**Votility Protocol: An extensible and customizable governance building block**

Gleisson de Assis
gleisson@votility.io
www.votility.io

**Abstract:** This article presents the Votility Protocol, a customizable and extensible platform for developing decentralized autonomous organizations (DAO), which rewards users for using it. Thus, in a simplified way, it explains the development tools used and the methodology applied in this technology. The protocol differs from some existing tools, it does not focus on being the DAO of existing protocols but on allowing the standardized and parameterized construction of new DAOs. By using the existing functionality in the protocol, developers can focus exclusively on the final event in the governance chain, having total control over what is executed, leaving the basic logic of governance to the tools available in Votility, which presents itself not only as a code library, but also as a simple and effective way to implement DAOs.

**Keywords:** decentralized protocol, autonomous decentralized organizations, building blocks, smart contracts, Ethereum, blockchain.

# 1. INTRODUCTION

To begin the subject and present all the advantages of the protocol it is important to conceptualize a few points. Thus, the starting point is the definition of what is a decentralized autonomous organization, or DAO. Think of a vending machine, such as those available at self-service stations selling soft drinks and other food products, imagine a hypothetical scenario of self-management of this machine:

- When it runs out of a product in stock, it automatically places the order and pays the supplier.
- When it realizes that a product does not usually sell, it is removed from the supply list and brings in a better one.
- When making purchases you earn the right to select products to enter the offer list for the next period.

A DAO or Decentralized Autonomous Organization works on similar principles. The idea of this management model has been circulating in the cryptocurrency community ever since Bitcoin [1] managed to get rid of intermediaries in financial transactions. Similarly, the main idea behind a DAO is to establish an organization represented by codified rules like a computer program that is transparent, controlled by the members of the organization and not influenced by a central government. A DAO represents an innovation in the design of organizations in terms of its emphasis on computerized rules and contracts, however the structures and functions of the DAO also raise governance issues [2].

The DAO concept is empowered by blockchain, as it provides the key requirements for operationalization: transparency, immutability, and traceability. The economic system itself provided by Bitcoin can be understood as a DAO, as there is no need for intermediaries, trust is guaranteed by cryptographic criteria and the decentralization of the network, however, another organization became better known which was The DAO. The DAO was intended to function as a venture capital fund for the cryptocurrency universe. The fundraising period was a success with about 150 million dollars, however the fact that made it more famous was precisely a major flaw, a hacker found a loophole in the coding of the smart contract that allowed him to drain funds, stealing about 60 million dollars [3].

Even surrounded by several inconsistencies and controversial issues, the idea of autonomous organizations and especially the post-bitcoin scenario brings new opportunities. Applications can run in a decentralized manner and operate autonomously, and this is the historical and evolutionary basis that smart contracts are based on, which at the time of writing this text explodes in the concept of decentralized finance.

Another important subject to progress with the understanding of this paper is the concept of building blocks and how computing environments benefit from the use of previously established solution structures. By simple definition, building blocks are entities that when put together make it possible for something to exist[4]. There are huge benefits when developers use code libraries to speed up development in computational environments, reducing the time spent on recurring tasks. In addition to libraries (small pieces of code that maintain responsibility for a specific point), architects pass on knowledge through design patterns [5]. Problems commonly found in software development are quickly resolved with solutions already tested and established. Using libraries and design patterns, developers create better solutions, as they do not need to expose themselves to the performance and security risks and problems previously identified and reduce time in the designing of complex solutions. Building blocks bring the best practices in solving a given problem and constructing new modular and configurable software.

Suppose the reuse of solutions (patterns) and code (libraries) is already relevant in a traditional computing environment. In that case, it is even more necessary in a distributed and immutable computing environment such as blockchain. Projects such as OpenZeppelin[1] present tested and consolidated standards, Chainlink[2] bringing external data queries to the blockchain, and even initiatives such as a date/time library [5] help and optimize the smart contracts development, as well as minimizing the problems of new codes and the implementation cost. In the case of deployed libraries, there is a saving in the computational power required to publish a new contract.

With Ethereum's[3] blockchain usage and other protocols that enable the writing of smart contracts, and in particular with the explosion of the topic of decentralized finance, the concept of building blocks has been widely used [6]. Protocols are built on top of other protocols creating a network of dependencies that makes each smart contract an integral part of a large decentralized computing solution. As stated by Onyekwere (2020) "A DeFi project has to be decentralized. This means that if at any time, you decide to change the protocol, the developer

[1] OpenZeppelin - The standard for secure blockchain applications - https://openzeppelin.com
[2] Chainlink- Connect your smart contract to the outside world - https://chain.link
[3] Ethereum - https://ethereum.org

cannot just decide on your own to make this change, you need to involve all the stakeholders and the community".

In this context, the Votility Portocol is presented, which aims to bring all the benefits already discussed in the previous paragraphs to the universe of DAO creation. The protocol forces the developer to use known design patterns in the creation of the base token, standardizes essential security issues, defines a communication mechanism between components, and drastically reduces development time, since it delivers the deployed smart contract and a user interface (dApp) to manage the proposals that will govern the DAO. These issues will be discussed in the next topics of the text.

## 2. METHODOLOGY

Before going into the technical aspects of implementing DAOs using the Votility Protocol, it is essential to highlight the principles that the methodology relies on. Tools like Snapshot.page[4] bring an interesting feature to the community around a project, because the participation in the governance of the protocol is done off-chain (centralized storage and processing) and there is no transaction cost in blockchain, this significantly increases the engagement of users, even being centralized in the voting control, the use of the IPFS protocol [8] is a step towards the decentralization of data, however, a critical point in the off-chain approach is that the execution of the direction pointed by the community (result of a vote) depends exclusively on the team or the project owner. On the opposite extreme we can think of the totally on-chain process (decentralized storage and processing) which guarantees total transparency and independence, but there is a strong obstacle which are the transaction costs, which makes it prohibitive for the great majority of users, benefiting a small part of the token holders, breaking the main concept of a DAO.

The Votility Protocol leaves it up to the proposing user to choose between a fully on-chain model, which is understood to be necessary in critical cases, and a hybrid model: a proposal is created and executed on-chain but the voting is off-chain. The hybrid model allows the community participation since there is no need to compromise operating costs in the voting. Even with a centralized point, this is minimized by using the IPFS protocol, user signatures, and protocol signature, generating a decentralized data chain with a low storage cost.

Another essential point for validating the participation of members of a DAO is the unit that determines the vote's weight, which is usually associated with the amount of tokens for a given project called voting power. We believe that the ideal scenario is to consult this information directly on the blockchain, but it needs to be timed, otherwise, a malicious user could choose an option (vote) and then transfer his voting power to another user, manipulating the result of the proposal. For this, we use the concept of snapshot, the owner of a given token needs to implement the ERC20Snapshot standard:

> *"This can be used to safely create mechanisms based on token balances such as trustless dividends or weighted voting. In naive implementations it's possible to*

[4] Snapshot - https://snapshot.page

*perform a "double spend" attack by reusing the same balance from different accounts. By using snapshots to calculate dividends or voting power, those attacks no longer apply. It can also be used to create an efficient ERC20 forking mechanism." [9]*

Having defined the strategy that will best bring the community to participate in a proposal (on-chain or off-chain) and the way to establish the voting power of each user, based on a snapshot, it is necessary to observe other aspects that will guarantee security, clarity and trust. In the methodology we highlight the minimum time required for a proposal to receive votes and the minimum quorum, this will avoid that, at the moment that a given protocol chooses to be fully autonomous, unfounded and opportunistic proposals are sent by malicious users, besides of course defining a set of users and other autonomous entities that may submit proposals, creating an expansive and secure inter-protocol communication ecosystem, which will always give the community the opportunity to dictate the course of decisions.

To finalize the modeling of a DAO the developer needs to think about how proposals will be parameterized, this is done through the concept of input parameters and voting options. Input parameters define the target of the vote and are extensible as input parameters of a function or method (concepts associated with programming language), while options define the course of the vote. Note the following proposals, even if hypothetical they define scenarios very common in DAOs:

1. "Send 10% of the allocated fund to build a new user interface. The amount should be sent to the responsible developer at 0xBff...90A. The community must vote Yes or No".
2. "Investment of 10 ETH to buy new assets for the investment fund. The purchase will be made in proportion to the community's choice. The options will be Asset #1 (0x658...B00), Asset #2 (0xf0c...d85 ), Asset # 3 (0xeed...7e7) and Asset #4 (0x5C9...881)".
3. "Change of the MONTH_FEE parameter to 10%. The community must vote Yes or No".

For each proposal, a developer must set up what are the input parameters and what are the voting options:

- Parameters: allocation percentage (10%) and destination address (0xBff...90A).
  Options: Yes and No.
- Parameters: quantity of ETH (10).
  Options: Active #1 (0x658...B00), Active # 2 (0xf0c...d85), Active # 3 (0xeed...7e7) and Active # 4 (0x5C9...881).
- Parameters: configuration parameter (MONTH_FER) and the new value (10%).
  Options: Yes and No.

These settings, made on-chain, ensure full transparency of what the proposal is. Users can audit the options and check the behavior of the DAO smart contract. Regardless of the

purpose of the organization or the proposal, both the input parameters and the result after the choice are determined at the beginning and are public knowledge.

The final step in the deployment of a new DAO is for the developer to publish a smart contract describing what will be accomplished at the end of the proposals. All other aspects are handled by the protocol. Below is a high-level code example referring to the first hypothetical proposal.

```
int amount = intputData[0] * totalAmount;
address target = inputData[1];

if winnerOption === YES then
     send(amount, target);
end if;
```

FIGURE 1 - High-level code. Source: Author.

Only these lines of code are necessary to describe the whole purpose of the DAO, since all the aspects discussed are left to the responsibility and management of the protocol. A critical point to be noted is that in the example, the funds are never held by the Votility Protocol, but by the smart target contract written by the developer, i.e., at no time will the developer cede control over his DAO under any aspect, because he will only be building the decentralized organization under the foundations of the methodology.

## 2.1. ON-CHAIN VOTE vs OFF-CHAIN VOTE

Choosing between on-chain and off-chain voting requires additional care by the proponent. What is being placed in the balance is total decentralization with a high operational cost and centralization with no operational cost. It is undeniable that the removal of the cost factor increases the participation of the community that makes up the DAO, however it figures as a point of failure, which is the centralization of voting data. The Votility Protocol reduces the burden of this decision by using 4 components: on-chain voting power query based on a snapshot generated by the token that determines the voting power, voting signatures by the community member using the private key associated with the account used in the voting, signing of the data by the private keys representing the Votility Protocol ensuring data integrity, and finally the decentralization of the data through the IPFS protocol.

When a user casts a vote and the data is propagated via IPFS there is a link to the previous vote, this way any user can audit the sequence of votes and the associated metadata (weight and signatures). At the end of the proposal a final report is generated presenting the voting summary and is signed by the Votility Protocol. This data set is in turn sent to the smart contract, which validates the signatures and only accepts it as a valid result if it comes from an integer data source. This process reduces the risk of centralizing the vote collection process and keeps the flow on-chain (creation, data input and proposal finalization), increasing the

autonomy of the decentralized organization, since the flow remains automated and independent of human operation.

## 2.2. DAO TECHNICAL DETAILS

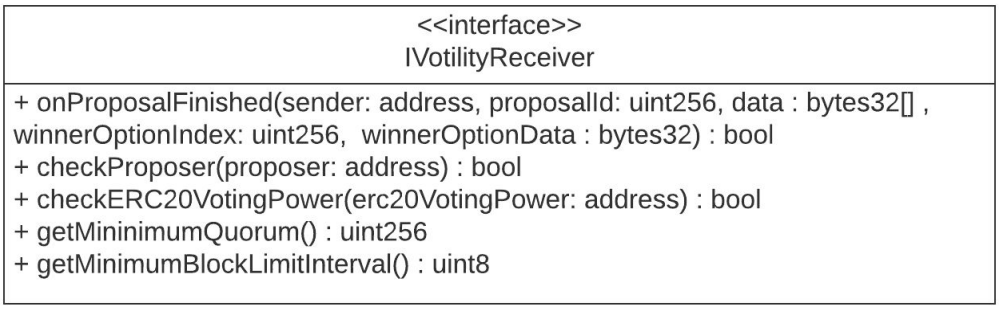


FIGURE 2 - IVotilityReceiver interface. Source: Author.

Following the steps described in the methodology, the developer needs to implement a new smart contract based on the IVotilityReceiver interface, which has the following definitions:

- **onProposalFinished**: event fired when the user finalizes the proposal.
- **checkProposer**: the Votility Protocol calls the method to check if the sender can
- create a proposal. It is recommended to implement a permission list and/or a black list.
- **checkER20VotingPower**: the Votility Propotol calls the method to check if the specified ERC-20 token will be accepted as voting power. Normally, you can use only one token as voting power, however, perhaps the project has many options, such as liquidity pools or other underlying tokens, so this method will give you the power to expand.
- **getMinimumBlockLimitInterval**: in a fully autonomous organization, anyone can submit a proposal, including a malicious proposal. By setting the amount of minimum blocks, the developer can control how long the proposal will wait to complete.
- **getMinimumQuorum**: following the same idea as the previous method, this method will prevent a malicious idea from taking effect with a small amount of votes. The return is based on the total supply of the token used as voting power.

## 2.3. THE ONPROPOSALFINISHED EVENT

The event that finalizes the process is the call of the onProposalFinshed function. This function receives the parameters that characterize the proposal:

- **sender**: the user who made the proposal finalization call.
- **proposalId**: unique identifier of the proposal, it is used to get other proposal data in the main Votility Protocol smart contract call.
- **data**: array that corresponds to the input parameters. The input parameters are mapped into bytes32 variables that can be easily converted to other primary types.

- **winnerOptionIndex**: index of the winning option starting from zero.
- **winnerOptionData**: the content of the winning option.

For proposals that are based on Yes and No type answers, the index check (winnerOptionIndex parameter) is enough, for more complex proposals the conversion of the winner option content (winnerOptionValue parameter) will be required. And for those proposals that need to retrieve the voting result, just request the complete data through the proposal's unique identification (proposalId parameter).

### 2.4. VOTING POWER WRAPPER

Existing tokens that do not have the snapshot-based implementation can use the Voting Power Wrapper (VPW) tool, which is an ERC-20 token factory [10], with the necessary features for use on the platform and is created based on another ERC-20 token. The wrapper allows users with a balance in the original token to block the balance by generating an equal amount in VPW that will be used as voting power in an ongoing proposal, at any time the user can get back the original tokens by burning the VPW tokens. Once created, the VPW token will be available and can be reused in other proposals.

## 3. THE DAPP

With people's growing interest in blockchain technology, dApps (decentralized applications) are becoming increasingly important for developers around the world. A dApp is different from traditional apps in that it can connect developers and users directly, without the involvement of intermediaries [9].

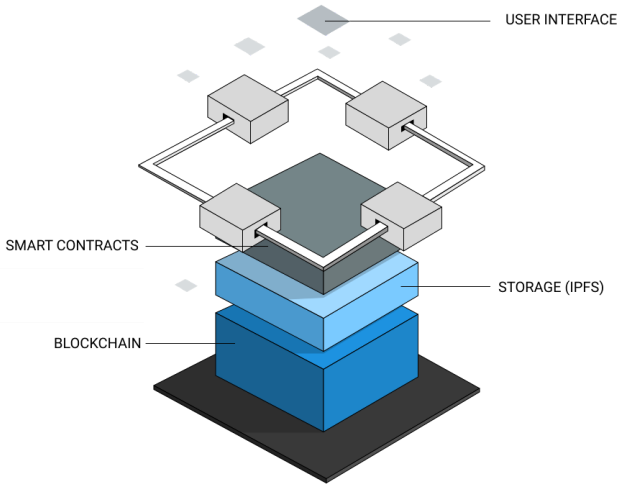


FIGURE 3: Representation of the composition of a dApp. Source: Adapted by the author of [9].

The Votility Protocol dApp contains two major components: the frontend and the backend. The frontend is the application that the user runs through the browser and is developed using the Vuetify[5] framework, which is a Vue.js[6] graphical interface library focused on Material Design[7]. The backend is responsible for processing off-chain votes and is developed using Nodejs[8], storing the data in the NoSQL MongoDB[9] database.

The frontend connects to the Ethereum blockchain via Metamask[10] (or another plugin compatible with web3.js[11]) and allows all on-chain operations without the need for access to the backend, in fact, if the proposal is configured to be executed on-chain only the frontend component is necessary since the metadata associated with a proposal is propagated on IPFS nodes, however, as there is the possibility of off-chain processing, the centralized backend is required for the complete solution.

The components are available in the project repository[12] and can be checked at any time.

## 3.1. PROPOSAL CONFIGURATION

To start in the dApp you need to connect to the Ethereum blockchain via Metamask (or another compatible plugin) as highlighted by figure 4.

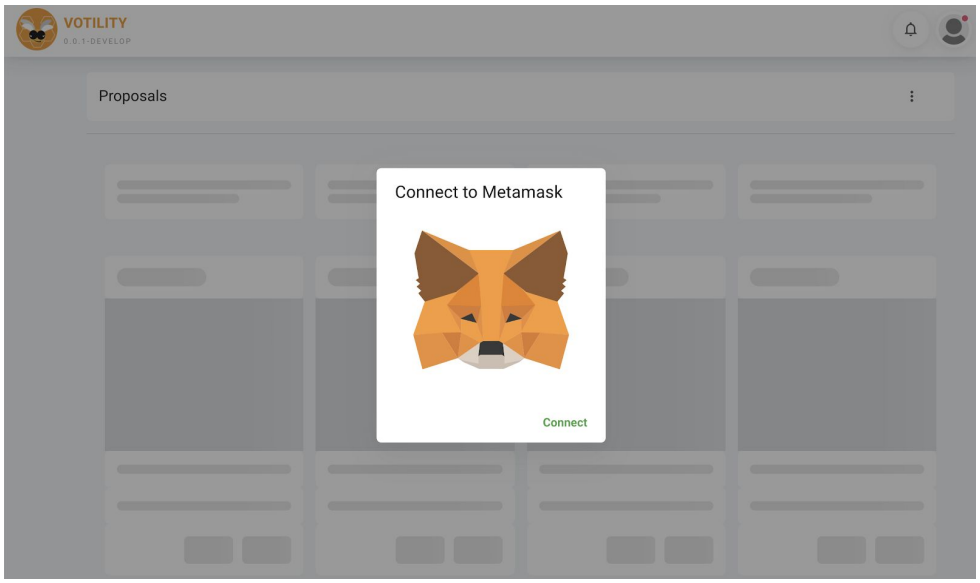


FIGURE 4: Connection of the dApp to the Metamask extension. Source: Author.

[5] Vuetify - Material Design Framework - https://vuetifyjs.com
[6] Vue.js - The Progressive JavaScript Framework - https://vuejs.org
[7] Material Design - https://material.io/design
[8] Nodejs - https://nodejs.org
[9] MongoDB A complete data framework - https://www.mongodb.com
[10] Metamask - A crypto wallet & gateway to blockchain apps - https://metamask.io
[11] web3.js - Ethereum JavaScript API - https://web3js.readthedocs.io/en/v1.3.0
[12] Votility Protocol Github repository - https://github.com/votility

After the connection is configured, the user can execute several activities, among them configuring a new proposal. Figures 5 and 6 show how the user can parameterize the proposal, mapping known types to a byte vector representation (bytes32 type in the Solidity language).

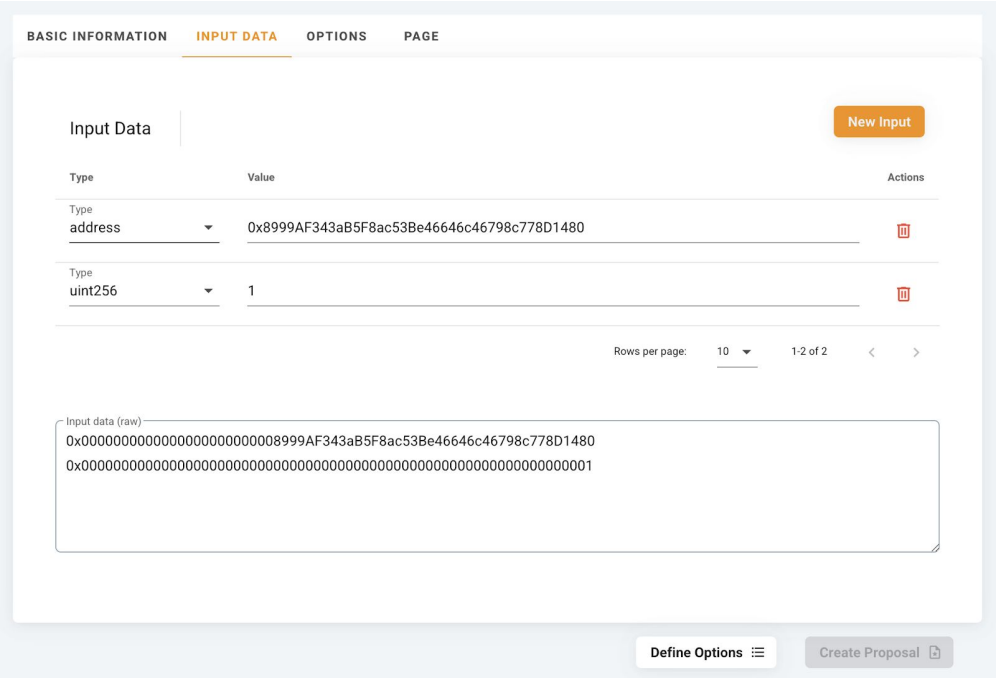


FIGURE 5: Configuration of the input parameters. Source: Author.

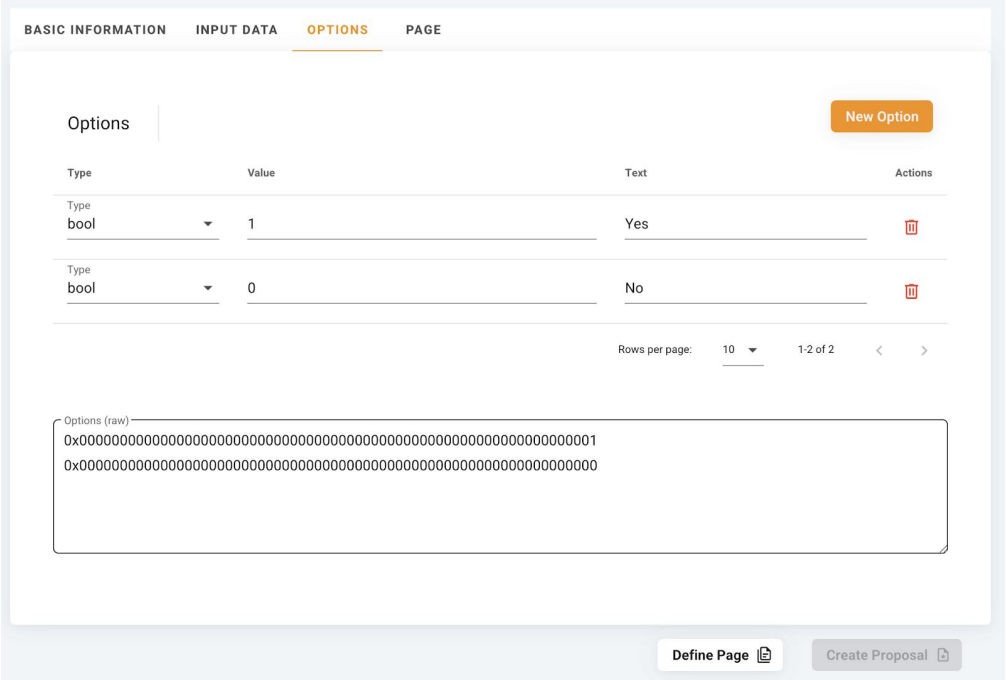


FIGURE 6: Configuration of voting options. Source: Author.

### 3.2. EVOLUTION OF THE PLATFORM: DAO MARKETPLACE AND DAO MAKERS

According to the implementation features, the proposed methodology, and the standardized form of creating new DAOs, it is natural to determine the platform's evolution path towards becoming a proposal management channel and a preconfigured DAO marketplace . Developers will either work on demand, develop customized DAOs and assume DAO Makers' role or create preconfigured solutions sending them to the marketplace.

## 4. VLTY TOKEN

The VLTY utility token, native to the protocol, will allow the user to configure new proposals and carry out the voting process if the voting is on-chain. The VLTY has been published on the Ethereum blockchain and has the technical specifications defined in Table 1.

| | |
|---|---|
| **Name** | Votility Protocol Token |
| **Symbol** | VLTY |
| **Specification** | ERC20 (*Pausable* and *Burnable*) |
| **Supply** | 100.000.000 VLTY |
| **Decimal places** | 18 |

TABLE 1 - Technical specifications of the VLTY token.

The VLTY is based on the ERC-20 standard [10], which defines a typical list of rules for tokens to follow within the platform, allowing developers to predict the interaction between tokens accurately. This standard allows the VLTY to be quickly integrated into any system that already operates in this way, making the token highly pluggable without significant technical efforts.

Every transaction within the Ethereum blockchain needs to pay a fee to be executed and validated by the miners. These fees are calculated in Gas, which is the execution fee for each transaction made on Ethereum. The value of Gas is expressed in ETH and is decided by the user. However, miners may refuse to process transactions with low amounts. When performing transactions on the dApp, the user will need ETH to transact.

### 4.1. VOTILITY MINING CYCLE

The off-chain voting process requires vote storage, responsibility of the Votility Protocol, and on-chain data entry, which can be performed by any user holding the VLTY token, i.e., member of the Votility Protocol DAO. Since this process requires the spending of ETH to compensate the miners, the user who performs the operation will receive VLTY tokens as a reward for the

transaction. The rewards will be distributed in blocks of 2,000,000 VLTY with initial rewards of 8,000 VLTY per on-chain data entry, with each block of 2,000,000 VLTY being a 10% reduction in rewards, in the last block the rewards will be 77.58 VLTY per on-chain data entry. The decreasing curve of the reward scheme will be as shown in Figure 7.

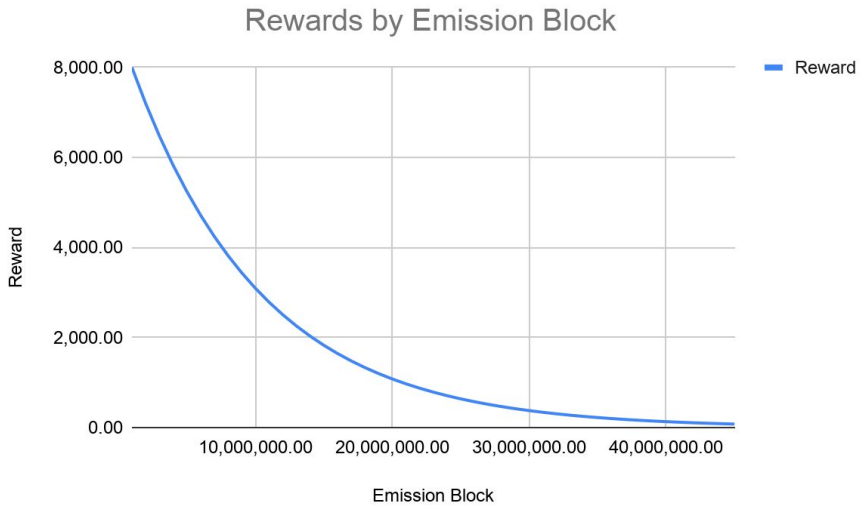


FIGURE 7: Representation of the reward emission curve. Source: Author.

The total rewards, corresponding to 45% of the total supply of the VLTY token, will be delivered after the completion of approximately 128,000 proposals. After this period, the reward for entry of tokens will be fixed and corresponding to half the proposal creation amount if an off-chain vote is chosen. Figure 8 shows the emission curve by the number of proposals completed.

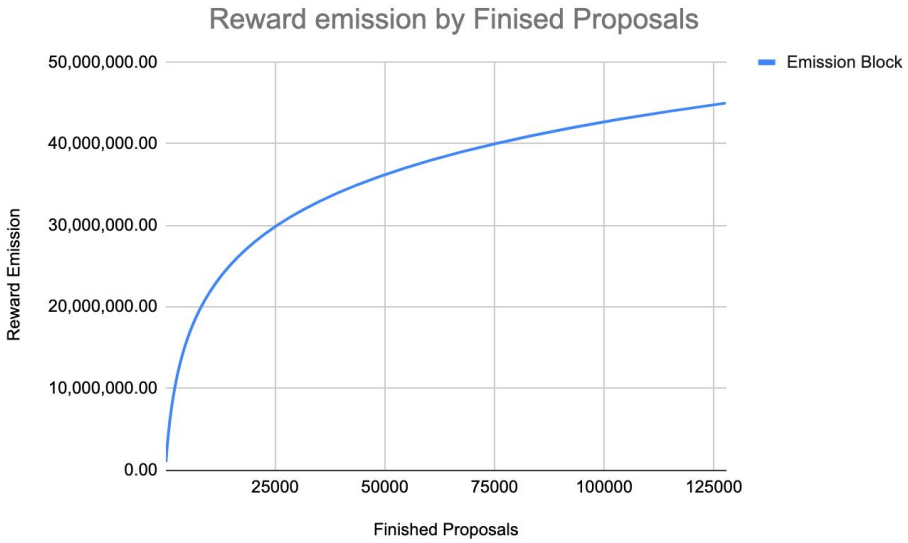


FIGURE 8: Representation of the emission curve versus the finalized proposals. Source: Author.

There is an intrinsic value by using Votility Protocol, as the protocol is decentralizing governance and putting users directly involved with the project evolution, however the reward model using the Votility Cycle Mining will bring another aspect of incentive for the execution of on-chain transactions. This rationale considers only the cost aspect, but the real value is to run and govern DAOs using Votility Protocol, which is intangible.

## 4.2. BURNING MODEL

The use of the token in the dApp will happen through the burning process at a rate defined by the community members, such rates will be differentiated between proposal creation and voting, when the configuration is on-chain. Each proposal creation and voting on-chain will reduce the total tokens available in the market, the VLTY can be understood as an utility access ticket to the existing and future functionalities of the platform. With the reduction of the total token supply the configuration of the fees by the community will ensure the correct pricing for access to the functionalities.

## 4.3. TOKENS DISTRIBUTION

The total supply of tokens will be distributed as follows:

- 15% reserved for the founders.
- 15% seed sale.
- 10% private sale.
- 10% airdrop.
- 5% platform development.
- 45% rewards for on-chain data entry.

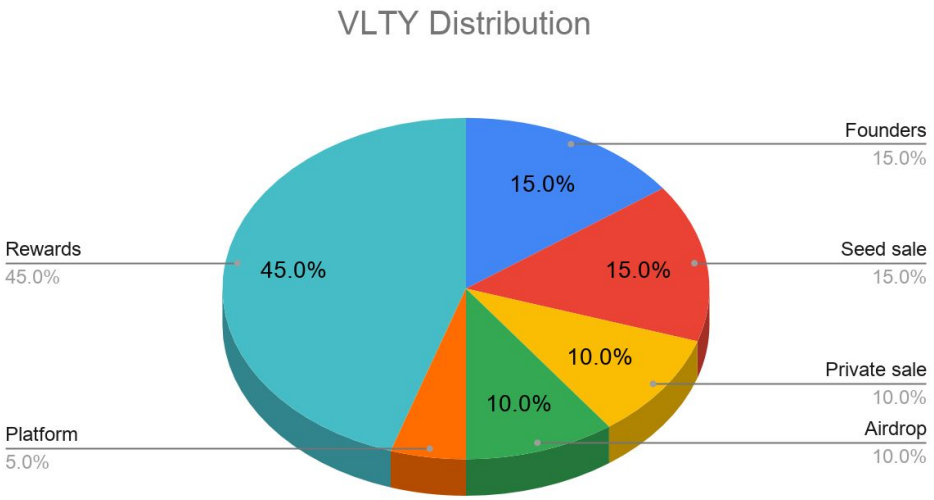


FIGURA 6: Distribuição de tokens VLTY. Fonte: Autor.

The seed sale stage will finance the first version of the project since it is an open-source non-profit initiative. The private sale stage and the amount reserved for the development will ensure the evolution of the platform and the allocation of resources necessary to keep the application running (material and human resources) over time. The project will use a percentage of the amount raised on the private sale to provide liquidity, allocating it in a Liquidity Pool. The amount reserved for airdrop will help the platform's dissemination, motivating the use and stimulating interest. Most will go towards rewards for the Votility Mining Cycle.

## 5. CONCLUSION

This paper presented the Votility Protocol, which is an extensible and parameterizable governance management platform focused on building autonomous decentralized organizations. The text discusses the fundamentals, methodology, technical details, and possible expansions of the ecosystem. Thus, the advantages of using building blocks to build decentralized applications are undeniable, as they reduce development time and deployment costs. These modular buildings create a dynamic and agile ecosystem, as new solutions are built quickly by relying on established and tested building blocks.

Because it provides a hybrid model of voting, on-chain and off-chain, there is a need to reward users who input results on-chain. To this end, the VLTY utility token is an integral part of the platform, as it not only guarantees access to existing and future functionality, but also serves as an incentive to the on-chain flow and fund the resources needed to pay the fees in ETH.

The platform is open and can receive contributions from the community.

# 6. REFERENCES

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system
[1] Chohan, Usman W., The Decentralized Autonomous Organization and Governance Issues (December 4, 2017). Available at SSRN: https://ssrn.com/abstract=3082055 or http://dx.doi.org/10.2139/ssrn.3082055
[2] The DAO: What Was the DAO and How Was it Hacked? (2021, January 27). Gemini. https://www.gemini.com/cryptopedia/the-dao-hack-makerdao
[3] Cambridge Dictionary. (2021, February 17). Building Block Meaning. Cambridge English Dictionary. https://dictionary.cambridge.org/us/amp/english/building-block
[4] Gamma, E. (1995). Design patterns: elements of reusable object-oriented software. Pearson Education India.
[5] B. (2018, September 8). BokkyPooBah's Gas-Efficient Solidity DateTime Library. Medium. https://medium.com/@BokkyPooBah/bokkypoobahs-gas-efficient-solidity-datetime-library-92bf96d9b2da
[6] Onyekwere, S. (2020, July 10). Building Blocks of DeFi (Decentralized Finance) - Sammy Onyekwere. Medium. https://medium.com/@sammyonyekwere/building-blocks-of-defi-decentralized-finance-2612cab3273d#:%7E:text=The%20fundamental%20DeFi%20building%20blocks,%2C%20Decentralized%20Exchanges%2C%20Oracle%2C%20e.t.c.]
[7] Nizamuddin, N., Hasan, H. R., & Salah, K. (2018). IPFS-Blockchain-Based Authenticity of Online Publications. Lecture Notes in Computer Science, 199–212. https://doi.org/10.1007/978-3-319-94478-4_14
[8] ERC 20 - OpenZeppelin Docs. (n.d.). OpenZeppelin. https://docs.openzeppelin.com/contracts/3.x/api/token/erc20
[9] L.H. (2019, April 25). dApp Development — A Simple Guide for Innovators and Entrepreneurs. Hackernoon. https://hackernoon.com/dapp-development-a-simple-guide-for-innovators-and-entrepreneurs-46922f98a6f2
[10] Vogelsteller, F. (2019, November 19). ERC: Token standard · Issue #20 · ethereum/EIPs. ERC: Token Standard #20. https://github.com/ethereum/eips/issues/20