dávila, “European Blockchain Services Infrastructure,” European Blockchain Partnership, [Online]. Available: https://www.itu.int/en/ITU-T/webinars/20201104/Documents/Emilio%20Davila-EBSI%20presentation%20ITU%20ED%2004112020.pdf?csf=1&e=CAHb8b. [Accessed 21 12 2021].

[49] European Commission, “eIDAS Regulation,” [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation. [Accessed 16 12 2021].

[50] D. Pöhn, M. Grabatin and W. Hommel, “eID and Self-Sovereign Identity Usage: An Overview,” [Online]. Available: https://www.mdpi.com/2079-9292/10/22/2811/pdf. [Accessed 16 12 2021].