IoT

MDPI

Article

Decentralised IOTA-Based Concepts of Digital Trust for Securing Remote Driving in an Urban Environment

Juhani Latvakoski *, Vesa Kyllönen and Jussi Ronkainen

VTT Technical Research Centre of Finland, Kaitoväylä 1, 90570 Oulu, Finland; vesa.kyllonen@vtt.fi (V.K.); jussi.ronkainen@vtt.fi (J.R.)

* Correspondence: juhani.latvakoski@vtt.fi; Tel.: +358-40-5200-149; Fax: +358-20-722-2320

Abstract: The novel contribution of this research is decentralised IOTA-based concepts of digital trust for securing remote driving in an urban environment. The conceptual solutions are studied and described, and respective experimental solutions are developed relying on digital identities, public key cryptography with a decentralised approach using decentralised identifiers (DIDs) and verifiable credentials (VCs), and an IOTA-based distributed ledger. The provided digital trust solutions were validated by executing them according to the remote driving scenario but with a simulated vehicle and simulated remote driving system. The hybrid simulation mainly focused on the validation of functional, causal temporal correctness, feasibility, and capabilities of the provided solutions. The evaluations indicate that the concepts of digital trust fulfil the purpose and contribute towards making remote driving more trustable. A supervisory stakeholder was used as a verifier, requiring a set of example verifiable credentials from the vehicle and the remote driver, and accepting them to the security control channel. The separation of control and data planes from each other was found to be a good solution because the delays caused by required security control can be limited to the initiation of the remote driving session without causing additional delays in the actual real-time remote driving control data flow. The application of the IOTA Tangle as the verifiable data registry was found to be sufficient for security control purposes. During the evaluations, the need for further studies related to scalability, application of wallets, dynamic trust situations, time-sensitive behaviour, and autonomous operations, as well as smart contract(s) between multiple stakeholders, were detected. As the next step of this research, the provided digital trust solutions will be integrated with a vehicle, remote driving system and traffic infrastructure for evaluation of the performance, reliability, scalability, and flexibility in real-world experiments of remote driving of an electric bus in an urban environment.

Keywords: cyber–physical systems; machine-to-machine communications; Internet of Things; smart energy systems; smart mobility systems; communications; security; trust

Citation: Latvakoski, J.; Kyllönen, V.; Ronkainen, J. Decentralised IOTA-Based Concepts of Digital Trust for Securing Remote Driving in an Urban Environment. *IoT* **2023**, *4*, 582–609. https://doi.org/10.3390/iot4040025

Academic Editor: Amiya Nayak

Received: 15 September 2023
Revised: 10 November 2023
Accepted: 21 November 2023
Published: 29 November 2023

1. Introduction

Recent developments in the technologies related to distributed computing and communications have led to an essential paradigm change in the operation of embedded physical devices. These devices no longer only operate as autonomous embedded products, but they are now more and more part of cyber–physical systems (CPSs), where information-based interaction with back-office IT systems is an essential part of the service that the devices provide to users. This paradigm change is especially challenging in the context of critical CPSs because cyber security seriously challenges the safety of operations. The contribution of this research is related to such a critical CPS in the context of autonomous vehicles, which are targeted to remotely driven vehicles in urban areas. The focused research problem of this work is related to the lack of digital trust in such remote driving of an autonomous electric bus in urban environments [1].

Remote driving refers to assisting (semi)autonomous vehicles to navigate from one location to another without a driver but with the help of a remote driver/operator. It is

expected that the vehicle has the necessary state-of-the-art automated solutions capable of detecting and reacting to normal situations rapidly in real time. It has been seen that such a vehicle is capable of operating autonomously for a short period of time (e.g., some seconds), but the events and objects in the surroundings may be so complicated that help from a human remote operator and artificial intelligence (AI) assistance are needed to help the journey to the intended destination in a trustworthy way. In addition, the remote operation can take care of supervision, increase safety, and help to solve exceptional situations in traffic. When such remote driver and back-office services are included, the importance and need for security solutions related to enterprise IT systems is obvious.

The contribution of this research is related to the research and development of decentralised IOTA-based concepts of digital trust for securing remote driving in an urban environment. The applied research method is an experimental approach using hybrid simulation [2]. The challenges related to securing remote driving were analysed, the concepts for digital trust were studied and defined, and then the experimental solutions for them were developed. The experimental solutions were built on digital identities using public key cryptography with decentralised identifiers (DIDs), verifiable credentials (VCs), and an IOTA-based distributed ledger. An example of a remote driving scenario was executed in a hybrid simulation manner, and evaluations were carried out against the challenges detected from the real remote driving scenario.

The remainder of this paper is organised as follows. The methods, challenges and prior art of the concept development are discussed in Section 2. The concepts of digital trust are described in Section 3. The IOTA-based experimental solutions of the concepts are clarified in Section 4, and the results are evaluated in Section 5. Finally, the concluding remarks are presented in Section 6.

2. Methods for the Concept Development

The applied research methods for the concept development are clarified in this section, following an analysis of challenges detected in securing remote driving in urban environments and a discussion on prior art related to the concept development for solving the digital trust challenge.

2.1. Research Challenge and Methods

The main focused research challenge in this work is the lack of digital trust between various stakeholders and resources in the remote driving of autonomous vehicles in urban environments, Figure 1. The operational environment of an autonomous vehicle contains a number of physical objects (such as traffic signs, lights, other vehicular objects, and people). When connecting such a vehicle over wireless networks to a remote driving system, there are a number of stakeholders and persons that could potentially challenge the cyber security of the system. Because of the criticality of the remote driving operation, only the stakeholders/people/objects that are allowed to access, share information, and control the vehicle shall be allowed to do so. Thus, the lack of digital trust in this context is the major targeted research challenge.

The applied research was carried out step by step. First, the challenges and requirements for securing remote driving were analysed, and then the analysis of prior art solutions was carried out as a parallel process for the development. Based on all this information, the first prototyping of the digital trust solutions was carried out, and the validation aspects were considered part of it. The remote driving of autonomous vehicles in an urban context is very demanding to validate. Validation cannot be performed in a real urban environment because it may cause big risks to the safety of other stakeholders on streets and introduce high R&D costs, and the tests are impossible to control and repeat in practice. Therefore, the validation needs to be carried out in some other way, for example, using simulation environments, but the challenge is how the results of the validation match with the reality on the streets. Based on preceding research and experiences, here, we select the application of the hybrid simulation methodology, where software in the loop (SIL),

hardware in the loop (HIL), and environment simulations are applied in a mixed manner with real objects to reach the validation objective [2]. This method enables validation even if the other operational components are not available in the development time. In addition, it can make validations cheaper and safer. Therefore, the hybrid simulation methodology is applied in the experimental R&D of the digital trust solutions. The first step in the validation of the digital trust concept solutions using the hybrid simulation approach is depicted in Figure 2. The major focus in this first step is to validate the functional and causal temporal correctness and evaluate the feasibility and capabilities of the provided solutions to contribute towards increasing digital trust in remote driving. As the next step of this research, the provided digital trust solutions are integrated with a vehicle, a remote driving system, and traffic infrastructure to evaluate the performance, reliability, scalability, and flexibility in real-world experiments of the remote driving of an electric bus in an urban environment. This article focuses on the solutions and results from the first step of the hybrid simulation-based validation of the digital trust solutions.

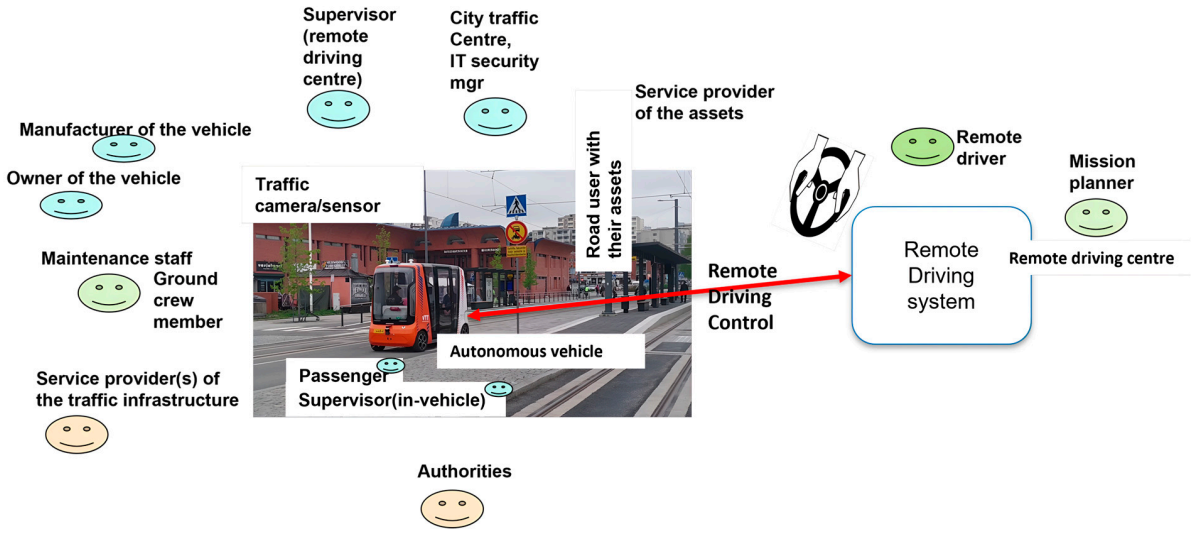

Figure 1. A view to the research question—an example of stakeholders and resources that require mutual digital trust in order to safely participate in remote driving of an autonomous vehicle in an urban environment.

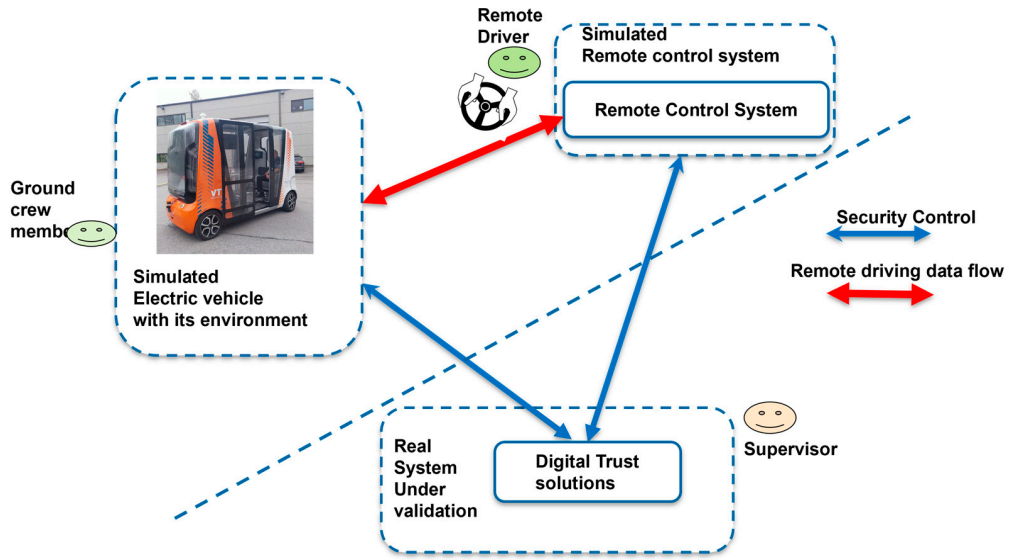

Figure 2. The first step in the validation of the digital trust conceptual solutions (real system under validation) using the hybrid simulation approach.

2.2. An Analysis of Challenges in Securing Remote Driving

The remote driving system has several actors and physical entities, which work in close interaction with each other and the surrounding environment, Figure 1. An example of such interactions is depicted in Figure 3 to present an overview of the remote driving process. The remote driving system has several different actors that work in close interaction with each other to make remote driving possible and secure. A security/safety check of the physical vehicle needs to be carried out by a ground crew member in close interaction with the remote driving centre. The presence, location, and mobility of the other objects in the environment need to be detected by the remote driving centre in a trustworthy way. Particularly, the situation of the vehicle as well as its environment needs to be known by the remote driver when the remote-control operations of the vehicle are performed. The role of guiding, warnings, and alerting is essential in ensuring successful and trustworthy remote driving. At the beginning of this research, an analysis of critical actors and assets was carried out. There are at least 14 critical physical assets related to information, operation, and devices that may or may not be critical for the operation when system elements interact with each other. These critical interactions were analysed to detect potential risks, problems, and challenges for security, privacy, and trust and then used for detecting potential threats that may occur in remote driving operations.

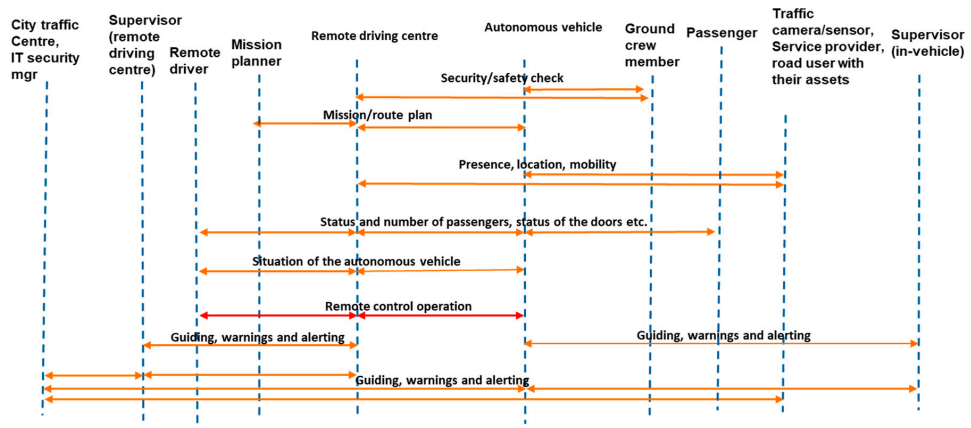

Figure 3. Actors and information based on interactions among them (see also Figure 1). The red arrow represents the remote driving related control of autonomous vehicle, and orange arrows to the information required in it.

The remote driving case includes serious risks to safety. For example, there is a risk that some external stakeholder such as a cyber attacker is able to attack the targeted system and perform remote driving control actions that may trigger or lead to an accident, or the system can be used in a terroristic attack, causing the loss of human life and damage to traffic and infrastructure of urban areas. An external stakeholder may steal the vehicle and use it for their own actions that are against the interests of the owner of the vehicle and related service provider(s). Alternatively, they may mislead the remote driver or the positioning/route following/situational awareness subsystems of the autonomous vehicle to make wrong decisions by providing wrong or misleading information, which may result in an accident with potential damage and loss of human life. An attacker may also expose privacy-sensitive information from the system and misuse it for some purposes without the permission of the owner, against GDPR regulations.

These are only examples of the risks that can be serious for the safety of remote driving. There are also several other risk areas such as the possibility of erroneous operation within the autonomous vehicle. For example, there may be errors in the positioning of the autonomous vehicle, the steering control actuator system, or the braking control system. In addition, risks arise from stakeholders' mistakes, there may be errors in the communication channels, unexpected situations in the surroundings of a vehicle may lead to erroneous actions, and an attacker may trigger some surrounding entity to work in a malicious manner.

The decentralisation of any such a risk may also lead to further challenges. For example, when an accident occurs, it is naturally essential to know the reasons for it. Because the vehicles are autonomous, it can be very challenging to know what actually happened in the system just before the accident occurred. Another challenge arises in cases where someone uses sensitive data for illegal purposes, e.g., the misuse of privacy-sensitive data that contravenes GDPR regulations. In addition, an important challenge area, especially in remote driving, is related to the need to gain the confidence of stakeholders (people on the streets and authorities) in the operation of the system.

Based on the analysis of the critical actors (13), critical physical assets (14), and interactions between them, a threat analysis was carried out. It revealed 30 potential threats related to the identified actors, physical system-related assets or operations, the consequences of which were estimated to compromise security-, privacy-, and or trust-related issues leading to potential safety problems in the remote driving case. Based on the referred threat analysis, the requirements presented in Table 1 were considered.

Table 1. An analysis of requirements for securing remote driving operation.

Requirement	Justification
Requirement 1 (R1). The source/sender of the mission plan must be trusted. It must be verified that the plan is sent by a real mission planner and that the plan is not modified.	The autonomous vehicle is not misused.
R2. Mission plan must be stored encrypted and can be updated only by a trusted source.	The autonomous vehicle is not used for illegal purposes.
R3. The autonomous vehicle must be able to drive safely.	Safe autonomous driving.
R4. Information on the presence, location, and mobility of humans/animals/artificial entities on the road must be trusted. It must be verified that input information is sent by real entities and the information is not modified.	Fraudulent information is not sent to the system.
R5. The location of input information must agree with the location of the vehicle.	Safe autonomous driving.
R6. The results of the emergency reasoning based on vehicle situation information must be kept safe. Manipulating the results may cause wrong emergency operations.	Emergency stops and vehicle pullovers are performed as they should be.
R7. The vehicle information for the remote driver must be trusted. It must be verified that vehicle information is sent by a real vehicle and that the information is not modified.	The remote driver receives correct information from the vehicle.
R8. The vehicle information for the remote driver must be real-time, i.e., the delay must be below a defined threshold (ms/s). Otherwise, the remote driver may perform fatal remote driving operations. This can be checked with timestamps, for example.	The remote driver receives correct information from the vehicle.
R9. The information from the remote driver to the vehicle must be trusted. It must be verified that information is sent by a real remote driver and that the information is not modified.	The vehicle receives correct information from remote driver.
R10. The information from the remote driver to the vehicle must be in real-time. This can be checked, for example, with timestamps.	The vehicle receives correct information from the remote driver.
R11. The operating systems of all system components must be kept up to date. Firewalls and antivirus software are used. Complex passcodes and passwords are used. Secure networks are used. Router security is checked, which can be low by default.	Emphasises system confidentiality.
R12. The communication between all the components of the system should be secure. Secure communication protocols (HTTPS, SSH, SFTP, FTPS) and encryption should be used. Cryptographic keys should be protected, for example, using subsystem isolation.	Emphasises system confidentiality.
R13. The system must be traceable. This makes it possible to analyse reasons for problems, which increases the system safety in the future.	To be able to analyse what happened in a dangerous situation or accident. Developing system safety.

After the analysis, it was estimated that identification, access control, and traceability are essential digital trust problems. Here, identification refers to the secure identification of the physical entities, service providers, users, and owners in the remote driving ecosystem. Access control refers to the capabilities of the owners to control the use of their resources by giving access rights to other users. Traceability refers to the capability to monitor events/data from multiple resources owned by different stakeholders in a reliable way. In addition, it was seen that when unexpected events occur during remote driving operations in an urban traffic context, it is essential to know the situation just before such events. For example, who was in charge of the remote driving? What interactions happened between the vehicle and the remote driver? What other vehicles and road users were nearby? What information was provided by traffic infrastructures (e.g., traffic lights, traffic cameras)? What were the positions of the entities? And so on.

2.3. A Discussion of the Prior Art

The traditional perimeter-based network security model has serious risks for the assets of an enterprise because an attacker may in one way or another gain access to the enterprise system. The likelihood of attackers gaining access to systems has increased, as remote work seems to increase the risks of security threats and the phishing of credentials; therefore, the likelihood of a malicious user being able to access the resources in enterprise systems is increasing. Zero Trust security models have been developed to contribute towards solving these problems, focusing on resource protection and the premise that trust is never granted implicitly but must be continually evaluated [3]. When speaking about safety-sensitive cases, like the remote driving of autonomous vehicles, it is obvious that the traditional perimeter-based network security model is not sufficient, and a Zero Trust type of security model needs to be applied instead. Traditional schemes are not enough for the remote driving case because of the need to ensure trust relationships between multiple persons, organisations, and physical assets simultaneously and to control access to the related monitoring and controlling data streams. This is also emphasised in [4], where an essential challenge is described as minimising the risk of an unauthorised takeover of a remote-driven vehicle. The responses raised particularly interesting questions with respect to the concerns addressed with the solution described in this paper. A need was seen for using data to overcome problems in establishing fault in case of, e.g., an accident. Similarly, open and transparent data sharing by an independent body was considered necessary, reflecting our concept of Trust Storage. The responses also highlighted that it is *"essential that there is an efficient process in place that will enable companies to verify the driver without delay"*. Driver verification and authorisation are at the centre of this paper.

The World Wide Web Consortium (W3C) created the Verifiable Credentials Data Model 1.0 specification, which was approved as a full W3C standard in September 2019. The specification applies self-sovereign identities, also called decentralised identifiers (DIDs), as the basis for the solution [5–7]. Applying DIDs and distributed ledger technologies makes it possible to avoid dependence on centralised registers/certificate authorities for key management, which are typically used in public key infrastructure (PKI) [8]. Therefore, the approach is also called a decentralised PKI [9]. The system works so that a holder (person, item, service, etc.) obtains a decentralised identifier (DID) together with its public key from a reliable provider, who also stores it in some type of verifiable data registry, such as blockchain/distributed ledger, a distributed database, or any other sufficiently trusted publicly accessible utility. After that, the holder requests verifiable credentials from various issuers who, after determining that the credentials can be granted, use their private key to digitally sign the credential (and any other cryptographic material needed to verify the issuer's credentials), and issue it to the holder to store in their digital wallet. Note that to preserve privacy, this issuance process does not need to involve any interaction with a verifiable data registry—in other words, no personal data needs to be written to a blockchain or third-party data repository. The process can be fully confidential between the issuer and the holder. Later, when the holder needs to gain access to some resource

controlled by a verifier, the verifier requests digital proof of one or more credentials from the holder. If the holder consents, the holder's wallet generates and returns the proofs to the verifier. Since the proof contains the issuer's DID, the verifier can use it to read the issuer's public key and other cryptographic data from the verifiable data registry. In the final step, the verifier uses the issuer's public key to verify that the proofs are valid and that the digital credential has not been tampered with [10]. Because of the safety-sensitive nature of remote driving, the confidence of the involved stakeholders and users could benefit from such a digital trust ecosystem so that the control concept could be acceptable in a public urban traffic system.

The other challenge is related to the ability to monitor events and data from multiple resources owned by different stakeholders of the traffic ecosystem in a reliable way. Because remote driving happens in an urban traffic environment, it is obvious that trust in the monitored trace is very important, including from the point of view of authorities. When applying the W3C approach to digital trust, the application of blockchain/distributed ledger technologies for tracing provides a possible approach, and these are applied to the remote driving case in this study. Distributed ledger technology (DLT) refers to the storage, distribution, and exchange (sharing) of data among the users of private or public distributed computer networks located in multiple sites [11]. One example of DLT is blockchain, which is the underlying technology of Bitcoin [12]. Blockchain is a linked-list type of data structure, which is updateable only via consensus among a majority of existing peers in the network, and thus, there is not a single CA controlling the ledger. Each block contains a set of transactions and their hash, with a link to the previous block hash. Only after successful consensus can a new block be added to the chain. Another type of DLT is the directed acyclic graph (DAG), where each transaction is represented as a node that is linked to one or several other transactions. The links are directed so that they point from earlier transactions to newer ones without allowing loops [11]. The transactions provide validation for each other, but a transaction cannot validate itself. A new transaction has to validate one or more previous transactions to join the DAG. Every new transaction refers to its parent transactions, signs their hashes, and includes the hashes in the new transaction. One essential difference compared with blockchain is that a DAG does not need miners, which makes it cheaper (no mining fee), faster, and more scalable. This makes the DAG quite an interesting technology for the CPS, which has a large number of transactions that need to be almost free to be realistic. An example of a DAG application is IOTA [11,13–15], which calls its distributed ledger the Tangle. The current realisations of IOTA (after 1.5 Chrysalis) apply elliptic curve cryptography (ECC) and binary operations [16].

A survey of applications of blockchain technologies for securing vehicular networks was analysed, e.g., in [17]. The identified major future challenges are related to scalability, privacy, quantum computing attacks and prototyping/simulations. For example, Fang et al. studied a zero trust-based protection scheme for users of the Internet of Vehicles [18]. Their simulation-based study on the application of Zero Trust network architecture shows that the security level of the system related to data transmission, stability, and trust can be improved. Their comparison with traditional boundary-centred security protection shows that the solution can protect a wider range of application security challenges. Another study suggests using a blockchain to secure smart vehicle communications [19]. Their proposal is a decentralised, privacy-preserving architecture for the smart vehicle ecosystem that uses public keys and overlay networks to establish connections between different stakeholders for different data. The use cases they mention are similar to this paper, i.e., trusted data sharing from the vehicle to various stakeholders. However, their study does not include the concept of a remote driver or how to secure the driver's identity and other required credentials.

Existing publications focusing specifically on remote driving do not seem to cover driver validation as a central issue. For example, ref. [20] describes a remote driving architecture but states only that the vehicle and remote driving station use a secure login to a mediating gateway. Another architecture described in [21] focuses on communications,

sensing, and AI control, but does not address driver authentication or authorisation at all. Other identified papers do not address these issues either: e.g., ref. [22] focuses on the usability of mobile networks for remote driving and [23] uses a survey with emphasis on other technical remote driving characteristics, such as latency, user experience, and situational awareness. The security of remote-driven vehicles generally refers to the security during the driving process, including the security of the sensors, operating system of the vehicle, control system, and vehicle-to-everything (V2X) communication. Existing solutions are mainly focused on those areas of security and trust. Our solution focuses mostly on digital trust before the remote driving process starts.

Based on the state-of-the-art analysis, it seems that there is a lack of digital trust solutions, especially practical experiments for securing remote driving in urban environments, to which our contribution is specifically related. Thus, the contribution of this research is related to the application of a Zero Trust kind of approach as the starting point for reducing the risk of attacks coming from inside the perimeter network. The actors are always verified first before they are allowed to remotely operate with the critical physical assets—autonomous vehicles in this case. The trust relationships between system actors are ensured using verifiable credentials and self-sovereign identities according to [5]. The security control process is applied according to the Trust over IP (ToIP) approach [10], but applying IOTA channels for secure messaging, and the IOTA Tangle is applied as the verifiable data registry. The concepts of digital trust are represented as a layered model, IOTA-based experimental solutions are described, and finally, the evaluations and lessons learned are discussed.

3. Concepts of Digital Trust

Establishing and maintaining digital trust between different stakeholders is vital in safety-critical scenarios such as remote driving. In this section, we present a conceptual approach and concepts of digital trust for achieving the necessary assurance between all participating entities in a safety-critical application and describe how it is used in the remote driving case.

3.1. Conceptual Approach

The key elements of the digital trust concepts are depicted in Figure 4 by dividing them conceptually into trust, credentials, control data, and trust storage levels. The trust level is related to the relationships between people, organisations, and physical assets (resources), which are called trust entities in this study. For example, in the remote driving case, several trust relationships between stakeholders are needed, such as between the autonomous vehicle owner and the autonomous vehicle, between the autonomous vehicle owner and the remote driving company, between the remote driving company and the remote driver, and between the remote driver and authorities (e.g., driving license). The credentials level is related to the digital identities of trust entities, the means to provide credentials from the issuers to holders, storing the credentials to wallets, and checking the credential proofs by verifier(s). The control data level is related to exchanging control data between the entities in an end-to-end manner in a secure way. The control data can include credentials, security keys or other cryptographical material, or meta information on the data stream related to the real data flow between the trust entities required to be known by the other parties in the communication. The trust storage level is related to storing the transactions related to critical trust relationships between trust entities, smart contracts, verifiable credentials, and other security, privacy, and safety-related critical events (traces monitored from the system) so that they cannot be changed after they are verified and added to the distributed ledger.

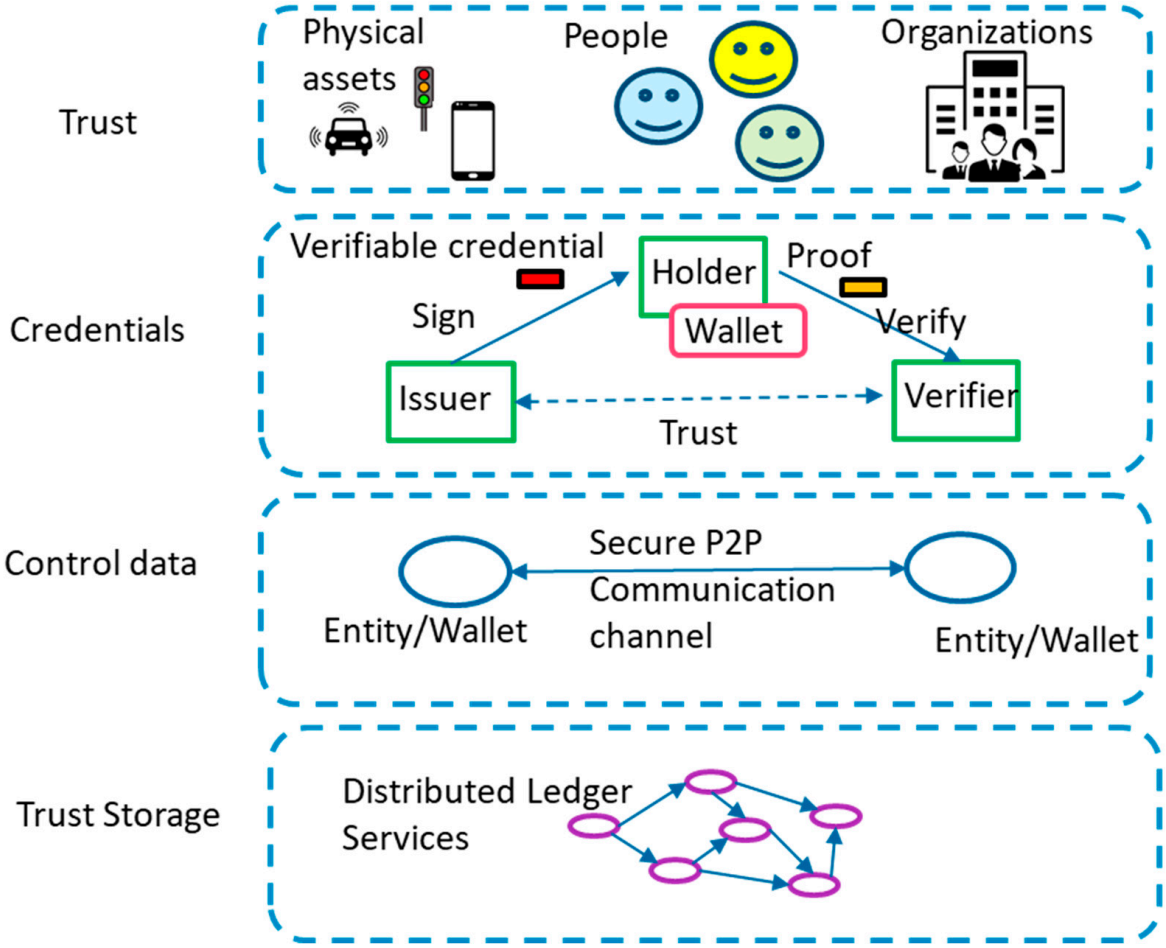

Figure 4. The key conceptual elements of digital trust.

In this research, the referred conceptual solutions were experimentally developed by relying on verifiable credentials and digital identities. In the solution, the PKI is combined with a decentralised approach using decentralised identifiers (DIDs) and verifiable credentials (VCs). An IOTA-based distributed ledger is applied for traceability. We developed a component called the trust monitor, also used here as the role of the supervisor (CPShub Trust@vtt), which applies referred technologies in order to study the operation of the provided digital trust concepts.

3.2. Trust Relationships among System Actors

An example of trust relationships among authorities, organisations, people, and physical assets is depicted in Figure 5. Typically, physical assets are owned by a person or an organisation. In the example, the vehicle is owned by a person, and the ownership is registered into the authorities' system and thus enforced by legal means. The vehicle is manufactured by a certain OEM organisation, and it is accepted for use in road traffic in the target country by authorities. The owner may make a contract with some organisation for a certain service, e.g., a remote driving service. The organisation may have multiple contractual relationships with each other. A person may be an employee of a certain organisation. That person may own physical assets, which might also have multiple relationships of similar kinds. A specific organisation may own and host the traffic lights based on the authorised relationships.

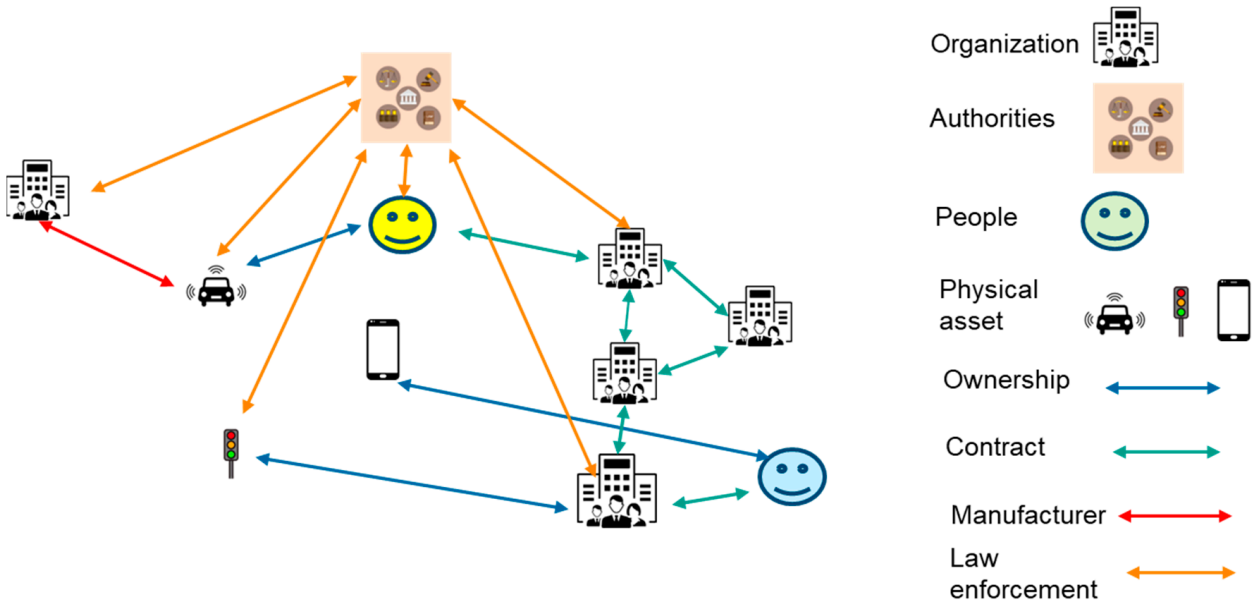

Figure 5. An example of trust relationships among authorities, organisations, people, and physical assets.

It is obvious that such trust relationships can be very complex, dynamic, and hierarchical, and the represented example is just a very simple view of the kind of trust network referred to here. The starting point of the proposed concept is that no trust relationship is automatically trusted, but it is always verified to ensure that the trust relationship is valid, similar to the work in [3]. When speaking about remote driving in an urban context, trust relationships are critical for security and safety. Therefore, these relationships need to be known and proven in order to allow such remote driving to happen.

3.3. Credentials as Proof of Trust

Trust needs to be translated to the real digital world using certificates called digital credentials. These credentials are valid only if they are issued by an acceptable organisation, and they can be verified by some other trusted organisation. This has been represented in the form of the trust triangle, Figure 6 [5]. Our concept relies on the W3C model of decentralised identifiers (DIDs), and verifiable credentials rely on the use of PKI public/private keys, as depicted in the trust triangle. An *issuer* (person, item, service, etc.) obtains and stores its DID with the related public key and any related cryptographic proof to a verifiable data registry. Then, the issuer signs the verifiable credential related to the requested certificate content using the issuer's private key and gives it to the credential requester (the holder), who can store the verifiable credential in their wallet or some trustworthy file storage. When some organisation (verifier) needs to ensure that the holder has a certain certificate, it can request digital proof of one or more credentials from the holder. The holder can then sign and provide the proof to the verifier using the holder's DID and private key, the certificate content in full or just the requested part, and the signature received from the issuer. The verifier can then find the holder's and issuer's public keys from the verifiable data registry. Finally, the verifier can use the public keys to ensure that the proof was generated by the holder for the specific request, to ensure that the digital signature in the proof was given by the issuer indicating the proof is valid, and to check that the hash of the content is correct. After the application of this W3C DID methodology, it is assumed that there is a high level of trust that the system actors are what they claim to be. After this step it seems possible to continue to the actual control process between the physical resources.

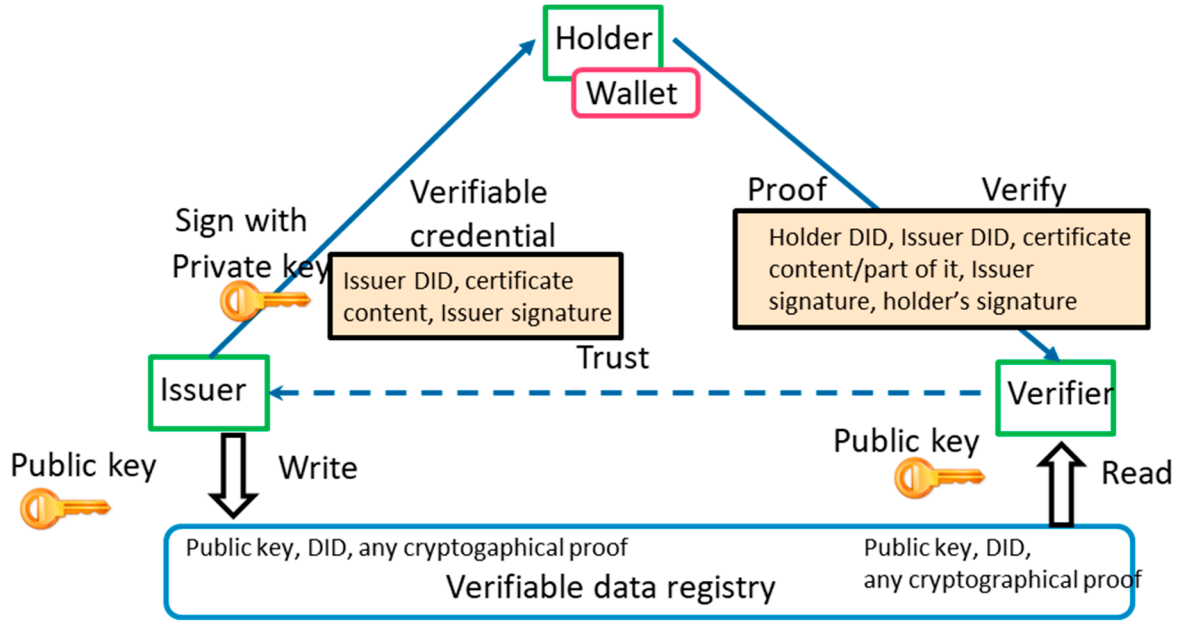

Figure 6. The trust triangle. If the verifier trusts the Issuer (dashed line), trust can be established with the Holder (solid line).

3.4. Control of Data Exchange

When the proof of the trust relationships between system actors and assets is achieved, then the process for creating end-to-end (E2E) security data flow can be initiated. A key selection in our concept is the separation of the control and data planes from each other, Figure 7. The control plane is needed for exchanging security- and communications-related information, such as encryption keys, other cryptographical material, the required quality of service (QoS) level from communications services, and meta information related to the real E2E data flow, which is to be established between the entities. It is envisaged that the control data are exchanged using the DID method according to the Trust over IP (ToIP) protocol [10]. The method applies DIDs with public/private keys as well as a verifiable data registry to ensure the validity of the control data. It is estimated that when the control data for security are exchanged, the secure E2E data flow can be established in a secure way according to the defined security-, communications- and data-related parameters. In this way, it is envisaged that the resulting E2E data flow fulfils all the requested quality levels of the application, which are very high in our focused remote driving scenario.

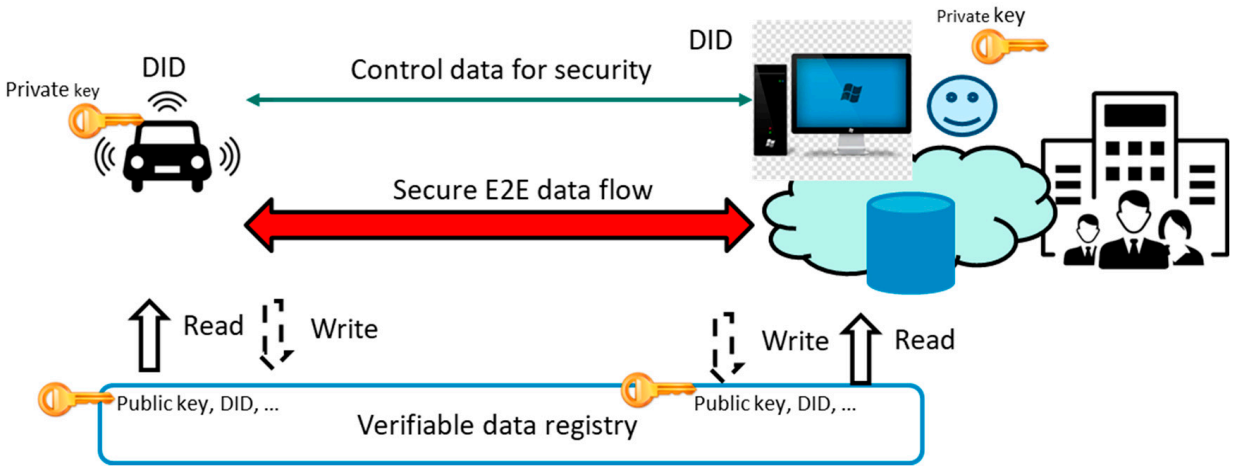

Figure 7. Control data for security to establish a secure end-to-end data flow among physical entities.

3.5. Verifiable Data Registry—Trust Storage

A key element in the proposed concept is the application of a verifiable data registry for storing trust-related data. It is obvious that such a data registry needs to be open, shared among stakeholders, and reliable enough that all the stakeholders can be sure that its content cannot be changed, modified, or compromised by anyone. In this research, the application of blockchain/distributed ledger technologies (DLTs) as the basis for the verifiable data registry was selected. In addition, the application of the directed acyclic graph (DAG) technology was considered to scale better and be cheaper because it does not need miners [16]. It is estimated to perform better in transaction-rich applications like the remote driving case. Therefore, it was selected for our experimental validation.

An example of a simplified verifiable data registry applying the principles of an IOTA-type DAG is depicted in Figure 8. Each transaction is represented as a node that is linked to one or several other transactions. When a new transaction would like to join the DAG, it needs to be linked with two transactions that are already linked with the DAG, i.e., reach a cumulative weight (CW) of 3. For example, the new transaction G may be linked with transactions A and B in the joining process, resulting in added hashes and digital signatures in the transaction. However, transaction G is not yet an approved transaction itself in this phase; it is called a 'tip' in IOTA terminology [16]. It needs to have at least two transactions linking to it to obtain a confirmed transaction status. For example, transactions C, B, and D are confirmed transactions. Transaction D is strongest because its CW is 5. Transactions A and W are only partially confirmed transactions because they have only one transaction linked with them (their CWs are 2).

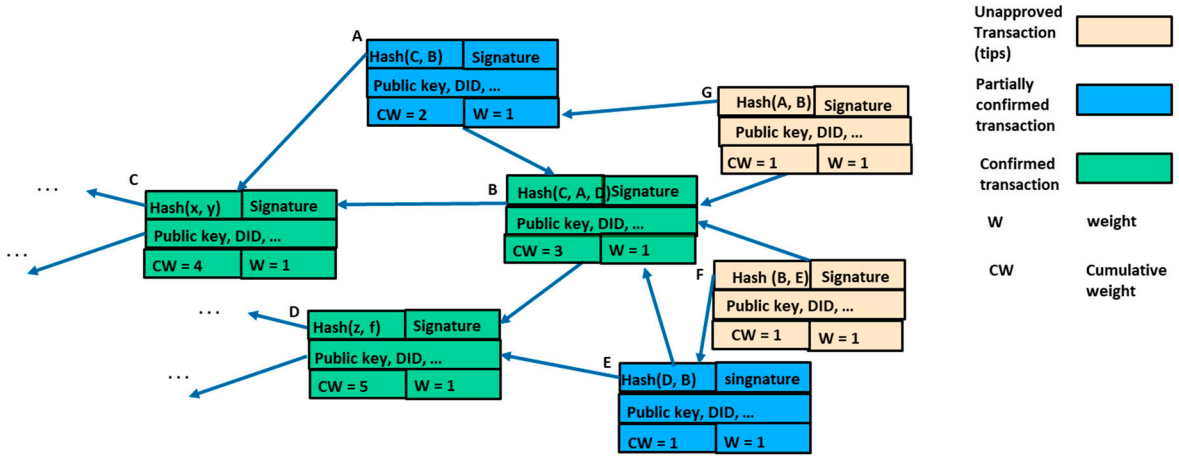

Figure 8. An example of a verifiable data registry with application of the principles of an IOTA-type DAG. Letters A–G denote individual transactions in the registry.

3.6. Application of the Concept for Securing Remote Driving

The application of the concept for securing remote driving is depicted in Figure 9. The practical ecosystem in the traffic context is simplified here to highlight the basic assumptions of this work. Accordingly, the system consists of an autonomous vehicle and remote-control system, which collaborate with the supervisory system. The supervisory system (CPShub Trust@vtt in the figure) takes care of the services required for the use of the distributed ledger, which is based on the IOTA Tangle. The blue arrows represent the security control process, which is needed to reliably identify the stakeholders and their resources. This is carried out by the supervisory system, which first executes the verification of the credentials of the stakeholders (e.g., remote driver) and endpoints (e.g., autonomous vehicle). If all the credentials are valid, then the entities are accepted to join the security control channel (secure IOTA channel) for the second security step. In the second security step, the entities exchange the security tokens and the parameters of the end-to-end remote driving communication channel via the secure IOTA channel. This exchange is secured

using PKI so that only the endpoints, i.e., autonomous vehicle (e.g., AUNE@vtt) and the remote driving system controlled by the specific remote driver (e.g., remote–driver–A@RemoteControlCentre) have access to the tokens and parameters. After the security tokens and parameters are exchanged between the endpoints, they are used to establish an end-to-end remote driving communication channel. The channel is depicted by the red arrow in Figure 7, and it is used for actual remote driving operations without any additional overhead. In normal operation, this end-to-end driving session can work as long as needed to remotely drive the vehicle to the intended destination.

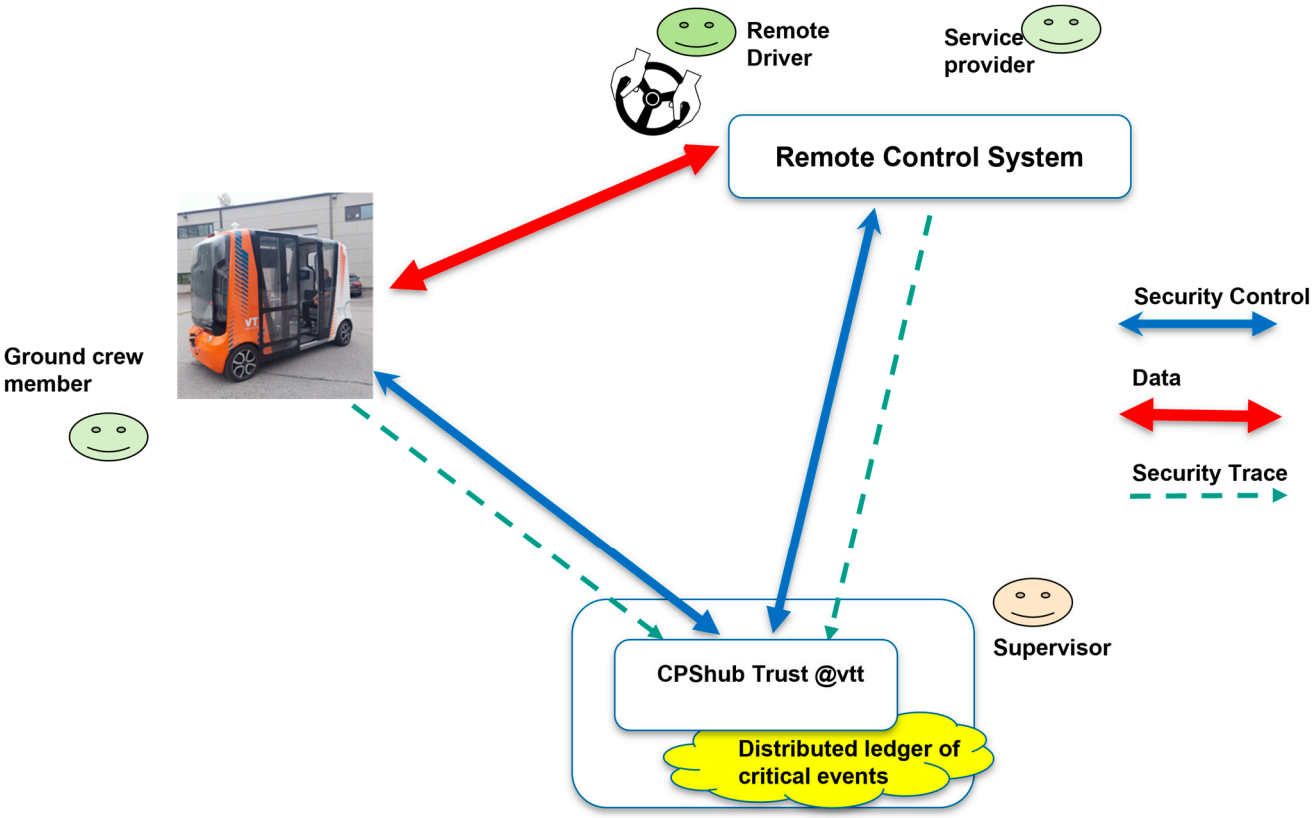

Figure 9. The application of the concept for securing remote driving.

When something exceptional happens, e.g., the remote driver is no longer able to control the vehicle as a result of some problems, then the vehicle needs to operate autonomously without any remote control. These kinds of events are called driving mode changes, which are important to store with timestamps for accountability reasons. The same applies also for in-vehicle events such as a passenger pushing a safety button in the autonomous vehicle, accidents/problems detected with the vehicle sensors, any changes detected in credentials (including security control at start-up), and also the time and locations when these events happened. The dashed green arrow in Figure 9 represent the tracing of these critical events and storing them in the local database of the supervisory trust monitor (CPShub Trust@vtt) with verifiable transaction into the IOTA Tangle. When some unexpected situations happen in urban traffic, the IOTA Tangle can be applied to study the preceding situations related to remote driving operations.

The main estimated advantages of the concept for securing remote driving arise from the increasing trust level for lowering the risk of attacks coming inside the perimeter network. The security control process verifies the actors first before they are allowed to do anything related to remote driving or provide information to be used in the remote driving process. The trust relationships between system actors are ensured using verifiable credentials. Thus, all the actors and operations are verified first before any remote operation and control is allowed, which is estimated to reduce risks for misuse and attacks. In

addition, in the proposed concepts, the IOTA Tangle is applied as the shared data registry, which can be applied to store trust contracts and critical event recorded traces from the remote driving operation. The advantages of such a shared and trustworthy data registry arise from the reliability and transparency of all the stakeholders in the field, such as, e.g., authorities.

4. Experimental Solutions of the Concepts of Digital Trust

In this section, we describe the experimental solutions that were developed to prove the provided concepts of digital trust for securing remote driving. The validation and demonstration of the core elements of the security control process were performed using a hybrid simulation approach. We simulated a subset of the stakeholders and their interactions in the IOTA Tangle. The following sections describe the demo scenario in general and the authentication and remote drive connection steps in particular. Finally, we discuss the technical details of the implemented IOTA-based experimental solutions.

4.1. Validation Scenario Description

Our validation scenario focuses on the two challenges presented above. It verifies that participants who intend to participate in the remote driving operation have all the necessary credentials and that significant operational events are recorded in an unmodifiable way for future reference and auditing. Trust-building is achieved via W3C DIDs and verifiable credentials, which are created beforehand for the participants for the scenario. The credential-issuing members also are created specifically for the scenario.

The scenario consists of software modules simulating a remote drive-capable vehicle, a remote driver who is an employee of a remote driving company, a supervisor who monitors and controls access to the communication channel between the vehicle and the driver, and a ground crew member who approves the vehicle for driving after a pre-drive inspection. These stakeholders of the system are presented in the object diagram in Figure 10. Communication in the security control channel is presented with green arrows and communication in the data channel is presented with the red arrow. All the stakeholders using trust solutions have a user interface (UI).

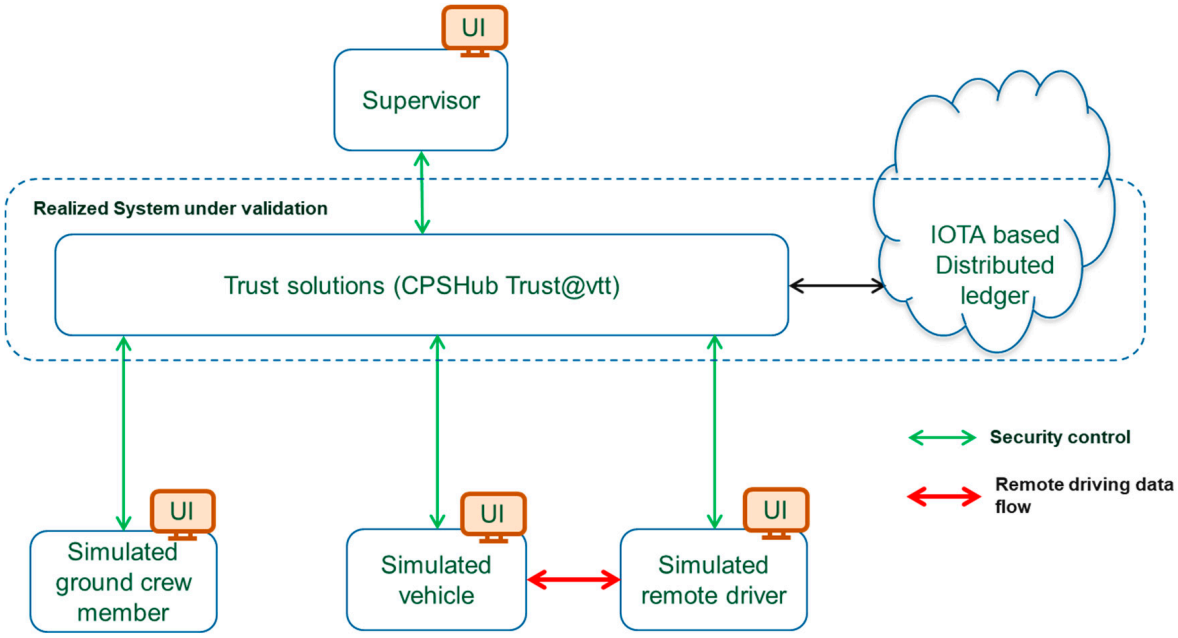

Figure 10. The stakeholders of the validation scenario. The dashed area depicts the realized system that is validated in the scenario.

The objective of the scenario is to demonstrate how the necessary trust among the participants can be established so that the driver can assume control over the vehicle and to demonstrate the immutable storage of significant driving events.

The timeline of the scenario is such that all the participants authenticate to the system, request access to the security control channel, and, once granted, post status messages and other requests to others on the channel to find potential counterparts—remote drivable vehicles for remote drivers, and vice versa. The remote driver initiates the exchange of point-to-point connection parameters, and once the vehicle approves, remote driving may commence via the data channel connection. Communication over this data channel in the scenario is limited to simulated requests of driving mode changes from autonomous to remote and back, which are initiated by either end. Both the vehicle and driver sides post notifications of changes in their status, such as driving mode changes, to the ledger. The timeline of the validation scenario is summarised in the sequence diagram in Figure 11. The commencement and closedown of the data channel connection is indicated by red arrows. Dashed arrows indicate return messages.

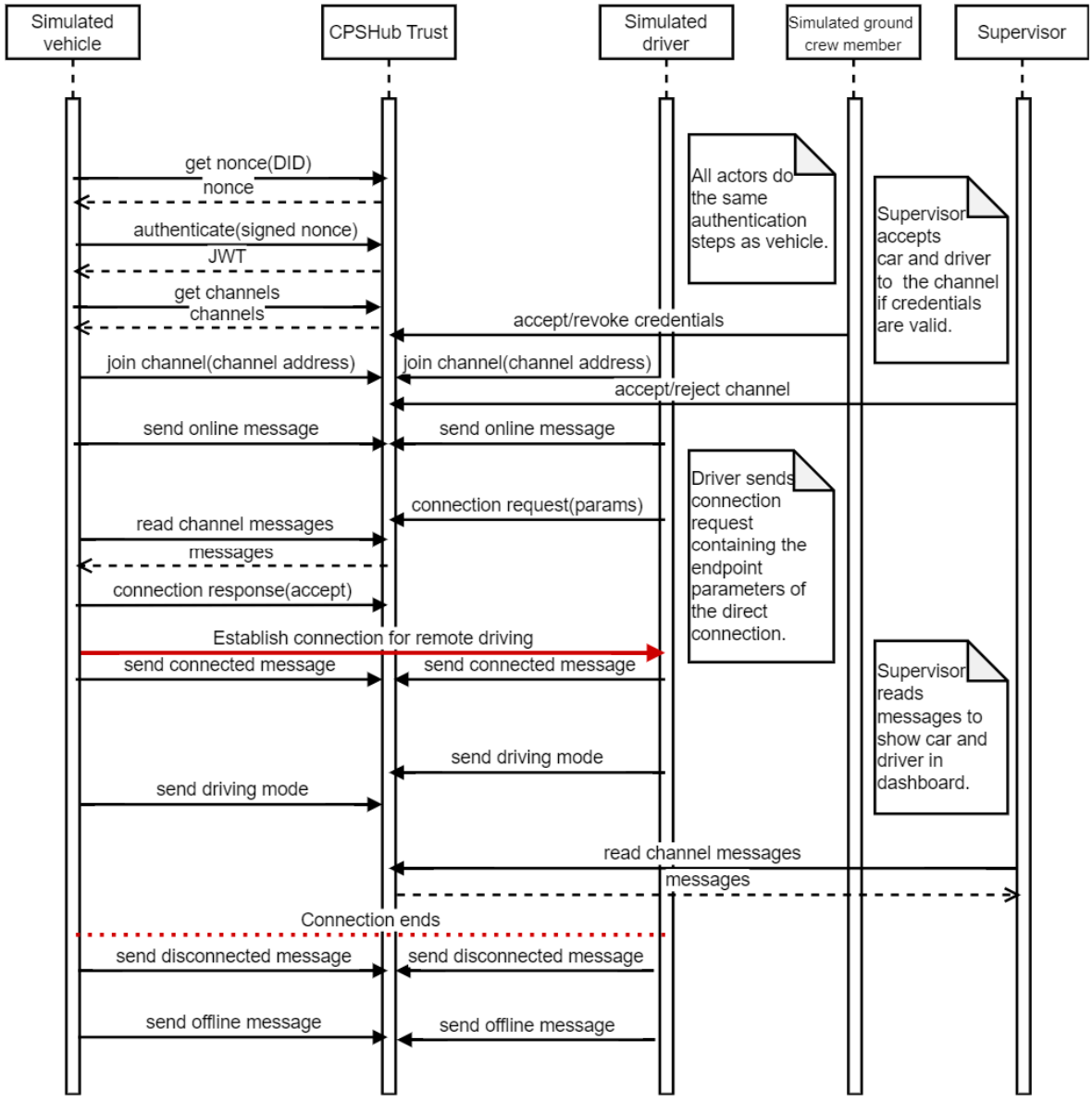

Figure 11. Sequence diagram of the validation scenario.

4.2. Authentication and Credentials

All participants independently authenticate to the system with their DID and a nonce via a challenge–response scheme that verifies control over the DID, i.e., the possession of the private signing key associated with the DID. After authentication, the driver and the vehicle find the communication channel address using a keyword query in the system. Then, each requests a subscription to the channel and remains on hold until their subscription is approved. The approval is completed by the supervisor, who is the owner of the channel and who periodically scans the channel for new subscription requests. The supervisor validates and inspects each subscriber's credentials and manually approves the subscribers to the channel based on the set of presented credentials. Figure 12 shows the supervisor's user interface where the vehicle has been accepted to the channel and the remote driver is waiting for the supervisor's acceptance. Acceptance subscription is indicated by green colour and waiting for acceptance by yellow colour. In this case, all the vehicle's three required credentials and the remote driver's two required credentials are drawn in green, indicating that they are present and valid.

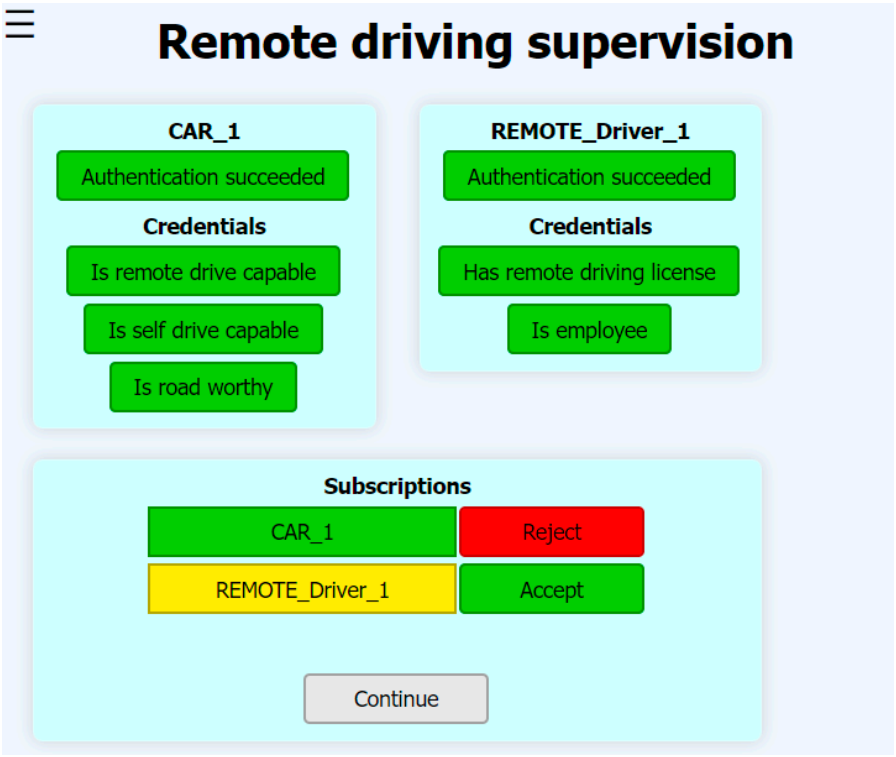

Figure 12. Credentials in the supervisor UI.

The supervisor decides which credentials to require from each participant before granting access to the common control channel. In the demo, most credentials are issued automatically with a script, and the supervisor client has a pre-set list of which credentials it requires. The credentials for the vehicle in the demo are issued by the fictitious vehicle manufacturer and vehicle inspection authority, concerning the vehicle's autonomous and remote drive capability. The driver is certified to be an authorised remote vehicle operator and an employee of the fictitious remote drive company, as issued by a fictitious licensing authority and the remote drive company. In addition, the use of a transient credential is demonstrated by the ground crew member software module, which issues limited-time credentials for the vehicle's current roadworthiness. Notably, the ground crew member does not need access to the common control channel, as the credentials are stored in the system rather than on the channel.

In summary, the key concepts for building the necessary trust for operating on the common control channel are as follows:

- Each participant proves control over their DID via a challenge–response scheme that requires possession of the private signing key associated with the DID.
- Participants have cryptographically solid credentials for all attributes required for channel access.
- Credentials are issued by known trusted sources (and their credentials can also be verified).
- The common control channel is encrypted so that only subscribers can read the messages and only for the time they are subscribed.
- The supervisor controls access to the channel and can read and cryptographically verify the prospect subscribers' credentials.
- The common control channel stamps every message with the sender's DID to indicate the source of the message.

4.3. Establishing the Remote Drive Connection

After approval, the vehicle and the driver clients post their status to the channel to become aware of each other and to be able to establish the point-to-point connection required for remote driving. The remote driver eventually sees the vehicle(s) in their UI on a list of potentially remote drivable vehicles on the channel. Connection establishment starts when the driver picks the vehicle from the list and requests a connection; see Figure 13. Optionally, the driver may also query and view the vehicle's credentials from the system. As part of the request, the driver's client software may include technical connection details so that the vehicle can reach the driver on the point-to-point channel. Alternatively, this information may be included in the vehicle's response if the point-to-point connection should be established with the vehicle as the server. However, point-to-point connection details should not be revealed to other subscribers of the common control channel, such as other drivers and vehicles. Thus, these details are additionally encrypted with the recipient's public encryption key. Public keys are part of the DID document, which each participant published in the ledger. As a result, only the recipient of the connection establish request can decipher the technical connection details.

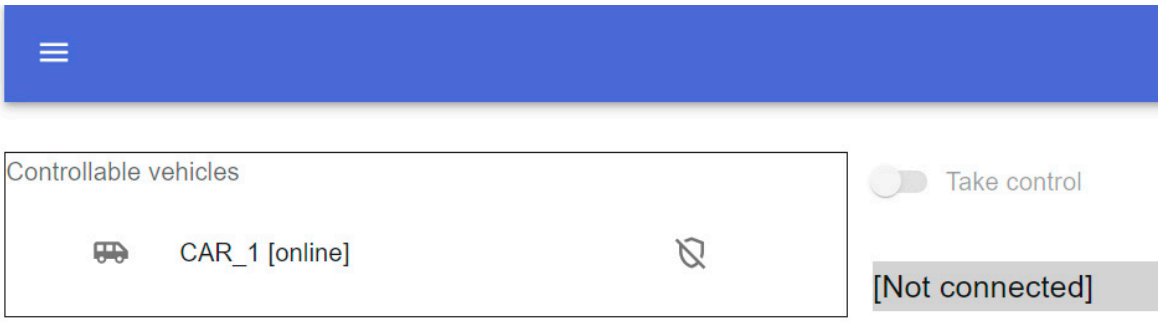

Figure 13. Car list in the remote driver UI.

The vehicle approves the connection establishment request and replies over the common control channel, leaving a trace with a timestamp of when the connection between the driver and vehicle was established. The parties can now communicate over the data channel, which in our validation scenario is a WebSocket connection hosted by the vehicle client.

4.4. Remote Drive Operation

It is not possible to conduct actual remote driving over the common control channel, which is based on a ledger and thus slow to operate due to the inherent delay (and potentially, cost!) of approving each transaction. This is what the point-to-point connection is for. However, the common control channel is also useful during point-to-point operation

as an immutable trace of important events encountered during a remote driving session. Our demonstration focuses on storing driving mode changes, which are posted to the channel individually by each participant, along with the sender's timestamp. Over time, a trace of significant events concerning each remote driving session from the viewpoint of both ends is thus accumulated. It is also noteworthy that the IOTA-based solution described in the next section makes it possible to generate a special auditor key for later inspection of all channel messages. This could be particularly useful in a later investigation of remote driving events in case of, e.g., an accident. Access to this key could be restricted to the relevant authorities. The owner of the channel (the supervisor in our case) can also read every message on the channel. Also, a dedicated user could be created, subscribed, and approved to the channel during channel creation for the sole purpose of being able to read all the messages ever posted to the channel.

The supervisor client monitors the trace and presents a visual status of the active remote driving session(s). This feature could make it possible for the supervisor to also have a traffic management or other authority role. Figure 14 presents a driving session in the supervisor UI. At the yellow points, the driver drives the vehicle remotely. At red points, the vehicle drives autonomously.

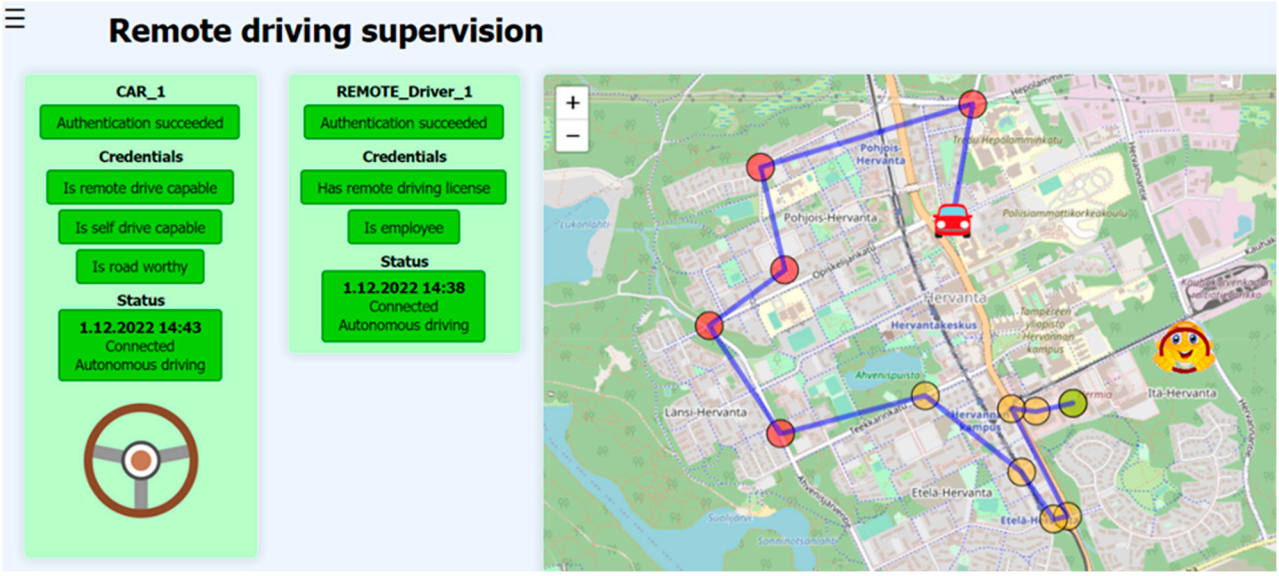

Figure 14. Driving session in the supervisor UI.

The supervisor can also periodically monitor the validity of the subscribers' credentials and, if channel access criteria are no longer met, revoke the subscription to prevent the user from receiving or sending messages on the channel.

4.5. Identities and Verifiable Credentials in IOTA Integration Services

Digital wallets are a convenient way to manage secrets such as private encryption keys, verifiable credentials, and value tokens stored in a ledger. Wallets can be beneficial for both security and convenience, as the wallet can unify access to various ledgers such as those used by different cryptocurrencies. Thus, the owner of the wallet only needs to authenticate to the wallet, not separately to each ledger and application. In this way, the owner does not have to directly handle the highly sensitive authentication data, decreasing the risk of the data getting into the wrong hands (provided the wallet itself is secure). A popular example of such wallets is the MetaMask [24] browser extension that is currently in version 11.4.

In our demo scenario, both the remote driver and the supervisor are humans and log on with their personal DIDs. IOTA claims that its services are aligned with the EU's upcoming eIDAS and the European Identity Wallet, but details of these are not provided [25]. Also,

we were not able to identify any existing wallet solutions that would enable the use of an IOTA DID or VCs associated with it for authentication.

One option to enable the use of a wallet is to explore the use of IOTA Smart Contracts [26], which does support MetaMask, but even if it had turned out to be usable for wallet identification, it would certainly have complicated the demo. The autonomous vehicle client would still need a different approach, as it is not a human actor but a device. In the demo scenario, the car's DID document and private keys are stored in a file on the (simulated) device. In a real-world case, however, storing the highly sensitive private keys on a filesystem might be too risky in the case that an attacker has physical access to the vehicle. Thus, some hardware-protected storage of the secrets would be preferred. Some hardware modules were identified that support the EdDSA/Ed25519 cryptography used by IOTA Identity, but these were not experimented with in practice. The identified modules include the Microchip CEC1702 [27], the USB-pluggable YubiHSM 2 [28], and SC-4HSM [29].

In principle, a network of trust with verifiable credentials can be formed based on the fact that each credential has an issuer, who in turn has credentials that can be cryptographically verified up the trust chain as necessary. The IOTA Integration Services has only one (cryptographic) issuer for verifiable credentials, which is the root identity of the IS instance. Credential hierarchy is possible in IS, however, in the form of an initiator field that conations the DID of the identity that initiated issuing the credential. The initiator, in turn, needs to possess a certain type of credential ('VerifiedIdentityCredential') to be able to issue credentials to other users. In practice, the initiator can thus be considered as the logical issuer of the credential. For the functionality of the demo, it makes no difference whether the cryptographic issuer of the credential is also the logical issuer, but the W3C Verifiable Credentials data model does not define such an initiator. Should there be a need for the logical issuer of each credential to also be the cryptographic issuer, a separate instance of the IS service would have to be run for each credential issuer. This would not be practical.

The IS instance holds each issued verifiable credential in its database. Any authenticated user can query the credentials of all other users registered on the same IS instance. Clearly, in a real-life scenario, this would raise privacy concerns, as the credentials could contain very private data. Again, in our demo scenario, this is not an issue, and it is possible to construct the credentials so that only those required to build the necessary level of trust between participants are stored on the IS. Nevertheless, this is a shortcoming in the current IS version. In credentials handling, as specified by W3C, each participant—the credentials holder—stores their own credentials, and credentials are presented only when requested by another participant, i.e., the verifier. The credentials should be presented as a W3C Verifiable Presentation (VP) that is constructed explicitly when requested, with cryptographic proof provided by the holder. The verifying party could then ascertain that the VP was produced specifically for the request by the holder of the credentials, and then request verification of the individual credentials of the VP from the issuer of those credentials. Cryptographic verification of a credential is possible even without the active participation of the issuer by using the issuer's public key. However, to make credential revocation possible, the issuer needs to store some information about the credentials it has issued. The IS seems to use its credentials database for revocation purposes, as the credential is deleted from the database when it is revoked, and subsequent verification requests for the credential fail. The IS REST API contains a method for verifying a VP, but no methods for creating one, so it does not fully support the Verifiable Presentations model.

In order to create hierarchies of trust, the IOTA Integration Service supports setting other IS instances as trusted roots. In practice, the root DID of the other instance is set as trusted, after which the first instance will report all cryptographically correct credentials issued by the other instance as valid. The addition of trusted roots would make it possible to manually create trust relationships between organisations that are running their own IS instances. In such a case, verifiable credentials issued by other IS instances would have to be transferred via some out-of-band method, as they could not be simply queried from the

(local) IS. This would also protect the credentials issued by an IS instance from queries by users of other instances. Revoked credentials, however, would be more difficult to detect because the IS instances do not communicate with one another. A credential issued in one instance is always considered verified in the other if the credential is cryptographically valid and the DID of its issuer is set as a trusted root.

In a restricted scenario, such as the remote driving case presented here, using trusted roots could be a viable option for establishing a network of trust between a limited set of participating organisations. In a general case, however, the concept of identities and credentials is much broader than we have covered in our demo scenario. In a real-life situation with different types of remote drive-capable vehicles from multiple manufacturers, various remote-driving companies—perhaps operating in several countries—and with different authorities involved, the range of required and potentially acceptable credentials quickly becomes huge. Fundamentally, the problem boils down to which credentials are required by each participant to allow an operation to take place, which kinds of variants of the credentials they are willing to accept, and which organisations they trust as issuers of those credentials. In this demo, the credentials and their issuers are predefined. A more real-life approach would be to use verifiable data registries, as suggested in the W3C Verifiable Credentials data model. The participants could use such registries to determine, for example, which credentials they can accept and from which issuers, and to check them automatically. However, such data registries are still very much under development and will not be covered here. As examples, the reader may refer to Hyperledger Aries [30] for an approach to managing credentials and the Sovrin Governance Framework [31] for identity management.

4.6. Discussion of the Implementation

Our solution makes use of the IOTA Identity [32] component as a W3C-compliant DID implementation, and the Channels application of the IOTA Streams [33] component for an immutable messaging channel with optionally encrypted content. The Channels application provides Ed22519-based signature and X25519-based encryption schemes but does not handle authentication or the association between data and its source [34]. These are up to the application. The Channels application supports both plaintext and encrypted message parts, but in our case, we only use fully encrypted messages. We wanted to use DIDs as the authentication method for channel access, so we chose to use the experimental IOTA Integration Services' [35] (hereinafter, "IS") microservice that provides APIs for both Identity and Channels services and, as the name implies, integration between the two. Our demo was set up on one computer running the Windows 10 operating system with a Minikube Kubernetes cluster running on a virtual machine. The setup was tested with both HyperV and VirtualBox virtual machines, which functioned identically. IOTA Integration Services were accessed using the IOTA-provided Node SDK and REST API. We used the IOTA Mainnet ledger using IOTA's Chrysalis network.

The IS consists of two subcomponents—the SSI Bridge [36] provides a wrapper for Identity operations, and the Audit Trail Gateway [37] wraps the Channels application. The two services are linked in IS so that authentication and message source stamping in Channels messages are performed using DIDs provided by the SSI Bridge. The IS has some local bookkeeping for the provisioning of services, such as a verifiable credentials registry for identities. It also caches ledger content (e.g., identities, messages) to speed up ledger read operations. Communicating with the IOTA IS requires a service instance-specific API key.

In our scenario, each participant has a DID document that was created using IS with a root account. The DID documents are rooted in the IOTA ledger and are publicly available. Each participant stores their document and its corresponding private signing and encryption keys in a file. Authentication to the service uses the IS-provided challenge-response method paired with the DID as input. As a response to successful authentication, each client is given a JSON Web Token (JWT) that they use as the authorisation token for

all subsequent requests. The IS can then map each request to the requestor's DID using the token.

All verifiable credentials are created using the IS and are stored primarily in the IS local database. The clients do not have locally maintained copies of their or their counterparts' credentials but rather request them from the IS as needed. Cryptographic verification of credentials is also provided by the IS API. Checking the content of the credentials is, however, entirely up to the participant acting as the verifier. In our case, the contents of each required credential for every participant are fixed design-time and checked by the supervisor before granting channel access. We found that the IOTA IS does not satisfactorily support the W3C-preferred way of exchanging verifiable credentials, which is to bundle them into a verifiable presentation. Therefore, in our setup, we implemented an option to request and present individual VCs as part of control channel messaging. However, this option was not extensively used as it is not the W3C preferred way, and it would complicate the control messaging compared with the chosen approach of using a predefined list of credentials to check for each participant.

The clients use the IS channel keyword search with a known topic keyword to find the address of the common control channel. After that, they request a subscription to the channel. The supervisor client periodically lists all subscription statuses in the UI, along with the credentials of each active and requested subscriber. The human user of the UI can authorise or revoke the subscribers manually after evaluating the credentials.

In the scenario, messaging on the channel is mostly based on status notifications that each subscribed client sends independently of others. The status messages include, among other data, the client's online status, point-to-point connection status, location, and current driving mode. Each client periodically polls the channel for new messages. The clients interpret the status messages as they see fit—for example, a remote driver client lists potentially remote drivable vehicles based on their reported availability. Extensive use of notification-style status messages is partially due to the potentially lengthy verification of the corresponding transactions to the ledger. Currently, with the IOTA network, message verification typically takes around ten seconds, which makes a request–response cycle quite lengthy. The only request–response message pair used in the demo is the exchange of remote drive point-to-point connection parameters. Both the request and response include their recipient's DID, as well as the connection parameters that are encrypted with the recipient's public encryption key. The encryption key is obtained from the IS by requesting the DID document of the recipient DID.

The Integration Services wrapper for Channels simplifies messaging over the IOTA ledger. Reading messages using the wrapper is based on polling, and there are no read cursors to determine which messages were sent to which recipient. Thus, each client needs to periodically read the channel for messages from a period starting a bit before their last read and filter out all messages that were sent by or have already been handled by that client. From the remaining messages, the client needs to determine those it needs to process. In our case, status notifications do not have a named recipient, so each client needs to consider their significance based on the message sender and content. For request–response messages, we added a recipient DID.

In our simulated environment, direct messages between the remote driver and vehicle are transferred through a WebSocket connection that is hosted by the vehicle client. The address of this server is sent encrypted to the driver client during the point-to-point connection exchange procedure. The only information transferred over the direct connection in the demo are indications of driving mode changes and indications of either party going offline. These indications act as examples of point-to-point events that trigger events to the immutable common control channel log. Both the driver and vehicle clients timestamp the events and report the changed status independently to the common control channel. In real life, the point-to-point connection could use multiple protocols and any type of content; WebSocket is used here to demonstrate a simple real-time connection.

5. Evaluation Results

The results are discussed in this section from three perspectives. First, a technical evaluation of the performed IOTA-based experiment is given. Next, the experiment is evaluated against the digital trust concepts discussed in Section 3. Finally, the experiment is evaluated with respect to the challenges of securing remote driving presented in Section 2.2.

5.1. Evaluation of the IOTA-Based Experimental Solution

In our evaluation scenario, we used IOTA Integration Services (IS), which combines the IOTA Identity and IOTA Streams services. The combination proved to be sufficient to demonstrate the key components of our trust solutions. With Integration Services, we could create W3C DID-based users and achieve authentication and trust establishment using W3C-verified credentials. Using IS, we were also able to create an encrypted common control channel with restricted access controlled by a supervisor. Channel messages were visible to subscribers of the channel, but partial encryption of message content was still possible to establish private data exchange between two subscribers. Using private data exchange, we were able to pass point-to-point connection parameters between two endpoints so that the necessary out-of-band communication channel required by real-time remote driving could be established, but only the endpoints would have access to it.

The association between messages and their authenticated sender was achieved with IOTA Streams, as was message storage in a public, immutable ledger. Public accessibility and immutability of messages in the IOTA ledger make it possible to audit the entire message trail later with a predefined auditor key, but we did not experiment with this feature in our demo scenario.

One limitation we found with IOTA Identity was that it did not support digital wallets as storage for identities and verifiable credentials. Therefore, we had to use a file-based approach for storing and accessing the private and public keys needed in authentication, which is not optimally secure as access to the file is protected only by the file system of each client. A wallet could have, e.g., a password, biometric, or two-factor authentication for access and would be preferable in a real-life scenario.

Another shortcoming with Identity and/or its Integration Services wrapper was its lack of support for the W3C-preferred way of using verifiable presentations to exchange the verifiable credentials needed for authorisation. Therefore, we opted for a simplified approach where the needed credentials are predefined and checked by the supervisor prior to authorising the credentials holder to the channel. This approach is less flexible than the verifiable presentation method, but enough to prove the concept of using verifiable credentials-based authorisation to securely establish a point-to-point remote driving connection.

The IOTA Channels application was found to be sufficient for our case, although its lack of support for determining which messages have already been sent to which client made implementation unnecessarily complex. IOTA Channels application was found to facilitate authenticated messaging with access control, immutable tracing of messages, and the association between each message and its sender. The message validation time, i.e., the time it takes for the message to be added to the ledger, was found to vary, typically around ten seconds. This is obviously too long for any real-time use, but it suits the connection establishment and message trace purposes of our demo scenario. The long message round-trip time—further increased by the periodic polling for new messages—makes request–response messaging especially long. Long delays mean that it is not possible to establish a precise timeline of events based on the timestamps set by the system onto the ledger messages. To mitigate this, we added a sender timestamp to the messages. However, relying on each sender to provide such metadata brings problems of its own; for example, each sender will have some clock offset. Also, a dishonest sender could manipulate the metadata before sending, if the data indicates liability for the event being reported. Figure 15 shows a histogram of validation times for 1000 ledger messages using a local demo setup. The average validation time is 11.0 s, with a median of 9.3 s. The

validation took less than 5 s for 10.1% of the messages and more than 20 s for 9.0% of the messages.

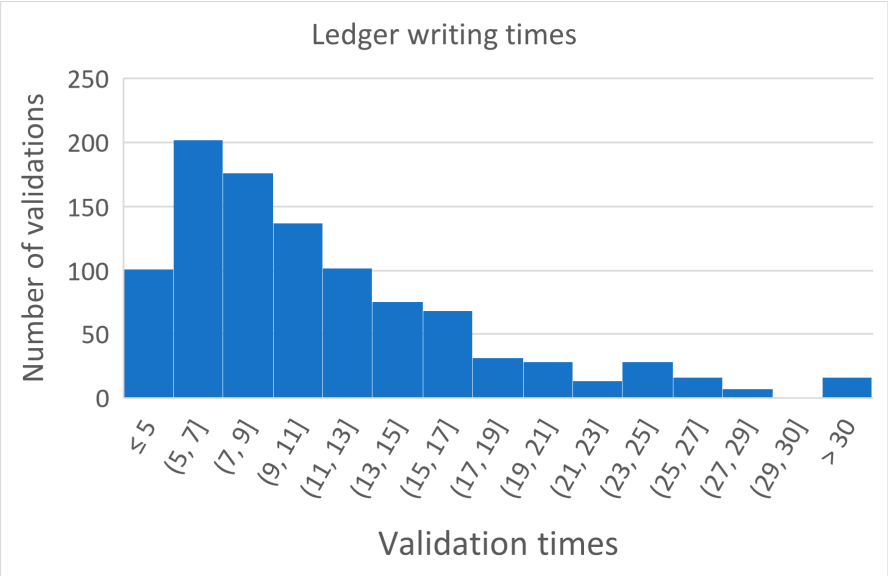

Figure 15. Message validation times.

Posting data on the IOTA ledger is free of charge in the sense that there are no transaction fees. However, permanent storage of data obviously incurs costs somewhere, so a real-world solution cannot rely on the storage being totally free of cost. The IOTA ledger prunes old transactions from the node network but provides the option of setting up special client-hosted network nodes for long-time transaction storage. A real-world solution for the presented scenario would likely have to host such a long-term storage node.

Another issue is that the more transactions there are in the node network, the more processing nodes are needed to approve those transactions. IOTA anticipates the use of incentives to encourage clients to host their own nodes in the future [38], which is something that a real-life solution would also need to consider.

5.2. Evaluation of the Digital Trust Concepts

The conceptual approach and provided concepts focus on the trust establishment phase and the related user identification and attribute verification. The contributions are quite well in line with Zero Trust architectures; however, there are some differing aspects in our approach. Zero Trust can be seen as a collection of cybersecurity paradigms that aim to shift authentication away from static, network-based perimeter defences to ones focusing more on users, assets, and resources [3]. In other words, authorisation to a certain service or resource is not tied to a specific user or their role, but rather to a context-aware set of dynamic rules for granting or revoking access. In our approach, trust and related access control is at the centre. While many access control paradigms exist, a few of them are of specific interest from the Zero Trust point of view, as listed in [39]. Policy-based, attribute-based, and function-based access control methods each determine the access based on an evaluation of the request, the requestor, and other dynamic parameters. Of these methods, our approach bears the most similarity to attribute-based access control *(ABAC)* [40], as channel access is based mostly on the attributes of the requestor presented in the form of verifiable credentials. The attributes may also verify the requestor's identity. A supervisor entity grants control to the security control channel based on credentials. Our approach makes it possible to use even finer-grained access control based on ABAC, as other stakeholders' credentials can be requested and mutually verified by each participant on the security control channel without a need for the supervisor to act as a mediator. However, our first implementation in the IOTA-based experimental solution relies solely

on verifying the attributes when joining the security control channel, which fulfils the lightweight mutual authentication requirement highlighted in [39]. Also, as access to the security control channel is effectively dependent on the verified identities of each member before granting access, it can be seen as a form of a software defined perimeter, an overlay network for securing resource access, as defined in [3].

The traceability of certain control events on the security control channel is a key feature of our approach. An immutable log of control events could be needed, for example, by authorities to determine the legal responsibility of each participant in case of an accident. In general, Zero-Trust architectures typically provide network and access logs but usually not in a way intended for public, immutable reference [39]. While ledger-based logs can be seen as a problem due to their inherent transparency [39], in our case, they are essential. Any potential harm from the public visibility of the logs is handled by our use of a message-encrypting library (IOTA Streams) that limits visibility to security control channel members and predefined auditors, and the possibility to use point-to-point encryption among members to mask certain parts of messages from other channel members.

Blockchain-based logging is resource-intensive, as [39] points out. Even though in our case we do not use a blockchain, but rather the IOTA DAG, the associated consensus mechanism limits messaging throughput. However, since the ledger usage is limited to security control and critical event logging, we find that the benefits outweigh the disadvantages.

5.3. Evaluation of Challenges against Securing Remote Driving

The remote driving case has several required trust relationships: the system has 13 identified types of actors (mission planner, ground crew member, road user, remote driver, backup remote driver, in-vehicle safety supervisor, IT/Traffic security manager, passenger, autonomous vehicle, traffic assets (signs, lights, cameras), provider/services provider, city traffic service provider, authorities). The required trust relationships in a real remote driving case are therefore much more complicated than what was evaluated in the presented scenario. In addition, we assumed that remote driving in an urban environment requires some supervisory stakeholders to be included. It could be, e.g., the traffic security manager of a city or suburb. The evaluations indicate that the basic approach works; however, there seem to be missing solutions for shared (trust) contract(s) between multiple stakeholders.

The process related to the use of credentials for proving trust is applied in the experimental solution so that the supervisor acts as a kind of verifier, and the issuers are simulated. The supervisor requires certain verifiable credentials from the vehicle and the remote driver before these entities are accepted into the IOTA channel. The verification process seemed to work sufficiently in remote driving; however, the number of entities and required credentials was limited, and the final acceptance into the IOTA channel was performed manually by the supervisor. In addition, the lack of a wallet application in the experiment was somewhat limiting. In a real remote driving case, the number of entities and credentials will increase heavily and, therefore, the scalability and autonomous operations need to be studied and experimented more.

The separation of control and data planes from each other was found to be a good solution because messaging via IOTA channels has quite high delays, as Figure 15 indicates; therefore, they cannot be used to exchange real end-to-end data flows related to remote driving. On the other hand, it was estimated that slightly more delay in the security control process may be acceptable in the start-up phase for the users. In addition, the requirements for security in exchanging critical parameters are very strict. Therefore, the decision to use the encrypted and access-controlled IOTA channel for security control and for securely exchanging the parameters of an out-of-band data plane connection proved to be a reasonable solution. In this way, any additional delays and overheads in the actual end-to-end data flow between the autonomous vehicle and remote driving system were not caused in the simulation-based experiment.

The application of IOTA Tangle as the verifiable data registry seems quite reasonable based on the experiment. This is because remote driving in an urban ecosystem has a very high number of transactions, especially because there are very many stakeholders and assets that need to have verifiable credentials, and because of the need to record security traces. It was estimated that the applied DAG is scalable enough for the remote driving case based on the use of the IOTA test network. However, there is a need to continue and proceed towards the application of the public IOTA.

The challenges for securing remote driving were analysed in Section 2. An evaluation of the results against these challenges is briefly discussed below.

- Requirement (R1): The solutions contribute towards ensuring that the source/sender of the mission plan is correct and that the plan has not been modified. This helps to ensure that the autonomous vehicle is not misused.
- R2: The source of the mission plan can be verified, which helps to prevent the use of an autonomous vehicle for malicious purposes.
- R3: The solutions contribute towards making autonomous driving safe.
- R4: The solutions can be used to verify the credentials of entities, which send information on the presence, location, and mobility of humans/animals/artificial entities on the road. Therefore, the trust level related to the referred information is improved, and the likelihood of the system suffering from fraudulent information is lower.
- R5: The sources of location information can be ensured, and they can be visualised on the dashboard of the supervisor. This is seen to improve the trust and safety level of the operation.
- R6: The application of IOTA for security trace can help in emergency reasoning because information concerning the vehicle and its surrounding situation can be stored on the occurrence of a critical event in a way that prevents its later manipulation.
- R7: The solutions can be used to ensure that the remote driver obtains information on the status of the vehicle from the correct and real vehicle.
- R8 and R10: The solutions do not cause additional delays or overheads for the e2e data flow between the autonomous vehicle and the remote driving system. The delays and overhead are estimated to stay the same, but they depend on the applied encryption/decryption between the vehicle and the remote driving system.
- R9: The solutions can be used to verify the credentials of the remote driver; therefore, the vehicle can better trust the information received from the remote driver.
- R11. The solutions do not rely on username/password systems only but always verify the actors, endpoints, and related credentials before allowing any real actions. This is expected to improve the system reliability and confidentiality, without relying too much on network-level security and usernames/passwords.
- R12: The solutions focus on the security control process, contribute towards application of PKI-based solutions, and rely on the distributed ledger as the shared trust register.
- R13: The application of IOTA for the security trace is estimated to help in enabling analysis of reasons for problems, dangerous situations, or even accidents. This is expected to contribute towards improving system safety in the future.

The division of the security-, privacy-, and trust-required solutions to trust, credentials, control data, and trust storage levels turned out to be a reasonable selection. It helps to divide the very complex real-world system needed for solving the digital trust puzzle into different abstraction levels. The critical parts of these systems are related to identities and relationships of between people, organisations, and physical assets, especially because of the dynamic nature of trust relationships. When changes occur at that abstraction level, the result should be very rapidly visible in the real world where physical assets interact with each other. In this research, the resulting conceptual solutions were experimentally developed by relying on digital identities, PKI with a decentralised approach using decentralised identifiers (DIDs) and verifiable credentials (VCs), and an IOTA-based distributed ledger. Based on the evaluations, this approach looks feasible, and work towards integrating the provided digital trust solutions with a real autonomous vehicle, traffic infrastructure, and

remote-control system is ongoing to also validate the performance, reliability, scalability, and flexibility of the provided conceptual solutions.

6. Concluding Remarks

The key contribution of this research is related to the application of the Zero Trust type of approach as the starting point for reducing the risk of attacks coming inside the perimeter network. In the proposed solution, the digital trust of the actors is always verified first before they are allowed to remotely operate autonomous vehicles. The concepts of digital trust are represented as a layered model by dividing the system into trust, credentials, control data, and trust storage abstraction levels. The conceptual solutions are studied and described, and respective experimental solutions were developed, relying on digital identities, public key cryptography with a decentralised approach using decentralised identifiers (DIDs) and verifiable credentials (VCs), and an IOTA-based distributed ledger. The provided digital trust solutions were validated by executing them according to the remote driving scenario but with a simulated vehicle and simulated remote driving system. The hybrid simulation mainly focused on the validation of functional, causal temporal correctness, feasibility, and capabilities of the provided solutions.

The evaluations indicate that the concepts of digital trust fulfil the purpose and contribute towards making remote driving more trustable. However, a real-world remote driving case has much more required verifiable trust relationships than those validated in the experiments. In addition, remote driving in an urban environment was assumed to require a supervisory stakeholder, such as a traffic security manager of a city or suburb, to be included. In the solution, the supervisor acted as a kind of verifier, requiring a set of example verifiable credentials from the vehicle and the remote driver, and accepting them manually in the security control channel. The verification process fulfilled its purpose and worked in a sufficient way. However, the number of entities and required credentials were quite limited compared with the real remote driving case. The separation of control and data planes from each other was found to be a good solution because delays caused by required security control can be limited to the initiation of the remote driving session without causing additional delays in the actual real-time remote driving control data flow. The application of the IOTA Tangle as the verifiable data registry was found to be sufficient for security control purposes.

During the evaluations, clear needs for further studies related to scalability, application of wallets, dynamic trust situations, time-sensitive behaviour, and autonomous operations, as well as smart contract(s) among multiple stakeholders, were detected. Studies targeted at more detailed solutions for these areas are ongoing, and the work towards integrating the provided digital trust solutions with a real autonomous vehicle, traffic infrastructure, and remote-control system is ongoing to validate the performance, reliability, scalability, and flexibility of the provided conceptual digital trust solutions.

Author Contributions: J.L. contributed to the concept development and evaluation and acted as the main editor. V.K. and J.R. contributed to the prior art descriptions, experimental solutions description and development, and evaluation parts. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the H2020 ECSEL Joint Undertaking project TRANSACT, Business Finland and VTT Technical Research Centre of Finland.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors would like to thank all the colleaques and the anonymous reviewers for their comments, which have greatly improved the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. EU Transact Project. Available online: https://transact-ecsel.eu/ (accessed on 16 December 2022).
2. Latvakoski, J.; Mäki, K.; Ronkainen, J.; Julku, J.; Koivusaari, J. Simulation-Based Approach for Studying the Balancing of Local Smart Grids with Electric Vehicle Batteries. *Systems* **2015**, *3*, 81–108. [CrossRef]
3. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; 2020 NIST Special Publication 800-207; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2020. [CrossRef]
4. The Law Commission (UK). Remote Driving: Advice to Government SUMMARY. 2023. Available online: https://s3-eu-west-2.amazonaws.com/lawcom-prod-storage-11jsxou24uy7q/uploads/2023/02/RD-Summary-for-20-02-23.pdf (accessed on 1 August 2023).
5. World Wide Web Consortium (W3C). Decentralized Identifiers (DIDs) v1.0—Core Architecture, Data Model, and Representations. W3C Proposed Recommendation 3 August 2021. 2021. Available online: https://www.w3.org/TR/did-core/ (accessed on 28 April 2022).
6. Sovrin. Self-Sovereign Identity and IoT. Sovrin Foundation SSI in IoT Task Force. 2020. Available online: https://sovrin.org/wp-content/uploads/SSI-in-IoT-whitepaper_Sovrin-design.pdf (accessed on 28 April 2022).
7. Kronfellner, B.; Merey, T.; Beron, D.; Terbu, O. Me, Myself and (SS)I. Boston Consulting Group. 2021. Available online: https://web-assets.bcg.com/6b/6d/84e00cad4c939c870d833b96321c/white-paper-me-myself-ssi.pdf (accessed on 28 April 2022).
8. Housley, R. Public Key Infrastructure (PKI). Available online: https://onlinelibrary.wiley.com/doi/10.1002/047148296X.tie149 (accessed on 2 January 2023).
9. Shi, J.; Zeng, X.; Han, R. A Blockchain-Based Decentralized Public Key Infrastructure for Information-Centric Networks. *Information* **2022**, *13*, 264. [CrossRef]
10. Trust over IP Foundation. Introduction to Trust over IP. Version 2.0. 17 November 2021. Available online: https://trustoverip.org/wp-content/uploads/Introduction-to-ToIP-V2.0-2021-11-17.pdf (accessed on 22 November 2022).
11. Liu, X.; Farahani, B.; Firouzi, F. *Distributed Ledger Technology. Intelligent Internet of Things*; Springer: Cham, Switzerland, 2020; pp. 393–431, ISBN 978-3-030-30367-9. [CrossRef]
12. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 22 November 2022).
13. IOTA Foundation. IOTA Overview. 2022. Available online: https://wiki.IOTA.org/learn/about-IOTA/an-introduction-to-IOTA (accessed on 23 November 2022).
14. Green, M. Hash-Based Signatures: An Illustrated Primer. Available online: https://blog.cryptographyengineering.com/2018/04/07/hash-based-signatures-an-illustrated-primer/ (accessed on 23 November 2022).
15. Silvano, W.F.; Marcelino, R. IOTA Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Gener. Comput. Syst.* **2020**, *112*, 307–319. [CrossRef]
16. Cech, J. Chrysalis (IOTA 1.5) Phase 2. Update and Next Steps. Available online: https://blog.IOTA.org/chrysalis-IOTA-1-5-phase-2-update-and-next-steps-eecabe55d7bd/amp/ (accessed on 23 November 2022).
17. Alladi, T.; Chamola, V.; Sahu, N.; Venkatesh, V.; Goyal, A.; Guizani, M. A Comprehensive Survey on the Applications of Blockchain for Securing Vehicular Networks. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1212–1239. [CrossRef]
18. Fang, L.; Wu, C.; Kang, Y.; Ou, W.; Zhou, D.; Ye, J. Zero-Trust-Based Protection Scheme for Users in Internet of Vehicles. *Secur. Commun. Netw.* **2022**, *2022*, 9896689. [CrossRef]
19. Dorri, M.; Steger, S.; Kanhere, S.; Jurdak, R. BlockChain: A Distributed Solution to Automotive Security and Privacy. *IEEE Commun. Mag.* **2017**, *55*, 119–125. [CrossRef]
20. den Ouden, J.; Ho, V.; van der Smagt, T.; Kakes, G.; Rommel, S.; Passchier, I.; Juza, J.; Tafur, M. Design and Evaluation of Remote Driving Architecture on 4G and 5G Mobile Networks. *Front. Future Transp.* **2022**, *2*, 801567. [CrossRef]
21. Saez-Perez, J.; Wang, Q.; Alcaraz-Calero, J.M.; Garcia-Rodriguez, J. Design, Implementation, and Empirical Validation of a Framework for Remote Car Driving Using a Commercial Mobile Network. *Sensors* **2023**, *23*, 1671. [CrossRef] [PubMed]
22. Neumeier, S.; Walelgne, E.A.; Bajpai, V.; Ott, J.; Facchi, C. Measuring the Feasibility of Teleoperated Driving in Mobile Networks. In Proceedings of the 2019 Network Traffic Measurement and Analysis Conference (TMA), Paris, France, 19–21 June 2019; pp. 113–120. [CrossRef]
23. Amador Molina, O.; Aramrattana, M.; Vinel, A. A Survey on Remote Operation of Road Vehicles. *IEEE Access* **2022**, *10*, 130135–130154. [CrossRef]
24. MetaMask. A Crypto Wallet & Gateway to Blockchain Apps. Available online: https://metamask.io/ (accessed on 9 January 2023).
25. IOTA Foundation. Successful Conclusion of ENSURESEC Part 2. Available online: https://blog.iota.org/conclusion-of-ensuresec-part-2/ (accessed on 9 January 2023).
26. IOTA Foundation. IOTA Smart Contracts Beta Release. Available online: https://blog.iota.org/iota-smart-contracts-beta-release/ (accessed on 9 January 2023).
27. Microchip Technology Inc. CEC 1702 Cryptographic Embedded Controller Data Sheet. Available online: http://ww1.microchip.com/downloads/en/DeviceDoc/00002207C.pdf (accessed on 9 January 2023).
28. Yubico. YubiHSM 2 Hardware Security Module. Available online: https://www.yubico.com/fi/product/yubihsm-2/ (accessed on 9 January 2023).
29. Spark Innovations Inc. Introducing the SC4-HSM. Available online: https://sc4.us/hsm/ (accessed on 9 January 2023).

30. Hyperledger Foundation. Hyperledger Aries. Available online: https://www.hyperledger.org/use/aries (accessed on 9 January 2023).
31. Sovrin Foundation. Sovrin Governance Framework. Available online: https://sovrin.org/library/sovrin-governance-framework/ (accessed on 9 January 2023).
32. IOTA Foundation. Unifying Digital Identities. Available online: https://www.iota.org/solutions/digital-identity (accessed on 9 January 2023).
33. IOTA Foundation. IOTA Streams. Available online: https://www.iota.org/solutions/streams (accessed on 9 January 2023).
34. IOTA Foundation. Streams Specification, Rev:1.0 A, p. 20. Available online: https://github.com/iotaledger/streams/blob/develop/specification/Streams_Specification_1_0A.pdf (accessed on 9 January 2023).
35. IOTA Foundation. Integration Services Wiki. Available online: https://web.archive.org/web/20230604211029/https://wiki.iota.org/integration-services/welcome/ (accessed on 1 August 2023).
36. IOTA Foundation. SSI Bridge. Available online: https://web.archive.org/web/20230604203630/https://wiki.iota.org/integration-services/explanations/services/SSI-bridge/introduction/ (accessed on 1 August 2023).
37. IOTA Foundation. Ecommerce-Audit Trail Gateway (GW). Available online: https://web.archive.org/web/20230528153958/https://wiki.iota.org/integration-services/explanations/services/audit-trail-gateway/introduction (accessed on 1 August 2023).
38. IOTA Foundation. Incentives to Run an IOTA Node. Available online: https://blog.iota.org/incentives-to-run-an-iota-node/ (accessed on 9 January 2023).
39. Syed, N.F.; Shah, S.W.; Shaghaghi, A.; Anwar, A.; Baig, Z.; Doss, R. Zero Trust Architecture (ZTA): A Comprehensive Survey. *IEEE Access* **2022**, *10*, 57143–57179. [CrossRef]
40. Hu, V.; Farraiolo, D.; Kuhn, R.; Schnitzer, A.; Sandlin, K.; Miller, R.; Scarfone, K. *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*; Technical Report NIST 800-162; NIST: Gaithersburg, MD, USA, 2014. Available online: https://csrc.nist.gov/publications/detail/sp/800-162/final (accessed on 1 August 2023).