A privacy-preserving approach to grid balancing using scheduled electric vehicle charging

Antonio Antonino

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 27.9.2019

Supervisor

Prof. Antti Ylä-Jääski

Advisors

Yki Kortesniemi, D.Sc. (Tech.)

Dmitrij Lagutin, D.Sc. (Tech.)

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Antonio Antonino

Title A privacy-preserving approach to grid balancing using scheduled electric vehicle charging

Degree programme ICT Innovation

Major Cloud Computing and Services **Code of major** SCI3081

Supervisor Prof. Antti Ylä-Jääski

Advisors Yki Kortesniemi, D.Sc. (Tech.), Dmitrij Lagutin, D.Sc. (Tech.)

Date 27.9.2019 **Number of pages** 81+3 **Language** English

Abstract

The introduction of renewable energy generation (e.g. solar and wind) in the energy distribution infrastructure makes balancing the total energy load and production in the grid more challenging due to the weather-dependent nature of these energy sources. One approach to mitigate the issue is to use weather forecasts to predict the production and then offer incentives to electric vehicle users (EVUs) to charge their vehicles during the times of energy surplus. However, doing this without leaking sensitive information about the EVUs location and identity presents challenges to the system design.

This thesis proposes a privacy-preserving architecture that allows the grid operator to offer incentives for contributing to the grid stability, and to reliably and automatically quantify the extent of each contribution while still maintaining the privacy of the EVUs. Furthermore, the architecture enables decentralised privacy-preserving dispute resolution without leaking any personally identifiable information (PII).

The architecture fulfils the goal by utilising self-sovereign identity technologies, such as decentralised identifiers (DIDs), and privacy-preserving digital credentials solutions, such as verifiable credentials (VCs). They allow the solution to utilise ephemeral identifiers and to compartmentalise the information into three different knowledge domains to ensure that only the minimum amount of information needed crosses any domain border.

An analysis of the solution indicates that the architecture ensures relatively strong privacy guarantees to the EVUs and solves the grid balancing problem while reducing the number of assumptions to the minimum. This makes the architecture applicable to a wide set of use cases in the EV charging field. Future work includes a detailed performance analysis of a proof-of-concept (PoC), although the information available from related research already indicates relatively low latency and a good level of deployability even on resource-constrained Internet-of-things (IoT) devices.

Keywords Privacy, Grid Balancing, Electric Vehicle (EV), Vehicle-to-Grid (V2G), Verifiable Credential (VC), Decentralised Identifier (DID), Self-Sovereign Identity (SSI)

Preface

I would like to say that I am very proud of the work presented in this thesis. Nevertheless, I am extremely aware that such work might not have been possible without the help and the support that I have received.

First and foremost, I would like to thank my two advisors, Yki Kortesniemi and Dmitrij Lagutin, for the large and constant support they have given me throughout the whole process of this thesis work, from the definition of its scope, through the design of the architecture, to the review of the thesis document. At the same time, I would like to thank my supervisor, Prof. Antti Ylä-Jääski, for giving me clear guidance and freedom to work on a topic that I feel particularly concerned about.

I would also like to thank Ericsson, my employer, for allowing me to work on the thesis during regular working time, without sacrificing too much of my spare time and allowing me to finish within the deadlines set. I would especially like to thank my project manager, Mikael Jaatinen, and my line managers, Lauri Salmio and Adam Peltoniemi, for understanding my situation and for offering me all the help I have ever needed since I started working on this thesis.

Last but not least, I would like to thank EIT Digital Master School for the amazing opportunity that they have offered me. I am very happy to have chosen this double-degree programme, during which I have learnt a lot, travelled a lot, and met many interesting people from all over the world.

Living away from home is not always easy. I left Italy to live abroad four years ago, in September 2015. Nevertheless, my parents have given me constant support and love for the whole time, and I want to thank them for this. The same I can say for my friends, both the ones that I have left home, because I know I can always count on them, and the ones that I have met during these four years around Europe. Without them, I could not have found that balance that is extremely important for one's well-being.

A deep and warm thank you, everyone. May this be only yet another wonderful checkpoint in the journey called life. Off to the next one!

Antonio Antonino

Contents

Abbreviations

BFT	Byzantine Fault Tolerant
CA	Certificate Authority
CS	Charging Station
CSO	Charging Station Owner
DApp	Decentralised Application
DID	Decentralised Identifier
DLT	Distributed Ledger Technology
DNS	Domain Name System
DoS	Denial-of-Service
DSO	Distribution System Operator
ER	Energy Retailer
EV	Electric Vehicle
EVU	Electric Vehicle User
FM	Fleet Manager
GDPR	General Data Protection Regulation
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet-of-Things
JSON	Javascript Object Notation
JWT	JSON Web Tokens
MitM	Man-in-the-Middle
PKI	Public Key Infrastructure
PII	Personally Identifiable Information
PoC	Proof-of-Concept
PoLP	Principle of Least Privilege
PoW	Proof-of-Work
PoS	Proof-of-Stake
RBFT	Redundant Byzantine Fault Tolerant
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SOFIE	Secure Open Federation for Internet Everywhere
SSI	Self-Sovereign Identity
TSO	Transmission System Operator
URI	Uniform Resource Identifier
V2G	Vehicle-to-Grid
VC	Verifiable Credential
ZKP	Zero-Knowledge Proof

1 Introduction

This section introduces the challenges of grid balancing, describes the scenario considered in this work, and presents the problem statement and the research questions to be answered in the rest of the thesis.

1.1 The challenges of grid balancing

The energy grid infrastructure handles the generation and distribution of electrical energy across well-delimited geographical areas, e.g. within a national border. The long-range, high-voltage infrastructure transmitting energy from production plants to municipalities and few large customers is managed by a transmission system operator (TSO) [22], while the grid delivering the rest of the energy to end customers, e.g. houses and small businesses, is managed by a distribution system operator (DSO). In some countries, the two roles are performed by the same actor.

Operating and maintaining a grid infrastructure brings several challenges to the DSO [18], including optimising the efficiency of the distribution grid, and avoiding reverse power flows. *Reverse power flows* indicate the processes in which the energy in the grid moves back towards the generation sites. Such reverse flows can generate significant issues for the DSO due to the design of the energy grids, which are only meant to handle unidirectional electricity flows from the generation sites to the points of consumption.

To mitigate such issues, the DSO typically divides its coverage area into different *energy districts*. The DSO can then deploy additional power generation facilities within each energy district, typically producing energy from renewable sources [2], as shown in Fig. 1. This is done to both distribute part of the energy generation process and to reduce the dependency of the DSO on the main transmission infrastructure and the TSO. Nevertheless, since the sources of energy in these smaller generation facilities are typically renewable, the amount of energy generated to the grid cannot be fully predicted as opposed to traditional power plants. For instance, a local plant producing solar energy will be able to produce more energy during the daytime, while a wind plant is more productive when there is a strong presence of wind. However most of the peaks and lows in the energy generation can be forecasted with an acceptable level of accuracy days before, on a per-district level.

The forecasted excess of energy can be mitigated using different approaches including reducing the amount of energy produced locally, temporarily storing the excess energy in facilities specifically built for the purpose, or by allowing users other than the traditional customers to access the energy services, thus taking out more energy from the grid and reducing the probability of reverse power flows. Due to their increasing adoption [26][67] and their high mobility, electric vehicles (EV) represent a convenient means for a DSO to level out peaks in the energy grid. For instance, users of electric vehicles (EVUs) can be incentivised to charge during peak times, to meet both DSO and EVU's needs.

The interactions between an EV and the electric grid, including charging interactions, are known as vehicle-to-grid (V2G) interactions. In the case of energy

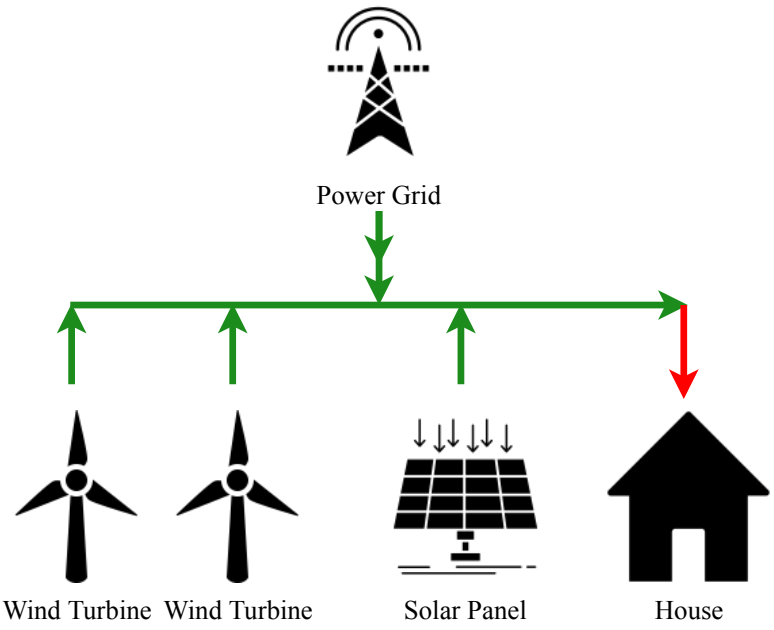

Figure 1: A conceptual model of the energy grid, with transmission, distribution and distributed power generators.

charging interactions, EVs interact with the grid by communicating with charging stations (CS) which might be owned by the DSO or by some independent, third-party entity known as the charging station owner (CSO). For this reason, many of these interactions raise several privacy concerns towards the privacy of the EVUs. Being able to identify a CS in a charging transaction means being able to derive the exact location where that transaction took place. If the identity of the EVU involved in the charging interaction is also known or is derivable, it is possible to know who (which EVU) was where (which CS), and at what time (the transaction timestamp). In the long run, the entity or entities having access to this information can easily track each EVU and determine typical patterns that can be used for activities such as targeted advertising.

Thus, there is a need for a solution in which the needs of both the DSO and the EVU can be addressed without threatening the privacy of the EVUs. In a system like this, the DSO would still be able to incentivise EVUs to charge their EVs in a specific district and within specific time frames, while EVUs would be able to charge their vehicles without the risk of being tracked and profiled over time.

1.2 Use Case

In the scenario used in this thesis, illustrated in Fig. 2, a distribution system operator (DSO) is willing to incentivise electric vehicle users (EVU) to charge at specific times and in a specific district to reduce the amount of excess energy and avoid reverse power flows. Whenever a new peak time is forecasted by the DSO, it publishes an energy flexibility request on an energy marketplace. An *energy flexibility request*

is a request by the DSO that a certain amount of energy needs to be consumed in a specific energy district within a certain time range. The marketplace is not accessible directly by individual EVUs but by energy retailers (ER). In addition to providing charging services to their customer EVUs, *energy retailers* also fulfil the role of mediators matching the energy needs of the DSO with the charging needs of their customers. For their work of mediation, ERs are rewarded by the DSO every time they succeed in completely satisfying an energy flexibility request published by the DSO (i.e. when their customers charge for the amount of energy the DSO asked for, which was enough to avoid the reverse power flow in a specific energy district).

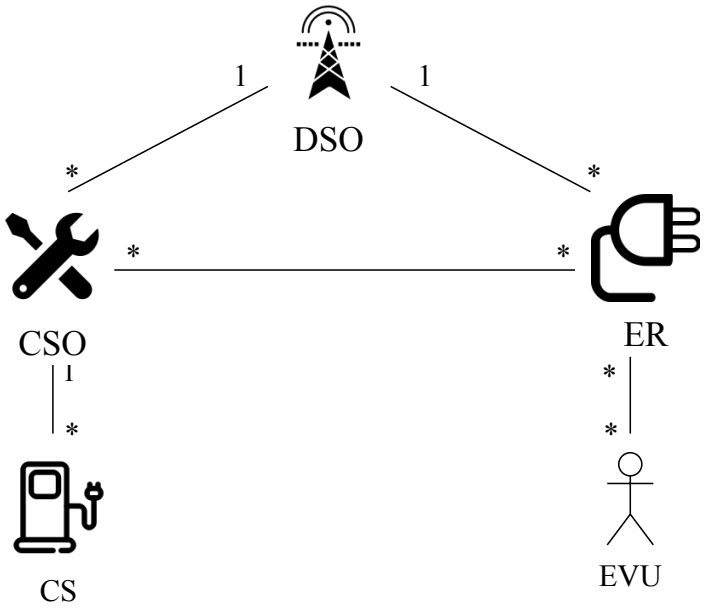

Figure 2: Entity-relationship scheme showing the different relationships among the actors in the market. The assumption here is that there is a single DSO.

The market relationships are defined as follows:

- A DSO has agreements with one or more charging station owners (CSOs) which have deployed CSs in one or more energy districts.
- A DSO has agreements with one or more ERs to allow them to sell energy to their customers.
- An ER has contracts with several different customer EVUs, and each EVU can be a customer of one or more ERs.
- An ER can provide its services through the CSs belonging to one or more CSOs.
- The CSs of a CSO can be used by one or more ERs.

Every time an EV charges at a charging station (CS), a *charging event* takes place. Each charging event includes details such as the specific CS and the EVU

involved in the transaction. However, having a per-CS granularity in the charging events would represent a large threat to the privacy of the EVUs. Furthermore, with the new General Data Protection Regulation (GDPR)[1], businesses are required to reduce the amount of personal information collected to the minimum needed to offer the service, and to adopt suitable measures to properly collect, process and store such information.

Specifically, in these types of scenarios where a DSO needs to keep the energy grid balanced, a district-level granularity for charging events is sufficient to verify that the customers of a specific ER have charged for a certain amount of energy within that district. Furthermore, a DSO is only interested in the amount of energy that has been charged by the customers of an ER in a district during the peak time, hence they should not care about the identities of the EVU or the CS involved in any charging event. District-level information is also sufficient for the ER to receive the incentives from the DSO. The ER also needs only information about the owner of a specific CS, for billing purposes, without the need to identify the specific CS that took part in the charging event. In the same fashion, CSOs should not be able to obtain the identities of the EVUs or to derive patterns of EVUs' habits by simply collecting information from the CSs they own. The only information they need concerns the ER that a specific charging EVU is a customer of, to correctly bill that ER.

1.2.1 Requirements

Considering the aspects discussed above, the solution to the presented use case must fulfil the following privacy requirements:

1. **PR 1**: The DSO MUST NOT be able to identify specific EVUs.
2. **PR 2**: A CSO and its CSs MUST NOT be able to identify a specific EVU engaged in a charging event.
3. **PR 3**: A CSO and its CSs MUST NOT be able to infer that the same EVU has taken part in two different charging events.
4. **PR 4**: An ER MUST NOT be able to infer that the same CS has taken part in two different charging events.
5. **PR 5**: An ER MUST NOT be able to link its customers' charging events to specific CSs, but only to specific districts.
6. **PR 6**: Communication between the different parties (e.g. between EV and CS, or between CS and ER) MUST NOT leak more information than needed that would make correlation attacks against the EVUs easier (e.g. network-layer identifiers such as IP and MAC addresses).

[1] https://eugdpr.org/

To make the use case generally deployable in real-world scenarios, the following business requirements must also be addressed:

1. **BR 1**: The DSO MUST be able to understand how much each ER has participated in maintaining grid stability in relation to the agreements for the energy flexibility requests.

2. **BR 2**: ERs bill their customers in a postpaid fashion, thus they MUST be able to monitor how much energy each EVU has charged over the billing period.

3. **BR 3**: CSOs are paid by ERs based on the charging services provided through any of their CSs. For this reason, CSOs MUST be able to claim payments from ERs by proving the authenticity of the charging events involving such CSs and the ER customers.

4. **BR 4**: CSs MUST always verify that an EV is authorised to perform a certain charging operation before letting the EV charge for the agreed amount of energy.

5. **BR 5**: EVUs COULD get some kind of reward from their ER every time they contribute to the grid balancing.

1.3 Problem statement and scope

The goal of the work presented in this thesis is to understand to what extent the application of the latest technologies in the field of Self-Sovereign Identity (SSI) allows the creation of a system meeting the privacy and business requirements presented in Section 1.2.1.

This thesis answers the following research questions:

1. **RQ1**: To what extent can the identity and location of the EVUs be protected, considering a scenario in which such an identity must be known by the ERs they are customers of?

2. **RQ2**: What are the possible relevant trade-offs between the privacy of an EVU and the business requirements of the other parties involved, i.e. DSO, CSOs and especially ERs?

3. **RQ3**: How can the system be designed so that the DSO can reliably and automatically evaluate the contribution of each of the ERs to the grid stability, without getting access to the identities of the individual users charging?

4. **RQ4**: How can charging events be logged and/or stored without revealing any personal information so that it is still possible to rely on them for dispute resolution between parties?

To achieve the aforementioned goal, an architecture will be designed that addresses all the privacy and business requirements introduced above. Specifically, the architecture will utilise *Verifiable Credentials* (VC), *Decentralised Identifiers* (DID) and *Distributed Ledger Technologies* (DLT). Furthermore, the architecture will be analysed in relation to the research questions and to the privacy and business requirements, to verify whether it addresses all of them.

1.4 Structure

The rest of the document is organised as follows. Chapter 2 introduces the main technologies and frameworks used in the solution. Chapter 3 presents related work that has been performed in the field of IoT privacy-enhancement solutions as well as within the context of energy markets related to electric vehicles. Chapter 4 describes the architectural choices taken during the design of the architecture. Chapter 5 provides details about the design of the solution. Chapter 6 performs an analysis of the implemented solution and answers the request questions presented in the previous section. Chapter 7 suggests possible future work. Finally, Chapter 8 presents the conclusions.

2 Background

This section introduces and explains the main technologies that have been used for the design of the solution.

2.1 Hyperledger Indy

For the scope of this thesis, the Hyperledger Indy framework has been used. Hyperledger is an open source blockchain-related project started by the Linux Foundation in 2015. Since its start, the Hyperledger Foundation has developed several tools and solutions to help spread the development of DLTs, blockchains, and blockchain-based solutions [36][37].

Indy, one of the projects, has been developed to provide a solution to manage digital identities that are authentic and privacy-preserving thanks to the usage of a permissioned blockchain[2]. The Indy framework has been used to create the Sovrin Network[3], a public network for issuing and verifying digital credentials, launched in July 2017. The network is regulated by a consortium of companies, known as the Sovrin Foundation, and new nodes can join the network after going through an identity verification process.

Since Indy relies on a permissioned DLT, the entities in an Indy network must fulfil one or more of the following roles: trustees, stewards, endorsers, and users.

Trustees are the entities that regulate the network. They have full authorisation to elevate/demote other entities privileges, as well as to accept new entities in the network.

Stewards can add one and only one validator node to the network. This node would connect to the other nodes in a peer-to-peer manner, would get its state in sync with the global state by running the consensus protocol, and would then be able to serve requests received by the users. A steward can also register identities for endorsers to on-board them onto the network.

Endorsers correspond to the public entities that are capable of issuing digital credentials. Public institutions usually join the network as endorsers, even though they might be willing to run a node as well, in which case they would also be stewards.

Users are the entities interacting with endorsers to receive digital credentials. They are usually on-boarded onto the network by endorsers, upon a verification process taking place by some other means (e.g. offline) that leads to the issuance of some type of digital credential.

2.2 Distributed Ledger Technologies (DLTs)

The term *ledger* denotes a registry in which an entity, e.g. a company, logs all its expenses and income over a certain period of time[4]. Every single expense or income is defined as a *transaction*. A transaction T changes the ledger state from S_{s-1} to

[2] https://www.hyperledger.org/projects/indy
[3] https://sovrin.org/
[4] https://www.dictionary.com/browse/ledger

the new state S_s such that the new state S_s includes the effects of executing the transaction T on the system at state S_{s-1}. Transactions involve two or more *accounts*, uniquely identified across the entire ledger, with the structure of the identifier specific to each ledger.

One key property of ledgers is *immutability*: all the transactions that are performed are permanently stored in the ledger history, with no possibility to delete them. The only possible way to revert the effects produced by a transaction T_1, if possible, is to execute another transaction, T_2, producing the opposite effects as T_1. If no other transaction is executed between T_1 and T_2, the ledger state can be reverted to the one in place before T_1 was executed. Nevertheless, even though the final and initial states are equivalent, the two transactions T_1 and T_2 are both registered into the ledger history.

With the advent of computers and digital devices, physical ledgers have been replaced by their digital counterparts. In the case of *distributed* ledgers (DLTs), the ledger is replicated or split onto different machines geographically spread across several connected sites connected. Each machine that is part of the network is called a *node*.

A subset of DLTs, blockchains, organise transactions in blocks and link blocks with one another [24]. In a *blockchain*, transactions are not executed as they are submitted to the ledger, but they are temporarily collected in a specific space of the working memory, called *transaction pool*, by each node in the network. Due to their geographical distance, the pool of each node can contain a different set of transactions. Every B seconds, the *block time*, these transactions are submitted to the system as a block for evaluation and execution.

Which transactions are included in each block and in what order depends on the node that is responsible for committing the block to the blockchain's history. This process depends on the specific implementation of each blockchain and on the set of rules regulating how transactions are evaluated, executed and distributed across the nodes: the consensus protocol. Once a block of transactions is added to the blockchain history, it is cryptographically linked to the previous one to form a chain, from which the term blockchain comes. The cryptographic link makes the new block immutable and serves as a proof of the blockchain current state so that changes to any previous blocks would result in a change of all the successive blocks up to the current last block as well. The inconsistency between the last cryptographically valid block and the altered one as the result of the change of a previous block in the chain would easily be detectable by the network [24]. This mechanism ensures the immutability of the ledger in blockchains.

2.2.1 Properties of DLTs

The distribution of data across several machines has benefits but also poses some challenges to the implementation of DLTs.

The first benefit is availability. *Availability* is the capability of the system to provide its services with no downtime [28]. It refers to the time the system has been working properly over the total time it has been deployed. For instance, a system

that has worked for 99 hours over a 100 hour period in which it has been deployed has 99% availability. In a distributed ledger, availability is provided by the presence of multiple nodes storing the same replicated global state: the failure of a single node or a small subset of nodes does not affect the capability of the system to perform its job and to satisfy requests by clients.

The second benefit is *data replication*, denoted as the capability of the system to store the same copy of the data on multiple nodes. This is very similar to performing several backups of the stored data every time the data is updated. The higher the number of nodes in the system, the higher the guarantees that the data stored will not be corrupted/lost. This, however, also leads to a higher cost for updating such data, since more nodes need to update their copy of the data.

One challenging aspect to take into account when talking about DLTs is that due to their decentralised nature, the nodes replicate a part or the totality of the stored data. For this reason, the data cannot be considered private since it is not stored on only a single node. This issue is even more relevant if the DLT has no special policies for regulating access and participation of nodes to the network, like in public DLTs, or if the data is not encrypted before being replicated. These two aspects represent great privacy issues to the data that is stored if they are not carefully considered during the design of a system relying on one or more DLTs.

Another challenge for distributed ledgers is to ensure that the global state is always consistent, i.e. all the nodes that are part of the DLT network have the same, unique state at any given time [28]. This property is called *consistency*.

In a DLT, the mechanism to keep the global state consistent across all the nodes is called *consensus*. The main goal of consensus is to verify that the transactions submitted to the ledger fulfil a specific set of conditions and to agree on their order, which is critical to determine the resulting state. For instance, there could be two transactions T_1 and T_2 in a block that is currently being evaluated by a node, where T_1 moves some money from account A to account B, currently with a balance of 0, and T_2 moves a smaller amount of money from account B to account C. If T_1 is evaluated before T_2, the entire block is considered correct since all the transactions in it can be executed with no errors, so the block can be committed to the blockchain history and the balances of the accounts A, B and C are updated accordingly. On the other hand, if T_2 is evaluated before T_1, the block will be considered invalid since T_2 fails due to the lack of funds in account B to operate. Either choice can be made, but the consensus must ensure that the same choice, i.e. the same order of transactions, is executed by all the nodes in the network. In case this does not happen, there might be nodes in which the block is executed, increasing the balance of accounts B and C, and nodes in which the block is not executed, leaving the balance of account B to 0.

In a centralised ledger, all the data resides on the same machine or in a set of machines controlled by the same entity, hence it is easier to keep consistent. Nevertheless, in a centralised ledger, there is a single regulating entity that is responsible for the correct working of the system, and users of the system must have trust that the system works as it is supposed to.

As an example, in a financial context, such as a bank, there are two possible

ways in which a malicious actor can perform unauthorised transactions on behalf of a user: by compromising the user's account, e.g. by stealing security codes and banking credentials, or by compromising the bank IT system. The former case is typically easier but still leaves to each user the choice to implement all the security precautions to secure his/her account. In the latter case, on the other hand, the user is not responsible for the unauthorised transactions. Instead, the trust relationship established with the signature of a customer contract is broken, with the bank as the only responsible party for the event. Alternatively, a distributed financial ledger would make the second vector of attack more difficult to implement, since each transaction needs to be explicitly authorised by the owner of the account and there is no central authority that can be compromised.

2.2.2 Types of DLTs

A DLT can belong to one of two categories: public (permissionless) or private (permissioned) [71]. A recap table is shown in Table 1.

A *public and permissionless* DLT is open for anyone to join, i.e. anyone can install the required software and join the network by running one or more nodes. Since it is a public DLT where anyone can propose new transactions to execute and can also participate in the consensus process, there is a need for strong security guarantees against malicious/invalid transactions and nodes running a malicious version of the consensus algorithm. The requirements for stronger security measures usually negatively impact the efficiency of the DLT both in terms of the number of transactions that can be evaluated and executed per unit of time, and also in terms of energy consumed by the nodes participating in the consensus protocol.

A *private and permissioned* DLT is in nature a private DLT, but capable of accepting new nodes and participants and assigning them specific roles. The nodes are typically organised in consortia, where legal contracts are signed and enforced. In such DLTs, usually, a distinction between roles is evident: some nodes can execute transactions against the ledger state and participate in the consensus process, some nodes can only perform read operations, while some other nodes might only be running the consensus protocol.

-	**PUBLIC**	**PRIVATE**
ACCESS	READ/WRITE to anyone	- READ to anyone - WRITE restricted
NETWORK ACTORS	Untrusted	Trusted/Semi-trusted
SECURITY	- PoW - PoS - Economic incentives	- Legal contracts - Proof of Authority
SPEED	Slow	Fast
EXAMPLES	- Bitcoin - Ethereum - Monero	- Hyperledger Indy - Ripple - Libra

Table 1: Different types of DLTs and their properties [71].

2.2.3 Consensus protocols

The consensus algorithm is the most important part of a DLT since it defines the key properties of the DLT such as scalability, energy consumption and degree of openness. *Scalability* refers to the capability of a system to handle a growing amount of work by adding resources to the system [13]. *Energy consumption* represents the amount of energy consumed by the nodes running the consensus protocol, which is strictly linked to their environmental impact. For instance, the consensus algorithm used in the Bitcoin cryptocurrency system consumes the same annual amount of energy as Austria [23]. *Degree of openness* represents the capability of including new nodes in the network once the network is already working.

Since nodes in a DLT network might behave maliciously, the class of consensus protocols used to coordinate the nodes within a DLT network is called Byzantine Fault Tolerant (BFT), from the very famous example of the Byzantine generals trying to reach consensus whether or not to attack (all together) or not a village owned by the enemy [45]. BFT protocols have been adopted in distributed systems long before blockchains and newer consensus protocols were developed [7][19]. However, due to the high degree of openness characterising some of the DLTs today, special consensus protocols have been developed to increase the security guarantees of an open DLT, to increase the costs of Denial-of-Service (DoS) attacks targeting the system availability, and to reduce the possibility of committing malicious transactions to the ledger history. The two most important protocols that have been developed are Proof-of-Work (PoW) and Proof-of-Stake (PoS).

Proof of Work (PoW): this consensus protocol is usually adopted in public DLTs, and specifically in blockchains, where the set of validator nodes cannot be trusted. The PoW consensus relies on specific mathematical problems that require a large amount of resources (RAM, CPU or network) to be solved but that are relatively easy to verify [39]. The validator nodes that compete with each other to first solve the problem are called *miners*. The fastest miner to correctly solve the mathematical challenge is given the possibility to add the new block to the blockchain. Such a block contains the transactions that node has received, organised in an order specified by the node.

The main goal of PoW is to reduce the possibilities of denial of service (DoS) attacks, as well as to reduce the possibilities of attacks altering the past state of the ledger. Nevertheless, PoW is vulnerable to a vector of attacks known as *51% attack* [72], in addition to being very energy inefficient due to the large amount of energy required by miners to solve the complex mathematical problem for each block [33][23].

Most well-known examples of blockchains using PoW-based consensus are Bitcoin [51] and Ethereum [15]. The IOTA[5] DLT also uses a PoW-based consensus, even though transactions are not stored in blocks and linked to form a chain, but in a different structure more suitable for IoT use cases: the Tangle [54].

Proof of Stake (PoS): this consensus protocol is also used in public blockchains, albeit its adoption has only recently spread. In PoS, the validators do not compete

[5] https://www.iota.org/

to solve a difficult mathematical problem, as in PoW. Instead, the validator of the next block, the *forger*, is chosen based on a random selection where the weight is represented by the stake each node has in the network, e.g. the number of coins it owns [14]. For instance, a validator node whose weight is 20% will, on average, validate 20% of the total blocks submitted to the blockchain history. Validating a new block requires the forger to put part of its stake "at stake": in case the forged block is declared invalid by the other nodes in the network, the stake is lost by the forger.

The idea behind PoS consensus is that by allowing nodes that have more stake to validate blocks of transactions, they have a larger interest in running the network correctly. If the transactions are not properly verified and the network is not correctly governed, the assets exchanged on the network lose value, since the remaining nodes would have no interest in being part of a system where business rules are not enforced. Nevertheless, if not correctly implemented, PoS is vulnerable to other attacks different from the 51% attack affecting PoW consensus algorithms, such as the nothing-at-stake or the false-stake attacks [41][10].

Main examples of blockchains implementing PoS consensus protocols are Nxt[6] and Dash[7]. Ethereum has also planned to move from PoW to PoS consensus with its Casper protocol [32][76].

The DLT on which Hyperledger Indy relies follows a permissioned model, with new nodes that can join the network upon acceptance from some of the entities that are already part of the network: the trustee entities [35]. The consensus protocol run by Indy, called Plenum [38], is an enhancement of the Redundant Byzantine Fault Tolerant (RBFT) protocol [5]. All the validator nodes that are part of the network, called the *pool*, contribute to keeping the global state consistent by participating in the consensus process. These nodes also interact with external applications and clients to support the features provided by the framework, i.e. issuing, managing and verifying digital credentials.

2.2.4 Smart contracts

An important feature of modern DLTs is the possibility to execute programs that run on all the nodes of the network and that can access data stored on the DLT. Such programs are called *smart contracts*, and the distributed applications relying on the functionalities of one or more smart contracts are called DApps.

Several smart contracts have been developed for a variety of purposes, as in the case of the Golem project [31], which has the goal of creating a global and decentralised supercomputer composed of the machines of all the participants. These participants are awarded tokens depending on the amount of processing power the other users of the network have used.

Smart contracts can also implement more complex business logic and business rules.

[6] https://nxtplatform.org/
[7] https://www.dash.org/

For instance, in the context of the SOFIE project[8], one of the SOFIE framework components consists of a smart contract template, customisable for the specific use case and deployable on the Ethereum blockchain, enabling a decentralised marketplace[9]. By interacting with the smart contract representing the marketplace, different parties can perform actions such as publishing an auction, making offers to participate in the auction, and closing an auction. The end of an auction would automatically move ETH tokens from the winning account to the auction owner and the asset specified in the auction to the auction winner's account. The fact that each operation invoked on the smart contract is performed by the whole Ethereum network provides the security guarantees offered by the network and its consensus rules.

2.3 Verifiable Credentials (VCs)

A credential is an attestation that, according to the *issuer*, someone, the *subject*, has certain properties, e.g. granting him/her the authority to perform certain operations. These properties can then be verified by the *verifier*, for example, to give access to a certain service. For the verifier to consider a credential valid, there needs to be a certain level of trust between it and the credential issuer. The relationships are shown in Fig 3. Typically, credentials are issued from the issuer to the *holder* so that it can be used by the holder to identify himself. Sometimes, as in the case of children or disabled people, the holder of a credential might not be the same as the subject: in such case, the holder covers the role of *guardian* and is allowed to use the credential on behalf of the subject, when requested.

Examples of credentials are a passport, a driving licence, or a university degree. All these credentials are affected by two main problems. The first problem is that physical credentials are easy to spoof and to clone, making the verification process usually more challenging and less reliable. The second problem is that usually showing a credential, e.g. the driving licence, reveals more information than what is requested by the situation. For instance, buying beer at the supermarket might require the cashier to verify that the buyer is over 18, so all the extra information revealed, e.g. name, address or even birthdate is not relevant for the use case and might allow the verifier to collect information that is not supposed to collect.

Furthermore, upon issuance of a credential, the issuer and the holder are required to be physically close, or there needs to be a third-party entity delivering the new credential from the issuer to the verifier, with all the risks involved. Moreover, if the credential to be issued is a digital one, it needs to be readable and verifiable by machines to be exchanged over the Internet, stored and verified by a digital device.

A *verifiable credential* (VC) is a specific solution for digital credentials that can be cryptographically verified to be authentic and not spoofed [65]. A verifiable credential contains claims describing the properties of the subject: these claims are

[8] https://sofie-iot.eu
[9] https://www.sofie-iot.eu/news/decentralised-marketplace-using-smart-contracts

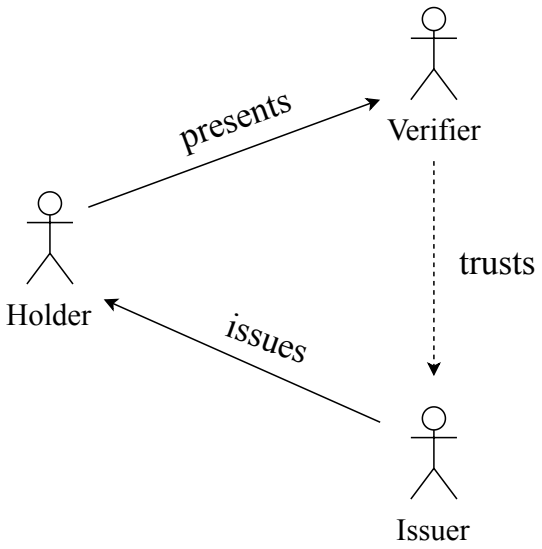

Figure 3: Interactions between the different actors involved in a typical credential usage from issuance to verification.

called *verifiable claims*. The flow for issuing and verifying VCs is not logically very different from that of a traditional physical credential and it is shown in Fig. 4.

Verifiable credentials are usually signed by someone, and such a signature must be verifiable. This implies that the identity of the signing entity must be retrievable by or already known to the verifier to verify the conformance of the credential to the expected requirements. For this purpose, a *verifiable data registry* stores the identifiers used in a specific domain, the keys needed to issue/verify credentials, and other relevant data that depend on the specific domain. This role can be fulfilled by a centralised authority or, to reduce centralisation and enable the development of privacy-preserving solutions, by a DLT.

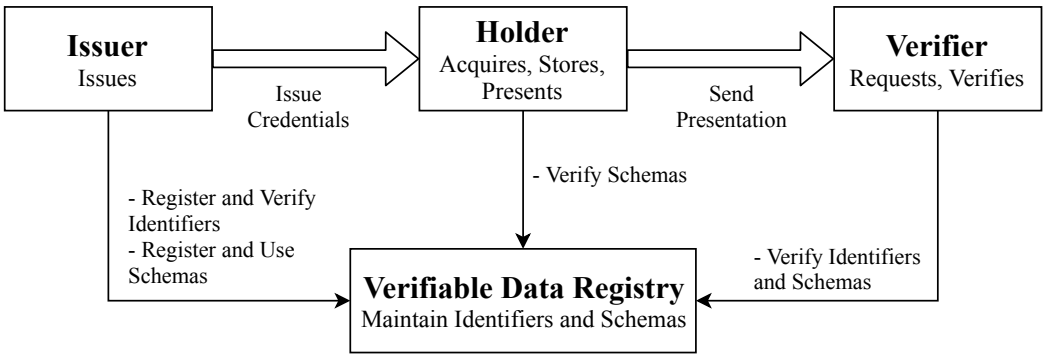

Figure 4: Different interactions taking place when using VCs [65].

When credential holders need to use one or more credentials for verification, they can generate verifiable presentations (or proofs). A *verifiable presentation* is proof that the holder of some VCs creates to prove to the verifier the ownership of one or more credentials satisfying the requirements set by the verifier. Once the presentation

is built and sent by the holder to the verifier, the latter can cryptographically verify the validity of the presentation (by verifying the validity of all the credentials used in the presentation), thus being certain the presentation has not been spoofed or tampered with.

A peculiar characteristic of VCs is that they allow a VC holder to selectively choose only a subset of the claims to be included in a verifiable presentation. Thus, a verifiable presentation can be built from several VCs without revealing to the verifier all the claims included in each of the credentials. Instead, the VC holder can create a combination of claims that minimally satisfy the requirements imposed by the verifier without leaking any additional, unnecessary information. This technique is known as *selective disclosure*.

2.3.1 VC Structure

A VC contains metadata (e.g. expiration date, revocation mechanism), a set of verifiable claims, and related proofs to verify the authenticity and integrity of the credential.

VCs have a JSON structure [65], and the set of all the fields present in a VC is called *credential definition*. Since VCs must be interoperable and machine-readable, they include one or more *contexts* containing the semantics of the fields of the VC. The value of a context is a set of URIs that, when dereferenced, allow the VC receiver to retrieve machine-readable information about the context and the meaning of any attribute in the credential. Including one or more contexts in a VC ensures extensibility of a credential with fields that are not presented in the official draft and that might be domain-specific.

The *credentialSchema* attribute defines the verification logic that verifiers will use when validating the credential. It contains information that a verifier uses to determine whether the data in the credential conforms to the expected schema. Each claim in a credential can reference a different schema, and its verification logic will depend on the specific schema that has been referenced. An example of a schema is the JSON Schema [74], which is used to describe the structure of a JSON document, expressing which fields, i.e. which keys, it should have and what is the semantic meaning of each of the fields.

A key field in a VC is the proof field, which ensures verifiability for a credential. The *proof* field is a set of cryptographic proofs that allow verifying integrity and authorship of the credential. *Credential integrity* ensures that the credential has not been tampered with since its issuance. *Credential authorship* indicates that the entity presenting the credential has been issued the credential or has the right to use the credential, as in the case of guardianship. This field is particularly useful against credential spoofing since a copy of the original credential is not usable if the authorship of the credential cannot be proved by the malicious holder. Because VCs represent a very recent technology, there is not yet a standard nor recommended proof mechanism. The two most actively used proof formats are JSON Web Tokens (JWT) [40] and Linked Data Proofs [64].

Furthermore, verifiable credentials, like traditional credentials, can sometimes

also be revoked by the issuer. For this reason, the validity status of a verifiable credential should be verifiable. This is achieved using the method specified in the *credentialStatus* field. As with proofs, there is no standard or recommended approach to provide such service, so different approaches have been taken by different implementations, preferring some properties over others, such as privacy or ease of implementation and deployment.

2.3.2 Zero-Knowledge Proofs (ZKPs)

Zero-knowledge proofs (ZKPs) are privacy-preserving proofs that allow proving predicates about the properties of a subject, e.g. his/her age, without revealing the values used for the proof, i.e. the subject's birthdate [29][55].

A valid ZKP must satisfy three properties: completeness, soundness and zero-knowledge.

- *Completeness* states that a verifier, upon receiving a valid ZKP, can trust that the prover knows the secret with very high probability.
- *Soundness* specifies that, with a very high level of confidence, a prover who does not know the secret is not able to cheat the verifier into believing that he does. The level of confidence is called *soundness error*.
- *Zero-knowledge* property ensures that a verifier has no way of knowing the secret used within the proof. The only information the verifier obtains is whether the statement that has been made by the prover is true or not.

ZKP can be either interactive or non-interactive. The first class of proofs [30], as the name implies, requires real-time communication between prover and verifier at the time of the proof presentation. Although still powerful, they have received less attention than their non-interactive counterpart due to their lack of flexibility and their synchronicity requirement.

On the other hand, a non-interactive ZKP is a proof that a prover can present to a verifier, without the requirement that both of them need to be able to communicate synchronously with each other during the process [12]. This class of proofs has received much attention and has also been adopted by some blockchain protocols to preserve the privacy of the users as well as to hide some transaction details. An example of this application is represented by Zcash, which has developed a type of non-interactive ZKP called zk-SNARK, acronym for zero-knowledge Succinct Non-interactive ARgument of Knowledge [60].

2.3.3 ZKPs and Verifiable Credentials

VCs can be included in ZKPs as long as they fulfil two requirements:

1. the VC must have the *credentialSchema* property so that all parties, i.e. prover and verifier, can perform cryptographic operations using zero-knowledge technology, i.e. proof construction and verification.

2. the VC must have the *proof* property that indicates a type of zero-knowledge proof can be built for the credential.

In a verifiable presentation (an illustration is shown in Fig. 5), credentials (or part thereof) can be included as-is, i.e. as they were issued by the issuing party, or they can be used to derive secondary verifiable credentials with privacy-preserving proofs.

For example, a credential containing a claim about birthdate of the subject might be used in two different ways. In the first case, whenever the proof that the age of the subject is over a certain value is required, the birthdate can be used directly by adding it to the presentation. In this case, the verifier will learn the age of the subject, and even just the knowledge of the birthdate might represent unneeded information leakage. In the second case, the claim about the subject's birth date can be used to generate a derived credential, called *secondary credential*, with a predicate proving that the subject age is greater than the required value. This solution does not reveal any more information to the verifier than needed.

2.3.4 VCs in Hyperledger Indy

Although different implementations of VCs have been developed, such as Credly[10], uPort[11] and VC-JS[12], this thesis considers VCs within the context of Hyperledger Indy, which is relevant for the scope of this work.

In Indy, a very strong focus has been devoted to the development of an identity system which would follow privacy-by-design principles [20]. The main idea of Indy is to use the underlying DLT to only store the elements that do not represent personal information for an entity, i.e. only information that can be made publicly available, such as the public identity of a business or a government. These elements must be accessible by everyone able to interact with the ledger.

In addition to public identities, credential schemas and credential definitions are also stored on the ledger. In this way, proofs built from a set of credentials by a prover can then be independently verified in a non-interactive and privacy-preserving way by the verifier by retrieving the needed schemas, definitions, and verifications keys used in the proof directly from the ledger. This avoids the requiring verifier to contact the issuer to verify each proof, preventing the latter from deriving usage patterns of specific credentials. A detailed description of the structure of credential schemas and definitions can be found on one of the Indy GitHub repositories [35].

Verifiable credentials in Indy have been implemented as a type of privacy-preserving credential called *anonymous credentials* [8] by following the design choices of the Idemix anonymous credential system, developed by IBM [17]. For maximum privacy, anonymous credentials have been combined with DIDs (discussed in Section 2.4) as identifiers for credential subjects and issuers.

[10] https://info.credly.com/
[11] https://www.uport.me/
[12] https://github.com/digitalbazaar/vc-js

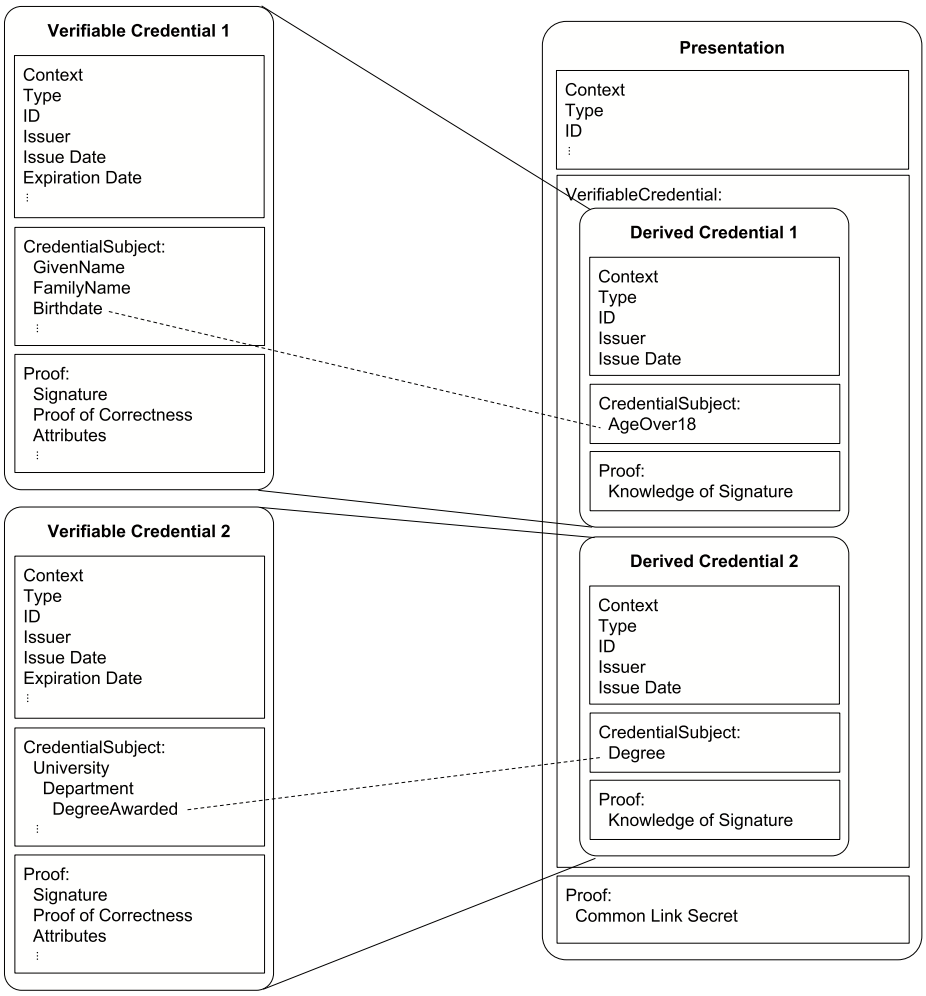

Figure 5: Structure of a verifiable presentation obtained by two different VCs [65].

The Idemix protocol provides privacy by allowing a credential holder to use different identities when interacting with each party, known as *pseudonimity*. The holder will be known with identity H_1 when communicating with the issuer to receive a credential and with identity H_2 when communicating with the verifier to present a proof.

Before requesting his/her first credential, the future credential holder generates and blinds, by the means of some cryptographic function, a private value known only to him/her, called *master secret*. When requesting a credential from an issuer, the holder sends the blinded master secret along with the credential request. The value of the blinded secret is then embedded by the issuer into the credential, so that the credential holder is the only entity that can prove the knowledge of the master secret and, hence, the ownership of the credential. The cryptographic details of how this is achieved are explained in [11].

Credentials, and hence also verifiable credentials, can be issued as well as revoked. The latter process must be designed carefully to not break the privacy or the identity of the credential owner. The typical approach for privacy-preserving credential validation schemes is based on either signature lists [52] or forward revocation lists [69]. The former is more time-efficient than the latter, but both solutions are computationally expensive for most devices, especially in Internet-of-Things (IoT) scenarios.

Therefore, Indy has taken a different approach by using cryptographic accumulators with bilinear maps [16] and tails files [9]. At a high level, a *cryptographic accumulator* can be considered as a product of several different prime numeric values, which gives it the name accumulator. Considering an accumulator $V = a \cdot b \cdot c \cdot d$, the values a, b, c, and d are defined as *tails*, while any product with one missing tail, for example the product $b \cdot c \cdot d$, is defined as a *witness*. Fig. 6 gives a visual example of a cryptographic accumulator.

For a secure cryptographic accumulator, it must not be possible to retrieve any of the factors contributing to the total product starting from the value of the product, a special type of pre-image attack. In Indy, resistance to pre-image attacks is ensured by using modular multiplication, since division is not defined in the modular world, making such attacks harder to put into practice. Moreover, to lower even further the probabilities of successful attacks, Indy makes use of tails that are larger than any numeric type can contain, and for this reason, they are stored on the underlying DLT in binary files. Each line of these tails files represents a tail, corresponding to a very long binary number. The line of each tail within the file is called the *index*.

Before issuing any credential of a certain type, its credential definition must be created and stored on the ledger by the issuer. Furthermore, if the credential supports revocation, the issuer must store on the ledger a *revocation registry*. Such a registry is composed of a set of metadata indicating what cryptographic accumulator is used and what tails file is used for the credentials conforming to the credential definition.

When a new credential is issued, the issuer communicates to the credential holder also the index of his/her tail within the tails file as well as the witness, i.e. the product of all the other tails of the credentials that have been issued up to that moment. The tail represents the private factor known only to the issuer and the identity holder. The knowledge of one of the valid tails as well as of the witness allows the holder to generate a proof of non-revocation without requiring the intervention of the issuer during each proof.

The proof of non-revocation proves that the credential holder can provide the mathematics to derive the value of the accumulator saved on the ledger by multiplying the private factor known only by him by the value of the witness. In this way, the privacy of the identity holder is preserved to a higher degree, since the issuer is not involved at all in the proving process and does not obtain any information about where a credential holder has used a specific credential.

Each time a credential is revoked, or periodically, the issuer must update the value of the cryptographic accumulator by removing the values corresponding to the tails of the credentials that have been revoked. In this way, all the holders of the

revoked credentials will not be able to provide anymore the mathematics to compute the new value of the accumulator since their tail is not part of the product anymore.

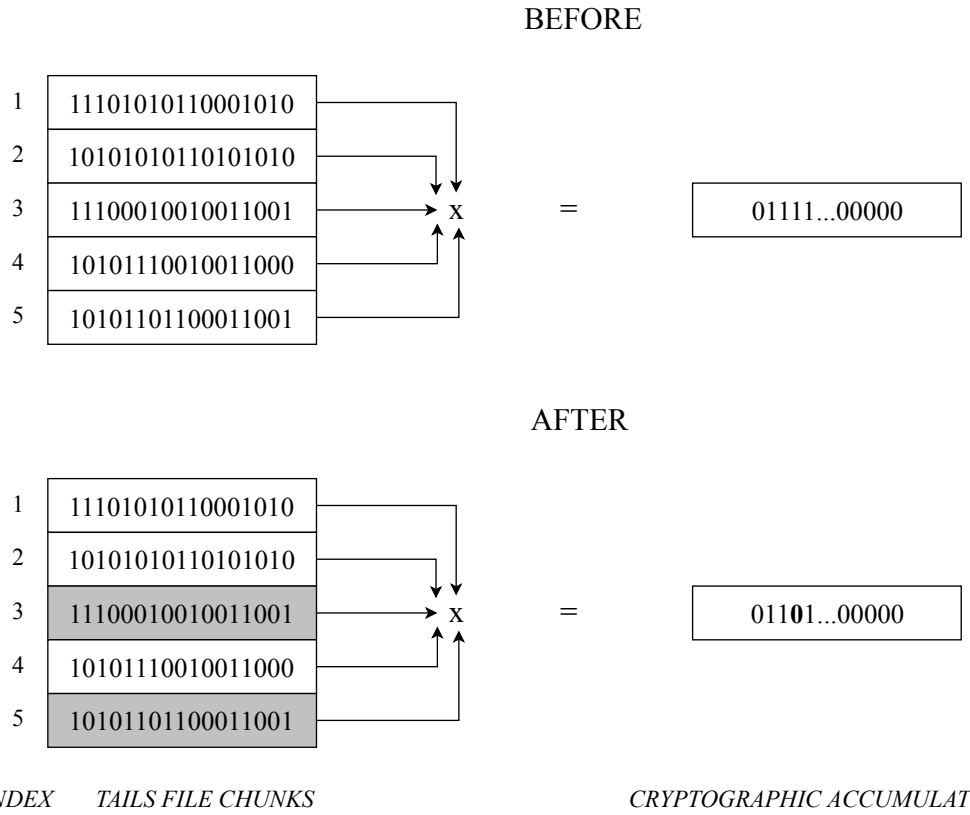

Figure 6: Example showing how credentials are revoked and how it affects the cryptographic accumulator. Greyed-out chunks indicate chunks that have been revoked and do not contribute to the final value of the accumulator.

2.4 Decentralised Identifiers (DIDs)

According to the official definition, "Decentralised Identifiers (DIDs) are a new type of identifier for verifiable, self-sovereign digital identity. DIDs are fully under the control of the DID subject, independent from any centralised registry, identity provider, or certificate authority."[56]

DID technology is very recent, and its design still under development by the W3C Credentials Community Group[13] [56]. Fig. 7 shows an example of how DIDs work in the case of an individual, Jane, interacting with several different entities, e.g. her bank, a public institution, or her employer's IT system.

According to Cristopher Allen, one of the main contributors to the design of several identity-related specifications, among which also DIDs, an identity is self-sovereign if the identity owner plays a central role in the administration of his/her own identity [3]. This means that identity owners must be able to control the identities

[13] https://www.w3.org/community/credentials/

they own and disclose them to the interested parties when needed. Furthermore, they must be able to stop sharing such information with the certainty it cannot be used anymore by any other party. Some of the advantages of using DIDs are in Table 2.

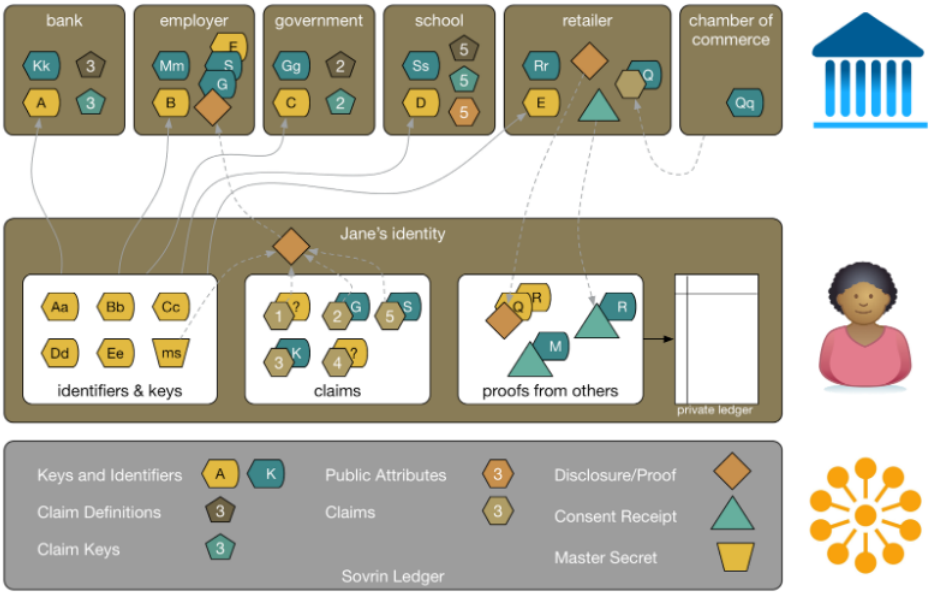

Figure 7: Self-sovereignity achieved by allowing each user to control his/her own keys and identities and disclose them when needed [48].

For self-sovereign identities to be extended to a large number of services and use cases, they ought to be as interoperable as possible. Thus, DIDs have been designed to follow all the self-sovereign identity principles and to differentiate from the traditional Public Key Infrastructure (PKI) [73].

2.4.1 Properties of DIDs

A DID is entirely controlled by its owner and does not depend on any central authority for its generation, verification, or revocation. The structure of a DID is *did:method:identifier*. The first component is the URI scheme identifier and always has the value *did* to indicate that the URI represents a DID. The second and third parts are specific to each class of DID.

Since DIDs are decentralised, also their governance does not involve any centralised party to create or delete classes of DIDs. Hence, anyone can create a new class of DIDs by specifying a new method. A *DID method* must define the rules to generate new identifiers (the identifier component in a DID URI), as well as how all the DID-related information are created, read, updated and deleted. A list of all the currently-known DID methods and their specifications can be found in [66].

A DID can be resolved to a DID Document. A *DID Document* is a set of data that describes the subject of a DID, including mechanisms, such as public keys, that

the DID subject can use to authenticate itself and prove their association with the DID.

To achieve maximum decentralisation, the resolution of a DID to a DID Document (very similar to what happens with DNS today) should take place in a decentralised yet reliable manner, e.g. via a DLT. Nevertheless, as long as there is a trust relationship between a DID resolver and the parties interacting using its services, a DID resolver can also be a central entity offering a DID lookup service. A draft is currently under development to try to design a set of common characteristics and features that all DID resolvers should provide [58].

GOAL	DESCRIPTION
Decentralisation	Eliminate the requirement for centralised authorities or single points of failure in identifier management, including the registration of globally unique identifiers, public verification keys, service endpoints, and other metadata.
Control	Give entities, both human and non-human, the power to directly control their digital identifiers without the need to rely on external authorities.
Privacy	Give entities, both human and non-human, the power to directly control their digital identifiers without the need to rely on external authorities.
Security	Enable sufficient security for relying parties to depend on DID Documents for their required level of assurance.
Proof-based	Enable the DID subject to provide cryptographic proof when interacting with other entities.
Discoverability	Make it possible for entities to discover DIDs for other entities to learn more about or interact with those entities.
Interoperability	Use interoperable standards so DID infrastructure can make use of existing tools and software libraries designed for interoperability.
Portability	Be system and network-independent and enable entities to use their digital identifiers with any system that supports DIDs and DID Methods.
Simplicity	Favour a reduced set of simple features in order to make the technology easier to understand, implement, and deploy.
Extensibility	When possible, enable extensibility provided it does not greatly hinder interoperability, portability, or simplicity.

Table 2: Benefits of DIDs listed by W3C working group in the current community draft [56].

2.4.2 DID Documents

A DID Document contains information that makes DIDs secure, interoperable (between DID methods) and machine-understandable. This widens the set of use cases in which DIDs are suitable by including scenarios relying on machine-to-machine communication, such as in IoT.

The *context* attribute adds semantic context to clarify the meaning of all the properties included in the document. This is fundamental when two digital systems need to communicate with each other and must use terminology and a protocol that both systems can understand.

Another important component of a DID Document is the *service* attribute: it enables the discovery of additional service endpoints under the control of the DID subject. For instance, an endpoint can be the URL of a REST API where the DID subject offers additional services. For this reason, DIDs are a very suitable way of bootstrapping communication between two untrusted parties.

A DID Document also contains information about the set of keys used by the owner of the identity. The keys can be used, for instance, to encrypt the data addressed to the DID's owner so that it can be decrypted only by him/her, or to verify the authenticity of the owner's identity, e.g. via a challenge-response authentication protocol.

Hence, by parsing the different components of a DID Document, a party is able to verify the integrity of the document itself, verify that the identity of the DID's owner is authentic, fetch any encryption keys for the data to be sent, and possibly bootstrap the communication according to the DID's owner preferences. All the functionalities are achieved without relying on any supervising/intermediary party (exclusion made for a possible resolution service resolving DIDs to DID Documents).

2.4.3 DIDs in Hyperledger Indy

As explained in Section 2.1, entities in Indy can fulfil one or more of the following roles: trustee, stewards, endorser, user.

Indy allows users to use *pairwise pseudonymous* DIDs in the interactions with the different endorsers (e.g. banks, public institutions, government services): the user generates and uses a different DID for each relationship he or she establishes. All generated DIDs are stored in one or more offline wallets, each protected by a master key.

Public institutions whose identity needs to be verified, e.g. for credential signature verification, have also a public DID that is used to create credential schemas, credential definitions and to issue credentials. These DIDs are the only DIDs that are used more than once across several different transactions. In particular, endorsers use per-connection pairwise DIDs to communicate with users, but will then use the same public DID to issue the required credentials.

As an example, if Alice wants to start using digital credentials, she will need to be on-boarded into the digital ecosystem by, for instance, her town hall municipality. Upon proving her identity with a physical identity document, the town hall will create a connection with her. The establishment of a connection is a multi-step

process that results in Alice and the town hall each generating a pairwise DID. The DID that Alice and the town hall have generated are unique to this connection, and will not be used in any other connection by either party. This constraint enhances the level of privacy and security of the system. If Alice will need to communicate with another entity, e.g. her university, she and the university would go through the same process of generating a new pairwise DID and use it in the connection establishment process. An example of the different DIDs created to communicate with different entities is shown in Fig. 8.

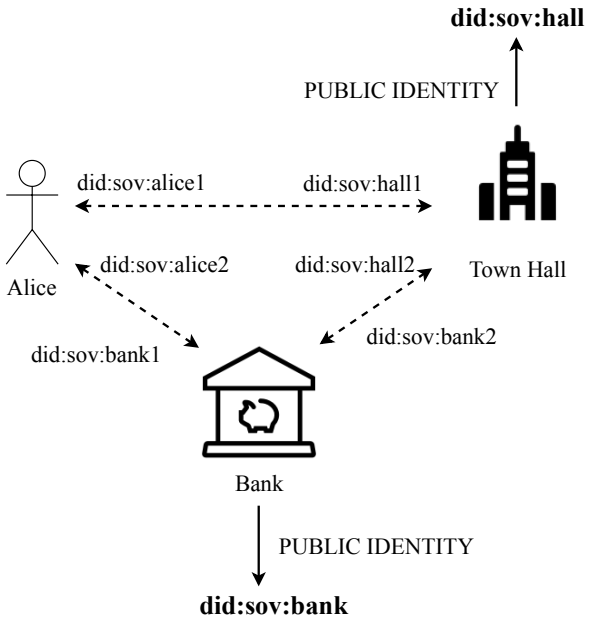

Figure 8: Difference between public DIDs and private, per-connection DIDs. Dashed lines represent pairwise connections, while stroked lines indicate the public DID of both bank and town hall.

The main goal of generating a new DID for each connection is to reduce the possibilities of linking the different interactions an entity is involved in. Since different DIDs are used, it is not possible to derive any interaction pattern for any user and understand which interactions involve the same users and with which entities they have interacted with.

Since DID Documents contain also cryptographic information, they are used to encrypt messages that only the receiver can decrypt, and also, from the receiver side, to verify the authenticity of the sender. For a public institution, DID Documents are also used to let users discover additional endpoints via which the communication can be bootstrapped. For instance, a bank might expose three different REST endpoints, and randomly include one of those every time a user tries to create a new connection. In this way, the bank can implement a DID-based load balancer.

Once a connection has been created, the issuer can issue a digital credential. In Indy, this operation is also a multi-step process which leads Alice, in the example of the town hall introduced above, to have a verifiable and anonymous credential

containing her registry information issued by the town hall. Now Alice has been onboarded since she has a digital credential and she can start collecting more credentials by using her initial digital registry credential.

Nevertheless, a credential is not useful if it cannot be used to create presentations (or proofs) and collect other credentials or access services. For this reason, Indy supports proofs. If an institution requires Alice to prove certain attributes before issuing a credential, it will use the previously created connection to publish a proof request. Once Alice receives the proof request, it will build a proof (a verifiable presentation) containing some or all of the following attributes, depending on the case:

- **Disclosed attributes**: verifiable claims that are included in the presentation without obfuscating their value.
- **Predicates**: zero-knowledge predicates about specific verifiable claims. They can be either built directly from a verifiable claim or derived from them.
- **Links**: if the presentation includes claims from more than one credential, they are used to prove that all the credentials used in the proof have been issued to the same user.
- **Non-revocation**: if any of the credentials used in a proof support revocation, it is used to prove that all such credentials have not been revoked.

Once the institution has positively verified the proof, it can publish a new credential that Alice can obtain.

Before being an Hyperledger project, the Indy framework was developed by Evernym[14] under the name of Sovrin. For this reason, the DID method used in Hyperledger Indy is called Sovrin and has the structure *did:sov:<identifier>* [48]. The Sovrin DID method is defined as follows:

CREATE: DID creation follows specific rules. The DID must be unique across the entire system. The entity willing to have its DID registered will need to first generate the needed encryption and signing keys. Then, it will add the public components into a DID Document, along with other details, and will send the DID Document to the party that has the rights to write the DID Document onto the ledger. Trustees can write DIDs belonging to other trustees, stewards, endorsers, and users. Stewards can add DIDs only for endorsers and users. Endorsers can add DIDs only for users, who in turn cannot register any new identity but only update the one they own, e.g. rotating encryption/verification keys.

READ: anyone can look up from the ledger a DID and retrieve its associated DID Document. The authenticity of the answer is either guaranteed by the fact that several validators return the same document, or by returning also a cryptographic proof based on Merkle trees [49][50].

[14] https://www.evernym.com/

UPDATE: the owner and the optional guardian of a DID are the only entities that can update the associated DID Document since they are the only ones owning the associated private keys. An UPDATE transaction is sent to the ledger, which must be signed with one of the keys specified in the *authentication* property of the DID Document.

DELETE: since DLTs are append-only, a DID and its document cannot be truly deleted from the ledger. A semantically equivalent result is achieved by updating a DID Document and setting its authentication keys to an empty list. In this way, operations on that DID cannot be signed/verified anymore, making the DID unusable.

3 Related work

This section presents previous research in the field of privacy-preserving vehicle-to-grid (V2G) interactions, grid balancing via electric vehicle charging, and privacy-preserving charging schemes. Privacy-preserving authentication and authorisation schemes for IoT devices, including decentralised identifiers (DID) and verifiable credentials (VC), are also discussed.

3.1 Grid balancing with EVs

Some of the solutions to the grid balancing problem have considered electric vehicles that act both as additional energy generators (by discharging at CSs in case of scarcity of energy in the grid) and as energy consumers (by charging at CSs in case of excess energy).

Since additional (dis)charging operations impact the EV battery performance over time and require extra effort by the EVUs, they need some form of incentives to participate in the grid balancing activities and to be rewarded once they have offered their vehicles for the purpose. Whenever they decide to contribute to the grid stability, EVUs must be able to claim such rewards from the requesting entity, be it the DSO or some second-level actor, by proving their participation. This is where most of the privacy issues arise for them since the information revealed can allow the other parties involved in the verification process to track each user's activity over time.

Furthermore, since it would be challenging for a DSO to interact with each EV due to their number and their level of mobility, the role of the *aggregator* is typically present in these use cases. Similarly to an energy retailer (ER), the main goal of an aggregator is to assist the DSO in shaving the peaks and lows of energy production and keep the grid stable, for which it gets paid by the DSO. An aggregator signs agreements with several EVUs, and, during periods of potential grid instability, rewards the EVs that help to keep the grid stable by charging/discharging at the right time in the right place.

In *Yang, Zhenyu et al.* [75], the authors present a solution in which DSOs and aggregators interact via energy charging/discharging requests published on a marketplace accessible by both parties. After fulfilment of a request, the DSO pays the aggregator who, in turn, issues digital tokens, called E-cash, to the EV owners. E-cash can later be redeemed for some services such as battery maintenance or parking discounts.

The solution is innovative with regard to how it handles the privacy of EV owners, by using blind signatures and pseudonyms. On the other hand, there are design choices that might lead to situations in which the system is compromised.

Specifically, the system includes a trusted entity generating and distributing key pairs to all the other parties; the distributing party also knows the real match between EV owners and all the pseudonyms they have used, for traceability reasons. This represents a single point of failure for both the availability and the security of the system: if such a central authority were to become compromised, the identities

of the users would be compromised as well, and generation/verification of signatures would no longer be considered reliable.

Also in *Chen, Jie et al.* [21], a marketplace-based solution is described where the DSO publishes service requests in the form of auctions on an electricity market. These service requests ask charging stations (CSs), acting as local aggregators and aggregating several EVs, to charge or discharge a certain amount of energy within a specific zone and within a certain time. Thus, the CSs compete with each other to win the auction by requesting the lowest amount of reward to fulfil it.

Using traditional identifiers for the CSs would raise several privacy concerns especially for the end users that agree to offer their vehicles to let the CS achieve its goal. For this reason, a solution in which the identities of the CS are hidden ensures higher levels of privacy.

In the work, the anonymity for the CSs in the transactions is achieved by the means of group signatures and group membership certificates, while EVUs make use of pseudonyms to hide their real identity. Nevertheless, it would be desirable not to have a centralised authority, albeit trusted, in privacy-preserving solutions, since the security of the entire system might be nullified for to the reasons explained in the previous paragraph.

Nicanfar, Hasen at al. [53], *Abdallah, Asmaa et al.* [1] and *Liu, Hong et al.* [47] also present solutions relying on a centralised identity for generating and managing identities of the parties involved. The papers propose different approaches to preserving the privacy of the end-users but fail in providing an adequate level of decentralisation which would make it possible to not rely on a single authority for the security and privacy of the whole system.

In the first work [53], a centralised trusted authority, the Smart Grid Server (SGS), generates keys and identities for both CSs and EVs. Furthermore, the SGS has full knowledge of any interaction taking place between EVs and CSs thanks to a private database containing all the needed data and metadata. Cyber attacks represent a serious threat to the security of the system, as the SGS can soon be identified as *the* single point of failure for both system security and availability.

The second paper [1] proposes a solution relying on temporary pseudonymous identifiers for EVs and different types of message requests for charging/discharging between a local aggregator, managing a fleet of EVs, and the DSO control centre (CC) unit. The paper presents a protocol and proves its lightweight nature, which makes the protocol very suitable for IoT scenarios as in the cases with CSs or EVs.

The payment model described in the paper involves the intervention of the aggregator for each EV-CS interaction, e.g. for each charging event, since the EV must pay each time it charges at a CS, and the aggregator remotely unlocks the charge at the designated CS. Nevertheless, even though the identity of EVs is protected by the usage of pseudonyms, the identity of CSs is not and is known to the aggregator for each charging event, thus posing new privacy threats for EVUs.

The third work [47] analyses the three roles that an EV can play during its interactions with the grid (request, store or provide energy), along with the privacy implications of each of the scenarios. Requesting, storing and providing energy are

slightly different operations relying on the same set of techniques to keep the identity of EVs protected: group/ring signatures, a trusted certificate authority (CA) issuing and managing keypairs for the EVs as well as providing authentication service to querying aggregators, and session (pseudonymous) identifiers.

Although the proposed solution solves some of the privacy challenges typical of role-based access control scenarios like the one presented in the paper, it does not address others. Firstly, the presence of a centralised CA reduces the overall security of the system and represents a single point of failure, in addition to being a bottleneck. Furthermore, the use case considered does not take into account the CSs, and possible identifiers used for them.

While more focused on optimisation algorithms to minimise the difference between the amount of power supplied by the grid and the amount of power released by the EVs batteries, the work presented in *Rottondi, Cristina et al.* [57] addresses the privacy of the EVs involved in the process. For the purpose, the authors propose the use of an anonymiser: an intermediary between the EVs and the local aggregators acting as a proxy in the communication that replaces the real identity of the EVs with pseudonyms and forwarding the request to the intended receiver.

Nevertheless, the introduction of a new entity in every interaction to hide the identity of the EVs poses new challenges for securing the system and moves the critical point of the system security to the anonymiser, which represents a bottleneck of the proposed system.

Some work has also been performed on a higher level, aimed at identifying all the security and privacy challenges typical of vehicle-to-grid (V2G) interactions. In an attempt to propose a general framework architecture that V2G systems can implement to enforce security and privacy by design, the work described in *Saxena, Neetesh et al.* [61] addresses many of the challenges for that purpose. Among the security objectives, the paper mentions *mutual authentication* for interacting parties, *information confidentiality*, and *message integrity*. Regarding the privacy objectives, much attention is paid on *identity anonymity* and *vehicle untraceability*.

The proposed architecture includes several entities interacting with each other to achieve the goal: communication servers, authentication servers, control centres, billing generation, and payment management servers. Anonymous authentication schemes, i.e. authorising users without identifying them, might be used to ensure identity anonymity and untraceability for EVs. Data confidentiality and privacy could be implemented by using the latest encryption techniques, such as homomorphic encryption, to enable computations over encrypted data without the need to decrypt it.

The paper also highlights the challenges of deciding the best payment scheme to use in such systems, with some schemes more suitable because of their flexibility (debit/credit cards) but lack of privacy, and some other more privacy-preserving, such as E-coins and prepaid cards, but less flexible to be adopted in a wide range of different scenarios.

3.1.1 Network anonymity

Some of the works presented above, specifically [21] and [47], do not address another important concern: the leak of identifying information proper of long-lived network identifiers, e.g. IP and MAC addresses. The nature of such identifiers (long-lived or even fixed over time) is a real attack vector used in profiling and fingerprinting the identities behind a specific identifier. For this reason, the information leaks that can be generated by not properly masquerading those identifiers are very important to address.

One work that properly addresses network-level information leaks is described in *Stegelmann, Mark et al.* [68]. The solution presented in the paper proposes a system designed so that communication between an EV and the aggregator, which do not share full trust, takes place using an anonymity network like Tor[15].

This issue is very important to address since network-layer identifiers correlation cancels out any other attempt of hiding the identity of an EV and its owner from the parties it interacts with. Nevertheless, the solution proposed also relies on a centralised certificate authority for certificates management. Furthermore, the use case considered in the scenario is quite restricted since aggregator and CS owner are the same entity. Removing this assumption makes the management of the EV owner's identity and verifiability of transactions more challenging in the proposed system.

Overall, the solutions proposed in the aforementioned sections address differently the issue of hiding the identity of EV and EVUs from other parties. This is typically achieved by relying on a trusted third-party that all the actors need to blindly trust to operate correctly and to be secure enough to ensure the security of the whole system. However, having a higher-level regulating authority creates dependencies that can easily be removed by using decentralised approaches, where each entity is responsible for its security and privacy.

The third-party introduced in the works, assuming it is not a specific aggregator or ER, can identify specific EVUs by analysing the charging transactions, which violates PR 1, described in Section 1.2.1. Furthermore, the business context and requirements of such use cases are largely different than the ones given in Section 1.2.1.

3.2 EVs in charging transactions

Other work has focused only on the interactions between EVs and CSs during the event of charging and their privacy implications, with no specific use case such as grid balancing.

The authors in *Langer, Lucie et al.* [46] provide a high-level explanation of what are the privacy challenges related to authentication and billing when charging an EV at a CS, without proposing a concrete solution to address them all. The authors

[15] https://www.torproject.org/

divide the use cases into four categories: controlled/uncontrolled customer-premises charging, and controlled/uncontrolled public-premises charging.

The former two cases indicate charging events taking place at the EV owner's private premises, in which the grid operator might (controlled) or might not (uncontrolled) be involved in the charging process (e.g. by offering discounts to charge in specific times of the day).

The latter two cases refer to charging events in which an EV is charged in a place different than its owner's premises, introducing additional privacy-related precautions that must be taken into account when designing a system offering such feature.

The paper also suggests possible solutions to enhance the privacy of such charging events, such as the use of district-level information instead of per-CS identifiers or hiding the identity of the user charging by using pre-paid smart cards. The latter approach raises new issues about linkability of transactions since smart cards are typically statically identified over their lifetime.

Au, Man Ho et al. [4] propose an easy-to-deploy solution which allows the EV owner to hide his/her identity across charging events by using pseudonyms and zero-knowledge proofs (ZKP). At the same time, the system provides auditability of transactions by introducing a third-party judging authority that can intervene only upon EV owner request; to do so, the EV owner is required to disclose the secret used to generate the pseudonyms and the ZKPs.

In the scenario described in the paper, the CSs only act as a front-end terminal routing the traffic from the EV to the billing server (for heavier cryptographic operations) and vice versa. Nevertheless, no mention is made about possible side-channel leaks, such as IP-related information, which would allow a CS to link two separate charging events to the same vehicle.

To introduce some degree of decentralisation to address the typical problems affecting centralised systems, in *Knirsch, Fabian et al.* [42] the authors have proposed a solution in which the main interactions between EVs and CSs take place via a blockchain-based energy marketplace. On such a marketplace, the EVs publish requests to charge in a specific area, before a specific deadline and for a certain amount of energy, and the CSs compete with each other to offer the lower price to charge.

Even though the scope of the paper does not cover any specific payment method, it does not take into account possible malicious behaviours of either the CS or the EV. The paper also does not consider possible interruptions in the energy flow through the CS during a charging event, resulting in the CS being paid anyway for the total amount of energy since such payment is made beforehand. Such a case can only be resolved by third-party trusted authorities.

Furthermore, no mention is made about how an EV can be sure that a winning CS is located within the region specified in the request, so some kind of authentication mechanism should be in place to let entities mutually authenticate.

The use of ZKPs and decentralised technologies such as blockchains offer ways to address several privacy and security challenges, including the presence of a central authority. The idea of a decentralised marketplace where contracts can be signed and immutably stored on a shared ledger increases the probability that the parties

signing the contract adhere to their obligations. Nevertheless, more challenging is the resolution of possible disputes, where the privacy of the entities involved must still be maximised.

In a decentralised solution, this should be possible to achieve in a way that proves that all the parties involved in a transaction agreed to execute the transaction, without identifying the specific parties involved. Decentralised solutions, like blockchains, are also extremely beneficial for addressing information leaks from Internet-based communications since there is no direct communication between two entities, but everything takes place through the execution of transactions. In this case, the privacy problem is moved onto the identifiers used in such transactions, e.g. whether they are reused or used only in a single transaction. For single-use identifiers, proving the ownership of an identifier (e.g. proving the binding between a single-use identifier and a real identity) becomes a big challenge, especially for identifiers used in past transactions.

3.3 EVs location privacy with known identity

Since almost no ER is yet willing to accept anonymous payments from its customers, the fact that traditional payment systems are mostly used by EV owners to pay aggregators/retailers for their charges limits the potential benefits that can be derived from introducing privacy-preserving payment solutions.

For this purpose, *Frosch, Tilman et al.* [27] aim to find a viable way to preserve location privacy of an EV and its owner when his/her identity *must* be known to the aggregator/retailer for business and billing purposes. The paper identifies three main problems in the area and proposes three solutions.

1. When an entity, e.g. a CS, makes use of digital signatures with the same signing key, it is easily identifiable across multiple interactions.

2. For regulatory purposes, often smart meter information (e.g. its identifier) must be included in the transaction details. This is very similar to including details about a specific CS in a transaction.

3. Network-based identifiers like IP and MAC addresses can also leak information.

All of the aforementioned issues seriously threaten the location privacy of the EVs involved in the charging transactions, since the specific CS involved in each transaction is easily identifiable and can be located on a map.

To address the listed problems, the following solutions are proposed in the paper:

1. Use of a group signature scheme. Such a scheme allows signatures to be verified as belonging to a specific group, i.e. the group of charging stations belonging to CSO_1, without revealing who within the group has signed a certain transaction.

2. The smart meter information can be added in the transaction as blinded, e.g. hashed, and the original value can be disclosed to a trusted third party only if needed, e.g. upon dispute resolution.

3. Use of an anonymizing communication network such as Tor for all Internet-based communications.

Although the solution addresses many of the problems typical of cases in which the identity of the EV owners must be known to their ERs, it still contains some potentially exploitable flaws, such as the key management procedure which is still performed in a centralised way. Furthermore, in the solution proposed, the regulatory entity, which also issues all the keys and the certificates, has access to every single CS identity, charging event and metering data. The resulting concentration of responsibilities makes the regulatory entity a sensitive target to attack to leak EV identities. Hence, a more decentralised and distributed approach would be recommendable.

3.4 ISO 15118 and the POPCORN protocol

In the recent years, the ISO standardisation body has started looking into electrical vehicle charging scenarios and has produced a set of standards, ISO 15118[16], to regulate the communication between EVs and CSs and to enhance the level of security in the related use cases. The standard defines details and protocols for V2G interactions, such as network and application requirements, physical and data link layer requirements, and various conformance tests. Nonetheless, while security has been widely considered and thoroughly analysed in the standard definition, privacy protection has not been addressed, and the results of a formal analysis presented in [6] highlights it.

The POPCORN protocol, presented in *Höfer, Christina et al.* [34], has been developed to be compliant with this standard, while at the same time improving its shortcomings regarding the privacy of EVs and their owners. The enhancements introduced by the POPCORN protocol concern four areas: minimising PII (Personal Identifiable Information), privacy-preserving functional alternatives, privacy-preserving information flows, and privacy-preserving payments.

The minimisation of PII involves the usage of dynamic identifiers to authenticate an EV to a CS, with the authentication process happening offline between CS and EV and not relying on any external service.

The privacy-preserving functional alternatives aim to replace privacy-sensitive procedures with their privacy-preserving counterparts: for instance, replacing traditional X.509 certificates with anonymous credentials due to their property of selectively disclosing the minimum information required depending on the authentication context.

Privacy-preserving information flows address leaks at the communication channel level and suggests using anonymising communication channels to reduce such leaks, as in the case of IP and MAC addresses.

The last enhancement, adoption of privacy-preserving payments, refers to the introduction of a trusted payment handler acting as the intermediary between the ER and the DSO and hiding the details of the payer from the payee while ensuring that parties are fairly remunerated for the services they offer.

[16] https://www.iso.org/standard/69113.html

Based on the identified improvements, the POPCORN protocol defines 5 phases to allow an EV, customer of an ER, to charge at a CS, owned by a CSO while retaining its privacy.

A key element of the protocol is the continuous exchange of metering receipts between the CS and the EV. At regular intervals, the EV sends a signed meter reading to the CS which can verify it and, in case of successful verification, continue supplying energy. The process can be interrupted at any time if the EV sends readings that are not consistent with the information that the CS has or vice versa. The rest of the protocol definition details how such receipts are sent to both DSO and ER for correct billing, and how the privacy of the entities involved is preserved.

Since the work performed is very relevant (especially because of its conformance to a very recent ISO standard), the protocol has been formally analysed in *Fazouane, Marouane et al.* [25], with the result that the properties such as anonymity and strong unlinkability cannot be formally proven. Furthermore, the challenge of providing privacy-preserving payments has been solved with the introduction of a trusted payment handler acting as the intermediary between payer and payee.

A more decentralised and distributed approach would be desirable in such case so that the actors do not need to rely on a single entity (also a single point of failure) to ensure that payments are correctly processed.

3.5 DID-based authentication and authorisation in IoT use cases

The application of decentralised and privacy-preserving techniques such as DIDs and anonymous credentials in the IoT world has only recently been studied, as in *Lagutin et al.* [44] and *Kortesniemi et al.* [43].

The authors in [43] study the application of DIDs and VCs in the energy sector, in a scenario very similar to the one depicted in section 1.2. In the work, the authors perform a feasibility study of using DIDs and VCs in such scenario, classifying electric vehicles and charging stations as IoT devices. The solution proposed uses single-use DIDs for both EVs and CSs and specifically-crafted ZKPs to allow EVs to charge without the risk of privacy leaks.

By using changing DIDs, the EV identity is not revealed to the CS. Furthermore, by combining DIDs with digital signatures, the ERs can prove the fulfilment of an energy flexibility request published by a DSO without revealing any personal information about the users involved in such transactions.

The usage of anonymous credentials containing the district information about a CS make sure that charging transactions do not contain information related to any specific CS. This is sufficient, since for grid balancing district-level information is the minimum level of granularity needed by both DSO, to reward ER, and by ERs, to bill their customers. In this way, CSs and CSOs are not able to link multiple transactions to the same EV, since dynamic identifiers are used for them. In the same way, the ER is not able to derive whether the same CS has been involved in several charging events. The usage of anonymous credentials and ZKPs ensures that the information exchange is verifiable, authentic and that only the minimum amount

of information required is shared between parties.

For its privacy-preserving measures, its decentralised nature and its applicability to IoT scenarios, several cues have been considered as the starting point for the architecture presented in the rest of this thesis.

3.6 SOFIE Decentralised Energy Flexibility Marketplace

In the context of the SOFIE[17] project, one of the pilots developed aims to solve the problem of grid balancing by allowing owners of EVs to charge at defined times of the day: by doing so, they can receive incentives, discounted prices, and rewards [63].

This section describes the use case implemented in the pilot and analyses its limitations with respect to the privacy of the EVUs and suitability for large, real-world deployments.

3.6.1 Use Case

The entities interacting in the pilot are very similar to the ones presented in Section 1.2, albeit some design choices make it only suitable for a small set of scenarios and not fit for more general use cases.

As depicted in Fig. 9, a distribution system operator (DSO) manages the distribution grid and is mainly interested in keeping the grid balanced and avoiding reverse power flows. Owners of electric vehicles (EV) are interested in charging their vehicles, perhaps taking advantage of peak times to obtain discounted prices or some other form of reward. The link between the needs of the DSO and those of the electric vehicle users (EVU) is represented by the fleet managers (FM). A FM has several EVU customers to which the FM provides charging services at any of the CSs it owns. These CSs can be remotely controlled by the FM to lock/unlock charges, and their metering data can be accessed to both DSO and their FM. The main business of a FM is to offer charging services to EVUs, and to obtain rewards by the DSO every time they actively and successfully contribute to the grid stability by shifting EVUs charges to the grid peak times.

Upon forecasting a possible reverse power flow in a specific energy district of the grid, the DSO can publish an energy flexibility request on the Ethereum-based energy marketplace in the form of an auction. These requests are related to a specific energy district, include a certain amount of excess energy that needs to be taken out of the grid, and have a deadline before which such energy must be removed to keep the grid balanced. The FMs then compete with each other on the marketplace by offering the lowest amount of rewards requested to fulfil the DSO request (i.e. to take out of the grid all the excess energy before the deadline).

Technically, publishing a new energy flexibility request on the marketplace means deploying a new smart contract on the Ethereum blockchain where the FMs can make bids for the number of rewards, and once an agreement is reached (i.e. the DSO chooses the winning FM), the DSO finalises the request and escrows the agreed amount of rewards in the form of digital assets.

[17] https://sofie-iot.eu

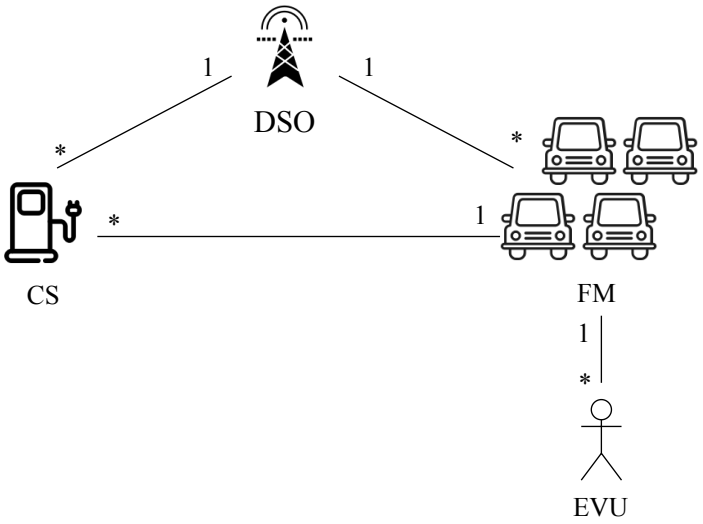

Figure 9: Relationships between parties in the SOFIE Italian pilot. The *1* indicates that only one entity is involved in a relationship, while the * indicates the involvement of one or more entities. E.g. an EVU can be a customer of only one FM, while a FM can have several customer EVUs.

The conditions for the reward are periodically verified through the interaction between the smart contract and an external oracle. The oracle is owned by the DSO and it is implemented as an API that allows querying a specific marketplace request and returns the total percentage of completion. This is achieved by fetching the total amount of energy used to charge EVs in the district since the energy flexibility request start time. If the 100% is reached before the expiration time, the smart contract moves the escrowed assets to the FM, otherwise, they can be claimed back by the DSO. To work, the oracle API relies upon the smart meters embedded within each deployed CS.

Once the FM has won the energy auction for a specific district, it will notify its customers of the new opportunity available. Customers that choose to charge following the indications of the FM can typically benefit from discounted prices for their charges. At the same time, they would allow the FM to increase the possibilities to fulfil the request and to receive the escrowed rewards from the DSO.

When EVUs need to charge their EVs, they contact the FM via their device and ask for a charging code for a specific CS. The FM generates and returns the code which is then shown to the CS as a proof the EVU has been authorised by the FM to charge. The CS forwards the code to the FM, which unlocks the charging if the code matches one of the codes generated for that CS and has not yet been used, and charges the related EVU accordingly.

3.6.2 Limitations

The novelty in the approach adopted in the pilot is the automation of contract agreement and contract resolution (i.e. paying the escrowed rewards to the FM if

the energy flexibility request is fulfilled). Such automation is achieved via the usage of smart contracts deployed on an Ethereum blockchain. The blockchain stores the signed contract between the DSO and the winning FM to make the terms of the contract immutable once the agreement has been reached.

The verification of the progress of the request fulfilment is also performed by the smart contract which, by querying an external oracle [70] maintained by the DSO, can detect when the required amount of energy has been taken out from a certain district. Once the request has been fulfilled, the smart contract moves the rewards, in the form of Ethereum assets, from the DSO account to the FM account which, in turn, might issue some rewards to the customers involved in the transactions that contributed to the request fulfilment, in addition to the cheaper price offered to them for the charging service.

Nevertheless, some limitations are affecting the system. First, a more realistic scenario would assume that the CSs are not owned by the FM, but rather by an independent, third-party entity such as the charging station owner (CSO). FMs would then be allowed to use the CSs owned by the CSO, which will then charge the FMs based, e.g. on the amount of energy provided in charging services using such CSs. Therefore, the process of proving the extent to which an FM has contributed to the grid balance becomes challenging due to the larger number of actors involved.

Another limitation of the system is the use of static identifiers for both CSs and EVs. With the use of long-lived identifiers, anyone having access to the history of transactions can easily start tracing patterns of usage for EVUs and know exactly where each EVU was at a specific time, due to their interactions with the CSs. In the long term, each EVU can be profiled by the other parties, making him/her a very attractive subject for targeted activities, such as advertising.

By evaluating the implemented solution in the SOFIE Decentralised Energy Flexibility Marketplace pilot with the privacy requirements presented in Section 1.2.1, and by assuming that the role of the ER in the requirements is performed by the FM in the pilot, we come to the conclusion that the pilot does not fulfil the privacy requirements 1, 4 and 5. Depending on who has access to the history of transactions, also privacy requirements 2 and 3 might be violated.

Overall, the assumptions that have driven the pilot development make it not properly fit for more general scenarios, in which more actors are involved and CSs are owned by independent businesses. Moreover, the privacy of EVUs has not been carefully considered in the design of the interactions between the different entities. The result is a system in which EVUs put their privacy at risk to receive discounted prices for their charges and some form of rewards.

4 Architectural choices

This section discusses the key choices that have been taken during the design of the architecture: how to bootstrap the communication between a CS and an EV at the beginning of a charging event, how to validate charging credentials before supplying energy, and how to reliably prove the correct amount of energy supplied.

4.1 CS-EV communication bootstrap for a charging event

Whenever an EV is near a CS in order to charge, the two entities must become visible and start communication with each other. This means that either the CS or the EV needs to trigger the beginning of the interaction. Furthermore, the CS and the EV cannot utilise any information collected about each other during previous transactions, since the system must ensure that transactions are not linkable with each other.

One approach would be to let the CS continuously advertise its information, such as the proof-of-ownership by the CSO, the ERs authorised to offer charging services at that CS, and the district identifier. In this way, an EV can automatically detect that a CS is advertising information and, if the CS fulfils the charging requirements (e.g. a specific ER is among the authorised ones and the district identifier is a specific one), the EV can proceed with the charging transaction.

Nevertheless, several possible solutions implementing automated charging transactions without explicit involvement of the EVU have been designed and analysed before the final proposed architecture. However, it has been challenging to identify some that would not be vulnerable against different classes of attacks, from MitM (Man-in-the-Middle) attacks to DoS (Denial-of-Service) attacks. In some scenarios in which the EVUs were showing charging certificates before cryptographically verifying the authenticity of the CSs, the identity of those EVUs were even at risk. The core of the problem is located in the fact that some bootstrapping information was sent unencrypted between the CS and the EV since they needed some initial handshaking procedure to create a temporary encrypted communication channel. This information is then exploitable by malicious third-party users to either prevent the honest parties to ever reach an agreement, to flood one of the parties with requests that would not lead to any successful charge (DoS attack), or to pretend to be one of the two parties (MitM attack).

For this reason, the final solution requires that the very first step in a charging interaction involves the manual intervention of the EVU. In particular, the EVU would interact with a CS using a short-range communication channel (e.g. NFC or QRCode) to exchange cryptographic information that the CS could then use to encrypt all the successive traffic. The use of a short-range communication channel makes the attacks described above harder to put in practice, because they would require either compromising the scanning tool used by the CS (e.g. a second NFC reader on top of the first one), or the malicious users to trick the EVUs to show them the same information that they show at the CS, so that the messages exchanged can be decrypted. Specifically, the manual trigger requires the EVU to communicate

to the CS a DID and its associated document specifying the encryption keys to successfully establish a connection between the CS and the EV. Upon connection creation, the communication can take place securely and the charging interaction can be correctly completed.

4.2 Charging credential validation process

Another choice that has been made concerns the charging credentials used by the EVUs and their issuance and validation. Specifically, one possible solution would require an EVU to register to the ER, before *each* charging transaction, the temporary DID used in that transaction. Then, during the charging transaction, the CS would ask the EVU for a proof containing the information about which ER was offering the charging service to the EV. The credential previously issued by the ER to the EVU would then be used to fulfil the proof request.

Once the CS has verified the proof, it establishes a temporary connection with the EVU's ER and queries the ER about the authenticity of the EVU's DID. In a correct scenario, that DID would have previously been registered by the EVU to the ER and then communicated to the CS in the initial steps of the transaction. To prevent any possible information leaks and reduce DoS attacks surfaces (e.g, external, unauthorised entities querying the ER about some EVUs), before a query, the CS must also prove to the ER that it belongs to one of the CSOs with which the ER has an agreement.

If the EVU had indeed previously registered the new DID with the ER, the ER returns a signed response stating that the EVU is one of its customers. Once the ER has confirmed and signed the identity of the EVU, the charging event can take place, and the signed response serves as a proof that, at the moment of the charging, the ER had authorised an EV to charge at a specific CS.

The disadvantage of this approach is that the ER endpoint needs to constantly be available for CSs to verify DIDs presented by the EVs; this represents a single point of failure both in terms of availability and security of the system. The charging of an EV at a CS should not be dependent on the availability of the ER endpoint during the transaction. Hence, a solution to this problem would be to use a more decentralised approach, which is the path taken during the design of the final architecture.

In the proposed solution, the EVUs generate a certain amount of DIDs e.g. at the beginning of the day, and then request their ERs to generate as many charging credentials, each authorising a different DID to charge. These credentials are short-lived and should preferably be used only once by the EVU, to avoid correlation attacks, since each credential reveals one of the identities used by the EVU in the charging transactions.

During a charging transaction, a CS could simply verify that a charging credential has been issued by an authorised ER and that it has not expired to let EVUs charge their cars and bill the ER accordingly. In this way, the ER is not involved during each charging transaction, and the charging certificates are only managed by the EVU to whom they were issued.

4.3 Energy supply confirmation

One relevant issue faced during the design of the interactions concerned how to validate the amount of energy that has been charged by a CS in a charging event. During a charging transaction, the parties might behave maliciously, e.g. an ER might be billed by a CS for energy it has never supplied, or the energy flow might be abruptly interrupted due to some issues in the energy grid. For these reasons, the effective amount of energy that is charged by the CS must be "certified" at the end of the transaction, so that it is possible for all the parties involved to rely on the certified information and to be sure that is the correct amount of energy supplied.

One solution would be to let the EV charge the maximum amount of energy it is willing to charge, and then let the DSO sign the effective amount of energy charged at the end of the transaction. This would be possible since the DSO has access to the smart metering data for each deployed CS, and could retrieve the information about the energy supplied by a CS in a charging transaction simply by having the information about the CS involved and the transaction timestamp.

Nevertheless, the solution has two main issues. The first issue is that usually, CSs have a smart meter measuring the activity of the entire CS, and not of single sockets. This means that in cases in which two concurrent charges are taking place at the same CS, a DSO cannot distinguish between the amount of energy supplied in each of the charging sessions. The second issue is that to certify the amount of energy charged by a CS, the DSO needs explicit information about the CS involved (and hence its location). This information is not relevant for the DSO to verify the fulfilment of energy flexibility requests, which only require information about the ER involved, the amount of energy supplied, and the energy district of the CS providing energy.

Hence, the solution proposed in this thesis adopts a different approach by letting the CS and the EV mutually sign micro-transactions within the larger charging transaction. The total amount of energy supplied is divided into small blocks that are signed by both the CS and the EV. Only when the EV has signed a new block, the CS will supply the next amount of energy specified in the block, and only when the CS has supplied the amount of energy asked, the EV will sign a new block asking for more energy. In this way, by reducing the risk affecting each party (e.g. by not being paid for the energy supplied or not receiving the energy that has been paid for), the charging interaction can be interrupted at any time by either party. The sudden interruption of an energy transaction does not affect the security and/or privacy of the transaction since the possession of the signed micro-transactions is enough for both parties to prove that a certain amount of energy has been provided by the CS to the EV.

Furthermore, for the same reasons explained in the previous subsection, with the approach taken in the final architecture, the DSO does not represent a single point of failure, as it would if it had to sign each charging transaction. The parties do not rely on the DSO to validate the amount of energy supplied, hence the charging transaction is solely dependent on the correct working of the CS and the EV.

5 Architecture Design

This section describes the use case, the architectural choices taken, and the details of the architecture implemented in this thesis.

5.1 System description

The main actors of the system have been presented in the introduction, but here they are presented in more details.

- **DSO**: controls the grid network, and needs to avoid reverse power flows by offering incentives to ERs if they can use a certain amount of energy for their customers to charge, within a certain district and time frame. These requests for energy flexibility are published on an energy marketplace in a form of auction where ERs compete with each other based on the number of rewards requested (i.e. the ER requesting the lowest amount of incentives as a reward wins the auction).
- **CSO**: owns the CSs and makes them available to different ERs so that they can offer their services to their customers via those CSs. Each time a CS offers a service to an EV customer of an ER, the owner of that CS is paid a service fee. The actual payment of the fees can be performed on a per-transaction basis or at regular intervals.
- **CS**: is owned by one CSO and can be used by the customers of one or several ERs at the same time, depending on the business agreements. They interact with the EVs using machine-to-machine communication and then report the details of the transactions back to the CSO system.
- **ER**: offers charging services to several EVUs. They compete in the energy marketplace to win energy auctions and obtain rewards in case of successful fulfilment. They collect transaction details from the CSO owning the CS involved in each transaction and bill their customers at the end of the billing period. They rely on the CSs from several CSOs to offer their services, even within the same energy district.
- **EVU**: interacts with CSs to make use of the charging service. They can be customers of multiple ERs, even within the same district. They obtain discounts or rewards from an ER in case they contribute to an energy flexibility request fulfilment by that ER, i.e. they charge in a specific district during a specific time frame.
- **EV**: interacts with the CS during a charging transaction, after the transaction has been initiated by the EVU. It also communicates regularly with the EVU's device to exchange information including new DIDs and charging credentials.

5.1.1 Assumptions

The following assumptions have been made during the design of the architecture:

- **GENERAL**:
    1. **GEN-A1**: DSO, CSO and ER are separate independent entities, i.e. no actor fulfils more than one role at any given time.
    2. **GEN-A2**: None of the parties involved, especially the ERs because of their knowledge of the EVU's identities, collude with each other to break the privacy of EVUs.
- **DSO**:
    1. **DSO-A1**: DSOs have knowledge about the precise location and the real-time energy consumption for each CS deployed within their coverage area, regardless of the CSO owning them.
    2. **DSO-A2**: DSOs do not behave unfairly towards either the CSOs or the ERs since unfair behaviour would make it impossible for them to make use of the ecosystem to release excess energy in the long run.
    3. **DSO-A3**: DSOs have a publicly accessible (e.g. stored on a distributed ledger) identity defined by a DID and its associated DID Document. This document defines the keys used by the DSO to sign credentials and the keys used by the other parties to verify those signatures. The DID Document also describes the DSO endpoint or endpoints that the other parties need to use to communicate with it.
- **CS/CSO**:
    1. **CSO-A1**: CSOs know the precise location and the real-time energy consumption for each CS they own.
    2. **CSO-A2**: CSOs are paid by ERs for each of their customers charging at any of the CSs owned by that CSO.
    3. **CSO-A3**: CSOs and their CS have an interest in offering as much energy as possible to EVs. CSs are assumed to never deviate from the protocol in a way that would lead to a valid charging event being classified as *uncompleted*.
    4. **CSO-A4**: CSOs have a publicly accessible (e.g. stored on a distributed ledger) identity defined by a DID and its associated DID Document. This document defines the signing keys used by the CSO to sign credentials and verification keys used by the other parties to verify those signatures. The DID Document also describes the CSO endpoint or endpoints that the other parties need to use to communicate with it.
    5. **CSO-A5**: CSs can deviate from expected behavior by validating charging transactions that have never taken place.

6. **CSO-A6**: CSs have enough processing power to perform public key cryptographic operations.
7. **CSO-A7**: CSs can communicate directly with EVs (e.g. using Wi-Fi Direct or Bluetooth).

- **EVU**:

1. **EVU-A1**: EVUs can deviate from expected behavior by charging without registering the charging transaction, causing him/her to charge for free.
2. **EVU-A2**: EVUs can trigger charging events with CSs and to communicate with their EVs via the usage of a mobile app.

- **EV**:

1. **EV-A1**: EVs have enough processing power to perform public key cryptographic operations.
2. **EV-A2**: EVs are able to communicate *directly* with CSs (e.g. using Wi-Fi Direct or Bluetooth).
3. **EV-A2**: EVs are able to communicate with the mobile device of their user.

- **ER**:

1. **ER-A1**: Multiple ERs can have agreements with a DSO within the same district. This extends the scenario implemented in the SOFIE Italian pilot, where the auction for an energy flexibility request can be won only by one ER.
2. **ER-A2**: ERs have a publicly accessible (e.g. stored on a distributed ledger) identity defined by a DID and its associated DID Document. This document defines the signing keys used by the ER to sign credentials and verification keys used by the other parties to verify those signatures. The DID Document also describes the ER endpoint or endpoints that the other parties need to use to communicate with it.

5.2 Architecture specification

This section presents the different interactions designed in the architecture and the message flow between the parties involved at each step. The interactions on the energy marketplace are outside the scope of this thesis, so the architecture will not include those interactions, even though the content of each energy flexibility request (i.e. the amount of energy to charge, the district information and the time limit) has been considered in the design of the architecture. For an ER to be able to prove that it fulfilled an energy flexibility request published by the DSO, it must be able to prove that a certain amount of energy has been charged within a specific district within the specified time frame. No other information is required, such as the specific

CS or the specific user involved. This proof must contain a set of transactions that, when evaluated, sum up to the total amount of energy that needed to be taken out of the grid, as requested by the DSO.

As specified in the requirements in Section 1.2.1, transactions must be verifiable, meaning that each party can independently verify that each transaction has taken place. The DSO must be able to verify that the total amount of energy requested to be released has indeed been supplied by the ER. The ER must be able to verify that its customer EVUs have charged within a specific district, and also to identify the CSO of each CS involved in the charging transactions, for billing purposes. The CSO must be able to identify the ER of which the EVU involved in each charging transaction is a customer of, to bill it accordingly.

The architecture relies heavily on the usage of single-use DIDs to enable security and privacy between any two parties. For instance, in a charging transaction involving a CS and an EV communicating with each other, the two entities each create a temporary pairwise DID. Using a different pairwise DID for each communication eliminates possible correlation attacks so that it is theoretically impossible for an external attacker to derive whether two different pairwise DIDs belong to the same entity.

5.2.1 UC-1: EVU on-boarding

The interaction, shown in Fig 10, takes place every time an EVU signs a contract with an ER that needs to initiate a new customer-provider relationship. Once the two parties sign the contract (this event is not part of the protocol, and it can happen either online or offline) the on-boarding of the EVU can start. **This process is repeated for each ER the EVU becomes a customer of.**

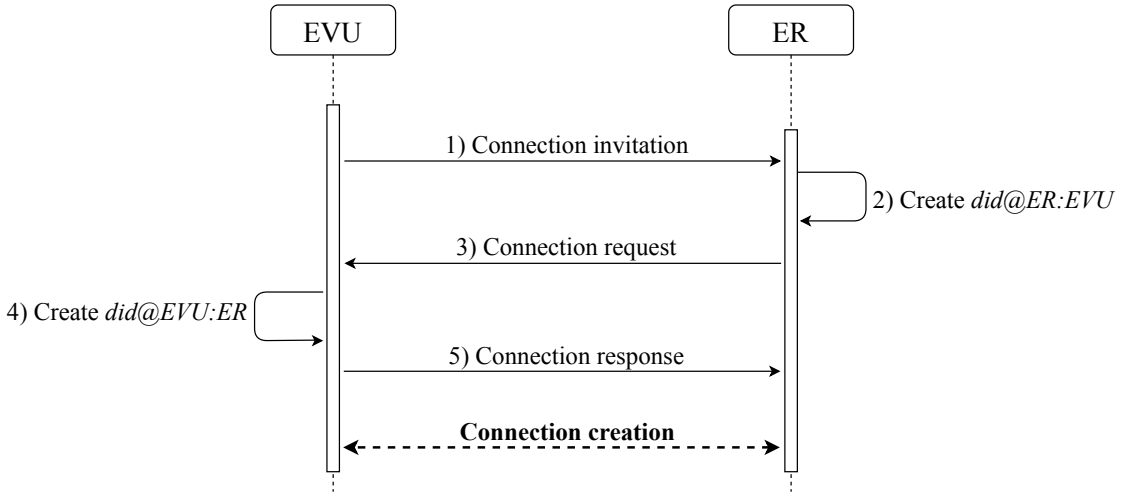

Figure 10: Message flow for on-boarding a new EVU as customer of a specific ER (UC-1).

The message flow is the following:

1. *Connection invitation*: this step involves the EVU communicating to the ER his/her intention to be issued a digital customer credential. Since the EVU and ER still do not know each other's endpoints, this message must be sent by the EVU to the ER over a traditional communication channel, preferably a secure one, to a well-known endpoint controlled by the ER. For instance, this message can be sent as a GET request to an HTTPS endpoint owned by the ER.

2. *Create did@ER:EVU*: upon receiving a connection invitation, the ER endpoint creates a new, unique pairwise DID that will be used only when communicating with the EVU. This DID will then be used by the EVU to encrypt future traffic addressed to the ER by using the ER public encryption key specified in its DID and to authenticate messages received by the ER by using the ER public verification key, also specified in the DID.

3. *Connection request*: the ER sends a connection request to the EVU. This message is sent to the EVU over the same channel used by the EVU in Step 1. The connection request message includes information about the DID created in Step 2 by the ER.

4. *Create did@EVU:ER*: upon receiving a connection request, the EVU endpoint creates a new, unique pairwise DID that will be used only to communicate with the ER endpoint.

5. *Connection response*: at this step, the EVU knows the pairwise DID used by the ER (received in step 3). The EVU can then encrypt the traffic using the public encryption keys specified in the DID received by the ER and sign it with the private signing key associated with its own pairwise DID. Furthermore, the DID also specifies the endpoint that the ER will use in the communication with the EVU, hence the EVU can start sending all the future messages, including the connection response, addressed to the ER to that endpoint, for instance a REST or SOAP service. The connection response contains the details about the EVU's DID, which also includes a public verification key that the ER can use to check the signature of the connection response message just received and of all the future messages sent by the EVU. **After this step, an end-to-end encrypted connection has been created between EVU and ER, and the communication will take place between the endpoints specified in the DIDs.**

5.2.2 UC-2: CS on-boarding to CSO

The interaction, shown in Fig 11 takes place every time a new CS is deployed by a CSO and if the ownership of a CS changes from one CSO to another. The message flow is very similar to the previous case of on-boarding an EVU and his/her EV. Furthermore, it is assumed that a credential definition for the credential issued to

CS by the CSO has already been created and is publicly accessible. The goal of this interaction is to create an encrypted communication channel between the CS and the CSO endpoint and *to generate a credential for the CS stating that it is owned by the CSO*.

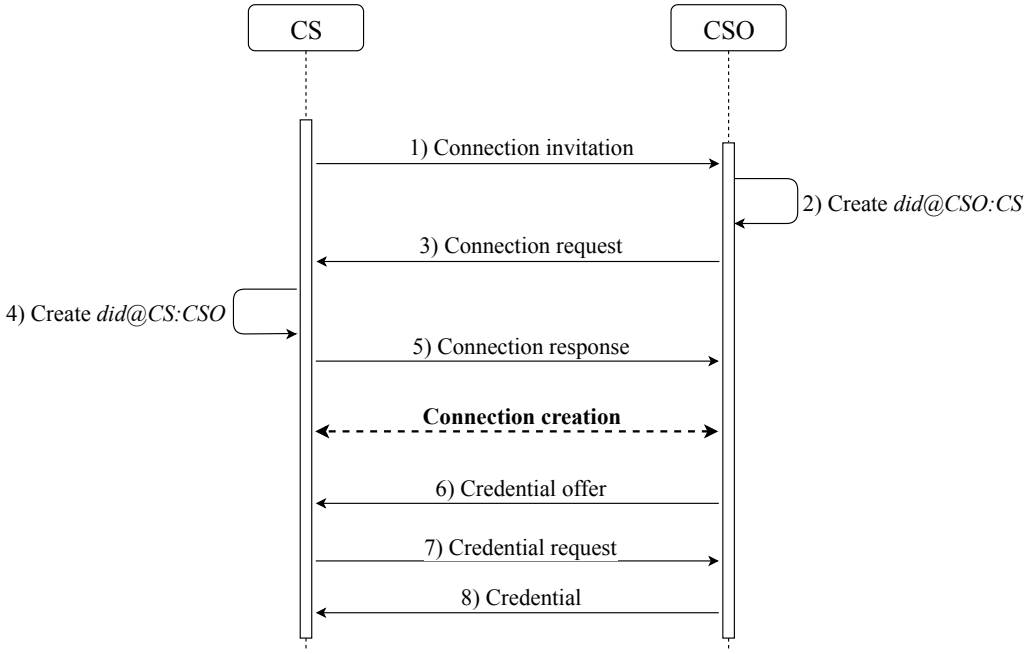

Figure 11: Message flow for on-boarding a new CS owned by a specific CSO (UC-2).

1. *Connection invitation*: this step involves the CS communicating to the CSO its intention to be on-boarded into the ecosystem. As with the EVU on-boarding, this message can be sent over any communication channel, preferably a secure one.

2. *Create did@CSO:CS*: the CSO endpoint creates a new, unique pairwise DID that will be used only when communicating with the CS endpoint.

3. *Connection request*: the CSO sends a connection request to the CS via the same channel used in step 1.

4. *Create did@CS:CSO*: upon receiving a connection request, the CS endpoint creates a new, unique pairwise DID that will be used only when communicating with the CSO endpoint.

5. *Connection response*: the CS sends a connection response encrypted with the CSO pairwise DID encryption key and signed with the private signing key associated with its pairwise DID. **After this step, an end-to-end encrypted connection has been created between CS and CSO, and the communication will take place between the endpoints specified in the DID.**

6. *Credential offer*: the CSO creates a credential offer for the CS. A credential offer contains references to the credential definition and the credential schema which the credential will conform to. Hence, a credential offer is created by the CSO to communicate to the CS what type of credential the CSO can issue, and what are the attributes that will be included in the credential. In this case, the credential will contain information about the owner of the charging station.

7. *Credential request*: the credential definition and schema included in a credential offer can be validated by the CS, i.e. the CS can make sure that the attributes in the credential are the ones the credential is supposed to have. Upon successful validation of the credential offer, the CS requests the credential with a credential request.

8. *Credential*: the CSO generates and issues the credential to the EVU. The credential is depicted in Listing 1, in Appendix A, and is signed with the signing key associated with the CSO public DID.

5.2.3 UC-3: CS registration with DSO

The interaction takes place after the CS has been correctly on-boarded by the CSO, which happens in UC-2. The interaction is shown in Fig 12. The registration happens between the CS and the DSO and is needed by the CS *to receive a credential containing the district information that the CS will then use when interacting with EVs*. The assumptions are that the schema and definition of the new credential that is going to be issued are already stored and publicly accessible.

1. *Connection invitation*: same as the previous use case.

2. *Create did@DSO:CS*: same as the previous use case.

3. *Connection request*: same as the previous use case.

4. *Create did@CS:DSO*: same as the previous use case.

5. *Connection response*: same as the previous use case.

6. *Proof request*: the DSO sends to the CS a proof request asking to show a valid credential defining the ownership of the CS. The proof also asks to include other CS-specific information (e.g. the authentic value of the smart meter identifier, perhaps obtained by a trusted module part of the CS) so that the DSO can issue a credential containing the right district identifier. Even though the proof of ownership is not strictly required by the DSO to issue a credential, it mitigates some types of attacks that might take place during the registration of a CS. For instance, if no proof is requested, any actor with a valid CS identifier, or any other details used at registration time, might register itself to the DSO as a valid CS in a specific district. On the other hand, the request of a valid proof about the CSO owning a CS allows the DSO to be sure that the entity asking for a district credential is indeed a valid CS. Hence, the risks

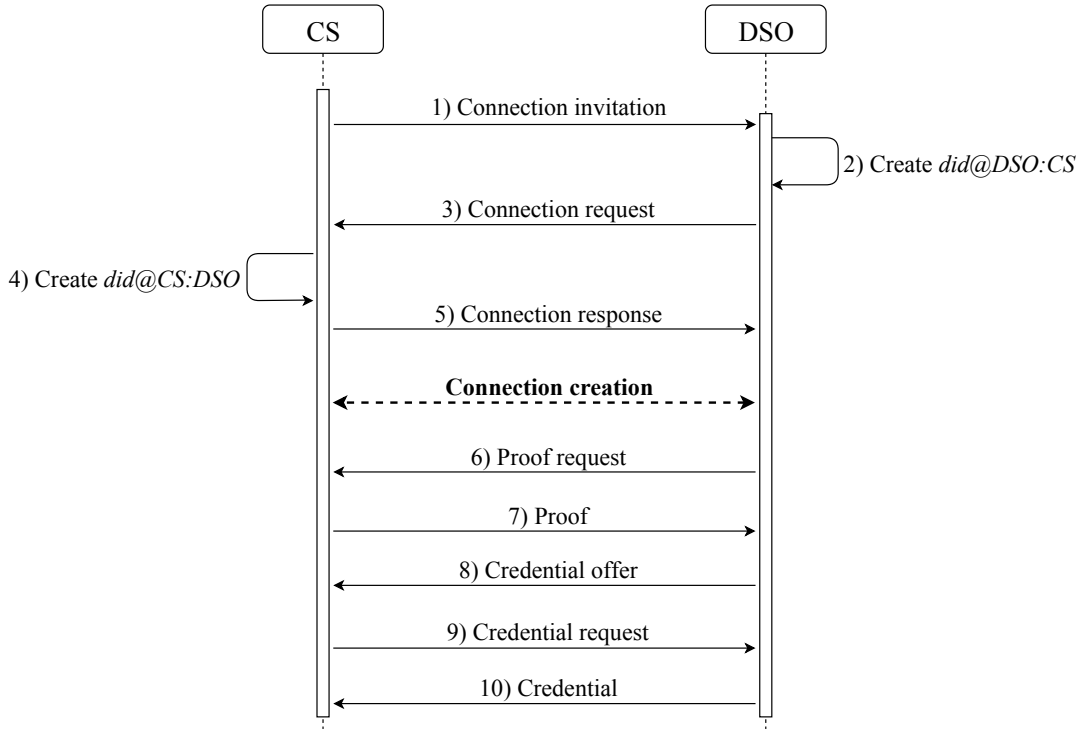

Figure 12: Message flow for registering a new CS as located within a certain district (UC-3).

of impersonation attacks are reduced, as long as the CS itself does not act maliciously for some other reasons.

7. *Proof*: the CS creates a verifiable presentation fulfilling the proof request and sends it back to the DSO for verification.

8. *Credential offer*: the DSO verifies that the CSO owning the CS is among its customers and, if the proof is successfully verified, the DSO sends to the CS a credential offer that makes it possible for the CS to obtain the district-identifying credential.

9. *Credential request*: same as previous cases.

10. *Credential*: the DSO sends to the CS the verifiable credential containing the district identifier for the CS. This credential will be used to interact with EVs. The credential is depicted in Listing 2, in Appendix A, and is signed with the signing key associated with the DSO public DID.

5.2.4 UC-4: Charging credentials generation

The purpose is for the EVUs *to collect credentials from the ERs that will allow them to charge their EVUs* during the day at the CSs that the ERs have an agreement with. The process can be repeated for each needed ER, and it uses the encrypted

connection channel that had been established upon EVU on-boarding in UC-1 with each of the ERs. Since these credentials are short-lived (e.g. expire at the end of the day in which they have been issued), ideally this transaction could take place at the beginning of each day in which the EVU thinks he/she is going to make use of them. There is no consequence if the charging credentials are issued and are never used or expire. As with UC-2 and UC-3, it is assumed that all the ERs issuing charging credentials have a public identity which is used to sign the credentials and later to verify those signatures. The flow is shown in Fig 13.

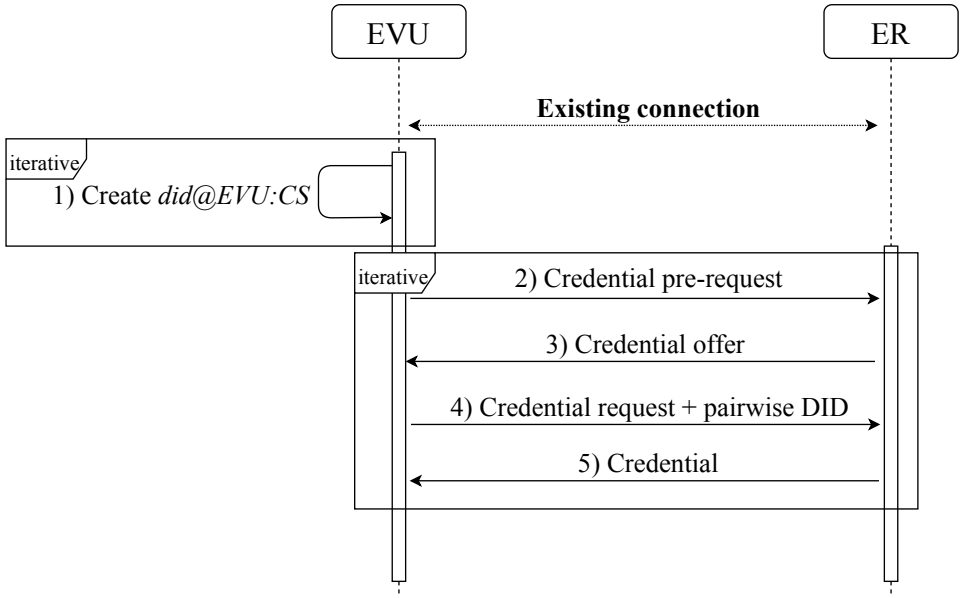

Figure 13: Message flow for an EVU to obtain charging credentials from its ER (UC-4).

1. *Create did@EVU:CS*: this is one of the fundamental steps to keep the identity of the EVU protected. The EVU generates as many pairwise DIDs as the number of credentials he or she is willing to be issued. The credentials can be used in charging transactions with CSs to prove the validity of the customer-provider relationship between the EVU and the ER. **Even though the EVU is not forbidden to re-use the same DID and credential twice, this is highly discouraged to reduce the probabilities of successful correlation attacks.**

2. *Credentials pre-request*: for each credential that the EVU wants the ER to issue, the EVU pings the ER endpoint (specified in the ER pairwise DID for the EVU-ER connection established in UC-1) to trigger a new credential offer-request-issuance iteration. No specific information is sent at this step.

3. *Credential offer*: the ER creates a new credential offer for each credential that needs to be issued. This is mainly done to avoid replay attacks (each credential offer has a nonce attribute). Nevertheless, the whole process can be achieved by

exchanging only one, larger message, containing the information about all the credentials that will be issued. In this case, the credential pre-request message at the previous step needs to specify the number of credentials that the EVU wants the ER to issue.

4. *Credential request + pairwise DID*: the EVU requests the ER to issue a charging credential (credential request) having as subject one of the pairwise DIDs generated at Step 1.

5. *Credential*: the ER issues the credential linked to the DID specified in the previous step. The credential will contain the information that the credential subject DID is a customer of the specific ER, as shown in Listing 3, in Appendix A. The credential is short-lived and not revocable.

5.2.5 UC-5: Charging event

The interaction, shown in Fig 14, takes place every time an EV needs to charge at a CS. Here the assumption is that there is an offline communication channel (either unicast or broadcast) between the CS and the EV. **Furthermore, if the communication channel relies on static identifiers (e.g. MAC addresses), such addresses need to be randomised before each interaction starts, so that the same EV cannot be linked by the CS and vice versa.** It is also assumed that EVU and EV can communicate via a previously-secured communication channel (e.g. via Bluetooth pairing) and that they have also agreed on which of the several DIDs previously generated and registered with the ER during interaction UC-4 will be used in this interaction.

The charging interaction relies on a series of key points. First, the protocol for charging is manually triggered by a user. This is done because it reduces the vector of attacks by making the communication bootstrapping between EV and CS more secure. It is also done so that the CS does not need to be constantly advertising information, but can advertise on-purpose whenever triggered by a valid (as explained later) human interaction.

Second, even in case the rest of the messages (other than the first human-triggered interaction) are broadcast between the EV and the CS, they are encrypted using each other's public encryption key, as specified in their DIDs. In this way, malicious actors cannot decrypt the content of the messages being broadcast, even though they might be able to capture them. This is also why the first step in the protocol, the human trigger, happens using a near-range communication channel (e.g. QRCode scanning or NFC tapping). In this way, there is no initial information advertised in clear, and the privacy of both the EV (and its EVU) and the CS can be guaranteed.

Third, during the actual exchange of electrical energy from the CS to the EV, there is a continuous exchange of signed data between the two parties. In this way, an EV can send signed requests for small units of energy (e.g. every kWh), and the CS can return signed responses that the energy required has been provided, similarly to the solution presented in the POPCORN protocol paper [34]. If either of the parties stops following the protocol, the communication can be interrupted, while

the total supply of energy up to that moment can still be proved due to the signed messages that the two entities have previously exchanged. The EV can suffer a small loss if the last request is not fulfilled by the CS, i.e. if the CS does not supply energy for the last signed request by the EV. The maximum amount of the loss depends on the amount of energy supplied within each signed block.

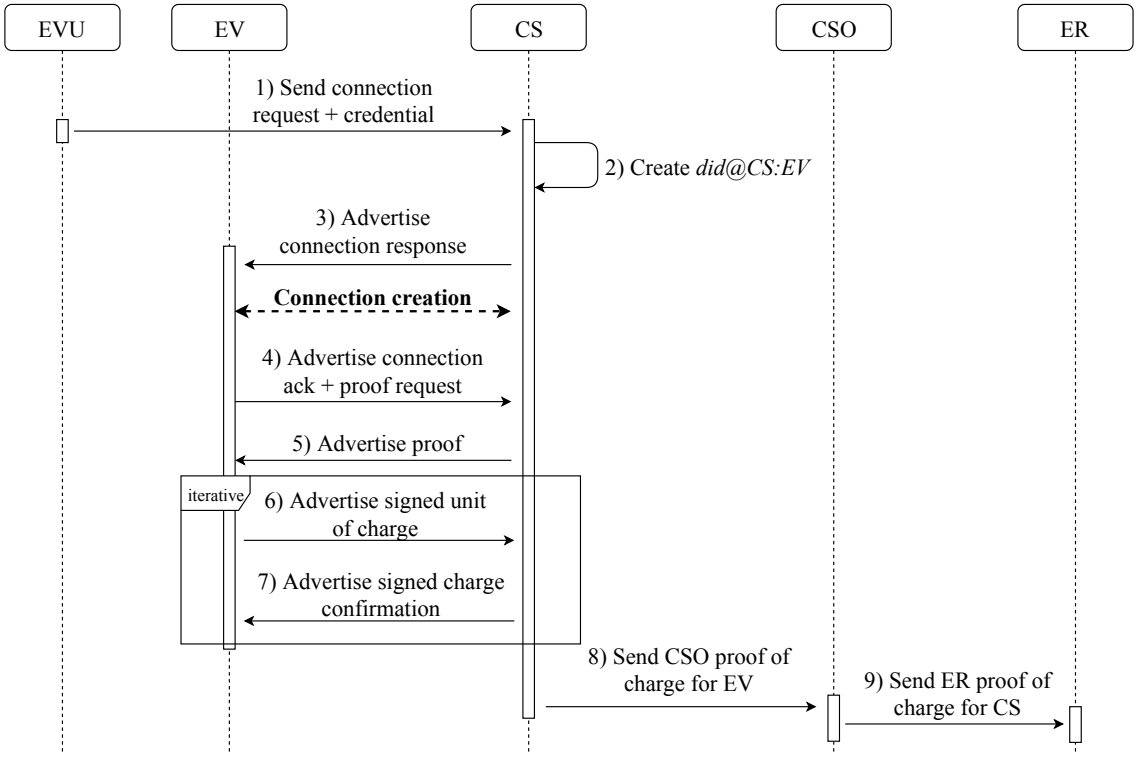

Figure 14: Message flow for a charging event (UC-5).

1. *Send connection request + credential*: this step requires the physical interaction of the EV owner that needs to get close to the CS and trigger the protocol. The action could be the reading of a QRCode by the CS or the tapping of an NFC device from the EVU to the CS reader, assuming the reader is properly secured. Such information is then embedded into the connection request and encoded in a format that depends on the communication channel used. In order to reduce the number of messages exchanged, and also to reduce the chances for malicious users to interact with a CS without being a customer of any of the ER authorised to use the CS, the message contains also the credential proving that the used DID has been issued a credential by an ER which has authority to use that CS to offer its services. Hence, the connection request (to create an encrypted communication channel) along with the credential (to short-circuit interaction with malicious users) is sent by the EVU to the CS using a near-distance communication channel, making it difficult for malicious users to alter or read the content of the message, which is exchanged in clear.

2. *Create did@CS:EV*: once the protocol has been initiated, the CS creates a new pairwise DID used only within the scope of the current charging event.

3. *Advertise connection response*: since the CS has the information about the DID used by the EV in the interaction, it can advertise encrypted data using the encryption key specified. In this step, the CS advertises the connection response, which makes it possible for the EV to assign a temporary identity to the CS (based on the temporary DID the CS has generated for this interaction), encrypt the traffic broadcast but addressed to it, and authenticate the traffic broadcast by it.

4. *Advertise connection ack + proof request*: from now on, a logical communication channel has been set: the traffic, even though broadcast, is encrypted and cannot be altered/understood by third parties. At this point, the EV sends back a connection ack along with a request for proof from the CS proving information about its owner as well as the district which the CS belongs to.

5. *Advertise proof*: the CS answers back by building and sending the requested proof, built from the credential obtained at on-boarding time by the CSO (UC-2) and registration time by the DSO (UC-3).

6. *Advertise signed unit of charge*: **this step, as the next one, is repeated several times, assuming both CS and EV successfully verify the information sent to each other in the previous iteration.** Each message is denoted with $S_{REQ}i$, where i represents the position of the message within the collection of messages sent by the EV to the CS. Each $S_{REQ}i$ contains the information about both DIDs used in the interaction, the amount of energy requested in this iteration, the timestamp, and the district information. The information is signed by the EV by using the private signing key associated with its pairwise DID. At the end, when the n messages are collected by the CS, it has undeniable proof of the total amount of energy requested by the EV during the specific charging interaction.

7. *Advertise signed charge confirmation*: each message is denoted with $S_{RES}i$. Each $S_{RES}i$ contains the same information as $S_{REQ}i$, but it is signed by the CS instead of the EV. When collected by the EV, it has undeniable proof of the total amount of energy supplied by the CS during the specific charging interaction.

8. *Send CSO proof of charge for EV*: once the EV has charged the amount of energy it needed, the communication between the CS and the EV interrupts. The CS then sends to the CSO endpoint all the n $S_{REQ}i$ and $S_{RES}i$, the charging certificate EV_{ER}, and the two proofs built to prove CSO ownership and district information. The information is not sent directly to the ER both because of possible information leakage deriving from an Internet-based communication, and also because there is no direct trust relationship between the CS and the ER, since the single CSs are indistinguishable to the ER, as this is one of the

goals of the protocol. The transaction-related information is stored by the CSO for possible dispute resolution, since it is enough to prove that the transaction took place (the EV has signed several micro-transactions) and that the EV was a customer of the ER (the charging certificate EV_{ER} proves it).

9. *Send ER proof of charge for CS*: upon receiving the transaction details from the CS, the CSO forwards them to the ER for billing purposes.

At the end of the charging transaction:

- **ER**: can prove to the DSO that a certain amount of energy (the sum of the energy charged in all the micro-transactions) has been charged in a district (the proof built by the CS about its district identifier) within a specific time frame (the timestamp in the micro-transactions). Such proof does not leak any information about the specific CSs or EVs involved in the charging interactions.
- **CSO**: can prove and bill the ER for the service since it can prove that a certain amount of energy (the sum of the energy charged in all the micro-transactions) has been charged to an EV belonging to the ER (the charging certificate presented by the EVU in Step 1) at a CS belonging to that CSO (the proof built by the CS, which has also been verified by the EV during the transaction).

Furthermore, being serialised in a machine-readable format, the verification of the transactions can be performed in an automated fashion, e.g. by a smart contract, which could then issue the agreed amount of rewards to the ER.

5.3 Privacy considerations

As shown in Fig. 15, the privacy of the solution is ensured by the fact that the system is logically divided into three areas of knowledge: the area of CSs and CSOs, the area of EVs/EVUs and ERs, and the area of the DSOs. By using temporary DIDs, and making the binding of those DIDs to real identities known only to the entities within the same area of knowledge, the privacy of the identities can be guaranteed if this binding is not made known to entities in other areas. For instance, after every charging transaction, a CS sends to its CSO the details about the charging transaction.

We assume that the CSO knows the real identity of the CSs it owns, and by having access to the transaction data for each charging event, it also knows the temporary identity that each of the CSs has used. Hence, the CSO can exactly say which CS was involved in which charging transaction. Nevertheless, neither a CS nor a CSO can derive the identity of the EVU involved in the transaction.

In the same way, an ER issues charging credentials to temporary DIDs created by its customer EVs, hence it has complete knowledge about the mapping between an EVU identity and all the DIDs generated (to which the credentials are issued). When receiving charging transaction details from the CSO, the ER is then able to derive the real identity of the EVUs involved in the charging transaction, since they will be

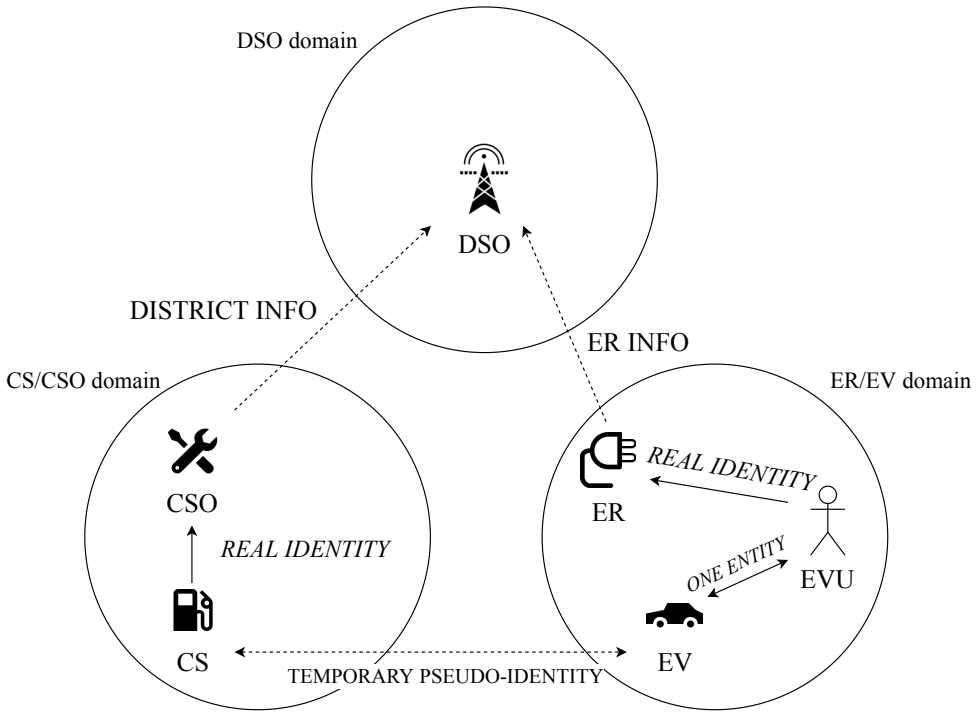

Figure 15: The different knowledge domains of the use case described. Solid arrows indicate personally identifiable information. Dashed arrows indicate information that do not relate to a single entity (either CS or EV).

billed accordingly, but has no knowledge about the CS involved. Since single-use DIDs are used, the ER cannot link two different charging events to the same CS, but only determine the district in which the charging event has taken place.

With regard to the DSO, from the transaction details it receives, it cannot derive any useful information about the real identity of the CS nor of the EV (and hence of its EVU) involved, but only the district in which the transaction took place and the ER the EVU involved in the transaction is a customer of. **It is worth pointing out that the DSO can still identify the specific CS involved in each charging transaction by comparing the consumption data from the smart meters installed on those CSs with the information in the charging transactions it receives. Nevertheless, such information does not affect the privacy of the EVUs, who are still protected from both the CSO and the DSO, as long as his/her ERs do not collude with them.** The different knowledge that each party gains at the end of a charging event is shown in Table 3.

This separation of concerns can be compared to the Principle of Least Privilege (PoLP) widely used in the IT security field [59]. The *Principle of Least Privilege* requires that every subject (e.g. a process, a program or a user) must be able to access only the information and the resources that are necessary to accomplish its task.

In the architecture presented, by preventing cross-domain information flow, we ensure that each subject has only limited knowledge of the system transactions and

-	CS	EV	TRANSACTION INFO
CSO	***- CS real identity***	- ER providing the charging service - EV pseudo-identity	- charging amount - charging time - district identifier
ER	- CS pseudo-identity - CSO owning the CS providing the charge	***- EV real identity***	- charging amount - charging time - district identifier
DSO	***- CS real identity***	- EV pseudo-identity	- charging amount - charging time - district identifier

Table 3: Table representing the knowledge that the main actors of the system, DSO, CSO, and ER, acquire after a charging transaction is completed between a CS and an EV. Rows represent the entities acquiring new information, while columns represent entities about which new information is obtained.

cannot threaten the privacy of the subjects (or entities) for which it is not responsible.

This is why the assumption GEN-A1, presented in Section 5.1.1, has been made about the different parties not colluding with each other: if there is cross-border information flow of personally identifying information, each party can obtain knowledge of the entire system, since the knowledge it gets from the other domains can be combined with its knowledge.

6 Analysis

This section evaluates how the solution proposed in the previous section satisfies the privacy and business requirements listed in Section 1.2.1. Furthermore, this section contains answers to the research questions presented in Section 1.3.

6.1 Privacy and business requirements

Overall, the use case has been made very general and flexible and therefore it can be implemented in several different scenarios: CSs can change their owner (CSO) during their lifetime; EVs can be customers of several ERs simultaneously; ERs can sign agreements with several CSOs at the same time, even within the same district; CSOs can have agreements with several ERs at the same time: the agreements specify in which energy districts the ERs can use the CSs owned by the CSO to provide their services.

Furthermore, when a party obtains some information about a charging transaction, such information must be authentic, i.e. it must be verifiable and the parties involved cannot repudiate their participation to the transaction.

6.1.1 PR1 + BR1

PR1*: The DSO MUST NOT be able to identify specific EVUs.*

BR1*: The DSO MUST be able to understand how much each ER has participated in maintaining grid stability in relation to the agreements for the energy flexibility requests.*

The PR1 has been set to protect the privacy of the EVUs against the distribution system operator (DSO). The only concern of the DSO is to reward the energy retailers (ERs) that successfully fulfil an energy flexibility request. For this reason, the only information they need to have access to, for each charging transaction, is the amount of energy charged, the district in which the charging event took place, and the ER allowing the EVU to charge in this transaction, i.e. the ER providing the charging service to the EVU. If the DSO obtains such information, the architecture would also be compliant with the business requirement *BR1*. Even though the descriptions of the various interactions (UC-1 to UC-5) do not describe the verification process by the DSO, the architecture and the information flow have been designed in a way that makes the process relatively straightforward to implement.

The proposed solution fulfils PR1 by only granting the DSO access to a subset of the whole energy transaction information, which is sent by the ER willing to prove its contribution to the energy grid balancing and hence to obtain the agreed rewards. One element of information that the DSO has access to is the amount of energy charged during the transaction. This information is double-signed by both the CS and the EV involved in the transaction (in steps 6 and 7 of UC-5). The DSO can be sure the signatures are authentic because for each charging interaction it also receives,

from the ER, the certificates and the proofs used in the transaction. Specifically, the DSO receives the proof built by the CS proving to the EV its district identifier and the charging certificate (issued by the ER) used by the EV to show to the CS it has the authorisation to charge at that CS.

By matching the two DIDs (of the EV and the CS) used to sign the charging amounts with the ones used both in the district proof and the charging certificate, the DSO can be sure that 1. the CS involved in the transaction is located in a specific district and 2. the EV involved in the transaction belongs to a specific ER.

The solution is designed to allow the DSO to verify contributions by the ERs without having access to the real identities of the EVs involved in the charging transactions. Hence, the verification process does not leak any information about the EV involved in the charging transaction other than the single-use DID used in the transaction which, if not re-used in any other transaction, does not make correlation or de-anonymising attacks any easier to implement. **This condition is true even if the DSO can identify the CS involved in the charging transaction by comparing the amount of energy charged in the transaction with the smart meter readings of the energy grid it has access to. Even by having information about the specific CS involved, the DSO still cannot identify the specific EVU, unless external tools are used (e.g. cameras that allow mapping CSs, timestamps and temporary DIDs to a specific car and registration plate).**

6.1.2 PR2 and PR3 + BR3 and BR4

PR2*: A CSO and its CSs MUST NOT be able to identify a specific EVU engaged in a charging event.*
PR3*: A CSO and its CSs MUST NOT be able to infer that the same EVU has taken part in two different charging events.*

BR3*: CSOs are paid by ERs based on the charging services provided through any of their CSs. For this reason, CSOs MUST be able to claim payments from ERs by proving the authenticity of the charging events involving such CSs and the ER customers.*
BR4*: CSs MUST always verify that an EV is authorised to perform a certain charging operation before letting the EV charge for the agreed amount of energy.*

The PR2 and PR3 have been set to protect the privacy of the EVUs against the CSO, assuming the ER and the CSO roles are not fulfilled by the same entity. The only information a CSO is supposed to obtain from a charging transaction is the amount of energy charged and the ER involved, through the EVU, in the charging transaction (for billing purposes). The information obtained by the CSO makes the architecture compliant with the business requirements *BR3* and *BR4*.

The proposed solution addresses PR2 and PR3 by using single-use DIDs and masquerading the network identifiers used in the communication between the CS and the EV. These measures allow the EVUs to hide their real identity and make it

very difficult to implement correlation attacks since each new interaction will involve a new DID and new network identifiers for their EVs.

Nevertheless, the CSO can still obtain reliable information regarding the transaction, i.e. it can be sure that an EVU belongs to a certain ER by using the charging certificate presented by the EVU during step 1 of UC-5. In the same way, as done by the DSO, the CSO can also reliably obtain the total amount charged by verifying the double signature used for each micro-charge transaction event. Furthermore, similarly as described above, the use of external means to gather additional information about the charging EVUs, e.g. cameras, represents a potential threat but is out of the scope of this thesis.

6.1.3 PR4 and PR5 + BR2 and BR5

PR4*: A ER MUST NOT be able to infer that the same CS has taken part in two different charging events.*
PR5*: An ER MUST NOT be able to link its customers' charging events to specific CSs, but only to specific districts.*

BR2*: ERs bill their customers in a postpaid fashion, thus they MUST be able to monitor how much energy each EVU has charged over the billing period.*
BR5*: EVUs COULD get some kind of reward from their ER every time they contribute to the grid balancing.*

The PR4 and PR5 have been set to partially protect the privacy of EVUs against their ERs, with the same assumption that ER and CSO roles are not fulfilled by the same entity. The only information that an ER is supposed to obtain from a charging transaction is the amount of energy charged, the district in which the charging interaction took place (to charge the EVU involved and to possibly issue additional rewards) and the CSO owning the CS used for the charge by the EVU (to verify the authenticity of the bills from the CSO). The information obtained by the ER makes the architecture compliant with the business requirements *BR2* and *BR5*.

The proposed solution addresses PR4 and PR5 by using single-use DIDs and fine-grained certificates. These measures prevent the ER to 1. gather knowledge about a specific CS involved in the transaction (it only receives information about the district in which the transaction took place and the CSO owning that CS) and 2. derive patterns for its customer EVUs, since CSs always use different identifiers, making linkage across charging transactions harder.

Nevertheless, the ER can still obtain reliable information regarding the transaction, i.e. it can be sure that one of its customers has charged a specific amount of energy in a specific district. The specific customer is identified by the charging certificate used in the transaction, while the district information is obtained with the district proof sent by the CS to the EV in step 5 of UC-5. Furthermore, by verifying the signatures of the micro-charges, the ER can reliably obtain the total amount of energy supplied in the charging interaction.

6.1.4 PR6

***PR6**: Communication between the different parties (e.g. between EV and CS, or between CS and ER) MUST NOT leak more information than needed that would make correlation attacks against the EVUs easier (e.g. network-layer identifiers such as IP and MAC addresses).*

The PR6 has been set to prevent that all the measures taken to fulfil the previous privacy requirements are invalidated. As also noted in some of the related works analysed, side-channel information leaks deriving from static network identifiers have usually been underestimated. Nevertheless, in a privacy-sensitive context and, more importantly, in a privacy-preserving solution, such issues must be addressed. For this reason, in UC-5 (in steps 2 to 7), the communication between the CS and the EV hides the real network identifiers used by masquerading them and randomising them before every transaction.

From a performance point of view, masquerading network identifiers has no impact of any kind, while it has very important and positive privacy implications.

6.2 Research questions

The design of the proposed architecture has made possible to answer the research questions presented in Section 1.3.

6.2.1 RQ1

To what extent can the identity and location of the EVUs be protected, considering a scenario in which such identity must be known by the ERs they are customers of?

The proposed architecture shows that the parties that know, for business reasons, the identity of the EVUs involved in the charging transactions, such as the ERs, are only able to locate those EVUs at a district level. On the other hand, the entities that know the exact location of a charging event, e.g. by knowing the CS involved, are not able to retrieve the identity of the EVU involved.

As mentioned above, the DSO does not obtain any information about the EVU involved in a charging transaction, only district information, amount of charge, timestamp of the transaction, and the ER involved. Similarly, the CSO does not obtain any identifiable information from the charging interaction between one of its CSs and an EV, only the amount of charge, the timestamp of the transaction, and the ER involved in the transaction. The ER, on the other hand, does not obtain any information that could lead to the identification of a specific CS, since that information could be combined with the knowledge of the EVU involved in the transaction and could breach the privacy that EVU should have. In particular, an ER receives only information about the amount of charge, the timestamp of the transaction, the district in which the transaction took place, and the CSO owning the CS offering the charging service.

Hence, the location of the EVUs is known only to the ERs that are involved in charging transactions and only at the district level. On the other hand, the identity of EVUs is kept hidden to CSO and DSO and is known, for billing purposes, to the ER offering the charging service at the specific CS involved in the transaction.

6.2.2 RQ2

What are the possible relevant trade-offs between the privacy of an EVU and the business requirements of the other parties involved, i.e. DSO, CSOs and especially ERs?

Most of the business requirements that are desirable for EV charging scenarios similar to the one presented in this thesis have been listed in Section 1.2.1 and the presented solution addresses all of them.

Nevertheless, giving the possibility to an ER to know both the identity of the EVUs in the transactions and the districts in which the transactions take place does not yet entirely protect the privacy of the EVUs. Yet, the ERs would like to know at least the district in which their customers charge, for instance, to derive patterns of the most crowded districts and subsequently reaching agreements with more CSOs to increase the offer in those districts.

It is not possible to enhance the identity privacy guarantees for the EVUs until the business requirement BR2 is removed. The requirement specifies that the customers of an ER are billed in a postpaid fashion, and for this reason, their identities in each charging transaction need to be known to the ER. Since the ER is the only party in the system that knows the real identities of the EVs, it is also the one in control of the most sensitive information, thus the most sensitive target for attacks.

For this reason, a better privacy-preserving approach would require the EVs not to share their real identities with any other party, including the ERs, so that they would be responsible for preserving and managing their own identities.

To achieve this, a different billing scheme is required. One possible solution would be to use anonymous payment schemes, e.g. some types of cryptocurrencies, and to introduce a pay-per-use model where EVUs would be paying for their charges during each charging transaction. Technologies such as smart contracts are very helpful to automate the entire process so that each block of charge (in steps 6 and 7 of UC-5) is unlocked only after the EV has paid for it, as suggested in [62].

By using an anonymous payment system and a pay-per-use model, the need for the ER to know the identity of the EV involved in each transaction is removed. It is enough for EVUs to build a proof that they are valid customers of a specific ER and that they have paid for the charge to unlock the supply from the CS. The built proof will still be used by the CSO to bill the ER for the service offered, but the identity of the EV will remain unknown to all the parties involved and its participation in different charging events will not be tracked over time.

6.2.3 RQ3

How can the system be designed so that the DSO can reliably and automatically evaluate the contribution of each of the ERs to the grid stability, without getting access to the identities of the individual users charging?

Reliability is ensured by signing the charging data that is exchanged between the EV and the CS during an interaction. Furthermore, in order for the signatures to be verified, there is need for additional credentials and proofs showing that the entity signing the data is either a certified CS belonging to a specific CSO (that the DSO can then bill) or that the EVU is a customer of a specific ER (for rewards issuance and future dispute resolution).

The machine-readable structure of the information exchanged among the parties enables the automation of some of the processes. For instance, the verification of the transactions can be performed in an automated fashion, e.g. by a smart contract that could then issue the agreed amount of rewards to the ERs fulfilling energy flexibility requests.

6.2.4 RQ4

How can charging events be logged and/or stored without revealing any personal information so that it is still possible to rely on them for dispute resolution between parties?

Auditability is one of the main properties that a system enabling interactions among untrusted actors needs to guarantee. The relationships between the different entities are regulated with contracts, so these entities are incentivised to behave fairly most of the time, even though things might not always go as expected. Moreover, if there is no dispute resolution mechanism in place, malicious parties would be able to trick the system without being punished for doing so: a system in which such actors are not punished for their actions would additionally lack the trust of the honest parties that, as a result, would abandon the system.

One solution to solve disputes is to introduce a trusted central authority storing all the transactions information and use it whenever needed. This approach is not very different from the ones proposed in other research that has been analysed and suffers from all the problems discussed in Sections 3.1 and 3.2.

Instead, a decentralised solution could be implemented in several ways. For instance with the creation of a trusted storage space, e.g. a DLT, where all the charging transactions could be stored and from where they could be retrieved and verified, in case of disputes. Nevertheless, a decentralised storage solution raises several concerns around data privacy, particularly if the data stored represents sensitive information. Another approach might require each party to store and manage the information it has access to, and present it when required by the system or in case of disputes resolution.

The solution presented in this thesis enables the implementation of decentralised solutions, since the data shared between parties does not represent, if taken independently, any personal information. Specifically, the amount of the charge, the

transaction timestamp, the district identifier and the temporary identities of EV and CS cannot, alone, be used by the actors of the system to point to a specific transaction involving a specific EV and a specific CS. Furthermore, since all the data that is exchanged is signed (e.g. charging data, proofs, and credentials), all of the parties can verify how much energy has been charged in a specific charging transaction without the need of a trusted central authority guaranteeing the authenticity or integrity of the data.

7 Future Work

This section presents the work that will be completed in the future, such as an evaluation of the performances of the system and the automation of some of the interactions presented.

7.1 Implementation and performance evaluation

This thesis has presented the design of an architecture enhancing the privacy guarantees of solutions currently used to mitigate electricity grid balance issues via scheduled charging of electric vehicles (EVs). First, it enhances the privacy of EVs during their interactions with the electricity grid. Second, it allows electric vehicles users (EVUs) to contribute to the electricity grid stability without compromising their privacy. Furthermore, the analysis carried out on the architecture has highlighted that it fulfils all the privacy and business requirements presented in Section 1.2.1.

Nevertheless, no proof-of-concept (PoC) has been developed to measure and evaluate the performances of the proposed architecture. For this reason, the development of a PoC using the Hyperledger Indy 2.1 framework could validate the claim that the architecture and the usage of DIDs are suitable for the use case. Since Hyperledger Indy uses a DLT as the underlying storage, the transactions within each use case presented in Section 5.2 might be divided into the following groups: time taken by the entities endpoints to perform computations (both cryptographic and not); time taken by an endpoint to interact with the ledger to save/retrieve credentials, schemas, and definitions; and time taken by any two endpoints to communicate with each other. Specifically, the last operation can take place in two different ways, either over the Internet (as in the case of CS-CSO communication) or using machine-to-machine protocols such as Bluetooth and Wi-Fi Direct (as in the case of CS-EV communication).

An implementation would allow measuring and categorising the time spent in each operation within each transaction to identify possible bottlenecks. For instance, if the generation of new DIDs by a party were to be identified as a bottleneck, then single-use DIDs might be re-used in a small number of interactions before being replaced by a new one.

7.2 Application to the SOFIE Energy Marketplace pilot

Among the related research considered, the SOFIE Decentralised Energy Flexibility Marketplace pilot [63] implements a real-world scenario based on similar, albeit less general, business relationships between distribution system operator (DSO), energy retailers (ERs), which are defined as fleet managers (FMs) in the pilot, charging stations (CSs), and electric vehicle users (EVUs) with their electric vehicles (EVs). Hence, the application of the proposed architecture to the SOFIE Decentralised Energy Flexibility Marketplace pilot can provide useful insights about the performance of the system in a real-world scenario and not in a simulation, as is the case of the previous section.

Even though not considered in the architecture, privacy-preserving identity technologies like decentralised identifiers (DIDs) can also be used in the interactions between the different entities to prove claims about themselves and to obtain access to some resources or services. For instance, the identities and the associated credentials might be used in the energy flexibility marketplace to bind Ethereum addresses to verified identities, if the access to the marketplace is regulated.

7.3 Automatic flexibility request fulfilment verification

The architecture has been designed to support the automation of as many processes as possible. One of the interactions that would benefit the most from increased automation is, as also pointed out at the end of Section 5.2.5, the verification, by the DSO, of the fulfilment of an energy flexibility request by an ER.

At the end of each charging interaction, the relevant metadata is sent first from the CS to the CSO, and then from the CSO to the ER. To prove the fulfilment of an energy flexibility request, the ER needs then to show to the DSO the "receipts" of all the charging transactions satisfying a particular energy flexibility request published to obtain the agreed rewards.

Since all the data exchange is machine-readable, the verification process can be automated in a way that prevents malicious behavior from either the DSO or the ER. Specifically, a smart contract can be used to enforce that 1. the energy flexibility request has been satisfied by the ER and 2. the correct number of rewards are transferred from the CSO to the ER.

The deployment of a smart contract for each energy flexibility auction that has been won by an ER and hence closed makes sure that its logic cannot be altered after the deployment and that it is verifiable by all the interested parties (in this case the DSO and the ER). The business rules specified in the contract are enforced fairly thanks to the security provided by the underlying DLT network.

Furthermore, the usage of smart contracts reduces to a large extent the workload requested to the DSO since it does not need to maintain any active endpoint to verify that each energy flexibility request has been fulfilled by the winning ER. Instead, the DSO just needs to make sure (when an energy flexibility request is won by and assigned to an ER) that the deployed smart contract correctly implements the business rules for the rewarding mechanism. Then, it is the ER's job to submit the receipts of the charging event to the smart contract to receive the rewards, without the need for the DSO to intervene.

7.4 Adoption of a privacy-preserving payment scheme

As a further improvement to the privacy of EVUs, the system could implement a payment system that would allow EVUs to pay for their charges without revealing their identity to any other parties, ERs included. Most businesses today are still reluctant to accept payments different than the traditional ones, i.e. debit/credit cards or cash. Nevertheless, there are several privacy-preserving payment solutions, mostly based on blockchain, that would allow EVUs to pay their ERs for the services

they use following a pay-per-charge model. The adoption of a privacy-preserving payment scheme would allow EVUs to pay for their charges without revealing their identities to the ERs, which is what happens in the proposed architecture due to the business requirements.

8 Conclusions

This thesis has presented an architecture enhancing the privacy guarantees of solutions currently used to mitigate electricity grid balance issues via scheduled charging of electric vehicles (EVs). Specifically, the information about charging transactions has been divided into different *knowledge domains* so that only the minimum amount of information needed crosses any domain border.

Section 1.2.1 lists the privacy and business requirements the final solution had to fulfil, while Section 5 introduces the architecture and lists the assumptions made during its design. To make the architecture applicable to a wide set of use cases in the sector, only the minimum needed number of business requirements have been made. On the other hand, the privacy requirements set make sure that the solution properly addresses the privacy needs of the EVUs.

The thesis has also analysed previous research performed in the field. The main points resulting from the literature review are that:

- Most of the solutions rely on centralised components for some critical tasks such as key material distribution and identities verification.
- Most of the solutions do not properly address side-channel information leaks deriving from communications taking place over the Internet. Long-lived (e.g. IP addresses) and permanent (e.g. MAC addresses) identifiers are as dangerous to the privacy of an entity as other types of static system-specific identifiers.

For the above reasons, the architecture designed uses privacy-enhancing and decentralised technologies such as decentralised identifiers (DIDs) and verifiable credentials (VCs). Furthermore, the communication involving entities whose identity needs to be protected (i.e. EVs) and other parties is reduced to the minimum needed. The network identifiers used in these communications are properly masqueraded and frequently randomised so as not to compromise the additional security and privacy provided by DIDs and VCs.

Because of its similarity in terms of business requirements and relationships between the different actors, the SOFIE Decentralised Energy Flexibility Marketplace pilot [63], developed within the context of the SOFIE[18] project, has been used as the starting point for the development of the architecture presented in this thesis.

Nevertheless, as considered during its analysis, the pilot does not properly address the privacy needs of the EVUs and makes some assumptions that reduce the deployability of the resulting system in more general contexts. However, because of its similarity with the architecture designed, the pilot has been considered as a suitable candidate to implement (by using the Hyperledger Indy framework) and apply the architecture presented in this thesis and test its performances in a real-world scenario. Aspects of the pilot such as marketplace interactions, charging transactions, and verification of energy flexibility request fulfilment would all benefit from the adoption of the proposed solution, in terms of increased privacy for the EVUs, increased levels

[18] https://sofie-iot.eu

of automation for some processes and increased deployability for more general and complex use cases.

Generally speaking, the solution presented in this work solves the issue of having a trusted centralised authority for most privacy-sensitive operations. The presence of a central authority represents a sensitive target from the information security point of view, and it might also represent a bottleneck for the performance of the system. Still, even with no central authority knowing all or most of the interactions taking place within the system, the architecture has been designed to ensure authenticity and reliability of the information exchanged.

In particular, since the use case considered involves different independent parties (with some not sharing any kind of trust), the system still ensures that disputes can be fairly solved by simply relying on past transactions. For instance, a DSO can reliably verify the contribution that an ER has made to the grid balance by analysing charging transactions within a specific time scope involving the customers of that ER, without threatening the privacy of those customers.

By deploying the proposed architecture in real-world use cases, EVUs will still be able to benefit from the incentives that DSOs will still make available to balance their energy grid. Furthermore, EVUs do not have to give away their privacy anymore to benefit from those incentives, as is the case with many other service providers that make heavy use of user data to offer customised solutions. On the other hand, businesses such as DSOs, CSOs and ERs can profit in a more ethical way that does not profile end users and does not make them targets for profiling activities. Moreover, by decentralising the management of each user's information and keying material, charging interactions solely and entirely depend on the entities interacting, i.e. CSs, EVs and EVUs, without relying on the services provided by an external, third-party entity that could represent potential single points of failure for the system availability.

References

[1] Asmaa Abdallah and Xuemin Sherman Shen. Lightweight authentication and privacy-preserving scheme for v2g connections. *IEEE Transactions on Vehicular Technology*, 66(3):2615–2629, 2016.

[2] Kari Alanne and Arto Saari. Distributed energy generation and sustainable development. *Renewable and sustainable energy reviews*, 10(6):539–558, 2006.

[3] Christopher Allen. The path to self-sovereign identity. URL: https://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (visited on 2019-06-29).

[4] Man Ho Au, Joseph K Liu, Junbin Fang, Zoe L Jiang, Willy Susilo, and Jianying Zhou. A new payment system for enhancing location privacy of electric vehicles. *IEEE transactions on vehicular technology*, 63(1):3–18, 2013.

[5] Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. Rbft: Redundant byzantine fault tolerance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 297–306. IEEE, 2013.

[6] Kaibin Bao, Hristo Valev, Manuela Wagner, and Hartmut Schmeck. A threat analysis of the vehicle-to-grid charging protocol iso 15118. *Computer Science-Research and Development*, 33(1-2):3–12, 2018.

[7] Michael Barborak, Anton Dahbura, and Miroslaw Malek. The consensus problem in fault-tolerant computing. *ACM Comput. Surv.*, 25(2):171–220, June 1993.

[8] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In *Annual International Cryptology Conference*, pages 108–125. Springer, 2009.

[9] Josh Benaloh and Michael De Mare. One-way accumulators: A decentralized alternative to digital signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 274–285. Springer, 1993.

[10] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*, pages 142–157. Springer, 2016.

[11] Patrik Bichsel, Carl Binding, Jan Camenisch, Thomas Groß, Tom Heydt-Benjamin, Dieter Sommer, and Greg Zaverucha. Cryptographic protocols of the identity mixer library. *Tech. Rep. RZ 3730, Tech. Rep.*, 2009.

[12] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 103–112, New York, NY, USA, 1988. ACM.

[13] André B Bondi. Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203. ACM, 2000.

[14] Vitalik Buterin. On stake. URL: https://blog.ethereum.org/2014/07/05/stake/ (visited on 2019-07-01).

[15] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[16] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *International Workshop on Public Key Cryptography*, pages 481–500. Springer, 2009.

[17] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 21–30. ACM, 2002.

[18] Capgemini. The dso model: The emerging challenges with the dso role. URL: https://www.capgemini.com/2018/09/the-dso-model-the-emerging-challenges-with-the-dso-role/ (visited on 2019-07-23).

[19] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.*, 20(4):398–461, November 2002.

[20] Ann Cavoukian et al. Privacy by design: The 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada*, 5, 2009.

[21] Jie Chen, Yueyu Zhang, and Wencong Su. An anonymous authentication scheme for plug-in electric vehicles joining to charging/discharging station in vehicle-to-grid (v2g) networks. *China Communications*, 12(3):9–19, 2015.

[22] Edvard Csanyi. Primary distribution voltage levels. URL: https://electrical-engineering-portal.com/primary-distribution-voltage-levels (visited on 2019-07-23).

[23] Digiconomist. Bitcoin energy consumption index. URL: https://digiconomist.net/bitcoin-energy-consumption (visited on 2019-08-26).

[24] The Economist. The great chain of being sure about things. URL: https://www.economist.com/briefing/2015/10/31/the-great-chain-of-being-sure-about-things (visited on 2019-07-01).

[25] Marouane Fazouane, Henning Kopp, Rens W van der Heijden, Daniel Le Métayer, and Frank Kargl. Formal verification of privacy properties in electric vehicle charging. In *International Symposium on Engineering Secure Software and Systems*, pages 17–33. Springer, 2015.

[26] Bloomberg New Energy Finance. Electric vehicles to be 35% of global new car sales by 2040. URL: https://about.bnef.com/blog/electric-vehicles-to-be-35-of-global-new-car-sales-by-2040/ (visited on 2019-06-28).

[27] Tilman Frosch, Sven Schäge, Martin Goll, and Thorsten Holz. On locational privacy in the absence of anonymous payments. In *Data Protection on the Move*, pages 75–100. Springer, 2016.

[28] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[29] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[30] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[31] Golem. The golem project. URL: https://golem.network/crowdfunding/Golemwhitepaper.pdf (visited on 2019-07-01).

[32] Alyssa Hertig. Ethereum's big switch: The new roadmap to proof-of-stake. URL: https://www.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proof-of-stake (visited on 2019-07-01).

[33] Michael Hiltzik. Is this scathing report the death knell for bitcoin? URL: https://www.latimes.com/business/hiltzik/la-fi-hiltzik-bitcoin-bank-20180618-story.html (visited on 2019-07-01).

[34] Christina Höfer, Jonathan Petit, Robert Schmidt, and Frank Kargl. Popcorn: privacy-preserving charging for emobility. In *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*, pages 37–48. ACM, 2013.

[35] Hyperledger. Hyperledger indy transactions. URL: https://github.com/hyperledger/indy-node/blob/master/docs/source/transactions.md (visited on 2019-07-03).

[36] Hyperledger. Linux foundation's hyperledger project announces 30 founding members and code proposals to advance blockchain technology. URL: https://www.hyperledger.org/announcements/2016/02/09/linux-foundations-hyperledger-project-announces-30-founding-members-and-code-proposals-to-advance-blockchain-technology (visited on 2019-07-01).

[37] Hyperledger. Open source blockchain effort for the enterprise elects leadership positions and gains new investments. URL: https://www.hyperledger.org/announcements/2016/03/29/open-source-blockchain-effort-for-the-

enterprise-elects-leadership-positions-and-gains-new-investments (visited on 2019-07-01).

[38] Hyperledger Indy. Welcome to indy plenum's documentation! URL: https://hyperledger-indy.readthedocs.io/projects/plenum/en/latest/index.html (visited on 2019-07-01).

[39] Markus Jakobsson and Ari Juels. *Proofs of Work and Bread Pudding Protocols(Extended Abstract)*, pages 258–272. Springer US, Boston, MA, 1999.

[40] M. Jones, J. Bradley, and N. Sakimura. Json web token (jwt). RFC 7519, RFC Editor, May 2015. http://www.rfc-editor.org/rfc/rfc7519.txt.

[41] Sanket Kanjalkar, Joseph Kuo, Yunqi Li, and Andrew Miller. Short paper: I can't believe it's not stake! resource exhaustion attacks on pos. In *International Conference on Financial Cryptography and Data Security. Springer*, 2019.

[42] Fabian Knirsch, Andreas Unterweger, and Dominik Engel. Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions. *Computer Science-Research and Development*, 33(1-2):71–79, 2018.

[43] Yki Kortesniemi, Dmitrij Lagutin, Tommi Elo, and Nikos Fotiou. Improving the privacy of iot with decentralised identifiers (dids). *Journal of Computer Networks and Communications*, 2019, 2019.

[44] Dmitrij Lagutin, Yki Kortesniemi, Nikos Fotiou, and Vasilios A Siris. Enabling decentralised identifiers and verifiable credentials for constrained iot devices using oauth-based delegation.

[45] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[46] Lucie Langer, Florian Skopik, Georg Kienesberger, and Qin Li. Privacy issues of smart e-mobility. In *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pages 6682–6687. IEEE, 2013.

[47] Hong Liu, Huansheng Ning, Yan Zhang, Qingxu Xiong, and Laurence T Yang. Role-dependent privacy preservation for secure v2g networks in the smart grid. *IEEE Transactions on Information Forensics and Security*, 9(2):208–220, 2013.

[48] Michael Lodder and Daniel Hardman. Sovrin did method specification. Editor's draft, W3C, June 2019. https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html.

[49] Ralph C Merkle. Method of providing digital signatures, January 5 1982. US Patent 4,309,569.

[50] Ralph C Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

[51] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. URL: https://bitcoin.org/bitcoin.pdf (visited on 2019-07-01).

[52] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In *International Workshop on Public Key Cryptography*, pages 463–480. Springer, 2009.

[53] Hasen Nicanfar, Seyedali Hosseininezhad, Peyman TalebiFard, and Victor CM Leung. Robust privacy-preserving authentication scheme for communication between electric vehicle as power energy storage and power stations. In *2013 Proceedings IEEE INFOCOM*, pages 3429–3434. IEEE, 2013.

[54] Serguei Popov. The tangle. *cit. on*, page 131, 2016.

[55] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, Michaël Quisquater, Louis Guillou, Marie Annick Guillou, Gaïd Guillou, Anna Guillou, Gwenolé Guillou, and Soazig Guillou. How to explain zero-knowledge protocols to your children. In *Conference on the Theory and Application of Cryptology*, pages 628–631. Springer, 1989.

[56] Drummond Reed, Manu Sporny, and Markus Sabadello. Decentralized identifiers (dids) v0.13. Editor's draft, W3C, June 2019. https://w3c-ccg.github.io/did-spec/.

[57] Cristina Rottondi, Simone Fontana, and Giacomo Verticale. Enabling privacy in vehicle-to-grid interactions for battery recharging. *Energies*, 7(5):2780–2798, 2014.

[58] Markus Sabadello and Dmitri Zagidulin. Decentralized identifier resolution (did resolution) v0.1. Editor's draft, W3C, June 2019. https://w3c-ccg.github.io/did-resolution/.

[59] Jerome H Saltzer and Michael D Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[60] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[61] Neetesh Saxena, Santiago Grijalva, Victor Chukwuka, and Athanasios V Vasilakos. Network security and privacy challenges in smart vehicle-to-grid. *IEEE Wireless Communications*, 24(4):88–98, 2017.

[62] Vasilios A Siris, Dimitrios Dimopoulos, Nikos Fotiou, Spyros Voulgaris, and George C Polyzos. Trusted d2d-based iot resource access using smart contracts. In *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–9. IEEE, 2019.

[63] SOFIE. Initial platform validation. URL: https://media.voog.com/0000/0042/0957/files/SOFIE_D5.2-Initial_Platform_Validation.pdf (visited on 2019-07-30).

[64] Manu Sporny and Dave Longley. Linked data proofs 1.0. Draft report, W3C, November 2018. https://w3c-dvcg.github.io/ld-proofs/.

[65] Manu Sporny, Dave Longley, Grant Noble, and Daniel Burnett. Verifiable credentials data model 1.0. Candidate recommendation, W3C, March 2019. https://www.w3.org/TR/2019/CR-verifiable-claims-data-model-20190328/.

[66] Manu Sporny and Drummond Reed. Did method registry. Community draft, W3C, June 2019. https://w3c-ccg.github.io/did-method-registry/.

[67] Statista. Electric mobility in europe - statistics facts. URL: https://www.statista.com/topics/1010/electric-mobility/ (visited on 2019-06-28).

[68] Mark Stegelmann and Dogan Kesdogan. Design and evaluation of a privacy-preserving architecture for vehicle-to-grid interaction. In *European Public Key Infrastructure Workshop*, pages 75–90. Springer, 2011.

[69] Eric R Verheul. Practical backward unlinkable revocation in fido, german e-id, idemix and u-prove. *IACR Cryptology ePrint Archive*, 2016:217, 2016.

[70] Shermin Voshmgir. Blockchain oracles. URL: https://blockchainhub.net/blockchain-oracles/ (visited on 2019-07-19).

[71] Shermin Voshmgir. Blockchains distributed ledger technologies. URL: https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/ (visited on 2019-08-31).

[72] The Next Web. Why proof-of-work isn't suitable for small cryptocurrencies. URL: https://thenextweb.com/hardfork/2018/05/24/proof-work-51-percent-attacks/ (visited on 2019-07-01).

[73] Joel Weise. Public key infrastructure overview. *Sun BluePrints OnLine, August*, pages 1–27, 2001.

[74] Austin Wright and Henry Andrews. Json schema: A media type for describing json documents. Internet-Draft draft-handrews-json-schema-01, IETF Secretariat, March 2018. http://www.ietf.org/internet-drafts/draft-handrews-json-schema-01.txt.

[75] Zhenyu Yang, Shucheng Yu, Wenjing Lou, and Cong Liu. p{2}: Privacy-preserving communication and precise reward architecture for v2g networks in smart grid. *IEEE Transactions on Smart Grid*, 2(4):697–706, 2011.

[76] Vlad Zamfir. Introducing casper "the friendly ghost". URL: https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/ (visited on 2019-07-01).

A Credential definitions

```
{
        "@context": [
                "https://energy-ecosystem/credentials/cs-onboarding/v1"
        ],
        "id": "https://cso.energy-ecosystem.org/credentials/1",
        "type": ["VerifiableCredential", "CSOnboardingCredential"],
        "credentialSubject": {
                "id": "did:example:CS_1",
                "CSO": "did:example:CSO_Public"
        },
        "issuer": "did:example:CSO_Public",
        "issuanceDate": "2010-01-01T19:23:24Z",
        "proof": {
                "type": "RsaSignature2018",
                "creator": "did:example:CSO_Public",
                "created": "2010-01-01T19:23:25Z",
                "nonce": "2bbgh3dgjg2302d-d2b3gi423d42",
                "proofValue": "eyJ0eXAiOiJK...gFWFOEjXk",
        }
}
```

Listing 1: Example of a credential issued from CSO to a CS during UC-2. DIDs are used as identifiers in the example. An example of Linked Data proof is given in the proof section. The CSO uses its public identity to make the signature of the credential verifiable.

```
{
        "@context": [
                "https://energy-ecosystem/credentials/cs-registration/v1"
        ],
        "id": "https://dso.energy-ecosystem.org/credentials/1",
        "type": ["VerifiableCredential", "CSRegistrationCredential"],
        "credentialSubject": {
                "id": "did:example:CS_1",
                "district_id": "13"
        },
        "issuer": "did:example:DSO_Public",
        "issuanceDate": "2010-01-01T19:23:24Z",
        "proof": {
                "type": "RsaSignature2018",
                "creator": "did:example:DSO_Public",
                "created": "2010-01-01T19:23:25Z",
                "nonce": "2bbgh3dgjg2302d-d2b3gi423d42",
                "proofValue": "eyJ0eXAiOiJK...gFWFOEjXk",
        }
}
```

Listing 2: Example of a credential issued from DSO to a CS during UC-3. DIDs are used as identifiers in the example. An example of Linked Data proof is given in the proof section. The DSO uses its public identity to make the signature of the credential verifiable.

```
{
        "@context": [
                "https://energy-ecosystem/credentials/ev-charging/v1"
        ],
        "id": "https://er.energy-ecosystem.org/credentials/1",
        "type": ["VerifiableCredential", "EVChargingCredential"],
        "credentialSubject": {
                "id": "did:example:EV_1",
                "ER": "did:example:ER_Public"
        },
        "issuer": "did:example:ER_Public",
        "issuanceDate": "2010-01-01T19:23:24Z",
        "proof": {
                "type": "RsaSignature2018",
                "creator": "did:example:ER_Public",
                "created": "2010-01-01T19:23:25Z",
                "nonce": "2bbgh3dgjg2302d-d2b3gi423d42",
                "proofValue": "eyJ0eXAiOiJK...gFWFOEjXk",
        }
}
```

Listing 3: Example of a credential issued from ER to an EVU during UC-4. DIDs are used as identifiers in the example. An example of Linked Data proof is given in the proof section. The ER uses its public identity to make the signature of the credential verifiable.