Governance In DeFi

Contents

- Introduction
- Governance Tokens
- Evolution of Governance in DeFi Protocols
- Token Distrubution & Centralisation
- Centralisation-enabled DeFi Vunerabilities
- AI-enabled On-chain Governance
- Findings
- Conclusion
- Content with notebooks
- Notebooks with MyST Markdown
- Self-Sovereign Identity: Technical Foundations and Applications

This article reviews the evolution of governance in DeFi protocols, security risks of associated with certain governance mechanisms and novel methods of on-chain governance.

Introduction

With centralisation the main theme of traditional finance, DeFi’s emerged as a viable alternative to the centralised financal institution building upon the peer-to-peer transactions functionality of cryptocurrencies.

Governance Tokens

Governance tokens are units that are interchangeable and serve to implement a voting mechanism among a group of participants. Through majority-voting schemes, holders can express their intentions for the development of the protocol `` {cite}`xu2022banks` ``.

Evolution of Governance in DeFi Protocols

The evolution of governance in a number of DeFi protocols often follows a pattern with three major stages, namely; centralised team-based governance, decentralised token-based governance and centralised token-based governancne `` {cite}`stroponiati5decentralized` ``.

Centralised Team-based Governance

Following the commencement of the DeFi protocol, the team often adopts a centralized governance framework to facilitate key

Skip to main content

Decentralised Token-based Governance

A gradual shift occurs from team to token based governance through the allocation of tokens to protocol users to forster decentralisation in the ecosystem.

Centralised Token-based Governance

Token Distrubution & Centralisation

Effective token distribution strategies are of paramount importance for governance in DeFi. This is because the distribution process determines the number of individuals who can exercise control over a project, as well as the extent of their voting power. Hence, a well-planned and executed token distribution strategy is a critical aspect of the success of voting rights tokens.

Centralisation-enabled DeFi Vunerabilities

MakerDAO Vunerbaility

An execuitve contract exists in MakerDAO which uses approval voting to alter the state of the entire system via an "executive contract". With this implementation, MakerDAO governance framework is vulnerable to attacks from malicious contracts, which could steal funds locked as collateral. An attacker could gain control of the contract with the most tokens staked (MKR) and potentially access $2 billion worth of collateral locked in Maker {cite}`stroponiati5decentralized

AI-enabled On-chain Governance

Findings

In conducting case studies, DeFi protocols analysed demonstrated a high degree of centrality {cite}`barbereau2022defi`. A combination of lending protocols, decentralised exchanges and yield aggregators were analysed to understand the governance centralisation. Another study conducted an empirical analysis of the top DeFi protocols by examining the wallet addresses and token holdings of participants and found evidence of centrality with voting power controlled by the top 5, top 100 and top 1000 addresses {cite}`jensen2021decentralized`. The protocols analysed in this study included Compound, Uniswap, Compound, Balancer and Yearn Finance. Compoound showed the highest evidence of centrality and Uniswap the least with the top 5 address for both constituting 42.1% and 12.05% respectively. Another unique approach involved the survey of users of DeFi protocols through interviews {cite}`ekal2022defi`. Voter collusion, low participation and voter apathy were identified as the main challenges in decentralised governance. In factoring the change in voting power over time, the centraliaiton of DeFI protocols increased with time with the majority token holders purchasing more tokens over a period of time {cite}`barbereau2022decentralised`.

Conclusion

Skip to main content

You can also create content with Jupyter Notebooks. This means that you can include code blocks and their outputs in your book.

Markdown + notebooks

As it is markdown, you can embed images, HTML, etc into your posts!

You can also add_{math} and

$$math^{blocks}$$

or

$$\mathbf{mean} la_{tex}$$

$$math blocks$$

But make sure you $Escape $your $dollar signs $you want to keep!

MyST markdown

MyST markdown works in Jupyter Notebooks as well. For more information about MyST markdown, check out the MyST guide in Jupyter Book, or see the MyST markdown documentation.

Code blocks and outputs

Jupyter Book will also embed your code blocks and output in your book. For example, here's some sample Matplotlib code:

Skip to main content

```
from matplotlib import rcParams, cycler
import matplotlib.pyplot as plt
import numpy as np
plt.ion()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 1
----> 1 from matplotlib import rcParams, cycler
      2 import matplotlib.pyplot as plt
      3 import numpy as np

ModuleNotFoundError: No module named 'matplotlib'
```

```
# Fixing random state for reproducibility
np.random.seed(19680801)

N = 10
data = [np.logspace(0, 1, 100) + np.random.randn(100) + ii for ii in range(N)]
data = np.array(data).T
cmap = plt.cm.coolwarm
rcParams['axes.prop_cycle'] = cycler(color=cmap(np.linspace(0, 1, N)))


from matplotlib.lines import Line2D
custom_lines = [Line2D([0], [0], color=cmap(0.), lw=4),
                Line2D([0], [0], color=cmap(.5), lw=4),
                Line2D([0], [0], color=cmap(1.), lw=4)]

fig, ax = plt.subplots(figsize=(10, 5))
lines = ax.plot(data)
ax.legend(custom_lines, ['Cold', 'Medium', 'Hot']);
```

There is a lot more that you can do with outputs (such as including interactive outputs) with your book. For more information about this, see the Jupyter Book documentation

Notebooks with MyST Markdown

Jupyter Book also lets you write text-based notebooks using MyST Markdown. See the Notebooks with MyST Markdown documentation for more detailed instructions. This page shows off a notebook written in MyST Markdown.

An example cell

With MyST Markdown, you can define code cells with a directive like so:

```
print(2 + 2)
```

```
4
```

When your book is built, the contents of any `{code-cell}` blocks will be executed with your default Jupyter kernel, and their outputs will be displayed in-line with the rest of your content.

Skip to main content

See also

Jupyter Book uses Jupytext to convert text-based files to notebooks, and can support many other text-based notebook files.

Create a notebook with MyST Markdown

MyST Markdown notebooks are defined by two things:

1. YAML metadata that is needed to understand if / how it should convert text files to notebooks (including information about the kernel needed). See the YAML at the top of this page for example.
2. The presence of `{code-cell}` directives, which will be executed with your book.

That's all that is needed to get started!

Quickly add YAML metadata for MyST Notebooks

If you have a markdown file and you'd like to quickly add YAML metadata to it, so that Jupyter Book will treat it as a MyST Markdown Notebook, run the following command:

```
jupyter-book myst init path/to/markdownfile.md
```

Self-Sovereign Identity: Technical Foundations and Applications

Key Insights!

- SSI systems use DIDs as unique, resolvable identifiers for each entity, allowing the secure management of digital identities without relying on a centralized authority.
- VCs provide cryptographically verifiable proof of an individual's attributes or personal information, enabling secure and trustworthy data sharing between issuers, holders, and verifiers.
- SSI incorporates privacy-preserving mechanisms such as zero-knowledge proofs and selective disclosure, allowing users to prove their credentials without revealing their actual identity or unnecessary information.
- While not mandatory, using blockchain as a decentralized data registry in SSI systems enables secure, tamper-evident, and verifiable storage of credentials, enhancing the trustworthiness and reliability of the identity management process.

Introduction

According to World Bank estimates, nearly 850 million people lack an official identity [JC23], and the proliferation of digital devices has made it increasingly essential to possess a verifiable digital identity. This has led to a rise in the number of digital transactions and the need for a secure and reliable identity management system. SSI is emerging as a decentralized alternative to traditional centralized identity management systems, in which identities are cryptographically verifiable. It allows individuals to control their digital identities and share them with trusted parties. Each entity in the SSI system is identified by a unique DID (Decentralized Identifier) (???), which can be resolved to reveal information such as the entity's public key and other metadata

Skip to main content

$$\underbrace{\texttt{DID}}_{\text{Scheme}} : \overbrace{\texttt{example}}^{\text{DID Method}} : \underbrace{\texttt{BzCbsNYhMrjHiqZDTUASHg}}_{\text{Method Specific Identifier}}$$

DID breakdown

See also

Find out more about some of the most commonly used DID methods:

- DID:INDY
- DID:UPORT
- DID:SOV

While centralized identities and federated identities offer convenience, control remains with the identity provider [LB15]. User-centric identities such as OpenID [RR06] and OAuth [FKustersS16] improve portability but do not give full control to the users. SSI is designed to provide users with full control over their digital identities, and involves guiding principles around security, controllability, and portability. In addition to providing full control, Bernabe et al. [BCHR+19] presented a classification of techniques for maintaining privacy in SSI, which included Secure Multiparty Computation and Zero-Knowledge Proofs, among others.

The three main parties involved in SSI systems are issuer, holder and verifier as shown in [Fig. 1]. The issuer issues a cryptographically signed credential to the holder, and the verifier is the entity that that confirm the authenticity of the credential using a decentralized data registry such as Blockchain. Holders store their credentials in secure digital wallets and can share them with other parties as needed. The holder can also create a presentation request and share it with the verifier.

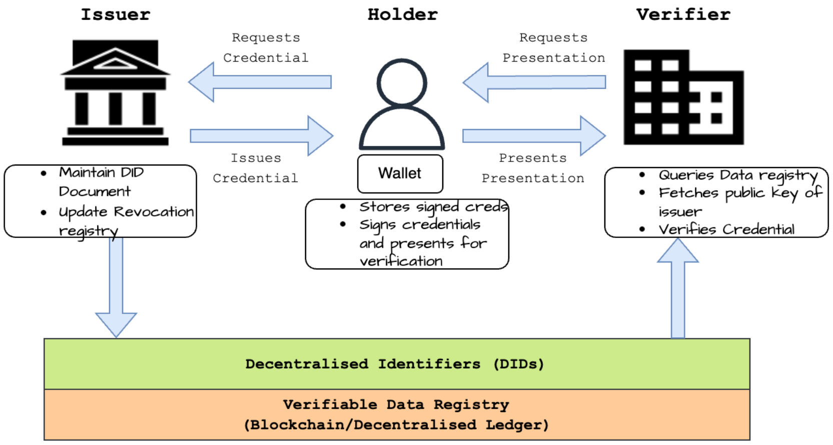
Issuer
Holder
Verifier
Requests Credential
Requests Presentation
Issues Credential
Presents Presentation
Wallet
Maintain DID Document
Update Revocation registry
Stores signed creds
Signs credentials and presents for verification
Queries Data registry
Fetches public key of issuer
Verifies Credential
Decentralised Identifiers (DIDs)
Verifiable Data Registry (Blockchain/Decentralised Ledger)

Fig. 1 SSI entities and their relations

Click here to see how a verifiable credential actually looks like

SSI

Self-Sover
decentrali
system wh
technolog
individuals
identities s

Verifiabl

A verifiabl
provides a
verifiable
informatio

Skip to main content

Nitty Gritties of SSI

- SSI solutions are designed to be blockchain-agnostic and adhere to W3C's specifications.
- The identity wallets (e.g., uPort, Trinsic, Connect.Me) are different from the digital wallets (e.g., Coinbase, Ledger, Trezor) that store cryptocurrencies in the sense that they store and manage DIDs and VCs instead of cryptocurrencies.
- To protect privacy, SSI solutions (e.g. - Hyperledger Indy and Aries) are increasingly using Zero-Knowledge Proofs (ZKPs) to prove the authenticity of credential```s without revealing the actual data.
- To facilitate secure communication between different SSI components (issuer-holder-verifier), DIDComm and CHAPI protocols have been developed and heavily used.

Applications for SSI

Recent studies have demonstrated the feasibility of using zero-knowledge proofs to disclose information selectively, such as proof of vaccination status, without revealing users' identities. These studies have employed interoperable open-source tools to implement these systems globally at a minimal cost. Schlatt et al. [SSFU22] illustrates how a customer can leverage a Zero-knowledge Proof concept called 'blinded link secret' to disclose information selectively. Similarly, Barros et al. [dVBSFCustodio22] implemented a prototype of an application for presenting proof of vaccination without revealing users' identities. Furthermore, it uses interoperable open-source tools across countries to implement this system globally at a minimal cost for each country's government. The NHS Digital Staff Passport solution [LC22] employs the Sovrin Network as a public key infrastructure (PKI) to manage verifiable credentials for staff onboarding. Hospitals register on the network and use their private keys to sign credentials, while staff members utilize Evernym's Connect.Me SSI digital wallet app to store and share credentials.

Shuaib et al. [SHU+22] suggest that a blockchain-based land registry system can be combined with a self-sovereign identity (SSI) solution to provide a secure and efficient identity management system for landowners. Three existing SSI solutions: Everest, Evernym, and uPort [Ame22], were evaluated based on SSI principles [All16] to determine their compliance and effectiveness in addressing identity problems in land registry systems. The Everest platform was found out to be the most compliant with the SSI principles, whereas Evernym and uPort had some limitations in terms of interoperability and user control.

Estonia is one of the few countries in the world that have managed to make e-voting a reality [SS22]. Sertkaya et al. [SRR22] proposed an EIV-AC scheme that integrates the Estonian Internet voting (EIV) scheme with anonymous credentials (AC) based on self-sovereign identity (SSI). The use of SSI-based anonymous credentials enables voters to prove their eligibility to vote without revealing their identity. The zero-knowledge proof of knowledge is used to prove that the voter has the right to vote without revealing any additional information. The EIV-AC scheme enhances the security and privacy of the EIV scheme, making it more compliant with privacy-enhancing and data minimization regulations.

Can SSI work without Blockchain?

Blockchain is one of many options when implementing the Self-sovereign Identity system. Alternatives like IPFS, Public-key cryptography and even traditional Certificate Authorities can be used to implement SSI []. However, the main advantage of using Blockchain is that it provides a decentralized and immutable ledger that can be used to store and verify credentials.

Conclusion

Self-sovereign identity can potentially revolutionize various industries, including healthcare, voting systems and many more. However, as research and development in SSI progress, it will be crucial to address interoperability, scalability, and usability

Skip to main content

[All16] Christopher Allen. The path to self-sovereign identity. *Life With Alacrity*, 2016. URL: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html.

[Ame22] New America. Three self-sovereign identity platforms to watch. *New America*, 2022. URL: https://www.newamerica.org/future-land-housing/reports/nail-finds-hammer/three-self-sovereign-identity-platforms-to-watch/.

[BCHR+19] Jorge Bernal Bernabe, Jose Luis Canovas, Jose L Hernandez-Ramos, Rafael Torres Moreno, and Antonio Skarmeta. Privacy-preserving solutions for blockchain: review and challenges. *IEEE Access*, 7:164908–164940, 2019.

[dVBSFCustodio22] Mauricio de Vasconcelos Barros, Frederico Schardong, and Ricardo Felipe Custódio. Leveraging self-sovereign identity, blockchain, and zero-knowledge proof to build a privacy-preserving vaccination pass. *Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass*, 2022.

[FKustersS16] Daniel Fett, Ralf Küsters, and Guido Schmitz. A comprehensive formal security analysis of oauth 2.0. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1204–1215. 2016.

[JC23] CLAIRE CASHER JULIA CLARK, ANNA DIOFASI. 850 million people globally don't have id—why this matters and what we can do about it. *World Bank*, 2023. URL: https://blogs.worldbank.org/digital-development/850-million-people-globally-dont-have-id-why-matters-and-what-we-can-do-about.

[LC22] Mary Lacity and Erran Carmel. Implementing self-sovereign identity (ssi) for a digital staff passport at uk nhs. *University of Arkansas*, 2022.

[LB15] Maryline Laurent and Samia Bouzefrane. *Digital identity management*. Elsevier, 2015.

[RR06] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, 11–16. 2006.

[SSFU22] Vincent Schlatt, Johannes Sedlmeir, Simon Feulner, and Nils Urbach. Designing a framework for digital kyc processes built on blockchain-based self-sovereign identity. *Information & Management*, 59(7):103553, 2022.

[SS22] Cyber Security and Society. Estonia leads world in making digital voting a reality. *Cyber Security and Society*, 2022. URL: https://www.ft.com/content/b4425338-6207-49a0-bbfb-6ae5460fc1c1.

[SRR22] Isa Sertkaya, Peter Roenne, and Peter YA Ryan. Estonian internet voting with anonymous credentials. *Turkish Journal of Electrical Engineering and Computer Sciences*, 30(2):420–435, 2022.

[SHU+22] Mohammed Shuaib, Noor Hafizah Hassan, Sahnius Usman, Shadab Alam, Surbhi Bhatia, Arwa Mashat, Adarsh Kumar, and Manoj Kumar. Self-sovereign identity solution for blockchain-based land registry system: a comparison. *Mobile Information Systems*, 2022:1–17, 2022.