energies

Article

Roaming Service for Electric Vehicle Charging Using Blockchain-Based Digital Identity

Joao C. Ferreira [1,2,*], Catarina Ferreira da Silva [1,3,*] and Jose P. Martins [1]

[1] ISTAR, Instituto Universitário de Lisboa (ISCTE-IUL), 1649-026 Lisbon, Portugal; jpmcs@iscte-iul.pt
[2] INOV INESC Inovação—Instituto de Novas Tecnologias, 1000-029 Lisbon, Portugal
[3] CISUC, 3030-290 Coimbra, Portugal
[*] Correspondence: jcafa@iscte-iul.pt (J.C.F.); catarina.ferreira.silva@iscte-iul.pt (C.F.d.S.);
Tel.: +351-210-464-277 (J.C.F.)

Abstract: We present a suitable approach to address the electric vehicle charging roaming problem (e-roaming). Blockchain technologies are applied to support the identity management process of users charging their vehicles and to record energy transactions securely. At the same time, off-chain cloud-based storage is used to record the transaction details. A user wallet settled on a mobile application stores user verified credentials; a backend application in the vehicle charging station validates the user credentials to authorize the energy transaction. The current model can be applied to similar contexts where the user may be required to keep several credentials from different providers to authenticate digital transactions.

Keywords: roaming; electric vehicle; EV charging process; blockchain; IoT; mobile app

check for updates

Citation: Ferreira, J.C.; Ferreira da Silva, C.; Martins, J.P. Roaming Service for Electric Vehicle Charging Using Blockchain-Based Digital Identity. *Energies* **2021**, *14*, 1686. https://doi.org/10.3390/en14061686

Academic Editor: Sangheon Pack

Received: 5 January 2021
Accepted: 15 March 2021
Published: 18 March 2021

1. Introduction

One of the significant challenges related to Electric Vehicle (EV) market penetration is the charging process. The charging process outside the home involves different entities and systems that create issues for EV owners. Although interoperability is available among systems in some regions and within a country, in particular, when a user needs to charge their EV outside of her/his country (or even regions) of origin, the interoperability between systems is nearly inexistent. Thus, previous planning and work will be needed to deal with different charging systems for an EV driver considering performing a European trip with an EV. Thus, the problem is not only to check where the charging stations are in their route and to identify the EV range, but also how it is possible to pay, to charge and to check if she/he has the suitable connector type. All over, the market is fragmented, and there is a bewildering number of providers.

In most cases, drivers need to access the charging process using a network RFID (Radio-Frequency Identification) card, a key-fob, or an app, some of which need to be pre-loaded with funds. Chargers that accept a contactless debit or credit card are few and far between. This has led many EV users to hold several subscriptions, one for each charging operator in each region or country [1]. Thankfully, some aggregators can provide an RFID card that works on several different networks, reducing the number of cards or apps that drivers need to obtain, but this usually only works in some countries. Consequently, travel abroad involves thorough planning and sometimes forces the owners to buy local charging cards and even establish local contracts. In the literature, we find that the term roaming refers to an EV driver's possibility of charging in different charging stations belonging to varying operators in different cities or countries. The increase in EVs and charging systems has developed concomitantly with extensive protocols and standards for charging system interoperability and e-roaming [2,3]. Charging system solutions for interoperability are only Available online: best at a national level or in a proprietary system such as Tesla, which has been recognized as a severe obstacle for the further uptake of electric mobility in Europe [3,4].

Concerns associated with energy transactions in security and privacy areas are raised due to several malicious threats [5,6], such as privacy leakage, falsification, node impersonation, and advertising a fraudulent charging service. As in [7–9], a security mechanism and monetary approach have been proposed and implemented to provide secure charging services for EVs. However, the security mechanism is vulnerable to a Sybil attack or a whitewashing attack. The monetary approach relies on trusted centers using a closed network with proprietary systems. However, trusted centers may leak users' privacy for profit and are vulnerable to attacks since they are based in a centralized approach. Blockchain (BC) appears to be an attractive solution since it provides a unique technology for secure energy transactions in a distributed network, without third party trusted agents, through the use of an immutable ledger, cryptocurrency, and the execution of smart contracts [10]. BC allows the creation of a trusted network, eliminates the intermediary participation's operation cost, and provides a cheaper way for energy transactions.

Given the decentralized nature of transport, with several stakeholders such as vehicles, drivers, charging stations, energy service providers, and others, EVs and e-mobility are natural applications for blockchains. Distributed ledger technology (DLT) enables one to avoid a centrally managed EV charging infrastructure, improving privacy and security concerns. It also reassures EV owners with more transparency in energy transactions, giving flexibility in choosing the energy supplier and simplifying the cross-border process. For operators, BC offers a market-based solution that can be used for optimized management, handling, validation, the storage of massive charging transactions, and the coordination of EV charging with renewable energy [11].

DLT has received much attention from both industry and academia due to its decentralized, persistent, anonymity for permissionless ledgers, and auditable properties. It allows one to record and handles digital asset transactions between parties over a decentralized, encrypted network in the energy sector while assuring transactions' integrity through secured encryption techniques [12]. The economic value of a BC-based approach in the energy market is enormous, and its impact is being studied [13]. Digital currencies such as Bitcoin [14], Ethereum, Hyperledger and BC-based Smart Contract (SC) approaches allow the performing of transactions in a new electricity market without centralized organizations or fixed prices. Complementarily, digital identity associated with BC [15] starts to play an essential role in authentication purposes in different systems, namely, checking and handling payment processes. BC enables self-sovereign identity management (SSIM), which is inherently unalterable and more secure than traditional identity systems, allowing users to [16]:

- Control their identities;
- Access and update information (though third-party verification may be required with some claims);
- Handle privacy issues;
- Transport the identity among different systems and organizations;
- Selectively disclosure information controlled by the holder.

Supported by blockchain-based SSIM, an EV driver can have one identity she/he can use across multiple systems and country boarders, establishing an interoperable roaming charging system. We propose and developed an EV prototype system enhanced with SSI, which provides EV owners with a unique approach for public charging, using a mobile device as an authentication process in a system that manages energy transactions among different players using BC and digital self-sovereign identity (SSI) to handle user authentication, securely record transactions, billing and security payments processing, and preserving owners identity from the Electric Vehicle Supply Equipment Operators (EVSEO) and simultaneously the location from the Electro Mobility Service Providers (EMSP). A key solution to the EV charging problem is the identity management of both drivers and charging equipment in a decentralized system allowing flexibility in the EV charging process. This approach creates the possibility of EV charging roaming and handles payments with the usage of digital cash. Our contribution is a roaming service-based solution for the EV charging process based on blockchain ledger technology coupled with decentralized identification and verifiable credentials supported by the permissioned Hyperledger Indy. We aim to contribute to creating flexibility in EV roaming charging process and stress the need to improve the interoperability of cross-border EV charging systems.

This paper is organized as follows. Section 2 highlights the related work, Section 3 presents our running scenario, while Section 4 illustrates the details of our BC-based identity service for EV roaming charging. Section 5 instantiates the system usage showing a standard user journey and the information exchanged between parties. Finally, Section 6 concludes the paper and provides future work.

2. Related Work

2.1. EV Roaming Protocols

There are many attempts to develop roaming protocols based on the concept of a central clearing house with varying success. However, none of these protocols has achieved any recognition from the national or the international standardization bodies yet. To avoid this, a BC solution overcomes these problems without the need for standardization of operations procedures. Several works introduce this approach mainly at a conceptual level, such as Mustafa et al. [17], who work on a specification of a set of security and privacy requirements for EV charging transactions. Gan et al. [18] proposed a decentralized protocol for EV charging to improve charging efficiency. Aitzhan et al. [19] presented a token-based decentralized energy trading system to enable peers to perform transaction anonymously and securely, and Mattila et al. [20] presented a BC use case for machine-to-machine (M2M) energy transactions in a housing society environment.

In 2017, Kang et al. [21] proposed a peer-to-peer (P2P) energy trading model with a consortium BC approach to address the privacy-preserving and transaction security issues in EV charging, and also Li et al. [22] used the consortium blockchain method to secure distributed energy trading market and formulated a novel energy BC system using the Internet of Things (IoT). Additionally, in 2018 a privacy-preserving BC incentive was proposed to motivate vehicle users to share traffic information by credit-coin [23], and Huang et al. [24] presented a decentralized security model to improve the security of trading between EVs and charging piles in a P2P network. Liu and his peers [25] propose an adaptive BC-based EV participation scheme in a smart grid platform. Erdin et al. [26] used an off-chain payment for EV charging to avoid high transaction fees and address the privacy exposure problem based on creating a payment network with permissions and signatures. In 2019, Martins et al. [27] proposed an EV charging process in shared spaces, such as condominiums, based on the IoT and decentralized BC. In 2020, Daghmehchi et al. [28] provided a hybrid BC with privacy-preserving and trustful energy transactions features for IoT platforms.

Despite all of these approaches, most of them at a conceptual level, none of them provide an integrated view of EV charging flexibility in a solution without associated charging cards. In this scenario, a BC-based identity process management plays an important role and is our main contribution, along with implementing digital BC-based decentralized identification and verifiable credentials which avoid the need for charging cards.

2.2. Identity Management Using Blockchain

Identity management (IdM) plays an important role in a universal EV charging process without centralized control [15]. IdM can be used by EV owners and charging stations to create a flexible charging system because of buyer and seller certification. Centralized models of IdM currently face challenges due to the increasing regularity of data breaches that lead to reputation damage, identity fraud, and above all, a loss of privacy for all concerned. These recurring events highlight a lack of control and ownership that end-users experience with their digital identities [29].

Today, most IdM schemes are centralized where a single entity such as an organization owns and controls the system [30]. Recently, several decentralized identity schemes have emerged that extend beyond naming and aim to provide a complete suite of IdM functions. However, until now, there has been no evaluation of these proposals. We are interested in studying whether DLT-based IdMs can go beyond previous approaches or simply create new “identity one-offs”.

A digital identity is a means to prove that someone or something is who/what they claim to be and to differentiate identities. In contrast, SSI is stated to be based on decentralized identifiers (DIDs) which should be fully under the control of the DID subject, independent from any centralized provider or certificate authority [31,32]. SSI affords more user control over her/his data and a more user-centric experience alongside distributed technologies. SSI allows the users to manage their own identity credentials and implements the W3C Decentralized Identifier (DID) [31], which is a structure containing the user identifier, cryptographic public keys, and other metadata necessary to transact with that identifier, as well as the Verifiable Credentials (VC) model [33]. VC are digital credentials associated with an identifier and cryptographic proofs such as digital signatures, enabling one to check if a credential is genuine and not tampered with and presented by the person/entity claiming to be the user.

BC identity management systems are emerging and are discussed in [16]. Sovrin is one of these systems. It is an open-source SSI network built on permissioned DLT that manages identity records [34]. The members of the Sovrin Foundation contribute to the DLT Hyperledger Indy and Hyperledger Aries. Hyperledger Indy provides the infrastructure to serve up BC-based digital identities, while Hyperledger Aries provides an identity agent enabling one to create, transmit, and store verifiable digital credentials. The identities in our EV e-roaming proposal are based on Hyperledger Indy and Aries. The next section presents the running scenario for our proposal.

3. Running Scenario

Our approach aims to mediate the relation between the following four entities: the EV Owner (EVO) who owns an EV; the Electro Mobility Service Provider (EMSP), which establishes the energy supply agreement with the EVO and mediates the relation between the EVO and the Electric Vehicle Supply Equipment Operators (EVSEO); and the Charging Stations (CS), owned by the EVSEO and used by the EVO to charge the EV.

We dematerialize the identification process based on physical cards, replacing it with the use of Decentralized Identifiers (DID) [31], Verifiable Credentials (VC) [33], and Verifiable Presentations (VP) [31] to identify the participating entities mutually. We use a permissioned Distributed Ledger (DL) to support the identification processes and securely record tamper-proof energy transaction information used by the EMSP to invoice the EVO.

An EVO has a DID, which is provided by a governing agency, which she/he uses to establish a contractual agreement with an EMSP in the existing electric energy service market. Using a mobile application, the EVO provides the information required to create a legally binding contract (e.g., DID and payment information) and signs that contract with a qualified digital signature [35]. Alternatively, she/he can receive and send a signed, legally binding contract by other channels (e.g., mail, e-mail). The EMSP receives the credential request and verifies the correctness of the information provided. As a counterpart of this contract agreement and to allow the EVO to identify her/his purchase when accessing the CS, the EMSP issues verifiable credentials (VC) with several claims associated with the contractual relation (e.g., customer identity on EMSP, charging limit/day, roaming validity zones) to the EVO, either directly proposed by the EMSP for registration in a node of the permissioned ledger or by an third party agency service provider such as a VC issuer. This VC issuer is a member of the permissioned ledger and can propose VCs on behalf of EMSP following a previous agreement between the latter and the VC issuer. This VC can be verified by the EVSEO, which own and maintain the CSs, by querying the ledger each time the EVO wants to charge her/his vehicle. For security and privacy reasons, only essential metadata regarding the EVO VC, such as proofs, timestamps, service endpoints, and public keys, are available for reading on the permissioned ledger. We rely on Zero-Knowledge Proof [36], which the Hyperledger Indy supports, to avoid undesirable disclosure of EVO information. Zero-Knowledge Proof (ZKP) permits the EVO to authenticate a VC's possession by providing an anonymous credential and VP without presenting the credential itself. This also enables stakeholders to verify the validity period of the specific information.

In a standard user journey, the EVO parks and connects the EV vehicle to the CS. With its mobile app, the EVO reads the QR code presented at the CS. However, other channels such as internet listing or beacons can also be used to identify the CS. Then, the EVO sends a connection request to the CS, which replies with its credentials (proving it is a valid CS) and requests to the EVO proof of owing a VC with specific properties (without requiring all the credential information) issued by an EMSP that allows the EVO to use that CS. The EVO presents her/his ZKP verifiable claim, stored in her/his mobile, to identify a valid contract. The CS queries the permissioned ledger to verify it is a legitimate user. Following the confirmation, the CS starts releasing the energy. The CS measures the power delivered, and at the end of the charging session the CS computes the total amount of energy loaded and presents a verifiable claim to the EVSEO, enabling it to record the energy transaction associated with the EVO device's proof on the transaction ledger, which the EMSP can read for invoicing the charging costs to the EVO. Figure 1 presents the entities participating in the system as well as the relevant service providers. The next section provides the technical details of our approach.

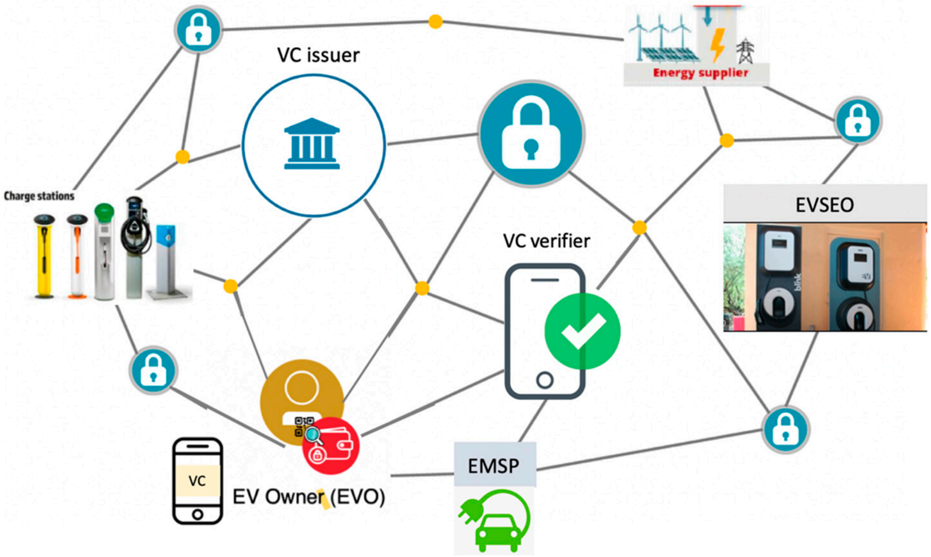


Figure 1. Electric Vehicle (EV) roaming system context.

4. Proposed Approach

In the current section, we present each platform component's technical detail, describing the high-level implementation, the role of the system, and the information exchanged with the other entities composing the entire platform, focusing on the EV-roaming process. The use cases and implementation details presented define the minimal requirements or behaviours an application participating in the system and fulfilling a specified role needs to implement. Due to the DL technologies' distributed nature, a system participant can develop its local implementation if the protocols and information flows are respected (e.g., exchanged messages, credentials schemas).

4.1. EVO Mobile Application

The EVO interaction with the system relies primarily on an internet-connected mobile device. Figure 2 presents a UML use case diagram [37,38] for the use cases supported by the mobile application (app) installed on the EVO device (i.e., smartphone, tablet).

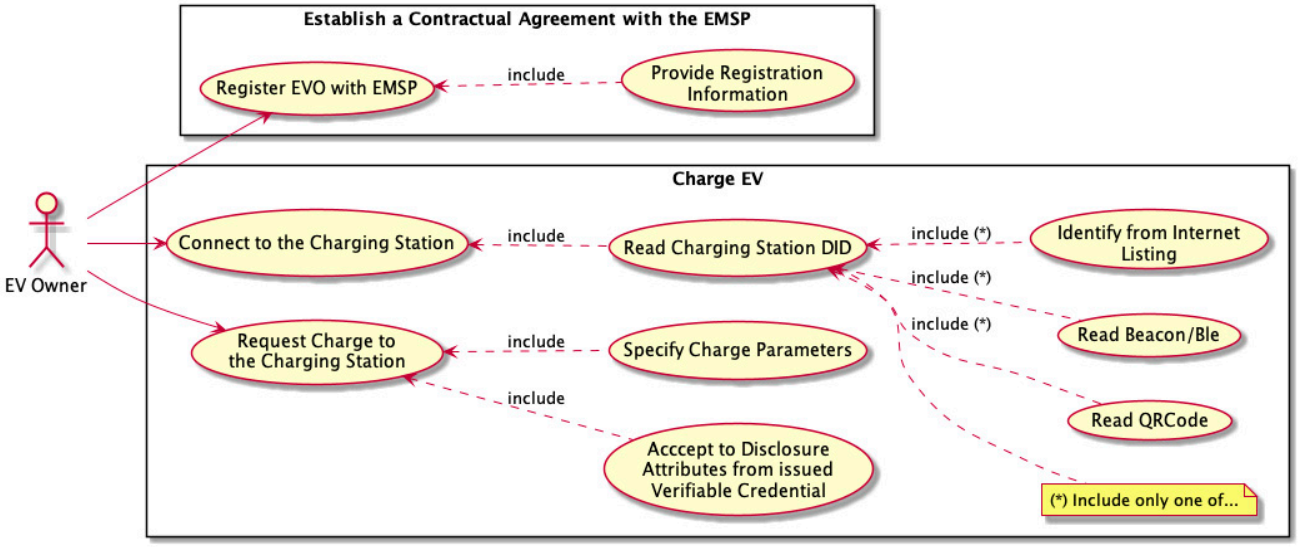

Establish a Contractual Agreement with the EMSP
Register EVO with EMSP
include
Provide Registration Information
Charge EV
EV Owner
Connect to the Charging Station
include
Read Charging Station DID
include (*)
Identify from Internet Listing
include (*)
Read Beacon/Ble
include (*)
Read QRCode
Request Charge to the Charging Station
include
Specify Charge Parameters
include
Acccept to Disclosure Attributes from issued Verifiable Credential
(*) Include only one of...

Figure 2. EV owner (EVO) use cases.

The mobile application (app) is the system's user interface element, allowing the EV owner to store and manage her/his own VC on a local/device-based wallet and simultaneously present those credentials to authenticate the user on the CS and present proof she/he can charge their vehicle. Aiming to fulfill this goal, the mobile app implements several use cases that can be grouped into two process groups: the contractual enrolment and vehicle charging. The upper part of Figure 2 supports the establishment of the contractual agreement between EVO and the EMSP. The EVO fills and submits a form with the required information to establish a binding contractual agreement with the EMSP. Depending on the contractual requirements for every particular EMSP, different information can be required and sent to different endpoints. Upon receiving and validating the information, the EMSP offers a VC, which is stored on the EVO device.

A UML sequence diagram [38] of the EVO enrolment journey is presented in Figure 3. The EVO initiates the mobile app's enrollment, allowing the user to provide the Uniform Resource Locator (URL) of the EMSP. As the contractual requirements may vary depending on the EMSP company, the mobile app queries the EMSP platform to obtain the list of information that is required to establish a contractual agreement with that specific energy supplier, such as DID and payment information. The user provides the required information on the mobile app and sends it to the EMSP for further validation. After validation of the entered information, assuming that the required conditions to establish a contractual agreement are fulfilled, the EMSP platform issues and offers a VC to the EVO that will be used to initiate a charging process on the CS.

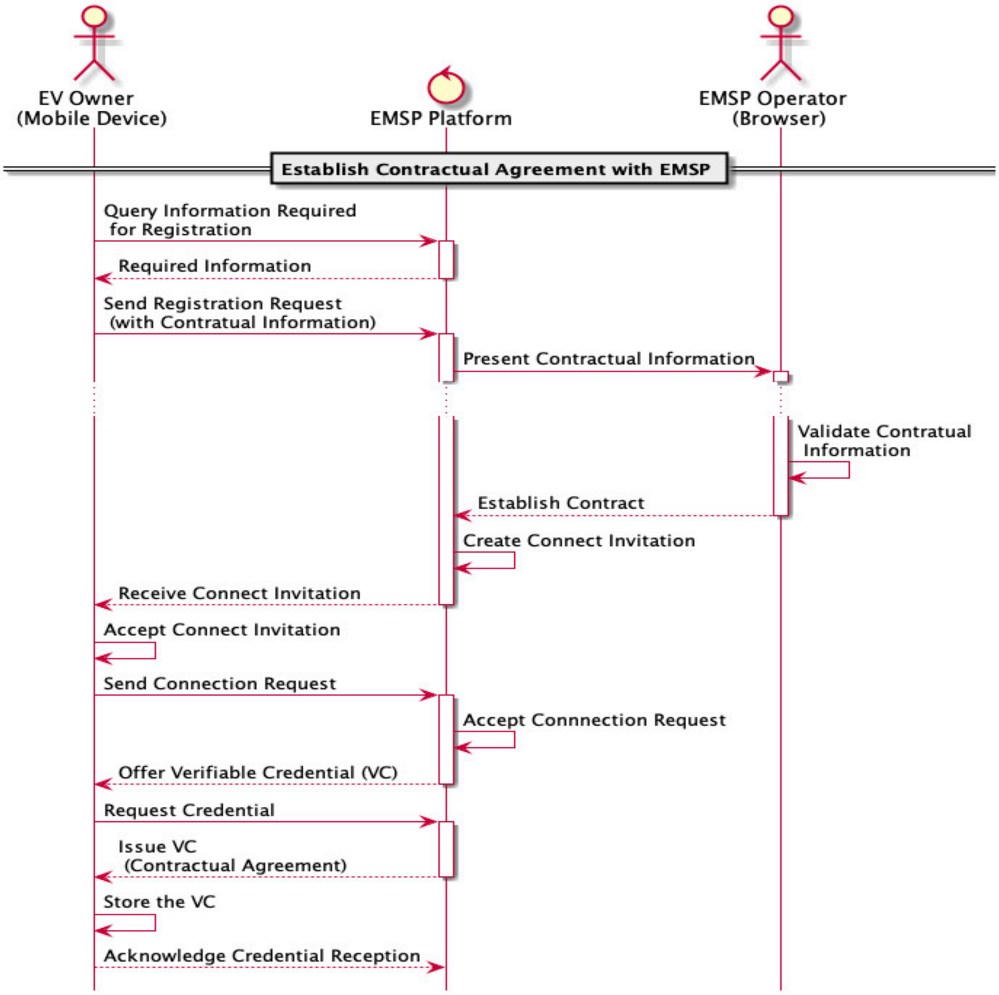

EV Owner
(Mobile Device)
EMSP Platform
EMSP Operator
(Browser)
Establish Contractual Agreement with EMSP
Query Information Required
for Registration
Required Information
Send Registration Request
(with Contratual Information)
Present Contractual Information
Validate Contratual
Information
Establish Contract
Create Connect Invitation
Receive Connect Invitation
Accept Connect Invitation
Send Connection Request
Accept Connnection Request
Offer Verifiable Credential (VC)
Request Credential
Issue VC
(Contractual Agreement)
Store the VC
Acknowledge Credential Reception

Figure 3. EV owner registration sequence diagram.

The second group of use cases is related to vehicle charging and the interaction between the EVO and CS. Figure 4 presents the information exchanged to initialize the charging process between these participants in the process. The EVO initiates the charging process by establishing a connection with the CS. The EVO, employing her/his mobile app, reads the connect invitation QR code [39] displayed on the CS or accesses that information from an internet listing (additional approaches can be explored such as the use of BLE and Beacons) and establishes a peer-to-peer connection with the CS. After establishing the CS connection, the EVO sends a message to the CS requiring a charge session to be initiated (and providing the CS with the charging parameters, i.e., time, max current, etc.). In response to the charging request, the CS requires the EVO to present proof that she/he owns a valid VC. The user selects the claim from the list of valid VCs that can be used to answer the presentation requests and presents the selected VC to the CS. The CS verifies the VC's validity by querying the permissioned ledger, thereupon enabling the charging process.

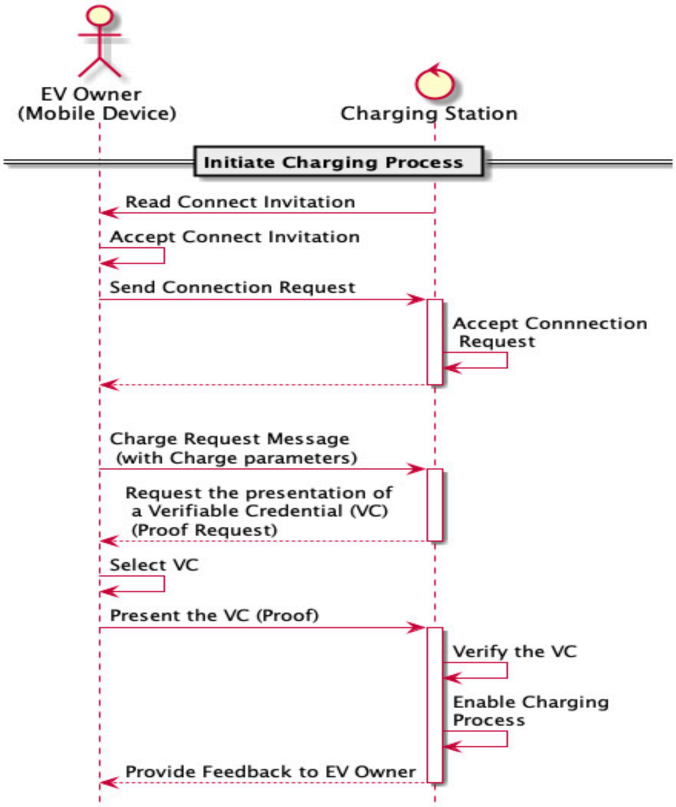

EV Owner
(Mobile Device)
Charging Station
Initiate Charging Process
Read Connect Invitation
Accept Connect Invitation
Send Connection Request
Accept Connnection Request
Charge Request Message (with Charge parameters)
Request the presentation of a Verifiable Credential (VC) (Proof Request)
Select VC
Present the VC (Proof)
Verify the VC
Enable Charging Process
Provide Feedback to EV Owner

Figure 4. Charging process sequence diagram.

4.2. The EVO Mobile App Implementation

Extending the approach used in [27], the mobile app was developed on the Xamarin Forms using the Microsoft® Visual Studio development platform (Microsoft, Seattle, WA, USA). To allow the developed application to connect to the identity management BC network used for this implementation, the Hyperledger Indy/Aries Framework for NET [40] is used, and the QR code and decoding were implemented using the ZXIng [41] Net port [42]. Our system comprises four software modules to support the required functionalities, as presented in Figure 5. The configuration and logging support modules that implement transversal features to the application, providing configuration management and logging services to the remaining software modules. Apart from these support modules, the application is composed of two major modules: the registration management and the charge management.

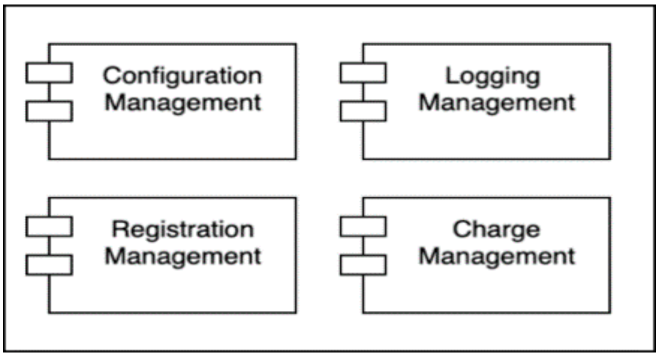
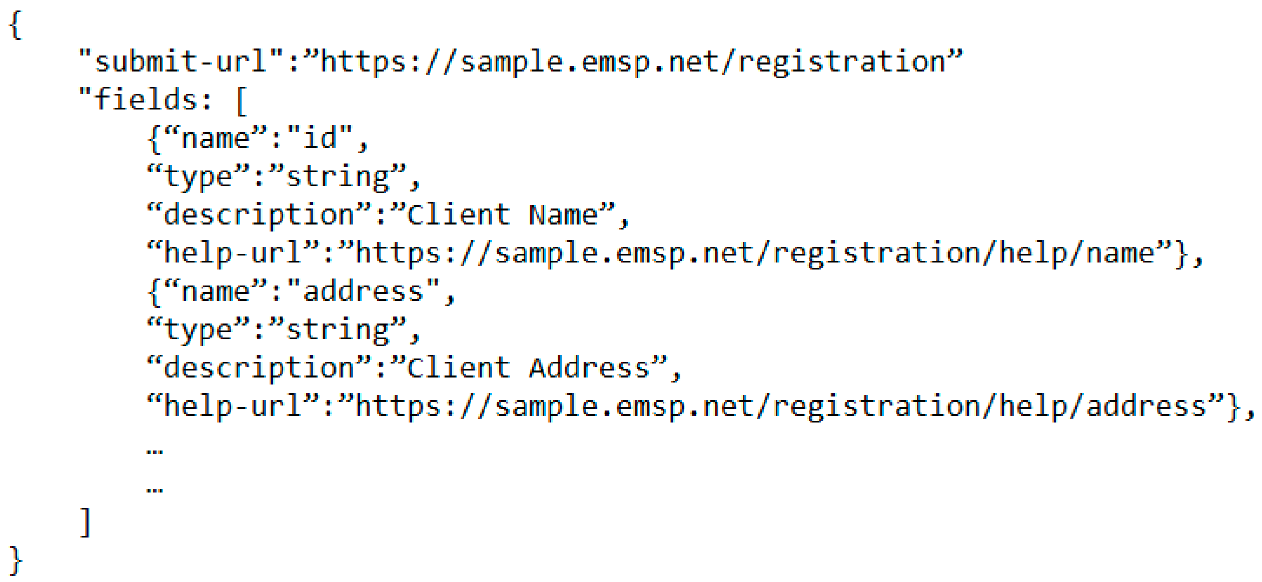


Figure 5. The EVO mobile app modules.

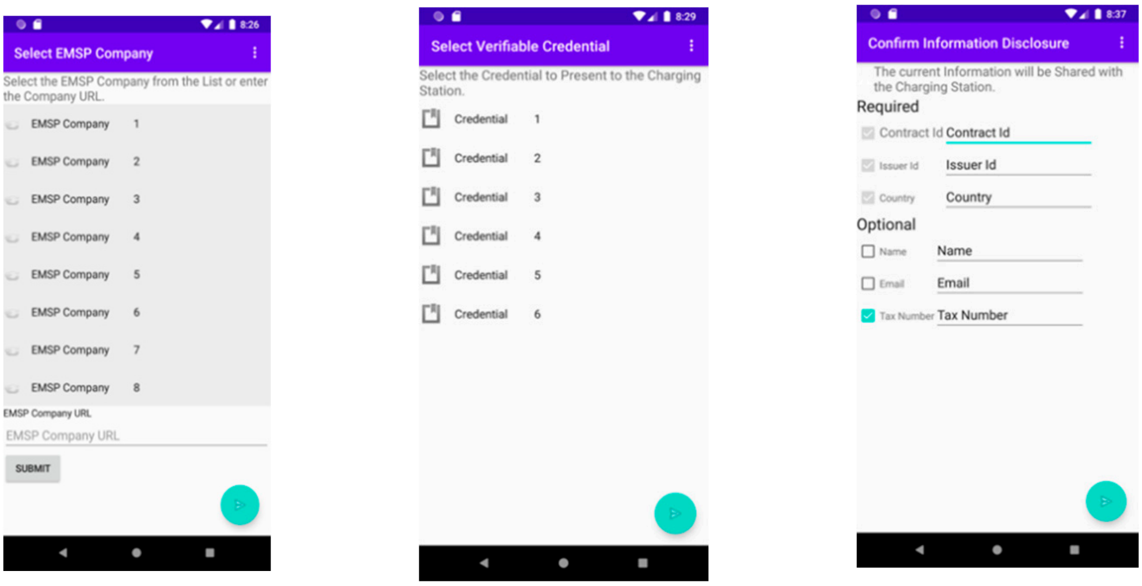
The Registration Management module supports the use cases related to the registration process implementing the flows presented in Figure 3, the interaction with the EMSP, the dynamic rendering, and the registration form submission process. Figure 6 illustrates the registration form metadata provided by the EMSP platform.



```
{
    "submit-url":”https://sample.emsp.net/registration”
    "fields: [
        {“name”:"id",
        “type”:”string”,
        “description”:”Client Name”,
        “help-url”:”https://sample.emsp.net/registration/help/name”},
        {“name”:"address",
        “type”:”string”,
        “description”:”Client Address”,
        “help-url”:”https://sample.emsp.net/registration/help/address”},
        …
        …
    ]
}
```

Figure 6. EVO registration form metadata.

The Charge Management module implements the use cases related to the charging process, illustrated by Figure 4, using the camera of the device to read the CS QR code, establishing the connection with the CS, requiring the charging operation of the CS, and providing the VC claim required by the CS to initiate the operation. Figure 7 presents some screenshots of the EVO mobile app, targeting the use cases shown in Figure 2.



Select the EMSP Company from the List or enter the Company URL.

Select the Credential to Present to the Charging Station.

The current Information will be Shared with the Charging Station.

Required

☑ Contract Id: Contract Id

☑ Issuer Id: Issuer Id

☑ Country: Country

Optional

☐ Name: Name

☐ Email: Email

☑ Tax Number: Tax Number

(**a**) (**b**) (**c**)

Figure 7. The EVO mobile app screenshots. (**a**) Selection of provider company for registration. (**b**) Selection of verifiable credential. (**c**) Accepting disclosure information.

4.3. EMSP Application

The EMSP are the organizations that contractually establish the energy supply contracts with the EVO. The EMSP provides the EVO with an authentication token, such as VCs, and is responsible for the ledger registration of the financial transactions, collecting the payment for the energy charged by the EVO and distributing the collected payment across the organizations that contributed to providing the service (e.g., energy suppliers, charging station owners). From a system perspective, minimal requirement is requested from the EMSP to be authorized to write in the permissioned authentication ledger to be able to issue the VC to the EVO. From a functional perspective, the EMSP prototype application implements the minimal requirements to allow an EMSP operator to receive the information required to establish a legally binding contract with the EVO and to validate the information provided by the EVO, and if considered valid, to establish a formal contract with the EVO and in the light of that contract, issue a VC to the EVO. Figure 8 provides minimal EV-roaming-related use cases implemented by an EMSP-owned application to participate in the system.

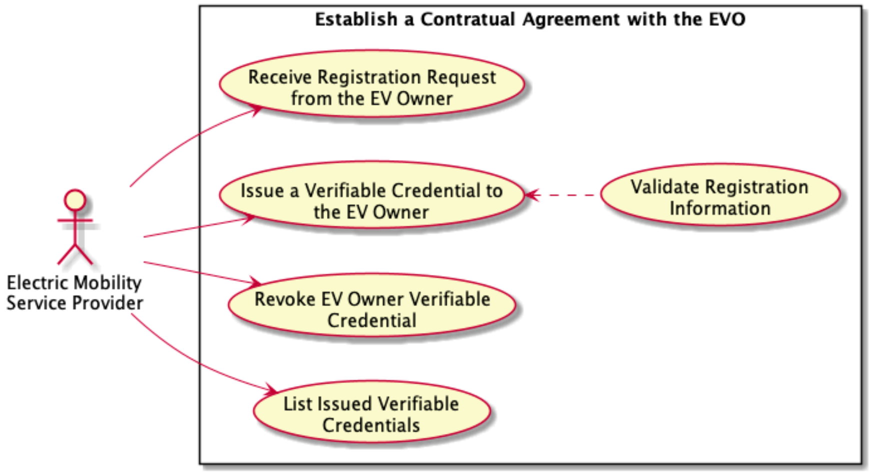
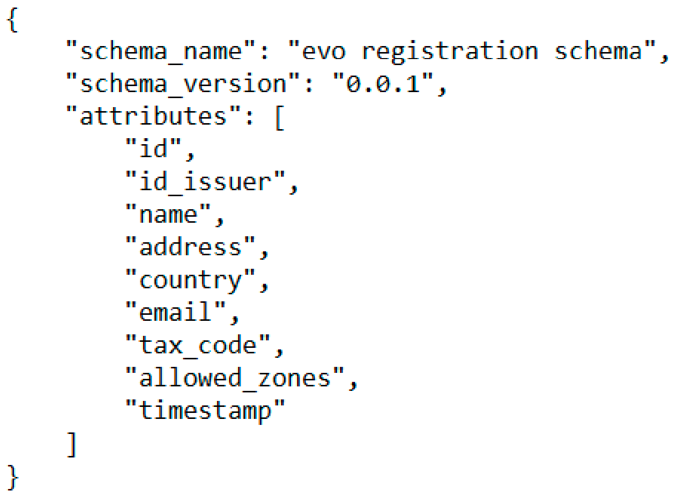

Establish a Contratual Agreement with the EVO
Receive Registration Request from the EV Owner
Issue a Verifiable Credential to the EV Owner
Validate Registration Information
Revoke EV Owner Verifiable Credential
List Issued Verifiable Credentials
Electric Mobility Service Provider

Figure 8. Electric Mobility Service Provider user cases.

The EMSP use cases are associated with the process of granting the EVO a VC. The EMSP receives the registration request, analyses, and validates the information. Figure 9 displays the schema definition for an issued credential.



```
{
    "schema_name": "evo registration schema",
    "schema_version": "0.0.1",
    "attributes": [
        "id",
        "id_issuer",
        "name",
        "address",
        "country",
        "email",
        "tax_code",
        "allowed_zones",
        "timestamp"
    ]
}
```

Figure 9. EVO verifiable JSON (JavaScript Object Notation) credential schema definition.

The attribute allowed zones (Figure 9) represents the list of the geographical zones where the EVO is allowed to charge the vehicle, which is used to filter the VCs.

The remaining use cases revoke the existing credentials and list the issued credentials, which are utility operations required to maintain the platform. Although the EMSP application is considered self-contained in the scope of our proof of concept, in a business context, the implementation of use cases can be achieved by integrating the internal systems and the service layers of EMSP platform.

4.4. EMSP Application Implementation

The software infrastructure of the EMSP, presented in Figure 10, was implemented with containers to simplify the deployment environment. The following components are packaged inside each distributed node docker container:

- Hyperledger Aries Agent Python (ACA-Py) mediates the relation between the authentication process in the distributed ledger and the application business logic components, exposing its services using a Representational State Transfer (REST) architectural pattern and webhooks;
- Application Logic implements all the business logic implemented using the NodeJS and follows the architectural approach used by the ACA-Py, exposing the provided services using a REST architectural style;
- The application storage is implemented using MariaDB, an open-source relational database;
- Management Console is implemented with a single web page architectural style; the applicational frontend (a layer that exposes the platform services to the end-user) is implemented with Angular;
- The webserver NGINX is used as a proxy to route all the requests from end-users to internal platform components and simultaneously to serve the static components that compose the Management Console. Additionally, it adds a TLS layer to secure the communications between the end-user browsers and the platform.

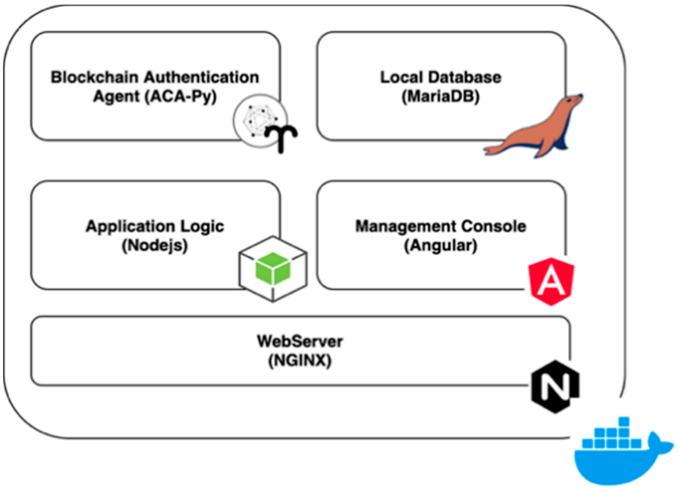

Blockchain Authentication Agent (ACA-Py)
Local Database (MariaDB)
Application Logic (Nodejs)
Management Console (Angular)
WebServer (NGINX)

Figure 10. Electro Mobility Service Provider (EMSP) platform software infrastructure.

Figure 11 presents the current interaction flow between the EMSP application components for a user-originated interaction. On the first user interaction, the user fetches from the server the code for the Management Console that runs on the user browser, after this operation, any user request will be translated in a simple service request that is received by the NGINX web server and delivered to the NodeJS application responsible by servicing the user request. If required, the NodeJS application accesses the database to read or write information, similar to the database access. If the user operation requires access to the BC ledger authentication network, the NodeJS application invokes the REST services provided by the ACA-Py.

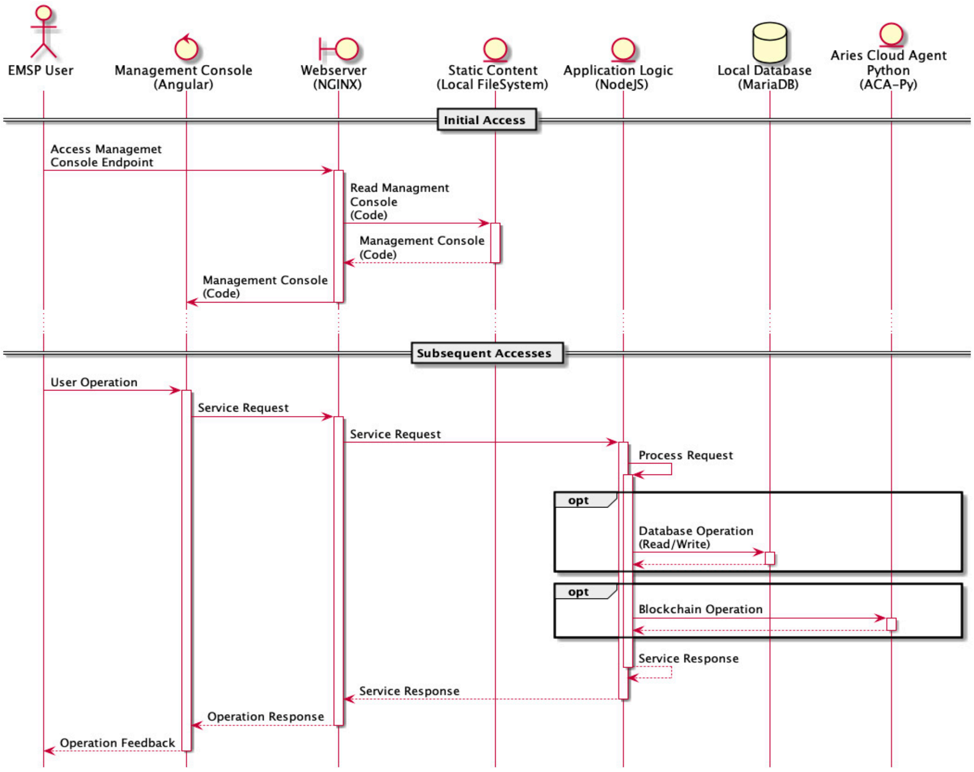

EMSP User
Management Console (Angular)
Webserver (NGINX)
Static Content (Local FileSystem)
Application Logic (NodeJS)
Local Database (MariaDB)
Aries Cloud Agent Python (ACA-Py)
Initial Access
Access Managemet Console Endpoint
Read Managment Console (Code)
Management Console (Code)
Management Console (Code)
Subsequent Accesses
User Operation
Service Request
Service Request
Process Request
opt
Database Operation (Read/Write)
opt
Blockchain Operation
Service Response
Service Response
Operation Response
Operation Feedback

Figure 11. EMSP user originated sequence diagram.

Figure 12 presents a network originated interaction regarding the EMSP. The reception on an event initiates the operation originated on the ACA-Py and delivered to the Application Logic component through a registered URL (callback/webhook). Upon the reception of an event, the Application Logic component processes the message, accesses the local database, or even invokes the ACA-Py services if required.

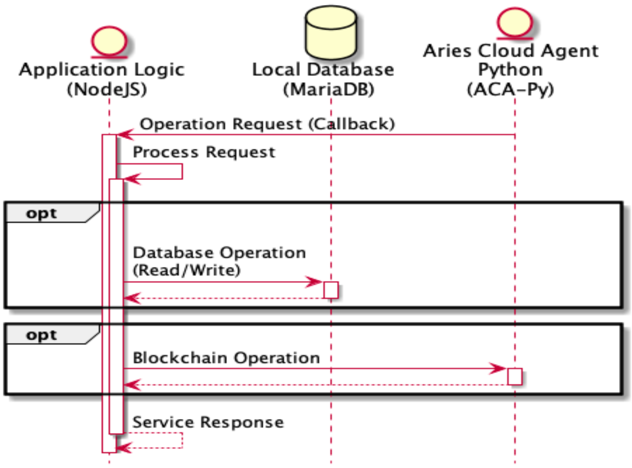

Application Logic (NodeJS)
Local Database (MariaDB)
Aries Cloud Agent Python (ACA-Py)
Operation Request (Callback)
Process Request
opt
Database Operation (Read/Write)
opt
Blockchain Operation
Service Response

Figure 12. EMSP BC network originated sequence diagram.

Figure 13 presents the list of modules composing the EMSP prototype application. The Application Logic is composed of two resource adapters responsible for establishing the interaction with the external components (the database and the ACA-Py engine), two support modules to centralize all the logging and configuration management components, and a registration management module that implements the business logic to support all the registration-related operations.

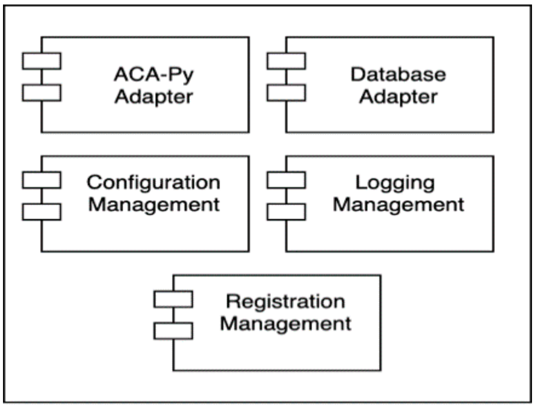
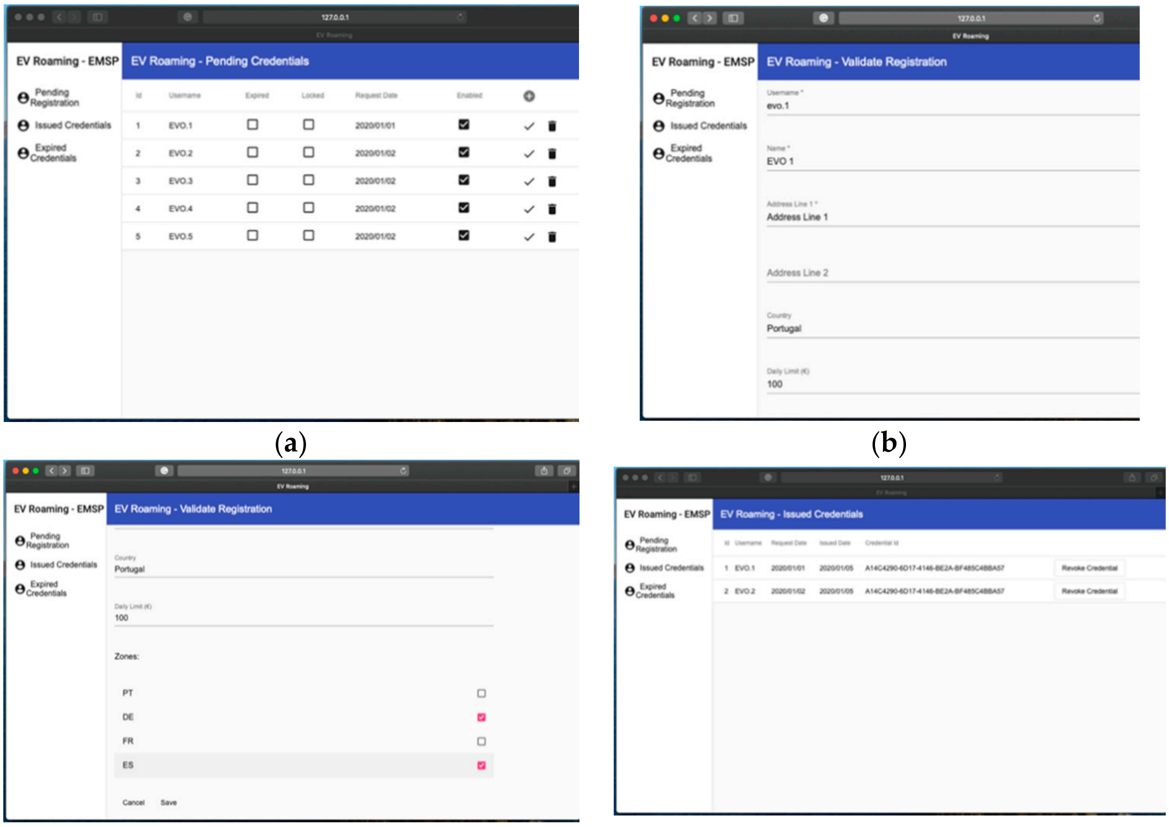


Figure 13. EMSP software modules.

Figure 14 presents some screenshots of the user interface (UI) EMSP application for the use cases in Figure 8.



EV Roaming - Pending Credentials

Id	Username	Expired	Locked	Request Date	Enabled	
1	EVO.1	□	□	2020/01/01	☑	✓
2	EVO.2	□	□	2020/01/02	☑	✓
3	EVO.3	□	□	2020/01/02	☑	✓
4	EVO.4	□	□	2020/01/02	☑	✓
5	EVO.5	□	□	2020/01/02	☑	✓

(**a**)

EV Roaming - Validate Registration

Username *
evo.1

Name *
EVO 1

Address Line 1 *
Address Line 1

Address Line 2

Country
Portugal

Daily Limit (€)
100

(**b**)

EV Roaming - Validate Registration

Country
Portugal

Daily Limit (€)
100

Zones:

PT □
DE ☑
FR □
ES ☑

Cancel Save

(**c**)

EV Roaming - Issued Credentials

Id	Username	Request Date	Issued Date	Credential Id	
1	EVO.1	2020/01/01	2020/01/05	A14C4290-6D17-4146-BE2A-BF485C4BBA57	Revoke Credential
2	EVO.2	2020/01/02	2020/01/05	A14C4290-6D17-4146-BE2A-BF485C4BBA57	Revoke Credential

(**d**)

Figure 14. EMSP application screenshots. (**a**) List of pending registration requests; (**b**) Registration request—user details; (**c**) Registration request—contract details; (**d**) List of issued verifiable credentials.

4.5. CS Embedded Application

The Charging Station (CS) implements a typical charging station device, although the implementation's prototype nature and the lack of loss of generality are achieved with a composition of commercial off-the-shelf available components. In essence, the CS device and its embedded software implement a minimal and specific set of functionalities; namely, the CS is required to authenticate a roaming user and allow the authenticated user to charge its EV while measuring the amount of energy charged. Figure 15 represents graphically the use cases implemented by the CS and briefly described above.

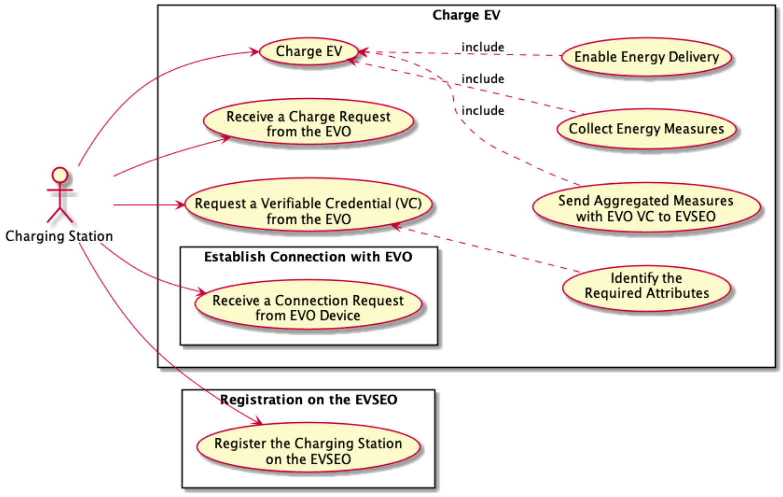

Charge EV
Charge EV
include
include
include
Enable Energy Delivery
Receive a Charge Request from the EVO
Collect Energy Measures
Request a Verifiable Credential (VC) from the EVO
Send Aggregated Measures with EVO VC to EVSEO
Charging Station
Establish Connection with EVO
Receive a Connection Request from EVO Device
Identify the Required Attributes
Registration on the EVSEO
Register the Charging Station on the EVSEO

Figure 15. Use cases implemented by the Charging Stations (CS).

After receiving the charging request message and a VC that confirms that EVO is authorized to use the CS and charge the EV, as previously presented in Figure 4, the CS starts the charging process. To charge the EV, the CS enables the energy delivery and continuously monitors the total time, the total amount of energy delivered, and instant energy flow until the values requested by the EVO are reached or the EV is fully charged; when that event is detected, the EV stops receiving energy. Figure 16 presents the activity diagram for the charging process.

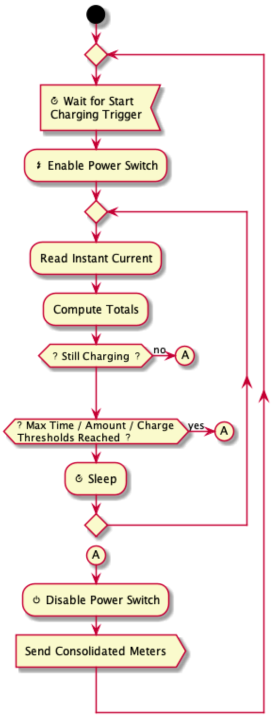


Figure 16. CS charging process activity diagram.

4.6. CS Implementation

The implementation of the CS unit is addressed in the current section. Figure 17 presents the hardware selected to manage and monitor the CS process. The CS controller is mainly composed of a Raspberry PI Zero W (a), providing the computing power required to support the BC-based identity management software and the controller-specific software. Attached to the computing unit we have a relay (b) to enable/disable the delivery to the EV and an analog-to-digital converter (ADC) (c) to feed the computing unit in near-real-time with the measure of the amount of current that is being delivered to the EV. The current delivery to the EV is measured with a non-intrusive device (d), which behaves as a current transformer, exposing the device terminals to a fraction of the current being passing through the circuit under measurement. The ADC used is designed to convert a tension value to a digital value; to convert the current measured into tension, a small electronic circuit (e) is added between the current sensor and the ADC.

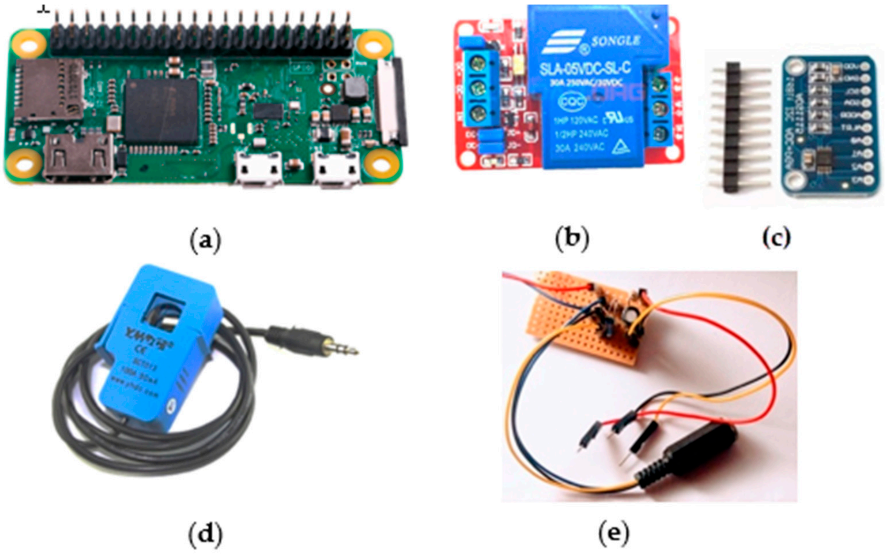


Figure 17. Main hardware components for the charging station (CS). (**a**) Raspberry PI Zero W; (**b**) ADS111 analog-to-digital converter 5; (**c**) SLA-05VDC-SL-C relay; (**d**) Current/tension converter; (**e**) SCT013 current transformer.

As presented in Figure 18, the CS embedded software relies on a reduced set of components installed in a Raspbian operating system from a software infrastructure perspective. The scope for the current implementation does not have substantial user interface requirements. All the application logic is implemented with a small Python application, running as a daemon on the operating system's top. The Hyperledger Aries Agent establishes the interface with the BC ledger, and a small SQLite database is used to store locally the transient measurement data as well as some local specific configurations.

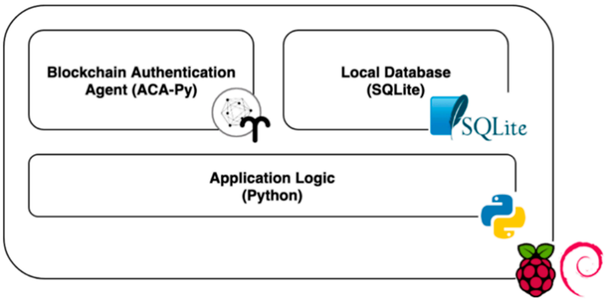

Blockchain Authentication Agent (ACA-Py)
Local Database (SQLite)
SQLite
Application Logic (Python)

Figure 18. The Charging Station software infrastructure.

The interaction between the software infrastructure components follows a similar pattern to the pattern presented in Figure 11, in this case, the Node Application and the MariaDB database, replaced by a local Python application and a local SQLite. Following the modular design presented in the previous software components, Figure 19 shows several software modules that compose the CS unit embedded software. The database adapter and the ACA-Py adapter aim to abstract the main application of the inherent complexities introduced by the ACA-Py (Hyperledger Aries) and the database components (SQLite). The Configuration and Logging Management implement support services to the application, and the Registration Management and the Charge Management are the modules that implement the application logic required by the CS. The Registration Management provides the required operations to register the CS with the EVSEO and implement all the required VC exchanging. The Charge Management implements all the logic required to interact with the EVO and manage the EV charging process.

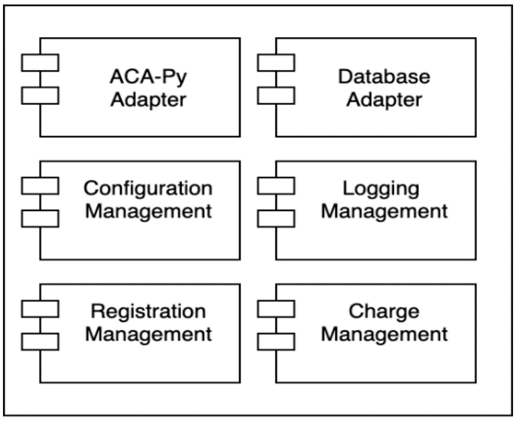


Figure 19. The CS software modules.

4.7. EVSEO Application

The Electric Vehicle Supply Equipment Operators (EVSEO) perform a key role in the ecosystem, which usually operate at the local or regional level due to the nature of the service they provide. The EVSEO are the edge interface with the EVO as they own and maintain the CS used to charge the EVs while releasing energy to the EVs and measuring the energy released. The EVSEO application serves two main purposes: to allow the EVSEOs to issue VCs to the CS devices that allow the EVO mobile application to recognize them as legitimate CS and to process and validate the information collected during the EV charging process; the consolidated information is recorded in a transaction ledger to allow further processing by the EMSP. Figure 20 identifies the set of use cases considered relevant to fulfil the EVSEO functional requirements.

The first set of use cases presented in Figure 20 are associated with the CS registration and credentialing, following a similar approach to that implemented within the EMSP to register EVOs. The lower set of use cases in Figure 20 involve the minimal set of operations needed by the EVSEO to manage, store, and share all the information gathered during an EV charging. Figure 21 presents the credential schema used for issuing a VC to a CS.

The EVSEO application's implementation follows the same software infrastructure and software architecture approaches used for the EMSP (previously presented in Figure 10). Considering those similarities, the current section will only highlight the differences between the components. Figure 22 presents the modules composing the EVSEO software prototype. Adding to the modules presented for the EMSP application in Figure 13, it adds a Transaction Ledger Adapter and a Transaction Management Module, responsible for processing the transaction data originating in the CS.

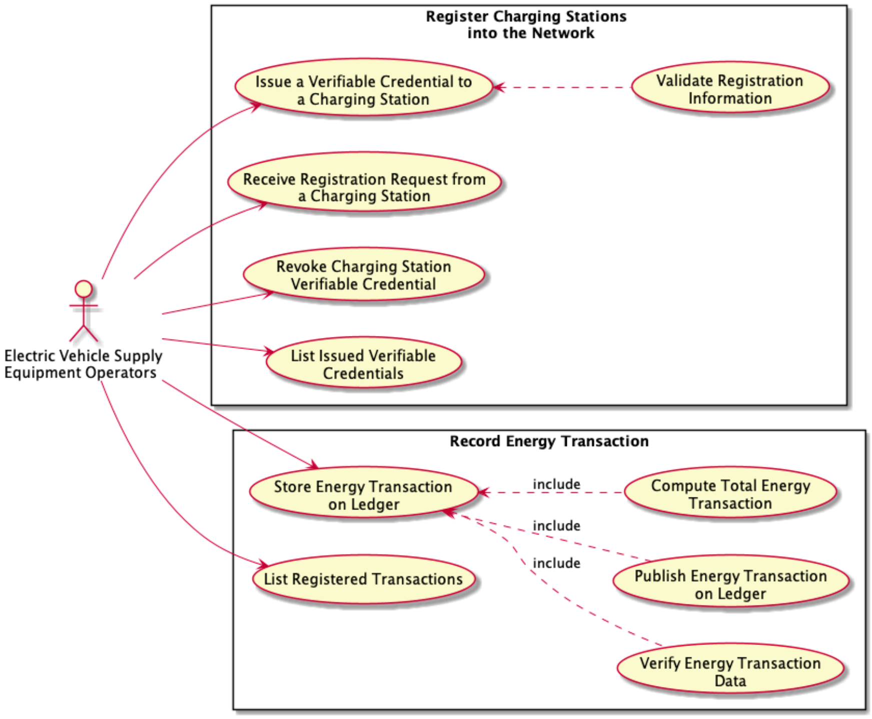
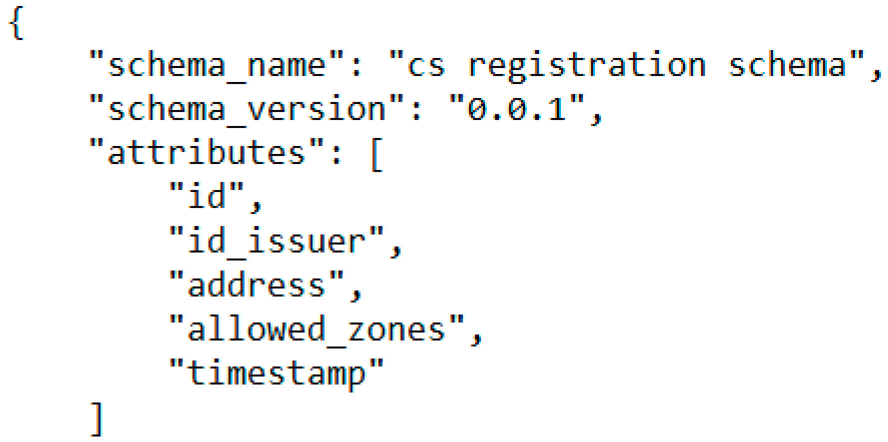

Register Charging Stations into the Network
Issue a Verifiable Credential to a Charging Station
Validate Registration Information
Receive Registration Request from a Charging Station
Revoke Charging Station Verifiable Credential
List Issued Verifiable Credentials
Electric Vehicle Supply Equipment Operators
Record Energy Transaction
Store Energy Transaction on Ledger
include
Compute Total Energy Transaction
include
include
Publish Energy Transaction on Ledger
List Registered Transactions
Verify Energy Transaction Data

Figure 20. Use cases implemented by the EVSEO.



```
{
    "schema_name": "cs registration schema",
    "schema_version": "0.0.1",
    "attributes": [
        "id",
        "id_issuer",
        "address",
        "allowed_zones",
        "timestamp"
    ]
```

Figure 21. CS verifiable JSON credential schema definition.

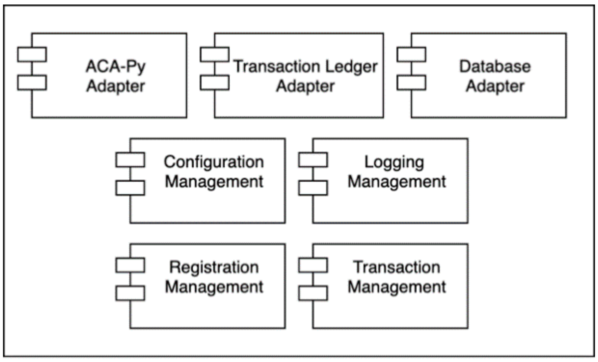


Figure 22. EVSEO software modules. EVSEO implementation.

5. Validation

Our proposition's validation was performed in two separate steps, targeting the validation of the physical devices' implementation and the complete solution as a platform. The physical device prototype calibration followed the same process used in [27]. With different consumption rates (between 0 A and 30 A), several electric devices were used to evaluate the precision and accuracy of the current measures gathered by the energy measurement device. The stability of the device after an extended period of continuous usage (24 h) was evaluated. The accuracy and precision levels measured are in line with the published information for the current transformer. After the device calibration, the current values measured reported a 5% error on average regarding the device stability; during extended periods of continuous usage, no significant impacts on the device behaviour were observed.

For implementing a sound validation scenario, it would be necessary to have several EVOs spread (and moving along) a wide geographical area and simultaneously engaging in partnerships with EMSP and EVSEO companies. In our scenario, which is a permissioned consortium one, the EMSPs act as energy brokers from EV owners to EVSEOs. Due to this scenario's inherent complexity, the platform design was validated with a simulation. We simulated several EVOs with contracts with different EMSPs, charging their EV within several CSs belonging to different EVSEOs. Figure 23 summarizes the scenario used for the simulation. The setup was built with four EVOs using VC issued by two different EMSP companies and four CSs conceptually belonging to two different EVSEO companies. Without loss of generality and establishing parallelism with a real use case, EMSP.1 and EVSEO.1 are companies operating in one geographical area or country, having customers (EVOs) and CSs, respectively, in that area. EMSP.2 and EVSEO.2 are companies operating in another geographical area, being the e-roaming concept exploited when customers from one area consume services in other areas provided by local companies. To simulate the physical devices composing the platform, respectively, the EVO Mobile app and the CS software agents were also built and packaged into independent software containers.

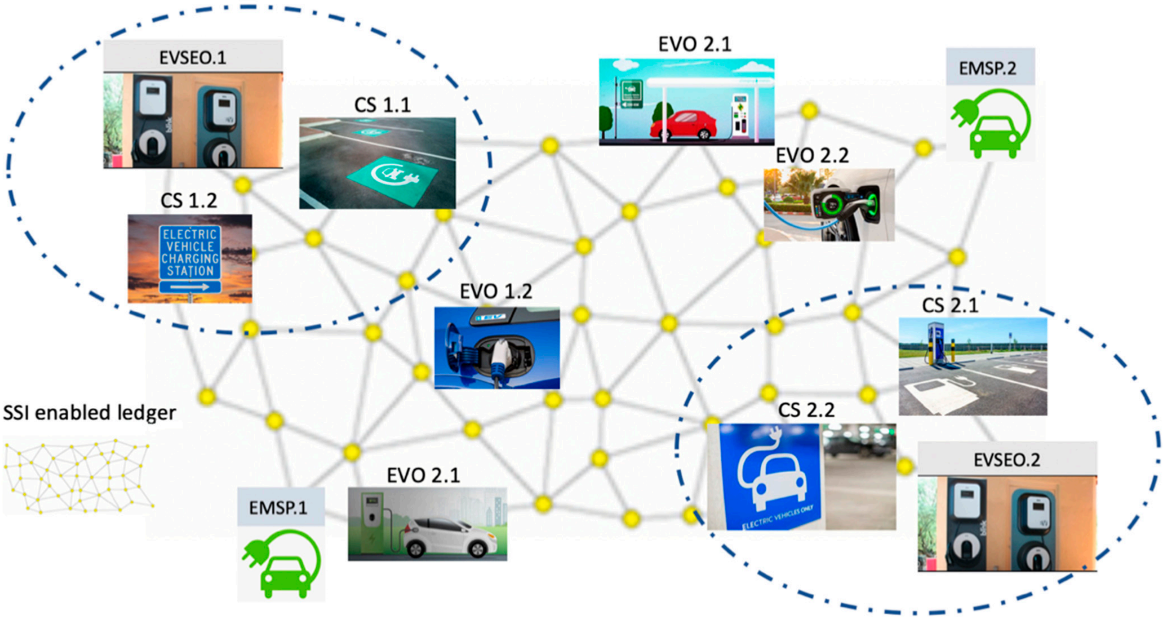
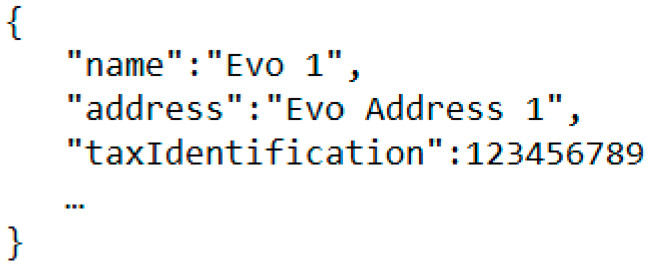


Figure 23. Our simulation scenario.

The simulation environment was set up using several separated containers: EVOs, EMSPs, CSs, and a self-sovereign identity (SSI) ledger, represented in Figure 23. This simulation environment runs on a laptop with a 2.8 GHz Quad-Core Intel i7, 16 GB 1600 MHz DDR3.

5.1. Simulation Setup

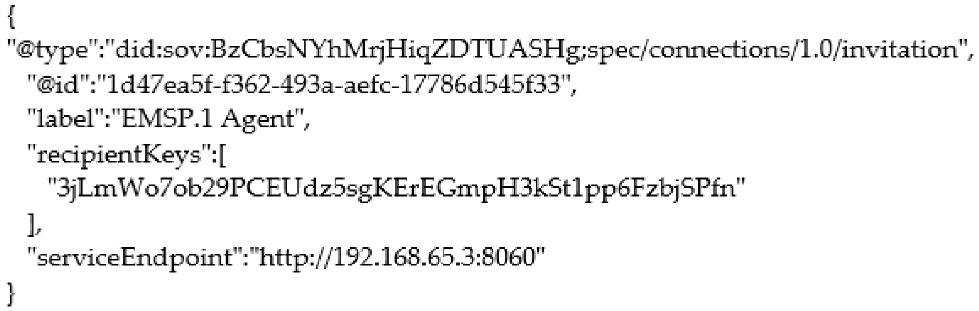
The simulation environment setup starts with the registration of the EVOs on each EMSP; each EVO simulation requests the selected EMSP the metadata describing the information required for registration as presented on Figure 6. Upon receiving the message, the EVO sends to the EMSP a message containing the required fields, as presented in Figure 24.



```
{
    "name":"Evo 1",
    "address":"Evo Address 1",
    "taxIdentification":123456789
    …
}
```

Figure 24. JSON format of EVO verifiable credential schema definition.

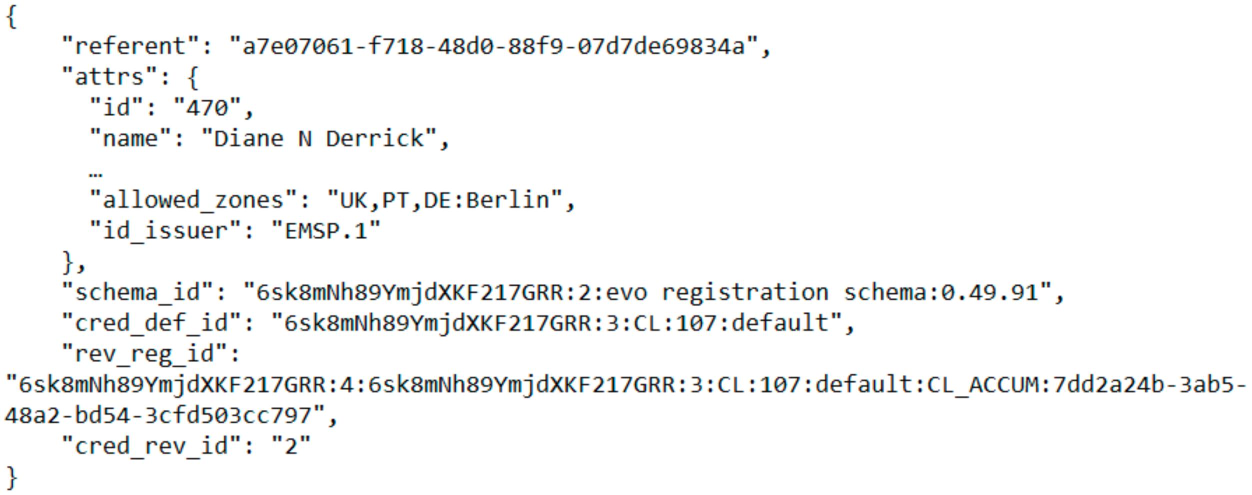
After processing the registration information (assumed as correct for simulation purposes), the EMSP sends a unique connection invitation message to the EVO, as presented in Figure 25.



```
 {
 "@type":"did:sov:BzCbsNYhMrjHiqZDTUASHg;spec/connections/1.0/invitation",
   "@id":"1d47ea5f-f362-493a-aefc-17786d545f33",
   "label":"EMSP.1 Agent",
   "recipientKeys":[
     "3jLmWo7ob29PCEUdz5sgKErEGmpH3kSt1pp6FzbjSPfn"
   ],
   "serviceEndpoint":"http://192.168.65.3:8060"
 }
```

Figure 25. EMSP generated connection invitation.

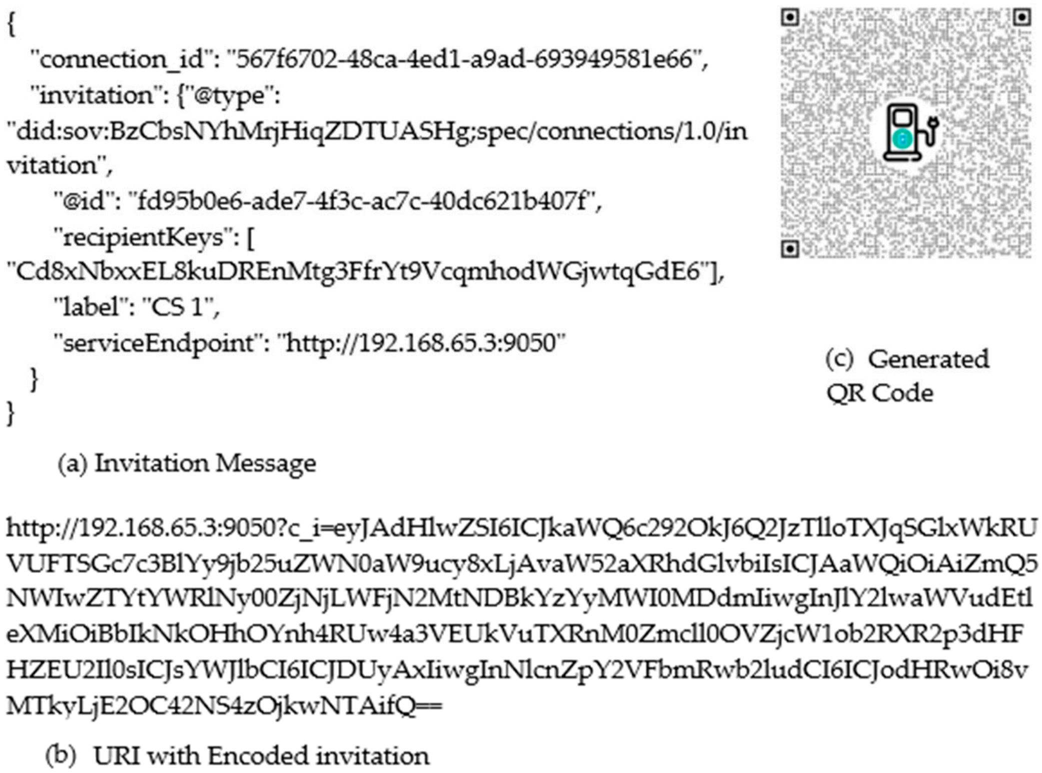
In answer to the connect invitation, the EMSP issues and offers to the EVO a VC, represented in Figure 26 (segment of the issued credential), to be used to prove her/his identity when connecting to a CS.



```
{
    "referent": "a7e07061-f718-48d0-88f9-07d7de69834a",
    "attrs": {
      "id": "470",
      "name": "Diane N Derrick",
      …
      "allowed_zones": "UK,PT,DE:Berlin",
      "id_issuer": "EMSP.1"
    },
    "schema_id": "6sk8mNh89YmjdXKF217GRR:2:evo registration schema:0.49.91",
    "cred_def_id": "6sk8mNh89YmjdXKF217GRR:3:CL:107:default",
    "rev_reg_id":
"6sk8mNh89YmjdXKF217GRR:4:6sk8mNh89YmjdXKF217GRR:3:CL:107:default:CL_ACCUM:7dd2a24b-3ab5-
48a2-bd54-3cfd503cc797",
    "cred_rev_id": "2"
}
```

Figure 26. EVO VC offered by the EMSP upon registration.

Similarly, using the CS VC schema, the same setup is performed for the CSs. They aim to identify the CSs, the EMSP issues, and provide the CS with a VC to allow the EVO to verify the CS identity. After the CS registration, the CS generates an invitation to connect message (Figure 27a), which is encoded in a URL to be used by the EVO to establish the connection to the CS. Figure 27b presents the generated URL with the encoded invitation to connect as well as the QR code encoding the URL in Figure 27c. To complete the environment setup, the CS endpoint URLs, as presented in Figure 27b, are configured in the EVO agents, and the EVSEO endpoints are configured in the CS.



```
{
  "connection_id": "567f6702-48ca-4ed1-a9ad-693949581e66",
  "invitation": {"@type":
"did:sov:BzCbsNYhMrjHiqZDTUASHg;spec/connections/1.0/in
vitation",
    "@id": "fd95b0e6-ade7-4f3c-ac7c-40dc621b407f",
    "recipientKeys": [
"Cd8xNbxxEL8kuDREnMtg3FfrYt9VcqmhodWGjwtqGdE6"],
    "label": "CS 1",
    "serviceEndpoint": "http://192.168.65.3:9050"
  }
}
```

(a) Invitation Message

(c) Generated QR Code

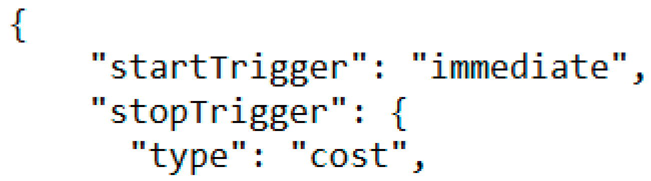
```
http://192.168.65.3:9050?c_i=eyJAdHlwZSI6ICJkaWQ6c292OkJ6Q2JzTlloTXJqSGlxWkRU
VUFTSGc7c3BlYy9jb25uZWN0aW9ucy8xLjAvaW52aXRhdGlvbiIsICJAaWQiOiAiZmQ5
NWIwZTYtYWRlNy00ZjNjLWFjN2MtNDBkYzYyMWI0MDdmIiwgInJlY2lwaWVudEtl
eXMiOiBbIkNkOHhOYnh4RUw4a3VEUkVuTXRnM0Zmcll0OVZjcW1ob2RXR2p3dHF
HZEU2Il0sICJsYWJlbCI6ICJDUyAxIiwgInNlcnZpY2VFbmRwb2ludCI6ICJodHRwOi8v
MTkyLjE2OC42NS4zOjkwNTAifQ==
```

(b) URI with Encoded invitation

Figure 27. CS invitation to connect message. (**a**) Invitation Message; (**b**) Generated QR Code; (**c**) URI with Encoded invitation.

5.2. Simulation

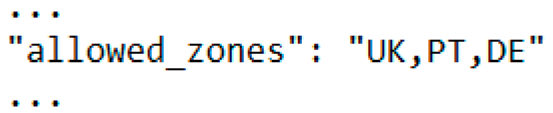
Using a random pattern, each EVO agent simulates a charging operation using one of the configured CS endpoints randomly drawn from the endpoints configured. Each charging operation is composed of the exchange of messages presented in this section. The initial connection is established by the EVO with the CS using a pre-configured invitation message, as presented in Figure 27c. After establishing the connection between the EVO and the CS, the EVO requires the CS to present a VC (issue by an EVSEO) verifying that the credentials are not expired or revoked to verify the CS identity. After successful verification of the CS identification, the EVO issues a charging request detailing the charging requirements, as presented in Figure 28.



```
{
    "startTrigger": "immediate",
    "stopTrigger": {
      "type": "cost",
```

Figure 28. Charging request message.

Upon receiving the charging request, the CS requires the EVO to present a valid verifiable credential (issued by the EMSP), containing the CS Zone in the allowed_zones attribute of the credential. In response to the CS request, the EVO presents a valid VC, with the allowed_zones attribute (Figure 29) granting the EVO permissions to charge her/his vehicle in that CS. Upon the credential reception, the CS verifies that the credentials are valid and not expired or revoked.



```
...
"allowed_zones": "UK,PT,DE"
...
```

Figure 29. Presented credential.

After receiving the VC claim, the CS enables the charging device and charges the vehicle until a stop condition is reached (user limits achieved, vehicle disconnected). For the simulation, the CS assumes that the charging process is finished after a certain time and sends one accounting message to the EVSEO containing the VC claim presented by the EVO as well as the accounting data associated with the current energy transaction. The communication process between the CS and the EVSEO follows a similar process used between the EVO and the CS; the CS connects to the EVSEO using the pre-configured EVSEO endpoint and after establishing an authenticated communication channel, sends the accounting message to the EVSEO with all the information related to that charging operation. During the simulation process, the metrics related to the exchange of messages between the EV charging network stakeholders were gathered (Table 1).

Table 1. Metrics for the simulation process.

Charging Events	**300**
Exchanged Messages (excl credentials exchange)	600
Exchange Credentials Time	244.53 s
Issued Verifiable Credentials	4
Exchanged (Presented) Verifiable Credentials	900
Average Time/Message	0.41 s
Average Connect Time	0.55 s

6. Conclusions and Future Work

We explore a blockchain ledger combined with Decentralized Identifiers usage for the Electric Vehicle roaming charging process, particularly to register charging transactions and identity verification. In this process, identity management regarding EV drivers and Charging Stations (CS) plays a critical role in allowing flexibility and interoperability among different charging systems. We implement blockchain-based digital identity management using Hyperledger Indy/Aries. This approach avoids charging specific cards used as an authentication process among charging systems. In this scenario, interoperability among different countries can be reached, allowing an EV charging roaming process. Additionally, CSs can be joined through the authentication process, allowing private entities participating in this charging approach to increase the number of charging spots. This approach allows energy accounting transactions in a secure and private environment, supported by public keys encryption and Zero-knowledge proof backed by Hyperledger Indy, and the payment process can be performed with digital currency. Price fluctuations and changes due to renewable availability can be managed with smart contracts, which we intend to address in future work.

Author Contributions: J.P.M. performed all the development work; J.C.F. is a thesis supervisor and organized all work in the computer science subject; C.F.d.S. collaborated on the blockchain, distributed ledger, self-sovereign digital identities and revised the document. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Foundation for Science and Technology (FCT) through ISTAR-IUL's project UIDB/04466/2020 and UIDP/04466/2020 and has been partially supported by Portuguese National funds through FITEC programa Interface, with reference CIT "INOV-INESC Inovação-Financiamento Base".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work is financed by national funds through FCT - Fundação para a Ciência e Tecnologia, I.P., under the project FCT UIDB/04466/2020. Furthermore, the author thanks the Instituto Universitário de Lisboa and ISTAR-IUL, for their support.

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

References

1. *Emerging Best Practices for Electric Vehicle Charging Infrastructure*; The International Council on Clean Transportation (ICCT): Washington, DC, USA, 2017. Available online: https://theicct.org/sites/default/files/publications/EV-charging-best-practices_ICCT-white-paper_04102017_vF.pdf (accessed on 18 March 2021).
2. Directive 2014/94/EU of the European Parliament and of the Council of 22 October 2014 on the Deployment of Alternative Fuels Infrastructure. 2014, 20. Available online: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014L0094&from=EN (accessed on 17 March 2021).
3. Adam, M. Accelerating E-Mobility in Germany. In *Springer Briefs in Law*; Springer International Publishing: Cham, Swizerland, 2016; ISBN 978-3-319-44883-1.
4. Navigant Research Electric Vehicle Charging Services. 2016. Available online: https://www.navigantresearch.com/-/media/project/navigant-research/reportfiles/wpsasev18navigantresearchpdf.pdf (accessed on 17 March 2021).
5. Saxena, N.; Grijalva, S.; Chukwuka, V.; Vasilakos, A.V. Network Security and Privacy Challenges in Smart Vehicle-to-Grid. *IEEE Wirel. Commun.* **2017**, *24*, 88–98. [CrossRef]
6. Paul, S.; Ni, Z. Vulnerability analysis for simultaneous attack in smart grid security. In Proceedings of the 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 23–26 April 2017; IEEE: Washington, DC, USA, 2017; pp. 1–5.
7. Su, Z.; Xu, Q.; Luo, J.; Pu, H.; Peng, Y.; Lu, R. A Secure Content Caching Scheme for Disaster Backup in Fog Computing Enabled Mobile Social Networks. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4579–4589. [CrossRef]
8. Wang, J.; Tang, J.; Yang, D.; Wang, E.; Xue, G. Quality-Aware and Fine-Grained Incentive Mechanisms for Mobile Crowdsensing. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016; IEEE: Nara, Japan, 2016; pp. 354–363.
9. Wu, Y.; Qian, L.P.; Mao, H.; Yang, X.; Zhou, H.; Shen, X. Optimal Power Allocation and Scheduling for Non-Orthogonal Multiple Access Relay-Assisted Networks. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2591–2606. [CrossRef]
10. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
11. NRG-X-Change. A Novel Mechanism for Trading of Renewable Energy in Smart Grids. In Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems, Barcelona, Spain, 3–4 April 2014; SCITEPRESS—Science and and Technology Publications: Barcelona, Spain, 2014; pp. 101–106.
12. Aste, T.; Tasca, P.; Di Matteo, T. Blockchain Technologies: The Foreseeable Impact on Society and Industry. *Computer* **2017**, *50*, 18–28. [CrossRef]
13. Mengelkamp, E.; Notheisen, B.; Beer, C.; Dauer, D.; Weinhardt, C. A blockchain-based smart grid: Towards sustainable local energy markets. *Comput. Sci. Res. Dev.* **2018**, *33*, 207–214. [CrossRef]
14. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://git.dhimmel.com/bitcoin-whitepaper/ (accessed on 17 March 2021).
15. Dunphy, P.; Petitcolas, F.A.P. A First Look at Identity Management Schemes on the Blockchain. *IEEE Secur. Priv.* **2018**, *16*, 20–29. [CrossRef]
16. Lesavre, L.; Varin, P.; Mell, P.; Davidson, M.; Shook, J. A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems | CSRC. NIST Report. 2020. Available online: https://csrc.nist.gov/publications/detail/white-paper/2020/01/14/a-taxonomic-approach-to-understanding-emerging-blockchain-idms/final (accessed on 20 May 2020).

17. Mustafa, M.A.; Zhang, N.; Kalogridis, G.; Fan, Z. Smart electric vehicle charging: Security analysis. In Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 24–27 February 2013; IEEE: Washington, DC, USA, 2013; pp. 1–6.
18. Gan, L.; Topcu, U.; Low, S.H. Optimal decentralized protocol for electric vehicle charging. *IEEE Trans. Power Syst.* **2013**, *28*, 940–951. [CrossRef]
19. Aitzhan, N.Z.; Svetinovic, D. Security and Privacy in Decentralized Energy Trading through Multi-Signatures, Blockchain and Anonymous Messaging Streams. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 840–852. [CrossRef]
20. Mattila, J.; Seppälä, T.; Naucler, C.; Stahl, R.; Tikkanen, M.; Bådenlid, A.; Seppälä, J. *Industrial Blockchain Platforms: An Exercise in Use Case Development in the Energy Industry*; ETLA Working Papers 43: Helsinki, Finland, 2016.
21. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3154–3164. [CrossRef]
22. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, 1. [CrossRef]
23. Li, L.; Liu, J.; Cheng, L.; Qiu, S.; Wang, W.; Zhang, X.; Zhang, Z. CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2204–2220. [CrossRef]
24. Huang, X.; Xu, C.; Wang, P.; Liu, H. LNSC: A Security Model for Electric Vehicle and Charging Pile Management Based on Blockchain Ecosystem. *IEEE Access* **2018**, *6*, 13565–13574. [CrossRef]
25. Liu, C.; Chai, K.K.; Zhang, X.; Lau, E.T.; Chen, Y. Adaptive Blockchain-Based Electric Vehicle Participation Scheme in Smart Grid Platform. *IEEE Access* **2018**, *6*, 25657–25665. [CrossRef]
26. Erdin, E.; Cebe, M.; Akkaya, K.; Solak, S.; Bulut, E.; Uluagac, S. Building a Private Bitcoin-based Payment Network among Electric Vehicles and Charging Stations. In Proceedings of the 2018 IEEE International Conference on Blockchain (Blockchain-2018), Halifax, NS, Canada, 30 July–3 August 2018.
27. Martins, J.P.; Ferreira, J.C.; Monteiro, V.; Afonso, J.A.; Afonso, J.L. IoT and Blockchain Paradigms for EV Charging System. *Energies* **2019**, *12*, 2987. [CrossRef]
28. Daghmehchi Firoozjaei, M.; Ghorbani, A.; Kim, H.; Song, J. Hy-Bridge: A Hybrid Blockchain for Privacy-Preserving and Trustful Energy Transactions in Internet-of-Things Platforms. *Sensors* **2020**, *20*, 928. [CrossRef] [PubMed]
29. Naik, N.; Jenkins, P. uPort Open-Source Identity Management System: An Assessment of Self-Sovereign Identity and User-Centric Data Platform Built on Blockchain. In Proceedings of the 2020 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 12 October–12 November 2020; pp. 1–7. [CrossRef]
30. Mühle, A.; Grüner, A.; Gayvoronskaya, T.; Meinel, C. A survey on essential components of a self-sovereign identity. *Comput. Sci. Rev.* **2018**, *30*, 80–86. [CrossRef]
31. Reed, D.; Sporny, M.; Longley, D.; Allen, C.; Grant, R.; Sabadell, M. Decentralized Identifiers (DIDs) v1.0—Core Architecture, Data Model, and Representations. IT Security and Privacy—A Framework for Identity Management (ISO/IEC 24760-1). Available online: https://www.w3.org/TR/did-core/ (accessed on 2 January 2020).
32. Tobin, A.; Reed, D.; Windley, F.P.J.; Foundation, S. The Inevitable Rise of Self-Sovereign Identity. 2017, 24. Available online: https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf (accessed on 17 March 2021).
33. Verifiable Credentials Data Model 1.0. Available online: https://www.w3.org/TR/vc-data-model/ (accessed on 2 January 2020).
34. Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on Electronic Identification and Trust Services for Electronic Transactions in the Internal Market and Repealing Directive 1999/93/EC. Available online: https://ec.europa.eu/futurium/en/system/files/ged/eidas_regulation.pdf (accessed on 17 March 2021).
35. Leveqda e-Identification. Available online: https://ec.europa.eu/digital-single-market/en/e-identification (accessed on 26 January 2020).
36. Feige, U.; Fiat, A.; Shamir, A. Zero-Knowledge Proofs of Identity. *J. Cryptol.* **1988**, *1*, 77–94. [CrossRef]
37. nified Modeling Language, v2.5.1. Unified Model. Lang. Available online: https://www.omg.org/spec/UML/About-UML/ (accessed on 17 March 2021).
38. Fowler, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed.; Addison-Wesley: Boston, MA, USA, 2004; ISBN 978-0-321-19368-1.
39. QR Code. Wikipedia. 2020. Available online: https://pt.wikipedia.org/wiki/C%C3%B3digo_QR (accessed on 17 March 2021).
40. Hyperledger/Aries-Framework-Dotnet. Available online: https://github.com/hyperledger/aries-framework-dotnet (accessed on 2 March 2020).
41. Zxing/Zxing. Available online: https://github.com/zxing/zxing (accessed on 2 March 2020).
42. Jahn, M. Micjahn/ZXing.Net. Available online: https://github.com/micjahn/ZXing.Net (accessed on 2 March 2020).