Blockchain for Optimized Digital Identity (DIDOs)

Abdessamad El akroudi
Paragraph research Lab, University of Paris VIII
Paris, France
e-mail: abdessamad.el-akroudi@etud.univ-paris8.fr

Rakia Jaziri
Paragraph research Lab, University of Paris VIII
Paris, France
e-mail: Rjaziri@univ-paris8.fr

***Abstract*—The blockchain is a new and emerging technology that gives us the opportunity to explore a new type of application, and a new class of problem solving. Blockchain has acquired massive popularity over the past few years. It has changed the way of thinking about data and trust, and the relation between them. This work is a part of our publication's series aiming to build a data trust on top of a decentralized identity (DID) which we name Data Trust Protocol. In this article, we focus on the DID. We propose an implementation of a DID based on W3C specification, as well as our vision about the future of data trust on top of a DID. We will answer some questions related to this implementation.**

Keywords - Decentralized Identity; Issuer; Holder; Verifier; MicroServices; DID Document; BlockChain; Data Registry;

I. INTRODUCTION

In the last decade, with the exploding number of applications that we use on a daily basis, we mostly use an email or a username or a key provided by organizations or services as our identifiers. Those providers become a single point of failure between users and every action they made which needs authentication. Google and Apple control two of the biggest federated login systems with "Login width" button. Users expose their identity details in a centralized manner with a major risk of identity theft, identity usurpation. The problem has become more apparent and centralized. The need for a new way to identify people and objects becomes more urgent with the development of the Internet of Things (IoT). IoT is defined as "An infrastructure of interconnected objects, people, systems and information resources together with intelligent services to allow them to process information of the physical and the virtual world and react" [1]. Every day we add a new connected object to our life starting from our new vision to define the place where we live: smart cities, parking management, energy management, personal health, and safety, remotely checking patient health. We even define agriculture, using connected and geolocated tractors and machines to optimize the energy spent and to improve the daily routine. The increasing growth of population and IoT infrastructure raises concerns about the feasibility of accommodating and managing interconnected objects and people effectively. Therefore, the W3 thinks about a new paradigm for identifying subjects, named the Decentralized Identity (DID) [4].

- Such a system is owned by the participants which means a total elimination of a single point of failure.
- Each entity should claim an attribute relation to the identity (themselves) and get trusted by an authority or an attested and verified one.
- The participants implement a system to agree on a common state.
- No single company controls the sensitive part of the system.
- Decentralizing the identity eliminates the risk of stealing a database storing millions of identifiers.
- The credentials should be stored in the same place as the owner (identity).

People can take back the ownership of their identity. With DID, you can manage the identity by yourself using a private key stored on a wallet; a very common example is social media. We can imagine our network in a special social network moved with our identity to another network, on the other side, so there is no need for companies to build a new social graph from scratch for any new social media. The ability to consent to the type of data you provide to a third party is the key in a DID system. This technique is achieved by Zero Knowledge proof which "are an elegant technique to limit the amount of information transferred from a prover A to a verifier B in a cryptographic protocol" [2]. As defined in the original GMR paper "a buyer A can assert to a seller B that his or her age is more than 18, which allows him or her to buy some specific product, without revealing the birth date" [3].

This paper proposes an implementation of a DID, which will allow an entity to transact in a seamless way with a decentralized ecosystem. A DID can be seen as a global unique identifier. In section 2, we present several DID projects that are relevant to our decentralized system under development. Moving on to section 3, we introduce the syntax specific to our system and provide an overview of the current architecture. Section 4 focuses on the DID Document and its role in addressing the challenge of public key rotation. In section 5, we showcase our architecture and propose the incorporation of a new layer to address the scalability issue.

II. RELATED WORK

DID systems have gained traction as an alternative to centralized identity management, ensuring security and privacy has emerged as a major challenge. Sybil attacks, secure key management, and privacy risks are just some of the issues that need to be addressed. In recent years, several studies have proposed solutions that leverage decentralized authentication mechanisms, consensus protocols, layered key

 ISBN: 978-1-68558-051-3

management approaches, and privacy-enhancing technologies.

The article "Self-Sovereign Identity Systems Evaluation Framework" [5] proposes a framework to evaluate self-sovereign identity (SSI) systems based on four dimensions: technical, functional, non-functional, and organizational. The technical dimension assesses the system's architecture, security, privacy, and interoperability. The functional dimension evaluates the system's features and capabilities, such as identity issuance, management, and verification. The non-functional dimension assesses the system's performance, usability, accessibility, and user's experience. Finally, the organizational dimension evaluates the system's governance, legal compliance, and economic sustainability. The proposed framework provides a comprehensive and structured approach to evaluating SSI systems, enabling a systematic comparison of different SSI solutions, and facilitating their adoption and implementation.

The DID solutions are often built on existing blockchains , such as Ethereum, Bitcoin, and others. While this approach provides a high degree of decentralization and security, it can also lead to scalability issues due to the limited processing capacity of these blockchains.

A. *uPort*

uPort [6] is built on the Ethereum blockchain and is available as open-source software. The primary goal of uPort is to provide DID services for all users. To create an identity, a user can use the dedicated mobile application provided by uPort, which stores all their identity data, including private keys for signing and sharing claims.

Once an identity has been created, two smart contracts named "controller" and "proxy" are automatically deployed onto the Ethereum blockchain. These contracts serve as the backbone for managing and controlling the user's identity and provide a secure and decentralized way of verifying identity and sharing data.

The uPort project has divided into Serto and Veramo, both dedicated to decentralizing the internet and restoring data control to individuals.

B. *Sovrin*

Sovrin [7] is a DID network built on Hyperledger Indy, which uses a "trust anchor" model and a modified Practical Byzantine Fault Tolerance consensus algorithm to establish the authenticity of verifiable credentials. Sovrin offers an SDK and APIs for easy integration and has a transparent governance model overseen by the Sovrin Foundation.

C. *EverID*

EverID [8] focuses on meeting the identity and the needs of developing countries for data management .The platform leverages blockchain technology, biometrics, and mobile devices to create a secure and verifiable identity for individuals in underserved communities. With a biometric verification system and mobile wallet feature, EverID aims to provide individuals with access to essential services and participation in the global economy. Built on the Komodo platform, EverID is highly scalable and provides fast and secure transactions through its hybrid consensus mechanism.

III. DID INFRASTRUCTURE OR DID PUBLIC KEY INFRASTRUCTURE

We need to introduce some definitions for the coming few sections, so let's gather the various definitions we need into one place:

- DID is a unique identifier; in our implementation it is generated from the public key generated by the DID issuer.
- The DID issuer generates the DID, writes the mapping of DID and DID Doc to Data registry. The DID issuer digitally signs any data and gives credential or VC to the Holder.
- Data Registry is the storage of the mapping of DID, and DID Doc, in our case, is a Blockchain.
- DID Doc is the DID documents which contain information associated with a DID. They typically express verification methods, for instance cryptographic public keys [9].
- DID Resolver resolves the DID-to-DID Doc from the Data Registry.
- A VC is a verifiable credential like a passport which serves to attest specific information about an entity associated with a specific DID. VCs can be created by any entity of the system and issued to the holder of the VC.

During a decentralized transaction, a VC Holder presents their VC to another entity, the verifier which is an entity, or an individual can verify the data of the VC as presented by the holder (Fig. 1).

The verifier can cryptographically check if the VC is connected with the Issuer and the Holder.

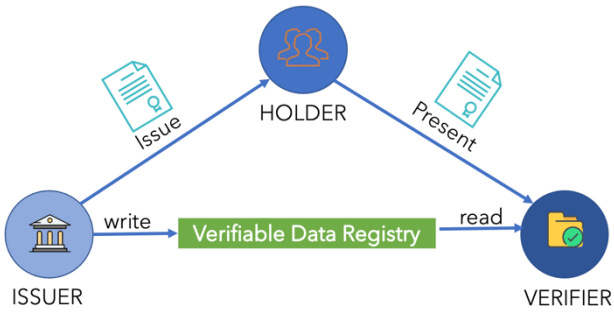

Figure 1. DID Architecture.

In the next section, we will discuss the DID Document and its role in addressing one of the major challenges associated with key rotation.

IV. DID DOCUMENT AND KEY ROTATION ISSUE

DID document is a type of metadata that contains information about a specific DID. This includes the public

keys that are associated with the DID. The DID document is often stored on a decentralized ledger called a Verifiable Data Registry (Fig. 1) or in a distributed file system and it is used to enable secure and decentralized identity management.

In order to maintain the security of a DID, it is often necessary to rotate its public key(s) periodically. This can be necessary in cases of compromise or to comply with best practices. When a public key is rotated, it is important to update the DID document to reflect the new key(s) associated with the DID.

To update the DID document with new public key(s), the DID owner can create and sign a new version of the document with their private key. This new version of the DID document can then be published to the decentralized ledger or distributed file system where the previous version was stored (Figure 2). This process allows the new public key(s) associated with the DID to be made available for use.

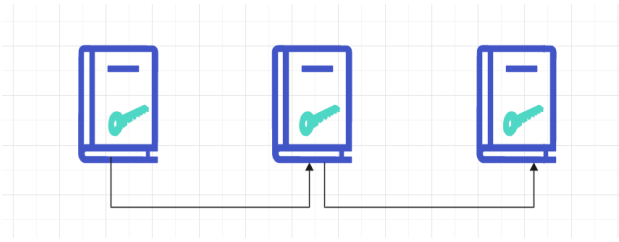

Figure 2. Multiple Version Chained of DID Document.

It is important to note that the process of public key rotation may vary depending on the specific DID system and implementation being used. Nevertheless, maintaining the security of DIDs is critical for ensuring the integrity of decentralized systems and their ability to support secure and DID management.

V. Our Contributions

Using a custom-built blockchain solution and the Substrate framework, our DID architecture addresses scalability challenges, ensuring a robust and high-performing ecosystem.

A. Implementation Using Substrate Framework

Our DID architecture utilizes a custom-built blockchain solution to overcome scalability issues present in existing blockchain platforms like Ethereum. We have chosen to employ the highly flexible and modular Substrate framework to enable tailoring of our DID implementation to our specific use case requirements. This approach offers the necessary scalability [10] and performance characteristics to facilitate a robust and reliable DID ecosystem. Substrate "enables developers to quickly and easily build future proof blockchains optimized for any use case" [11].

We will build our DID system on top of this framework because "Substrate takes the hard work out of blockchain development without imposing limits often found in other frameworks, it allows development teams to quickly build blockchains based on academically-researched and field-tested code that have proven their worth on many live networks worth billions of dollars" [12].

In this paper, we will show you how to create a custom pallet which is the key concept of substrate Framework. It will contain the logic code for our blockchain application. These Pallets are built with libraries called Frame, as well as the Rust programming language. Frame includes functions that make it very simple to build our application logic.

We need three components to build our DID system; the first is the DID pallet. The second is the VCS pallet, and the last one is the attestation pallet. The DID pallet is responsible for the DID creation and its storage on the blockchain.

The DID pallet is a critical component of our decentralized systems. It enables the creation, management, and control of Decentralized Identifiers (DIDs). Specifically, the DID pallet provides three core operations: create, update, and revoke, that allow DID owners to manage their identities within the blockchain system. By leveraging blockchain technology, DIDs are stored in a decentralized and tamper-proof manner, ensuring their immutability and high level of security.

In addition to the DID itself, the DID pallet also stores metadata related to the identifier. While some of this metadata is stored directly on the blockchain, other types of metadata will be stored on external databases to optimize storage requirements. For example, entity-related information, like the name and the logo of the DID owner may be stored on external databases.

This hybrid approach to DID management enables users to leverage the benefits of both blockchain and external databases to store and manage their identity-related metadata. It also provides a scalable solution regarding the performance issue.

The DID pallet at this level contains DID creation function called DID_create (1) and one struct named Details which store the details about DID on the blockchain.

(1) pub fn did_create(origin: OriginFor<T>, signing_key: T::SigningKey, boxing_key: T::BoxingKey, did_doc_ref: Option<Vec<u8>>) -> DispatchResult {}

The VCS pallet is responsible for the creation of Verifiable Credentials and revoke. The VCS is hashed and stored on the blockchain with the vcs_create function (2).

(2) pub fn vcs_create(origin: OriginFor<T>, hash:T::Hash) -> DispatchResult {}

Finally, the attestation pallet, which is responsible for the VCS trust (3), in this step is a trusted entity. For example, like a university will trust any data like a certification or a license, or any type of data regarding the DID and a certain claim.

(3) pub fn attestation_create(origin: OriginFor<T>, claim_vcs:T::Hash,vcs_hash:T::Hash) -> DispatchResult {}

 ISBN: 978-1-68558-051-3

B. *Distributed Real Time Layer*

Our DID architecture includes a distributed microservices layer (Figure 3) to manage DID metadata, for example VCS metadata, DID metadata, and Issuer metadata. This layer enables instant DID document resolution without requiring blockchain requests. Any changes to the Public Key Infrastructure are immediately reported to this layer via an Event System, ensuring that all metadata remains up-to-date and accurate.

Every subject {ISSUER,HOLDER,VERIFIER} in our scenario will receive a DID (decentralized identifier) that conforms to a particular format according to the W3C standard [4].

As an illustration, consider Alice's decentralized identifier, which can be constructed using her public key in the following manner:

did:dido:{Subkey(5GrwvaEF5zXb26Fz9rcQpDWS57CtERHpNehXCPcNoHGKutQY)}

Here, the "Subkey" function generates a public key derived from the Root public key.

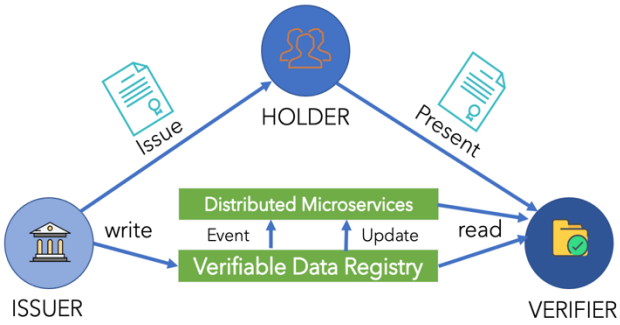

Figure 3. New Architecture Adding a Real Time Layer.

In our system architecture, a trusted entity, such as a university will possess a VC that they can use to sign and distribute to a particular HOLDER. This cryptographic proof will be recorded on our blockchain as an attestation.

The HOLDER will provide evidence to a VERIFIER, for example by displaying a QR code.

The VERIFIER will verify the evidence provided by querying our decentralized microservices layer.

VI. Conclusion

The issuer, holder, and verifier are key components of a DID system. Our work aims to create a DID system that enables secure and privacy-preserving interactions between issuers, holders, and verifiers by connecting the pallets responsible for managing the different components of the system, as shown in Figure 3. By establishing clear and well-defined links between the pallets, the system can work together efficiently and securely. The linking of pallets is crucial for ensuring the reliable exchange of identity information and creating a robust and privacy-preserving SSI ecosystem that can be trusted by all parties involved.

We are incorporating an additional microservices layer to manage the scalability issues that arises from using blockchain technology. Our blockchain serves as a secure ledger, while the microservices layer serves as a cache.

At this stage, we conduct tests using multiple approaches, extending the runtime level, client level, and off-chain level. We also attempt to request changes in the blockchain core when they arise.

This approach will enable us to develop a truly decentralized identity system that can handle a large volume of requests.

References

[1] ISO/IEC JTC 1. (2014). Internet of things: A review of frameworks, standards and action plans.

[2] Oscar Avellaneda, et al. “Decentralized Identity: Where Did It Come From and Where Is It Going?”. [Online]. Available from: https://doi.org/10.1109/MCOMSTD.2019.9031542 December, 2019

[3] U. Feige, A. Fiat, and A Shamir “Zero-Knowledge Proofs of Identity”, in Journal of Cryptology. [Online]. Available from: https://link.springer.com/content/pdf/10.1007/BF02351717.pdf

[4] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, C. Allen "Decentralized Identifiers (DIDs) v1.0 Core architecture, data model, and representations".[Online]. Available from : https://www.w3.org/TR/did-core/

[5] A. Satybaldy and M. Nowostawski "Self-Sovereign Identity Systems: Evaluation Framework".[Online]. Available from: https://www.researchgate.net/publication/339836401_Self-Sovereign_Identity_Systems_Evaluation_Framework March 2020

[6] O. Dib and K. Toumi, "Decentralized Identity Systems: Architecture, Challenges, Solutions and Future Directions".".[Online]. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3785452 March 2021

[7] L. Stockburger, G. Kokosioulis, A Mukkamala, R. R. Mukkamala, and M Avital, "Blockchain-enabled decentralized identity management: The case of self-sovereign identity in public transportation".".[Online]. Available from: https://www.sciencedirect.com/science/article/pii/S2096720921000099 June 2021

[8] B. Reid and B. Witteman, "EverID WHITEPAPER" https://neironix.io/documents/whitepaper/6176/EverID_Whitepaper_v1.0.2_July2018.pdf. ".[Online]. Available from : https://neironix.io/documents/whitepaper/6176/EverID_Whitepaper_v1.0.2_July2018.pdf July 2018

[9] W3C Recommendation, Decentralized Identifiers (DIDs) v1.0 Core architecture, data model, and representations W3C Recommendation, https://www.w3.org/TR/did-core/#dfn-did-documents

[10] A. I. Sanka and R. C. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research".[Online]. Available from: https://www.sciencedirect.com/science/article/abs/pii/S1084804521002307 December 2021

[11] Gavin Wood, et al. "Substrate Framework" https://docs.substrate.io/ June 2023

[12] A. El akroudi, " Data Trust Protocol" https://github.com/lsroudi/DataTrustProtocol/ June 2023