Enabling identity for the IoT-as-a-Service business model

Santiago de Diego¹, Cristina Regueiro², and Gabriel Maciá-Fernández³

¹TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, 48160, Spain and also with the University of Granada, Granada, 18071, Spain
²TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, 48160, Spain
³Department of Signal Theory, Telematics and Communications, University of Granada, Granada, 18071, Spain, and also with the Information and Communication Technologies Research Centre (CITIC), Granada, 18071, Spain

Corresponding author: Santiago de Diego (e-mail: santiago.dediego@tecnalia.com).

This work has been partially supported by the Basque Country Government under the ELKARTEK program, project TRUSTIND (KK-2020/00054) and by Spanish Government-Ministry of Science and Innovation through project SICRAC (PID2020-114495RB-I00).

ABSTRACT The IoT-as-a-Service (IoTaaS) business model has already been identified by some people from both industry and academia, but has not been formally defined. IoTaaS offers IoT devices on demand, with considerable cost savings and resource optimization. In addition, it enables different applications to reuse the existing devices. However, this business model is associated with different technological challenges that need to be addressed, one of which is the identity problem. Focusing on this, self-sovereign identity (SSI) schemes have proven to provide better privacy and scalability than traditional identity paradigms, which is especially important in the IoT owing to its characteristics. In this paper, we formally analyze an IoTaaS business model, identifying and detailing its main technological challenges. In addition, we tackle the identity problem of this business model and propose an SSI-based identity management system, which is compliant with the existing standards from the W3C, and include a performance evaluation.

INDEX TERMS IoT, as-a-Service, IoTaaS, Identity Management, SSI

I. INTRODUCTION The “as-a-service” label has become incredibly popular nowadays. In the context of the Internet, it is not necessary to own a resource to use it. In “as-a-service” models, users of resources can reduce costs, as they avoid investing on its acquisition, while owners of resources can monetize their investments by offering services based on those assets to interested users. In other words, “as-a-service” often implies a win-to-win situation for both the user and the owner of a resource. The Netflix use-case [1] is a perfect example of a successful “as-a-service” solution, where a traditional use-case (watching films on the Internet) is redefined as a subscription-based business model. However, other business models could also be mentioned, such as the pay-per-use model, where users pay for the time they use the service. Another widely known “as-a-service” solution is Amazon Web Services (AWS) [2], where any user can use cloud computing and storage capabilities, thus paying only for the utilized amount of time or traffic.

On the other hand, the expansion of the Internet of Things (IoT) is remarkable. IoT is being integrated into numerous solutions owing to its ubiquity. According to the study in [3], the number of connected devices around the world is expected to increase from 20.4 billion in 2020 to 75.0 in 2025. Considering the expected number of IoT devices, it is likely that many of them remain underused. Thus, it is necessary to think about ideas and solutions to reuse existing devices, rather than creating new ones, thus avoiding the negative consequences of overpopulating our environment with IoT devices [4]. In addition, IoT devices are often deployed in a particular environment, so solutions that effectively enable the reuse of these devices for other purposes are very welcomed. For example, an IoT device measuring temperature at a specific location may be useful for researchers interested in its data for its own application. In this context, the IoT-as-a-Service (IoTaaS) business model is being coined as a solution to these problems, adding special value when it comes to enabling different applications aimed at optimizing the use of IoT devices. However, despite some blog entries from industry [5] have scratched the surface over this term, as well as the research community [6], [7], these references do not go deep into the business model description and operation. Thus, as the first contribution of this paper, we describe a formal definition of this business model to identify its main technological challenges. This could help other research efforts focus on related problems.

In parallel with the increased popularity and ubiquity of the IoT, many issues and challenges that need to be carefully addressed have arisen. In particular, the security of IoT devices is crucial. As stated by numerous authors [8], [9], it is not trivial to achieve an overall security level in IoT...
ecosystems. Some IoT devices, such as wearables, keep sensitive information from their users and, for this reason, attackers are increasingly targeting these devices. Unfortunately, some well-known security incidents such as Stuxnet [10], the Mirai botnet, and other IoT zombie-related attacks [11], which infect millions of IoT devices, have emphasized the necessity of a security point of view when developing IoT-related solutions. Other more generalist attacks include data theft, device malfunction, and remote control [12]. In this sense, solutions that tackle security problems associated with IoT ecosystems are welcomed. Identity management falls under the umbrella of previously described security problems. It can be informally defined as the process of assigning identities to the different actors in an IT ecosystem and is a relevant aspect in the security process, as many attacks that are based on techniques such as impersonation and identity theft, can be prevented by solid identity management. For example, the Sybil attack also affects the IoT [13] and is directly caused by deficient identity management. Good identity management practices also help to guarantee non-repudiation by identifying the different actors in the system. To provide some insights about the challenges associated with identity, it is estimated that the cost of identity assurance processes exceeds 3.3€bn a year in the UK and $22bn in the USA by extrapolating the size of the population [14], without taking into account the costs associated with storage, protection, breaches, and regulations. By adding IoT into the equation, the numbers were completely exorbitant. Hence, it is necessary to manage the identity of all existing devices. Consequently, it seems clear that it is worth thinking about new strategies for identity management in the IoT world. Thus, as a second contribution of this paper, we present a system to manage identity in the IoTaaS business model.

Some authors [15] provided a broad overview of the concept of identity and how it evolved over time. To sum up, identity has suffered several transformations during the last few years, starting from an isolated and centralized model and later moving to a federated model. However, these models suffer from serious scalability and security issues. The last suggested model is the self-sovereign identity (SSI) [14]. As explained in the following section, this model alleviates some of the main problems in previous schemes and, therefore, our identity model for IoTaaS will be based on SSI.

In summary, the present research work provides two main contributions: a) to formally present the IoT-as-a-Service (IoTaaS) business model with its own characteristics and its relevant technological challenges, and b) to propose an IdM system based on SSI for this business model, including a performance evaluation.

The different sections of the paper are organized as follows: Section II shows the advances from the research community regarding the "IoT-as-Service" model and analyzes the current state of the art in self-sovereign identity for the IoT. Section III formally presents the IoT-as-a-Service business model and describes its associated technological challenges. Section IV introduces the reader to the basic concepts of the SSI paradigm. Section V describes the proposed design of the SSI-based identity system for the business model. Section VI validates the proposed identity system in terms of performance. Finally, some conclusions are drawn in Section VII, and further research directions are suggested.

II. STATE OF THE ART

As discussed in the introductory section, there is not much literature about the concept "IoT-as-a-Service" Some authors [6], [7] introduced this concept, but only described it as a feasible business model and did not analyze all the technological challenges that it implies. Others [16] go a little bit further by describing four essential components in the IoTaaS: sensors and gateways, middleware, backend servers, output interface, but still lacks in properly proposing a model. A search in Google Scholar using the keywords "IoT as a Service" or "IoT-as-a-Service" shows different works including these terms. However, none of them refers to the concept of IoT as a Service, as considered in the present work. For example, some authors [17] presented an IoT solution for public safety and disaster response (PSDR), while others [18] presented a formal model integrating IoT and eHealth, and others [19], [20] referred to IoT-as-a-Service as a combination between IoT and cloud. Finally, other authors [21] propose an infrastructure for hosting IoT workloads in the cloud.

Regarding the SSI model from a research point of view, some studies have proposed the application of the SSI paradigm to the IoT. First, some authors [22] thoroughly explained the SSI paradigm and presented decentralized identifiers (DIDs) and verifiable credentials for IoT. In addition, they analyzed the suitability of SSI for IoT, proving that it is a better method than the use of pretty good privacy (PGP) and X.509 certificates. Other studies have studied the applicability of the SSI paradigm to the IoT world. For example, some authors [23] introduced some SSI concepts and showed use cases for the industrial IoT, while others [24] briefly mentioned the intersection between SSI and IoT. These works show the “big picture” but they did not delve into any Credential Exchange protocols or technical details. Other authors [25] proposed the use of DIDs as identifiers for IoT devices and accurately studied the requirements for IoT devices to run an SSI-based identity management system, even proposing a proxy-based solution for extremely constrained IoT devices. Other proxy-based solutions were also proposed by other research works [26], in this case the IoT Exchange, to connect IoT devices with users. However, they do not suggest a specific verifiable credentials schema for the IoT and restrict themselves to just analyzing DIDs as suitable identifiers for the IoT. Other authors [27] presented an SSI protocol to trace the origin of IoT devices.

SSI uses a decentralized registry that stores information such as DIDs and DID documents (see Section IV for details). This registry can be implemented using distributed ledger technology (DLT), thus improving the overall security of the system by preserving the integrity and availability of the
stored information. For example, some authors[28] proposed an SSI scheme based on IOTA [29] for its DLT backbone, and applied it to a rental car use case. The main problem of this proposal is that IOTA is not fully decentralized because it still depends on the coordinator, an element that centralizes the consensus process. Similarly, others[30] also proposed IOTA as a DLT for implementing the SSI scheme. They accurately defend that IOTA is permissionless, has no transaction costs, and scales well; however, the main drawback is that IOTA is not fully decentralized. Furthermore, some mixed designs have been proposed in which SSI is used in combination with other elements in the same infrastructure [28][30].

Finally, there are studies that focus on specific applications. For example, some authors [31] proposed the use of a smart band for biometric identification following an SSI scheme, while others [32] particularized the SSI paradigm for Internet of Vehicles (IoV).

From an industrial point of view, there are several technologies and frameworks that enable SSI for people, including Hyperledger Indy (Sovrin) [33], uPort [34] (currently known as Serto), Blockstack [35], Veres One [36], Jolocom [37], Identity.com [38], Uniquid [39], ShoCard [40] and DIF [41], which gather some of them under his umbrella. Moreover, the Sovrin Foundation has conducted research on SSI and the IoT [42] and provides some useful insights for its application.

However, none of these works consider the particularities of the IoT-as-a-Service business model, such as the different actors who participate in it and the differences between usage and ownership of the devices. In addition, there is a lack of studies on the performance evaluation of SSI-based IdMs for IoT.

III. ANALYSIS OF THE IOTAAS BUSINESS MODEL

In this section, we analyze the IoTaaS business model to identify its main technological challenges.

A. ACTORS IN THE BUSINESS MODEL

Let us consider a person or entity denoted as owner, which possesses a set of IoT devices \( \{D_n \mid n \in \mathbb{N} \} \). Each device \( D_n \) can be “hired” by any interested consumer (person or entity) during a period of time. We will denote these consumers as \( \{C_m \mid m \in \mathbb{N} \} \). With the term “hired,” it means that the device is accessed for the data or services that it provides.

For example, a consumer might be interested in the data provided by an IoT device, such as temperature or wind power readings, but also in using some services from this device, such as a connectivity test against other systems, or a notification in response to certain events. The amount of money each consumer will pay for the data or services will depend on the amount of time or resources that he/she will consume from these devices, as well as the type of the requested service, following a pay-per-use business model. It is also possible to follow a subscription-based business model in which consumers pay a periodic amount of money (subscription) to access data or use services from devices. Examples of possible real-life applications for IoTaaS are sufficiently large to not be detailed in this paper.

This nomenclature can be extrapolated by considering several owners, \( \{O_k \mid k \in \mathbb{N} \} \), each having its own set of devices. Each consumer \( C_m \) will be interested in obtaining services or data from devices, \( D_n \), if and only if those devices meet some quality requirements. Certification entities, named as \( \{CE_l \mid l \in \mathbb{N} \} \), indicate that devices truly meet these quality requirements. In other words, they trust customers. Many actors, such as external auditors or manufacturers, can be certification entities because they can certify some attributes of the devices. Certification entities differ from Certificate Authorities (CAs). Certificate Authorities are companies or organizations that issue digital certificates (normally X.509) to different entities on the Internet. A certification entity, as described in the IoTaaS, is the actor who certifies some attributes of a device. This can be done by using digital certificates or by any other means. In the proposed identity model, this is done by using verifiable credentials, so we have decided to name it “certification entity”, as this is a broader term than Certificate Authorities. The different roles of the business model are presented in Fig 1.

B. TECHNOLOGICAL CHALLENGES IN THE IOTAAS BUSINESS MODEL

Now, we briefly describe some of the main technological challenges that appear when facing a real implementation of the IoTaaS business model.

1) SERVICE ENDPOINT IDENTIFICATION AND AVAILABILITY

To begin with, one of the relevant aspects to tackle is defining how the services provided by devices will be offered to and accessed by different consumers. With respect to access to services, the publish-subscribe pattern is an option for communication between devices and consumers. Regarding how to reach the services offered to consumers, it is quite straightforward that we need mechanisms to connect consumers with devices. One alternative taken from other as-a-service solutions is the use of a marketplace, that is, a mechanism to connect IoT devices with potential consumers. For our purpose, this marketplace is an interface where IoT devices are listed to be chosen by consumers. The marketplace also has the goal of preventing IoT devices from saturating consumers’ requests, thus alleviating possible
denial of service (DoS) attacks, which can be especially severe if these devices are resource constrained. However, the final design of this marketplace could vary and become more complex depending on the requirements of the final implementation, as stated by some authors [43]. Thus, the marketplace appears as an optional but highly desirable element, because it enables more-constrained IoT devices to participate in the proposed IoTaaS.

2) COMMUNICATION MODEL
This challenge has been identified as an open research topic by several researchers. For example, some authors [44] proposed a privacy-preserving query scheme for IoT using homomorphic encryption, which can add an extra layer of privacy to the communications. Transferring data among IoT devices themselves or to the marketplace can be made by using IoT-specific protocols, such as MQTT-S, as usual in wireless sensor networks [45], or any other IoT-related protocol. In addition, secure versions of the MQTT and MQTT-S protocols might be considered, such as SMQTT and SMQTT-S [46], if the security levels require them. Furthermore, the chosen solution must incorporate a mechanism to allow queries so that consumers can search for devices using different parameters, making it easier to select the most suitable one for their needs. In addition, the chosen mechanism must be standardized, and owners will need to be aware of this standard before publishing their devices in the marketplace. As will be shown in Section V, the present research proposes the use of queries over SSI verifiable presentations [47] to achieve this goal. Communication between consumers and the marketplace can be achieved using HTTPS or any other secure protocol.

Service semantic definition

3) SERVICE SEMANTIC DEFINITION
There are some attributes of IoT devices that are important to be shown and queried in IoTaaS, namely:
- Identifier
- Data type provided by the device
- Quality Certificates
- Cost
- Location and geographic scope
- Owner

Despite this being a minimum set of necessary attributes to implement the business model, depending on the final implementation of the business model, some more attributes might be included in this list. Two of these attributes are especially relevant for the identity problem to generate trust: quality certificates and owners. The quality certificate attribute collects all the information related to the overall quality of the IoT device, while the owner attribute defines the owner of the device. Both attributes are directly related to the identity problem, and in Section V, we deal with how they can be modeled.

4) OPTIMIZATION
Another challenge to be analyzed is optimization. The range of possible scenarios to be analyzed in IoTaaS was large. Further studies on IoTaaS may study different optimization problems associated with the business model. For example, suppose that the expense of an IoT device $D_n$ is $E_n$ and its expected benefit is $B_n$. Both $E_n$ and $B_n$ can be functions of other variables, such as the type of data provided by the device, its location, or its quality. Then, the optimization problems can be formulated with different objective functions that allow us to know, for example, how to distribute the devices among different locations, how many quality certificates must the devices have, or the number and type of devices to be acquired to increase the profit. Complex economic models can be built from this starting point, for example, models that consider the implicit cost of depreciation, which would impact $E_n$, or variable benefits $f(B_n)$ depending on non-stationary variables, such as season or demand.

5) PAYMENT
Finally, another technological challenge is to provide mechanisms that permit payment transactions. These mechanisms should be carefully designed and suited for different scenarios. For example, micropayments could be made from consumers to owners (traditional scheme), or directly carried out to IoT devices, leveraging infrastructures such as DLT (e.g., Blockchain) to manage and track transactions. With the latter solution, the process could be made without owners’ intervention, using smart contracts as a method to automatically send the money when a certain balance is reached. In such an approach, it would be possible to decentralize the payment solution, thus improving the privacy of involved parties by using any type of blockchain address, such as the ones implemented in Bitcoin [48] or Ethereum [49]. Still, it should be considered that these addresses are pseudo-anonymized, which means that privacy cannot be fully guaranteed, and attacks such as de-anonymisation attacks [50], which attempt to reveal the identity behind an address, can occur. Some research works have shown blockchain-based micropayment methods for IoT. For example, some authors [51] implemented a micropayment method for IoT using the Bitcoin network, while others [52] proposed the use of the Lightning Network (LN) of Bitcoin to enable micropayments in IoT. Others [53] presented the idea of using blockchain as a service for IoT to manage device configuration, store sensor data and enable micro-payments. IOTA technology [29] was initially conceived to enable free transactions between IoT devices and can be effectively used to implement the payment method. Other authors [54] have advanced in this direction by implementing an IOTA-based micropayment solution for IoT.

6) IDENTITY PROBLEM
The identity problem must be determined by assigning identities to the different actors of the system. Traditionally, the identity problem on the Internet has been dealt with using what is known as the isolated model. In the isolated model, the service provider (SP) and the identity provider (IP) are the same entity, being considered a relying party. Users need to trust these components to store their identities, thus making them become honeypots for attackers. A risk in the
isolated model is that IP might easily run away with all the identities, or prevent the users from using the service, leaving them completely unprotected. In addition, it acts as a single point of failure (SPoF) because the entire system relies on it. With the emergence and popularity of services on the Internet, the isolated model became unmanageable because of the huge number of identities to be managed by a single SP, so a centralized model was developed to solve this issue. In this model, the SP and IP are not the same entities. Instead, different SPs share the same IP. However, it does not solve the previous issues, because the users still need to trust an external entity and the IP remains as an SPoF, thus becoming a honeypot for attackers.

Finally, the federated model differs from the centralized approach in the creation of trust relationships among different IPs. It allows the establishment of networks of IPs, improving the flexibility and overall trust of the system. However, the subjacent problem still remains, that is, users need to trust the IP, which can still invalidate the user’s identity at will. In addition, scalability issues are not solved; as the number of identities increases, the IP needs to properly manage them; at the same time, it also becomes a honeypot for attackers and an SPoF. In this context, the self-sovereign identity (SSI) paradigm emerges, and it intends to solve the issues associated with the previous identity models.

IV. SELF-SOVEREIGN IDENTITY PRELIMINARIES

SSI works with the concept of verifiable credentials (VCs) issued to a holder by a trustable issuer as a digital equivalent to traditional paper credentials, generated by government authorities. A verifier could use a verifiable data registry (VDR) to check whether the VCs presented by a holder are valid.

These VCs are kept private under the control of the holder, who can show them as proof when required from a verifier. "Under the control of the holder" means that VCs are normally stored in a wallet, i.e., a private user storage that contains them, as well as different holder private keys. Each corresponding public key is associated with an address that is known as a decentralized identifier (DID), with a structure defined by W3C [55]. DIDs provide a standard way to create permanent, globally unique, crypto-verifiable identifiers for people and organizations under their own control. Consequently, DIDs are the first verifiable identifiers that do not require registration authority. A DID can be looked up in a blockchain or any anti-tamper decentralized registry (VDR) to retrieve a DID document that contains information such as the subject of the DID, the associated public key, public credentials, network endpoints, authentication protocols, and digital signs. Private DIDs are also possible when involved parties want to keep their relationships private, so DIDs do not necessarily need to be recorded in the VDR.

In a credential verification process, the holder is requested to present a proof verifying that he fulfills certain requirements. This is known as verifiable presentation (VP), and is based on the use of VCs. For example, an IoT device can present a credential stating that it is waterproof. The verifier is the person or entity that verifies the VP issued by the holder. A claim proves certain attributes of the holder without disclosing the entire credential information. The use of claims avoids sharing personal and undesired information with third parties, but only the required information. This is normally achieved by using zero-knowledge proof (ZKP) cryptography to prove that it has a specific attribute without having to show the attribute itself. The W3C [47] standard proposes at least three proof mechanisms: JSON web tokens (JWT) secured using JSON web signatures (JWS), link data signatures and Camenisch-Lysyanskaya zero-knowledge proofs. More information about the relationship between verifiable credentials and JWT tokens can be read in Section 6.3.1 of the standard. Technically, by using ZKPs, a holder can use cryptography to generate a VP based on a VC without disclosing the attributes of the VC, but still allow a verifier to check if the proof is valid or not. Deeper information about zero-knowledge proofs and their relationship with SSI can be found in other works [56].

As can be deduced from the above, SSI provides many benefits [14] against the rest of the models for identity management: isolated, centralized, and federated models. First, it removes the need for "identity hubs" by empowering the users to manage their own identities. Second, and more importantly, when it comes to the IoT, it scales better as a consequence of removing these identity hubs, which is especially true in IoT environments [15], where ubiquity or the existence of a myriad of devices are essential features to be considered [9]. This fact clearly complicates the identity management process using traditional approaches. Finally, some authors [42] recognized some additional benefits when applying SSI to the IoT, which can be summarized as an increase in revenue, cost, and risk reductions. This study also analyzes which cybersecurity threats affecting the IoT can be mitigated using an SSI-based IdM.

One question to be addressed is whether IoT devices have sufficient resources to run an SSI-based identity scheme. As stated in Section II, some authors [25] precisely analyzed the minimum requirements that IoT devices must have in order to run an SSI solution and conclude that most devices are
1. \(D_n\) generates a VP containing its new VC

2. This VP is signed and sent to the marketplace to be indexed

3. \(C_m\) can see this VP published in the Marketplace and can check its authenticity against the VDR

Fig 3. Verification process

V. IDENTIFICATION MANAGEMENT SYSTEM FOR THE IOTAAS

We divide the proposal for the identity management process in IoTaaS in the three subprocesses. The first subprocess is called identification and initialization, and it is partially represented in Fig 2. The second subprocess, shown in Fig 2 and Fig 3, is called the Credential management subprocess, and the third one, shown in Fig 4, is called the service consumption subprocess. In the following, we detail them.

A. IDENTIFICATION AND INITIALIZATION

Before describing the proposal for identity management in IoTaaS, it is relevant to indicate that all the aforementioned actors in IoTaaS, that is, certification entity (owner, manufacturer, or whoever), device, and consumer, can play different roles from the identity management perspective, that is, issuer, holder, or verifier, depending on the stage of the process in which they participate. In what follows, we clarify this.

The first relevant aspect of the architecture is the identification of participants. Here, each device \(D_n\) has its own DID to be identified and its own private and public keys stored in its wallet for signing verifiable presentations. This DID allows the identification of IoT devices while preserving their privacy. The DID and its corresponding DID document creation and registration in the verifiable data registry (VDR) follow the guidelines provided by some authors [25] and are represented in Fig 2. It is required for the owner to enable his devices to participate in IoTaaS. This can be done either manually \((the\ owner\ creates\ the\ devices'\ DIDs\ and\ DID\ documents)\) or automatically \((the\ owner\ sends\ a\ one-time\ token\ to\ his\ devices\ and\ they\ create\ their\ own\ DIDs\ and\ DID\ documents)\). Ultimately, these DID and DID documents will be registered in the VDR and, consequently, these devices will be enabled in the IoTaaS business model.

Similarly, each consumer \(C_m\) and certification entity \(CE_i\) perform the same process of generating their own DIDs. These DIDs, together with their corresponding DID documents, are stored in the VDR after their creation.

Another element will be part of this scheme as a solution for the service endpoint identification and availability challenge, which is the marketplace. This element acts as a directory for devices, where consumers can check whether devices are available to provide the required data or services. In addition, it behaves as a storage for verifiable presentations (VPs), where devices store their proofs for their verifiable credentials (VCs). By doing so, consumers can check these VPs directly from the marketplace, avoiding saturation of devices, as discussed in Section III; in other words, it acts as a cache for VPs. As VCs are not stored in the marketplace, devices can maintain control of the information sent to the marketplace via VPs, which can be generated using ZKPs, thus ensuring privacy as needed. Note that the marketplace cannot generate ZKPs because this element is not the original holder of these VCs, so only the devices can generate them. ZKPs could be enabled in devices; if required, the marketplace acts as a proxy for VPs containing these ZKPs. Note that there is a tradeoff between security and performance because ZKPs could improve the privacy of devices, but their use implies that these devices would need to answer queries and verification requests from consumers, as well as generate ZKPs, which clearly impacts performance. In addition, notice that the marketplace is preferably implemented as a distributed system, thus avoiding becoming an SPoF. Centralization can be avoided by using decentralized technologies to implement the marketplace or even using multiple marketplaces [57]. Blockchain technologies are not a good candidate to be used to implement the storage of VPs in the marketplace because VPs cannot be removed from it, which goes against the right to erasure defined in the GDPR [58], but using a blockchain to ensure transparency through smart contracts and adding an extra layer of security to the marketplace could be a good solution, which can be tackled in future works. Other alternatives, such as distributed versions of the LDAP [59] can be used to implement this component. In addition, we can minimize the risk of identity theft attacks against the marketplace by using X.509 certificates to identify this component. Furthermore, VPs are signed by each \(D_n\), so no tampering can be made by a malicious marketplace because consumers can check the authenticity of the VPs by using a VDR.

B. CREDENTIAL MANAGEMENT SUBPROCESS

This subprocess deals with two different aspects: i) how VCs are issued and ii) how these credentials are verified.
Regarding credential issuing, we assume that a set of certification entities issues certain VCs to a device \( D_n \). Hence, from the point of view of identity management, the certification entities act as issuers and the device is a holder. Note that these VCs are part of the identity of devices and contain information about each \( D_n \), such as, for example, what quality certificates it has. Hence, a device’s identity is composed of all its VCs. For example, a manufacturer (acting as a trusted certification entity) could certify the waterproof property for all his devices issuing them a VC for this property. In addition, these devices may have a different owner, \( O_h \). The owner could also act as a trusted certification entity by sending VCs for specific features, such as ownership. Fig 2 represents this VC issuing process by a certification entity after DID creation and registration.

With respect to the verification process, consumers \( \{C_m; m \in \mathbb{N}\} \) in the IoTaaS model act as verifiers from an identity management process, while devices \( \{D_n; n \in \mathbb{N}\} \) play the role of holders. Now, each device \( D_n \) generates a VP from its set of VCs, so anyone can check that the selected claims are valid. This VP is signed by the devices and sent to the marketplace to be indexed. Now, the marketplace knows the DID of the device. Consequently, consumers can query the marketplace to retrieve information about each device to determine whether it satisfies their needs by checking its authenticity against a trusted VDR. This process is illustrated in Fig 3.

C. SERVICE CONSUMPTION SUBPROCESS

A consumer credential, which serves as proof of authorization, is a special kind of VC that allows a consumer to access data or services from an IoT device. This third subprocess involves the consumer getting a consumer credential for accessing data or services from an IoT device. The consumer credential has its own attributes, being relevant the issuanceDate, allowing anyone to know when it has been issued and the OrganizationDate, which contains the date when the credential is no longer valid, both specified in the W3C [47]. These two dates define the timeframe during which the consumer will be able to consume data from the IoT device. A pay-per-use payment method would require assigning a timeframe long enough and a revocation method. Consumer credentials cannot be updated, so any update in the credential, such as a new expirationDate implies a new issuing process. In this subprocess, the device plays the identity management role of an issuer, while the consumer acts as a holder. In addition, the device acts as a verifier for its consumer credentials, as it can verify this credential issued by itself to the corresponding consumer every time he wants to consume data from it. Note that the marketplace could also act as a verifier by checking the message signature from the device against the VDR. In addition, it is important to have a mechanism for the IoT device to revoke the consumer’s access in case of payment failure or breach of contract. In addition, some payment methods, such as the pay-per-use, may require the revocation of consumer credentials, where the consumer pays for the number of resources consumed by the device, instead of the time. Focusing on the revocation, there are no differences in this procedure compared with the standard SSI revocation process described by authors in [47]. Revocation is made by using a revocation list recorded in the VDR, so the holder will no longer be able to prove that he has this valid credential anymore. This approach has immense value for the issuer, because the number of IoT devices is enormous and the issuer does not need to answer requests for every party desiring to check whether a VC issued to a holder is valid or not, he just needs to update this revocation list. The service consumption subprocess is illustrated in Fig.4. First, a consumer \( C_m \) queries the marketplace to select the most suitable device for his/her needs. Remember that in the Credentials management process, devices have sent a VP with all the required VCs to the marketplace containing their identity attributes, so the marketplace can perform queries over the attributes in the VCs and return a list with the
devices’ DIDs that match the consumer query. Once the consumer has selected an available device \( D_n \) by its DID, he requests access to this device and sends it a micropayment. As explained in Section III, micropayments can be made either to the owner or to the device, and the latter is preferable in terms of the owner’s privacy. Then, if \( D_n \) agrees with \( C_m \) request, it sends a signed message to the marketplace. The marketplace then has information about whether \( D_n \) is available for use, so it can include the device in his list of available devices. Consumers only have to check this list provided by the marketplace without interrupting the normal operation of \( D_n \).

This prevents other consumers from selecting this device. Note that some devices may allow several concurrent consumers, while others may not. This entirely depends on the device configuration and capabilities and is beyond the scope of this study. At the same time, \( D_n \) sends a consumer Credential to \( C_m \) so that he can connect to the IoT device, while this credential is valid. Finally, the marketplace can check against the VDR if the signature of the device is valid. However, the use of consumer credentials as an authentication method adds security to the process while penalizing performance, because devices must verify this credential each time an authentication attempt from a consumer takes place. In this case, alternative mechanisms could allow controlled access to devices, such as the issuance of temporary access tokens. Further research can study the optimal combination between tokens and consumer credentials to achieve the best performance-security ratio.

Another interesting research topic could be the generation of access tokens based on a given VC. However, in Section VI, some performance measures are provided to show that the access mechanism based on consumer credentials does not actually imply performance degradation unless the device is extremely constrained.

### A. SECURITY CONSIDERATIONS

Some security and privacy considerations about the proposed identity model are given in this subsection. First, we have already addressed the privacy of the exchanged device attributes in Section IV. These can be protected by devices by generating zero-knowledge proofs when compiling verifiable presentations, before sending them to the marketplace.

Another aspect to be considered is related to data aggregation. Data aggregation occurs when consumers aggregate information from the same device when asking for different verifiable presentations. This is a difficult privacy problem to address, but ZKPs can solve this aggregation problem by hiding unnecessary information. In addition, solutions to this problem are often policy driven, so if a device wants to avoid aggregation in his data, he can indicate it in the verifiable presentation he generates.

Furthermore, other security risks must be considered, even if ZKPs are used. Different side-channel attacks can be used to retrieve useful information about the subject of verifiable credentials. For example, there are mechanisms on the Internet to track individuals, such as cookies, web browser fingerprinting or position information. The proposed scheme cannot prevent the use of these tracking technologies if they have been installed in the consumers or devices. Hence, their use must be avoided in critical scenarios.

Another risk could appear when certification entities include links in verifiable credentials. A verifiable credential is tampering-protected, but the content outside this credential is not, so links can cause harm as the linked data can be modified by an attacker. This can be avoided by using content-integrity protection for links to external data, such as storing the content in IPFS [60]. The W3C standard [47] describes about how to protect against these and other attacks in Sections 7 and 8.

The protection of cryptographic keys, which is a cornerstone of signature processes, is also crucial to the identity management process. A trusted execution environment (TEE) [61] is commonly known as an isolated processing environment in which applications can run, separated from the rest of the system and, most relevant to the present work, can be used to protect the signing process as well as the data (cryptographic keys) involved in it. The signing process can also be protected by employing embedded cryptographic keys using a trusted platform module (TPM) for embedded systems [62].

### VI. PERFORMANCE VALIDATION

The proposed system can be implemented as long as it complies with W3C standards for DIDs and VCs ([55] and [47], respectively). However, some questions arise regarding the performance of the devices when the proposed protocol is implemented. Although, as stated in Section II, some authors [25] have suggested that most devices are able to work with DIDs in terms of performance, we conducted some tests to determine whether it is also possible for devices to follow the proposed protocol and, specifically, to create and validate consumer credentials. IoT devices frequently depend on batteries, and it is important to check how SSI processes consume them. As providing an accurate measure of battery usage is dependent on the target device, we analyzed CPU and RAM usage during the consumer credential creation and verification processes to evaluate the number of consumed resources, which provides more general conclusions. Regarding the marketplace, its implementation is expected in non-resource-constrained nodes that can be scaled with more resources or even with more nodes. Thus, we consider that, from a performance point of view, there are no specific issues. In addition, there are several possibilities to implement this component, so it is not possible to measure its performance as it depends on the final implementation. The most basic implementation could be a database and a web service. SQL databases do not seem to be very appropriate for implementing this component as it needs to store verifiable presentations, which are unstructured. Thus, a non-relational database, such as MongoDB seems to be more appropriate [63]. Regarding the webservice, Apache [64] is the most widely used web server in the world, but other alternatives are also possible.
MongoDB benchmark results are promising even with enormous amounts of data, performing even better than MySQL [65]. Traffic is normally managed through load balancers, such as Nginx [66], which allow to distribute the requests between different servers, then scaling the solution. Nginx can also substitute Apache as a web server. Network latency is another aspect to be considered when it comes to the final implementation, but it is always present and is not an SSI-related issue. However, it is true that the performed tests directly depend on network latency, so a `ping` command has been used to determine the base latency of the test network. To establish a baseline, each ICMP package in the test network takes an average of 0.28ms Round Time Trip (RTT), which is insignificant compared to the obtained times. Finally, regarding consumers, the VC creation and verification processes are equal to the processes in the devices. In addition, consumers can run SSI clients in a variety of hardware (laptops, smartphones, servers, etc). Hence, the results have not been included in the present work because they fully depend on the chosen hardware.

To conduct the tests, a Raspberry Pi 2 B was used as the device. It is equipped with 1GB of RAM memory and uses a 900 MHz quad-core ARM Cortex-A7 CPU. A low-resource Ubuntu [67] operating system with a Hyperledger Indy [33] client was installed in the device. Indy clients are currently under the Hyperledger Aries project, and the library used for testing has been the Aries Framework JavaScript project, available in Github [68].

A wallet created in Raspberry Pi stores the consumer credentials. Aries agents are the software used to connect different SSI actors. Another computer located in the same network plays the consumer role by first receiving a consumer credential and then sending a VP with this credential to the Raspberry Pi, which must validate it. In this setup, we first measure the time spent by both the consumer credential exchange and the consumer credential verification, both within the service consumption subprocess seen in Section V. Aries implements a several-step mechanism between consumer and device, similar to a TCP handshake, to complete these exchange and verification processes. We take 5,000 samples of every process and take the mean and standard deviation values. In addition, we created 10,000 consumer credentials in the device to evaluate the time spent on this process. Besides the time evaluation, RAM and CPU usage are measured in the device for both processes as well. The results are shown in Table I.

CPU percentages were averaged among the four available logic cores in the Raspberry Pi. Tests were conducted using JEST [69] scripts, each of which took approximately 50 seconds to complete. Consequently, CPU and RAM usage were measured during these 50 s. To establish a baseline, launching a JEST script that only loads the Aries libraries takes approximately 12 s and consumes 6MB of RAM memory and 4% of the CPU.

The following conclusions can be drawn from these results. First, higher times are obtained in the processes for exchanging and verifying consumer credentials, rather than in creating them, through a VP. Second, regarding RAM and CPU usage, it is not a resource-intensive task, so most devices should be able to perform these operations. Consequently, we can conclude that network latency will be the main factor that affects performance, although a slower CPU in the target device can degrade it. However, according to the experiment, the minimum required RAM was approximately 150MB. Tests conducted with several consumer credentials exchanges in parallel do not seem to have a significant impact on CPU and RAM usage. Finally, we can conclude that the consumer credential exchange process is slightly faster than verification. The reason for this might be that the device needs to verify the validity of the VP, which adds extra complexity. JavaScript is a quite efficient language thanks to its asynchrony and allows the Raspberry Pi to achieve reasonable times because devices can perform other tasks while waiting for the next step in the consumer credential exchange and verification processes. Hence, JavaScript is appropriate for the implementation of Aries clients for IoTaaS, avoiding the blocking of the devices between these steps.

It is important to note that these numbers can vary depending on factors such as network latency, intermediate network hops, and device capabilities. However, the results presented here are promising and show that most devices will be able to run the scheme without any issues.

### VII. CONCLUSIONS

This paper has analyzed the IoTaaS business model from a formal perspective, together with its associated technological challenges, and proposed an IdM based on self-sovereign identity (SSI). This system includes a marketplace as a mechanism to connect consumers with devices and store verifiable presentations. IoTaaS is a promising business model that will enable the reutilization of IoT devices for different use cases and applications. However, it will be necessary to propose an implementation for the different challenges exposed in this document, as it has been done with the identity problem. The performance results also show that it is possible to implement SSI-based IdMs for IoTaaS. Finally, once the fundamentals of SSI are on the table and their synergies with the IoT are clear, some domain-specific research works (IoTaaS or other) merging these two topics will appear and adapting the general SSI principles to these

### TABLE I. EXPERIMENTAL RESULTS

| Measure                        | Value    |
|--------------------------------|----------|
| **VCs creation**               |          |
| Average                        | 31.07 ms |
| Standard deviation             | 4.80 ms  |
| Consumer Credential exchange   |          |
| Average                        | 811.95 ms|
| Standard deviation             | 73.69 ms |
| Average CPU usage              | 18%      |
| Average RAM usage              | 150 MB   |
| Consumer Credential verification|         |
| Average                        | 923.92 ms|
| Standard deviation             | 44.47 ms |
| Average CPU usage              | 19%      |
| Average Ram usage              | 150 MB   |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
particular use cases will not be straightforward. In IoTaaS, the marketplace has shown this, acting as a storage for **verifiable presentations**. Finally, further research in this field includes the integration of SSI into other components, such as micropayment systems and access control engines.

REFERENCES

[1] Au-Yong-Oliveira, M., Marinheiro, M., & Tavares, J. A. C. (2020, April). The Power of Digitalization: The Netflix Story. In World Conference on Information Systems and Technologies (pp. 590-599). Springer, Cham.

[2] Varia, J., & Mathew, S. (2014). Overview of amazon web services. Amazon Web Services, 105.

[3] Review42. Internet of Things Statistics, Facts & Predictions. [Online]. Available: https://review42.com/internet-of-things-stats.

[4] GREU, V. (2015). Facing IoT-The New Giant Wave of the Information and Communications Technologies Development. Romanian Distribution Committee Magazine, 6(4), 18-25.

[5] Travers, J. IoT as a Service: A new business model?. [Online]. Available: https://www.ericsson.com/en/blog/2018/11/iot-as-a-service-a-new-business-model. Accessed on: Jan 1, 2021

[6] Deng, D. J., Pang, A. C., & Hanzo, L. (2019). Recent Advances in IoT as a Service (IoTaaS 2017). Mobile Networks and Applications, 24(3), 721-723.

[7] Brańczak, A. A., Pacifici, F., & Mazzola, F. (2019). IoT as a service for smart cities and nations. IEEE Internet of Things Magazine, 2(1), 28-31.

[8] Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82, 395-411.

[9] Zhou, W., Jia, Y., Peng, A., Zhang, Y., & Liu, P. (2018). The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. IEEE Internet of Things Journal, 6(2), 1606-1616.

[10] Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. IEEE Security & Privacy, 9(3), 49-51.

[11] Kambourakis, G., Koliadis, C., & Stavrou, A. (2013, October). IoT as a service for eHealth. In 2013 IEEE 15th International Conference on Internet of Things (pp. 1-6). IEEE.

[12] Kortesniemi, Y., Lagutin, D., Eio, T., & Fotiou, N. (2019). Improving the privacy of IoT with decentralised identifiers (dids). Journal of Computer Networks and Communications, 2019.

[13] Berzin, O., Ansay, R., Kempf, J., Sheikh, I., & Hendel, D. (2021, Mar). The IoT Exchange. [Online]. Available: https://arxiv.org/abs/2103.12131. Accessed on: Jun 1, 2021.

[14] Weingartner, T., & Camenzind, O. (2021). Identity of Things: Applying concepts from Self Sovereign Identity to IoT Devices. The Journal Of The British Blockchain Association, 21244.

[15] Naik, N., & Jenkins, P. (2020, December). uPort open source identity and decentralized trust. Whitepaper, The Sovrin Foundation.

[16] Windley, P., & Reed, D. (2018). Sovrin: A protocol and token for self-sovereign identity and decentralized trust. Whitepaper, The Sovrin Foundation.

[17] Abishek, S. V., Priyan, G. D., More, S., & Suganthan, A. IoT based Smart Band for Biometric Authentication Using Blockchain Technology.

[18] Theodoulis, A., Moschou, K., Votis, K., Tzovaras, D., Lauinger, J., & Steinhorst, S. (2020, June). Towards a Blockchain-based Identity and Trust Management Framework for Internet of Things. In 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-6). IEEE.

[19] Windley, P., & Reed, D. (2018). Sovrin: A protocol and token for self-sovereign identity and decentralized trust. Whitepaper, The Sovrin Foundation.

[20] Ali, M., Shea, R., Nelson, J., & Freedman, M. J. (2017). Blockstack Identity for IoT Devices. In 2017 IEEE International Conference on Blockchain (pp. 1-6). IEEE.

[21] Veres One - A Globally Interoperable Blockchain for Identity. [Online]. Available: https://veres.one/. Accessed on: Jan 1, 2021

[22] Jolocom - A Decentralized, Open Source Solution for Digital Identity and Access Management. [Online]. Available: https://jolocom.io/wp-content/uploads/2019/12/Jolocom-Whitepaper-v2.1-A-Decentralized-Open-Source-Solution-for-Digital-Identity-and-Access-Management.pdf. Accessed on: Jan 1, 2021

[23] Identity.com - Decentralized Identity Verification. [Online]. Available: https://www.identity.com/. Accessed on: Jan 1, 2021

[24] Uniquid - Connect and Control IoT Devices at Scale. [Online]. Available: https://uniquid.com/. Accessed on: Jan 1, 2021

[25] ShoCard – Personal identity for the people. [Online]. Available: https://www.pingidentity.com/en/lp/shocard-personal-identity.html. Accessed on: Jan 1, 2021

[26] DIF – Decentralized Identity Foundation. [Online]. Available: https://identity.foundation/. Accessed on: Jan 1, 2021
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131012, IEEE Access

[42] S. Foundation. (2020). Self-Sovereign Identity and IoT.[Online]. Available: https://sovrin.org/wp-content/uploads/SSI-and-IoT-whitepaper.pdf. Accessed on: Jan 1, 2021

[43] Zheng, W. (2006). The business models of e-marketplace. Communications of the IIMA, 6(4), 1.

[44] Lu, R. (2018). A new communication-efficient privacy-preserving range query scheme in fog-enhanced IoT. IEEE Internet of Things Journal, 6(2), 2497-2505.

[45] Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January). MQTT-S-A publish/subscribe protocol for Wireless Sensor Networks. In 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE’08) (pp. 791-798). IEEE.

[46] Singh, M., Rajan, M. A., Shivraj, V. L., & Balamuralidhar, P. (2015, April). Secure mqtt for internet of things (iot). In 2015 fifth international conference on communication systems and network technologies (pp. 746-751). IEEE.

[47] World Wide Web Consortium. (2019). Verifiable Credentials Data Model 1.0: Expressing Verifiable Information on the Web. https://www.w3.org/TR/vc-data-model/?c=vc-data-model.

[48] Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. Manubot.

[49] Buterin, V. (2017). A next-generation smart contract and decentralized application platform. white paper, 3(37).

[50] Birjuvoked, A., & Tikhonirov, S. (2019, June). DeHomologization and linkability of cryptocurrency transactions based on network analysis. In 2019 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 172-184). IEEE.

[51] Lundqvist, T., De Blanche, A., & Andersson, H. R. H. (2017, June). Thing-to-thing electricity micro payments using blockchain technology. In 2017 Global Internet of Things Summit (GloTS) (pp. 1-6). IEEE.

[52] Kurt, A., Mercana, S., Erdin, E., & Akkaya, K. (2021, May). Enabling micro-payments on IoT devices using bitcoin lightning network. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-3). IEEE.

[53] Samaniego, M., Jamsrandorj, U., & Deters, R. (2016, December). Blockchain as a Service for IoT. In 2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing ( CPSCom) and IEEE smart data (SmartData) (pp. 433-436). IEEE.

[54] Nakada, R., Nguyen, K., & Sekiya, H. (2021). Implementation of Micropayment System Using IoT Devices. Journal of Signal Processing, 25(4), 137-140.

[55] Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., & Holt, J. (2020). Decentralized identifiers (dids) v1. 0. Draft Community Group Report.

[56] Kulabukhova, N. V. (2019, January). Zero-knowledge proof in self-sovereign identity. In Proceedings of the 27th International Symposium Nuclear Electronics and Computing (NEC’2019) (pp. 381-385).

[57] Gupta, P., Kanhere, S., & Jurda, R. (2019). A decentralized IoT data marketplace. arXiv preprint arXiv:1906.01799.

[58] Voigt, P., & Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10, 3152676.

[59] Sermersheim, J. (2006). Lightweight directory access protocol (LDAP): The protocol.

[60] Benet, J. (2014). Ibs-content-addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561.

[61] Sabt, M., Achemlal, M., & Bouabdallah, A. (2015, August). Trusted execution environment: what it is, and what it is not. In 2015 IEEE Trustcom/BigDataSE/ISPA (Vol. 1, pp. 57-64). IEEE.

[62] Kinney, S. L. (2006). Trusted platform module basics: using TPM in embedded systems. Elsevier.

[63] Győrődi, C., Győrődi, R., Pecherle, G., & Olah, A. (2015, June). A comparative study: MongoDB vs. MySQL. In 2015 13th International Conference on Engineering of Modern Electric Systems (EMES) (pp. 1-6). IEEE.

[64] Laurie, B., & Laurie, P. (2003). Apache: The definitive guide. “O’Reilly Media, Inc.”

[65] Jose, B., & Abraham, S. (2020). Performance analysis of NoSQL and relational databases with MongoDB and MySQL. Materials today: PROCEEDINGS, 24, 2036-2043.

[66] Nedelcu, C. (2013). Nginx HTTP Server. Packt Publishing.

[67] Medina, F. A. B., Menjívar, T. G., & Flores, E. O. C. (2019). Lubuntu. SolTvIC UGB, 3(2), 21-30.

[68] Aries Framework Javascript. [Online]. Available: https://github.com/hyperledger/javascript. Accessed on: Jul 1, 2021

[69] Delightful JavaScript Testing. [Online]. Available: https://jestjs.io/. Accessed on: Jul 1, 2021

Santiago de Diego Mathematician and IT Engineer from the University of Granada (UGR) and the master’s degree in information security at the International University of La Rioja (UNIR). He currently works as a cybersecurity researcher at TECNALIA Research & Innovation and is pursuing a Ph.D. degree at the University of Granada under the supervision of Dr. Gabriel Macià and Dra. Cristina Regueiro.

His current research interests include distributed ledger technologies, cybersecurity for critical infrastructures, and identity management systems.

Gabriel Macià is an Associate Professor in the Department of Signal Theory, Telematics and Communications of the University of Granada (Spain) and researcher in the Information and Communication Technologies Research Centre, CITIC. He received an M.S. in Telecommunications Engineering from the University of Seville, Spain and the Ph.D. in Telecommunications Engineering from the University of Granada. His research interests are focused on system and network security, with special focus on intrusion detection, ethical hacking, network information leakage and denial of service.

Cristina Regueiro Telecommunications Engineer (2010) with a PhD in Information Technology and Communications in Mobile Networks (2017) from the University of the Basque Country. She started as a Researcher at the University of the Basque Country on issues related to digital signal processing, wireless communications, and mobile communications (2011-2017). She continued her research work at the Innova lab Association (2017-2018) with a deep focus on industrial communications, Industry 4.0, IoT, fog, and cloud computing. Her cybersecurity experience was later completed at Ikerlan (2018-2019) taking part in the cybersecurity IoT and cybersecurity on digital platforms teams. In December 2019, she joined the cybersecurity and blockchain team within Tecnalia, where she is currently a senior researcher with a focus on critical infrastructures and blockchain-based systems.

VOLUME XX, 2017

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/