First Deployed DAO with True Full Decentralisation

Brian Planje

To obtain the degree of Master of Science in Computer Science
Data Science & Technology Track
To be defended publicly on July 7th, 2023

Student number:	4599438
Thesis committee:	Dr. ir. J.A. Pouwelse, supervisor
	Dr. ir. Cyntha Liem
	Dr. ir. Leonard Franken

Delft University of Technology
Faculty of Electrical Engineering, Mathematics & Computer Science
Distributed Systems Group

First Deployed DAO with True Full Decentralisation

Brian Planje, Johan Pouwelse (thesis supervisor)
B.O.S.Planje@student.tudelft.nl, J.A.Pouwelse@tudelft.nl
Distributed Systems
Delft University of Technology
Delft, The Netherlands

–master thesis–

***Abstract*—Blockchain technology has allowed for the emergence of a new type of organization, the Decentralized Autonomous Organization (DAO). They have gained significant traction in recent years, reaching market capitalizations of up to 60 billion USD in 2021. These organizations coordinate economic activity by an unbounded group of people within an adversarial environment. However, despite their potential, currently deployed DAOs face notable challenges related to centralization in governance and infrastructure. This work addresses these limitations by proposing a novel architecture for a fully decentralized DAO with no compromises. We introduce a scalable governance protocol utilizing multi-signature schemes to manage shared assets effectively. To demonstrate the feasibility of our approach, we implement, deploy, and evaluate a real-world DAO called Music DAO. Music DAO serves as a compelling use case, enabling listeners to collectively invest in and listen to their favorite artists. This research represents a significant advancement in the field of decentralized organizations, with the potential to revolutionize the way people collaborate and organize themselves.**

***Index Terms*—Decentralized Autonomous Organization (DAO) and operation, blockchain, multi-signature scheme, protocol design, smart contracts, distributed control**

I. Introduction

Decentralized Autonomous Organizations (DAOs) are a mechanism for economic activity by an unbounded group of people within an adversarial environment. They present a fundamental way for people to organize themselves online. Absent of any managers, any person can join, propose, and vote on decisions. Bottom-up interaction and coordination allow such an organization to leverage the wisdom of the crowd [11]. Bitcoin has solved the problem of collective decision-making without a trusted third party by making an immutable ledger possible [17], which eventually led to the emergence of DAOs. Prior to this emergence, partially decentralized protocols and platforms such as BitTorrent and Wikipedia enabled millions of individuals to collaborate in file sharing and information accumulation. The growing emergence and popularity of decentralized protocols highlight their potential for fostering collaboration between individuals.

DAOs have a long-standing history, with the first DAO deployed a decade ago on Ethereum named ‘The DAO’ [6, 9]. Since then, the number of deployed DAOs has grown enormously. In 2021 there were over 2,000 DAOs deployed on Ethereum alone with an aggregated market capitalization exceeding $60 billion [5]. These DAOs are mostly built around *decentralized finance (DeFi)*, such as the decentralized exchange Uniswap. This exchange reached transaction volumes of up to $85.5 billion in November 2021 and maintains a pool of shared assets, distributed among its members in the form of tokens. The Uniswap DAO empowers its members to manage the exchange by investing its assets in development work or other initiatives. Members can also alter the exchange’s rules, such as transaction fees, through a governance process. In this process, members cast their votes using their tokens to indicate whether they agree with the proposed investment or changes.

Despite the rapid development of this paradigm, many DAOs exhibit forms of centralization in both their governance and technical infrastructure. For example, the second-largest DAO by market capitalization, APE DAO, had an initial token distribution in which 38% of tokens were allocated to various founders [21]. Since each token represents a vote, these founders now hold a disproportionate amount of voting power. Additionally, proposals are vetted by a centralized moderation team, and all execution of proposals is carried out by the DAO’s foundation members. Another example is Solend, one of the largest decentralized lending systems [24]. In 2022, an incident occurred in which the development team seized control of and liquidated an account belonging to a large-scale investor, who held approximately $170 million worth of cryptocurrency[1]. The team claimed that the account posed a systemic risk to the ecosystem. This incident highlights the prevalence of centralized decision-making in DAOs.

The root cause of the failure of contemporary DAOs to have decentralized governance lies in their inability to decentralize every component without compromising its infrastructure. Proof-of-work and proof-of-stake have failed to scale, despite a full decade of attempts to boost transaction rates, without the loss of decentralisation [28]. Attempts to circumvent this by working with fewer miners which process more transactions have resulted in systems akin to those of traditional authorities, such as VISA. Centralization might even be inevitable, with Cong et al. showing that in the long run, due to centralized mining pools, blockchains such as Bitcoin will have a centralized market structure [7]. Proof-of-stake distributed ledgers

[1] https://blog.solend.fi/slnd1-mitigate-risk-from-whale-1504285ab4d2

run the risk of reinstating a centralized elite. To validate the network, a substantial amount of capital must be placed at risk by a set of validators. This set of validators can then be subjected to regulatory pressure or collude with one another to alter transaction validation rules at the infrastructure layer. They run the risk of moving to a new centrality with a new elite, who can afford to buy enough tokens to put up to stake to validate the network.

In this paper, we propose a new architecture for completely decentralized DAOs. We argue that pure academic decentralisation within a viable and sustainable DAO represents a key milestone in the evolution of Web3. We believe an as-simple-as-possible DAO architecture with basic governance, membership voting, and management of shared assets is a key step forward towards achieving this goal. To demonstrate the feasibility of our architecture, we design, implement, and evaluate a prototype for a DAO centered around music, referred to as the Music DAO. This implementation solely utilizes smartphones and is currently deployed and live. We conduct a real-world test with users and analyze the performance of our governance protocol.

This work contributes the following:

1) **The Simple DAO Architecture** We design and justify an architecture for DAOs that is completely decentralized. To achieve this, we propose a set of requirements, infrastructure, and components that we deem necessary. We provide a detailed design for our components, including our novel governance protocol based on multi-signature schemes.
2) **Music DAO: a truly decentralised DAO** We implement a real-world DAO that revolves around the music industry using our Simple DAO architecture. We use a combination of networks, including the TU Delft-created IPv8 peer-to-peer network layer, TrustChain, Bitcoin and BitTorrent. We create a music platform where artists can share music and receive funds from a flexible DAO structure. This DAO runs on smartphones only, has no central components, and is deployed on the Android Play store.
3) **Performance Analysis** To evaluate the proposed infrastructure and implementation, we perform a set of performance experiments on our governance protocol. Additionally, we conducted an end-to-end experiment on our Music DAO to measure its loading time. We also conducted a real-world test amongst a set of individuals interested in DAOs. The results of these tests provide insights into the feasibility and effectiveness of our proposed architecture and implementation.

II. Problem Description

The goal of this study is to develop and deploy an academically pure decentralised DAO. There is no consensus on how to define a DAO. We define it as *a mechanism for economic activity by an unbounded group of people in a competitive environment devoid of infrastructure, leadership, and legal centralized authority.* An organisation that relies on no central intermediary nor central authority, thereby being truly unstoppable. We believe that the lack of a completely decentralized infrastructure leads to DAOs inheriting the problems of traditional organizations. If even a single component remains centralized while others are decentralized, the DAO may still be vulnerable to the drawbacks of centralization.

In traditional organizations, participants often have little influence on decision-making. Even when they do have influence, the process can be outdated and slow (as in democracy) or limited to a select group of wealthy individuals (as in the case of shareholders in companies). While the internet has helped combat these problems, the issue of digital democracy remains unsolved, as highlighted by Hindman's book 'The Myth of Digital Democracy' [12]. This problem is hard to solve because top-down hierarchies and layers of managers are required to enforce rules. Without the enforcement of rules, participants with conflicting interests may not cooperate due to a lack of trust. Rules are enforced by third-party authorities, such as the legal system or boards of companies. However, their interests may in turn not align with those of the participants, and they may alter or disregard the rules. Big-tech companies for example are ultimately concerned with profit maximization and do this at the expense of privacy-infringement and social problems they cause. This difficult problem of enforcing rules without a third party has seemingly been solved by the advent of Bitcoin [17], paving the way for the emergence of organizations without central intermediaries: DAOs.

The difficulty in creating a Decentralized Autonomous Organization is simultaneously achieving trust, full decentralisation, and scalability. The problem is similar in nature to the blockchain trilemma [28], with the inclusion of decentralisation in terms of governance [15]. Currently, every technology claiming to be a DAO has central points of control and critically relies on central servers [2, 19, 21, 24]. Real decentralized DAOs only exist in theory. Bitcoin and BitTorrent are the only examples of technology stacks that are not reliant on central infrastructure.

In addition, implementing and deploying a DAO is difficult in practice due to the many engineering challenges. It requires interacting with live networks, which are unreliable and hard to test. Rapid advancements in the field lead to badly documented code and libraries are mostly only available in low-level languages due to the performance requirements of cryptographic operations. Most importantly, the code must be secure and bug-free since large financial transactions may depend on the code.

III. Related Work

The concept of Decentralized Autonomous Organizations (DAOs) is relatively new in academia, leading to a scarcity of academic analysis on decentralisation in existing DAOs and theoretical frameworks. These topics are mostly discussed in grey literature such as blog posts, articles, and project documentation. In this section, we will focus on related work on efforts to create theoretical frameworks and architectures for DAOs, efforts to define decentralisation in DAOs, and analysis of current DAOs.

Vitalik Buterin introduced the concept of DAOs early on in his Ethereum whitepaper and in a 2014 blog post [10]. He described the ideal DAO as an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do. In 2016 Christoph Jentzsch successfully deployed the first DAO which is most similar to what we know them as today: ‘The DAO’. With a remarkable internal capital of $150 million USD from 11,000 investors, it demonstrated the potential scale of DAOs. However, ‘The DAO’ encountered a critical smart contract exploit, resulting in an Ethereum blockchain fork to rectify the situation [9]. This incident highlights the challenges of security and vulnerability in DAO implementations.

Considerable effort has been invested in creating theoretical frameworks and architectures for DAOs. This work is closely related to our work since we are also exploring ways to formalize, design and implement DAOs in an academic manner. Shuai et al. developed a comprehensive framework for DAOs that identifies their characteristics, problems, implementations, and upcoming trends [26]. They introduce a five-layer architecture for DAOs that separates governance, technology, incentives, organization, and manifestation. We share their vision of governance and technology as separate layers but find the other layers to be too subjectively defined. They do not, however, give a concrete implementation nor design of such a DAO. Qin et. al make a similar contribution by identifying fundamental principles and requirements for DAOs derived from the three terms present in its definition: decentralisation, autonomy, and organization. Their proposed architecture consists of an organizational, coordination, execution, and application layer. It mainly focuses on the organization and theoretical modeling of labor. Both papers lack in specifying and defining the technical decentralized infrastructure of DAOs.

Several papers have focused on defining and quantifying decentralisation within a DAO. Axelsen et al. created a general framework for assessing decentralisation through expert and literature reviews [3]. This framework consists of five dimensions, each with its own quantifiers. For instance, for governance, they define the number of distinct persons required for a vote to pass as an indication of decentralisation. Appel et al. show that decision-making in current DAOs is highly centralized [2]. Their findings indicate that for more than 69% of proposals, the top three token holders decide the result of the vote. They did this through the analysis of 151 DAOs with 10.639 proposals. Our work also focuses on the decentralisation aspect of DAOs and attempts to identify requirements that ensure decentralisation.

IV. The Simple DAO Architecture

We now present our architecture, which we coin The Simple DAO Architecture, visualized in Figure 1. We deliberately remove all unnecessary features and complexity in order to provide a future-proof, generic, and principled building block. Our architecture represents a milestone within the evolution of actual DAO realisations: it is the first to achieve full decentralisation. We first discuss our requirements, then go over the infrastructure we rely on and lastly go over our components. We deem these components necessary to reach our goal of making economic activity by an unbounded group of people in a competitive environment devoid of infrastructure, leadership, and legal centralized authority possible.

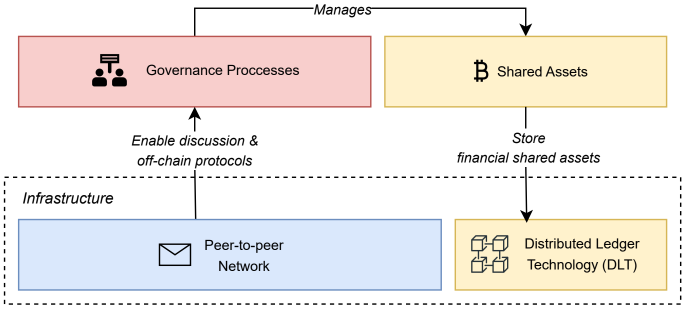


Fig. 1: The Simple DAO Architecture

A. Architectural Requirements

Our architectural requirements are based on the principle of decentralisation and the zero-server architecture [22]. These principles provide a foundation for designing infrastructure that serves the common good, emphasizing the absence of hierarchy in networks, intermediaries, and the inclusion of democratic decision-making processes. Guided by these principles, we have identified three key requirements that are crucial for the realization of fully decentralized DAOs.

1) Trustless: Interactions between participants must not require any inherent trust. Instead, distributed protocols based on cryptography should be employed, allowing each participant to independently verify the integrity of the system. This includes cryptographic protocols such as public-key cryptography and consensus mechanisms that rely on incentives, such as proof-of-work. Trustlessness ensures that no intermediaries are needed to provide that trust, which is essential in a DAO. Furthermore, it ensures that decision-making processes are verifiable fair, transparent, and resistant to cheating.

2) Permission-less: A fundamental requirement of our architecture is its permissionless nature, allowing anyone to participate in the organization without requiring approval from centralized authorities. Discrimination based on factors irrelevant to the functioning of the DAO should be eliminated. However, it is important to note that collective decision-making of existing members within the organization can still determine whether a person is allowed to join or not. The permissionless nature promotes decentralisation by removing barriers to entry and fostering inclusivity.

3) Transparent: Transparency is a core requirement of our architecture, ensuring that all relevant information regarding the organization is accessible and visible to everyone. This includes details about participants, their actions, decision-making processes, and other relevant data. Transparent access to information empowers participants to inspect and verify the state of the organization, enabling informed decision-making and eliminating unfair information asymmetry. Furthermore, transparency promotes accountability, as participants can be held accountable for their actions. Transparency is essential for both internal and external stakeholders to help foster trust between the organization and the broader community.

B. *Infrastructure*

Decentralized infrastructure is a necessity to realize our goal. In our architecture, we rely on peer-to-peer (P2P) networks to facilitate communication without the need for intermediaries. We now describe the technologies we consider necessary for a DAO infrastructure.

1) Distributed Ledger Technology: Distributed Ledger Technology (DLT) enables secure and decentralized financial transactions. Data is replicated and validated across a network of nodes communicating in a peer-to-peer network. The most commonly used form of DLT is blockchain, a tamper-resistant data structure consisting of linked blocks that store transactions. A consensus mechanism, such as Proof of Work [17], ensures the immutability and trustworthiness of recorded information. This mechanism effectively addresses the problem of double spending and provides a high level of trust required for financial transactions. Furthermore, DLT can also be utilized for storing non-financial information that benefits from high availability and immutability.

For our architecture, we consider it crucial that the DLT employed is open-source, permissionless, transparent, and sufficiently decentralized. Open-source code enhances trust and security by making it more difficult for a codebase to be maliciously altered, as it allows for community review and verification. The network must be permissionless to enable open participation and foster decentralisation due to a larger and more diverse range of nodes being able to verify the network. Transparent transactions allow for the verification of governance processes and allow members to hold each other accountable. The notion of sufficient decentralisation can be measured in terms of the difficulty to attack the network, the longevity of the network, and a number of other quantitative measures [15]. Without this decentralisation, components such as governance run the risk of becoming centralized again.

2) Peer-to-peer Network: In order to coordinate governance and other activities, participants need to be able to communicate with one another in a peer-to-peer manner. This includes both communication in the form of human conversations and technical protocols. Communication must be tamper-proof and authenticated so that participants can hold each other accountable for any decisions they make in governance processes.

In order to verify cryptographic protocols and enable informed decision-making by new participants, it is crucial to maintain a historical record of communication. This record should adhere to the principles of *local-first-data-storage* [13]. With local-first data storage, the responsibility for data availability lies collectively with network participants. The network should not rely on any specialized data providers, as this would reintroduce centralization. Instead, data is stored on the numerous devices available to users, such as smartphones, computers, and tablets. By employing protocols such as gossiping protocols built on top of peer-to-peer networks, data replication can be employed to ensure data availability.

Peer-to-peer overlay networks facilitate the aforementioned type of communication. Traditional internet communication methods such as bulletin boards, forums, and social media platforms do not satisfy our strict requirements. They are inherently centralized and subject to moderation and censorship. Overlay networks offer an abstraction layer that shields underlying infrastructure complexities and enables authenticated messaging between peers in a decentralized network architecture. Public-key cryptography is employed within these overlay networks to establish participant identities. Overlay networks provide a foundation for deploying decentralized protocols, such as our governance protocol, which we will describe later in Section V.

C. *Components*

We build two features that we deem necessary for economic activity on top of our infrastructure: governance and shared assets. We now describe what these components should look like. In Section V, we will present the design for these components.

1) Governance: Governance processes make economic activity possible by enabling participants to collectively make decisions in a trustless manner. Any member has the opportunity to submit *proposals* on various topics, ranging from fund management to policy changes. Other members participate in the exchange of ideas and perspectives, ideally on decentralized messaging boards, to improve the proposal and outcome. After proper discussion, members can vote on proposals through a governance protocol. This is a protocol that enables voting in a trustless manner using cryptography. The execution of voting results must be automated, without requiring human intervention. Iteratively repeating this process allows the organization to evolve organically, incorporating feedback and learning from past mistakes.

Distributed ledgers are the only way to make governance protocols possible while satisfying our requirements. This type of governance is also referred to as on-chain governance. Proposals, votes, the result, and the execution of the vote can be stored and executed on-chain in an immutable and secure manner. Typically this is done through the use of smart contracts, which suffer from high blockchain space usage and other limitations. We describe an alternative approach in Section V.

We do not deem governance which is solely off-chain real governance. This is governance that is not stored or validated on-chain at any point during the process. It relies on the counting of signatures posted on a bulletin board on a platform such as Snapshot[2]. It lacks trustlessness since an external party,

[2] https://snapshot.org//

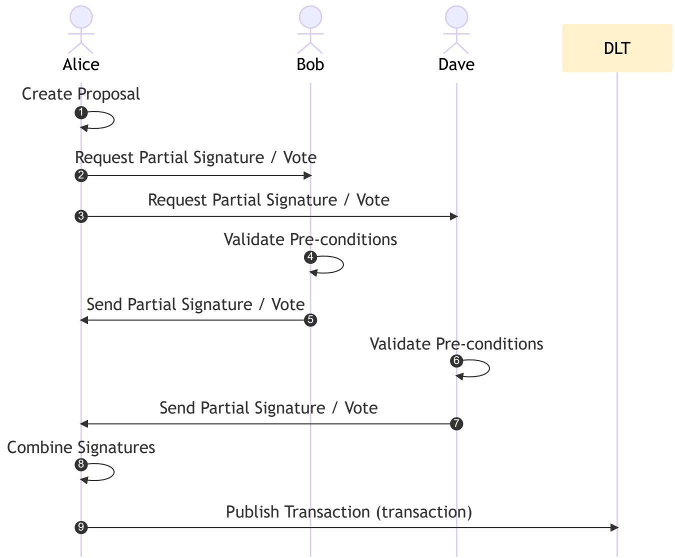

Alice
Bob
Dave
DLT
Create Proposal
1
Request Partial Signature / Vote
2
Request Partial Signature / Vote
3
Validate Pre-conditions
4
Send Partial Signature / Vote
5
Validate Pre-conditions
6
Send Partial Signature / Vote
7
Combine Signatures
8
Publish Transaction (transaction)
9

Fig. 2: Sequence diagram of our online democratic governance.

such as an internal commission, must be trusted to execute the voting result. In case of collusion or other malpractices, there is no recourse available.

To ensure orderly and "fair" decision-making within a DAO, a set of governance rules should be established. Generally, individuals who contribute more and take on responsibility should have more benefits in the decision-making process than others. Digital tokens for the DAO itself can enable this differentiation. This concept is often a matter of debate, and the concept of "fairness" in decision-making is also an open research question still [26]. We deviate from this mainstream model by deploying the one-human-one-vote model of democratic governance. It prevents power from going to the wealthy and ensures that existing institutions cannot lay claim to power on the basis of their authority.

2) Shared Assets: In order to fund its activities and achieve its objectives, a DAO must have shared assets. Although DAOs without any assets can rely on altruism to some extent, typically financial incentives are needed to make work possible in practice. These assets belong to the DAO members and can be managed through governance processes enabled by DLT. Members should be able to lock funds and transfer them to external entities. Cryptocurrencies are a suitable choice as they satisfy all three established requirements. They can be programmed for trustless transfers following a governance vote. Conversely, real-world assets face challenges in achieving trustless transfers. Although they can be digitized into digital assets, doing so requires a custodial entity, which compromises decentralisation and trustlessness.

V. Design

We now introduce our design for the components specified in Section IV-C. Recall that all prior DAOs lack full decentralisation. Our design doctrine dictates true full decentralisation. The design we describe adheres to our architectural requirements and makes use of the infrastructure in Section IV-B. In Section VI we implement this design in our deployed Music DAO.

A. Governance

To address the problem of DAO governance, we propose a novel protocol that combines cryptographic multi-signature schemes and blockchain technology, enabling secure and decentralized decision-making as outlined in Section IV-C1 of our architecture. Existing solutions rely on smart contracts, which perform the voting process entirely on-chain. Our approach involves conducting the voting process off-chain while only storing and executing the result on a DLT such as a blockchain, as depicted in Figure 2. This approach effectively reduces transaction costs, saves blockchain storage space, and decreases the total time required for a vote. However, it is important to note that smart contracts allow for extendability and advanced functionalities such as vote delegation, automatic fund transactions after successful proposals, and additional requirements for initiating proposals.

Our design builds upon established multi-signature schemes and carefully avoids the need for costly smart contracts [14, 16, 18]. These are cryptographic schemes in which a set of participants jointly have ownership over a shared public key. The creation of this shared public key is done securely through the aggregation of all individual public keys. In order to create a signature, each participant creates a partial signature using their own public key. These partial signatures are then combined into a single joint signature valid for the shared public key. Furthermore, threshold schemes can be employed, which necessitate only a subset of the partial signatures, rather than all of them. While a less complex solution for implementing our protocol on a blockchain such as Bitcoin involves using simple scripts, we consider this approach to be non-scalable [1]. The list of public keys and the signatures are all stored on the blockchain within the script. The size of the transaction scales with the number of members, unlike our solution in which the transaction size remains static.

In our design, the act of creating a partial signature is analogous to casting a vote in favor of a proposal. As illustrated in Figure 2, participants engage in an exchange of messages within an overlay network, described in Section IV-B2. Each participant has a unique public and private key pair, and all participants are aware of each other's public keys. First, a single user creates a proposal. This proposal can be any arbitrary text message since the signature will be created over a hash of this message. It then informs other participants of the proposal. Participants vote in favor by signing the message and returning it. Participants implicitly vote against the proposal by not participating. If sufficient partial signatures are available, the vote is over and the proposal has been accepted by virtue of the creation of the signature. Sufficient here is defined as either all participants in the case of multi-signature schemes or the threshold amount in the case of threshold schemes. If the proposal involves a financial transaction on a blockchain, it can be published and stored on the blockchain. It is important to note that time limits for voting and the ability to revoke votes are not possible within this context.

With our design, members can vote off-chain, and only the result of the vote needs to be stored on-chain. This approach enables us to achieve a high level of security and scalability while minimizing the complexity of the system. Our solution does not rely on general-purpose smart contract capabilities but instead utilizes the simplicity and security of the chosen blockchain, specifically Bitcoin. One significant advantage of our approach is the substantial reduction in the number of on-chain transactions required, which can be up to n, where n represents the number of members in the DAO. This reduction can be seen in the comparison of governance protocols in Table I, in which we compare the blockchain space usage needed for a single proposal and vote. Our governance protocol, which can use any of the multi-signature or thresh-hold schemes, demonstrates a constant blockchain storage requirement, in contrast to smart contracts that scale with the number of members n.

The time complexity of our design depends on both the cryptographic operations required and the communication overhead of messages. The most expensive operation in multi-signature schemes is the number of (multi-)exponentiations required. The MuSig2 paper provides a comparison of a large number of schemes and shows that this number scales linearly with the number of participants in all schemes [18]. The communication overhead is dependent on the network topology and scheme used. These schemes require at least one, but often multiple rounds of full communication between all peers to guarantee security. To realize this, a full broadcast is needed. The lower bound for a full broadcast is $\Omega(D)$, where D represents the network diameter, defined as the maximum distance between any two nodes in the network [20]. While in practice difficult, using optimized network topologies linear overhead can be achieved. In conclusion, our design can operate in linear time with respect to the number of participants.

We additionally introduce the concept of a *pre-condition* to make management of shared assets possible, which will we describe in the proceeding section. This is a function in an arbitrary programming language that verifies a condition, such as the state of the blockchain at that moment. This function is verified by members locally as a pre-condition for creating a vote. Note that this *pre-condition* is not secured through additional cryptographic means: if sufficient people want to collude and ignore the *pre-condition*, they can do so and still create a valid signature.

B. Shared Assets

Building on our governance protocol described in the previous subsection, we can enable members to manage shared assets as described in Section IV-C2. The shared assets we use are the native cryptocurrency of the blockchain since we are not using any smart-contract capabilities. At all times the DAO has a single *shared public key*. A shared public key is created using a multi-signature scheme and allows the participants to jointly have ownership over the shared assets. All cryptocurrencies locked up using this key in transactions are considered the shared assets of the organization. Managing these shared assets requires being able to lock up funds, transfer these funds and add or remove members from co-managing the funds. New members must also pay an entrance fee in order for the DAO to keep functioning.

We will now go over the three functions we need for managing funds and how they are handled.

Locking funds - Contributing funds to the DAO entails publishing a signed transaction that includes an output locking the sender's funds using the DAO's shared public key. Subsequently, these funds can now be spent by the members of the DAO.

Transferring funds - To transfer locked-up funds, members can propose an unsigned transaction that unlocks the current DAO funds, enabling them to be transferred to external parties. This proposal is then voted on using our governance protocol, resulting in a valid signature for the transaction. Once the signature is generated, any member can publish the signed transaction on the blockchain, ensuring the irreversible transfer of funds

Member addition and removal - In addition to locking and transferring funds, our governance protocol also enables members to add or remove people from co-managing the shared assets. For a new member to join, all funds must be moved to a new address by locking it using a new shared public key that includes the new member. Typically, the new member must first pay the pre-agreed upon entrance fee to keep the DAO functioning. This requires two sequential transactions: one in which the new member locks up the entrance fee funds and one in which the old funds are moved to the new shared public key. The problem is that the existing members could commit fraud by not fulfilling their promise to add the new member after the new member has paid the entrance fee.

We meticulously design the *pre-condition* and make use of a special unsigned transaction to avoid this problem. The key idea is to enable the new member to atomically pay the entrance fee and join the DAO simultaneously. The new member first generates a new shared public key, which includes their own individual key, and creates an unsigned transaction. This transaction, visualized in Figure 3, has two inputs and outputs. The first input is the entrance fee, signed by a personal wallet of the new member. The second input is the previously locked-up DAO funds. The output of the transaction is equal to the amount of the previously locked-up DAO funds plus the entrance fee and is locked up using the new shared public key. Additionally, an output can be added to return change to the new member, since in Bitcoin all outputs of a transaction need a destination for them not to be lost. This transaction is then subject to our governance protocol and published if successful. It is now impossible for a new member to join without paying the entrance fee or for a new member not to

TABLE I: Comparison of different governance protocols for 1 single proposal. pk is the size of a single public key. sig is the size of a single signature. n is the amount of members participating in the voting process. N/A is due to the protocol only requiring a single transaction, thus not being applicable.

Governance Protocol	**Type**	**Year**	**Transactions Required**	**Size Single Transaction**	**Size All Transactions**
Smart Contract [19]	Smart Contract	2013	n	$pk + sig$	$n \cdot (pk + sig)$
Naive Bitcoin [1]	Bitcoin Script	2008	1	$n \cdot (pk + sig)$	N/A
MuSig [16]	Multi-signature	2018	1	$pk + sig$	N/A
MuSig2 [18]	Multi-signature	2020	1	$pk + sig$	N/A
FROST [14]	Thresh-hold signature	2020	1	$pk + sig$	N/A

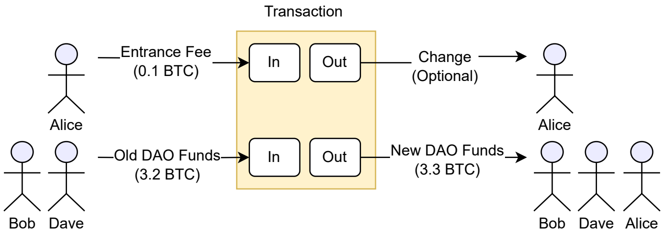


Fig. 3: Transaction which adds a member to the shared assets of the DAO

be accepted into the DAO after paying. The pre-condition is set so that all parties check whether the transaction is not fraudulent and exactly as specified here before signing.

The procedure for removing a member follows a similar approach and involves excluding their key from the new shared public key. Any member can initiate the removal procedure by first creating a new shared public key that excludes the departing member. They then use this new key to create an unsigned transaction that moves the old funds to this new key. Optionally, if existing members wish to do so, they can add an output returning some of the funds to the departing member. If a sufficient number of members vote in favor, the transaction can be signed and published, resulting in the departing member losing their voting rights and leaving the DAO.

The implicit governance structure exhibited here is founded on the ownership of private key shares. As mentioned in Section IV-C1, we ideally want a one-human-one-vote governance structure. It is important to note that our current design does not address the issue of Sybil attacks. A one-human-one-vote model can be implemented using Sybil-resistance mechanisms [27]. In the absence of this restriction, a single user can create sybils to acquire additional shares based on the required criteria for membership. This can be desirable if, for instance, the members of the DAO wish to incentive greater participation in the DAO (financial or otherwise), which can be rewarded with additional private key shares.

VI. Music DAO: a truly decentralised DAO

We have created Music DAO to reshape the music industry. We meticulously designed Music DAO to replace any existing intermediary with open-source code. We choose this industry since it is plagued by intermediaries: streaming platforms, record labels, distributors, and payment processors. The goal is to re-distribute the power back to end-users and away from any large intermediaries. In short, our DAO enables artists to earn a living through music and allows listeners to listen to their preferred music and support artists. Various roles such as talent scouting remain, but no longer require any human labor. A music curator is no longer required if real-time viral music statistics exist. Current cloud-based architecture restricts such vital business information.

Our DAO allows listeners to directly contribute to artists. Artists receive a 100% revenue split and do not have to share up to 30% of their revenue with streaming platforms such as Spotify [25]. This allows them to completely focus on music and further incentivizes listeners to support their artists. Listeners can do this through simple donations on the Bitcoin network, or more importantly through DAO functionality. This functionality is based upon our governance protocol described in Section V. Any listener can start a new fund that other listeners can join. Together they can make proposals to fund the projects of their favorite artists.

Our usage of open-source technologies and permissionless networks keeps users fully in control of their music and funds. Vendor lock-in, a phenomenon prevalent among streaming platforms, poses significant challenges for artists as it restricts their ability to move their music to alternative services. Furthermore, the coercive practices of record labels, requiring artists to give up their music rights indefinitely, magnify the issue of limited autonomy within the industry. A small number of platforms take up the majority of the market share: Spotify, Youtube, and Apple Music. The monopolization of this space forces artists to succumb to the power of these platforms, to have a chance at succeeding.

The absence of an open API or protocol for artists to seamlessly share their music across multiple platforms further exposes the challenges they face. Artists have no control over how their music is consumed, with many platforms being riddled with advertisements. They cannot instead offer their listeners alternative open-source software, unlike our solution. Even if an artist decides to use multiple platforms, they must agree to all their terms and conditions, which are subject to change and unfavorable. Moreover, the DAO's censorship resistance qualities address the concerns of artists residing in jurisdictions with strict censorship policies, granting them the freedom to express their art without fear of suppression or unjust moderation.

The Music DAO comprises two core components: the music

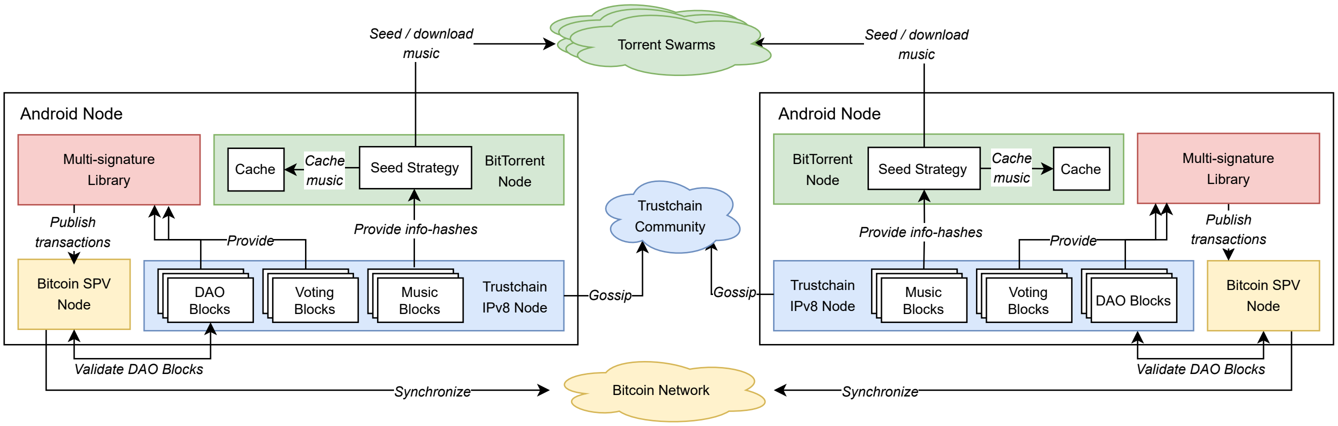
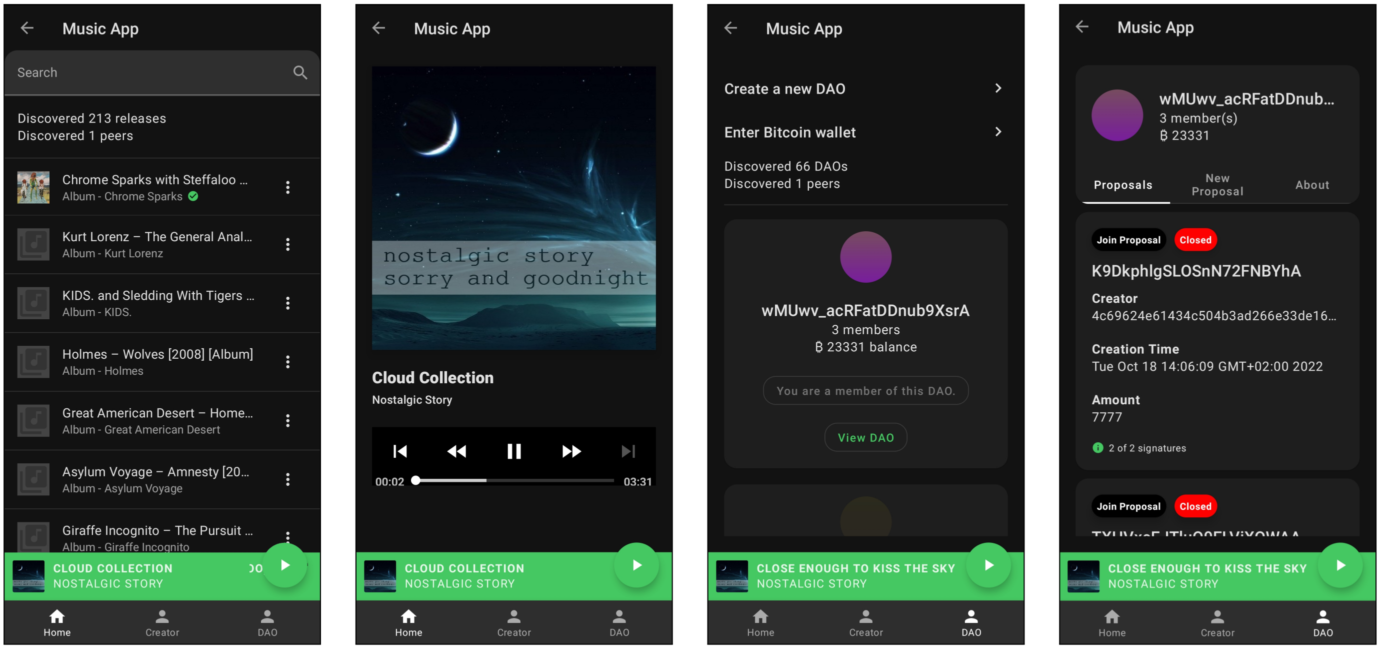


Fig. 4: A visual representation of the Music DAO based on our architecture.



Discovered 213 releases
Discovered 1 peers

Chrome Sparks with Steffaloo ...
Album - Chrome Sparks

Kurt Lorenz – The General Anal...
Album - Kurt Lorenz

KIDS. and Sledding With Tigers ...
Album - KIDS.

Holmes – Wolves [2008] [Album]
Album - Holmes

Great American Desert – Home...
Album - Great American Desert

Asylum Voyage – Amnesty [20...
Album - Asylum Voyage

Giraffe Incognito – The Pursuit ...
Album - Giraffe Incognito

CLOUD COLLECTION
NOSTALGIC STORY

(a) Our album discovery overview screen

nostalgic story
sorry and goodnight

Cloud Collection
Nostalgic Story

00:02 03:31

CLOUD COLLECTION
NOSTALGIC STORY

(b) Our playback screen for a downloaded album

Create a new DAO

Enter Bitcoin wallet

Discovered 66 DAOs
Discovered 1 peers

wMUwv_acRFatDDnub9XsrA
3 members
₿ 23331 balance

You are a member of this DAO.

View DAO

CLOSE ENOUGH TO KISS THE SKY
NOSTALGIC STORY

(c) Our DAO discovery overview screen

wMUwv_acRFatDDnub...
3 member(s)
₿ 23331

Proposals | New Proposal | About

Join Proposal | Closed

K9DkphlgSLOSnN72FNBYhA

Creator
4c69624e61434c504b3ad266e33de16...

Creation Time
Tue Oct 18 14:06:09 GMT+02:00 2022

Amount
7777

2 of 2 signatures

Join Proposal | Closed

CLOSE ENOUGH TO KISS THE SKY
NOSTALGIC STORY

(d) Our DAO voting screen of a proposal

Fig. 5: Screenshots of the Music DAO

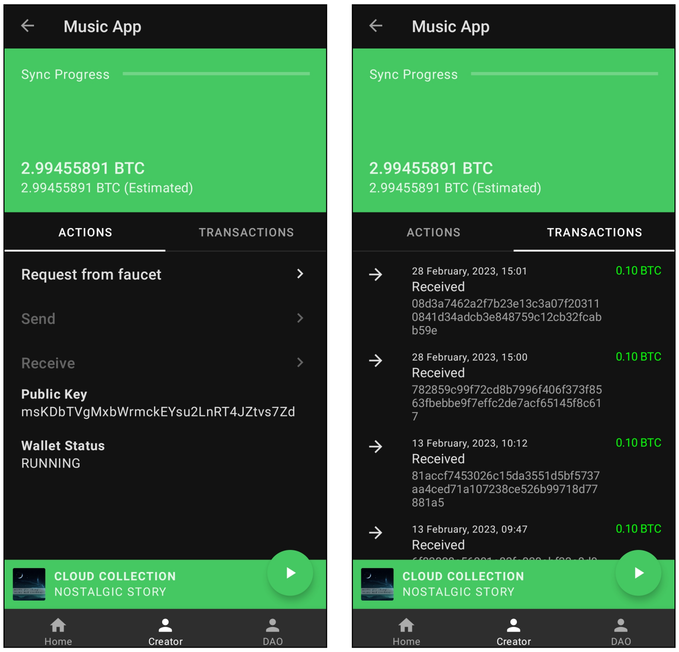
platform and the DAO itself. The music platform serves as a hub for disseminating music and its associated metadata, ensuring accessibility for artists and listeners. The DAO enables collective asset management, empowering listeners to collectively fund new projects from their favorite artists.

A. Implementation and Deployment

We use our Simple DAO architecture and design in Sections IV and V to create the Music DAO. Our implementation spans 8.661 lines of Kotlin code and can be found on our Github[3]. We have successfully deployed our Music DAO on the Google Play Store [4]. An overview of our implementation is visualized in Figure 4. In the following sections, we discuss the implementation of the Music DAO and the accompanying music platform, including the UI and UX of our application.

The DAO runs on Android and is integrated into the TrustChain Superapp[5], an Android application written in Kotlin housing many other applications built on top of IPv8 and TrustChain. Our DAO solely makes use of smartphones, since they have a low barrier to entry and can upkeep peer-to-peer networks through background services. The choice of Android as the platform for our DAO is driven by its open-source nature, extensive service APIs, and the capability to maintain peer-to-peer networks through background services.

[3] https://github.com/Tribler/trustchain-superapp/pull/123
[4] https://play.google.com/store/apps/details?id=nl.tudelft.trustchain
[5] https://github.com/Tribler/trustchain-superapp



2.99455891 BTC
2.99455891 BTC (Estimated)

Request from faucet
Send
Receive

Public Key
msKDbTVgMxbWrmckEYsu2LnRT4JZtvs7Zd

Wallet Status
RUNNING

28 February, 2023, 15:01 — Received — 0.10 BTC
08d3a7462a2f7b23e13c3a07f20311
0841d34adcb3e848759c12cb32fcab
b59e

28 February, 2023, 15:00 — Received — 0.10 BTC
782859c99f72cd8b7996f406f373f85
63fbebbe9f7effc2de7acf65145f8c61
7

13 February, 2023, 10:12 — Received — 0.10 BTC
81accf7453026c15da3551d5bf5737
aa4ced71a107238ce526b99718d77
881a5

13 February, 2023, 09:47 — Received — 0.10 BTC

CLOUD COLLECTION
NOSTALGIC STORY

(a) The overview screen of the wallet

(b) A list of transaction from the personal wallet

Fig. 6: The integrated Bitcoin lite wallet

To facilitate peer-to-peer communication, we utilize IPv8, a networking layer that enables the establishment of overlay networks without relying on central infrastructure. IPv8 supports authenticated communication with privacy, utilizing public-key cryptography. It can establish connections through firewalls (NATs) and even using Bluetooth connections. Messaging is done through UDP for efficiency reasons. These characteristics align well with our implementation's requirements for decentralisation and public-key cryptography. Our communication is done in our own overlay network, which restricts our interactions to peers who are directly involved with our Music DAO.

In terms of the distributed ledger technology (DLT) used for shared assets, we have opted for Bitcoin. Bitcoin is the most long-standing and secure blockchain currently deployed. Given the bandwidth and storage limitations of smartphones, running a full Bitcoin node is impractical with a full node exceeding 400GB worth of transactions in 2023. We do not comprise and use the API of a "trusted" full node. Hence, we employ an SPV (Simplified Payment Verification) node using the BitcoinJ library, which stores and validates block headers while connecting to full nodes. This approach allows us to achieve lightweight node operation on smartphones. We created a separate page to let users manage their crypto wallet, seen in Figure 6. Users can view their addresses, transactions and request funds from a faucet. Please note that our current implementation connects to a Bitcoin Regtest network instead of the full live Bitcoin network. The BitcoinJ library lacks the necessary functionality for our governance protocol, and alternative updated libraries in Kotlin are not available at present. Connecting to the full Bitcoin network is considered future work.

Additionally, we leverage TrustChain, another DLT, to distribute metadata within our Music DAO. TrustChain enables peer-to-peer transaction creation and maintains personal ledgers for each user, facilitating efficient distribution of metadata such as proposals, votes, and music albums. Transactions are in the form of blocks which can contain arbitrary data. These blocks form a blockchain structure and represent the personal ledger. Metadata such as proposals, votes, and music albums are stored using blocks in these ledgers. These blocks can be sent, broadcast, and accessed by other peers, allowing for the sharing and retrieval of metadata across the network. To ensure the availability of metadata blocks to all peers in the overlay network, we have implemented a simple gossiping protocol. This protocol involves periodic broadcasting of blocks to a fixed number of peers, enabling widespread access to metadata.

1) Music DAO: The main functionality of the DAO is the management of shared assets in the form of Bitcoin using our governance protocol. We implement our design specified in Section V to fulfill this.

The Music DAO hosts many DAOs within it, each catering to the diverse music tastes and investment preferences of users. Fans of specific artists can gather and establish dedicated DAOs for those artists. In Figure 5c the screen with the list of all DAOs can be seen. On this screen, users can create a new DAO. Anyone can make a new DAO. They must first specify an entrance fee and threshold percentage for votes. A transaction is then created and published on Bitcoin and this metadata is disseminated and stored on TrustChain. Other users receive this metadata and can view the new DAO on the screen and attempt to join it. The current members must then vote on the join proposal. The list of proposals within a DAO can be seen in Figure 5d. If sufficient members in the DAO vote in favor, the aspiring members will be added to the DAO. Once a DAO is established with assets and members, its assets can be invested in artists. Artists disseminate their Bitcoin addresses using a special artist block on TrustChain. Any member of the DAO can propose to invest in an artist. This is made convenient by showing all the possible artists and accompanying addresses as an option to invest in when proposing. If the vote is successful, the specified amount of assets will be sent to that artist. In addition to this, users can donate directly to artists themselves on the artist profile screen.

Table I provides a comprehensive comparison of the multi-signature schemes we evaluated for our project. We considered various options, including MuSig, which is a Bitcoin-compatible multi-signature scheme that utilizes Schnorr signatures. MuSig2 [18] is an improvement over MuSig and stands out for its efficiency, requiring one less round of communication compared to MuSig. FROST [14] is the state-of-the-art threshold signature scheme. While FROST offers advanced features, it is complex and requires substantial engineering effort to implement and thus we consider this out of scope. In any case, adopting any of these schemes for our governance protocol significantly reduces the storage requirements.

Considering our specific project requirements and constraints, we opted to choose MuSig due to its comparatively low complexity and the availability of an early Python prototype version. We have developed a Kotlin implementation of the MuSig [16] scheme. This implementation, the first of its kind in Kotlin, is based on an early Python prototype which we ported and bug-fixed[6]. Notably, we have chosen to avoid implementations written in C++ and use code bridges. We modify the algorithm to support the specification of Schnorr signatures in Bitcoin described in BIP340 [8], which has several cryptographic and encoding caveats we successfully worked around. It's important to note that a limitation of governance capabilities is the requirement for 100% voting participation. A single missing vote blocks any proposal. A BIP340-compatible version of FROST would solve this problem but is out of the scope of our work.

2) Music Platform: The core features of the music platform are the streaming of music and the discoverability of music and artists. We consider these features essential since they are the first step towards competing with industry platforms such as Spotify. We have implemented these functionalities and integrated them into the SuperApp, right alongside our DAO functionality.

Streaming of music without centralized infrastructure is implemented using the BitTorrent protocol. BitTorrent has a proven track record of stability and security, with 19 years of incremental improvements to the protocol. While other technologies such as IPFS offer similar functionality, BitTorrent is more widely adopted and has a larger user base. Using BitTorrent, we can avoid large centralized data centers for music streaming and instead rely on the peer-to-peer transfer of audio files from phone to phone.

Music is presented in the form of albums and exists in two forms: metadata and actual binary files. Binary files include audio files and album cover art. The ID3 metadata in the audio files is used to further enrich the albums with i.e. genre information. Metadata is in the form of TrustChain blocks, disseminated in the network using our previously mentioned gossiping strategy. This TrustChain block contains the album title and date, a reference to the artist's public key, and the magnet link needed to download binary files. The artist's public key can be used to query locally available blocks on the device and find more albums by that artist. Every album and its block also have a UUID that uniquely identifies it in case of duplicate albums. The magnet link is used to query the BitTorrent DHT to retrieve complete torrent information and find peers seeding the album. This is a distributed hash table of torrent users providing this information. With this information, users can join torrent swarms and start streaming albums. Streamed albums are cached locally for later playback and seeding purposes. This streaming process is also visualized in Figure 4.

We assume a form of altruistic seeding from users on the platform using a seeding strategy. Solving the problem of selfish seeding is considered out of scope. Strategies such as tit-for-tat can be implemented to further incentivize users to make their bandwidth and local storage space available. Clients cannot seed all their cached music due to limited bandwidth and must use a seeding strategy to choose which albums to stream. This strategy can be optimized to increase music availability across the network, which is especially challenging since music demand varies greatly. Due to the lack of popularity metrics on our platform, we opt for a simple strategy in which a random set of albums are seeded to other users.

An artist wishing to publish an album can do so from their phone. They must provide a set of audio files, cover art, and required metadata. The binary files are packaged into a torrent, and a TrustChain block is serialized and added to their personal ledger before being disseminated. At the same time, the phone starts seeding the album to ensure its availability for initial users to stream. Since blocks are cryptographically linked in a blockchain, artists can prove their history of publishing albums and show that they are worthy of investment. Additionally, artists can publish information about themselves, such as a biography and a Bitcoin address, in the form of a TrustChain block so that other users can find more information about them after discovering their albums.

Figure 5a showcases the list of discovered albums, while Figure 5b showcases the music playback screen. Our UI/UX efforts focus on simplicity and intuitiveness. We present users with a single list of discovered albums that can be easily searched using the search bar. The list shows all albums whose album blocks have arrived at the user's device through gossiping. When users select an album, songs start downloading immediately, with a sequential downloading approach ensuring quick playback of the first few seconds of each song. On artist profile pages, users can access all the artist's songs and have the option to donate Bitcoin from their personal wallet. This personal wallet can be seen in Figure 6.

To bootstrap the platform for early users, we curate a dataset comprising hundreds of albums with Creative Commons licenses, obtained from PandaCD[7]. This initial dataset is seeded from a single phone and serves as a valuable resource, allowing early users to explore and enjoy a diverse collection of music while the platform grows.

VII. Performance Analysis

In this section, we present an analysis of our implementation's performance. We analyze the performance of our governance protocol described in Section V-A and perform a limited set of end-to-end performance evaluations for our music platform. We measure the time to first screen load, UDP packet, and album discovery. We also perform a real-life deployment test to validate our application involving

[6] https://github.com/bitcoinops/taproot-workshop

[7] https://pandacd.io/

experts in the field of DAOs, who actively engaged with our implementation.

We measure the performance of our governance protocol described in Section V both in terms of its cryptographic performance and its performance in a networked setting. The purpose of our analysis is to explore whether our protocol is capable of supporting large DAOs, and if not, which trade-offs have to be made.

For both experiments, we measure the time it takes to create an aggregated shared public key and a joint signature of a constant 32-byte string using our BIP340 [8] MuSig implementation with an increasing number of individual keys or nodes. We do not concern ourselves with making this string a Bitcoin transaction and validating it against the Bitcoin network, since our goal is to analyze our governance protocol and not the Bitcoin network. We conducted the experiments on an Android Emulator emulating a Pixel 2 Phone to simulate the performance of our implementation on smartphones while allowing us to have easier debug access for experiments. This setup closely resembles the hardware specifications of typical smartphones and allows us to analyze the performance under realistic conditions.

A. Cryptographic Performance

Firstly, we measure cryptographic performance to get insight into a best-case runtime. This experiment runs in a single application process on the emulator. Before the experiment, all individual keys are generated and cached in memory, since public keys of all participants are known in practice as well. During the experiment, all the individual keys are aggregated into a shared public key. The individual keys are stored in memory and are directly accessible by the MuSig scheme. Afterward, a joint signature over a 32-byte message is signed using the partial signatures of all the individual keys. Lastly, the signature is verified to make sure it is correct. We run the experiment for up to 10.000 individual keys with a 100-key interval, to have the experiment run in an acceptable time amount in the range of hours while exploring large key amounts. We conducted the experiment 10 times to account for the presence of non-deterministic processes in the scheme.

As shown in Figure 7, the run-time of both key aggregation and joint signing scales linearly with the number of nodes. 10,000 individual keys are aggregated in ±12.5 seconds and a message can be signed in ±2.5 seconds. Key aggregation takes considerably longer than message signing, which can be attributed to the number of elliptic point multiplications required for aggregation compared to signing [16]. This difference can be disadvantageous for new DAOs as opposed to established DAOs. In new DAOs, aggregation of keys is more common due to the influx of new members, making them more impacted by this difference. The linear increase in runtime for public key aggregation and signing as the number of nodes increases suggests that scalability may be a concern when scaling the governance protocol to accommodate millions of users. While the cryptographic performance remains reasonable for most consumer-grade hardware, further optimization or alternative approaches may be necessary to ensure efficient performance at larger scales.

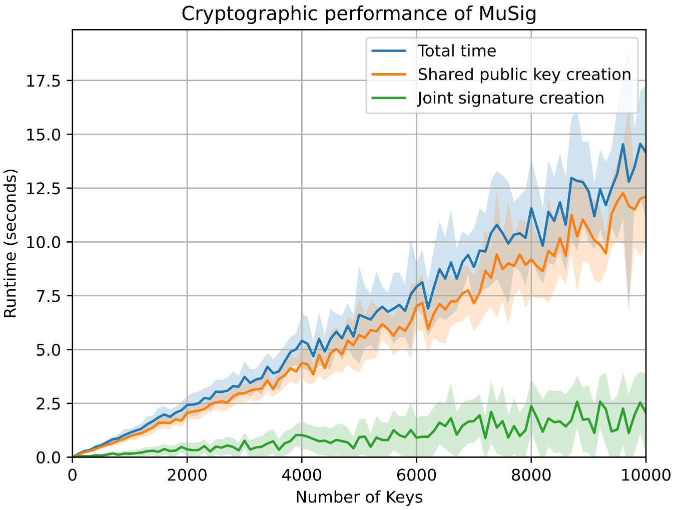


Fig. 7: Cryptographic cost of democratic voting using our governance protocol

We also observe that the standard deviation can be quite large, as indicated by the shaded region. Upon further inspection, we determined this is due to the BIP340-specific changes made to MuSig. In BIP340, public keys are encoded so that the y-coordinate is always "even". If this is not the case, the point is "negated". The definition of "even" and "negated" in this context refer to specific elliptic curve operations that require expensive elliptic point multiplication operations. In 50% of the cases, the shared public key will be odd, requiring all participants to negate their individual keys. This process results in a significant increase in runtime in 50% of cases.

Figure 8 shows the flame graph of the cryptographic operations of a single key aggregation and joint signing round for 10.000 individual keys. The purpose of this graph is to gain insight into the resource utilization of our implementation and identify potential bottlenecks. The entire experiment takes ±12.5 seconds. We observe that 60% of the time is spent on aggregating the public key and ±15% is spent on aggregating the nonces, creating the partial signatures, and combining these signatures into the final signature. The rest of the time of ±25% 8s mostly used for the negation of keys, as described earlier.

These findings are consistent with our cryptographic performance results and suggest that shared public key aggregation is the most computationally demanding cryptographic task. This is because it requires the multiplication of elliptic curve points, while other operations either do not require such multiplications or require only a constant number of them. Furthermore, we observe that the negation of keys is an expensive task because it requires generating a new public key for each negation. This is an artifact of the Bitcoin specification of Schnorr signatures and can be avoided if necessary by using other blockchains that do not require this encoding.

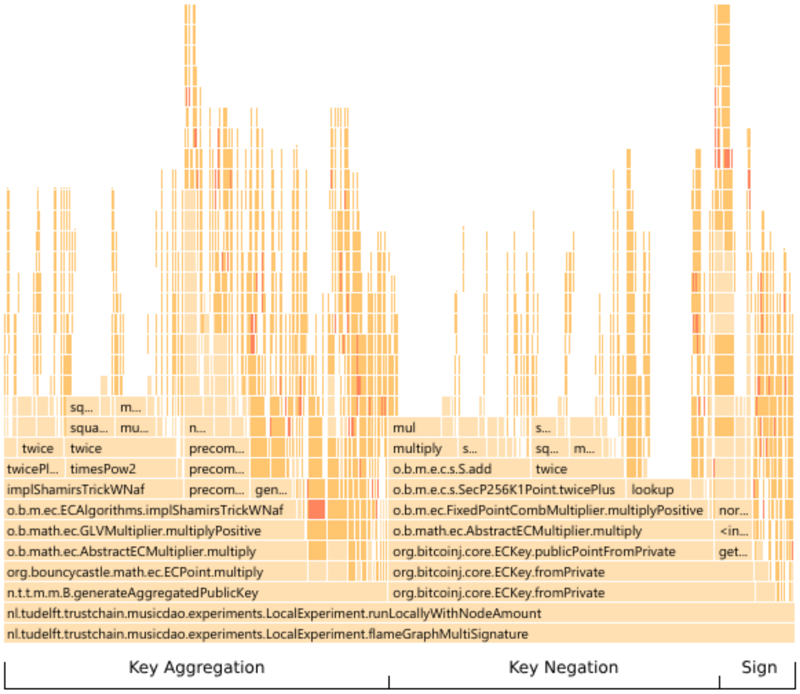


Fig. 8: Flame graph of the cryptographic operations in the governance protocol for 10.000 keys

B. Networked Performance

To get insight into the viability of this governance protocol in real-world settings, we evaluate the performance in a networked peer-to-peer setting. As described in Section VI, our deployed implementation is based on a gossiping protocol using TrustChain blocks. Since we have not optimized our gossiping protocol and to simplify our evaluation, we implemented an additional simple event-based IPv8 protocol using UDP messages which assumes full connectivity among all peers. While this setup allows us to assess the protocol's performance under optimal conditions, it does not account for the challenges and optimizations associated with real-world gossiping protocols or the constraints imposed by the UDP packet size limit.

We run all IPv8 nodes on a single emulator, each assigned to a unique port using our local network IP address. This minimizes latency since all packets are confined to a local network. The nodes run the aggregation and signing collectively using the protocol and a special single node measures and stores the run time. The experiment is repeated for up to 20 nodes 10 times. The node amount limit is due to certain messages scaling with the number of nodes, eventually exceeding the UDP packet size limit. Although this limitation could be addressed by using protocols such as the EVA protocol [4], it falls beyond the scope of this experiment.

As shown in Figure 9, for 20 nodes, the runtime for aggregating keys is ±2.2 seconds, and for signing it is ±2.1 seconds, resulting in a total runtime of ±4.2 seconds. Comparison between cryptographic and peer-to-peer performance reveals that the latter is the limiting factor, even under our optimal conditions with event-based communication, local networking, and no Bitcoin transaction validation. It must be noted that this outcome can be partially attributed to the inherent packet loss in UDP messaging. Additionally, the Kotlin IPv8 implementation used is not very well maintained and suffers from several bugs and performance issues.

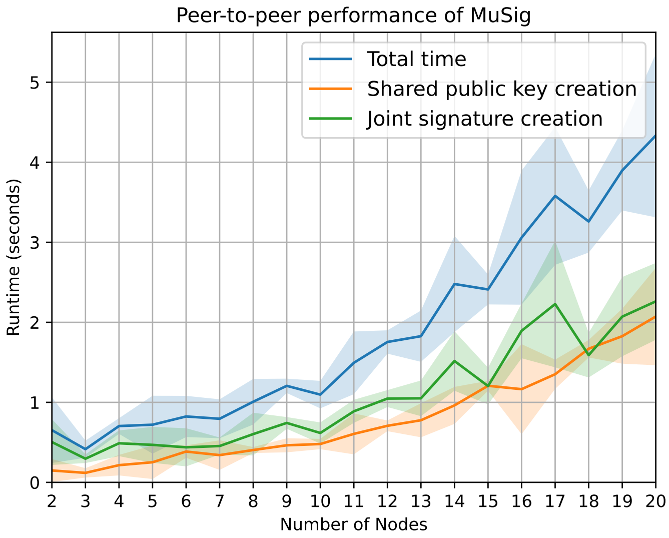


Fig. 9: Performance of democratic voting using governance protocol in networked setting

Furthermore, we observe a reduced time difference between the aggregation and signing processes in terms of runtime. This can be attributed to the fact that both processes require a full round of communication between all nodes. The time taken by the cryptographic operations performed on the nodes is minimal compared to that of the full round communication. We conclude that networking solely bottlenecks the governance protocol and not cryptographic operations.

If voting is required to be time-sensitive, a peer-to-peer governance protocol using P2P is not feasible for a large amount of DAO users. Here, we define "time sensitivity" in voting as the necessity to reach a decision within a very short timeframe, typically within seconds. An example of such time-sensitive voting is making investment decisions based on rapidly fluctuating financial market activities. However, voting where time is not highly sensitive can make use of this protocol. For instance, voting on funding an album for an artist can be held open for days if needed, allowing enough time for votes to be collected and combined using the peer-to-peer protocol. Furthermore, note that smart-contract-based governance faces similar limitations in highly time-sensitive voting, while additionally being constrained by transaction space requirements on the blockchain and the transaction speed of the underlying blockchain infrastructure.

C. End-to-End Performance

We measure various aspects of the application's performance, including the time required for the initial loading of the application. We categorize this into a pipeline of processes to identify any bottlenecks or variations in time. The measured metrics include the time to load the application, the arrival time of the first UDP packet, the processing time of the TrustChain block containing music metadata, the rendering time of the UI logic, and the total time until the music content is displayed.

To conduct the experiment, we designate one phone as the seeder phone, preloaded with a library of two hundred albums. Another phone, referred to as the benchmark phone, receives the new music. Both phones are connected to a shared local network. We repeat the experiment ten times and present the results in Figure 10.

The total time to the first display of music is on average under ±3.2 seconds, which we consider a reasonable loading time for users' first load. Subsequent openings of the application will show locally cached music, resulting in instant access and reduced loading times. We observe that processing the TrustChain block and UI logic takes considerable time. We hypothesize that the UI logic can be substantially optimized through further engineering efforts. Note that in a setting with more phones, this time until the display of music will decrease due to more releases being gossiped to the receiver phone. This result can thus be interpreted as an upper bound for our application's loading performance.

Lastly, in order to evaluate the usability of our application, a real-life deployment test was conducted. A picture can be seen in Figure 11. Participants were given a presentation on DAOs and were subsequently provided access to the application which is deployed on the Google Play Store. Through the deployment test, we acquired practical insights into how users perceived and utilized the application. User feedback during this real-life scenario provided valuable information for refining and improving the application's usability, ensuring that it meets the needs and expectations of its intended users.

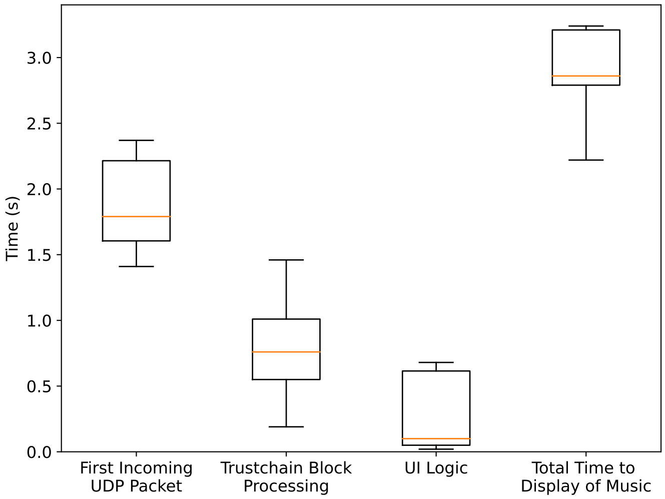


Fig. 10: End-to-End performance of loading albums

VIII. Conclusion

In an increasingly centralized landscape dominated by big tech companies, Decentralized Autonomous Organizations (DAOs) offer an alternative through a bottom-up approach to collaboration on the Internet. While DAOs are supposed to be decentralized, many DAOs still suffer from centralization in both their infrastructure and governance. In this work, we proposed a simple and robust architecture for DAOs that allows for economic activity while maintaining complete decentralisation. To realize this, we present a novel governance protocol based on multi-signature schemes which enable off-chain voting, resulting in reduced blockchain usage and increased scalability. This protocol empowers participants to collectively manage shared assets in the form of cryptocurrencies. To demonstrate the viability of our architecture, we implement and deploy the Music DAO, a fully decentralised smartphone-based DAO serving as a music platform. Our DAO enables listeners to collectively invest cryptocurrency into artists and listen to them without the use of any intermediaries. Performance analysis of our implementation demonstrates that our governance protocol is suitable for non-time-sensitive voting scenarios and its limitations are primarily due to communication overhead rather than cryptographic operations.

References

[1] Multi-signature Specification of Bitcoin. https://en.bitcoin.it/wiki/OP_CHECKMULTISIG. Accessed: 2023-06-21.

[2] Ian Appel and Jillian Grennan. Control of decentralized autonomous organizations. *Available at SSRN 4322917*, 2023.

[3] Henrik Axelsen, Johannes Rude Jensen, and Omri Ross. When is a dao decentralized? *arXiv preprint arXiv:2304.08160*, 2023.

[4] Joost Bambacht and Johan Pouwelse. Web3: A decentralized societal infrastructure for identity, trust, money, and data. *arXiv preprint arXiv:2203.00398*, 2022.

Fig. 11: Real-world deployment test

[5] Cristiano Bellavitis, Christian Fisch, and Paul P Momtaz. The rise of decentralized autonomous organizations (daos): a first empirical glimpse. *Venture Capital*, 25(2):187–203, 2023.

[6] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[7] Lin William Cong, Zhiguo He, and Jiasun Li. Decentralized mining in centralized pools. *The Review of Financial Studies*, 34(3):1191–1235, 2021.

[8] Bitcoin Core. BIP0340 - Schnorr Signatures for secp256k1. https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki. [Accessed 30-Jun-2022].

[9] Vikram Dhillon, David Metcalf, Max Hooper, Vikram Dhillon, David Metcalf, and Max Hooper. The dao hacked. *blockchain enabled applications: Understand the blockchain Ecosystem and How to Make it work for you*, pages 67–78, 2017.

[10] Ethereum Foundation. Daos, dacs, das and more: An incomplete terminology guide.

[11] Ralph Hertwig. Tapping into the wisdom of the crowd—with confidence. *Science*, 336(6079):303–304, 2012.

[12] Matthew Hindman. The myth of digital democracy. In *The Myth of Digital Democracy*. Princeton University Press, 2008.

[13] Martin Kleppmann, Adam Wiggins, Peter Van Hardenberg, and Mark McGranaghan. Local-first software: you own your data, in spite of the cloud. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pages 154–178, 2019.

[14] Chelsea Komlo and Ian Goldberg. Frost: flexible round-optimized schnorr threshold signatures. In *International Conference on Selected Areas in Cryptography*, pages 34–65. Springer, 2020.

[15] Bartosz Kusmierz and Roman Overko. How centralized is decentralized? comparison of wealth distribution in coins and tokens. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2022.

[16] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multi-signatures with applications to bitcoin. Cryptology ePrint Archive, Paper 2018/068, 2018. https://eprint.iacr.org/2018/068.

[17] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, Dec 2008.

[18] Jonas Nick, Tim Ruffing, and Yannick Seurin. Musig2: Simple two-round schnorr multi-signatures. Cryptology ePrint Archive, Paper 2020/1261, 2020. https://eprint.iacr.org/2020/1261.

[19] OpenZeppelin. Governance - openzeppelin governer. https://docs.openzeppelin.com/contracts/4.x/api/governance, 2023. [Accessed 05-Jun-2023].

[20] Gopal Pandurangan. *Distributed Network Algorithms*, pages 15–18. Gopal Pandurangan, 2023.

[21] Dilip Kumar Patairya. What is ApeCoin and how does it work? — cointelegraph.com. https://cointelegraph.com/news/what-is-apecoin-and-how-does-it-work, 2022. [Accessed 21-Jun-2023].

[22] Johan Pouwelse. Towards the Science of Essential Decentralised Infrastructures. In *Proceedings of the 1st International Workshop on Distributed Infrastructure for Common Good*, pages 1–6, Delft Netherlands, December 2020. ACM.

[23] Alex Preukschat and Drummond Reed. *Self-sovereign identity*. Manning Publications, 2021.

[24] Solend. Solend: algorithmic, decentralized lending and borrowing protocol on Solana. https://solend.fi/, 2023. [Accessed 21-Jun-2023].

[25] Ruth Towse. Dealing with digital: the economic organisation of streamed music. *Media, Culture & Society*, 42(7-8):1461–1478, 2020.

[26] Shuai Wang, Wenwen Ding, Juanjuan Li, Yong Yuan, Liwei Ouyang, and Fei-Yue Wang. Decentralized autonomous organizations: Concept, model, and applications. *IEEE Transactions on Computational Social Systems*, 6(5):870–878, 2019.

[27] E Glen Weyl, Puja Ohlhaver, and Vitalik Buterin. Decentralized society: Finding web3's soul. *Available at SSRN 4105763*, 2022.

[28] Qiheng Zhou, Huawei Huang, Zibin Zheng, and Jing Bian. Solutions to Scalability of Blockchain: A Survey. *IEEE Access*, 8:16440–16455, 2020.