Using Self-Sovereign-Identity principles to prove your worth in Decentralized Autonomous Organizations

Vid Keršič[1,*], Andraž Vrečko[1], Urban Vidovič[1], Martin Domajnko[1] and Muhamed Turkanović[1]

[1]*Faculty of electrical engineering and computer science, University of Maribor, Institute of informatics, Blockchain Lab:UM, Maribor, Slovenia*

Abstract

Decentralized autonomous organizations (DAO) have many use cases and are becoming increasingly more popular in recent years. The paper analyses the disadvantages and problems of current DAOs (i.e., the Plutocracy problem) and presents a solution, in which the existing implementations of DAO are combined with the Self-Sovereign Identity (SSI) ecosystem to solve those issues. The solution is presented in the form of an extension to the crypto wallet MetaMask, using Snaps, which enables DAO users to create SSI-based identifiers (DID) and manage verifiable credentials (VC), which are later on used as a voting ticket in a DAO. We validate the solution by presenting the prototype of the SSI Snap and demonstrating its usage on the Snapshot decentralized voting system.

Keywords

decentralized, DAO, decentralized autonomous organization, SSI, self-sovereign identity, verifiable credential, MetaMask, proof of concept, Veramo, Snaps

1. Introduction

Decentralized autonomous organizations (DAOs)[1] are slowly emerging as a new digital and management structure without a typical real-world organizational hierarchy, hence the decentralized prefix. Since the DAOs are running on public permissionless blockchain networks [2], the barrier to joining and contributing to an organization (and being paid for the work) is more accessible than in a typical setting in the real world. While the mentioned structure provides many new opportunities and advantages, it comes with several problems. One of the problems is the so-called Plutocracy Problem, described by Serto [3] and Vitalik Buterin [4, 5]. This problem affects tokens-based (ERC20, ERC721, or ERC1155) DAOs, which are currently the dominant form of DAOs, by a wide margin.

In a plutocracy, the wealthiest members of the organization/society are the people with the most power and influence, regardless of their expertise on the given topic. Adding to that, most

SQAMIA 2022: Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, September 11–14, 2022, Novi Sad, Serbia

*Corresponding author.

vid.kersic@um.si (V. Keršič); andraz.vrecko@student.um.si (A. Vrečko); urban.vidovic2@um.si (U. Vidovič); martin.domajnko@student.um.si (M. Domajnko); muhamed.turkanovic@um.si (M. Turkanović)

https://ii.feri.um.si/en/person/vid-kersic-2/ (V. Keršič); https://ii.feri.um.si/en/person/muhamed-turkanovic-2/ (M. Turkanović)

0000-0002-7340-7501 (V. Keršič); 0000-0002-5079-5468 (M. Turkanović)

proposals are decided by only a few members who control the majority of tokens. Therefore, most token-based DAOs are not decentralized and give their holders a false sense of security, leading them to believe that they hold any weight in the decision-making process, while it actually is in the hands of few. The problem lies in the possibility of obtaining and transferring the reputation that represents your worth in the DAO. The plutocracy problem can never be solved if the reputation can be bought. Suppose we want to make DAOs more aligned with the web3 ethos and strive for meritocracy, where the power of members is determined based on their skills. In that case, different ways to represent reputation must be implemented.

Imagine a made-up metaverse DAO. There are several decisions that the community can make: changes in the smart contracts [6], connecting and creating new partnerships, designing new worlds, creating new graphical content for the project, etc. Who should be able to vote on the changes in a smart contract code, and who on the future of marketing? The former should be decided by the proficient developers in Solidity [7], while the latter by the digital marketing experts. The question is, how to verify the potential voter's specific skills?

Reputation should be gained through experience, achievements, and other successful contributions to some cause (e.g., DAO). This could be anything from completing a quest on Rabbithole, committing new code to a project, or being active in the DAO communities. A credible third party should then give a reputation to the contributors in the form of proof of contribution or attestation to their skills and knowledge. It's only natural to do that in a digital, cryptographically verifiable, and tamper-proof way. And the best way to do that in a structural and reusable way is to use verifiable credentials (VC) and verifiable presentations (VP), which are one of the main concepts behind the Self-Sovereign Identity (SSI) paradigm [8, 9]. But specialized software must first be developed to bring together web3 structures (DAOs), where data resides on public blockchains or other networks, and SSI principles, such as offline and local private data.

1.1. Aim and contribution

The aim of our research was to solve the above-presented challenge of plutocracy. The main idea was to enable DAOs to use a fairer way of operating and decision calling, which however should still be based on digital and decentralized principles and enable secure verification of voters' skills and knowledge. For this, we analyzed the possibility of using SSI principles.

As such, we had to enable the integration of SSI and DAOs in a web3 user-friendly way, which would enable seamless binding of the two ecosystems. Currently, there is no straightforward way to operate within SSI and DAOs on the same terms and conditions, i.e., solutions on the market require separate mobile/web applications for blockchain (DAOs) and SSI operations. Therefore, in this research, we define the solution to these challenges in the form of software architecture and components, which would enable seamless integration of the two ecosystems. Our solution is based on the MetaMask wallet, which is one of the most adopted crypto wallets in the market, especially in the Ethereum ecosystem [10]. As such, we leverage the MetaMask's extension, called Snaps, which we designed and implemented in a way that supports the core SSI workflows. To showcase and validate the prototype in the real-world setting, we connect the most popular dApp for DAO governance Snapshot to our SSI Snap [11]. Adding the voting mechanism based on decentralized identifiers (DIDs) and VCs, representing people's expertise and experience, blockchain projects and organizations can target people with specific skills to

decide on the organization's future direction. Our approach enhances the decentralization of DAOs since the currently most adopted token-based voting mechanism is heavily influenced by large shareholders, thus making DAOs centralized and controlled by a small group of people.

The paper has the following structure: Chapter 2 provides an overview of applied technologies, Chapter 3 describes the proposed solution, Chapter 4 showcases the performed validation, and Chapter 5 discusses the advantages and disadvantages of the approach. The paper is concluded in Chapter 6.

2. Preliminaries

Our proposed solution builds upon several concepts and technologies. This chapter provides their definition, description, and role in the solution.

2.1. Decentralized Autonomous Organizations (DAOs)

Decentralized autonomous organizations (DAOs) are community-controlled organizations, where rules are enforced and governed by a computer program, instead of a central government. The computer program is usually represented as a smart contract, which is a program for the automated execution of agreements stored on a (decentralized) blockchain and run on the underlying virtual machines (e.g., EVM) representing the blockchain network. Their main purpose is to provide a way for autonomous, decentralized, and transparent governance of organizations. The first point is achieved by the usage of smart contracts, while the other two points are provided by the underlying blockchain technology on which the smart contracts are run. There are different types of DAOs supporting a wide range of use cases from Protocol DAOs, Investment DAOs, Grant DAOs and Gaming DAOs [12]. DAOs can run on different voting mechanisms, the most popular being token-based single-choice quorum and quadratic voting. In both types of governance, voting power relies on token balances of voters, such as ERC20 or ERC721 tokens. In single-choice voting, voters can vote only for a single choice, and each token represents a voting power of 1. Therefore the voter with a higher balance has more voting power than one with a lower. Quadratic voting enables voters to vote on several choices, with the results being calculated quadratically, giving the number of different voters more influence on the final result than the number of tokens [13].

Snapshot plays a vital role in the decentralized world and DAOs [11]. Snapshot is a decentralized voting platform that provides flexibility and supports various voting mechanisms. It is also user-friendly and does not cost gas, as the voting process is done off-chain and based on decentralized file storage like (IPFS).

2.2. Crypto Wallets

One of the key components for interacting with blockchain technology are crypto wallets. Their primary functionality is secure storage of cryptographic keys, with which we control the blockchain addresses, support transaction signing, and in some cases also data encryption. When compared with digital wallets, the major difference is that crypto wallets are focused on the management of blockchain addresses and thus the control over their coins and tokens,

while digital wallets have a more general purpose, enabling users to control (qualified) digital identities and identifiers, and as such use those for purposes of authentication, digital signing, as well as the collection and management of attestations in the form of digital documents [14, 15]. Crypto wallets also enable users to create several accounts, each with its own key pairs, between which they can easily switch.

2.2.1. MetaMask and MetaMask Snaps

The most popular wallet, MetaMask, introduced Snaps, which makes building plugins for additional functionality possible [16]. MetaMask is a crypto wallet and gateway to blockchain apps, providing a simple interface for users to interact with EVM-based blockchains, sign and send transactions, etc. Snaps make a wide specter of new applications possible. They can enable support for previously unsupported chains like Polkadot, Solana, Bitcoin, etc. They allow dApps to modify MetaMask's state to store and retrieve data, like VCs. They also enable access to the web and the possibility to leverage practically any API and much more. New functionality is only limited by the creativity of developers.

Technically speaking, MetaMask Snaps is a system that allows anyone to expand the capabilities of MetaMask safely. It is a JavaScript program that runs in an isolated, sandboxed environment inside the MetaMask. In addition to the existing MetaMask RPC methods, Snaps can create new RPC methods for websites to call. Unfortunately, that is the only way to interact with the Snaps, as modifying MetaMask UI is not possible (at least at the moment).

Snaps are currently only supported in the MetaMask Flask, a separate desktop browser extension for developers. But it is expected that the Snap system will be integrated into the main MetaMask in the future, with much more information found in their documentation [17].

2.3. Self-Sovereign Identity (SSI)

There are three main components to a digital self-sovereign identity: DIDs, VCs, and VPs.

Decentralized Identifiers, or DIDs in short, are the new type of unique and persistent identifiers (URI) that enable verifiable and decentralized identity [18]. They are entirely controlled by the identity owner and are independent of centralized authorities. Each individual can create as many DIDs as they wish and use each in different contexts to prevent data correlation.

DID Document forms the root record for a DID and is a set of data that describes a DID, including mechanisms, such as public keys and pseudonymous biometrics, that an entity can use to authenticate itself as the DID. While a public key can be obtained from the DID document by anyone, a private key used for proofs and digital signatures is safely stored in the user's wallet. DID Document may also include other attributes or claims describing the entity, such as service endpoint, delegates, etc. These documents are often expressed using JSON-LD.

DIDs are verifiable, their corresponding DID documents are usually stored on a trusted data registry (typically a blockchain) and can be accessed by anybody. There are multiple methods for storing and resolving DIDs. For example, the method did:ethr uses a Smart Contract on Ethereum to store the DID data. Similar to blockchain addresses, DIDs are pseudonymous, however, they offer additional capabilities such as key rotation, delegation, and a way to link a service endpoint (social media account, etc.) to the identity.

However, DIDs are not enough to represent our entire identities as they merely provide a "basket" for them. This basket must be filled with all kinds of data, usually presented in the form of credentials in the real world. Credentials are ubiquitous in our daily lives they take the forms of passports, various licenses, and certificates, ownership of bank accounts, and much more. The problem with credentials is that until recently, there had been no standard ways of representing them organized online.

Verifiable Credentials, or VCs for short, are an open standard for digital credentials to solve this issue [19]. They are digitally signed and can be verified cryptographically, which makes them tamper-proof. VCs work well with data privacy, which goes well with data regulations pushed by the European Union (GDPR) and some other countries.

VCs are interoperable and can use a lightweight Linked Data format. It is an extension of an already successful JSON format that provides a way to include object and data typing, JSON-LD keyword aliasing, creating links via nesting or referencing, and internationalization features (describes how to express data values in different languages). Another format for VCs is (JSON Web Token), a popular internet format for transferring data with digital signatures. Because of that, current SSI tools often provide better support for JWTs.

Verifiable Presentation, or VP for short, expresses data from one or more VCs and is packaged so that the authorship of the data is verifiable. The data in a VP, which is often about the same subject, could have been issued by multiple issuers.

There is a lot of ongoing work on further enhancing data privacy when presenting VCs. This can be done with Selective Disclosure and Zero-Knowledge Proofs (ZKP). Selective disclosure enables generating proofs from only a few attributes of a credential. Using ZKPs, one could prove the necessary condition for the attribute without revealing the actual value. In practice, this means one could prove that they are above the age of 18 without showing their ID card, and third parties would instantly be able to verify that data [20].

Now that we have a base understanding of the individual SSI components, we can look at how they work together.

2.3.1. VC Trust Model

SSI completely changes the paradigm of online data sharing and brings it closer to the physical world. There are three entities in the VC trust model:

- Issuer that issues the credential
- Holder that is the owner and subject of the credential
- Verifier that receives and verifies the credential

As seen in the Fig. 1, the issuer is the entity that issues VC to the holder whom the VC is about. The holder then presents the VC to the verifier, who verifies the validity of the VC and checks if it meets the established criteria. For example, a government issues an ID card in the form of a VC to Alice. Alice is the holder of the VC. Alice wants to go to a concert at a club. Alice has to prove that she is 18+ and does so by presenting a VP, which she generates using her VC. The club then verifies if Alices VC is valid and if she is indeed older than 18. The verification process is based as follows. When the government issues a VC, they add into the VC

their DID and sign the VC with their private key of the coresponding DID. The DID document of the corresponding DID is registered on a blockchain. When the club wants to verify the authenticity and validity of the VC and its proof, they can check the DID and its associated public key on the blockchain to see who issued it without contacting the issuing entity. DIDs enable VCs to be verified anywhere, at any time.

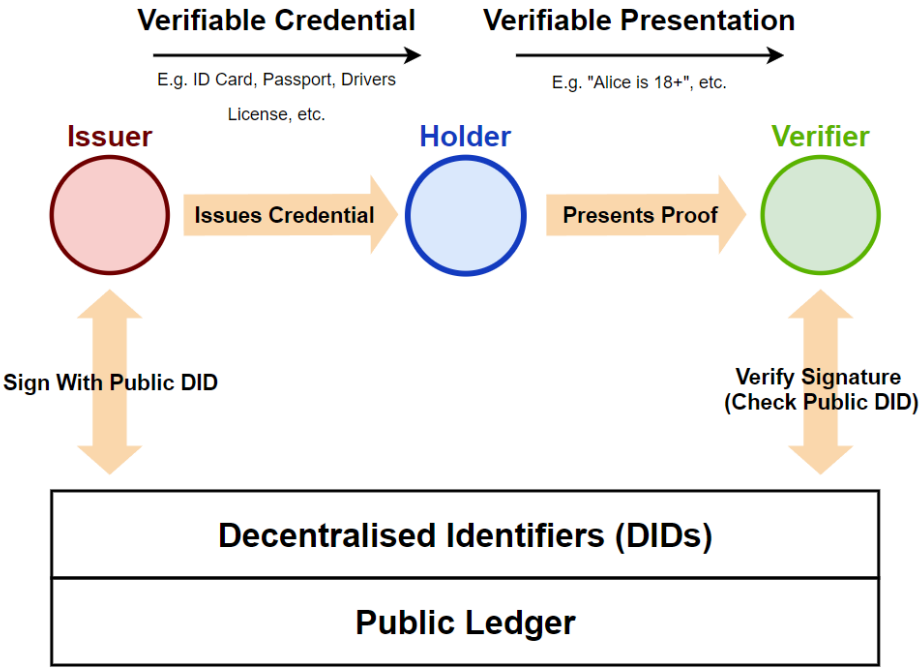

Figure 1: SSI trust model [8].

3. Solution

At the core of our solution is the so-called SSI Snap. A Snap as already mentioned in Chapter 2.2.1, is a plugin-like extension of the popular crypto wallet MetaMask. As shown in Fig. 2, the SSI Snap resides inside the MetaMask wallet. The idea of the Snap is to provide blockchain-based accounts, that are controlled with MetaMask, handling the core functionalities of SSI wallets, i.e., generation and control of DIDs, management of VCs, etc. With such a solution, a possible DAO stakeholder, which currently is only able to participate in voting through the management of blockchain-based tokens, would now be able to vote using the same tools (MetaMask) by passing VPs as the voting ticket. To enable such functionalities, the Snapshot, which is the decentralized voting system for DAOs, also needs to support the management of votes based on

VCs and not just tokens. In this example, the VC-based votes are then validated using Snapshot and Ethereum and stored inside IPFS to make the experience fee-less.

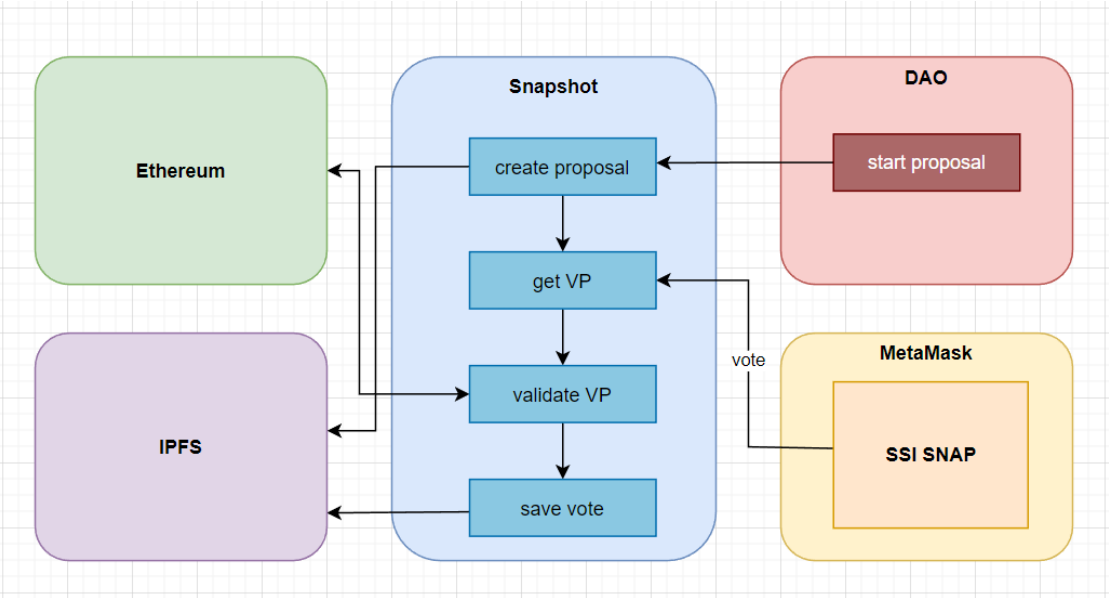

Figure 2: A high level technical overview of the proposed architectural landscape.

3.1. SSI Snap Design

Every user in the system needs to have a unique id and full control over his online identity. As previously mentioned, in the SSI world, this identity is called DID. To work correctly, DIDs require a DID method, which specifies how DIDs and DID documents are created, resolved, updated, and deactivated [18].

There are various DID methods. One of the most popular methods is called did:ethr. This method uses Ethereum addresses as fully self-managed DIDs. In other words, every Ethereum account is a DID (DIDs are Ethereum addresses with a "did:ethr::" prefix). Ethereum accounts in MetaMask, used daily by millions, are essentially DIDs. What is missing is the functionality to use them and leverage their potential correctly.

Our demo will use SSI Snap to store Solidity Course Completion VC, proving that the user controlling the MetaMask account has completed a Solidity course and is qualified to vote on Snapshot governance proposals.

In order for the SSI Snap to handle functionalities related to DIDs, VCs, and VPs, we decided to use a Veramo framework. Veramo is a performant and modular API for Verifiable Data and SSI [21]. Essentially it's a client that allows the creation and management of DIDs, VCs, and VPs and makes developers' lives working with them much easier.

Veramo is used to generate and store DIDs and additional keypairs. The team behind Veramo implemented plugins called DIDManager, KeyManager, and PrivateKeyManager to do precisely that. However, these plugins do not come with a way to store data inside the MetaMask State.

Luckily, due to the extendable nature of these plugins, it made it easier to implement a custom datastore plugin that allows the Managers to store data inside the MetaMask State.

Veramo is also used to verify and store VCs and generate VPs. Unfortunately, Veramo does not have a VCManager plugin. Nevertheless, we developed the VCManager plugin with an additional datastore plugin to save VCs in the MetaMask state.

The ability to create additional datastore plugins is also great for future implementations. In the future, we plan to implement additional ways to store data, starting with storing everything (of course encrypted) in a cloud. This will make syncing with other MetaMask wallets possible. Having multiple ways of storing data and quickly changing between them will create a better user experience. The final SSI Snap architecture is shown in Fig. 3.

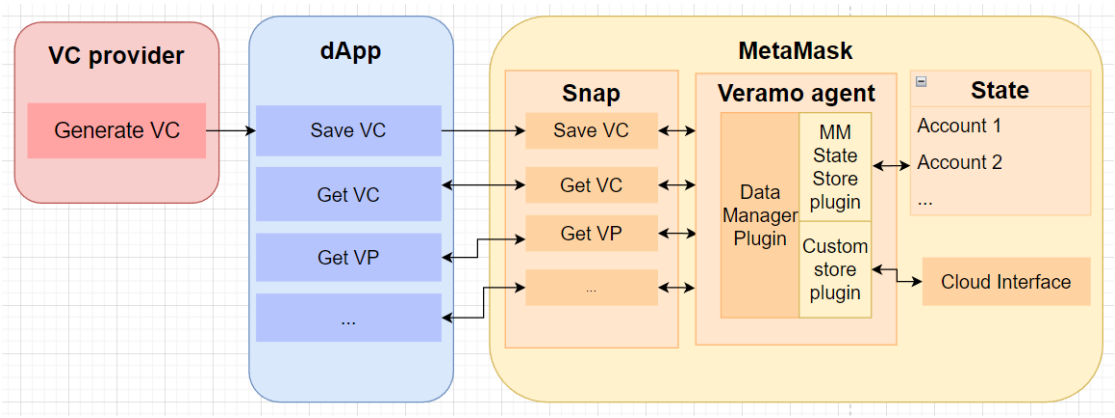

Figure 3: The architecture of the proposed SSI Snap.

To maintain as much security as possible, we have decided not to expose private keys from existing MetaMask accounts but to create and use an additional DID (Ethereum account) exclusively for generating VPs. Private keys are needed for digitally signing VPs with Veramo, since core MetaMask RPC methods do not offer a way to sign them properly. Essentially, this means that a separate DID is generated for every MetaMask account that wants to store and manage VCs. This DID lies in the MetaMask state and is only used for generating VPs.

However, a DID can only use its own VCs, and since VCs are issued to a MetaMask account DID, the newly generated DID can't use them. To make things right, we authorize the new DID to use VCs explicitly. Thus, the DID has to be registered as a delegate to the DID document of a MetaMask's account. We hope this won't be necessary in the future as MetaMask is constantly updated with new features and tighter Snaps integrations are on the horizon.

4. Validation

To showcase the workflow of the SSI Snap, we have designed a proof of concept (PoC) and developed a demo platform. The PoC is designed with the Ethereum blockchain platform. One of the main reasons for this is the fact that: (1) it's one of the decentralized and public permissionless blockchain platforms, (2) the most popular and most commonly used blockchain

platform for smart contracts, (3) designed and tested did:ethr method, (4) plenty of already established frameworks, including various SSI & DID frameworks and, (5) DID Documents do not need to be changed often (or even never in some cases). A more detailed description of the prototype is accessible on the GitHub repository (Online Resources).

In PoC demo, a user will install and approve the SSI Snap, add a delegate to the DID Document of the selected MetaMask account, get and store a VC after completing a straightforward course and display the VC on the profile page.

The demo can be tested freely (Online Resources). To follow it, you need to use MetaMask Flask (version >10.9.2) and have some ETH on Rinkeby testnet.

To start using the platform, the user needs to connect to the platform using the MetaMask crypto wallet. After the connection is established, the user gets a prompt to install and Connect to the SSI Snap. The user needs to give it specific permissions for the Snap to work. Besides the standard permissions, SSI Snap also needs permission to manage the MetaMask state. When the user has successfully installed the Snap and connected to the platform, they can start the Solidity course (Fig. 4). First, the SSI Snap needs to initialize for the current account.

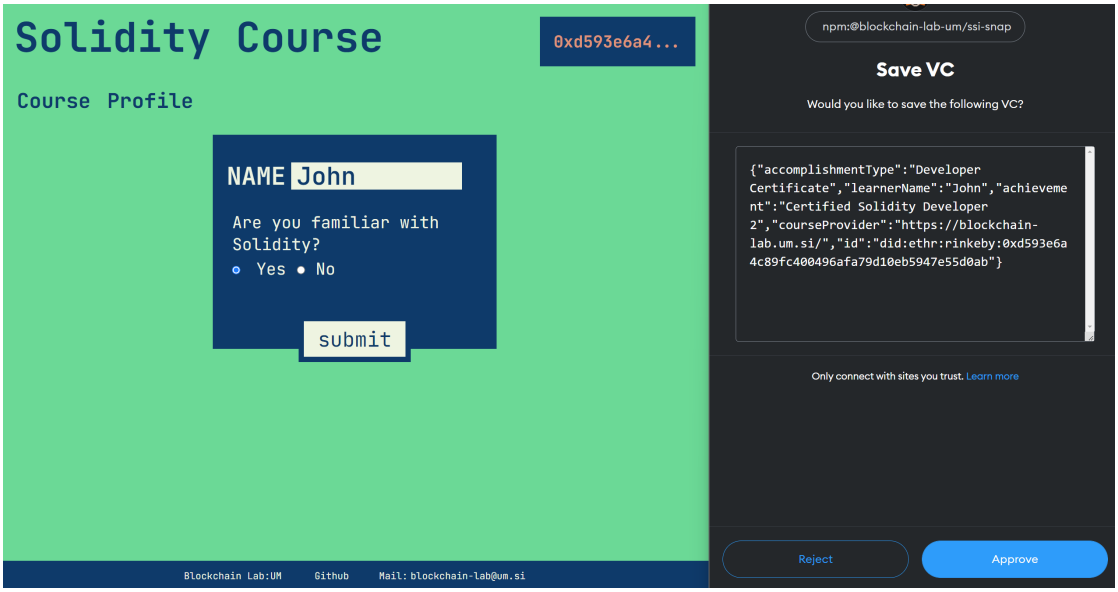

Figure 4: A mockup of the Solidity Course, where the user can get a VC after passing a course.

Adding a delegate will cost the user some ETH, as it modifies the blockchain state. Once the transaction is confirmed, a new delegate is added to the DID document. To make sure a new delegate has been added correctly, we can resolve the DID Document using the DID Universal Resolver. You can see that the delegate has been added to the DID Document of the user's account.

The next step is to fill out the form and request the VC. We also use the Veramo in the platform's backend to generate a VC. The user will be prompted to save the VC in the MetaMask state. Currently, this is done in a non-standardized way, but we are looking into OpenID Connect standards for VC Issuance.

A VC should appear under 'My VCs' on the profile page if everything goes well. This VC can then be used to create a VP when the user votes on the Snapshot platform.

To use the SSI Snap, dApps only need to implement a Connect MetaMask button and call our custom RPC methods.

As a next step in the PoC we have to expand the demo with the Snapshot voting mechanism that only allows those users to vote on specific proposals within a DAO, who can provide a valid VP. As seen in Fig. 5, the user voting on the Snapshot platform selects a valid VC, for which a VP is generated, and signs a transaction containing the content of the vote, including the VP.

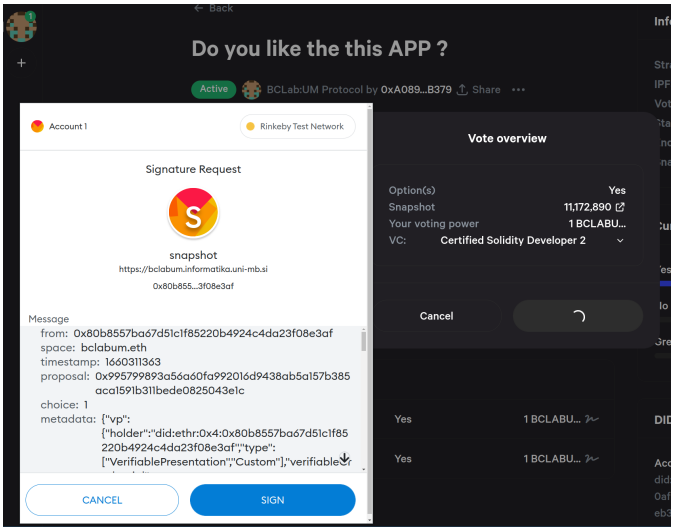

Figure 5: Demonstration of usage of the proposed solution on the modified Snapshot platform, where the voting based on VC is evident.

5. Discussion

Our main goal was to bring support for SSI principles to DAO, i.e., enable DAOs to leverage stakeholders' proofs of skills and knowledge as a ticket for DAO voting. This had to be achieved with the notion of bringing DIDs and VCs to the MetaMask crypto wallet. The MetaMask Snaps system allowed us to add those functionalities to the most popular crypto wallet. With our SSI Snap we provided MetaMask users access to SSI without the need to install redundant applications or software. Existing MetaMask accounts become DIDs and users can store VCs directly in MetaMask and generate VPs when needed. The SSI Snap leverages the security of MetaMask, a secure and tested wallet, used by millions, so there is no need to worry about the security of storing data in a new application. Another advantage is the ability to create additional DIDs that are not necessarily MetaMask accounts and the ability to implement additional ways of storing data. The SSI Snap will be configurable, giving the users an option to decide where their data should be stored (e.g. IPFS) and which VC serves what purpose.

A disadvantage we see in the MetaMask Snaps system is the lack of support for developing a custom user interface, hence any interaction with SSI Snap requires a dApp. There is currently

also no way to choose a single VC inside MetaMask which means that all VCs need to be sent to the dApp where the user gets to select one and a dApp receives more data than necessary. Regarding configurability, another disadvantage is again the need for dApp for a user to configure the Snap to his needs. The SSI Snap also currently does not provide any way to synchronize data with other MetaMasks applications - all data is stored locally.

Overall, the SSI Snap enhances the MetaMask with SSI principles and provides unified support for DIDs and VCs without the need for users to install and use any additional applications.

6. Conclusion

Decentralized autonomous organizations are community-controlled organizations. The currently dominant form of DAOs, token-based DAOs suffer from the Plutocracy Problem. DAOs should strive for meritocracy, where the power of members is based on their skills. To achieve meritocracy, DAOs have to evolve from token-based form to a form that uses self-sovereign identity with credentials. SSI enables individuals to create and control their identities online. SSI consists of unique decentralized identifiers, verifiable credentials, and verifiable presentations.

With the goal of meritocracy in DAOs, we developed a MetaMask snaps application that allows users to securely store and use SSI in their existing MetaMask wallet. This application gives millions of existing MetaMask users easy access to the SSI.

Acknowledgments

This work was supported by the Slovenian Research Agency (Research Core Funding) under Grant P2-00577.

References

[1] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, F.-Y. Wang, Decentralized autonomous organizations: Concept, model, and applications, IEEE Transactions on Computational Social Systems 6 (2019) 870–878. doi:10.1109/TCSS.2019.2938190.

[2] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, ArXiv abs/1906.11078 (2019).

[3] Serto, The dao plutocracy problem, 2021. URL: https://serto.medium.com/the-dao-plutocracy-problem-a8841546a0f2.

[4] V. Buterin, Soulbound, 2022. URL: https://vitalik.ca/general/2022/01/26/soulbound.html.

[5] V. Buterin, Moving beyond coin voting governance, 2021. URL: https://vitalik.ca/general/2021/08/16/voting3.html.

[6] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, A. Hobor, Making smart contracts smarter, CCS '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 254–269. URL: https://doi.org/10.1145/2976749.2978309. doi:10.1145/2976749.2978309.

[7] C. Dannen, Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners, 1st ed., Apress, USA, 2017.

[8] A. Preukschat, D. Reed, Self-sovereign identity, Manning Publications, 2021.

[9] Š. Čučko, M. Turkanović, Decentralized and self-sovereign identity: Systematic mapping study, IEEE Access 9 (2021) 139009–139027.
[10] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.
[11] M. Hussey, What is snapshot? the decentralized voting system, 2021. URL: https://decrypt.co/resources/what-is-snapshot-the-decentralized-voting-system.
[12] U. W. Chohan, The decentralized autonomous organization and governance issues, 2017. URL: http://dx.doi.org/10.2139/ssrn.3082055. doi:10.2139/ssrn.3082055.
[13] A. Wright, The rise of decentralized autonomous organizations: Opportunities and challenges, Stan. J. Blockchain L. & Pol'y 4 (2020) 1.
[14] M. A. Hassan, Z. Shukur, Review of digital wallet requirements, in: 2019 International Conference on Cybersecurity (ICoCSec), 2019, pp. 43–48. doi:10.1109/ICoCSec47621.2019.8970996.
[15] S. Schwalm, D. Albrecht, I. Alamillo, eidas 2.0: Challenges, perspectives and proposals to avoid contradictions between eidas 2.0 and ssi, in: H. Roßnagel, C. H. Schunck, S. Mödersheim (Eds.), Open Identity Summit 2022, Gesellschaft für Informatik e.V., Bonn, 2022, pp. 63–74. doi:10.18420/OID2022_05.
[16] Y. K. Chaturvedi, A quick guide to metamask snaps, 2022. URL: https://etherworld.co/2022/01/19/a-quick-guide-to-metamask-snaps/.
[17] MetaMask, Introduction | MetaMask Docs, 2022. URL: https://docs.metamask.io/guide/snaps.html.
[18] W3C, Decentralized Identifiers (DIDs) v1.0, 2021. URL: https://www.w3.org/TR/did-core/.
[19] M. Sporny, D. Longley, D. Chadwick, Verifiable credentials data model v1.1, 2022. URL: https://www.w3.org/TR/vc-data-model/.
[20] O. Goldreich, Y. Oren, Definitions and properties of zero-knowledge proof systems, Journal of Cryptology 7 (1994) 1–32.
[21] Veramo, Veramo - A JavaScript Framework for Verifiable Data | Performant and modular APIs for Verifiable Data and SSI, 2022. URL: https://veramo.io/.

A. Online Resources

To learn more about the SSI Snap, its architecture, and how to use it, a GitHub repo and the course demo are available.

- GitHub: https://github.com/blockchain-lab-um/ssi-snap,
- Demo: https://blockchain-lab-um.github.io/course-dapp/,
- MetaMask Flask: https://metamask.io/flask/.