Ljubljana, 13.6.2023

Vid Bolčič, Lovro Klenovšek, Andraž Čeh, Gal Lindič, Jan Krivec

GitHub repository: https://github.com/jan-krivec/FOG

Decentralized Publishing Organization(DPO)

In this section we will present how our Decentralized Publishing Organization would work. It consists of 2 parts: Frontend (UI) and decentralized components (Smart Contracts). Users would access the application through a web page.

For the authentication we are using Metamask, which can be used to ensure that only authorized users can access and interact with the platform. The user would need to have its own Metamask account which would be used to store information about his reviews, reputation, etc. The authentication with Metamask would work in the following way. When a user enters the platform they would be prompted to connect their Metamask wallet to the platform and if they proceed to do so they would be granted access to interact with the platform. If a user just wants to read journals he would not need to authenticate himself, but for paid articles he would have to buy access with Metamask, or prove he has done so in the past.

DPO application

We created a basic working implementation of the DPO, where authors can publish their articles and submit them to the review process. As the article is submitted, three reviewers and one editor are automatically chosen from DAO to review / edit the article. Reviewers and editors can see all articles that still have to be reviewed. All three reviewers have to submit their grade and comment for the article. After this is done, the editor has to give the final approval for the article to be published or denied. If the article is denied, the author can resubmit his article and the reviewing process starts over again. When the reviewer or editor leaves their review, they can no longer change their vote until the article has been denied and the reviewing process starts over again.

Each article has to have a unique name. The editor can not give the final approval without all three reviewers leaving their review. The reviewers also have to leave a grade and a comment for the grade for the review to be valid. Only the reviewers and editor that were assigned to this article can leave a review and after they left their vote, they can not vote

again (until the article is being revised). All validation logic is held inside the smart contract. Each article pdf file is stored in IPFS which can be accessed anytime.

When the editor approves the article, he will also mint the NFT of the article to the author. When a normal user accesses the

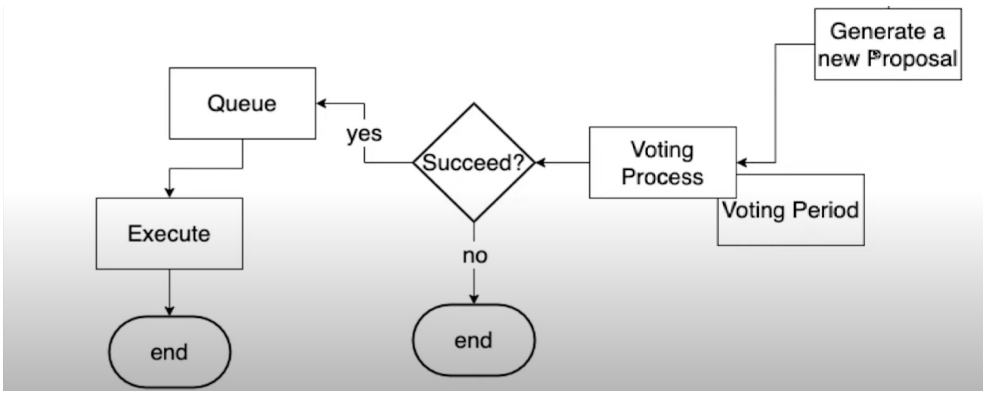
DAO

In the DPO we differentiate between multiple user roles such as Reader, Author, Reviewer and Editor. By using a DAO (Decentralized Autonomous Organization), we enable users to be promoted from reader all the way to the editor role. The way this would work is as follows. The DAO would be a smart contract on the blockchain, with a set of rules and governance mechanisms that determine how the platform operates. Each user would have a vote in the DAO, based on the number of tokens they hold. These tokens are used to represent each user's stake in the platform.

User can request the promotion through the DAO, which requires the approval of other users with sufficient voting power. Once the promotion is approved the change is written on blockchain.

Here are a few steps explaining how this system works:

- Token allocation: When someone joins our organization, they are initially allocated 1000 tokens. These tokens symbolize their stake in the organization and simultaneously determine their voting power.
- Proposal creation: Once a user receives these tokens, they can create a proposal to promote their role within the organization. For instance, if they aspire to become an editor, they can propose this to the remaining members.
- Voting: Once the proposal is created, a voting process begins. Every member of the organization who holds tokens can vote 'for', 'against', or 'abstain'. The voting power of each member depends on the number of tokens they hold.
- Proposal execution: If the proposal is accepted, the changes in roles are carried out accordingly. The user who proposed the promotion moves to the new role - for example, from an author to an editor.
- Token adjustment: Upon promotion, the user's token count is adjusted. For example, if a user is promoted to an editor, the number of their tokens may increase, reflecting their new role and greater responsibility within the organization.



This is done through collection of smart contracts written in Solidity, intended to facilitate a decentralized autonomous organization (DAO) governance system.

- The Governor contract extends various contracts that give it different abilities like voting, setting the voting period, handling quorum, etc. It also has a proposal counter (s_proposalCount).
- The TimeLock contract extends the TimelockController, which provides mechanisms for delayed execution, allowing scheduling of contract calls to be made at a future date.
- The UserRoles contract, extending the Ownable contract, is meant for managing user roles. Each address can have a specific role assigned, which can be managed (added or removed) only by the owner of the contract. It also provides an overview of all the current users and their respective roles.
- The GovernanceToken contract, which extends the ERC20Votes contract, is the token for the system. It allows users to claim tokens (up to a defined maximum per user), keep track of token holders, and mint or burn tokens as necessary. It also emits events for important actions such as token transfers, minting, and burning. The contract also allows for some tokens to be kept back (a percentage defined at the time of contract deployment) while the rest are stored in the contract, ready to be claimed by users. The '_afterTokenTransfer', '_mint', and '_burn' functions are overridden to emit events whenever these actions are performed.

In conclusion, implementing a DAO-based model for a decentralized publishing organization (DPO) provides a transparent, democratic, and secure way to manage user roles and their evolution within the organization. The use of blockchain technology and smart contracts ensures a decentralized system, where all actions are permanently recorded, and no changes can be made without consensus from the stakeholders.