Systematization of Knowledge for Decentralized Identities & Verifiable Credentials —Phase 1 of Research for Lido DAO: Complete

Nethermind

Follow

Nethermind.eth

--

Listen

Share

The following is a continuation of a previous post,"A Path To Permissionless Liquid Staking," which is an overview of Lido DAO and decentralized liquid staking. In that post, the Nethermind Cryptography Research team outlined the problems of building such a protocol and put forward a collaboration proposal on how to go about solving them.

In this blog post, we are proud to share with the community the outputs from Phase 1 of this work proposal, which produced a systematization of knowledge on the fields of Decentralized Identity and Verifiable Credentials.

Table of Contents

Introduction

In this systematization of knowledge, we examine the current innovations and approaches to the field of decentralized identity

(also self-sovereign identity

), as well as its relevant foundations. In the words of the Ethereum foundation decentralized identity is "the idea that identity-related information should be self-controlled, private, and portable." Accordingly, an introductory article by Dock Labs defines decentralized identity as "a type of identity management that allows people to control their own digital identity without depending on a specific service provider."

Users can construct their decentralized identities from various data sources: interactions on a blockchain, information gathered from major social networks or centralized websites, and even an ID issued by a government or an educational institution. Self-sovereign identity implementations then store this data (encrypted or not) in a document on a distributed ledger, such as a blockchain. This document is associated with a set of keys that the user controls and can be used to assert ownership of the data.

Furthermore, to facilitate access and communication, the document is tagged with a unique identifier from which a URL to the document can be derived. This ID is known as a decentralized identifier

(DID — not to confuse with decentralized identity, which may use DIDs).

To utilize this identity, users can request or create verifiable credentials (VCs). VCs are cryptographically-verifiable claims to a third party about the data conforming to their identity or properties. Verifiable credentials give the user control of which information is shown and provide information requesters tools to combat forgery and fraud.

Description of the work

Understanding current solutions in the landscape of self-sovereign identity is relevant to Lido's aim of increasing the quality of its validator set in a distributed fashion. A mechanism that decentralizes Lido's validator set will require robust identity management and authentication methods. To this end, Lido DAO agreed to fund Nethermind in delivering a systematization of knowledge for decentralized identities and verifiable credentials. This will be the first step towards an informed design of a mechanism that fulfills Lido's goals.

During this project, we investigated what the state-of-the-art is, what solutions could be used in practice, and how to use them. More specifically, after a preliminary examination of over 100 papers and protocols related to the fields mentioned earlier, we selected and studied a collection of 70 of them and placed the results in the following database: Decentralized Identity and Verifiable Credential systems. The papers in the database have been analyzed as follows:

- A summary note was prepared for each paper, accessed by clicking on each paper's title.
- Papers were rated from 1 to 5 according to their quality and originality. This is reflected in the "quality score" column.
- Papers were rated according to their relevance to Lido's mechanism design problem. This is reflected in the "relevance score" column.

Decentralized Identities

Let us now give a more detailed picture of Decentralized Identities, including their advantages and the mechanisms (such as DIDs and VCs) that govern them. As previously stated, Decentralized Identity

is a type of identity management system where users are the owners of their private data that they store in a digital wallet, with the option to prove claims about these data to potential verifiers. "Decentralized Identity" is used interchangeably with the term "Self-Sovereign Identity" (SSI).

Following the analysis of different types of identity management systems, we will discuss the components of Decentralized Identities and, based on an introductory article by Dock Labs, illustrate the advantages of Decentralized Identity compared to other systems.

- In a Centralized Identity Management system,

the users create a username and a password for each service they want to access. The main disadvantages of this type are that (i) the users should store or remember multiple sets of usernames/passwords, (ii) there is a single point of failure for data breaches, as there is one server that stores sensitive data that belongs to many users. A few examples of such incidents are described in the paper "Digital Identities and Verifiable Credentials".

- In a Federated Identity Management system,

users can access multiple applications using a single credential set. An example is when the user signs in to multiple applications using their Google or Facebook account. This solves the first disadvantage of the Centralized Identity Management system, but it comes at a cost: there is no

[unlinkability

](https://github.com/decentralized-identity/snark-credentials/blob/master/whitepaper.pdf).

This means that if multiple applications collude, it is possible to detect if the same user has registered to all of them using the same account.

- In a Decentralized Identity Management system

, these disadvantages are mitigated. Since there is no single server where the data of multiple users are stored, there is no single point of failure — every user stores their private data (or the keys to decrypt it) in a digital wallet. In addition, most applications of DI that use cryptographic tools — such as Verifiable Credentials

and Zero-knowledge Proofs —

provide some level of unlinkability

and allow the users to prove claims about themselves without revealing their private data. For example, users can prove that they are above 18 without revealing their actual date of birth.

Components & mechanisms of Decentralized Identity

Typically, an implementation of a decentralized identity system involves the following components:

Decentralized Identifiers (DIDs):

a cryptographically verifiable identifier created and managed by the user. The user can manage this identifier by using a private control key. Specifically, the DID is a URI (uniform resource identifier) that has a recommended structure specified by the World Wide Web Consortium (W3C):

Figure 1: Parts of a decentralized identifier. Source:

[“Decentralized Identifiers (DIDs) v1.0 W3C recommendation”

](https://www.w3.org/TR/did-core/#a-simple-example).

[Copyright

](https://www.w3.org/Consortium/Legal/ipr-notice#Copyright)2022

[W3C

](https://www.w3.org/)(

[MIT

](https://www.csail.mit.edu/),

[ERCIM

](https://www.ercim.eu/),

[Keio

](https://www.keio.ac.jp/),

[Beihang

](https://ev.buaa.edu.cn/)). W3C

[liability

](https://www.w3.org/Consortium/Legal/ipr-notice#Legal_Disclaimer),

[trademark

](https://www.w3.org/Consortium/Legal/ipr-notice#W3C_Trademarks), and

[permissive document license

](https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document)rules apply.

In the DID above, we see the following components:

1. The DID URI scheme identifier, which for this standard is always “did

”.

1. The identifier for the DID method

refers to the particular implementation of the DID standard being used. Examples include “ethr

”, “tz

”, “cosmos

”, “dock

”, among many others. A full list of existing implementations (curated and maintained by W3C) can be foundhere.

1. The DID method-specific identifier makes the DID unique and specific to a particular subject.

DID document:

It contains information about the Decentralized Identity of the user, such as attributes or authentication methods. It can be updated by the user via their private control key.

According to the W3C standard, it corresponds to a JSON, likethe following example:

Example 1: A simple DID document. Source:

[“Decentralized Identifiers (DIDs) v1.0 W3C recommendation”

](https://www.w3.org/TR/did-core/#a-simple-example).

[Copyright

](https://www.w3.org/Consortium/Legal/ipr-notice#Copyright)2022

[W3C

](https://www.w3.org/)(

[MIT

](https://www.csail.mit.edu/),

[ERCIM

](https://www.ercim.eu/),

[Keio

](https://www.keio.ac.jp/),

[Beihang

](https://ev.buaa.edu.cn/)). W3C

[liability

](https://www.w3.org/Consortium/Legal/ipr-notice#Legal_Disclaimer),

[trademark

](https://www.w3.org/Consortium/Legal/ipr-notice#W3C_Trademarks), and

[permissive document license

](https://www.w3.org/Consortium/Legal/2015/copyright-software-and-document)rules apply.

In the above example, we see how the DID document is bound to the DID via the “id” entry. An “authentication” entry also contains a public key. If the user holds the respective private key and signs messages with it, they can assert ownership of the associated identity.

Verifiable data registry:

Commonly a blockchain, it is used to record the DIDs and DID documents.

Verifiable Credentials (VCs)

: a cryptographically secure, digital version of credentials (paper or digital) that users can present to possible verifiers. This digital version should comply with [Verifiable Credentials Data Model v1.1

](https://www.w3.org/TR/vc-data-model/).

We will explain the mechanisms behind VCs more thoroughly in the next section.

Decentralized Identity Wallet

: an application that allows users to create their DIDs and the associated DID documents, and store related keys, data, and verifiable credentials.

The following diagram on decentralized identifiers, provided by the W3C, may shed light on how all the pieces fit together:

Figure 2: Elements of DIDs and their functions. Source:

[“Decentralized Identifiers (DIDs) v1.0 W3C recommendation”

](https://www.w3.org/TR/did-core/#architecture-overview).

For clarification, let us note that:

- “DID subject” refers to the entity identified by a DID. It need not refer to persons or organizations — it can also describe groups, things, or concepts.
- “DID controller” refers to the entity controlling the cryptographic keys bound to the DID document. In the case of a decentralized identity associated with a single person, the DID subject and controller can be the same. In the case of an organization, the DID controller can be one of its members.

Verifiable credentials

The above overview characterizes how decentralized identities are stored and accessed via DIDs. However, we can construct a higher level of functionality — that of authentication

. When we think of how identities are used and proven in the physical world, we note that the subject normally shows a credential — issued

by an authorized party — that also certifies attributes about themselves (e.g., date of birth). This credential’s authenticity is verified

by a third party at that time, accepting the subject’s claims about their identity if the credential is deemed legitimate.

The same three-party paradigm — with a user (or holder/subject), an issuer, and a verifier — can be employed in the world of decentralized identities and gives rise to the concept of verifiable credentials

. In a nutshell, verifiable credentials

work as follows:

1. Both the issuer and the holder create a DID; each holds a private key to manage their DID.
2. The issuer signs a claim for the holder (e.g., the issuer is a university and signs a degree) using its private key; the claim is bound to the holder’s public key. This will constitute a verifiable credential.
3. The holder holds this credential in a digital wallet

and presents

it to the verifier.

1. The verifier checks which DID corresponds to the issuer and verifies the signature.

Once again, W3C provides us with a diagram of the entire process:

Figure 3: Verifiable credentials. Here, “schemas” refer to the structure and syntax of the relevant credentials and identifiers

. Source

: “

[Verifiable Credentials Data Model v1.1

](https://www.w3.org/TR/vc-data-model/#ecosystem-overview)”.

We note that the “presentation” in step 3 may take on different levels of complexity, depending on the privacy the holder wishes to maintain. In the simplest scenario, the holder shows the verifiable credential to the verifier. More generally, the holder can construct a verifiable presentation,

a tamper-evident presentation derived from the data inside the credential, authenticated by cryptography. Examples and use cases include:

- The holder wants to share a VC of their university diploma without disclosing their GPA. Then a verifiable presentation is created without including the GPA attribute, which the original credential did include.
- The holder wants to prove they are over 21 years of age without disclosing their exact age. Then a zero-knowledge proof of this claim is constructed from the original credential and used as a verifiable presentation.

Transferring identity data to Web3

We now focus on the data sources that can be used to construct a decentralized identity. Among these, we’ve mentioned how we may use inputs from Web2 to achieve this. Examples include a reputation score on Reddit, the number of stars on repositories created by a user on GitHub, and even the existence of an open TLS connection with a government website, which the user presents as evidence of citizenship of a given country.

In the Web3/blockchain context, one reason to be interested in bridging Web2 data to Web3 is that it may provide some degree of resistance to Sybil attack — that is, the ability for a malicious entity on a decentralized protocol to create an arbitrary number of identities and gain disproportionate influence over it. If, for instance, we require a decentralized identity to bridge a reputation score from Web2 that is valuable enough, then this mechanism can complicate the creation of numerous identities by a single entity.

Besides Web2, there are alternative sources of off-chain information that one may attempt to transfer to Web3 to create an identity. Among these, we count government IDs, institutional credentials, and even biometrics. The goal behind using this data remains the same — using valuable sources to facilitate the identification and obfuscate the creation of Sybils.

The main technical challenge when following this approach is: How do we pull this data in a verifiable way so that the system is not likely to be exploited?

For example: are oracles to be used? If so, what incentivization mechanism is used to enforce their honesty?

Due to their potential for building Sybil-resistant solutions (and for making decentralized identities more meaningful as a whole), we have paid special attention to implementations that explore transferring identity data to Web3 in a trustless or verifiable way.

Selected papers

Finally, we highlight a selection of database papers — rated as highly relevant and recommended as a starting point for reading.

Classical papers

This is one of the first papers discussing the problem of Sybil nodes in a network. It outlines how powerful Sybil nodes could be if we require the network nodes to control a given amount of resources for identity authentication. Furthermore, it shows that malicious nodes can easily spin off an unlimited number of nodes if the only requirement to access the network is the possession of resources checked on an irregular basis.

- The Sybil Attack

The following paper(s) showed how to build anonymous credentials. There are no more distinguished researchers in creating a cryptographic toolbox for credentials than Camenisch and Lysyanskaya, who, over a number of years, introduced several crucial ideas for dealing with credentials.

- Work of Camenisch and Lysyanskaya on Anonymous Credentials

Decentralized Identity & Verifiable Credentials

Next, we present some papers that aim to standardize how decentralized identifiers and verifiable credentials look. This effort, led mostly by W3C, is tremendously important since the lack of standardization may redeem identifiers useless. If we want people to use verifiable credentials, we must ensure they are as widely recognized as possible.

- [W3C's Decentralized Identifiers data model v1.0

]
(https://nethermindeth.github.io/lido_phase_1/Database%2011b9b21206a8466191f8587fb73edf58/W3C%E2%80%99s%20Decentralized%20Identifiers%20data%20model%20v1%200%20fd885cccdf81

- [W3C's Verifiable Credentials Data Model v1.1

]
(https://nethermindeth.github.io/lido_phase_1/Database%2011b9b21206a8466191f8587fb73edf58/W3C%E2%80%99s%20Verifiable%20Credentials%20Data%20Model%20v1%201%2068fcef3902ac44

The next piece comes from Vitalik et al. The authors discuss tokens bound to Web3 identities. Soulbond tokens can potentially change the landscape of Web3 and make it more identity-oriented.

- [Decentralized Society: Finding Web3's Soul (Soulbound tokens)]
(https://nethermindeth.github.io/lido_phase_1/Database%2011b9b21206a8466191f8587fb73edf58/Decentralized%20Society%20Finding%20Web3%E2%80%99s%20Soul%20(Soulbou%20d1a47

The problem with most classical papers on verifiable credentials is the centralized issuer which does not fully embrace the decentralized potential of Web3. The following article deals with this issue by showing how to decentralize that party.

- Decentralized Anonymous Credentials

Sometimes the credentials need to be revoked. Most VC schemes deal with revocation as follows: the system keeps a list of all revoked certificates, and the user needs to show that their credentials are not on that list. The following paper explains how to accelerate this process, and without the need to interact with the user — the verifier can check on its own whether the issuer revoked the credentials.

- Zero-knowledge credentials with deferred revocation checks

Web2 to Web3 data

The next set of papers is of immense importance. They deal with the problem of getting Web2 data to Web3. There is a lot of data in Web2, yet most of it is hidden behind a TLS protocol. In other words, it is accessible to users, but those cannot show it in a verifiable manner to other parties. This is because the TLS protocol establishes a symmetric cryptography-based connection between the user and the server. The papers aim to solve that problem by allowing the user to include a verifier in the communication with a server who could verify the correctness of data.

- CanDiD
- DECO
- TLSNotary

Project implementations

Even with a set of magnificent papers, Web3 wouldn't progress without the implementation effort of several teams and companies. Here we name only a few, but there are more exciting projects to follow in our database.

- Polygon Id
- Sismo
- [EBSI (joint initiative from the European Commission and the European Blockchain Partnership)]
(https://nethermindeth.github.io/lido_phase_1/Database%2011b9b21206a8466191f8587fb73edf58/EBSI%20(joint%20initiative%20from%20the%20European%20Commissio%20ba190ea5d9c64af
- Interep (by PSE)
- Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers