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

***Abstract*—Blockchain technology has allowed for the emergence of a new type of organization, Decentralized Autonomous Organizations (DAOs). They have gained significant traction in recent years, reaching market capitalizations of up to 60 billion USD. These organizations can 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 artists to distribute their work in a decentralized manner enabling listeners to collectively invest into 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 new fundamental way for people to organize themselves in society. 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 [12]. Bitcoin has solved the problem of collective decision-making without a trusted third party by making an immutable ledger possible [18], 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 increasing emergence and popularity of decentralized protocols highlight their potential for fostering collaboration between individuals.

DAOs have a long-standing history, with the the first DAO deployed a decade ago on Ethereum named "The DAO" [8, 10]. Since then, the number of deployed DAOs has grown exponentially. In 2021 there were over 2,000 DAOs deployed on Ethereum alone with an aggregated market capitalization exceeding $60 billion [7]. 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 [4] and is governed by its own token. Members can manage the collective funds and change the rules of the exchange by voting with their tokens on proposals. Tokens were initially sold through an initial coin offering (ICO) and are now traded on exchanges.

Despite the rapid development of this paradigm, many of them exhibit forms of centralization in both their governance structure and technical infrastructure. This centralization is reflected in the lack of true decentralized governance. For instance, the second-largest DAO by market capitalization, APE DAO, is characterized by an initial token distribution in which 38% of tokens were distributed to various founders. Since every token is equivalent to 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 foundation members of the DAO. Another example is Solend, one of the largest decentralized lending systems. In 2022, there was an incident in which the development team took control of and liquidated the account of a whale with approximately $170 million worth of cryptocurrency. The team claimed it allegedly posed a systemic risk to the ecosystem at the time. 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 its inability to decentralize every component without compromising in 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. 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, Bitcoin will have a centralized market structure [9]. Proof-of-stake distributed ledgers run the risk of reinstating a centralized elite. To validate the network, a substantial amount of capital must be placed at risk. This set of validators can then be subjected to regulatory pressure or collide 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 with basic governance, membership voting, and treasury management is a key step forward in achieving this goal. To demonstrate the feasibility of this 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.

1) **The Simple DAO Architecture** We design and justify an architecture for DAOs which is completely decentralized. To achieve this, we propose a set of requirements and components that must be used. In particular, we make a distinction between a governance protocol and a separate settlement mechanism for decisions.
2) **Music DAO: a truly decentralised DAO** We design and implement a real-world DAO that revolves around the music industry using the our simple DAO architecture. We use a combination of networks, including the TU Delft created IPv8, to 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) **Evaluation** To evaluate the proposed infrastructure and implementation, we perform a set of performance tests on our governance protocol to assess the performance. Furthermore, we assess a number of aspects of our music platform. Additionally, we have done a real-world test amongst a set of people interested in DAOs. The results of these tests provide insights into the feasibility and effectiveness of our proposed architecture and implementation.

II. Problem Description

Participants in traditional organizations have a diminishing level of influence on decision making. Even if they have influence, it often is an outdated and slow process (democracy) or relegated to a select wealthy group (shareholders). 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" . Top-down hierarchies and layers of managers result are required to enforce rules. Without enforcement of rules, participants who do not trust each other do not corporate due to their conflict of interest. Rules are enforced by third-party authorities, such as the legal system or boards of companies. However, their interest may in turn not align with the interests of participants. They can alter the rules or not follow them at all. 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 soflved by the advent of Bitcoin [18] and has allowed for the emergence of organizations without any central intermediaries: DAOs.

The difficulty in creating a decentralized autonomous organization is simultaneously achieving trust, complete decentralization and scalability. The problem is similar in nature to the blockchain trilemma [24], with the inclusion of decentralization in terms of governance [16]. Currently every technology claiming to be a DAO has central points of control and critically rely on central servers. Real decentralized DAOs only exist in theory. Bitcoin and Bittorrent are the only examples of technology stacks which are not reliant on central infrastructure.

In addition, implementing and deploying a DAO is a 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 performance requirements of cryptographic operations. Most importantly, security must be guaranteed since large financial transactions may depend on the code.

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. 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 which relies on no central intermediary nor central authority and one which is truly unstoppable.

III. Related Work

The concept of Decentralized Autonomous Organizations (DAOs) is relatively new in academia, leading to a scarcity of academic analysis on decentralization 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 pertaining to the history of DAOs, efforts to create theoretical frameworks and architectures for DAOs, analysis of current DAOs and efforts to define decentralization in DAOs.

Vitalik Buterin introduced the concept of DAOs early on in his Ethereum whitepaper and in a 2014 blog post [11]. He described the ideal DAO as än 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 a Ethereum blockchain fork to rectify the situation [10]. 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 [22]. They introduce a five-layer architecture for DAOs separating governance, technology, incentives, organization and a manifestation layer. They do not, however, give a concrete implementation of such a DAO utilizing the design. Qin et. al make a similar contribution by identifying fundamental principles and requirements for DAOs derived from the three terms present in its definition. Their proposed architecture consists of an organizational, coordination, execution and application layer. The infrastructure in the execution layer is discussed very briefly and has not enough requirements and specification. Both papers lack in specifying and defining the decentralized infrastructure of DAOs.

Decentralization plays a crucial role in the design and effectiveness of DAOs. Axelsen et al. developed a framework that quantifies decentralization through expert and literature reviews, employing five dimensions with specific quantifiers. Appel et al. discovered a high level of centralization in decision-making processes within current DAOs, with the top token holders significantly influencing the voting outcomes. Additionally, Bellavitis et al. conducted empirical work, observing a steady growth in the number of DAOs, active users, and proposals across the ecosystem.

Several papers have focused on defining and quantifying decentralization within a DAO. Our work also focuses on the decentralization aspect of DAOs and attempts to identify requirements which ensure decentralization. Axelsen et al. created a general framework for assessing decentralization through expert and literature reviews [5]. This framework consists of five dimensions, each with their own quantifiers. For governance, they for instance define the amount of distinct persons needed for a 51% vote as an indication for decentralization. Appel et al. show that decision-making in current DAOs is highly centralized. 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.

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 complete decentralisation. We first elaborate on our requirements, then go over our required infrastructure 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.

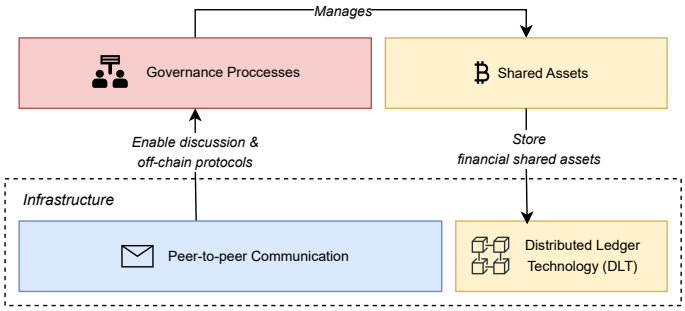


Fig. 1: The Simple DAO Architecture

A. Architectural Requirements

Our architectural requirements are based on the principle of decentralization and the zero-server architecture [20]. This architecture provides design principles to realize infrastructure for the common good, which include having no hierarchy in networks, no intermediaries and democratic decision-making processes. Using these principles as a base we identify three requirements which are necessary for completely decentralized DAOs.

1) Trustless: Interactions between participants must not require any inherent trust. Instead, distributed protocols based on crytography should be used which can independently be verified by each participant. This includes cryptographic protocols such public-key cryptography and consensus mechanisms based on incentives such as proof-of-work. Lack of required trust ensures that no intermediaries are needed to provide that trust, which is essential in a DAO. Furthermore, it ensures decision-making processes are verifiable fair and no cheating can occur.

2) Permission-less: Anyone should be able to participate in the organization, without needing approval of centralized authorities. They should not be discriminated based on factors which are not relevant for the workings of the DAO. This does however mean that members in the organization can still collectively decide to block or not allow a person in the organization. Permission-less promotes decentralization, since barriers of entry are removed.

3) Transparent: All information regarding the organization should be available and visible for everyone. This includes all information about participants and their actions, decision-making processes and other relevant data. It enables participants to inspect and verify the state of the organization and make informed decisions, without any unfair information asymmetry. Furthermore it ensures participants can be held accountable for their actions. Transparency should be for both internal and external stakeholders to help foster trust between the organization and the wider community.

B. Infrastructure

Decentralized infrastructure is a necessity to realize our goal. In particular, we make use of P2P networks to enable communication without intermediaries. We now describe the technologies we deem necessary in a DAO infrastructure.

1) Distributed Ledger Technology: Distributed Ledger Technology (DLT) enable secure financial transactions without centralized parties. Data is replicated and validated across a network of nodes communicating in a peer-to-peer network. Blockchain is the most commonly used DLT, a tamper-resistant data structure consisting of chained blocks of transactions which are cryptographically linked. A consensus mechanism, such as Proof of Work [18], ensures that once information is recorded into the ledger, it is very extremely to change. This property solves the problem of *double spending* in practice and provides the high amount of trust required for financial transactions. Additionally, some non-financial information can also benefit from high availability and immutability.

We deem that a DLT must be open-source, permission-less, transparent and sufficiently decentralized in order for it to adhere to our architectural requirements. Open-source code ensures that code cannot be maliciously changed, which is essential to verify security.

Permissioned networks are not democratic in nature and can easily be colluded within. The requirement for permission to join such networks introduces the potential for collusion, as nodes have the authority to selectively add nodes that align with their own interests or beliefs. The network must be permissionless to enable open participation and foster decentralisation due to more nodes verifying the network. Transparent transactions allow for verification of governance processes and allows members to hold each other accountable. The notion of sufficient decentralization can be measured in terms of difficulty to attack the network, the age of the network and a number of other measures [16]. Without this decentralization, components such as governance run the risk of becoming centralized again.

2) Peer-to-peer Communication: In order to coordinate governance and other activities, participants need to 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 i.e. governance processes.

A history of communication must be kept in some way for cryptographic protocols to be verified and to allow new participants to make informed decisions. It is important that this must be done using *local-first-data-storage* [14]. Local first data storage entails that network participants themselves are collectively responsible for the availability of data. There should be no special data providers in the network, since this re-introduces centralization. Data is stored on the many devices users nowadays have: smartphones, computers and tablets. Using some protocol built on top of P2P communication, for instance gossiping protocols, data can be replicated to ensure data availability.

Peer-to-peer overlay networks facilitate the aforementioned type of communication. Typical communication methods on the internet such as bulletin boards, forums and social media platforms do not satisfy our strict requirements. They are centralized in nature subject to moderation censorship. Overlay networks abstract away underlying infrastructure and provide authenticated messaging between peers in a decentralized network architecture. To enable authenticated messaging, such an overlay network must use public-key cryptography to make participants identifiable. Overlay networks allow for decentralized protocols to be deployed on top of them, such as our multi-party signature protocol for governance we will describe later in Section V.

C. Components

We build two features which we deem neccesary 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 show our 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 make *proposals*, which can be about anything from fund management to policy changes. Other members participate exchange 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 which enables voting in a trustless manner. After voting, the result must be executed automatically with no human intervention. Iteratively repeating this process enables the organic evolution of the organization, driven by 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 execution of the vote can be stored and executed on-chain in a 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 which 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 platform such as Snapshot[1]. It is not trustless, since some external party such as an internal commission must be entrusted to execute the result of the vote. If they decide to collude and for instance not transfer funds, no one can do anything about this.

In order for a DAO to achieve its objectives in an orderly and "fair" manner, a set of governance rules should be established dictating how decisions are made in the organization. Generally, individuals who contribute more and take on responsibility should have more benefits in the decision-making process than others. This can be enabled through digital tokens

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

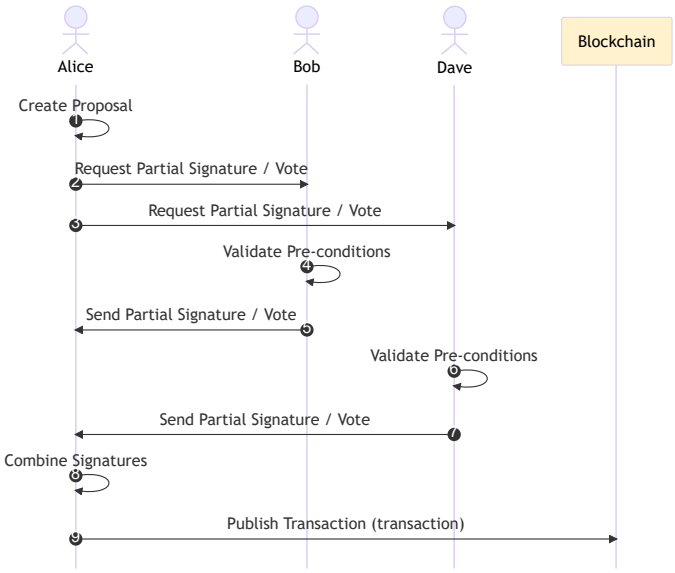


Fig. 2: Sequence diagram of our voting protocol for a transaction proposal

for the DAO itself. This concept is often a matter of debate, and the concept of "fairness" in decision-making is also an open research question still [22]. We do however argue that in the most ideal case a one-human-one-vote model is ideal for organizations concerning themselves with the common good. 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: For a DAO to fund its activities and achieve its objectives, it must have some notion of shared assets. Although DAOs without any assets can rely on altruism to some extent, most of the time financial incentives are needed to make work possible in practice. These assets belong to the members of the DAO and can be managed through governance processes, made possible by DLT. Members should be able to 1) lock funds and 2) transfers fund to some external entity. Cryptocurrencies are the most obvious choice, as they conform to all three requirements we previously established. They can be programmed to be transferred in a trustless manner after a governance vote. This is hard, or perhaps impossible, to do for real-world assets. They can digitized into digital assets, but this requires some entity to keep the assets into custody and act fairly, violating decentralization and trustless-ness.

V. Design

We now introduce our design for the components specified in Section IV-C. The design we describe adheres to our architectural requirements which we deem necessary for decentralization. Our design 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.Our approach involves conducting the voting process off-chain, while storing and executing the resulting votes on the blockchain, as depicted in Figure 3. By adopting this approach, we effectively reduce transaction costs, save blockchain storage space, and decrease the total time required for a vote when compared to existing solutions.

Existing solutions rely on smart contracts. The industry standard for such contracts is the OpenZeppelin Governor v2.0.0, a widely-used smart contract framework for DAO governance in the blockchain industry [2]. The smart-contract stores all the state for the DAO: the proposals, the vote count, and other information. Participants can create proposals and vote on these proposals by creating a transaction from their personal crypto-wallet which interacts with the contract. Interacting here entails changing the state of the contract according to a set of pre-defined rules. Casting a vote increases the vote counter on a proposal. Once the voting period of the proposal ends, the proposal is closed and the result is considered final. The main advantage of utilizing smart contracts for DAO governance is their extendability. Custom smart contracts can incorporate advanced functionalities such as vote delegation, automatic fund transactions after successful proposals, and additional requirements for initiating proposals. However, this approach requires many transactions, consumes significant blockchain space, incurs high transaction fees, and is time-consuming due to limited blockchain throughput.

Our design builds upon established multi-signature schemes [15, 17, 19]. These are cryptographic schemes in which a set of participants jointly have ownership over a single 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 signature valid for the shared public key. Both of these processes can require multiple full rounds of communication in which every participant sends a message to each other. At no point during key creation, aggregation or signing does the private key materially exist. Thresh-hold signatures are a subset of multi-signature schemes and allow for a thresh-hold of partial signatures to be sufficient for signing, while multi-signature schemes require all partial signatures.

While a less complex solution for implementing our protocol on a blockchain like Bitcoin involves using simple scripts [3], we consider this approach to be non-scalable. Bitcoin scripts are stack-based programs that specify under which condition funds from a transaction can be spent. These scripts are stored on-chain in the transactions themselves. Using scripts, a transaction can be programmed to be multi-signature. The approach for this is trivial: one must store a

list of public keys which are required to spend the transaction in the script. To spend the transaction, the accompanying signatures of the keys must be supplied. The list of public keys and the signatures are thus all stored on the blockchain. While this approach avoids all complexity of the multi-signature protocols, this is not a scalable solution. 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 3, participants engage in an exchange of messages within an overlay network, described in Section IV-B2. Each participant possesses a unique public and private key-pair, with all public keys shared among participants. 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 thresh-hold amount in the case of thresh-hold schemes. If the proposal involves a financial transaction on a blockchain, it can be published and accepted 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.

Using our design we can have the members vote off-chain and merely have to store the result of the vote on-chain. In this way, we can achieve a high level of security and scalability, while keeping the complexity of the system at a minimum. We do not rely on general purpose smart contract capabilities. Instead, we use a blockchain of choice, namely Bitcoin, which is simple and secure. We greatly reduce the number of on-chain transactions needed by up to n, n being the number of members in the DAO, compared to governance processes using smart contracts. This lets us avoid the aforementioned problems of smart-contract-based governance.

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 crypto-currency of the blockchain, since we are not using any smart-contract capabilities. At all times the DAO has a *shared public key*. A shared public key is created using a multi-signature scheme and allows the participants to jointly have ownership over a the shared assets. All cryptocurrency locked up using this key in transaction is considered are the shared assets of the organization. Managing these shared assets requires being able to lock up funds, transfer these funds and add or remove people 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 undergoes the governance protocol described earlier, resulting in the creation of a signature. 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. The initial member of the DAO has the authority to define parameters, such as the threshold for decision-making within the DAO. For a new member to join, all funds must be moved and locked under a new shared public key that includes the new member. Typically, the new member must first pay a pre-agreed upon entrance fee to keep the DAO functioning. This requires two sequential transactions: one in which the new member locks up 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* to avoid this problem. The key idea is to enable the new member to atomically pay the entrance fee and join the DAO simultaneously. This is achieved through the creation of an unsigned transaction with multiple inputs and outputs. All distinct parties, the new members and existing members, then can sign the transaction safely since the outputs cannot be changed. The pre-condition checks whether the transaction is sound before signing.

The new member generates a new collective public key, which includes their own, and creates an unsigned transaction. This transaction is visualized in Figure 5. The transaction has two inputs. The first input is the entrance-fee, signed by a personal wallet of the new member. The second input are the previously locked up DAO funds. The output of the transaction are the funds combined, locked up using the new collective public key. Additionally, an output can be added to return change to the new member, since in Bitcoin all outputs of a

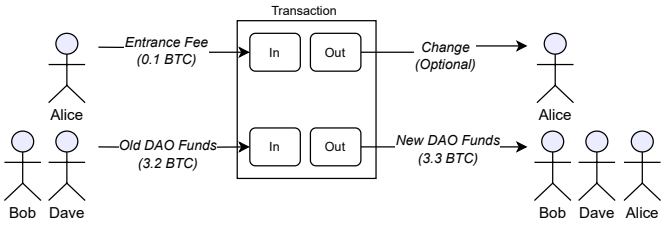


Fig. 3: Transaction which adds a member to the shared assets of the DAO

transaction need a destination for them not to be lost. The transaction is set-up in this way to make it impossible for a new member to join without paying the entrance fee, since it is done atomically.

Similarly to spending funds, this transaction is subject to a voting procedure using our protocol. Before voting, the pre-condition is set-up in such a way that members first validate whether the member actually has signed its entrance fee input. If the vote is successful, the member will have paid the entrance-fee and joined the DAO through key addition.

The procedure for removing a member follows a similar approach, involving the exclusion of their key. Any member can initiate the removal procedure by creating a new key that excludes the targeted member and subsequently transferring the funds to the new key. If a sufficient number of members vote in favor, the member will be successfully removed and lose their voting rights. Furthermore, during this transaction, the members can decide whether to return any remaining funds to the departing member.

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. However, it is important to note that our current design does not address the issue of sybil attacks. A one-human-one-vote [23] model can be implemented using sybil-resistance mechanisms. 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 re-shape 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 to allow listeners to listen to their preferred music and support artists. Various roles such as talent scouting remain, but no longer require any human labour. A music curator is no longer required if real-time viral music statistics exist. Current cloud-based architecture restrict 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 [21]. This allows them to completely focus on music and further incentives 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 permissionsless 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 market share: Spotify, Youtube and Apple Music. The monopolization of this space force artists to succumb to the power of these platforms, in order 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 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[2]. We have successfully deployed our Music DAO on the Google Play Store [3]. An overview of our implementation is visualized in Section IV. In the following sections, we discuss

[2]https://github.com/Tribler/trustchain-superapp/pull/123

[3]https://play.google.com/store/apps/details?id=nl.tudelft.trustchain

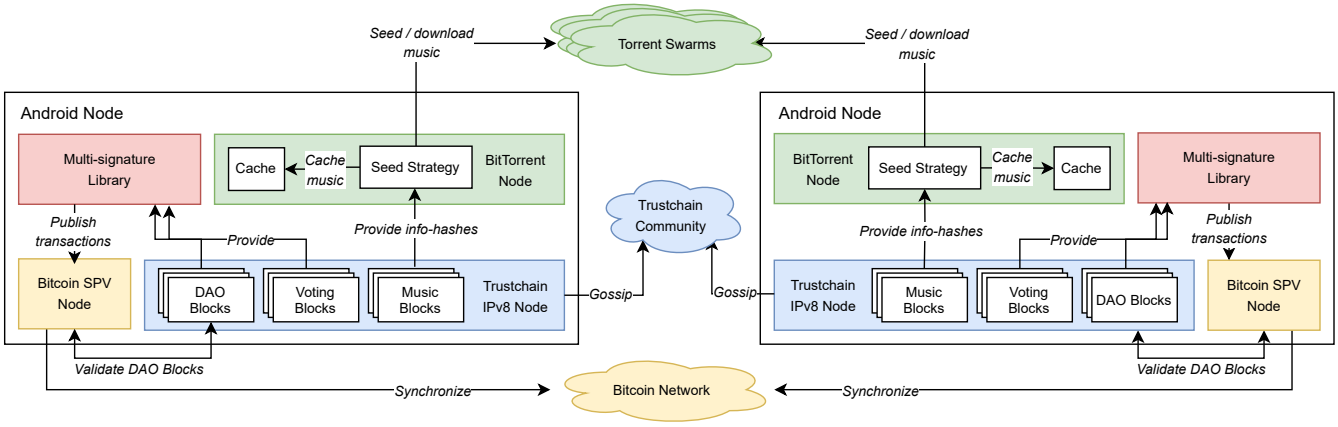


Fig. 4: A visual representation of the Music DAO based on our architecture.

TABLE I: Comparison of different governance protocols for 1 single proposal. pk is the size of public keys. sig is the size of signatures. n is the amount of members participating in the proposal.

Governance Protocol	Type	Year	Transactions Required	Size Single Transaction	Size All Transactions
Smart Contract [2]	Smart Contract	2013	n	$pk + sig$	$n \cdot (pk + sig)$
Naive Bitcoin [3]	Multi-signature	2008	1	$n \cdot (pk + sig)$	N/A
MuSig [17]	Multi-signature	2018	1	$pk + sig$	N/A
MuSig2 [19]	Multi-signature	2020	1	$pk + sig$	N/A
FROST [15]	Thresh-hold signature	2020	1	$pk + sig$	N/A

the implementation of the Music DAO and the accompanying music platform, including the UI and UX of our application, which required considerable effort.

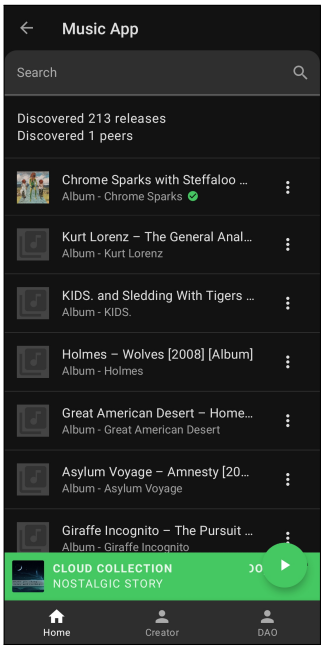
The DAO runs on Android and is integrated into the TrustChain Superapp[4], an Android application written in Kotlin housing many other applications built on-top of IPv8 and TrustChain. Our DAO solely makes use of smartphones, since they have a low barrier to entry and can upkeep peer-to-peer networks through background services. The choice for Android as opposed to other platforms is due to its open-source nature, the availability of an IPv8 implementation on Android and its extensive service APIs that allow services to continuously run in the background, allowing for the upkeep of the network.

Peer-to-peer communication is done through IPv8, a networking layer enabling peer-to-peer overlay network without any central infrastructure. It can establish connections through firewalls (NATs) and even using bluetooth connections. It provides authenticated communication with privacy using public-key cryptography. Messaging is done through UDP for efficiency reasons. These qualities form a natural fit for our implementation, which also requires decentralization and public-key cryptography. We created a custom overlay network specifically for this DAO with our custom functionality.

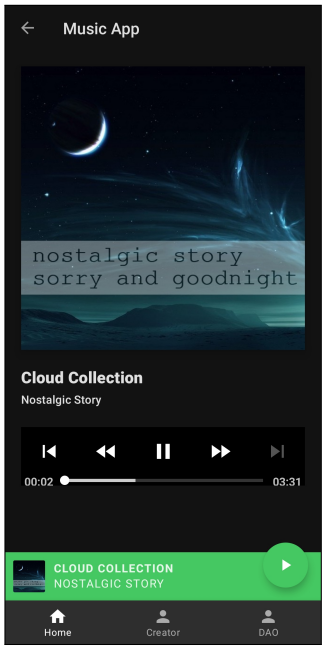
The DLT we opted to use for shared assets is Bitcoin. Bitcoin is the most long-standing and secure blockchain currently deployed. Running a full node on smartphones is infeasible due to bandwidth and storage limitations, with a full node exceeding 400GB of transactions in 2023. We do not comprise and use the API of a "trusted" full node. We run a SPV node, otherwise known as a light node, using the BitcoinJ library. A SPV node only stores and validates block headers and connects to full nodes, allowing us to run a lightweight node on a phone. We created a separate page to let users manage their own crypto wallet, seen in Figure 8. Users can view their address, transactions and request funds from a faucet controlled by us.We currently connect the regtest network instead of the full live Bitcoin network. The functionality we need in order to implement our governance protocol is not available in BitcoinJ, and there is no other updated library available on Kotlin. We consider connecting to the full Bitcoin network future work.

In addition to Bitcoin, we leverage a DLT called Trustchain for distributing meta-data within our Music DAO. Trustchain enables peer-to-peer transaction creation and maintains personal ledgers, allowing for efficient distribution of meta-data 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. Meta-data 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 meta-data across the network.To ensure the availability of meta-data 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 meta-data.

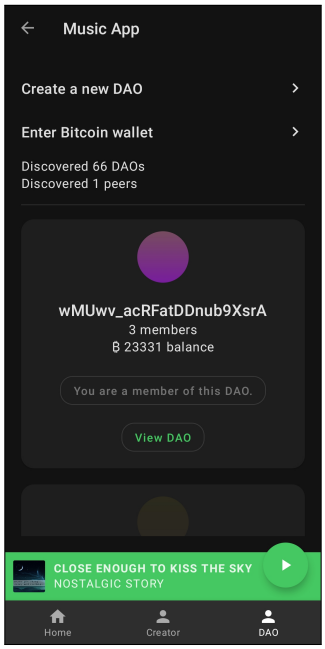
[4]https://github.com/Tribler/trustchain-superapp



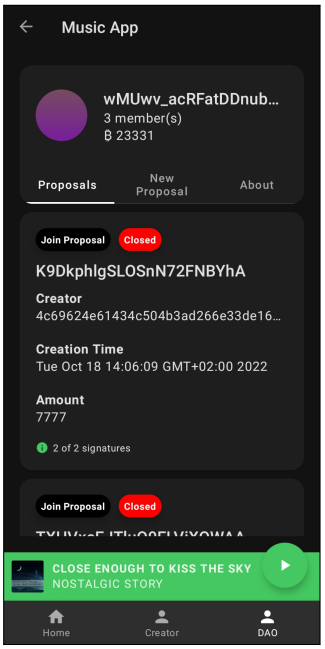
(a) Our album discovery overview screen



(b) Our playback screen for a downloaded album

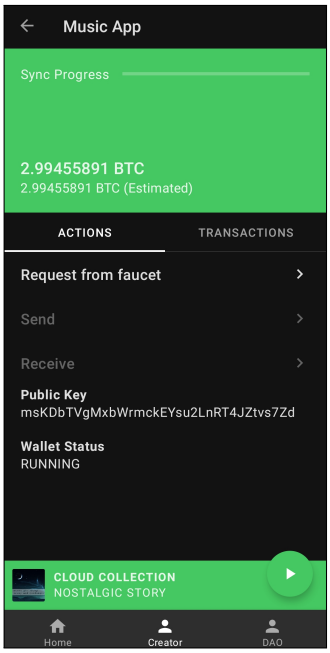


(c) Our DAO discovery overview screen

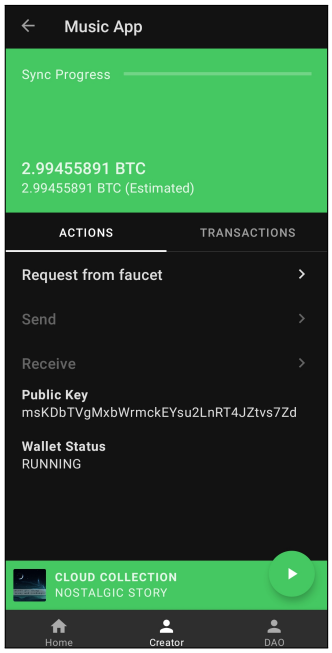


(d) Our DAO voting screen of a proposal

Fig. 5: Screenshots of the Music DAO



(a) The overview screen of the wallet



(b) A list of transaction from the personal wallet

Fig. 6: The integrated Bitcoin lite wallet

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 encompasses multiple DAOs, 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 7c 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 thresh-hold percentage for votes. A transaction is then created and published on Bitcoin and this meta-data is disseminated as described earlier. Other users can view the new DAO on the screen and attempt to join it. The current members must then vote on the proposal. The list of proposals within a DAO can be seen in Figure 7d. If sufficient members 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 into artists. Artists disseminate their Bitcoin addresses using a special artist block. Any member of the DAO can propose to invest into an artist. This is made convenient by showing all the possible artists and accompanying addresses as an option to invest in when making a proposal. 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 in the artist profile screen.

To implement our governance protocol, we have developed a Kotlin implementation of the MuSig [17] scheme. This implementation, the first of its kind in Kotlin, is based on a Python implementation that we partially ported [5]. Notably, we have chosen to avoid native implementations and code bridges. MuSig outputs Schnorr signatures, which are supported by the

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

Bitcoin network since the recent Taproot upgrade. We modify the algorithm to support the specification of Schnorr signatures in Bitcoin described in BIP340 [1], which has a number of cryptographic and encoding caveats we succesfully worked around.

A comparison of various multi-signature protocols is presented in Table I. MuSig2 [19] stands out as a modern multi-signature protocol that requires one less round of communication compared to other options, but unfortunately lacks compatible implementations. FROST [15] is the most modern thresh-hold signature scheme, but is complex and hard to implement. Considering our specific project requirements and constraints, we opted to choose MuSig due to its comparatively complexity and availability of compatible implementations in other languages. However, it is worth noting that by selecting MuSig, our governance capabilities are limited to voting scenarios where unanimity among participants is required for a vote to be successfully passed. This restriction may impose limitations on certain decision-making processes within our system. We consider the usage of a BIP340 compatible thresh-hold scheme future work. This would allow for more flexible voting mechanisms and potentially accommodate scenarios where consensus can be achieved with a predefined threshold of participant approvals.

2) Music Platform: The core features of the 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 like Spotify. We have implemented these functionalities and integrated it into the SuperApp, right alongside our DAO.

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 peer-to-peer transfer of audio files from phone to phones.

Artists publish albums right from their phones. They need to provide a set of audio files, a cover art image and a title. The ID3 metadata in the audio files is used to further enrich the albums with i.e. genre information. An UUID is created for the album so that it can uniquely be identified on the platform. Every album has its own torrent file, however instead of distributing this .torrent file to other users we rely on its info-hash. This is a SHA-1 hash of the contents of the folder. We use this info-hash in combination with the Bittorent DHT to avoid usage of torrent trackers since they are a centralized solution. The BitTorrent DHT is a distributed hash which can be used to find other peers which are interested in torrents with the same info-hash. This info-hash, along with all previously mentioned meta-data, is then serialized into a Trustchain block. This block is then added to the personal Trustchain ledger of the artist and gossiped around the community. Since the blocks are cryptographically linked in a blockchain, artists can prove that they have history in publishing albums and build up reputation.

Figure 7a showcases the list of discovered albums, while Figure 7b showcases the music playback screen. Our UI/UX efforts are focused on simplicity and intuitiveness. We present users with a single list of discovered albums, which can be easily searched using the search bar. These are all albums which album blocks have arrived at the device of the user 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 to the artists from their personal wallets. On the profile pages of artists all their songs can be found. Additionally here users can choose to donate some Bitcoin to the artists from their personal wallet.

To bootstrap the platform for early users, we curate a dataset comprising hundreds of albums with Creative Commons licenses, obtained from PandaCD[6]. 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.

We assume a form of altruistic seeding from users on the platform using a seeding strategy. We do not attempt to solve the problem of selfish seeding and consider this out of scope. Some strategies such as tit-for-tat can be implemented to further incentive users to make available their bandwidth and local storage space. Clients cannot simply seed all their cached music due to limited bandwidth. They must use a seeding strategy to choose what albums to stream, and this strategy can be optimized to increase music availability across the network. This is especially difficult, since music differs in demand greatly. Due to the lack of popularity metrics on our platform, we opt to choose for a random strategy wherein a random set of albums is streamed.

VII. Performance Analysis

In this section, we present an analysis of our implementation's performance. We analyze the governance protocol both in terms of cryptographic performance and its performance in a networked setting. We do 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 performed a real-life deployment test to validate our application involving 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. We 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 public key and a signature of a constant 32-byte string using our BIP340 [1] MuSig implementation. Our

[6]https://pandacd.io/

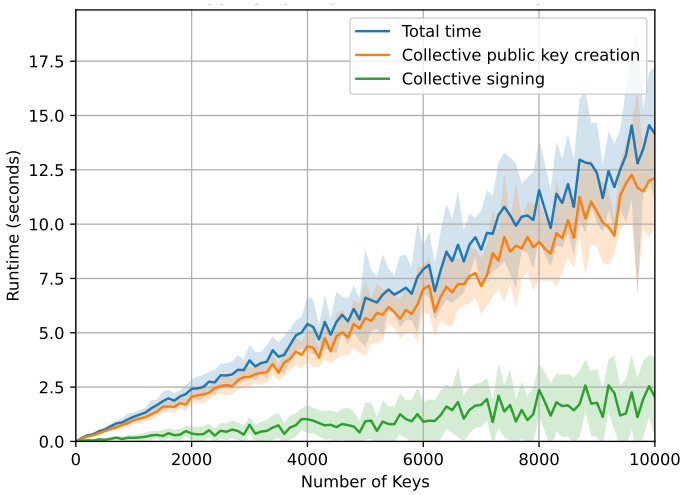


Fig. 7: Cryptographic cost of democratic voting using our governance protocol

goal is to find a best-case runtime for our governance protocol. We thus do not concern ourselves with making this string a Bitcoin transaction. To simulate the performance of our implementation on smartphones, we conducted the experiments using an Android Emulator running on a consumer-grade PC with 32GB RAM and an Intel i7-12700H processor. 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. The experiment runs on the emulator with one process, storing all public keys in memory.. This is not possible in real-world scenarios without compromising security. Before the experiment, all public keys are generated and cached into memory. This is because key generation is an expensive operation, and in practice is done beforehand as well. We run the experiment for up to 10.000 keys with a 100-key interval, to have the experiment run in an acceptable time amount while exploring large key amounts. We run the experiment 10 times due to the non-deterministic nature and performance of key generation.

Figure 9 shows run-time of both aggregating and signing scaling linearly with the amount of nodes. 10.000 keys are aggregated in 12.5 seconds and sign a message in 2.5 seconds. Aggregation of keys takes considerably longer than signing of messages. This can be attributed to the amount of elliptic point multiplications required in aggregation compares to signing [17].

The observed linear increase in runtime for key aggregation and signing as the number of nodes increases indicates that scalability might be a concern when scaling the governance protocol to millions of users. Although 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.

This difference can be unfavorable for new DAOs as opposed to established DAOs. In new DAOs, aggregation of keys is more commonplace due to new members joining, and as such would be more impacted by this. In either scenario, we can conclude that the cryptographic performance is reasonable for most consumer grade hardware. The linear increase in runtime might pose a problem however if we attempt to scale a DAO to millions of users.

We also observe that 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. Public keys in BIP0340 are encoded in such a way that the y-coordinate is always even. If this is not the case, the point is negated. The aggregated public key will be odd in 50% of the cases, which requires all participants to negate their own keys as well. This process causes increases runtime in 50% of the cases.

Figure 10 shows a flame graph of the cryptographic operations of a single aggregation and signing round. The function shown computes a signature for 10.000 keys in 12.5 seconds. We observe that 60% of time is spent is on aggregating the public key. The rest of the time of 25% is mostly used for the negation of keys, which is only required for Bitcoin signatures in theory. The rest of the time 15% is spent on aggregating the nonces, creating the partial signatures and combining these signatures into the final signature.

The results indicate that the aggregation of public keys is the most computationally expensive cryptographic task. This is due to the fact that aggregating public keys requires multiplication of elliptic curve points. The other operations do not require this or only require this in a constant amount of time. Furthermore, we observe that negation of keys is an expensive task. This is because a new key has to be generated for every negation. This is an artifact of the Bitcoin specification of Schnorr signatures, and can be avoided by using other blockchains.

B. Networked Performance

In order to get insight in the viability of this governance protocol in real-world settings, we examine 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. This protocol is hard to evaluate due to its gossiping nature. To simplify the evaluation, we assumed full connectivity between all peers and implemented a simple IPv8 based protocol using UDP messages. 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 IP address. This minimizes latency, since all packets are confined to a local network. The nodes run the aggregation and signing collectively using the

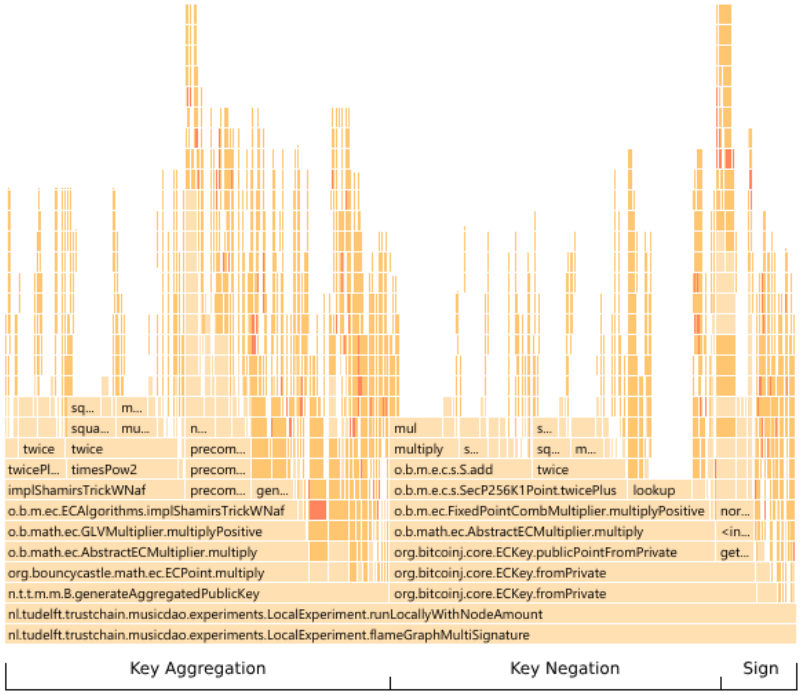


Fig. 8: Flame graph of the cryptographic operations in the governance protocol for 10.000 keys

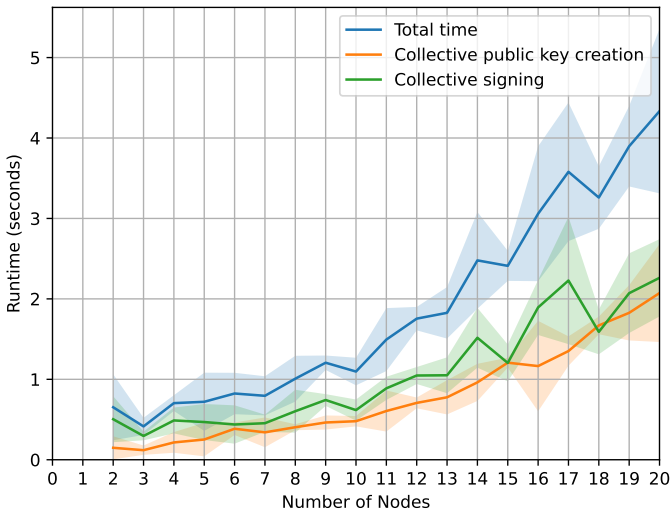


Fig. 9: Performance of democratic voting using governance protocol in networked setting

protocol and a special single node measures and stores the run time. The experiment is repeated for up to 20 nodes for 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 [6], it falls beyond the scope of this experiment.

As shown in Figure 11, 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. A comparison between cryptographic and peer-to-peer performance reveals that the latter is the limiting factor, even under optimal conditions such as event-based communication, local networking, and no Bitcoin transaction creation logic.

Moreover, we note a decreased time gap 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 round communication. We conclude that the governance protocol is bottlenecked solely by networking and not by cryptographic operations.

If voting is required to be time-sensitive, a peer-to-peer governance protocol using P2P is not feasible. We define *time sensitivity* in voting as the requirement for a decision to be made in a very short amount of time, in the range of seconds. An example of this would be investment decisions made based on activity in financial markets, which can fluctuate wildly in seconds. Voting where time is not sensitive can however make use of this protocol. For instance, voting on a decision to fund an album for an artist. This vote can be held open for weeks if needed, and throughout the weeks the votes can be collected and combined. During this period there is enough time for the peer-to-peer protocol to complete.

C. End-To-End Performance

We measure the time it takes for our application to load, for the first packages with music data to arrive, and for music to show up. One phone will act as a seeder and the benchmark phone will receive new releases. The phones are connected to the same local network. The experiment is run 10 times and

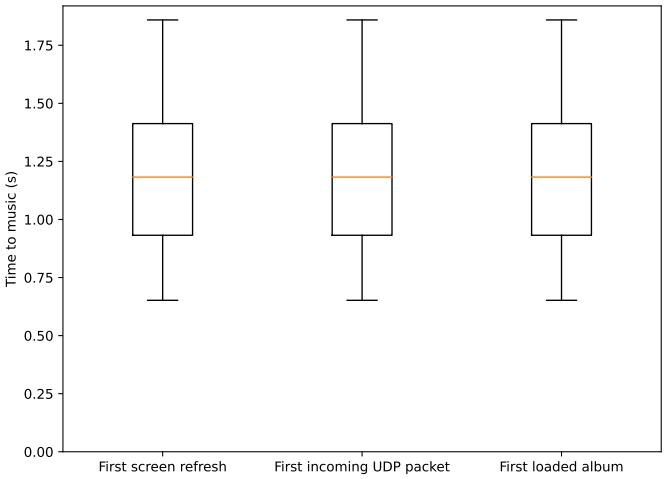


Fig. 10: End-To-End performance of loading albums

Fig. 11: Real-world deployment test

the results can be found in Figure 12. All measurements end up being under two seconds, which we consider a reasonable loading time. Notice that in a setting with more phones, this time will decrease due to more releases being gossiped to the receiver phone. This result can thus be interpreted as an upper bound for the loading performance of our application.

Lastly, in order to evaluate the usability of our application, a real-life deployment test was conducted. A picture can be seen in Figure 13. 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.

VIII. Conclusion

In an increasingly connected world where big tech and governments are centralizing power, decentralized autonomous organizations (DAOs) offer a bottom-up approach to collaboration on the Internet. While DAOs are supposed to be decentralized, many suffer from centralization in both infrastructure and governance. In this work, we have proposed a simple and robust architecture for DAOs that allows for economic activity while maintaining complete decentralization. We show a novel design for governance based on multi-signature schemes which enabled off-chain voting. Using our architecture and design, we create the Music DAO, which utilizes the most robust currently live-deployed networks and demonstrates the viability of our architecture. Our performance analysis on our governance protocol shows that performance is not limited by cryptography operations but by communication overhead.

References

[1] bips/bip-0340.mediawiki at master · bitcoin/bips — github.com. https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki. [Accessed 30-Jun-2022].

[2] Governance - openzeppelin docs — docs.openzeppelin.com. https://docs.openzeppelin.com/contracts/4.x/api/governance. [Accessed 05-Jun-2023].

[3] Multi-signature specification of bitcoin. Accessed: 2023-06-21.

[4] Uniswap combined metrics. Accessed: 2023-06-21.

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

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

[7] 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.

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

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

[10] 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.

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

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

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

[22] 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.

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

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