UPPSALA
UNIVERSITET

UPTEC IT 22005
Examensarbete 30 hp
Juni 2022

Fair Voting System for Permissionless Decentralized Autonomous Organizations

Erik Hellström

Civilingenjörsprogrammet i informationsteknologi

UPPSALA
UNIVERSITET

Fair Voting System for Permissionless Decentralized Autonomous Organizations

Erik Hellström

Abstract

The increasingly adapted technology called blockchain can be viewed as a distributed append-only time-stamped data structure which is made possible by a distributed peer-to-peer network. The network uses cryptography and different consensus mechanisms to ensure immutability, security, transparency, and speed in a decentralized fashion. A permissionless decentralized autonomous organization (DAO) is an application deployed on a blockchain that enables people to govern and coordinate themselves in a decentralized manner through self-executing rules where anyone can join. A foundational function of a DAO is the voting system which dictates how the governance of the DAO is conducted. Voting systems in DAOs are currently not well researched and the currently used solutions have flaws, they are either not secure or they have the risk of resulting in unfair outcomes. This is the problem that this project focuses on. The problem was approached by conducting research in the field and through the conclusions of the research a new solution for a voting system was proposed and implemented. The proposed solution can be used to gain inspiration in further studies or be tested and developed to evaluate it in practice.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Uppsala

Handledare: Olena Burutina Ämnesgranskare: Riccardo De Masellis

Examinator: Lars-Åke Nordén

Sammanfattning

En blockkedja kan ses som ett verktyg för att spara och lägga till information. Till skillnad från de flesta tekniska system i dagens samhälle som är centraliserade, så är en blockkedja decentraliserad. Vilket innebär att det inte finns någon central entitet eller mittpunkt som bestämmer vad som är korrekt eller inte. Utan detta bestäms av hela nätverkets noder. En nod är i själva verket en dator som hjälper till att hålla igång blockkedjan och bygga nya block av information som läggs till. Användaren kan på ett säkert sätt interagera med blockkedjan med hjälp av kryptologi. Blockkedjornas omfattande användning av kryptologi ligger till grund för hur kryptovalutor fått sitt namn. Blockkedjors främsta användningsområde är transaktioner av kryptovalutor och är den initiala anledningen till varför blockkedjor kom till. Informationen på blockkedjan är då som en lista av transaktioner som håller koll på vilken person som skickar pengar till vem. Noderna i en blockkedja kommer överens om vad som är sant, när det kommer till informationen som läggs till i blockkedjan. För att komma överens använder de olika konsensus-mekanismer, det är olika procedurer som noderna tillsammans utför för att behålla integriteten av blockkedjan. Tillsammans med den underliggande teknologin och dessa consensus-mekanismer så åstadkommer blockkedjorna oföränderlighet, säkerhet, transparens och hastighet på ett decentraliserat sätt.

För att ta till vara på fördelarna som blockkedjor har i andra syften än transaktionssystem så uppfanns decentraliserade applikationer. I decentraliserade applikationer sparar man programmeringskod på blockkedjan så att ens program kan köras på ett decentraliserat sätt och på så sätt dra nytta av fördelarna av att ha ett fullt decentraliserat system. Ett mål med decentraliserade applikationer är att kunna ha en applikation som är fri från underhåll och styrande från utsidan. Alltså en applikation som är autonom och fungerar utan mänsklig interaktion. En sådan applikation kallas för en decentraliserad autonom organisation (DAO). Styrandet av en DAO refererar till hur beslut tas och hur makten är fördelad. En DAO är självstyrande av en samling regler och ingen central entitet kan ändra dessa regler då de är utplacerade på blockkedjan. För att användare ska kunna bidra till styrandet av en DAO så måste det finnas något sätt att registrera användarnas kollektiva preferenser, det vill säga någon form av röstning. Röstandet i en DAO är för närvarande inget väl utforskat område och de nuvarande lösningarna har brister, antingen så är de inte säkra eller så finns det en risk för att de leder till orättvisa utfall. Detta är problemet som detta projekt fokuserar på. För att undersöka problemet så studerades befintliga lösningar och en ny lösning föreslogs och implementerades.

Den implementerade lösningen går ut på att användarna röstar med sina kryptovalutor

men för att göra det rättvist och säkert så justeras vikten av rösterna. Vikten av en röst räknas ut med något som kallas *Quadratic Voting* samt *Coin Age*. *Quadratic Voting* innebär att man tar roten ur mängden av kryptovalutor som en användare äger för att jämna ut makten mellan rika och fattiga användare. *Coin Age* innebär att vikten av en röst baseras på hur länge användaren har ägt kryptovalutorna. Detta medför ett värde av investerad tid i en DAO. *Coin Age* kan även mildra risken för attacker där man distribuerar kryptovalutor till flera användare för att få mer makt. Den implementerade lösning är inte redo för produktion utan det är tänkt att andra ska få inspiration och utvärdera lösningen i praktiken.

Contents

List of Figures

List of Acronyms

DAO	Decentralized Autonomous Organization
Dapp	Decentralized Application

Glossary

Cryptography Cryptography is the practice and study of techniques for secure communication. It is used to achieve different aspects of information security such as data confidentiality, data integrity, authentication, and non-repudiation [6].

Hash The result of a hash function which performs a one-way compression of data of any given length to a fixed-length hash value. The hash value can be used as a digital fingerprint for the data which was compressed [1].

Governance Governance is the process of political administration, including management of public resources, using political authority to exercise administrative power in a defined environment [30].

Blockchain A blockchain is a distributed database that keeps track of transactions by organizing them into a chain of blocks. The blockchain is maintained by a peer to peer network and cryptography is used in order to maintain its integrity and immutability [48].

Cryptocurrency Wallet or Wallet Users of blockchain store their private keys in a cryptocurrency wallet and uses the wallet to interact with and manage their investments and transactions [2].

Decentralized Autonomous Organization A DAO is an application deployed on a public blockchain that enables people to coordinate and govern themselves through a set of self-exectuting rules. The governance of the DAO is decentralized, no central entity has control over the application [26].

1 Introduction

Blockchain was first introduced by the anonymous person behind Bitcoin Satoshi Nakamoto in 2008 and it was created in order to solve the problem of double-spending. Double spending is when a person spends their assets two or more times. For example, person A sends X amount of an asset to person B and C while person A's balance is only deducted with the amount X once. A solution to this was to create a peer-to-peer network with timestamped transactions by hashing them and putting them into a block that contains the hash of the previous block, creating a chain. The validity of a block on the Bitcoin blockchain is controlled by the majority of CPU power in the network. According to Satoshi Nakamoto, this would be more robust than present financial solutions at the time and reduce transaction costs [38]. Since blockchain was introduced in 2008 it has continued to gain traction and its use case have been increased and developed over time. Blockchain can be viewed as a distributed append-only time-stamped data structure which is made possible by a distributed peer-to-peer network [43]. The network uses cryptography and different consensus mechanisms to ensure immutability, security, transparency, and speed in a decentralized fashion.

In order to take advantage of the blockchain to do more exciting things than just serve as a transaction system, smart contracts were developed [7]. A smart contract is a way of implementing decentralized applications (Dapps) with the use of blockchain [41]. Today there exist multiple different blockchains that enable the use of smart contracts but the most popular one is Ethereum [33]. Smart contracts are based on the idea that a legal contract could be notarized as well as executed automatically with the use of blockchain. Given that the program is deployed on the blockchain it will be immutable due to the nature of blockchains. This means that once a smart contract is developed and then deployed to the network, it will not be possible to revise the logic of the program [7]. By running the code on the blockchain the application gains all the advantages of having a decentralized system, contrary to the majority of applications in today's society which are centralized. These benefits are the following:

- Data is immutable since it is stored on the blockchain.
- The application and storage of data are fault-tolerant since the system is running on thousands of different machines, there is no single point of failure.
- Transparency, everyone has the possibility to inspect the blockchain and inspect the code that is running on it.
- Seamless integration with payment functionality as this is the initial role of the blockchain.

- Anonymity, people are represented by their cryptographic keys.

The ultimate goal of running an application on the blockchain is to have an application that is completely hosted by the blockchain network. An application that is in no need of maintenance or governance from the outside. Meaning that the application is in a way autonomous and works without human interaction [7]. This is referred to as a decentralized autonomous organization (DAO). One could say that it is like any other decentralized application but a bit more complex in that it allows for a governance system to be used. A permissionless DAO is a DAO where everyone is free to join. The governance in a DAO refers to the way in which decisions are made and how the power is distributed. The DAO is self-governed by a set of self-executed rules, and no central entity can change these rules as they are written in code deployed on the blockchain [26]. This basically means that the code runs the show and no one person can change it, this is where the decentralized governance comes in. In order for the users to be able to have a say in the decisions that are being made and rules that are set there has to be some way to record the collective opinions of all the users in the DAO, which is some form of voting. A foundational function of a DAO is the voting system which dictates how the governance of the DAO is conducted.

Voting systems in DAOs are currently not well researched and the currently used solutions have flaws, they are either not secure or they have the risk of resulting in unfair outcomes. This is the problem that this project focuses on. The problem was approached by conducting research in the field and through the conclusions of the research, a new solution for a voting system was proposed and implemented. The proposed solution can be used to gain inspiration in further studies or be tested and developed in order to evaluate it in practice.

This project is done with a company called ChromaWay as the external party. ChromaWay is a blockchain technology company that has created a new blockchain architecture called relational blockchain which combines the benefits of a relational database together with the fault-tolerant decentralized security of a blockchain [13].

1.1 Problem Statement

DAOs are an exciting topic that many are convinced will revolutionize the way in which organizations and possibly even companies are structured and governed. However, DAOs are still a fairly novel concept and in order to achieve the possible benefits which can theoretically be achieved with a DAO further research needs to be done and more DAOs need to be created and tested [26].

As previously stated a DAO is in no need of central authority or governance, the governance is decentralized across the users and the code runs the show. Given this premise, it is made clear that the governance relies heavily on the ability to record the collective opinion of the users. In other words some form of voting. The specific problem that this project will focus on is the voting procedure within DAOs and how it can be made fair. Fair in the sense of making it as democratic as possible or equalize everyone's vote weight as much as possible. During the project it is made clear that current DAO platforms often use very different approaches to this problem. Currently, there is no clear approach which is the best one, at least in the general case. Blockchain inherently handle cryptocurrencies and their transactions of them very well and secure which is why most of the current DAO platforms use a coin-based voting system, that is people vote with their money. For example, in a democracy, we take for granted that one person is eligible to one vote. With coin-based voting this is not the case, people with more money have more voting power, which is not really fair and can lead to problems and vulnerabilities in the DAO. It should be noted that a purely coin-based approach could be considered fair in certain cases.

Now the problem with a permissionless DAO, meaning anyone can participate and anyone can create a new wallet and interact with the DAO is that one person - one vote is difficult to achieve as it makes the DAO very susceptible to sybil attacks. A sybil attack is when an attacker creates multiple agents in order to act as multiple people in this case with the intent to gain a large influence on the DAO and get malicious proposals to pass the voting system.

Another approach is to use something other than a coin-based approach for the voting system. However, it is not straightforward how such an approach should be implemented. There are many factors that need careful thought, consideration, and research. The coin-based approach has apparent flaws and opting for another approach is difficult but the purpose of this project is to investigate possible solutions.

The entire governance of a DAO will not be considered, only the actual voting system where votes are cast by members on proposals which in turn will dictate how the DAO should be governed.

The goals of this project are the following:

- Conduct a research study on the current state of the research within the field, specifically targeting currently available DAO platforms, focusing on their voting systems.
- Attempt to formulate and implement a new approach for a fair voting system based on the findings from the research.

What is currently occurring in the DAO space is that new innovative forms of governance are taking place. However, the people that create these forms of governance rarely do so with prior extensive testing and research. These aspects result in DAOs being a difficult research field in their current state. It is particularly important to look at how the voting systems within these DAOs function and how well they work in their environment [21].

The available DAO platforms or frameworks that currently exist provide approaches, conceptions, and tools for building DAOs but they rarely share or attempt to share the underlying technical concepts in their platforms. Which in turn makes it difficult for normal people or beginners within programming or blockchain technologies to join the DAO space, at least when it comes to implementing or understanding the underlying technology [45]. This is another reason why research regarding the topic is important to evolve the technology further.

2 Background

2.1 Blockchain

Blockchain can be viewed as a distributed append-only time-stamped data structure. It is made possible by a distributed peer-to-peer network [43]. Blockchain has in past years gotten a lot of attention from a broad range of industries as well as private people. Before blockchain, it was very difficult to let certain applications that were in need of trust and security to run in a decentralized fashion. These applications were previously needed to have a trusted intermediary, a central source of truth. It is now possible for these applications to achieve the same security and truth but in a decentralized fashion, which they could not do in the absence of blockchain technology. The way in which blockchains are engineered leads to often much-desired properties such as security, auditability, robustness, and transparency [11]. It achieves this through its clever architecture and its extensive use of cryptography. A blockchain can be viewed as a distributed database that is structured as a linked list where every element in the list is a block of transactions and every block is immutable, so the data which is stored on the blockchain is immutable. Through cryptography all users become anonymous but the blockchain is still transparent since it is open for everyone to audit its history. Given the positive properties of blockchains, it is obviously a very attractive solution for banking and financial services. There are also other exciting use cases, which will be discussed in this report.

As previously mentioned each block in the blockchain contains the cryptographic hash

value of the previous block, which creates a chain of blocks, see figure 1. The only block which does not point to the previous block is the first block, called genesis [11]. This fact is one of the reasons why it is very hard for one malicious attacker to alter the chain of blocks, because if one block is changed all subsequent blocks also need to be recreated. This entire chain of blocks can be read by anyone which is where the transparency comes in. The entire chain of blocks is also stored on multiple nodes all over the network. This is another reason it is so difficult to alter a blockchain, there is no single point of failure [11]. In order to understand how the blockchain works, we need to understand what a node is in the network. A node keeps a copy of the blockchain and tries to add new blocks to the blockchain, one could say that a node is a gateway in which transactions are added to the blockchain. All nodes together form the complete peer-to-peer network which enables the existence of the blockchain [11].

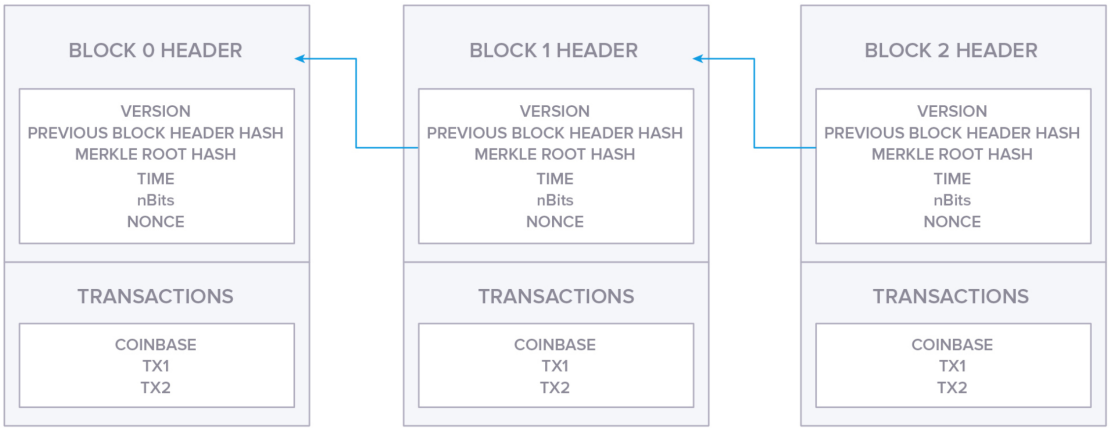

Figure 1 Block structure [27]

Every user or account which is existing on the blockchain is represented by its public and private keys. These keys are generated using cryptography and it is a way to communicate securely across the internet. These keys are generated randomly with asymmetric encryption. The owner of the private key can encrypt data so that the public key can decrypt it, by doing so the person with the public key knows that the encryption was done by the owner of the private key. Anyone with the public key can also encrypt data that only the private key can decrypt. This results in the possibility of completely encrypted communication on the internet. Users on the blockchain are represented by their public key and when people receive transactions on the blockchain they are sent to one's public key. The private key is as the name implies private and is

used to send transactions on the blockchain [34]. It is very important to store these private keys in a secure location in order to not lose them or let anyone steal them because the private key is all that a person needs in order to steal the funds connected to it. There are multiple different approaches for storing private keys. In order to not get hacked the most secure way is generally to have a *offline key storage* which is when the private key is stored on an offline device or some kind of object, for example, a paper. This method drastically reduces the risks of getting hacked but it will not provide quick and easy access to the private keys. For this reason, it might be a good idea to have another account with a smaller amount of assets, where the private key is stored on a less secure but more easily accessed online storage or wallet [39].

If a user wants to send some assets to another user it will need to sign the transaction with its private key. After a user has signed their transaction it is then added to a node that is broadcasting the transaction to its one-hop peers in the network. Before sending the transaction further in the network the peers control that the transaction is valid. One might wonder what it means for a transaction to be valid and in what way this is determined. A transaction is when person A sends assets to person B [9]. One thing that can make the transaction invalid is if person A tries to send the same assets to person C, also known as double spending which is what Satoshi Nakamoto solved with Bitcoin [38]. An important property of a valid transaction is of course also that the correct amount is added to the receiver and the correct amount is deducted from the sender. After a while, the transaction has been sent across the complete network and all transactions that has been broadcasted during a given time frame are put inside of a new block. The process of creating new blocks is referred to as *mining*. The newly mined block is sent out through the network. All nodes in the network go through all transactions in the block and ensure that they are valid, they also check that the block points to the previous block in the chain. If the new block is valid every node adds it to the chain and the global state of the blockchain has been updated [11].

2.1.1 Consensus Mechanisms

The rules which determine if a transaction is valid or not depend on which kind of consensus mechanism the blockchain uses [9]. All the nodes in the network need to come to an agreement on which transactions are valid and should persist in the blockchain. This is done through a consensus mechanism [18]. It is important to remember that the blockchain network is built with multiple nodes that do not need to trust each other and as mentioned before there is no central source of truth. In order to prevent chaos and multiple different forks with different states of the blockchain, there needs to be some consensus mechanism for all the nodes to reach a consensus on which transactions are valid [11]. A fork is when the blockchain is being split into two

or more different blockchains because the nodes could not reach a consensus. The consensus mechanism that might first come to mind is simply to let all the nodes vote on which transactions are valid or not, this would need way less computation power and reduce the environmental impact that certain consensus mechanisms cause. The problem with going with this approach is that it is extremely susceptible to sybil attacks. Sybil attacks are when an attacker creates multiple different fake identities or in this case nodes in order to gain a majority vote regarding the voting of the consensus mechanism [47]. Whenever it is possible to freely create new identities, accounts, or sources of votes you must be aware of the risk of sybil attacks and mitigate it in order to keep the integrity of the system [47].

There are a lot of different consensus mechanisms out there and this section will describe a few of them. The most popular consensus mechanism is *Proof-of-Work*, this is used for Bitcoin [38]. *Proof-of-Work* is what it sounds like, the mining node needs to perform some kind of work and be able to prove that it has been done. This is what one means when mentioning that the most amount of CPU power dictates the truth on a blockchain, in contrast to the most amount of nodes in the example mentioned above. In public blockchains, anyone has the right and ability to join the system as a user as well as to join the network of nodes, as previously mentioned this makes public blockchains susceptible to sybil attacks. With *Proof-of-Work* sybil attacks are prevented in blockchains [44].

The work in *Proof-of-Work* is usually some kind of computational process like guessing which hashes result in some value. The point of using *Proof-of-Work* is to make it difficult, time-consuming, and costly to mine new blocks, which in turn makes it really hard to create fake blocks. All miner nodes are doing this work simultaneously, trying to create the new block. The one which succeeds to create the new block gets a reward and the new block can be verified by all the other nodes and then added to the blockchain. The reason why this works is that in order for a malicious node to create a bad block it would need to compute all subsequent blocks as well. The entire networks nodes will also go through the new block and verify that everything is correct, including all new transactions. This means that miners cannot cheat. If the new block does not pass the verification it will be discarded and the miner has wasted its time and resources on creating the malicious block. The consensus mechanism is a key part of what makes blockchains secure, robust, and trustworthy [25]. With *Proof-of-Work* the ability to verify transactions is dependent on the computing power instead of the number of nodes. The incentive that miners have is that they are rewarded with assets whenever their block makes it onto the blockchain, in the case of Bitcoin they are rewarded with some new Bitcoin and with the transaction fees paid by the people doing transactions. The approach of *Proof-of-Work* is based on the basic assumption that it is increasingly more difficult to obtain the majority of computing power in the network than it is to control

the majority of nodes [44]. This approach also has the ability to create a scarcity of the asset. If we look at Bitcoin, the way in which new Bitcoins are made is through the mining reward. As computational power and Bitcoin gain traction the work in the consensus mechanism is solved faster and faster. In order to maintain the scarcity of newly created Bitcoin, there is a difficulty put in place that is adjusting itself so that new blocks are only created once every 10 minutes [44]. See figure 2 where Bitcoins blockchain has been analyzed to see how the difficulty is adjusted over time. The miner reward is also halved every 210000 blocks which increase the scarcity, much like gold or diamonds are a scarce commodity in the real world. The scarcity is wanted to create value in the asset, a value which is backed up by computational resources in the real world [44].

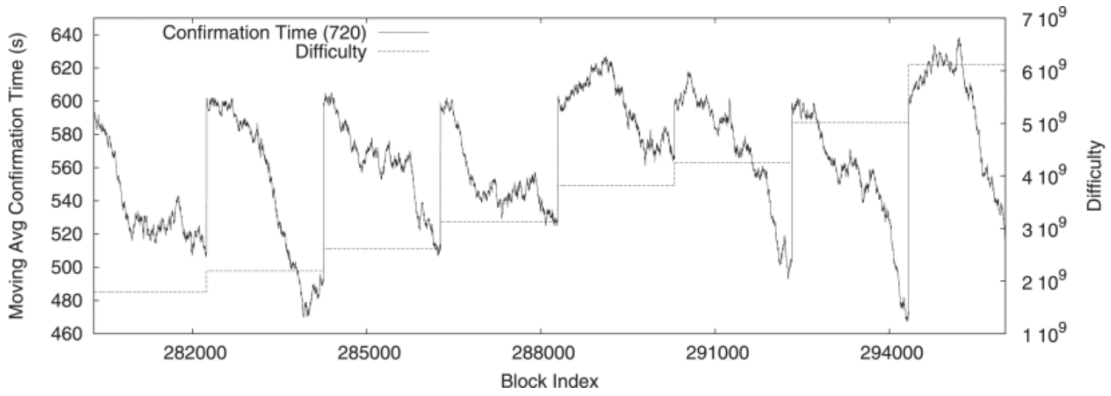

Figure 2 Difficulty of the work in Bitcoins consensus mechanism being adjusted over time in order to maintain the scarcity of the asset [44]

Since *Proof-of-Work* is designed to use a huge amount of computational power, it of course uses huge amounts of electricity. The enormous energy consumption of certain blockchains is one of the main arguments against the use of blockchains. In 2019 Bitcoin mining in Asia alone used up in 33.2Twh [3]. There are also other drawbacks of using *Proof-of-Work*. For example, the fact that the rewards for mining are decreasing as time goes on, after a while, the rewards will be so small that miners will lose some of their incentive and risk leaving the network [44]. The miners will still have the incentive of transaction fees, the fee that the miners charge from the people of which transactions are being processed. An alternative to *Proof-of-Work* is *Proof-of-Stake*. A stake is an amount of cryptocurrency that a user has locked up in exchange for the chance to validate new blocks on the blockchain. *Proof-of-Stake* also introduces a concept called *Coin Age* the amount of time that has passed since the cryptocurrency was staked [18]. The *Coin Age* is used to determine the power that the

node has within the network when it comes to reaching consensus. The longer a node stakes the currency the more power it obtains. The *Coin Age* also determines the reward of creating a new block [18]. *Coin Age* also determines the difficulty of the work needed to be done in order to create the block. This way *Coin Age* could be seen as the actual resource rather than computing power as seen in *Proof-of-Work* [44]. The staked amount together with the *Coin Age* determines the chance a node has to be picked to validate a new block, so only one block does the work of validating the block. This work that is done can be compared with the work that all of the nodes do on a *Proof-of-Work* blockchain. The picked node is then either rewarded for validating only valid transactions or punished for validating invalid transactions. One way to explain why *Proof-of-Stake* works is because the validating node has so much faith in the validity of the new block that it is willing to back it up with its own money. A malicious node can basically only lose if it does something malicious. This is how *Proof-of-Stake* can provide a solution to the fact that blockchains waste a lot of energy, by drastically reducing the need for computational power which certainly is beneficial to the environment [44]. It can also be argued that *Proof-of-Stake* combats the issues of the rich getting richer and instead results in the poor getting richer since anyone can provide a stake and the rewards are linearly proportional to the amount you are able to obtain and hold [44]. Another argument for using *Proof-of-Stake* is that it seems like it is far more difficult to attack. Since it would be very expensive to acquire the majority of the assets in the network, and if the attack is successful the digital asset would almost certainly lose all of its value. It would basically be very inefficient to attack the network when you look at what outcomes the attacker could reach [44][18].

What kind of consensus mechanism a blockchain chooses to use is often based on the blockchains use case and desired properties. For example, *Proof-of-Work* and *Proof-of-Stake* will not work for a smaller blockchain as it is very easy to gain a majority in that case. If very low latency is desired then these two methods will not be the best either, at least for most implementations [20]. Another consensus mechanism that is worth mentioning is *Proof-of-Authority* which is when you limit the number of validator nodes in the network and you make sure that you choose trusted nodes [20]. Going with this approach you trade some of the decentralization to gain some of the benefits of a centralized system [12].

Together with a consensus mechanism and the underlying design previously discussed blockchains are a way to obtain immutability, security, transparency, and speed. Immutability is reached because it is impossible to alter the passed blocks in the blockchain. Security is reached through extensive use of cryptography together with the consensus mechanism which validates transactions. Transparency is achieved by the decentralized nature of the blockchain, anyone can access all of the transactions. When compared to current banking systems and other ways of securely sending assets

globally, blockchain achieves much faster transaction speeds [32].

2.2 Decentralized Applications

In order to take advantage of the blockchain to do more exciting things than just serve as a transaction system, smart contracts were developed [7]. A smart contract is a way of implementing decentralized applications with the use of blockchain [41]. Today there exist multiple different blockchains that enable the use of smart contracts but the most popular one is Ethereum [33]. Smart contracts are based on the idea that a legal contract could be notarized as well as executed automatically with the use of blockchain. Programming languages like Solidity, which is the language for the blockchain Ethereum can be used to implement different smart contracts which in practice is a program that is deployed on the blockchain [7]. Given that the program is deployed to the blockchain it will be immutable due to the nature of blockchains. This means that once a smart contract is developed and then deployed to the network, it will not be possible to revise the logic of the program. The beauty in this is that the program becomes transparently visible to the public and the logic can be trusted since it will not change [7]. The way in which smart contract executions can be verified by any node in the decentralized blockchain network can also be trusted in the same manner as any transaction can be trusted on the blockchain. Which in turn allows for the absence of a central point of failure and truth [41]. It is worth noting that when a user interacts with a smart contract and some sort of writing action is done to the blockchain, then all the nodes need to run that code. However when a reading operation is done then it is only the node to which the user is connected that needs to run that code. A smart contract is one way of implementing decentralized applications, also referred to as dapps [7]. A smart contract enables functions that can be used by the rest of the blockchain's users, kind of like an API (application programming interface) which is hosted by the blockchain itself [41]. The need for decentralized applications might not always be obvious but it offers something that is lacking in past technologies such as web, cloud, and others [41]. The ability to have decentralized applications allows for many interesting implementations, probably many of which we have yet to observe. By running the code on the blockchain the application gains all the advantages of having a decentralized system, contrary to the majority of applications in today's society which are centralized. These benefits are the following:

- Data is immutable since it is stored on the blockchain.
- The application and storage of data are fault-tolerant since the system is running on thousands of different machines, there is no single point of failure.

- Transparency, everyone has the possibility to inspect the blockchain and inspect the code that is running on it.
- Seamless integration with payment functionality as this is the initial role of the blockchain.
- Anonymity, people are represented by their cryptographic keys.

2.3 Decentralized Autonomous Organization

The goal of theoretical dream of a blockchain application is an application that is completely hosted by the blockchain network. An application that is in no need of maintenance or governance. Meaning that the application is in a way autonomous and works without human interaction [7]. The definition of a decentralized autonomous organization (DAO) can differ depending on the source. The definition that seems to be used most broadly is some version of the following: "A DAO is a blockchain-based system that enables people to coordinate and govern themselves mediated by a set of self-executing rules deployed on a public blockchain, and whose governance is decentralized (i.e., independent from central control)." [26]. One could say that it is like any other decentralized application but a bit more complex in that it allows for a governance system to be used. The governance in a DAO refers to the way in which decisions are made and how the power is distributed. The DAO is self-governed by a set of self-executed rules, and no central entity can change these rules as they are written in code deployed on the blockchain. This basically means that the code runs the show and no one person can change it, this is where the decentralized governance comes in. In order for the users to be able to have a say in the decisions that are being made and rules that are set there has to be some way to record the collective opinions of all the users in the DAO, which is some form of voting.

There are people out there that would argue that the Bitcoin blockchain network is the actual first DAO since it fits all the criteria [7]. Although the term is today generally used to describe an organization or application deployed on top of a public blockchain, rather than it being the blockchain network itself [26]. Possibly one of the more beautiful attributes of a DAO is the fact that there is no central authority to shut it down. As long as members are active within the DAO, the DAO will exist [21]. The first DAO that really gained some attention was a DAO called *The DAO* which was created in 2016 [26]. After that, there have been multiple popular examples, some of which are discussed in section 3. As previously mentioned the definition of a DAO has never really been established and some academic studies deliberately try to avoid using a certain definition. Samer Hassan has in his paper *Decentralized Autonomous Organization* [26]

compiled the main points that usually occur within the used definitions of a DAO within academic literature:

- DAOs should enable a group of people to coordinate and self-govern themselves online.
- The source code which enables the DAO should be deployed on a blockchain, preferably a public blockchain. Meaning that anyone is able to join and participate in the blockchain.
- A DAOs smart contract or code specifies the rules of how people should interact and govern themselves.
- The rules are self-executed independently of the will of any party within the DAO.
- No central control is able to influence the governance of the DAO.

A DAO can be used for a lot of different things. At first glance, it might sound that the use of DAOs is fairly limited to certain use cases but in fact, there are many different functions that a DAO could fulfill [26]. For example, a DAO could be a company, the backend of a video game, a communication platform, or a simple way for a group of people to make decisions. Basically, any application where a central intermediary has previously been needed can now often be replaced by DAOs. When it comes to technical online solutions most of them have previously needed some sort of centralization. DAOs are gaining a lot of traction and it is a really exciting business to be involved with [28].

There are some really interesting legal questions regarding DAOs. If a DAO is really autonomous, who is legally responsible for the actions of the DAO? There are debates about whether a DAO should be seen as its own legal entity or if the people legally responsible for its actions are the people that participate in the DAO [26].

As previously mentioned there are some uncertainties regarding the definition of a DAO. It is unclear if a DAO should be able to be completely isolated from human interaction in order to work as intended, basically how autonomous does it has to be? It is also unclear how decentralized the DAO needs to be, should the government of the DAO be completely decentralized or not? [26]. This report will use the definition that was stated at the beginning of this section.

A fast-growing genre within decentralized applications is video games. Video games fit the nature of blockchain technologies very well [7] and it is a good example to look at if you want to get a good idea of the possibilities that decentralized applications and DAOs can bring to the table. Imagine a game where the items and assets obtain real value like a cryptocurrency does on the blockchain in a decentralized manner such that

as long as people are playing the game, the game will exist. The governance of the game would also be given to the players so that the players decide on what changes should be implemented in the game. This would bring meaning and a sense of realness to the gaming experience.

DAOs are still very novel technological systems where the field is being explored and theories are being tested [21]. As previously mentioned there are some uncertainties when it comes DAOs and both legal and technological research needs to be done within the field.

3 Related work

There are a lot of DAOs in the making and the entire blockchain industry is extremely active at the moment but there is not a sufficient amount of research being done on this specific subject. The specific subject refers to the voting procedure of DAOs, more specifically how the votes are registered and weighted in order to create a fair system. Developers are really thinking about these questions and DAOs are being created while trying to figure out these problems. However, since this is a new area of technology there is not that much scientific research available. Nevertheless, there are similar topics like computational social choice which is a well-studied area and is described in section 3.3. There are also interesting studies being done analyzing current DAOs and their voting systems, but very few of them focus on the specific topic that this report is trying to tackle though the act of voting is often referred to as just “voting“ and no more details are given.

3.1 Voting Systems

This subsection describes a few voting systems that have been used on existing DAO platforms.

3.1.1 Voting with Holographic Consensus

There is a common scaling problem when it comes to the voting procedures in DAOs. As the DAO grows, so will the number of proposals that need to be discussed and voted upon (if members are allowed to create proposals). However, the members will still have the same amount of time that they can spend on the voting procedure, which certainly won’t be enough. One way to combat this problem is to use something called

Holographic Consensus, introduced to the DAO space by DAOstack, described in section 3.2.2. In *Holographic Consensus* members can create proposals, which need an absolute majority to pass, that is more than 50% of all available votes. There is also another way to pass votes, *Holographic Consensus* introduces a sort of betting layer where members are able to bet on proposals if they are going to pass or get rejected. If the number of bets reaches a certain amount of staked tokens the proposal will be able to pass with a relative majority, that is a majority of the cast votes. This way the DAO will focus on the proposals which attract the stakers. The work done by the betting members filters out proposals and allows for a more smooth, efficient, and scalable voting system. The DAO can in theory trust the betting members as they are incentivized to be aligned with the DAOs opinions since it will financially reward them [21].

A study was done to analyze the effectiveness of *Holographic Consensus* on the DAOstack platform. The study showed that it is working as intended. For larger DAOs the proposals fairly often skip the majority voting requirement by members betting with a pretty high prediction rate, over 90% [22].

3.1.2 Conviction Voting

Conviction Voting is a voting system that tries to represent the aggregated opinions of members in a DAO, recorded and expressed continuously. That is the voting period is not within a small window of time but recorded over time. Members vote on a proposal and the longer they keep their voted stakes (a locked amount of the asset) the more their vote will grow and they show a greater conviction towards the proposal. Members can at any time change their vote however when they do so the conviction of the vote is reset. This voting system can be used on a DAO created on the Aragon platform, described in section 3.2.1 [21].

3.1.3 Reputation Voting

There are multiple different ways in which a reputation voting system can be used. The main idea is to move away from the more commonly used coin-based systems. A DAO which uses *Reputation Voting* uses a non-transferable reputation token which can then be either counted as one vote or can be used as a ticket for the member to vote [45]. The DAO framework Colony uses a reputation-based voting system, described in section 3.2.4.

3.1.4 Quadratic Voting

Quadratic Voting has been discussed within the field of Social Choice for quite some time and it is an interesting concept that can be used for voting in DAOs. In general, outside the DAO space *Quadratic Voting* was introduced and used in order to eliminate the downside of having one vote per person. This arguable downside is that in a voting scenario with multiple different alternatives to vote on, the voter is limited to only showing their support for one alternative. Possibly more importantly it does not enable the person to express intensity in their preference [31]. The way in which *Quadratic Voting* tries to solve these issues is by letting people pay to vote as many times as they like but at an escalated price for each vote. The cost for a number of votes is the quadratic value of the number of votes, so the cost for more votes rapidly escalates. This way a majority of voters who barely care about an issue will not win against a little bit smaller group of voters who intensely believe in the preference [40]. Within DAOs and the blockchain world *Quadratic Voting* can be used in a public blockchain to achieve the same results as mentioned above. However, the reason behind using it is to reduce the power of the richest token holders when transitioning from a coin-based voting system in contrast to the example above where it is used to transition from a voting system where one person has one vote. *Quadratic Voting* is used by a DAO platform called Illuvium [29]. One big drawback from using *Quadratic Voting* for the voting system in DAOs is that it opens up the risk for sybil attacks. Members can simply create multiple accounts and distribute their assets in order to bypass the reduction of the votes. In order to use *Quadratic Voting* this added risk has to be mitigated.

3.2 DAOs

This subsection examines a few of the most popular existing DAO platforms as well as which voting systems they use.

3.2.1 Aragon

Aragon is a DAO platform where people can create and manage their own DAOs. There are currently over 1700 DAOs who are collectively managing 900 million dollars on Aragon. DAOs created through Aragon are deployed on the Ethereum blockchain [21]. Aragon enables people to customize their DAOs with different apps, which are smart contracts. Each of these apps is associated with a web interface in order to interact with them [45]. Members are also able to develop their own apps and connect them to their

DAO [46]. Most DAOs on Aragon use the default vote app for voting. When voting with the default vote app the members of the DAO vote with their tokens, i.e. the power is in the money as mentioned in section 1.1. The default voting app has two parameters that can be changed by voting:

- The required percentage of support.
- The minimum acceptance quorum.

The required percentage of support is how much of the voted tokens need to be in favor of the proposal. The minimum acceptance quorum is the percentage of tokens of all available tokens in DAO needed to be cast in the vote for the voting result to be valid [21]. There are other voting apps that can be used to customize a DAO on Aragon. For example there are apps which uses *Holographic Consensus*, described in section 3.1.1. A *Conviction Voting* app has also been implemented on Aragon, *Conviction Voting* is described in section 3.1.2. In short, you can use a wide variety of voting apps for your DAO on Aragon, but as previously mentioned the majority is using the default application [21].

3.2.2 DAOstack

DAOstack is another DAO platform that is trying to solve the scalability problem that occurs within DAOs caused by the governance. Consider a DAO which uses a simple voting system where tokens are used to cast votes and a majority of all votes are needed in order for the proposal to pass. This approach is where a huge part of the members are required to participate in the voting in order for something to pass scales really bad. As the number of members in the DAO grows, so will the proposals, but the member's ability to participate and read up on the proposals is not growing. The members will not have time to participate in the governance. The DAOstack platform is trying to solve this by introducing *Holographic Consensus*, described previously in section 3.1.1. DAOs created on DAOstack are also running on Ethereum [21]. DAOstack sees a DAO as a network of stakeholders where decisions are made through a non-hierarchical governance process [45].

3.2.3 DAOhaus

DAOhaus lets users create DAOs running on the Ethereum blockchain which implements a very simple voting system, also a coin-based system. On DAOhaus the

voting system has no quorum in place, proposals will always pass as long as they have a relative majority. This approach leads to some flaws, mostly related to the risk of being attacked with malicious proposals that take advantage of this attribute. In order to combat these flaws, DAOhaus lets members of a DAO *rage quit* during a grace period after the vote is passed. A member can *rage quit* with its resources if the member does not agree with the outcome of the vote. Functionality is also implemented which will automatically revoke the proposal if 30% of the members *rage quit*. The effect that this functionally has is similar to the possibility of forking the entire DAO. It essentially motivates all shareholders to keep the members happy and stay within the organization [21].

3.2.4 Colony

Colony is also a DAO platform or framework where people can launch DAOs on the Ethereum network but with quite a different way of voting. On Colony voting power is determined by reputation [45]. Reputation is represented as a number associated with the member which tries to represent the value that that member contributes to the DAO. This number is also associated with different domains where the member has the most expertise. The reputation is used to weigh members' votes on proposals that are linked to the domain in which the user has expertise. Reputation is also used to hand out different amounts when rewards are given out. Reputation is handed out by peer assessments of the member's action. This system is created in order for the right people to get more voting power on certain proposals where they have shown expertise. Reputation can be lost by inactivity, misbehavior, or in general bad actions [42].

3.2.5 Governor DAO

Governor DAO is a DAO that offers a suite of products and services for organizations and projects to add DAO qualities to their own communities. The most interesting thing about the Governor DAO is their solution for a one-voice-one-vote governance system which they claim is an industry-first and sybil-resistance product [23]. They implement this with something that they call a *Proof of Existence*-Token. This token is given to users after going through something called a biometric enrollment. This biometric enrollment is done by creating a hash-value based on the user's face and voice. This token is then given to the user's wallet and then the Governor DAO can make sure that there is only one user with that specific face and voice and a fair and completely democratic governance system can be used where each user has one vote. It is not only argued that it is fair but they also claim it is a completely sybil resistant

system. The *Proof of Existence*-Token is also not transferable [24].

3.3 Computational Social Choice

Social choice theory is an entire field of study where methods for collective decision-making are created and analyzed [10]. This field has recently gained some traction in the computer science and artificial intelligence world. Both when it comes to enhancing decision-making protocols as well as using the findings within the field to solve problems that rise up within artificial intelligence [10]. An example of this is when designing voting protocols, it might be difficult to make it impossible to cheat, but the computational difficulty of cheating might be astronomical [5].

There are a lot of very interesting findings within the field of computational social choice. Different voting rules and protocols can be used in order to achieve different outcomes [4]. The way in which political elections are structured can easily be taken for granted but there is a reason why Sweden mainly has two alliances and USA has two popular parties. This research field really highlights the challenges that arise when adding multiple candidates or alternatives in a vote [4]. An example of these difficulties is when participants start to vote against alternatives contrary to voting for, in order to achieve the best outcome. This ends up with the vote being dominated by two alternatives.

In 1972 a man called Kenneth Arrows came up with a theorem that he calls the impossibility theorem. This theorem can be viewed as a paradox within the social choice and collective decision-making. It basically proves that while following a few principles of fair voting procedures it is impossible for a group of people to agree on an order of preference within a set of alternatives. Arrow won a Nobel Prize for this and the impossibility theorem has ever since been widely used within economics [37].

The voting principles mentioned above are the following [37]:

- Social Ordering: All voting participants give an order of the alternatives.
- Independence of Irrelevant Alternatives: If any alternative is removed from the vote, the outcome between the remaining alternatives should not change.
- Weak Pareto: Unanimous individual preferences are respected.
- Nondictatorship: There is no dictator. Every participant's votes should be counted.

- Unrestricted Domain: All possible individual preferences must be taken into account.

The impossibility theorem states that when following these principles it is impossible to construct a social ordering of the alternatives. The principles are incompatible [37].

There are of course other voting systems that are not a victim of this theorem. For example, plurality voting, approval voting, and Condorcet voting [5]. However, there seems to be no system without flaws. There is always discussion of which voting systems are fair or not, which in some cases is not so straightforward and sometimes raises philosophical questions. One of the more distinct problems is whenever participants are incentivized to not vote for their preferred choice in order to manipulate the vote. One goal of constructing these voting systems is to make it computationally difficult to compute how to manipulate the vote in order to achieve the desired outcome [5].

However, the impossibility theorem created by Arrow is built on the assumption that there are more than two alternatives [5]. Whenever there are more than two alternatives it becomes difficult to construct a fair voting system. Within this field, there is another interesting theorem, published by Kenneth May in 1952. The theorem is referred to as May's Theorem and it claims that majority voting is the only voting system that achieves certain properties which makes it a fair rule [19]. These properties are the following [35]:

- Anonymity: The social preference is not biased for or against any participating voter.
- Neutrality: The social preference is not biased for or against any alternative.
- Positive Responsiveness: If an alternative A rises relative to another alternative B for one or more participants' preferences, then A does not fall relative to B in the social preference and if A and B were previously tied, A is now ahead of B.

Kenneth May was able to prove that these properties were satisfied by the majority rule in the case of a vote with only two alternatives [35]. The majority voting system is simply that each voting participant gets to vote on one alternative and the alternative with the majority of votes wins [5]. It can also be seen that the majority voting system is easily the most used democratic method for choosing between two alternatives. Also, more complex voting systems such as plurality voting, runoff voting, and others simply become majority voting in this simple case [35].

3.4 Hedget Proposals

Hedget is a decentralized application for decentralized finance (also referred to as DeFi) created by ChromaWay. More specifically Hedget is a dapp for options trading. Hedget allows users to buy and sell options products by providing cryptocurrencies as collateral. Hedget also allows users to hedge the risk for their holdings [15].

ChromaWay has also implemented a voting dapp for Hedget which is called Hedget Proposals, its open-source code can be found on GitHub [14]. The dapp records votes on proposals created by admins or core users and the votes are cast with a token called HGET [15]. It is a very simple and straightforward approach where one token equal one vote [14]. The Hedget Proposals dapp is used as a base for the implementation discussed in section 5.

4 Method

It is not very straightforward which methods for voting are the best since there are multiple different factors that one can consider, and a lot of these factors can be difficult to measure. One way might be to measure how successful the DAO is, another way might be to ask how satisfied the members are with the voting. These two measurements might give very different results. The fairest and most democratic measurement is probably the latter, but is it fairer to measure the voice of a more invested member proportionally to his or her investment? How should the investment be measured, time as a member or active participation, or financial investments? The answer to these questions probably differs depending on what the use case of the DAO is. However, the goal of this project is not to evaluate current methods but to study what has previously been done within the field and try to explore new ways of achieving a fair way of voting and then see how it can be implemented. The definition of fair can be a bit vague but in this report, a fair voting system refers to a system that is democratic or at least as democratic as possible while remaining secure.

This section covers the chosen approach used for the implementation in section 5 and the motivation behind it. The technologies used for the implementation are also described.

4.1 Proposals

One could argue that it would be reasonable to be able to add multiple different alternatives to each proposal, or at least it would be nice to have the option to do that.

Having multiple different options on the proposals leaves more engagement being required by the voting users. It is probably wiser to reduce the amount of effort needed by the voters in order for people to keep voting. Given what was discussed in section 3.3 about how much more complicated it gets with more voting alternatives when it comes to making the voting system fair. It seems like a very good choice to limit the alternatives which a proposal can have to two, simply pass/fail or yes/no. This way the amount of effort required by the participants is minimized and a voting system that is proven to be fair is used.

4.2 Voting System

As discovered in section 3 about related work there are DAO platforms that use very different approaches to create their voting system. However, not much information was able to be found regarding research being done to back the majority of these approaches. This in turn often leads to the lack of a clear technical and scientific motivation for most of the approaches in the DAO platforms that were researched. Even though certain elements of the voting systems that are being used have been researched before it is important to keep in mind that DAOs are built on top of blockchains and certain voting systems and mechanics might lead to unexpected outcomes that were not a possibility when looking at these voting systems outside of a blockchain setting. The current DAOs that are being created can be seen as experiments that over time will refine and perfect themselves, however, this does not neglect the importance of continued research within the field.

In section 1.1 it is mentioned that the problem with a straightforward coin-based approach is that it is not fair for richer people to have greater voting power. For example, in a democracy, we take for granted that one person is eligible to vote. With a coin-based approach, people are voting with their money. In a DAO this could be problematic, it opens up the risk of rich people voting with the intent to make rich people richer. However, in certain scenarios, it might be considered the fairest solution since it allows for people with more “skin in the game“ to have a bigger voice. For example a DAO with the main goal of increasing its financial position of the DAO, much like a public traded company that tries to increase the value of its shareholders. In this example, it could be argued that the best decision would be to go with a completely coin-based approach where one coin equals one vote.

In the example of a video game that is operating as a DAO it might make more sense for the DAO to try and implement a voting system where one person equals one vote. Since the primary goal of the DAO will most likely be to enhance the gaming experience of all players and increase the number of players. An assumption can be made that this

would be easier to achieve with an approach where one person equals one vote since the video game should not only cater to the richest people in the game. In order for the game to grow and for everyone to have a good experience the game should also cater to new people, otherwise, they will not stay. But this approach is very difficult as it makes the DAO very vulnerable to sybil attacks, that is one person can create multiple different wallets and gain the voting power of an entire group of people. The only DAO platform mentioned in section 3.2 which has implemented this approach is Governor DAO, described in section 3.2.5. The way in which Governor DAO achieves this is by not making the DAO permissionless but instead requiring users to go through a biometric enrollment that creates a hash-value of the user face and voice, the user is then awarded a *Proof of Existence*-Token which enables the user to cast a vote on proposals, this token is then non-transferable [24]. If this approach actually works and a person is not able to obtain two different *Proof of Existence*-Tokens it can be argued that an attractive attribute of the DAO is removed, namely the fact that users are not anonymous anymore. It can also be argued that a biometric enrollment is a lot to ask of a user and will be a big enough hurdle for people to not join the DAO.

Holographic Consensus discussed in section 3.1.1 is a very interesting solution to the scalability of the governance system in a DAO and there was a study that proved its effectiveness. The study was not very extensive though as it was mentioned in the report [22]. The implications of creating a betting system for the voting system could be bigger than expected and more studies should be done on the matter. By being able to bet on the outcome of a vote the DAO runs the risk of having its members vote for which alternative is most probable to win instead of the alternative that the member prefers. This fact could disturb the voting process in a pretty significant way. But as mentioned in section 4 these are difficult things to study and prove. Another way to achieve the possibility of having a quorum of a relative majority is by going with the approach discussed in section 3.2.3 about DAOhaus. DAOhaus allows people to *rage quit* with their assets after a proposal has passed so that every one that is voting will not want to make people unsatisfied, because then people will leave and the DAO together with the financial value will diminish. *Holographic Consensus* and *rage quitting* or forking is a solution to the scalability issue in the governance of a DAO, which is not the main focus of this project. This fact together with the previously mentioned arguments leads to these approaches not being a part of the proposed solution and implementation in this project.

Conviction Voting is a way of letting time be a factor in how much influence a member has which can be seen as fair in some cases. The downside of this is that the member's tokens are locked in the vote for a long period of time, which will result in proposals taking a long time to pass and members cannot use their tokens for other things in the meantime.

Reputation Voting is an attempt of moving away from the coin-based solution by introducing a non-transferable reputation token. The example mentioned in the related work section is the DAO platform called Colony, described in section 3.2.4. Colony's way of implementing the reputation token allows for people with more experience and skill within a certain domain to gain more voting power in that specific domain. It sounds like a fairly complex system and the way in which peer assessments are involved with how much reputation a member will gain seems a bit unclear. This complex reputation system sounds like it would be vulnerable to different attacks and exploits. A simpler reputation voting system could be implemented in such a way that you simply just need one token to be eligible to cast one vote and this token is given to a member by a vote cast by the rest of the token holders. This approach sounds fair, however, it is still not bulletproof and it is also not a permissionless DAO. The DAO would also run the risk of members selling their reputation tokens by selling their entire wallet, this would be very difficult to prohibit. The system also runs the risk of having a small group of people deciding everything and not letting more people join. Which might be optimal for some use cases, but not for a permissionless DAO.

4.2.1 The chosen approach

Quadratic Voting has a very interesting property in that it is drastically reducing the voting power of a rich person compared to a not-so-rich person. This is a property that solves a majority of the problems with rich members having too great of voting power, but the aspect of a financial investment being valuable in the DAO is not completely removed. A few simple examples of how this would work:

- 1 token = 1 vote.
- 25 tokens = 5 votes.
- 100 tokens = 10 votes.

As mentioned in section 3.1.4 about *Quadratic Voting* it does open up a great vulnerability to sybil attacks. Attackers can simply create multiple wallets and bypass the reduction of the votes. The DAO mentioned in 3.1.4 called Illuvium does not mention how this vulnerability is handled in their DAO. Nevertheless, if the DAO is permissionless it will need a way of mitigating the risk of sybil attacks.

While the research was being done on blockchain technology and its different consensus mechanisms described in 2.1.1 a question emerged. If the entire technology behind blockchain is built so that a network of untrusted nodes can reach a consensus without

failure, why cannot these same consensus mechanisms be used in a DAO in order for all members to reach a consensus? The answer to this question probably lies in the fact that the blockchain network is working towards a global state of truth, if a node claims the truth is something else, it is lying. When it comes to voting on proposals in a DAO there is no right or wrong answer which dictates a truth that can be reached. It would also be odd for the member with the most computational power (if *Proof-of-Work* was used) to have the most voting power. However, there is an interesting thing that is used within *Proof-of-Stake* which is called *Coin Age* described in section 2.1.1. *Coin Age* could be used as a way of introducing a value to invested time in the DAO much like *Conviction Voting* is trying to do but without the downsides. In certain cases *Coin Age* could hopefully also be an efficient way to reduce the risk of sybil attacks. The voting power could be determined by a members *Coin Age* of its tokens multiplied by the amount of tokens the member possesses. If the *Coin Age* of a resource is reduced to zero when it is transferred between two wallets it is at the very least impossible to do a swift sybil attack. Depending on how the variables are set up and how long time it takes for the *Coin Age* to reach a meaningful number the sybil attack could be delayed and detected since the blockchain is transparent. Meaning that if suspicious activity or a large amount of tokens are distributed to multiple wallets a system could be set in place to detect such activities and thanks to *Coin Age* sufficient amounts of time would be available to defend against the attack. The research suggests that the idea of using *Coin Age* as a feature within DAO voting has not yet been discussed or attempted.

For these reasons mentioned above the voting system developed during this project is using *Quadratic Voting* together with *Coin Age*. The system is built on the open-source project Hedget Proposals described in section 3.4 which is built with the programming language called Rell on the blockchain platform called Chromia further described in section 4.3.

4.3 Chromia

Chromia is a blockchain platform created in order to try and combat the flaws previous platforms have had when it comes to creating decentralized applications. Chromia is invented and developed by ChromaWay. There exist multiple platforms on which to create decentralized applications. The most popular one is the Ethereum blockchain, where developers can create applications with the programming language solidity. However, the creators of Chromia had bad experiences with Ethereum, such as bad user experience, high fees, frustrating developer experience, and poor security. If dapps is going to become a mainstream technology and live up to its potential these flaws have to be dealt with. Chromia brings a platform that has a completely new architecture and programming model with the following goals [17]:

- Allow dapps to scale to millions of users.
- Improve the user experience of dapps to achieve parity with centralized applications.
- Allow developers to build secure applications by using familiar paradigms.

The creators of Chromia envision blockchain to serve as a shared database in a decentralized system, where additions and updates are safe, authorized, and consistent. Chromia is built as a relational model, where data is stored in a relational database in order to achieve a flexible, versatile, and consistent platform on which dapps can be created. In turn, a relational programming language called Rell is used for developers to create dapps on Chromia with ease, Rell is further described in section 4.3.1. Chromia is built in such a way that each dapp has its own blockchain and with every blockchain being run on a subset of nodes, it enables the platform to scale horizontally by adding nodes. Since each dapp is run by its own blockchain (or sidechain) it is not deployed with smart contracts which allow for the freedom of having its own resources which enables the developers to create their own gas fees and other uses of the resource. Dapps on Chromia can quickly retrieve information from nodes running the dapp due to the benefits of a relational database with indexing, another benefit is that the application is able to handle large amounts of queries and updates [17].

One of the reasons that Chromia is able to achieve these benefits apart from the fact that it is built as a relational database is the fact that Chromia uses a version of *Proof-of-Authority* mentioned in section 2.1.1. A set of provider nodes is chosen by the creators of Chromia and when a sufficient amount of nodes is chosen they will be able to vote on adding new nodes to the system. When this happens Chromia will be a system that is not tied to the creators as a gatekeeper. Consensus is reached with a set of validator nodes that run a practical byzantine fault tolerance algorithm (PBFT [8]), the validator nodes are a subset of all the provider nodes which is running Chromia. In order to further strengthen the security of Chromia it also uses a *Proof-of-Work*-blockchain such as Ethereum or Bitcoin to anchor itself every few blocks. This way it can be argued that Chromia has at least the same confirmation strength as any of these *Proof-of-Work*-blockchains [17]. Since Chromia uses this technique the dapps are able to confirm transactions within 2 seconds, since the huge delay of a *Proof-of-Work* approach is mitigated and the relational model allows for efficient queries. But then again, some of the decentralization is lost in the process.

Chromia is a general-purpose blockchain platform that is optimized to serve as a shared database. It can be used to develop all kinds of dapps [17].

4.3.1 Rell

Chromia is built upon a framework called Postchain which is also developed by ChromaWay. Postchain is compatible with PostgreSQL which is a open source database software but the way in which the SQL queries are made needs to be safe and regulated so that the blockchain is secure, intact, and not abused. This fact together with the reasons that the SQL language is a bit unintuitive to modern developers as well as highly verbose led to ChromaWay developing a new programming language for Chromia [17].

The language is called Rell which stands for “Relational Language“. Rell code is first compiled into a binary format and then it is translated by the Chromia nodes into SQL queries while making sure the queries are not malicious or incorrect [17].

Rell was designed with the intention for it to be very easy to learn for developers. It enables programmers to use relational programming idioms that most developers are already familiar with. The language is also designed to be similar to modern popular programming languages like JavaScript and Kotlin.

Here is a really simple example of some Rell code:

```
entity player { key name; }

operation add_player(name) {
    create player(name);
}
```

In the example above we can see an entity being created, which is much like a class in Java. The entity is called a player and has a key attribute called name. Then an operation called *add_player* is declared, an operation is a data-modifying request. This operation takes in a name and creates a new player with that name [16].

The following JavaScript code is an example of how a client can interact with the Rell code above [16].

```
const tx = gtx.newTransaction([user.pubKey]);

tx.addOperation("add_player", "Kevin");

tx.sign(user.privKey, user.pubKey);
```

```
6
7 return tx.postAndWaitConfirmation();
```

When it comes to dapps which are often handling a lot of financial resources it is very important to catch programming errors in the compilation stage which is why Rell has implemented static type checks. Rell is up to seven times more compact when compared to SQL since Rell is able to derive a lot of information automatically in order to ease the developer of this work. In short, Rell is a compact, convenient, efficient, and safe programming language for a relational blockchain [17].

5 Implementation

This section goes through some of the more important parts of the implemented code and the thoughts and theory behind it.

5.1 Coin Age

The *Coin Age* is implemented in such a way that a date is stored together with the asset balance. When the balance is reduced for example by sending a transaction, the *Coin Age* does not need to be reduced. The remaining balance still has the same *Coin Age*. Although when the balance of the asset is increased for a wallet, for example when a transaction is received the *Coin Age* needs to be adjusted according to how much of the asset is received. An equation for how the *Coin Age* is going to be calculated was created in order to determine how the *Coin Age* would be adjusted after receiving a transaction. In equation 1 below $coinAge_{new}$ and $coinAge_{old}$ refers to the new and old duration between the current time and the *Coin Age* date which is stored with the asset.

$$coinAge_{new} = \frac{balance_{old} \times coinAge_{old} + balance_{new}}{balance_{old} + balance_{new}} \tag{1}$$

The reasoning behind the equation might become more clear after realizing that it is derived from equation 2 below. Multiplying the new *Coin Age* with the sum of the old and the new balance should result in the same value as multiplying the old balance with the old *Coin Age* and then adding the new balance.

$$coinAge_{new}(balance_{old} + balance_{new}) = balance_{old} \times coinAge_{old} + balance_{new} \tag{2}$$

Equation 1 enables the new *Coin Age* to be reduced in accordance with the relation between the old and the new balance.

In Chromia time is represented in milliseconds which means that when it is used to adjust the tokens when casting a vote by multiplying the tokens with the *Coin Age* it will most likely be a very large number that will make the voting results difficult to read as a participant in the vote. Therefore it should be adjusted so that it could easily be observed and compared. To do this the *Coin Age* could be multiplied by a constant. The suggested constant for this project is the following:

$$k = \frac{2}{1000 \times 60 \times 60 \times 24 \times 90} \tag{3}$$

The denominator is 90 days represented as milliseconds. Meaning when the *Coin Age* is multiplied by this constant after 90 days it will result in 2. After 90 days the voting power will be the token balance of the wallet multiplied by 2 (if disregarding the effect of the quadratic voting).

A good idea would also be to cap the *Coin Age* when it is used to adjust the token balance. Depending on the use case of the DAO and how active the members are it could be a good idea to create a cap after a certain amount of months.

The code below represents the entity called balance where the key of the entity is a member's account together with the asset. The actual amount of the balance is a mutable integer, which is also true for the *Coin Age* date.

```
1 entity balance {
2     key acc.account, asset;
3     mutable amount: integer = 0;
4     mutable coin_age_date: integer;
5 }
```

In the following code snippet, we can observe how the balance entity is created. It is done in a function that ensures that a user has a certain asset, and then the balance is returned. If the user has no balance a new balance entity is created. When the balance is created the *Coin Age* is derived from the last block time of the blockchain.

```
1 function ensure_balance(acc.account, asset): balance {
2     val balance = balance @? {asset, account};
3     if (balance != null) {
4         return balance;
```

```
5       }
6       else return create balance(account, asset, amount = 0,
         ↪  coin_age_date = op_context.last_block_time);
7   }
```

The code that can be seen below this paragraph is a part of the function which processes the output of a transaction on the blockchain. In these lines of code the *Coin Age* is adjusted during the transaction. On line 3 the usage of equation 1 can be observed.

```
1 val balance = ensure_balance(target_account, asset);
2 val coin_age_old = op_context.last_block_time -
   ↪  balance.coin_age_date;
3 val coin_age_new = (balance.amount*coin_age_old +
   ↪  o.amount)/(balance.amount + o.amount);
4 balance.coin_age_date = integer(op_context.last_block_time
   ↪  - coin_age_new);
```

5.2 Quadratic Voting

In order to be able to look at the total amount of assets that have been cast to vote we need to store the actual amount in the vote entity. This would be needed if a quorum were to be implemented for example. In the following piece of code, the $amount$ attribute refers to the vote weight that the vote has after being adjusted with Quadratic Voting and Coin Age. Alongside the amounts, there is also the poll entity and the poll option as well as the user's account. Indices are made out of the poll and the poll option for faster queries.

```
1 entity poll_vote {
2     key poll, account.eth_account;
3     index poll, poll_option;
4     amount: integer;
5     actual_amount: integer;
6 }
```

In this implementation, the vote weight or amount is always an integer but in certain cases such as bitcoin, the value can become so large that most people only deal with amounts far below one. When this is the case one might think that a problem would occur because the square root of numbers between one and zero becomes a greater

number. However, this is not a problem. The idea behind *Quadratic Voting* is that a smaller amount of assets gains a bigger advantage than a larger amount of assets. With numbers between zero and one, this fact is still true.

The following code represents the function that is called when a user votes on a proposal. First, the user and the poll are retrieved, then it is checked that the poll has not yet been completed. Then the square root function is called on the staked amount and then an instance of the $poll\_vote$ entity is created, as seen in the code snippet above. In Rell there is actually not any implemented functionality to take the square root of a number, so in the following code, a custom function called $sqrt$ is used.

```
1 operation vote_for_option_in_poll (account_id: byte_array,
   ↪ auth_descriptor_id: byte_array, id: name, option:
   ↪ text) {
2     val user = account.retrieve_verified_user(account_id,
       ↪ auth_descriptor_id);
3     val the_poll = poll @{ .proposal.id == id };
4     require(the_poll.proposal.endTimestamp >
       ↪ op_context.last_block_time, "Proposal already
       ↪ completed");
5     var staked_amount = account.eth_account_state @ { user
       ↪ } .staked_amount
6     var reduced_amount;
7     reduced_amount = sqrt(staked_amount);
8     create poll_vote (the_poll, user, poll_option @{
       ↪ the_poll, .text == option }, amount =
       ↪ reduced_amount, actual_amount = staked_amount;
9 }
```

It should be mentioned that the amount is staked for at least the amount of time that the proposal is open. This means that it is not possible to vote on one proposal with the same tokens multiple times. It is also in the function above that the *Coin Age* is meant to be used but the actual effect of the *Coin Age* has not yet been implemented. How the final vote weight would be calculated can be seen in equation 4.

$$reducedAmount = \sqrt{stakedAmount} \times coinAge \tag{4}$$

6 Results and Discussion

Revisiting the problem statement in section 1.1 and the two goals mentioned we can see that both of the goals have been successfully achieved. The research study on the current state of the field, specifically targeting currently available DAO platforms, focusing on their voting systems has been done. Multiple sources and articles have been found and studied regarding this subject although the information about the specific details of the voting systems has been scarce. The scarcity of this information was anticipated and made clear fairly early in the project and is one of the reasons why the research was challenging, due to the difficulty of finding relevant sources and figuring out which resources were relevant. Prior to beginning the more specific research, background research had to be conducted. In order to understand the problem, blockchain technology had to be understood as well as decentralized applications and decentralized autonomous organizations.

Different reoccurring and interesting voting systems were discussed in the related work section, section 3. A few DAOs that use these systems were also discussed. Later on, in section 4 regarding the chosen method conclusions about the research were made in order to begin and formulate a new approach for a voting system that attempts to be fairer but still remain secure. This new approach uses *Quadratic Voting* and *Coin Age* to do this. The approach also limits proposals to only have two alternatives to vote for, due to the findings from computational social choice in section 3.3. By using *Quadratic Voting* the cost of voting power exponentially increases which reduces the unfair voting power richer people might have in a permissionless DAO where a coin-based voting system is used. *Coin Age* is then also used in order to mitigate the added risk of sybil attacks when using *Quadratic Voting*. *Coin Age* functions in such a way that the voting power of a coin is increased over time, beginning from when the asset is added to the wallet. One might think that the effect that *Coin Age* has is pretty much the opposite of the effect of *Quadratic Voting* voting has. Meaning that with *Coin Age* newer people will have less voting power due to a low *Coin Age*. However, having a financial investment is very different from having invested time in the DAO. Financial investments can of course be bought but time cannot.

The method explained above was then implemented with the relational blockchain Chromia and its programming language Rell. This implementation was challenging as blockchain programming is fairly difficult and prior experience with Rell and blockchain programming was very limited. A lot of time was spent on integrating the software cryptocurrency wallet MetaMask [36], which was not completed due to lack of time and lack of experience. A smart contract for staking (the act of locking an amount of the asset) had to be implemented which made it fall out of the scope of the project. These factors led to the fact that a complete prototype was not finished

however the main implementations which are linked to the project were finished, the most prominent of these are found in the implementation section 5.

7 Conclusions

The conclusions that can be made about the research is similar to the claims of the sources mentioned in the problem statement. They claim that more DAOs need to be created and tested [26]. It is also mentioned that the DAO platforms which currently exist and are being created often form the governance processes without much research or extensive testing. These DAOs can in turn be seen as live experiments which probably will perfect themselves over time. These claims make the topic a bit difficult to research [21]. It is also difficult to find information about the underlying concepts of most DAOs which also contributes to the fact that it is a difficult subject to research [45]. These facts have also been discovered during the research of this project and it is a reason why continued research is important.

There are a lot of different approaches being used for the voting systems in DAOs. None of them are currently flawless. For a lot of use cases, the current coin-based approaches for voting systems in permissionless DAOs come with either vulnerabilities or the possibility of unfair outcomes. Current voting systems that do not use a coin-based approach has a lot of uncertainties and a lot of questions regarding how they work or should work.

A mix of *Quadratic Voting* and *Coin Age* was proposed and implemented in this project in a voting system that limits the possible alternatives on proposals to two. *Quadratic Voting* has previously been used in a DAO but based on the research *Coin Age* has not been proposed or used in a DAO before. This approach stems from the existing research on DAOs and blockchain consensus mechanisms. The approach is not tested in this project, but its possibility to solve the discussed problems has been argued. The argument for this approach is that it reduces the voting power of the rich with *Quadratic Voting* in order to make the voting system fairer for everyone. Making the voting system fairer in this manner opens up vulnerabilities to sybil attacks, but the proposed approach has the potential to mitigate or remove these vulnerabilities. The approach is not meant to be ready for production but it is meant for others to gain inspiration and evaluate it in practice.

8 Future work

The proposed and implemented approach for a voting system could be further developed and tested by others in order to evaluate it in practice. More specifically more mathematics and real-world examples could be done on the approach in order to gain insight into how to tweak it and prove that it works. In this report, it is argued that *Coin Age* will delay a sybil attack and hopefully make it possible to detect an upcoming attack, but that idea still needs further development and testing to apply it in practice. As previously stated the entire topic of DAOs definitely needs more scientific research, specifically the governance process and the voting systems. It is vital in order to further develop the field and solve the existing problems regarding voting since the voting system is such an important foundation for DAOs. Hopefully, this report can be part of such further research.

References

[1] M. Alawida, A. Samsudin, N. Alajarmeh, J. S. Teh, M. Ahmad, and W. H. Alshoura, "A Novel Hash Function Based on a Chaotic Sponge and DNA Sequence," *IEEE Access*, vol. 9, 2021.

[2] H. Albayati, S. K. Kim, and J. J. Rho, "A study on the use of cryptocurrency wallets from a user experience perspective," *Human Behavior and Emerging Technologies*, vol. 3, no. 5, pp. 720–738, 12 2021.

[3] M. Bondarev, "Energy consumption of bitcoin mining," *International Journal of Energy Economics and Policy*, vol. 10, no. 4, 2020.

[4] F. Brandt, V. Conitzer, and U. Endriss, "Computational social choice," *Multiagent systems*, vol. 2, pp. 213–284, 2012.

[5] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, *Handbook of computational social choice*. Cambridge University Press, 2016.

[6] W. J. Buchanan, *Cryptography*, 2017.

[7] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized Applications: The Blockchain-Empowered Software System," *IEEE Access*, vol. 6, pp. 53 019–53 033, 9 2018.

[8] W. Cai, W. Jiang, K. Xie, Y. Zhu, Y. Liu, and T. Shen, "Dynamic reputation–based consensus mechanism: Real-time transactions for energy

blockchain," *International Journal of Distributed Sensor Networks*, vol. 16, no. 3, 2020.

[9] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," 2019.

[10] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet, "A short introduction to computational social choice," in *International Conference on Current Trends in Theory and Practice of Computer Science*. Springer, 2007, pp. 51–69.

[11] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," 2016.

[12] B. B. A. Christyono, M. Widjaja, and A. Wicaksana, "Go-Ethereum for electronic voting system using clique as proof-of-authority," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, 2021.

[13] ChromaWay. About chromaway. [Online]. Available: https://chromaway.com/about-us

[14] ChromaWay. Hedget proposals. [Online]. Available: https://github.com/snieking/hedget-proposals/blob/develop/README.md

[15] ChromaWay. Hedget protocol. [Online]. Available: https://www.hedget.com/papers/hedget_whitepaper_eng.pdf

[16] ChromaWay. Welcome to the rell sdk! [Online]. Available: https://rell.chromia.com/en/master/index.html

[17] ChromaWay. (2019) Platform white paper. [Online]. Available: https://chromia.com/documents/Chromia-_-Platform-white-paper2019.pdf

[18] M. Du, X. Ma, Z. Zhang, X. Wang, and Q. Chen, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-January, 2017.

[19] J. Duggan, "May's theorem in one dimension," *Journal of Theoretical Politics*, vol. 29, no. 1, pp. 3–21, 2017.

[20] P. Ekparinya, V. Gramoli, and G. Jourjon, "The Attack of the Clones Against Proof-of-Authority," 2020.

[21] Y. Faqir-Rhazoui, J. Arroyo, and S. Hassan, "A comparative analysis of the platforms for decentralized autonomous organizations in the Ethereum blockchain," *Journal of Internet Services and Applications*, vol. 12, no. 1, 2021.

[22] Y. Faqir-Rhazoui, J. Arroyo, and S. Hassan, "A scalable voting system: Validation of holographic consensus in Daostack," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2020-January, 2021.

[23] Governor DAO. Introduction. [Online]. Available: https://docs.governordao.org

[24] Governor DAO. Project voting & challenges. [Online]. Available: https://docs.governordao.org/dao-voting-concepts/project-voting-and-challenges

[25] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Generation Computer Systems*, vol. 107, pp. 760–769, 6 2020.

[26] S. Hassan and P. De Filippi, "Decentralized autonomous organization," *Internet Policy Review*, vol. 10, no. 2, 2021.

[27] Horizen Academy. Blockchain as a data structure. [Online]. Available: https://academy.horizen.io/technology/expert/blockchain-as-a-data-structure/

[28] Y. Y. Hsieh, J. P. Vergne, P. Anderson, K. Lakhani, and M. Reitzig, "Bitcoin and the rise of decentralized autonomous organizations," *Journal of Organization Design*, vol. 7, no. 1, 2018.

[29] Illuvium. Dao governance. [Online]. Available: https://docs.illuvium.io/whitepaper/dao/

[30] Y. Keping, "Governance and Good Governance: A New Framework for Political Analysis," *Fudan Journal of the Humanities and Social Sciences*, vol. 11, no. 1, 2018.

[31] S. P. Lalley, E. G. Weyl *et al.*, "Quadratic voting," *Available at SSRN*, 2016.

[32] B. Lashkari and P. Musilek, "A Comprehensive Review of Blockchain Consensus Mechanisms," 2021.

[33] A. López Vivar, A. L. Sandoval Orozco, and L. J. García Villalba, "A security framework for Ethereum smart contracts," *Computer Communications*, vol. 172, pp. 119–129, 4 2021.

[34] M. Malik, M. Dutta, and J. Granjal, "A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things," 2019.

[35] E. Maskin, "Arrow's theorem, may's axioms, and borda's rule," 2020.

[36] MetaMask. Metamask. [Online]. Available: https://metamask.io

[37] M. Morreau, "Arrow's theorem," 2014.

[38] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, 2008.

[39] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," *ICT Express*, vol. 7, no. 1, 2021.

[40] E. A. Posner and E. G. Weyl, "Quadratic voting as efficient corporate governance," *The University of Chicago Law Review*, vol. 81, no. 1, pp. 251–272, 2014.

[41] M. Pustišek, A. Umek, and A. Kos, "Approaching the communication constraints of ethereum-based decentralized applications," *Sensors (Switzerland)*, vol. 19, no. 11, 2019.

[42] A. Rea, D. Kronovet, A. Fischer, and J. du Rose. Colony technical white paper. [Online]. Available: https://colony.io/whitepaper.pdf

[43] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, "A systematic literature review of blockchain cyber security," 2020.

[44] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 2084–2123, 7 2016.

[45] M.-C. Valiente Blázquez, S. Hassan, and J. Pavón Mestras, "Evaluating the software frameworks for developing decentralized autonomous organizations," in *XVI Jornadas de Ciencia e Ingeniería de Servicios (JCIS)*, 2021. [Online]. Available: https://biblioteca.sistedes.es/submissions/descargas/2021/JCIS/2021-JCIS-001.pdf

[46] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, and F. Y. Wang, "Decentralized Autonomous Organizations: Concept, Model, and Applications," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 870–878, 10 2019.

[47] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 10 2014.

[48] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys*, vol. 52, no. 3, 2019.