IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 13, 2021, accepted July 21, 2021, date of publication July 26, 2021, date of current version August 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099837

A Trusted Approach for Decentralised and Privacy-Preserving Identity Management

RAFAEL TORRES MORENO, JESÚS GARCÍA-RODRÍGUEZ, JORGE BERNAL BERNABÉ, AND ANTONIO SKARMETA, (Member, IEEE)
Department of Information and Communication Engineering, University of Murcia, 30100 Murcia, Spain

Corresponding author: Rafael Torres Moreno (rtorres@um.es)

This work was supported by the European Union’s Horizon 2020 Research and Innovation Program (OLYMPUS Project) under Agreement 786725.

ABSTRACT Identity Management (IdM) systems have traditionally relied on a centralized model prone to privacy, trust, and security problems, like potential massive data breaches or identity spoofing. Identity providers accumulate excessive power that might allow them to become a big brother, analyzing and storing as much data as possible. Users should be able to trust identity providers and manage their personal information straightforwardly without compromising their privacy. The European OLYMPUS project introduces a distributed approach for IdM based on enhanced Attribute-Based Credentials (ABC) that splits the role of Identity Provider to limit their influence and chances to become a unique point of failure. However, the trust relationship between service providers, users, and identity providers is still a gap in those kinds of privacy-preserving ABC systems. Decentralized technologies are an opportunity to break away from the centralized model and propose systems that respect privacy while increasing users’ trust. This paper presents an evolution of the OLYMPUS architecture, maintaining all the privacy features and incorporating distributed ledger technologies to enhance trust and security in online transactions and IdM systems. The proposed system has been implemented, tested, and validated, showing its performance and feasibility to manage user’s identity in a fully privacy-preserving, distributed and reliable way.

INDEX TERMS Blockchain, digital identities, DLT, identity management, privacy, privacy enhancing technologies, privacy-preserving, security.

I. INTRODUCTION

Data has become a big concern. Smart cities, eHealth, Industry 4.0, and many cloud applications are putting traditional identity management systems into trouble that are not evolving with the same speed. Systematic analysis through algorithms, the reduction of storage costs, and the lack of user-friendly tools that allow users to improve their privacy pose a problem for consumers. Data, such as location or health monitoring, allow valuable information to be collected for companies, many times without users being aware of this collection.

Traditional identity management systems (IdMs) rely on the use of centralized identity providers (IdPs) that create, manage and maintain identity information of its users or smart devices and, at the same time, provide authentication mechanisms to service providers (SPs). This widely deployed solution enables the operation of single sign-on (SSO) technologies. Although the way it operates is very convenient due to its simplicity, achieving desirable levels of security and privacy is a challenge. Tracking and linking by IdPs is one of the main problems to be faced. For example, in applications dealing with sensitive data (e.g., health), the loss of privacy can be a major problem.

In this scenario, users must be careful and must have the necessary information about when, how and with whom they are sharing their personal information in order to be able to avoid massive data leakage [1] or collection without consent [2]. Furthermore, users should have at their disposal the necessary tools to enable them to exercise the rights described by the European Union in the General Data Protection Law (GDPR) [3], [4].

Identity management systems are evolving towards decentralised systems in order to address current shortcomings.

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

The metamorphosis of Internet services and the quest for greater security have brought distributed ledger technologies(DLT) [5] to the forefront. Among these technologies, the most famous is Blockchain [6], which is nothing more than a decentralized database organized by cryptographically linked blocks. It makes it not possible to alter the information once it has been introduced. Although the popularity of these technologies is due to the irruption of Bitcoin [7] and other cryptocurrencies, their application has been demonstrated in other cases such as border control, e-voting, e-residence, authorship management, supply chains, etc. In all these cases, the digital identity and privacy-preserving concepts are gaining strength and relevance, without forgetting that *what you write in the chain stays in the chain*. In any case, DLT scenarios still have to address numerous challenges [8] regarding linkability, network privacy, key management, or privacy regulations

The progression from web-centric approaches or federated identities to a self-sovereign system (SSI) [9] where users take control over their data to avoid constant tracking, IdPs impersonations, or massive data leakages is a trending topic [10]. Combining privacy-preserving IdM systems with DLT or Blockchain technologies can provide a sufficiently robust and user-friendly solution that would maintain security standards while improving confidence in the entire infrastructure and reducing the chances of fraud.

Although numerous solutions for identity management have been proposed, there is room for improvement. While widely deployed IdMs are weak on privacy, new solutions are cumbersome, unusable, and still leave trust between the entities involved unresolved. Users often find that their ability to manage their identity is lowered. Identity providers analyze as much information as possible for their commercial purposes. Even service providers act dishonestly by asking for more data than necessary or modifying their access policies without explanations to jeopardize user protection.

This paper presents a solution relying on distributed technologies and privacy-preserving Attribute-Based Credentials (P-ABC) [11], [12]. The main challenge is to reinforce the trust of users and service providers in the distributed entities that conforms the IdM system. The proposal is based on the European OLYMPUS project [13], eliminating the IdP as a critical point of failure, avoiding the tracking of users through their behavior, and erasing the risk of impersonation. It then modifies the OLYMPUS behavior to integrate a reinforced trust system. The inclusion of blockchain technology significantly enhances the trust features of the architecture. First, it increases users' trust in IdPs thanks to ledger-backed registration. Second, trust between users and service providers is also improved through the registration of the service providers. Finally, it makes it easier for service providers and identity providers to have a strong trust relationship from the start. All this while keeping usability and integration features as straightforward as possible.

The rest of this paper is structured as follows. Section II introduces the state of the art about identity management. Section III introduces the OLYMPUS proposal and describes the Blockchain distributed private attribute based credential (dP-ABC) approach introducing goals and proposed architecture. Section IV delves into proposal implementation details and evaluation. Section V provides a security analysis of the proposed solution. Finally, Section VI closes the paper with the main findings, conclusions, and future work.

II. STATE OF THE ART: IDENTITY MANAGEMENT

User authentication is a critical aspect nowadays. Users have multiple accounts with different service providers, and each of these services may require a specific set of data. For example, an airline will require user citizenship and passport number, while other services will accept any user data. Personal data is stored and protected by some authentication mechanism that allows users to prove they are who they claim to be. How this information is protected becomes a critical point. The typical username and password pair has been proven insufficient [1], [2], [4].

On the one hand, in terms of linkability, storing the same attributes in different providers allows these providers to track users across the services they use. Simply by relating or comparing attributes, a user can be identified unambiguously.

On the other hand, although legislation such as the GDPR [3] exists, not all service providers guarantee data security, either for ulterior motives or simple negligence, adding to the risk of sharing data with these providers, with database leaks facilitating attackers in their task and increasing the problem of reused passwords.

Beyond the typical username and password, other solutions try to increase privacy and security, such as public key infrastructures (PKI) [14], single sign-on (SSO) [15], or privacy-enhancing attribute-based credentials (P-ABC) [16]–[18] that we briefly introduce below.

Public Key Infrastructure (PKI) [14] and the use of X.509 certificates is one of the best-known and easy-to-implement solutions. Its operation is based on the trust of certification authorities (CAs) and issuing certificates endorsed by trust chains through these CAs. One of the advantages of this scheme is that the CAs only have to be available to obtain the certificate.

In contrast, service providers must agree to use and rely on intermediate CAs that users can use. However, the major concern is that users are responsible for managing their private keys. If this material is lost, they will not be able to log in and will have to create a new account. Even worse, an adversary could claim to have lost his private key to impersonate a legitimate user. In this sense, requests to create or close accounts are critical. In addition to these problems, it also has usability shortcomings. Since most users have more than one device, distributing cryptographic material becomes a tedious task. In addition, linkability remains a problem unless the user has a different pseudonym and certificate for each service provider. Finally, the X.509 is an all-or-nothing system and lacks the concept of minimal disclosure. This means that the user will always reveal the full content of his credential, including relevant and non-relevant data, during an authentication process. From a privacy point of view, this is a big problem.

Federated identity systems [15] is another well-known and widely implemented technology. They are based on protocols and standards, such as OAuth [19] or SAML [20]. These systems centralize trust in a single entity, the identity provider (IdP). This entity is responsible for storing all the attributes associated with a user. Users only have to register with an identity provider, and service providers must trust the identity provider's assertions about a user's attributes.

Trust centralization is an advantage over X.509. However, user attributes can still be stored in SPs and exposed in case of a security breach. The user no longer has to take care of cryptographic material; he only needs his username and password, making this option easy to use even on new devices. Revocation problems are no longer suffered, thanks to the generally short lifetime of the tokens. Only the account linked in the IdP needs to provide the appropriate revocation or closing mechanisms. In addition, this approach eliminates the repetitive task of filling the same data on each SP because all attributes are stored in the IdP.

Nevertheless, the main benefit of SSO over X.509 is applying a high level of granularity to user attributes. For example, if a service wants to verify that a user is older than a certain age, it is no longer necessary to disclose the complete date of birth. Asserting that the user is older than a certain age is enough. However, this mechanism is not standardized in all SSOs.

The biggest problem with this approach is precisely the centralization of trust in a single IdP, making it a critical failure point. In case of a compromised IdP, all private information will be affected. Furthermore, linkability is still a problem and is accentuated with traceability because every interaction the user makes inevitably goes through the IdP, which will learn everything about its users.

Traditional IdM systems share a lack of solutions concerning user privacy, often leaving too much data exposed, as in the case of X.509, or allowing an entity to act as a big brother tracking all user movements in the case of SSO systems. In this circumstance, Privacy-Enhancing Attribute-Based Credentials (P-ABCs) [16]–[18] are presented as a privacy-preserving solution. P-ABCs make a similar proposition to X.509, where a user receives a credential issued by a trusted issuer. The credential contains a set of attributes certified by the issuer for e.g., the user's name, age or nationality that can be used to convince a service provider of the validity of the declared attributes. Unlike X.509, P-ABCs enable the user to derive one-time use tokens that reveal strictly necessary information, for e.g., certifying that the user is older than a certain age.

In this scenario, users obtain from the service provider (SP) an access policy (P) with which they have to comply (i.e., older than certain age). If the user credential can satisfy the policy, a so-called presentation token is obtained. This token contains only the minimum amount of information required by the access policy, potentially being a predicate over an attribute. For example, proving that the user's age is higher than the required limit. An important aspect is that the user can only obtain presentation tokens consistent with the information certified in the credential, and the service provider can verify the token against the policy and be convinced of its correctness.

The benefits provided by P-ABCs are mostly related to the privacy they bring to the user. Apart from gaining selective disclosure, users have more control over their linkability because of the non-binding properties of the derived tokens. It is not possible to know which credential they come from once they are generated. However, P-ABCs suffer from usability problems. Implementations such as IBM's Identity Mixer [21], [22] or Microsoft's U-Prove [23] have not had the expected adoption rate. They also inherit the same management issues as X.509 since users are responsible for the security of their credentials. Compromising these credentials or their keys could allow an attacker to impersonate the user. Further, P-ABC systems are cryptographically complex. Users and service providers must have specific software to work with them, and they tend to be computationally expensive. Finally, previous P-ABCs still rely on a single identity provider; this exposes the IdP as a single point of failure.

Distributed technologies are emerging strongly, and applications such as Blockchain are taking identity systems to a new level where privacy and security are the challenges to address [8]. Proposals in the context of the Blockchain are growing in number, driven by the rise of cryptocurrencies. Hawk [24], Zcash [25] or Zerocoin [26] are cryptocurrencies that already add privacy features such as zero-knowledge proofs or linkability controls. Privacy-preserving solutions based on crypto-privacy techniques are emerging to empower users with mechanisms to become anonymous and take control of their data following a Self-Sovereign Identity (SSI) model. In that sense, solutions such as Sovrin [27], Serto (previously uPort) [28], Jolocom [29] and Shocard [30] are some of the foremost proposals.

Sovrin [27] is an identity management solution that runs on top of permissioned blockchain [8], in particular, Hyperledger Indy [31]. Sovrin supports DPKI (Decentralized Public Key Infrastructure), where every public key has its public address in the ledger (DID, decentralized identifier [32]) that enable universal verification of claims. Users can have different DIDs for each existing relationship, with different key pairs. Sovrin allows attestation, verifiable assertions, and anonymous credentials based on zero-knowledge proofs, with the scheme proposed by Camenisch-Lysyanskaya [21]. The Sovrin approach is very comprehensive, and its advantages, such as unlinkability, identity recovery, integration of DIDs, or zero-knowledge proofs, are well integrated. However, Sovrin does not provide an authentication service and lacks usability by not displaying clear and precise information on the privacy implications that may arise. Moreover, it does not support smart contracts, which is an explicit limitation of the scenario. As for the credentials used, the underlying cryptography is old, negatively impacting its efficiency.

Serto (formerly uPort) [28] is another identity solution that works on permissioned blockchains. It uses a 20-byte hexadecimal identifier to represent the user's ID, with the address of a Proxy Smart contract deployed over the Ethereum [33] network. The smart contract is used as an indirection method between the user's private key (hosted on their device) and the accessed service. The user's application contacts a smart contract that contains the access control logic.

This system provides some unlinkability by the possibility of having different user IDs. In addition, it adds selective disclosure with the possibility of attribute encryption. Finally, it additionally supports identity recovery in loss and integrates with the decentralized identifier (DID).

Serto's proposal lacks precise information on privacy implications. It also does not provide authentication, and how attributes are stored can be problematic. Stored attributes, even in encrypted form, are always publicly accessible and therefore analyzable.

Jolocom [29] is an identity framework that by default stores DIDs on the public Ethereum blockchain. Jolocom's approach aims to provide a lightweight, self-sovereign identity solution for decentralized systems that is easy to implement for non-technical users. In a nutshell, Jolocom provides decentralized identity based on hierarchically deterministic keys (HD keys) generated, supplied, and controlled by users. It supports key recovery through the use of a seed phrase and provides anonymity in context-specific interactions. Interoperability is a goal to be achieved, so it introduces Etherum-based smart contracts and includes support for other blockchains.

The advantages are structural and in terms of interoperability. Distributed operation and support for W3C standards on verifiable credentials and DIDs are very positive. However, it does not introduce any privacy preservation mechanism, nor does it support verifiable presentations (defined by VC-W3C). Moreover, it only allows the use of JWT tokens as an authentication mechanism.

Shocard [30] is an identity management solution built on a public blockchain. Generated ShoCards are stored in the blockchain while keys and other sensitive information are stored out of the band.

The significant advantage of Shocard is that it is very lightweight and easy to implement. On the other hand, aspects such as unlinkability are not fully guaranteed making it possible to track users. In addition, Shocard requires a server, which can potentially be a critical point as it stores information about users.

All the solutions described tend to be incomplete. They do not provide a complete ecosystem. While traditional identity systems are weak on privacy, blockchain proposals lack sufficient tools (authentication, authorization) to complete identity management systems. We need an ecosystem that brings together satisfactory privacy management while being easy to use and integrate by all parties involved, providing greater trust.

III. BLOCKCHAIN-BASED dP-ABC APPROACH

The previous section shows the current state of the art concerning traditional IdM systems and introduces some disruptive Blockchain-based solutions. Now we introduce a novel proposal based on the OLYMPUS identity management framework [13], [34] in combination with DLT technologies.

A. REFERENCE dP-ABC ARCHITECTURE

The European project OLYMPUS [13], [34], introduces a new proposal for identity management applying distributed techniques, intending to solve the problems presented in previous solutions. This approach devises a privacy-preserving identity management solution evolving from federated identity systems and eliminating the IdP as the single point of failure. OLYMPUS introduces the concept of oblivious identity management [35]. This approach distributes the capacity of the traditional IDP over multiple partial IDPs. No single server, or no collusion of servers smaller than a given threshold, can impersonate its users, track their online behavior, link their virtual identities across services, or recover their passwords. Figure 1 shows the OLYMPUS proposed architecture.

The proposal made by OLYMPUS identifies several requirements to be addressed: No Impersonation by IdPs, avoid offline attacks, short-lived authentication tokens, unlinkability across Relying Parties (RPs), hide RPs from IdP, minimize user-side hardware/software, data-minimization, and Easy integration with existing IdM technologies.

The OLYMPUS approach focuses on usability proposing a framework that imposes minimal requirements on user devices and does not rely on secure hardware tokens. Nevertheless, it respects users' right to privacy by enforcing unlinkability of authentications and minimal data disclosure concerning service providers and identity providers alike. In addition, to facilitate adoption, OLYMPUS works closely with existing technologies and standards, such as the introduction of the Verifiable Credentials proposed by the W3C [36].

The advanced cryptographic techniques used by OLYMPUS allow moving from a centralized to a distributed IdP model. Work is distributed among N IdPs that collectively work as a Virtual Identity Provider (vIdP). Only by compromising all IdPs would it be possible to compromise the integrity of the system. OLYMPUS provides distributed user and password-based authentication to operate (1) online or (2) offline. (1) During an online case, OLYMPUS behaves like a traditional SSO system in which the user obtains a single-use access token for a specific service given an access policy. (2) In the offline scenario, the behavior becomes a P-ABC system where the user can obtain a credential to derive access tokens later without using the identity provider again.

A distributed signature scheme introduced in PESTO [37] is used in the token-based scenario, where each of the

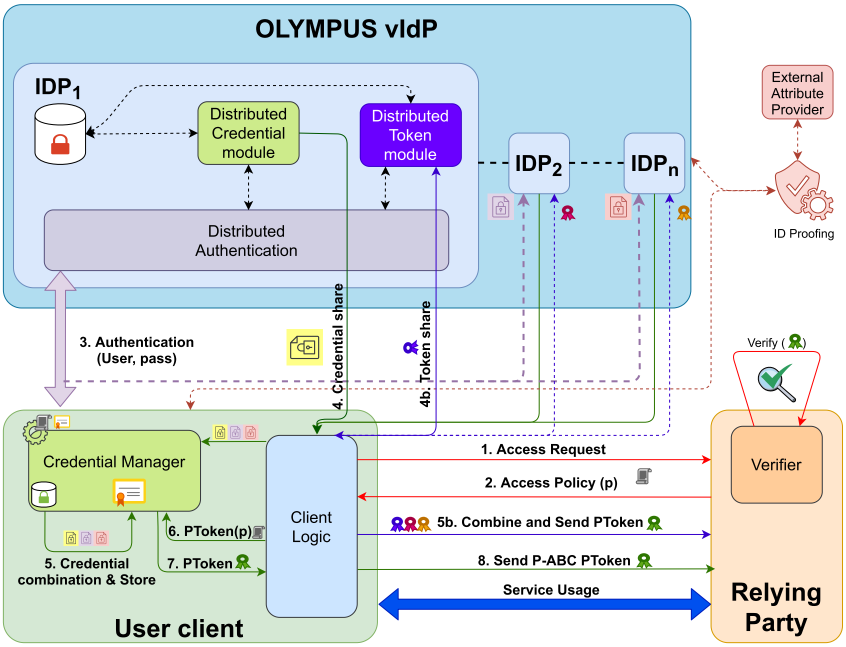


FIGURE 1. OLYMPUS dP-ABC architecture.

IdPs generates a signature share. By combining all the shares, the presentation token is generated. For the scenario based on dP-ABC credentials, a multi-signature scheme introduced in [38] based on Pointcheval-Sanders signatures is used. Each partial IdP signs the user attributes (with its independently-generated key), generating a credential share whose functionality is equivalent to a credential issued by a single IdP. The user combines the shares into a final verifiable credential with the aggregated public key of the vIdP. This public key is the aggregation of the different public keys of each partial IdP that mold the vIdP. A detailed explanation of the dP-ABC implementation and associated crypto-algorithms can be found in [39] and [40] respectively.

OLYMPUS has multiple benefits thanks to the combination of usability and privacy features that defined SSO and X.509. It offers two possible forms of operation under the same framework depending on the needs of the scenario or the user. The framework introduces critical features and capabilities needed for truly privacy-preserving identity management solutions. It includes unlikability between service and identity providers, protection against impersonation, and minimal disclosure capabilities. However, the significant advantage of the OLYMPUS scheme is the introduction of decentralized technologies eliminating the critical point of failure introduced in SSO systems.

Despite its advantages over traditional systems, the architecture can be improved in terms of trust. Although OLYMPUS introduces distributed technologies through the segmentation of the IdP and even in the issuance of cryptographic material, it does not address the trust relationships that are still necessary between the entities involved. Trust is put on the composition of the vIdP (partial IdPs) but ignores the necessary trust relationship between users, vIdP, and service providers. They still need to trust, as traditionally done, that everything is legitimate and reliable.

Users want to be sure that their identity provider is trustworthy. Total trust cannot be achieved through a typical relationship where the user trusts the vIdP; something more is needed. Secondly, users value having as much information as possible about the services they are going to use. Whether a service provider operates honestly or whether something looks suspicious, users need to be appropriately informed. It is necessary to provide them with tools that increase their confidence in these situations without diminishing their user experience.

B. PROPOSED EVOLVED BLOCKCHAIN-AWARE TRUSTED dP-ABC ARCHITECTURE

The proposed evolution of OLYMPUS aims to substantially improve confidence in the entire infrastructure without penalizing the user experience and maintaining the precepts of ease of use, deployment, and integration with other technologies.

OLYMPUS offers two modes of operation, (1) online and (2) offline, depending on the desired scenario. After analyzing those scenarios, in both cases, clients must trust the legitimacy of the vIdP (including partial IdPs) and service providers. In addition, they must make service access decisions according to a set of policies with no other help than their common sense. Experience has shown that burdening users with too many decisions often leads to problems and even loss of credentials and personal data. In other words, all parties (including users) trust that everything works as expected. For that reason, and although OLYMPUS is going on the right path, the trust of the infrastructure can

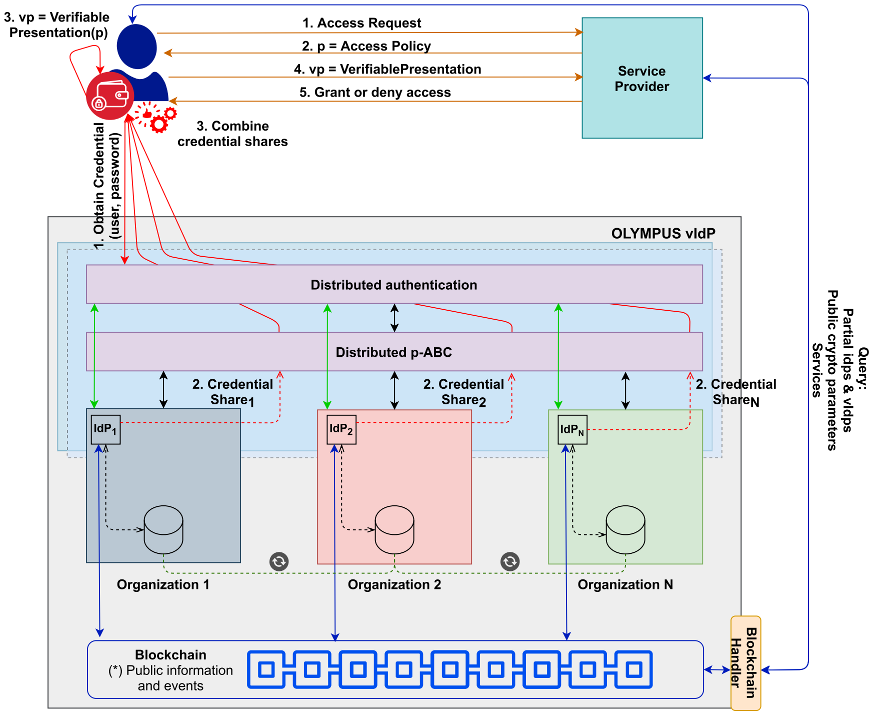

3. vp = Verifiable Presentation(p)
1. Access Request
2. p = Access Policy
4. vp = VerifiablePresentation
5. Grant or deny access
Service Provider
3. Combine credential shares
1. Obtain Credential (user, password)
OLYMPUS vIdP
Distributed authentication
Distributed p-ABC
2. Credential Share$_1$
2. Credential Share$_2$
2. Credential Share$_N$
IdP$_1$
IdP$_2$
IdP$_N$
Organization 1
Organization 2
Organization N
Query: Partial idps & vidps Public crypto parameters Services
Blockchain (*) Public information and events
Blockchain Handler

FIGURE 2. Proposed Blockchain-based and fully distributed privacy-ABC IdM system.

be increased by coupling technologies such as blockchain. In that sense, the evolution of the architecture is presented (Figure 2) including a blockchain infrastructure that allows increasing trust in the whole scenario, including user interactions.

The proposal is based on introducing a ledger to write or record specific events, serving as tamper-proof support to subsequently deploy processes such as identity provider discovery or verification of public cryptographic parameters. Thus, the ledger provides a cryptographic root of trust, which facilitates identity management without external authorities.

The proposed architecture shows an OLYMPUS scenario in which different organizations operate the vIdP. Each organization contributes one or more partial IdPs that end up forming the vIdP. The underlying cryptographic processes have not changed. The issuance of dP-ABC tokens and credentials still works in the same way. However, there is now a ledger that acts as a watchdog and common registry between the organizations. Each organization's partial IdPs are registered when they are launched, ensuring that their characteristics do not change or are modified without consent, adding an extra layer of trust. If the registration does not match the partial IdP, no further checks would be necessary when composing the vIdP. Another innovation of the new architecture is the possibility to query the ledger. Users and service providers can observe the ledger and make decisions based on the observed data. As we already know, the ledger is a tamper-proof database that provides almost total confidence for the data stored there, and therefore, the confidence in the decisions taken based on this data is increased. In addition, the proposed solution makes use of smart contracts [24] (also known as *chaincodes*) to handle data into the ledger. Smart contracts are blockchain stored programs that run when predetermined conditions are met. They can automate workflows by triggering actions when conditions are met. The typical usage is to automate executions over an agreement so every participant can be sure immediately about the outcome without intermediaries. Smart contracts are decentralized, immutable, and transparent, making it possible to provide the architecture with fully auditable and secure processes.

Data storage in the ledger is a critical process: *What goes into the ledger stays in the ledger*. How it is stored can make all the difference in terms of adoption, security, and trust. The proposed solution includes the use of W3C Verifiable Credentials [36] and Decentralised Identifiers [32] standards, and no personal or privacy-compromising data will, under any circumstances, be stored in the ledger.

In line with the introduction of blockchain technology, OLYMPUS entities must modify their behaviour to accommodate new processes.

1) PARTIAL IdPs AND vIdP

IdPs are the most important entities of the architecture. vIdPs are defined as a set of partial IdPs endorsed in the ledger through the use of smart contracts. Whenever a partial IdP starts its operation, it invokes an enrolment contract that records information relevant to its subsequent identification. A set of attributes defines each partial IdP: A DID document

```
{
    "status": "ACTIVE",
    "publicKey": "CnoKeAo6DeVv7T9T[...]",
    "spawnDate": "2021-03-10T10:48:20",
    "did": {
        "id": "did:umu:OL-Partial-IdP:0",
        "context": "https://www.w3.org/ns/did/v1",
        "service": {
            "serviceEndpoint": "10.1.6.6:9080",
            "type": "OL-Partial-IdP"
        }
    }
}
```

LISTING 1. Partial IdP enrolment.

consisting of an identifier (e.g. did:umu:OL-Partial-IdP:0:test1), context and the service definition (address and service type). It also includes information on whether it is active, the spawn date and the associated public key. An enrolment example is shown in the listing 1. In parallel, when the partial IdP is being enrolled, the composition of the vIdP of which it is part is also added or updated automatically, avoiding that any of the partial IdPs acts as some controller. No partial IdP should have more responsibility than another. Automatically manipulating the composition of the vIdP ensures that the architecture is not hierarchical and eliminates the possibility of one partial IdP having more weight than another. The vIdP is a virtual entity constituted by the endpoints, DIDs, and public keys (as well as the aggregated public key) of the partial IdPs that compose it (Listing 2). This process guarantees that any IdP and vIdP participating in the architecture was added by a trusted party, who must have the necessary permissions and cryptographic material to operate with the ledger, leaving a traceable and auditable record.

2) SERVICE PROVIDERS

Service Providers play a verifying role in OLYMPUS. They communicate access policies and verify accesses against that policy. The new approach goes one step further and performs a registration process for them through smart contracts. As soon as a service wants to trade with the new approach, the following information must be entered into the ledger: its endpoint, DID, registration date, status, and a set of predicates that define what data is required for its use, e.g., revealing the email address or proving that the user's age is in a specific range (listing 3). In this way, anyone could know in advance which services are part of the framework and which data they claim to consume. The ledger acts as a watchdog registering this relevant information in an immutable and auditable way. The service registration process is manual and therefore requires the involvement of administrators. In the future, this process can be automated.

```
{

"did":
{
    "@context": "https://www.w3.org/ns/did/v1",
    "id": "did:umu:OL-vIdP:test1",
    "services":
    [
        {
            "endpoint": "10.1.6.6:9080",
            "id": "did:umu:OL-Partial-IdP:0",
            "pk": "CnoKeAo6er2OxSH2lrVv7T9T[...]"
        },
        {
            "endpoint": "10.1.6.6:9081",
            "id": "did:umu:OL-Partial-IdP:1",
            "pk": "U8R21sGxIE9UebXNMISCdWaZ[...]"
        },
        {
            "endpoint": "10.1.6.6:9082",
            "id": "did:umu:OL-Partial-IdP:2",
            "pk": "aYVXzQ2qNYiJdAgBbPHzYAKA[...]"
        }
    ]
},
"docType": "VIdPRegistration",
"idps":
    [
        "did:umu:OL-Partial-IdP:0",
        "did:umu:OL-Partial-IdP:1",
        "did:umu:OL-Partial-IdP:2"
    ],
"schemas":
    [
        "did:umu:OL-PublicParameters:Scheme"
    ],
"spawnDate": "2021-03-10T10:14:44",
"status": "ACTIVE"

}
```

LISTING 2. Virtual IdP enrolment.

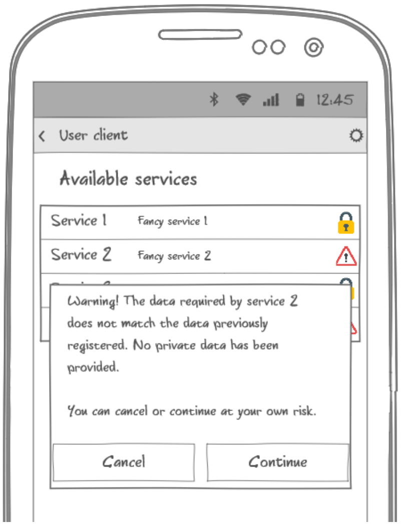
```
{

"date": "2021-03-10T11:28:52",
"did": {
    "@context": "https://www.w3.org/ns/did/v1",
    "id": "service",
    "service": {
        "serviceEndpoint": "https://myservice.com",
        "type": "Web service"
    }
},
"domain": "https://myservice.com",
"predicates": "[{\"attributeName\":\"url:Role\",\"operation\":\"
     REVEAL\",\"value\":null,\"extraValue\":null}]",
"status": "ACTIVE"

}
```

LISTING 3. Service enrolment.

3) USER CLIENT

The user is the main subject to be protected. In OLYMPUS, the user is shielded through the principles of minimal disclose and the issuing of distributed cryptographic material. This approach provides the client with connectivity to the ledger and gives it the capability of discovery. This allows it to find registered vIdPs and IdPs securely before even registering on the platform. Similarly, it can find the legitimate registered service providers along with the data they will require. This puts the customer in an advantageous situation since he can start making decisions without affecting his privacy or security at a glance at the ledger. Firstly, the connection configuration comes from a reliable source, the ledger, making the configuration process trustable for the end-user and even eliminating manual configuration by the user altogether. Secondly, the user receives information certified by the ledger about available services. What kind of service they are and what data they require. With this data, the user can make decisions based on objective data. He knows that his IdP is secure, and he knows that the declared services are being monitored. If a service changes its access policies to, for example, make them more aggressive unilaterally, the user would be warned of this fact.

In any case, no information about users is recorded in the ledger, preventing tracking or any data leakage. Although the user client can send events to the ledger (e.g., warn about a suspicious service), it behaves as an observer. Figure 3 shows a mockup example of what a user will see in his application.



Warning! The data required by service 2 does not match the data previously registered. No private data has been provided.

You can cancel or continue at your own risk.

FIGURE 3. User client mockup view.

C. ARCHITECTURAL IdM FLOWS AND PROTOCOLS

The new proposal slightly modifies the workflows and includes new entities (ledger and blockchain handler), having different interactions between entities. First of all, it is useful to detail the registration process of a vIdP and its partial IdPs (Figure 4).

When an OLYMPUS partial IDP is launched, it starts an internal configuration process in which it must generate the necessary cryptographic material to operate together with the other partial IDPs that will form the vIdP. During this process, they must also be registered in the ledger so that their connection data and public cryptographic parameters are safely stored in an unmodifiable way. To this end, the partial IDPs have been given direct integration with the ledger, using a preset configuration so they can connect and directly invoke smart contracts.

Once the connection between the partial IDP and the ledger is established, it performs the invocation of the *addpartialidp* contract. This contract stores the structure shown in listing 1. The ledger checks the existence of the IdP to be registered or updates it if it exists. In parallel, smart contracts that add or update a partial IdP also invoke the smart contract in charge of adding or updating the vIdP (*addorupdatevidp*), avoiding a partial IdP to become a controller, steps 3 to 6.

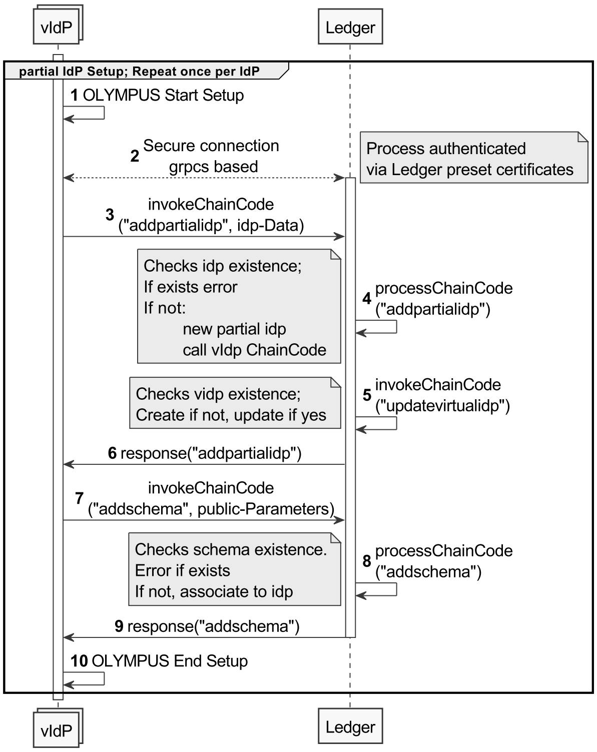

vIdP
Ledger
partial IdP Setup; Repeat once per IdP
1 OLYMPUS Start Setup
2 Secure connection grpcs based
Process authenticated via Ledger preset certificates
3 invokeChainCode ("addpartialidp", idp-Data)
Checks idp existence; If exists error If not: new partial idp call vIdp ChainCode
4 processChainCode ("addpartialidp")
Checks vidp existence; Create if not, update if yes
5 invokeChainCode ("updatevirtualidp")
6 response("addpartialidp")
7 invokeChainCode ("addschema", public-Parameters)
Checks schema existence. Error if exists If not, associate to idp
8 processChainCode ("addschema")
9 response("addschema")
10 OLYMPUS End Setup
vIdP
Ledger

FIGURE 4. Partial IdP and vIdP ledger registration.

As previously mentioned, preventing an IdP from becoming a controller is an important task. No IdP should have more tasks associated with it than another. That is, they all have the same responsibilities. None of them can decide when to update the vIdP, and as a consequence, the problem of a malicious partial IdP trying to control the creation or update of the vIdP is eliminated. The last step in the registration is to add the public parameters to the ledger, steps 7 to 9.

Service providers wishing to operate using the proposed solution must also go through a small configuration process as they include the necessary verifier to validate user presentations (Figure 5).

Service providers receive a configuration regarding the location of the provided Blockchain Handler. To operate with an identity provider based on the proposed solution, the service provider needs to obtain the corresponding vIdP information and to do so, it will query the Blockchain Handler (*getvidp*), obtaining the connection data of the vIdP from the ledger. In this way, the service provider already knows that the vIdP is legitimate or, at least, that it was reliably registered, steps 1 to 7. Next, to configure the verifier, it needs the cryptographic parameters it obtains through the Blockchain Handler, steps 8 to 12. At this point, the service provider can already verify presentation tokens generated by the proposed solution. The last step is to register the service, for which it again makes use of the Blockchain Handler and the *addservice* smart contract to which it passes the data shown in the listing 3.

With the vIdP and the services registered in the ledger, the scenario is ready for the users. Assuming that the user

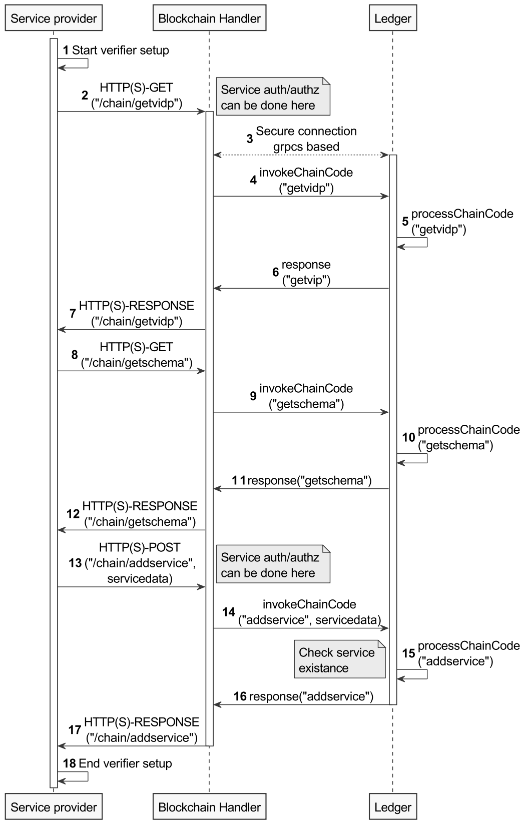

Service provider
Blockchain Handler
Ledger
1 Start verifier setup
2 HTTP(S)-GET ("/chain/getvidp")
Service auth/authz can be done here
3 Secure connection grpcs based
4 invokeChainCode ("getvidp")
5 processChainCode ("getvidp")
6 response ("getvip")
7 HTTP(S)-RESPONSE ("/chain/getvidp")
8 HTTP(S)-GET ("/chain/getschema")
9 invokeChainCode ("getschema")
10 processChainCode ("getschema")
11 response("getschema")
12 HTTP(S)-RESPONSE ("/chain/getschema")
13 HTTP(S)-POST ("/chain/addservice", servicedata)
Service auth/authz can be done here
14 invokeChainCode ("addservice", servicedata)
Check service existance
15 processChainCode ("addservice")
16 response("addservice")
17 HTTP(S)-RESPONSE ("/chain/addservice")
18 End verifier setup
Service provider
Blockchain Handler
Ledger

FIGURE 5. Service ledger registration.

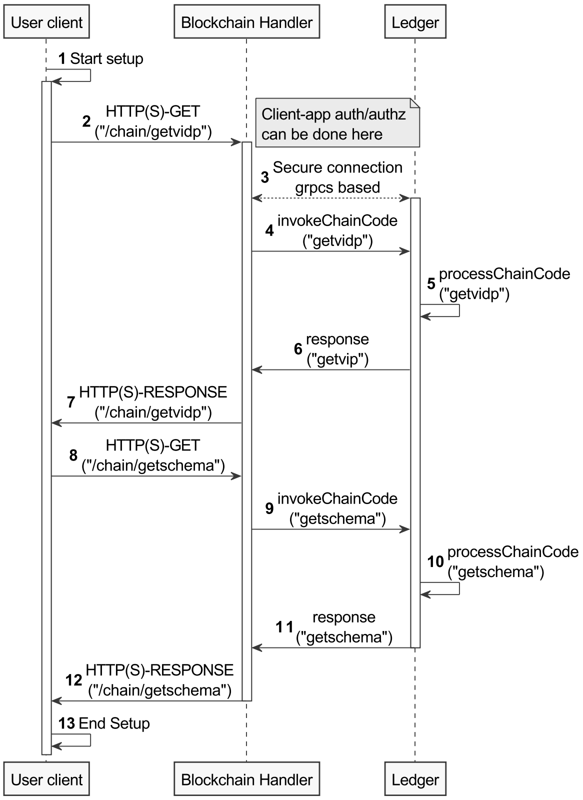

User client
Blockchain Handler
Ledger
1 Start setup
2 HTTP(S)-GET ("/chain/getvidp")
Client-app auth/authz can be done here
3 Secure connection grpcs based
4 invokeChainCode ("getvidp")
5 processChainCode ("getvidp")
6 response ("getvip")
7 HTTP(S)-RESPONSE ("/chain/getvidp")
8 HTTP(S)-GET ("/chain/getschema")
9 invokeChainCode ("getschema")
10 processChainCode ("getschema")
11 response ("getschema")
12 HTTP(S)-RESPONSE ("/chain/getschema")
13 End Setup
User client
Blockchain Handler
Ledger

FIGURE 6. Client auto-configuration process.

client has a previous minimum configuration concerning the Blockchain Handler endpoint, the user client start their interaction by launching an auto-configuration process that will culminate with the obtention of the vIdP connection parameters as well as the cryptographic primitives (Figure 6).

For this purpose, it makes a query to the Blockchain Handler that acts as an intermediary between the client and the ledger, deployed by the organization or organizations with a static configuration assumed as reliable. Step 3 indicates that a secure connection has been established between the entity and the ledger.

Once the connection between the ledger and the Blockchain Handler is established, the *getvidp* smart contract (chaincode) is invoked, returning the corresponding data (steps 5 to 7). The last step, performed by the client, is to obtain the public parameters associated with the vIdP it has received, and to do so, it starts the query through the Blockchain Handler who will invoke the *getschema* chaincode. The client will receive the encoded scheme public parameters as well as the attribute definitions associated with the vIdP. At this point, the client can recheck the parameters by directly asking the vIdP to verify that they are indeed the same. However, it is sufficient to assume that they are true because if they have changed, the following processes will not work.

Once all elements are deployed and configured, users are ready to operate and make use of the available services. The available services can be obtained through any discovery methods. The user application can check, using the Blockchain Handler and the ledger, that a service has been previously registered. For example, verify that a service has not changed endpoints without notification. This feature adds extra security and confidence against the possibility of a phishing attack or service spoofing.

According to these characteristics, Figure 7 describes how a user can make use of a service offered within the presented proposal. First, the user selects a service from those available by obtaining an access policy, steps 1 to 3. Then begins an internal verification in which the client retrieves information about the service he wants to access from the ledger through the *getservice* smart contract. The ledger contains the service record and the data it declared it would use from the users (policy), steps 3 to 7. The application compares the information received (service policy and the one recorded in the ledger) and warns the user if something has changed, step 8.

At this point, the user visualizes the policy applied to access the service and makes a decision. If the user continues, the next step is to obtain the dP-ABC credential, listing 4, with Verifiable Credential format [36], with the particularity of the *proof* field. We have extended the Verifiable Credentials and Verifiable Presentations data model to accommodate the cryptography introduced by OLYMPUS. It includes an epoch for expiration purposes, the purpose of the proof, the proof value, and the type of signature it contains. The credential is obtained from vIdP in steps 9 to 12. In this case, the

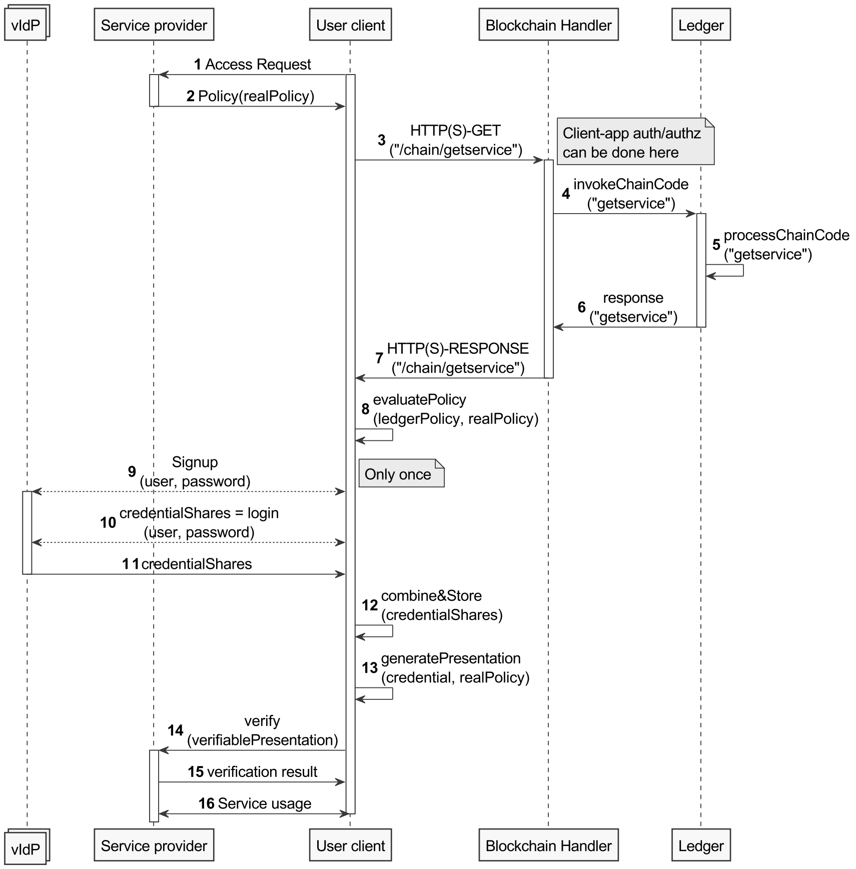

vIdP
Service provider
User client
Blockchain Handler
Ledger
1 Access Request
2 Policy(realPolicy)
3 HTTP(S)-GET ("/chain/getservice")
Client-app auth/authz can be done here
4 invokeChainCode ("getservice")
5 processChainCode ("getservice")
6 response ("getservice")
7 HTTP(S)-RESPONSE ("/chain/getservice")
8 evaluatePolicy (ledgerPolicy, realPolicy)
9 Signup (user, password)
Only once
10 credentialShares = login (user, password)
11 credentialShares
12 combine&Store (credentialShares)
13 generatePresentation (credential, realPolicy)
14 verify (verifiablePresentation)
15 verification result
16 Service usage
vIdP
Service provider
User client
Blockchain Handler
Ledger

FIGURE 7. Service access.

```
{
"@context": ["https://w3id.org/credentials/v1",
"https://olympus-deployment.eu/example/context"],
"type": ["VerifiableCredential", "OlympusCredential"],
"credentialSchema": [{
    "id": "https://olympus-project.eu/example/validationSchema",
    "type": "OlZkValidationSchema"
}, {
    "id": "https://olympus-project.eu/example/encodingSchema",
    "type": "OlZkEncodingSchema"
}],
"issuer": "did:meta:OL-vIdP",
"issuanceDate": "2021-05-31T11:29:28",
"expirationDate": "2021-06-01T07:29:28",
"credentialSubject": {
    "annualSalary": 35000,
    "role": "student",
    "mail": "mail@um.es",
    "organization": "UMU",
    "dateOfBirth": "1989-01-05T00:00:00"
},
"proof": {
    "epoch": 1622532568000,
    "proofPurpose": "AssertionMethod",
    "proofValue": "CjwKOgA[...]AAAAAAAAAA",
    "type": "OlPsSignature"
    }
}
```

LISTING 4. Example of leveraged verifiable credential.

credential obtained shows a series of attributes: annual salary, role, email, organization, and date of birth on which the user will be able to perform tests. The credential has a "proof" field that includes a digital signature. This signature is a group signature generated by the N partial IdPs that make up the vIdP.

The reconstructed credential is stored locally (until its expiration), avoiding the need to always go to the vIdP. Finally, with the obtained credential, a Verifiable Presentation Token, listing 5, is generated and presented to the service to obtain the requested access, steps 13 to 16. The presentation generated for a particular access policy is observed now. The disclosed attribute is included as well as the cryptographic proof.

Throughout the entire process, the components and services were always supported by the blockchain infrastructure and smart contracts. It is possible to consult the parameters of the vIdP or the registered services at any time without affecting the user's experience. In this regard, the following section presents concrete implementation details and an evaluation of the proposed solution.

IV. IMPLEMENTATION AND EVALUATION

This section describes details of the implementation accomplished and presents an evaluation of this implementation.

The realised implementation is structured as follows. An Android-based user client, three JAVA-based partial IdPs, a Javascript-based Blockchain Handler entity and a blockchain platform. The development of the user client on

```
{
"@context": ["https://w3id.org/credentials/v1",
    "https://olympus-deployment.eu/example/context"],
"type":
    ["VerifiablePresentation", "OlympusPresentation"],
"expirationDate": "2021-05-31T11:35:34",
"verifiableCredential": [{
    "credentialSchema": [{
        "id":
            "https://olympus-project.eu/example/validationSchema
                ",
        "type": "OlZkValidationSchema"
    }, {
    "id":
        "https://olympus-project.eu/example/encodingSchema",
    "type": "OlZkEncodingSchema"
    }],
    "credentialSubject": {
        "organization": "UMU"
    },
    "issuanceDate": "2021-05-31T11:29:28",
    "issuer": "did:meta:OL-vIdP",
    "expirationDate": "2021-06-01T07:29:28",
    "proof": {
        "epoch": 1622532568000,
        "nonce": "OLYMPUS-POLICY-1893291946",
        "proofPurpose": "AssertionMethod",
        "proofValue": "CvMBCvABCjoKd6Ke[...]",
        "type": "OlPsDerivedProof"
    },
    "type": ["VerifiableCredential", "OlympusCredential"],
    "@context": ["https://w3id.org/credentials/v1",
        "https://olympus-project.eu/context",
        "https://olympus-deployment.eu/example/context"]
    }]
}
```

LISTING 5. Example of leveraged verifiable presentation.

Android is simple, only needing the OLYMPUS dependencies and small code extensions for the scenario. Meanwhile, partial IdPs incorporate the functionality to communicate with the blockchain platform natively and are entirely dependent on the type of platform selected.

Selecting and integrating a blockchain platform is not a trivial decision. Among the available options, those based on the Hyperledger [31] project have been considered. The first option was to use the Indy [41] project since it is focused on identity management. A small integration test based on this platform was performed, but it did not meet all our expectations. Mainly it was because of the lack of support for smart contracts [42], [43]. The next option was the Fabric [44] project, with more functionalities. Among others, it includes smart contracts and a modular design. It also promises low latency and advanced privacy management. Although it is not a project specifically designed for identity management, it does provide the desired flexibility and functionality. In addition, the scalability and efficiency of Fabric have already been analyzed in several studies [45], [46] with promising results, such as the possibility of handling 200 transactions per second and more than 100,000 participants with an average response time of 0.01 seconds for 100,000 "query" transactions requests [47].

In Fabric, smart contracts are called Chaincodes and are closely related to the use of channels. Channels are private sub-networks for communication between two or more specific network members to perform private and confidential transactions. More detailed, the blockchain deployment has an organization composed of the organizations or members, anchor peers, the order node, chain codes, and the shared ledger, figure 8.

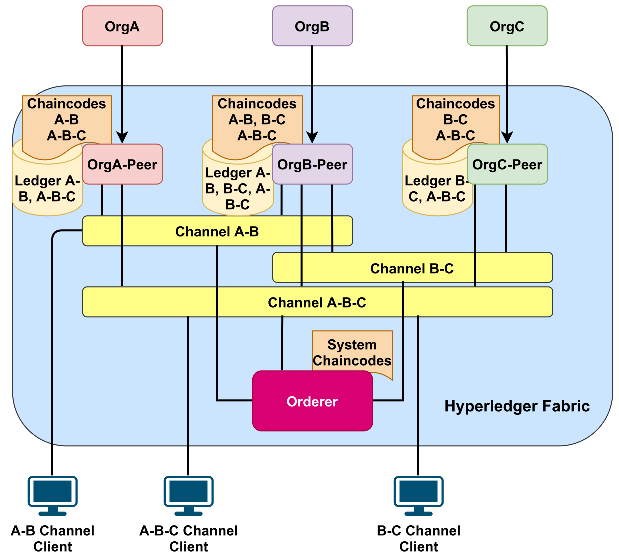


FIGURE 8. Hyperledger fabric channels and chaincodes.

Each transaction on the network is executed on a channel. In addition, each party must be authenticated and authorized to perform transactions on that channel. The proposal made (figure 2) includes an extended trust substructure that enhances the OLYMPUS architecture. The entities involved modify their behavior to use the blockchain infrastructure and chaincodes to query and enter data into the ledger.

Although OLYMPUS IdPs natively support interaction with Fabric, this is not the case for user clients and service providers. There are several reasons for this situation. Firstly, they would make deployments of these entities too heavy and secondly, these entities should be agnostic to the blockchain infrastructure being used. For these reasons, we have chosen to develop the Blockchain Handler entity, which allows both users and service providers to interact with the ledger without having to perform additional configurations.

A. BLOCKCHAIN HANDLER

It is a REST API that provides interaction with the ledger deployed by our solution via common HTTP methods. It is based on JavaScript and allows a light and easy to extend development. In addition to the methods for interacting with the ledger, it allows us to simulate different elements such as the service provider with its respective verifier.

Next, we present a performance evaluation based on a test scenario, figure 9, consisting of the following elements: First, Hyperledger Fabric v2 was deployed using OpenStack. Each Fabric virtual host consists of 2GB of RAM and a single virtual core. The Fabric infrastructure consists of 2 organizations with two peers each, a certification authority and an Orderer node. Every Hyperledger machine is running Docker v19, and Docker compose v6.14. Secondly, there

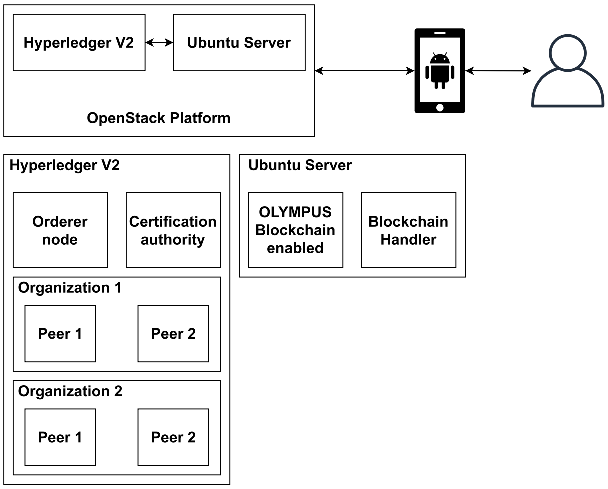


FIGURE 9. Deployed scenario.

is an Ubuntu Server 18.04, also virtualized by OpenStack, with four cores, 8GB of RAM, and the necessary software suite to run the tests: Docker v19, Docker compose v1.17, NPM v6.14, and JAVA v1.8. Finally, an Android emulator based on the Pixel 3A model with 4GB of RAM and a virtualized CPU is available, as well as an Android device model, OnePlus 6T, that has 6GB of RAM 8-core Qualcomm Snapdragon 845 processor.

First, the performance of the OLYMPUS vIdP was evaluated. Specifically during the setup phase of a partial IdP. The times involving accesses to the ledger for the smart contracts *getpartialidp, addourupdateidp* and *addschema* have been collected. Additional measurements have been taken depending on where the IdP is deployed, i.e., measurements marked as local mean that both the ledger and IdP are on the same network and remote measurements mean that the IdP and ledger are on different networks. While both are valid in real deployments, the remote scenario where an organization deploys its IdPs without the ledger necessarily being in its domain would be the closest to an actual use case. Figure 10, shows the results obtained. It can be seen how placing the deployment of IdPs and ledger in the same network increases the performance, being more noticeable in heavy processes that involve multiple interactions such as *addorupdateidp*.

The data obtained show that the setup process of a partial IdP can take between 33.43 to 36.1 and 41.1 to 45.86 seconds, depending on whether they are deployed in the same network or not (Figure 11). The difference is significant and is mainly in the remote scenario where there are many configuration hops. In contrast, the local scenario benefits from fewer hops and lower latency due to network sharing. Considering that the setup process must be performed only once, it is a manageable time that would not be detrimental for a production release.

Continuing with the evaluation, the next focal entity of the measurements is the **Blockchain Handler**. This element will receive queries from both user clients and service providers, and consequently, its performance is critical to the proper functioning of the scenario. Figure 12 shows the times obtained for each of the available methods: *getvidps, getvidp, getschema, addservice, getservices* and *getservice*.

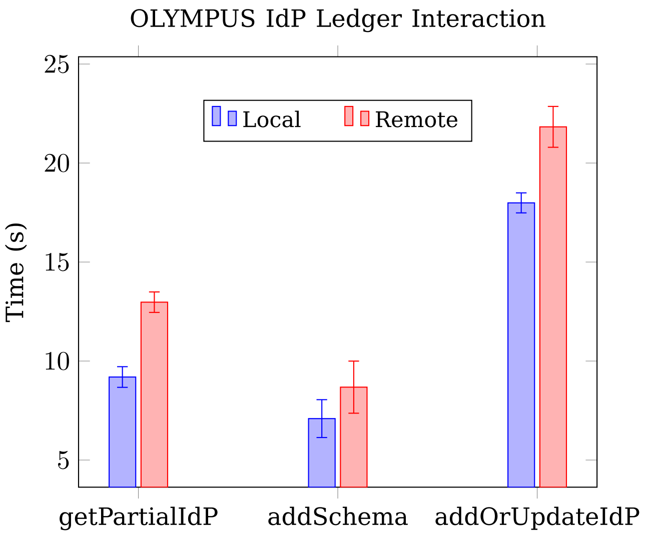


FIGURE 10. OLYMPUS IdP and Ledger interaction.

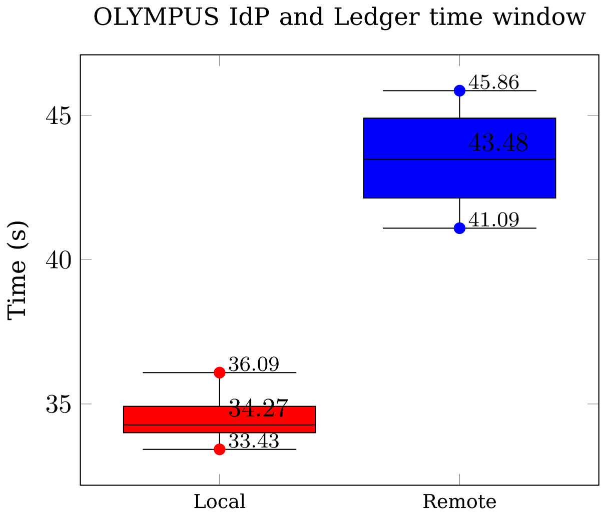


FIGURE 11. OLYMPUS IdP and Ledger time window.

In general, the times obtained with a deployment where ledger and Blockchain Handler are stationed in the same network are sensibly lower than those obtained with remote deployment. It is also observed that the most expensive methods are those that require collecting information from the ledger, such as *getvidps* and *addservice*. Both methods involve searches on the blockchain that become more expensive as the blockchain grows, which could be problematic in a larger scenario.

In addition, times for the verifier and the user application have been obtained. Precisely, the time required to launch and setup the verifier was measured along with the time required to auto-configure the user application.

The Figure 13 shows the auto-configuration times for both the verifier and the user application. As can be seen, for

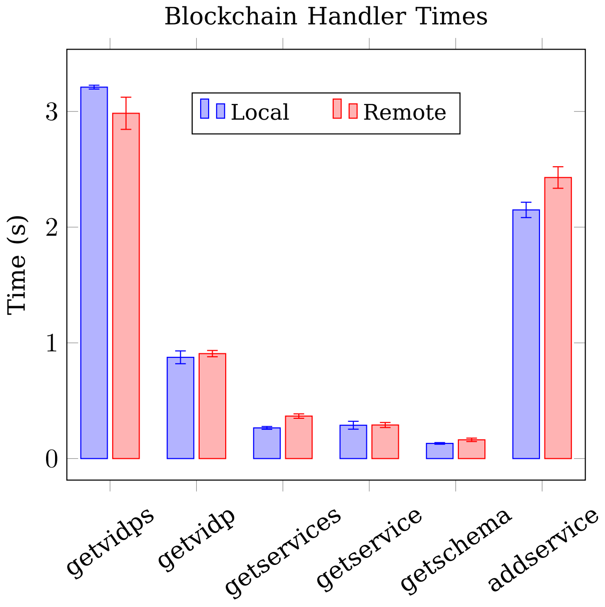


FIGURE 12. Blockchain handler times.

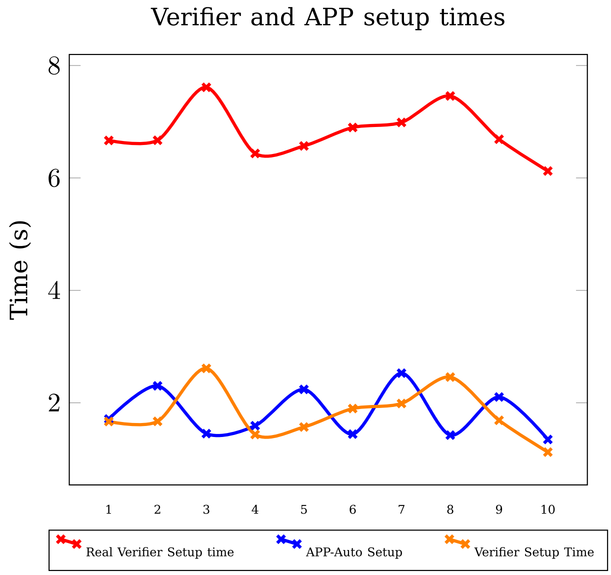


FIGURE 13. Verifier and APP setup times.

users, the auto-configuration process is not a major overhead. The auto-configuration includes retrieving the vIdP to use, and obtaining the necessary cryptographic material from the ledger through the Blockchain Handler and using it to perform OLYMPUS-related configuration.

The verifier is placed on the same server as the Blockchain Handler, responsible for instantiating verifiers when needed. When the Blockchain Handler receives a verify request for a given vIdP DID, it first checks if there are any running instances for that DID; if not, it initiates the instantiation process. When a new verifier instance is launched the Blockchain Handler waits 5 seconds for it to come online. The 5-second wait has to do with the need to instantiate a verifier that runs on JAVA and whose invocation is being performed from an element that is not directly compatible (JavaScript). The wait time is applied to make sure that the verifier has been launched correctly before interacting with it. In this sense, the obtained times distinguish between real-time, which includes the 5-seconds, and pure time, which excludes it. After that, the setup interface is used to configure it. That interface receives the result of a query to the ledger for the smart contract *getvidp*, whose times are shown in Figure 12.

As in the case of the IdP, the verifier only needs to be instantiated and configured once. It will only require reconfiguration in case of changing the cryptographic parameters of the IdP, e.g., the credential structure.

With the verifier ready, the user is ready to obtain Verifiable Credentials, generate Verifiable Presentations and perform verification requests. In this sense, different tests have been performed generating Verifiable Presentations, measuring their generation time and later the verification time. In our tests, a user credential is defined by the following attributes and may contain some or all of them:

- url:DateOfBirth
- url:Mail
- url:Organization
- url:Role
- url:AnnualSalary

Figure 14 shows the times obtained when doing a presentation process with different policies (i.e., attributes revealed and range proofs), while Figure 15 illustrates the proportion of time consumption of each sub-process. At a glance, it can be seen that revealing is much lighter than range

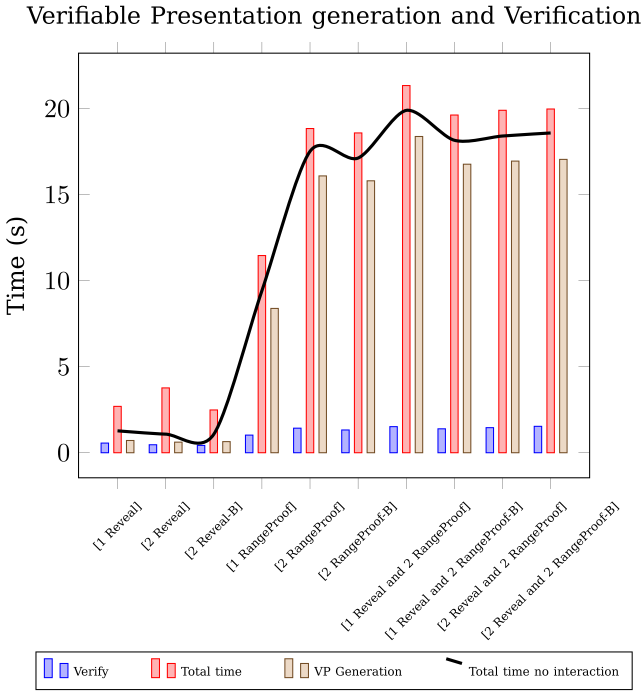


FIGURE 14. Verifiable presentation generation and verification.

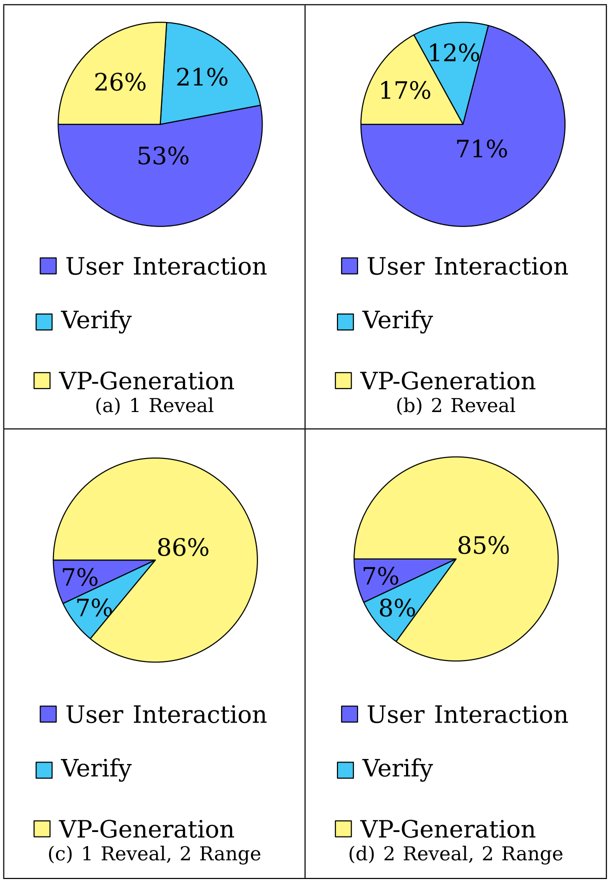


FIGURE 15. Time proportion between verify, generate the presentation and total time.

proofs, which are computationally more expensive. The figures also show the contrast between verification times when there is user intervention and no user intervention. User intervention indicates that the user must have consented to generate the verifiable presentation for a given access policy, while non-interaction only shows the time of internal processes in which the user has not had any interaction. We can see that, in scenarios where the proof only requires attribute revelation, generation and verification times are similar and have a similar impact as user interaction in total execution time (i.e., not a heavy overheard). However, when the computationally-complex range proofs are needed, the VP generation times are much higher. This is due to the range proof protocol itself (generation is heavier than verification), but above all because of the more constrained hardware on the user side.

V. SECURITY ANALYSIS

This section introduces a brief security analysis of the proposed solution, covering known problems and possible ways of mitigation.

The proposed solution is a combination of distributed systems. On the one hand, we have the OLYMPUS solution, and on the other, Hyperledger Fabric.

OLYMPUS follows a distributed security model, meaning that all partial IdPs must be corrupted for an adversary to jeopardize the security. The OLYMPUS system ensures that, as long as a single partial IdP is not corrupted, it is impossible to forge identities or impersonate any user. Moreover, it is not possible to brute force attack user passwords because of the partially-oblivious distributed pseudo-random functions [37] used for distributed password authentication. OLYMPUS incorporates proactive security, which means that if a partial IdP has had its secret cryptographic material compromised, it is possible to refresh this material, preventing the adversary from using the previously secret information to impersonate it.

The incorporation of Blockchain through the Hyperledger Fabric (HLF) platform leaves OLYMPUS's strengths completely preserved; however, systems like HLF enable distributed applications running smart contracts (*chaincodes*) on Nodes (peers) belonging to multiple cooperating organizations. These nodes intercommunicate on a network that updates multiple copies of a distributed network ledger that contain exact replicas of ordered blocks of data. The distributed nodes of these systems are usually running on standard computers and are spread over multiple domains, making them a good target for attackers. Works such as Brotsis *et al.* [48], Andola *et al.* [49], Yamashita *et al.* [50], and Dabholkar *et al.* [51] have already dealt with security analysis of HLF, including common attacks and problems (i.e., DDoS attacks, double registration problem or tampering), structural challenges and possible problems arising from smart contracts implementation (i.e., not using specific languages like Solidity[1]), without revealing major shortcomings. In that sense, the security of the HLF infrastructure will depend to a greater extent on good practices in the deployment and development.

We designed data models taking into account that some assets are potential targets of stale data attacks. Mainly those that define public parameters for vIdPs and service providers. The relevant entries include two fields that can thwart attempted attacks. First, the *status* can be modified to show that an asset is no longer active. Also, these entries contain a DID document. The secret key corresponding to the DID would only be available to the rightful owner (if not, the compromise would come from a different attack), so the validity of entries and the legitimacy of another party can always be checked during interactions by a challenge to that secret identity. Lastly, we remark that there is no additional risk of stale data attacks that harm user privacy, as the ledger will not store any sensitive information. What is more, possible attacks against service providers are mitigated because of P-ABCs' privacy features (i.e., minimal disclosure and unlinkability).

The security of the user identity during interactions is assured through the properties of the P-ABC scheme [38]. The IdPs are not involved in a presentation process, and

[1] https://solidity-es.readthedocs.io

service providers only receive a short-lived zero-knowledge presentation token. Thus, even if multiple service providers and the IdPs collude, they will not obtain more information than the user accepted or cryptographic data that can be used for impersonation attacks. What is more, unlinkability ensures that different presentations cannot be linked to the original user, ensuring privacy. Note that, as with any other mechanism, privacy may be breached through "external" attacks (fingerprinting…) or if the user decides to reveal identifying information.

In the proposed solution, connectivity between IdPs and the HLF platform is done directly, taking advantage of the security infrastructure provided by Fabric. This ensures a legitimate connection between IdPs and ledger. Nonetheless, this is not the case for users and service providers, where the Blockchain Handler entity becomes relevant. Although a direct connection of customers and service providers to the ledger would be desirable, this is not entirely realistic. Direct integration is cumbersome and complex to manage at these ends. In addition, problems arise, such as disparate devices or the need to handle excessive cryptographic material. Therefore, the Blockchain Handler entity provides an intermediate form of connection, which opens the door to other significant concerns about security (point of attack) and trust.

This entity provides connectivity data to users and service providers as a starting point to begin operations. Whether or not to trust this entity is not a trivial decision. While in the test scenario, this entity is deployed and trusted manually without users or providers having to intervene, the interaction is more complicated in a real scenario. As is already the case with other services such as e-banking or government services, trust in this entity is guaranteed through digital certificates issued by a trusted authority (i.e., government), thus placing the final decision in the hands of users or service providers.

However, these issues can be mitigated, increasing security and trust. For example, we can apply techniques to make the handler distributed (e.g., distributed computing). We could even leverage the permission mechanism of the ledger to allow different entities to perform the handler role. In this case, different approaches like trust scoring can be applied to improve the result. Lastly, we remark that it is possible to make the two options coexist: the chain offers a handler for users/devices that need it but is still directly accessible if an actor wishes to, avoiding the issues mentioned above.

Another advanced security functionality, revocation of credentials, is still a semi-open problem. Groups of P-ABC credentials can already be revoked through the *epoch* attribute, which is mandatory to reveal during presentations and is already used for managing credential lifetimes. The ledger would be an appropriate tool for maintaining a list of epochs revoked by each vIdP. Some new smart contracts would be needed, but they would be effortless. However, it is much more desirable to have fine-grained revocation. This is left for future work because the modifications to the P-ABC cryptography needed are still not fully developed. Again, the ledger would be a great assisting tool, suitable for storing the public information needed by the revocation scheme (e.g., publishing and updating accumulators if revocation is based on them).

VI. CONCLUSION AND FUTURE WORK

The purpose of OLYMPUS is to provide a user-friendly ecosystem in which privacy is the crucial element. The inclusion of blockchain seeks to increase trust in the infrastructure by providing the means for users, service providers, and identity providers to enjoy the features that OLYMPUS provides and perceive an ironclad trust throughout the infrastructure at all stages.

This work demonstrates that integration between distributed identity provider technology (OLYMPUS) and blockchain is possible and provides the desired trust. The pilot deployment and subsequent analysis performed indicate no usability penalty in the entities involved or considerable impediments hindering adoption.

The solution shown allows users to have confidence during all the stages; for example, the identity providers used are legitimate, and that the services do not unilaterally change their access policies with hidden intentions. When a service wishes to change its access policies, it must notify the infrastructure in advance; otherwise, the user will be warned of possible dishonest use.

However, during development, some problems have been encountered that need to be improved in future work. Firstly, the cryptographic elements need to be further optimized for more limited devices (e.g., IoT devices). Secondly, smart contracts and, in general, the way queries are made to the blockchain platform must be revised to avoid excessive growth of query times. It has been found that as the blockchain grows, so do the query times. The greater the number of blocks, the greater the distance to explore. In medium or small scenarios, the times may be manageable but they could be challenging in significant scenarios if no action is taken. In any case, this problem can be solved. Firstly by making use of the indexes already built into Hyperledger Fabric and secondly by applying updates and patches to Fabric as the project is optimised and reviewed by the community.

In addition, the existing P-ABC solutions on blockchain are not fully distributed. They do not split traditional IDP over multiple partial IDPs. This paper proposes the first fully distributed, trusted, oblivious, blockchain-based, and privacy-preserving ABC Identity management system to the best of our knowledge. This proposal [2] achieves the goal of providing an enhanced trust system. Partial IdPs, virtual IdPs, and even public cryptographic data are reflected in the ledger so that users and other entities can consult them at any time. Service providers go through a similar registration process so that users can easily detect changes in criteria. In addition, all operations are achieved through the execution of smart contracts and transparent to the users.

[2] The code will be publicly available in https://github.com/rafaeltm/OLChainEnabled once the paper is published.

To conclude, we are currently working on three ways to improve the proposed scenario. Firstly, automating processes (i.e., registration of SPs), thus avoiding excessive manual configurations that may lead to errors. Secondly, revocation of credentials with blockchain support. Revocation is an important issue and needs to be addressed to ensure full functionality. While it is possible to revoke groups of credentials with the current deployment, the goal should be to achieve fine-grained revocation. Blockchain and smart contracts are positioned as an excellent way to solve this challenge. Finally, the application of a blockchain-backed trust scoring system can significantly increase the prospects of the scenario, even allowing the application to IoT scenarios where devices often operate unattended and where the action is necessary to reduce attacks such as node hijacking or impersonation.

REFERENCES

[1] K. O'Flaherty, "Collection 1 breach–how to find out if your password has been stolen," *Forbes*, Jan. 2019. [Online]. Available: https://www.forbes.com/sites/kateoflahertyuk/2019/01/17/collection-1-breach-how-to-find-out-if-your-password-has-been-stolen/

[2] L. H. Newman, *Equifax Officially Has no Excuse*. San Francisco, CA, USA: Wired, 2017.

[3] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," in *Practical Guide*, 1st Ed. Cham, Switzerland: Springer, 2017.

[4] N. Hong, L. Hoffman, and A. Andriotis, "Capital one reports data breach affecting 100 million customers, applicants," *Wall Street J.*, Jul. 2019. [Online]. Available: https://www.wsj.com/articles/capital-one-reports-data-breach-11564443355

[5] R. Maull, P. Godsiff, C. Mulligan, A. Brown, and B. Kewell, "Distributed ledger technology: Applications and implications," *Strategic Change*, vol. 26, no. 5, pp. 481–489, 2017.

[6] D. Tapscott and A. Tapscott, *Blockchain Revolution: How Technology Behind Bitcoin is Changing Money, Business, World*. Baltimore, MD, USA: Penguin, 2016.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Bitcoin.org, Tech. Rep., 2019.

[8] J. Bernal Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. Torres Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[9] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *Sovrin Found.*, vol. 29, no. 2016, 2016.

[10] *European Self Sovereign Identity Framework*, Informatie Vlaanderen, Belgium, Jun. 2019.

[11] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[12] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, Oct. 1985.

[13] R. T. Moreno, J. Bernal Bernabe, J. García Rodríguez, T. K. Frederiksen, M. Stausholm, N. Martínez, E. Sakkopoulos, N. Ponte, and A. Skarmeta, "The OLYMPUS architecture—Oblivious identity management for private user-friendly services," *Sensors*, vol. 20, no. 3, p. 945, 2020.

[14] R. Housley, W. Ford, W. Polk, and D. Solo, *Internet X. 509 Public Key Infrastructure Certificate and CRL Profile*, document RFC 2459, Jan. 1999.

[15] J. De Clercq, "Single sign-on architectures," in *Proc. Int. Conf. Infrastruct. Secur.* Berlin, Germany: Springer, 2002, pp. 40–58.

[16] K. Rannenberg, J. Camenisch, and A. Sabouri, "Attribute-based credentials for trust," in *Identity in the Information Society*. Berlin, Germany: Springer, 2015.

[17] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Proc. Conf. Adv. Cryptol.*, 2001, pp. 93–118.

[18] P. Bichsel, J. Camenisch, M. Dubovitskaya, R. R. Enderlein, S. Krenn, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, F. Preiss, K. Rannenberg, and A. Sabouri, "An architecture for privacy-ABCs," in *Attribute-based Credentials for Trust: Identity Information Society*. Berlin, Germany: Springer, 2015, pp. 11–78.

[19] D. Hardt, *The Oauth 2.0 Authorization Framework*, document RFC 6749, Oct. 2012.

[20] J. Hughes and E. Maler, "Security assertion markup language (SAML) V2. 0 technical overview," in *Proc. OASIS SSTC Workshop Draft*, 2005, pp. 29–38.

[21] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 21–30.

[22] J. Camenisch, S. Krenn, A. Lehmann, G. L. Mikkelsen, G. Neven, and M. O. Pedersen, "Formal treatment of privacy-enhancing credential systems," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Berlin, Germany: Springer, 2015, pp. 3–24.

[23] C. Paquin and G. Zaverucha, "U-prove cryptographic specification V1. 1," Microsoft Corp., Albuquerque, NM, USA, Tech. Rep., 2011. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/U-Prove20Cryptographic20Specification20V1.1.pdf

[24] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.

[25] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, *Zcash Protocol Specification*. San Francisco, CA, USA: GitHub, 2016.

[26] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed E-Cash from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 397–411.

[27] D. Khovratovich and J. Law, "Sovrin: Digital identities in the blockchain era," *Github Commit Jasonalaw*, vol. 17, pp. 1–5, Oct. 2017.

[28] *Uport Project*. Accessed: Jul. 2021. [Online]. Available: https://github.com/uport-project/specs

[29] *A Decentralized, Open Source Solution for Digital Identity and Access Management*. Berlin, Germany: Jolocom, 2017.

[30] S. ShoCard. (2016). *Travel Identity of the Future*. [Online]. Available: https://whitepaper.uport.me/uPort_whitepaper_DRAFT20170221.pdf

[31] V. Dhillon, D. Metcalf, and M. Hooper, "The hyperledger project," in *Blockchain Enabled Application*. Berlin, Germany: Springer, 2017, pp. 139–149.

[32] M. Sabadello, D. Reed, D. Longley, M. Sporny, and C. Allen. (Feb. 2021). *Decentralized Identifiers (DIDs) V1.0*. [Online]. Available: https://www.w3.org/TR/2021/WD-did-core-20210222/

[33] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2014.

[34] R. T. Moreno, J. G. Rodríguez, C. T. López, J. B. Bernabe, and A. Skarmeta, "OLYMPUS: A distributed privacy-preserving identity management system," in *Proc. Global Internet Things Summit (GIoTS)*, 2020, pp. 1–6.

[35] J. Camenisch, "Distributed single sign-on," U.S. Patent 10 164 964, Dec. 25, 2018.

[36] M. Sporny, D. Longley, and D. Chadwick, "Verifiable credentials data model 1.0," W3C, Cambridge, MA, USA, Tech. Rep., Nov. 2019. [Online]. Available: https://www.w3.org/TR/vc-data-model/

[37] C. Baum, T. Frederiksen, J. Hesse, A. Lehmann, and A. Yanai, "PESTO: Proactively secure distributed single sign-on, or how to trust a hacked server," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 587–606.

[38] J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa, "Short threshold dynamic group signatures," in *Proc. Int. Conf. Secur. Cryptogr. Netw.* Berlin, Germany: Springer, 2020, pp. 401–423.

[39] A. Skarmeta, "D4.2 first reference implementation of oblivious IDM," Dept. Inf. Commun. Eng., Univ. Murcia, Murcia, Spain, Tech. Rep., Mar. 2020. [Online]. Available: https://olympus-project.eu/wp-content/uploads/2020/10/Olympus_pu_d4_2_v1_2.pdf

[40] H. L. Towa, "D4.1 cryptographic design of an oblivious idm system," Dept. Inf. Commun. Eng., University of Murcia, Tech. Rep., Oct. 2019. [Online]. Available: https://olympus-project.eu/wp-content/uploads/2019/12/Olympus_pu_d4_1_v1.0.pdf

[41] *Hyperledger Indy*. Accessed: Jul. 2021. [Online]. Available: https://indy.readthedocs.io/en/latest/

[42] N. Szabo, "Smart contracts," First Monday, Chicago, IL, USA, Tech. Rep., Sep. 1997, vol. 2, no. 9, doi: 10.5210/fm.v2i9.548.

[43] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[44] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, Chicago, IL, USA, vol. 310, 2016, pp. 1–4.

[45] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *Proc. IEEE 26th Int. Symp. Modeling, Anal., Simulation Comput. Telecommun. Syst. (MASCOTS)*, Sep. 2018, pp. 264–276.

[46] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, "Performance modeling of hyperledger fabric (permissioned blockchain network)," in *Proc. IEEE 17th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2018, pp. 1–8.

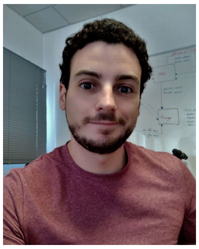
[47] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 536–540.

[48] S. Brotsis, N. Kolokotronis, K. Limniotis, G. Bendiab, and S. Shiaeles, "On the security and privacy of hyperledger fabric: Challenges and open issues," in *Proc. IEEE World Congr. Services (SERVICES)*, Oct. 2020, pp. 197–204.

[49] N. Andola, Raghav, M. Gogoi, S. Venkatesan, and S. Verma, "Vulnerabilities on hyperledger fabric," *Pervas. Mobile Comput.*, vol. 59, Oct. 2019, Art. no. 101050.

[50] K. Yamashita, Y. Nomura, E. Zhou, B. Pi, and S. Jun, "Potential risks of hyperledger fabric smart contracts," in *Proc. IEEE Int. Workshop Blockchain Oriented Softw. Eng. (IWBOSE)*, Feb. 2019, pp. 1–10.

[51] A. Dabholkar and V. Saraswat, "Ripping the fabric: Attacks and mitigations on hyperledger fabric," in *Proc. Int. Conf. Appl. Techn. Inf. Secur.* Berlin, Germany: Springer, 2019, pp. 300–311.



RAFAEL TORRES MORENO received the B.S. and M.Sc. degrees in computer science from the University of Murcia, in 2017, where he is currently pursuing the Ph.D. degree. He is also a Research Fellow with the Department of Information and Communications Engineering, University of Murcia. He has been a Visiting Researcher with IBM Research Zürich. He has participated in H2020 EU research projects, such as OLYMPUS, CyberSec4EU, and ARIES. His research interests include security and privacy-preserving technologies applicable in decentralized systems.

JESÚS GARCÍA-RODRÍGUEZ received the B.S. degree in mathematics and the B.S and M.Sc. degrees in computer science from the University of Murcia, where he is currently pursuing the PhD. Degree. He is also a Research Assistant with the Department of Information and Communications Engineering, University of Murcia. He has participated in the H2020 EU research projects OLYMPUS and CyberSec4Europe. His research interests include privacy-enhancing and security technologies, including the cryptography behind them and their application to the Internet of Things.

JORGE BERNAL BERNABÉ received the B.S., M.S., and Ph.D. degrees in computer science and the M.B.A. degree from the University of Murcia, Spain. He is currently an Assistant Professor with the University of Murcia. He has been a Visiting Researcher with Hewlett-Packard Laboratories and the University of the West of Scotland. He has authored several book chapters and more than 60 articles in international top-level conferences and journals. During the last years, he has been working in several European research projects, such as SocIoTal, ARIES, OLYMPUS, ANASTACIA, INSPIRE-5G, and CyberSec4EU. His scientific activity is mainly devoted to the security, trust, and privacy management in distributed systems. He is also interested in security and privacy aspects in the Internet of Things.

ANTONIO SKARMETA (Member, IEEE) received the B.S. degree (Hons.) and the M.S. degree in computer science from the University of Granada, Granada, Spain, and the Ph.D. degree in computer science from the University of Murcia, Murcia, Spain. Since 2009, he has been a Professor with the University of Murcia. He has worked on different research projects in the national and international area in the networking, security, and the IoT area, like Euro6IX, ENABLE, DAIDALOS, SWIFT, SEMI RAMIS, SMARTIE, SOCIOTAL, and IoT6. His main interests are in the integration of security services, identity, the IoT, and smart cities. He has been the Head of the research group ANTS, since its creation in 1995. He is also an Advisor to the Vice Rector of Research of the University of Murcia for international projects and the Head of the International Research Project Office. Since 2014, he has been the Spanish National Representative for the MSCA within H2020. He has published over 200 international articles and has been a member of several program committees. He has also participated in standardization for IETF, ISO, and ETSI.