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 rapdidly become popular in recent years hitting 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 significant challenges related to centralization in both governance and infrastructure. This work addresses these limitations by proposing a novel architecture for a fully decentralized DAO with no compromises. We also devise a novel scalable governance mechanism using multi-signature schemes to manage shared assets. We demonstrate the practicality of our architecture by implementing, deploying and evaluating a music-centered DAO. Our music-centered 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 work represents a significant advancement in the field of decentralized organizations and has the potential to revolutionize the way people collaborate and organize themselves.**

***Index Terms*—Decentralized autonomous organization (DAO) and operation, blockchain, multi-signature scheme, mechanism 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 [13]. 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 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" [9], [11]. 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 [8]. 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 [5] 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 [10]. 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 voting mechanism.

1) **The Simple DAO Architecture** We design and justify an infrastructure 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 voting mechanism 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 crowdfund 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 voting mechanism 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 increasingly less 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). 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 solved by the advent of Bitcoin [17] 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 [22], with the inclusion of decentralization in terms of governance [15]. 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. While 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, and as such, analysis on decentralization in current DAOs and possible theoretical frameworks on how decentralization can actually be achieved remain scarce. 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 theorertical 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 [12]. Hee 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”. The goal of the project was to create a new business model for non-profit enterprises. With an internal capital of 150 million USD from 11.000 investors at its peak, it was extremely large for its time. It however suffered from an exploit in the smart contract [3], after which the Ethereum blockchain was forked to return the money to investors.

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

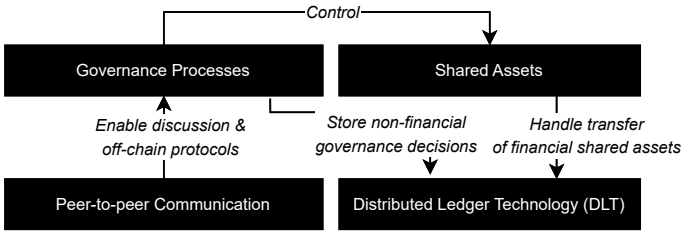

Fig. 1. The Simple DAO Architecture

ways to formalize, design and implement DAOs in an academic manner. There is however a lack of technical papers regarding the technological infrastructure of DAOs. Shuai et al. developed a comprehensive framework for DAOs that identifies their characteristics, problems, implementations, and upcoming trends [20]. They suggest a five-layer architecture for DAOs separating governance, technology and incentives. 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 definition of the DAO.

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 [6]. 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. Further empirical work on analyzing DAOs has been done by Bellavitis et al., who show a continuous increase in terms of DAOs, active users and proposals across the eco-system [8].

IV. The Simple DAO Architecture

We now present our simple DAO architecture, visualized in Figure 1. We deliberately remove all unnecessary features and complexity in order to provide a flexible and strong building block. Our architecture represents a milestone within the evolution of actual DAO realisations: it is the first to achieve complete decentralisation. We elaborate on our requirements and components in our architecture.

A. Architectural Requirements

Our architectural requirements are based on the principle of decentralization and zero-server architecture [19]. This architecture provides design principles for common infrastructure, which include having no hierarchy in networks, no intermediaries and democratic decision-making processes. We identify a set of principles which are required for all components in order for DAOs to function in a decentralized manner in an adversarial environment.

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.

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.

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. Distributed Ledger Technology

Distributed Ledger Technology is required to provide participants with a mechanism to interact with each other directly without having to trust each other. Typically this is solved through a blockchain, which is a distributed digital ledger of trust which records transactions in a transparent, secure and immutable manner. This makes financial decisions possible, since the problem *double spending* is solved in practice.

We deem that a DLT must be open-source, permission-less, transparent and sufficiently decentralized in order for it to adhere to our architectual 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. Transparency of transactions of blockchains allows for verification of processes and allows members to hold each other accountable. The notion of sufficient decentralization can be measured in terms of the difficulty to attack the network, the age of the network and a number of other measures [15]. Without this decentralization, components such as governance run the risk of becoming centralized again.

C. 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. 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.

D. 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 for 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. History of communication must be public. This ensures new participants can review the history of the DAO, thereby enabling them to make informed decisions that align with the objectives of the organization.

Peer-to-peer overlay networks facilitate communication in the way we described. 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. This is essential for fostering discussion about proposals for decisions and the collaboration within the organization. The creation and dissemination of proposals for instance must be communicated among all members. They also 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.

Storage of meta-data is done through the principle of *local first data storage*. Local first data storage entails that network participants themselves are collectively responsible for the availability of data. There should not be 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, for instance gossiping protocols, data can be replicated to ensure data availability.

E. Governance Processes

Governance processes make economic activity possible by enabling participants to collectively make decisions. Any member must be able to make a proposals and vote for a decision, to ensure the system is permission-less .The decision must be executed automatically, in a trustless manner. The primary use case is for the management of the shared assets. Members can collectively decide to invest in something, with the aim of furthering the objective of the DAO.

DLTs are the only way to make this possible and satisfy our requirements. 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 where-in every action is a separate transaction made on the ledger. We will describe an alternative way to do this using our voting protocol described in Section V.

We do not deem off-chain governance real governance. This is governance which is not stored on-chain, but relies on the counting of signatures posted on some bulletin board on platforms such as Snapshot [CITE]. 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 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 [20]. We do however argue that in the most ideal case a one-vote-one-human 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.

V. Multi-Signature Voting Protocol

We now introduce a novel protocol which combines multi-signature and blockchain technology in order to make governance possible. We visualize our protocol in Figure 2. We first describe current governance protocols, then describe our protocol and its advantages over them. We then show how to use it in the context of managing shared assets on a blockchain. In Section VI we implement this protocol for managing shared assets.

The most common voting mechanism in use currently is through the usage of smart-contracts. The industry-standard for such contracts is a smart contract by OpenZeppelin named Governor [4]. The smart-contract houses all the state for the DAO: the proposals, the vote count on these proposals and other rules. Participants use their wallets and tokens to interact with the contract. They can cast votes on a proposal by creating and publishing a transaction which changes the state of the contract according to a set of rules. Once the voting period ends, the proposal is closed and the result is considered final.

The main advantage of this approach is its extendability. Custom smart contracts can support advanced functionalities

TABLE I

COMPARISON OF SIZE REQUIRED ON BLOCKCHAINS FOR DIFFERENT VOTING MECHANISMS. SIZE IS MEASURED IN TERMS OF KEY SIZE. n IS THE NUMBER OF PARTICIPANTS IN THE DAO. m WHERE APPLICABLE IS THE NUMBER OF PARTICIPANTS REQUIRED TO VOTE.

Voting Scheme	Year	Signatures Required	Public Keys Published	Signatures Published	Transactions Required	Size
Smart Contract [4]	2013	$\leq \lvert n \rvert$	$\leq \lvert n \rvert$	$\leq \lvert n \rvert$	$\leq \lvert n \rvert$	$\lvert n \rvert$
Naive Bitcoin [1]	2008	$\lvert n \rvert$	$\lvert n \rvert$	$\lvert n \rvert$	1	$\lvert n \rvert$
MuSig [16]	2018	$\lvert n \rvert$	1	1	1	1
MuSig2 [18]	2020	$\lvert n \rvert$	1	1	1	1
FROST [14]	2020	$\leq \lvert n \rvert$	1	1	1	1

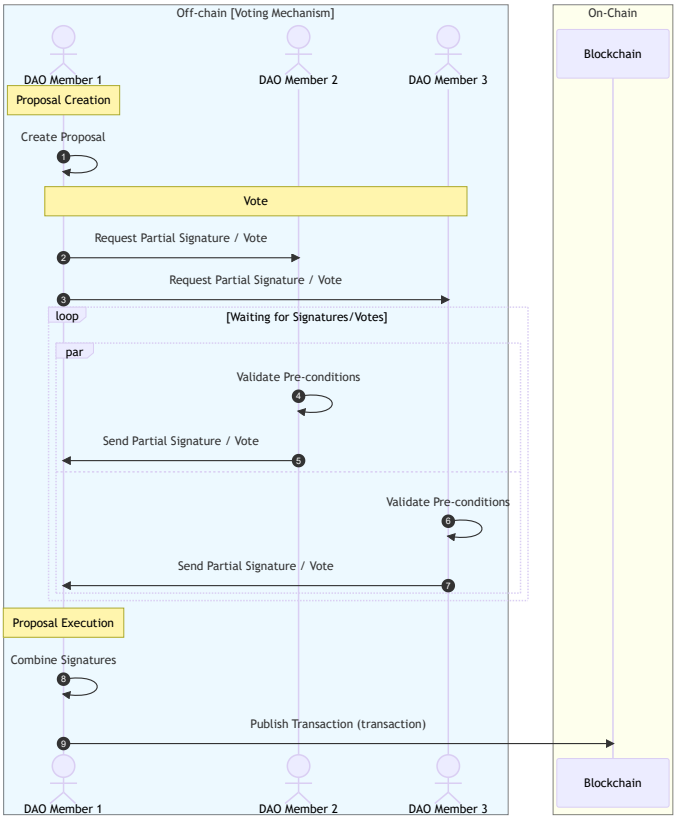

Fig. 2. Sequence diagram of our voting protocol for a transaction proposal

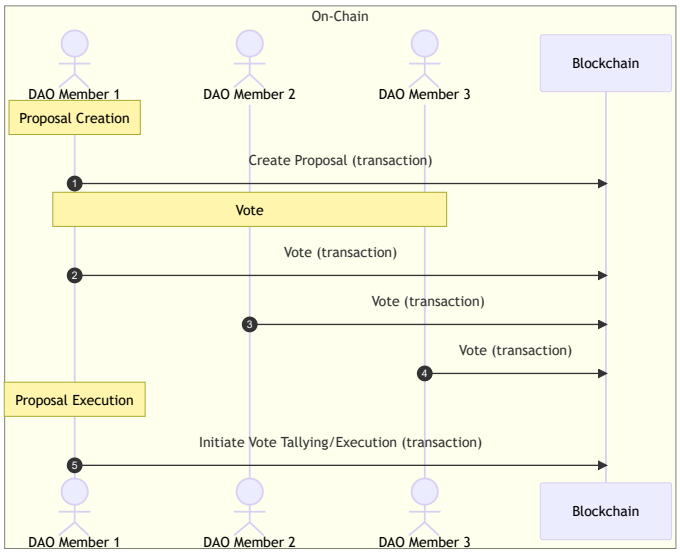

Fig. 3. Sequence diagram of voting protocol in a typical DAO deployed on Ethereum using smart-contracts

such as delegating votes, automatically transacting funds after a successful proposal and adding additional requirements for initiating proposals. The main downside is that in order to complete a vote many transactions are needed, which is hindered by the scalability problem of blockchains. The process for a vote without delegation is visualized in Figure 3.

A. Protocol Specification

We propose a voting mechanism based on multi-party computation (MPC), specifically, multi-signature and thresh-hold signature schemes [14], [16], [18]. These are schemes in which a set of participants jointly have ownership over a single public key. The creation of this shared public key is also done in a secure manner through key aggregation. In order to create a signature, each participant creates a partial signature. These partial signatures are then combined into a single signature. At no point during key creation, aggregation or signing does the private key exist. Thresh-hold signature schemes allow for a thresh-hold of partial signatures to be sufficient for signing, multi-signature schemes require all partial signatures.

They key idea behind our protocol is equating the creation of a partial signature as casting a vote in favor of a proposal. Using this, we can vote off-chain and merely have to store the result of the vote on-chain. We visualize our protocol and the exchange of messages between participants and the blockchain in Figure 2. Communication is done off-chain through an overlay network as described in Section IV. All participants have a public key known to all other participants through which they are identified.

The voting protocol works as follows. 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 enough partial signatures are available, the vote is over and the proposal has been accepted by virtue of the creation of the signature. In the case of a transaction, this transaction can now be published and accepted on a

blockchain. Note that a time limit is not possible and votes cannot be revocated. A *pre-condition* can be defined by the participants. This is a function in an arbitrary programming language which verifies a condition, such as the state of the blockchain at that moment. Note that this *pre-condition* is not secured through additional cryptographic means: if the enough people want to collude and ignore the *pre-condition*, they can do so and still create a partial signature.

In this design, we do not rely on advanced turing-complete smart contract capabilities. Instead, we use a blockchain of choice, namely Bitcoin, which is simple and secure, and does not require advanced smart contract capabilities. In this way,we can achieve a high level of security and scalability, while keeping the complexity of the system at a minimum.

In addition to this, we greatly reduce the number of on-chain transactions needed by up to n, n being the amount of members in the DAO, compared to governance processes using smart-contracts. This is since only the result of the vote of needs to be published on-chain and voting itself can be done off-chain through an overlay network. This lets us avoid large transaction fees and long transaction times.

In Table I we compare a number of multi-signature and thresh-hold schemes to smart contract governance in terms of the size of the transaction and the amount of transactions required. We have included the most recent and widely used schemes which support Schnorr signatures, which is a necessity for compatibility with Bitcoin. We also include Naive Multisignature in Bitcoin in our comparison. This multi-signature scheme relies on including multiple signature and public keys in the transaction, which defeats the purpose of avoiding on-chain storage usage.

B. Managing Shared Assets

Using our protocol we can enable participants to own and manage shared assets. For this, we need to carefully design our pre-condition and leverage the validation rules of the blockchain. We will be using the Bitcoin blockchain and a thresh-hold signature scheme. The shared assets wil be in the form of Bitcoin. Collective funds are locked up by a single collective public key. A set of transactions which outputs are locked up by this key represent the collective fund.

Locking up funds - Anyone can send funds to the DAO by publishing a signed transaction. The transaction should contain inputs from the sender's personal wallet with the outputs locking up the funds with the DAO's current public key. Subsequently, these funds can now be spent by the members of the DAO.

Spending funds - To spend locked up funds, members must publish a transaction sending previously locked up funds to a new address. A single member creates an unsigned transaction and sends this to the other participants as a proposal. The input of this transaction consist of previously locked up funds. The output is the receiving address for some amount of currency. The other members then run a voting procedure on this proposal. If the vote is successful, the transaction can be signed and published after which it is executed and stored on the blockchain.

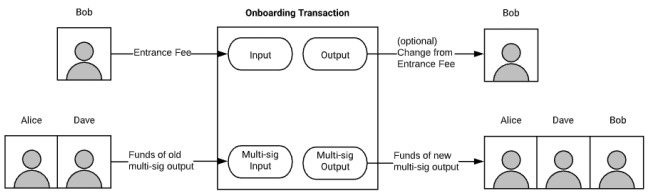

Fig. 4. Visualisation of transaction which can extend the DAO

Member inclusion and exclusion - In order for a new participant to join, all funds must be moved from the current collective key to a new collective key which includes the new participant. Typically, the new participant must also pay some pre-agreed upon entrance fee to join the organization. The new participant creates 1) a new collective public key which includes his own and 2) a new unsigned transaction. This transaction is visualized in Figure 4. The transaction has two inputs. The first input is the entrance-fee, signed by a personal wallet of the new participant. 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 participant. The transaction is set-up in this way to make it impossible for a new participant to join without paying the entrance fee, since it is done in an atomic manner.

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 participants first validate whether the participant actually has signed its entrance fee input. If the vote is successful, the participant will have paid the entrance-fee and joined the DAO through key inclusion.

The removal of a participant is done in a similar manner, through the exclusion of the key. Any member can start the procedure of removing someone by creating a new key excluding the members and moving the funds to the new key. If enough members vote, the member will be removed and not be able to vote in the future. Members can in this transaction also decide to return funds back to the leaving members.

Note that the first participant can set-up parameters such as the thresh-hold of the DAO. It is subject to voting procedures whether this thresh-hold is ever changed in the future.

Governance Structure - The implicit governance structure exhibited here is founded on the ownership of private key shares. A one-token-one-vote [21] 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.

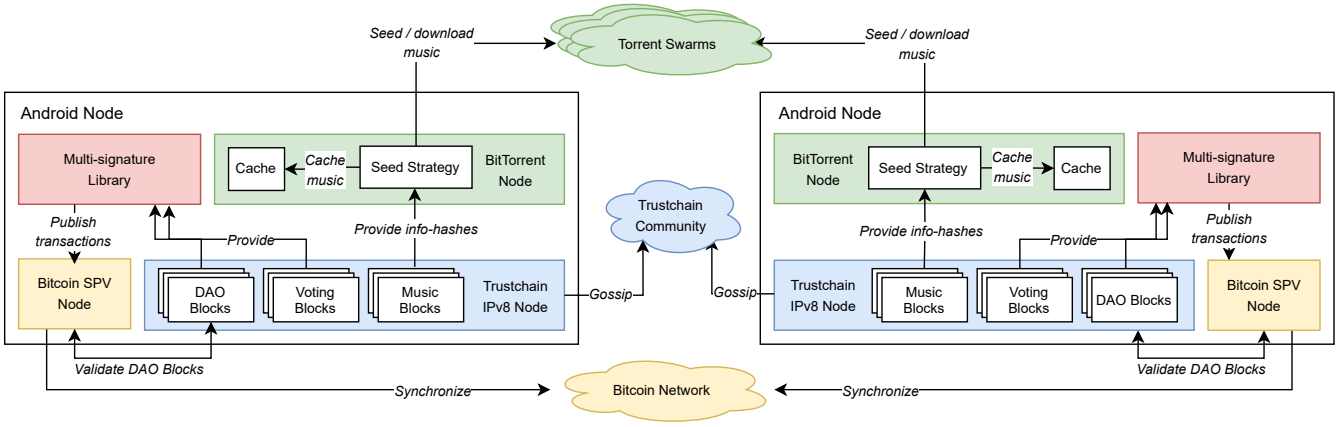

Fig. 5. A visual representation of the Music DAO based on our architecture.

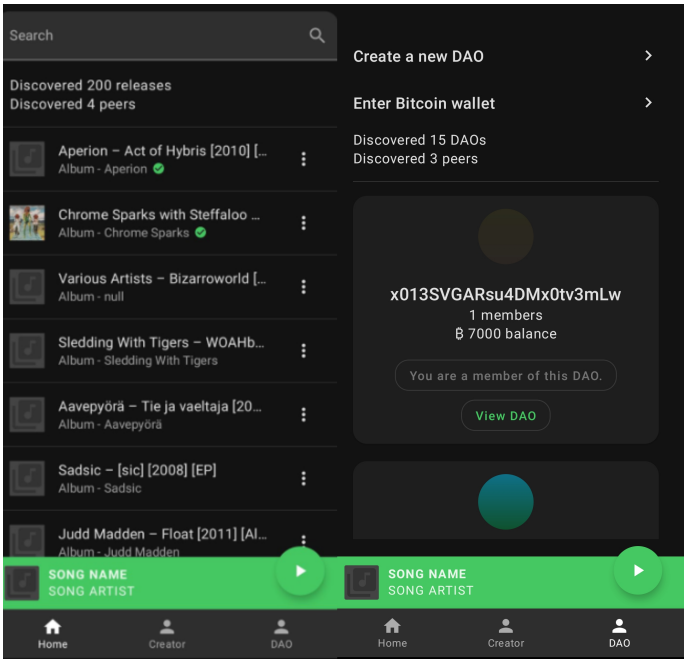

Fig. 6. A list of releases and DAOs in the application

VI. Music DAO: A truly decentralised DAO

We have created MusicDAO to re-shape the music industry. We meticulously designed MusicDAO 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 [CITE]. This allows them to completely focus and 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 mechanism described in Section V. Any listener can start a new fund which other listeners can join. Together they can make proposals to fund the projects of their favorite artists.

Our usage of open-source technologies and permissions-less networks keeps users fully in control of their music and funds. Streaming platforms have vendor lock-in, making it hard for artists to move their music to other services. This practice is even more of a problem with record labels, which makes upcoming artists sign away their music rights for the rest of their life. 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.

There is no open API or protocol for artists to share their music across all platforms. Artists have no control on how their music is consumed, with many platforms being riddled by advertisements. They cannot instead offer their listeners alternative open-source software, unlike our solution. Even if an artists decides to use multiple platforms, they must agree to all their terms and conditions, which are subject to change and unfavorable. Furthermore, censorship by moderation teams on platforms is now also impossible, which is an important feature for artists living in jurisdictions in the world imposing censorship.

In order to realize this objective, the DAO consists of two main components: the music platform, and the crowdfund platform. The music platform enables the dissemination and availability of music and it's meta-data. The crowdfund platform allows listeners to collectively manage funds, which they

can use to fund new projects of their favorite artists.

A. Implementation and Deployment

We now present our implementation, visualized in Figure 5. Our implementation is completely decentralized and based on our Simple DAO Architecture described in Section IV. For all components, we ensure that our architectural requirements for decentralisation are satisfied. Each user runs our open-source software on their Android smartphone. This software is available on Github and deployed on the Google Play store. We now elaborate on the components in our implementation.

The implementation is created using Kotlin and Android on the JVM platform. This allows for deployment on the Play Store and accessibility for hundreds of users. Cross-platform mobile application is outside the scope of our use case, due to many of our libraries not being available, such as our chosen overlay network IPv8. Android additionally provides extensive service APIs that allow services to continuously run in the background, allowing for the upkeep of the network.

We chose to limit our implementation to smartphones only for several reasons, all of which align with our primitive of creating a permissionless system. Additionally, smartphones have a lower barrier to entry, as almost everyone has a phone, especially in developing countries, and not everyone has a PC. The zero-architecture server stack also supports the idea that smartphones are the superior device for maintaining and using P2P networks.

BitTorrent Node - The use of BitTorrent in our implementation is due to its reliability and decentralization. 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. By extracting torrent info hashes from the platform, we can facilitate mass seeding of the network, or allow users to download content using popular torrent clients without the need for our application. The use of the accompanying Distributed Hash Table (DHT) network in our implementation is to remove the need for tracker servers, which are centralized and may be taken down by law enforcement agencies. DHT networks are much harder to take down and only require a simple bootstrap node, which can be any node with sufficient knowledge, after which you can get almost any swarm info about a info-hash in the network.

Listeners keep seeding a part of their music according to some strategy, for instance based on popularity. The optimization of this process is out of scope for this work. For this implementation, the most popular music and a selection of the less popular music (tail-end) is randomly selected and seeded.

Different users on the network can receive the signed trustchain blocks and add them to their local storage of published music. They use the meta-data in the block to query the DHT network and download peer information to download the torrent from seeders. After the music has been downloaded, everything is verified, and the listener can listen to the music with the accompanying data.

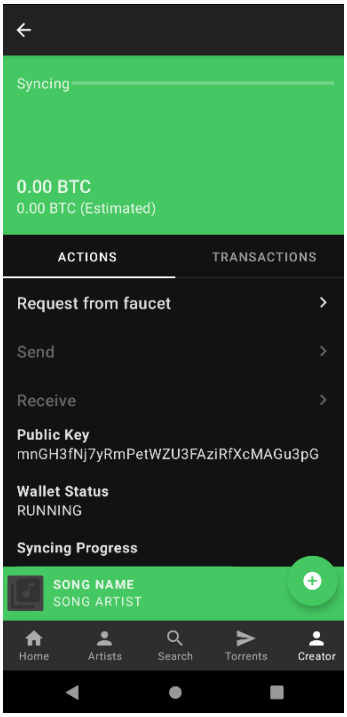

Fig. 7. The integrated Bitcoin lite wallet

IPv8 & TrustChain Node - Artists can publish music to the platform. Published music is shared on the IPv8 peer-to-peer overlay network. The music is first encoded to the correct format and an accompanying torrent file/torrent meta-data is created for the formatted data. This meta-data is then published on the personal trustchain of the user and gossiped around to other users. At the same time, the torrent file is published on the BitTorrent DHT network and is available to seed from the phone. Additional meta-data such as album art cover is also included in the published music and is displayed in the GUI.

Bitcoin SPV Node -

Multi-Signature Library -

VII. Performance Analysis

In this section, we present an analysis of our implementation's performance. We analyze the voting mechanism both in terms of cryptographic performance and its performance in a peer to peer setting with networking. To evaluate usability, we perform various experiments measuring the time to discovery and listening of music, and discovering DAOs. Additionally, we conduct a real-life deployment test involving experts in the field of DAOs, who actively engaged with our implementation.

We measure the performance of our voting mechanism described in Section V both in terms of its cryptographic performance and its performance in a peer-to-peer setting. We explore whether our mechanism 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 [2] MuSig implementation. Our goal

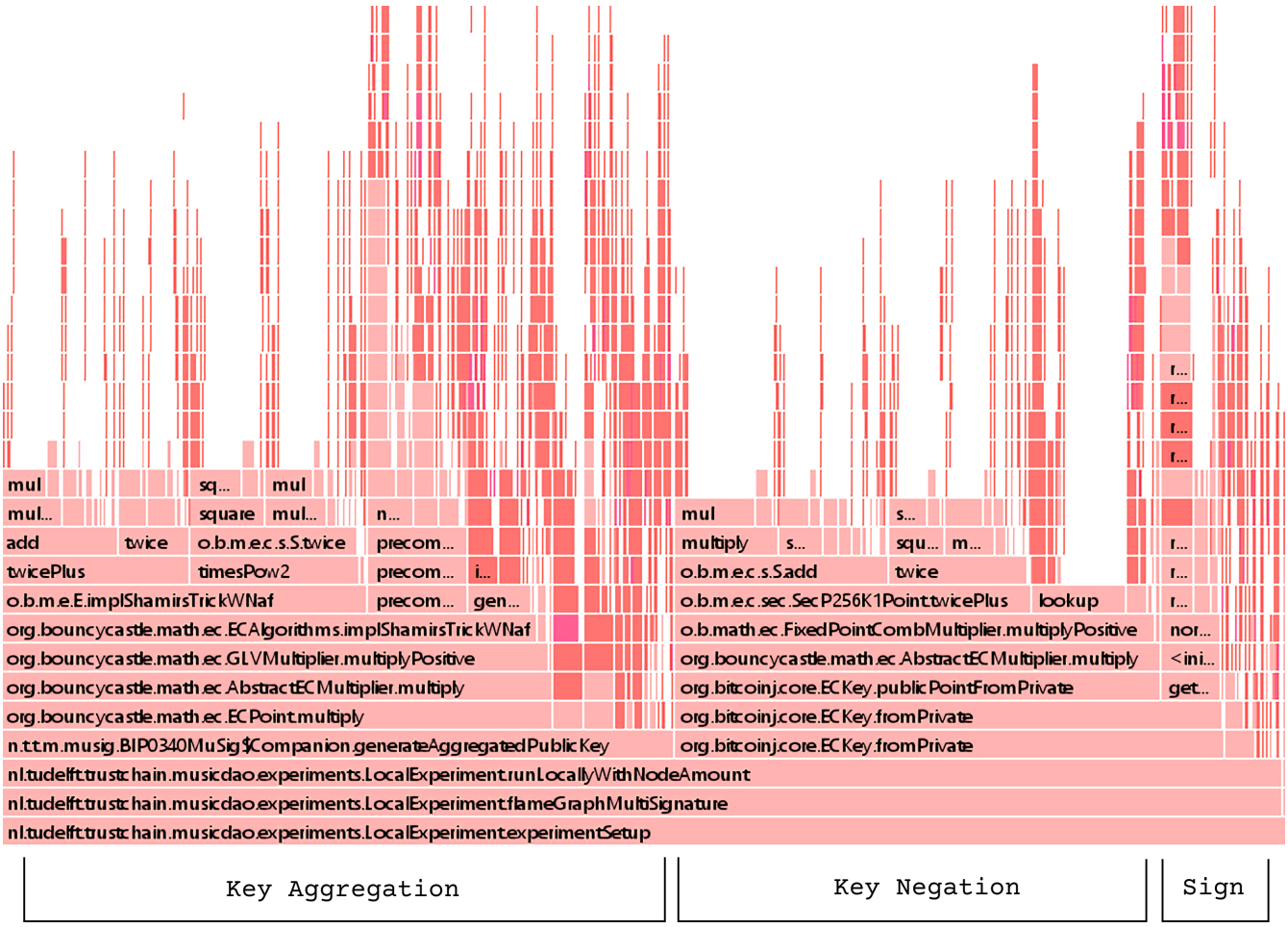

Fig. 8. Flame graph of the cryptographic operations in the voting mechanism for 10.000 keys

is to find a best-case runtime for out voting mechanism. We thus do not concern ourselves with making this string a Bitcoin transaction. We run the experiments on an Android Emulator within an consumer grade PC with 32GB RAM and a Intel i7-12700H, mimicking the conditions of our implementation being run on smartphones only.

A. Cryptographic Performance

Firstly, we measure cryptographic performance in order to get insight into a best-case runtime. The experiment runs in a single process on the emulator in a sequential manner: all public keys of all participants are stored in memory and accessible immediately. This is not possible in real-world scenarios without compromising security. Before the experiment, all public keys are generated and cached in memory. This is because key generation is an expensive operation, and in practice is done before-hand as well. We run the experiment for up to 10.000 keys with a 100 key interval, with the aim of having the experiment run in an acceptable time amount while exploring large key amounts. The experiment itself is repeated 10 times, since key generation is non-deterministic and can influence the performance.

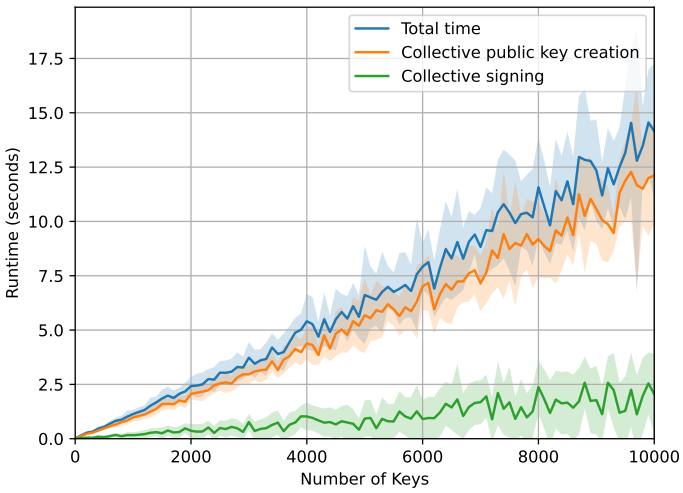

Fig. 9. Cryptographic cost of democratic voting using multi-signature aggregation scheme MuSig

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 [16] compares to signing.

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 impacted by this. In either scenario, we can conclude that the cryptographic performance are reasonable for a large number of users on consumer grade hardware. The linear increase in run-time 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.

Peer to Peer Performance In order to get insight in the viability of this voting mechanism 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. Instead we add the assumption that every user is online and can immediately reply to incoming messages. We implement and evaluate a simple IPv8 based protocol using solely UDP messages and use it on our experiment, assuming full connectivity between all peers. This eliminates the risk of us simply measuring the performance of the gossiping protocol, which optimization is out of scope.

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 protocol and a special single node measure and stores the runtime.

The experiment is repeated for up to 20 nodes for a total of 10 times. The node amount limit is due to certain messages scaling with the amount of nodes, eventually exceeding the UDP packet size limit. Although this limitation could be addressed by using protocols such as the EVA protocol [7], it falls beyond the scope of this experiment.

As shown in Figure 10, 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.

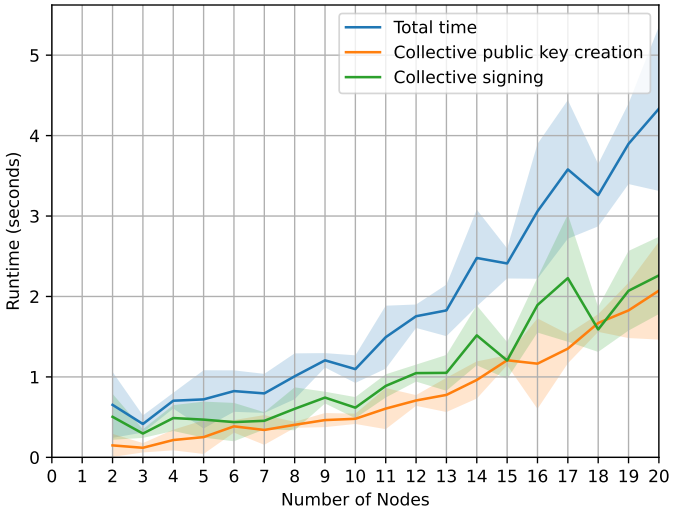

Fig. 10. Performance of democratic voting using multi-signature aggregation scheme MuSig a peer-to-peer setting with networking

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 voting mechanism is bottlenecked solely by networking and not by cryptographic operations.

If voting is required to be time sensitive, a peer to peer voting mechanism 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 on investing 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 mechanism. 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. In this period there is enough time for the peer to peer protocol to complete.

1) Flamegraph of Local MuSig: Figure 8 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. From the Figure we can show 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 until 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. Usability Experiment

We measure the time it takes for music to show up in the application using two phones using benchmark code within the application. One phone will act as a seeder and one phone will receive new releases. The phones are connected to the same local network. The experiment is run 10 times and the results can be found in Figure 11. All measurements end up being under two seconds, which is a reasonable time to wait. Notice that in a setting with more phones, this time will decrease due to more chance of releases being gossiped to the receiver phone. This thus can be interpreted as an upper bound.

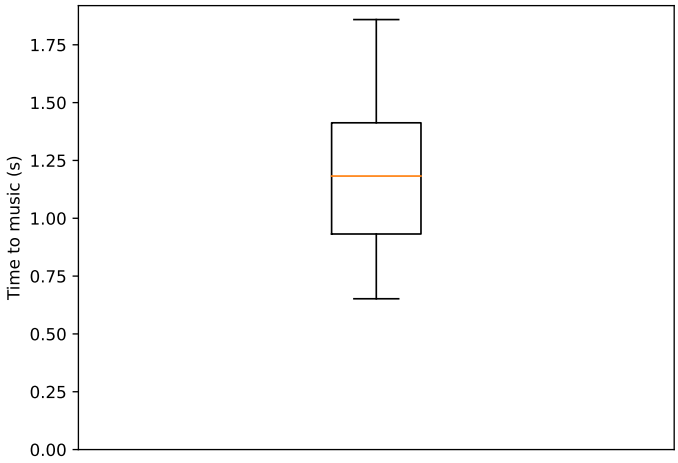

Fig. 11. Time to first discovery music

C. Real-life deployment test

In order to evaluate the usability of our tests, a real-life deployment test was conducted. 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 gained 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 for collaboration on the internet. However, many DAOs suffer from issues caused by managerial and infrastructure centralization. In this work, we have proposed a simple and robust architecture for DAOs that allows for economic activity while maintaining complete decentralization. The Music DAO, which utilizes the most robust currently live-deployed networks, demonstrates the viability of this architecture, and our evaluations show that it is both scalable and user-friendly.

References

[1]
[2] bips/bip-0340.mediawiki at master · bitcoin/bips — github.com. https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki. [Accessed 30-Jun-2022].
[3] A call for a temporary moratorium on the dao.
[4] Governance - openzeppelin docs — docs.openzeppelin.com. https://docs.openzeppelin.com/contracts/4.x/api/governance. [Accessed 05-Jun-2023].
[5] Uniswap combined metrics.
[6] Henrik Axelsen, Johannes Rude Jensen, and Omri Ross. When is a dao decentralized? *arXiv preprint arXiv:2304.08160*, 2023.
[7] Joost Bambacht and Johan Pouwelse. Web3: A decentralized societal infrastructure for identity, trust, money, and data. *arXiv preprint arXiv:2203.00398*, 2022.
[8] 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.
[9] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.
[10] Lin William Cong, Zhiguo He, and Jiasun Li. Decentralized mining in centralized pools. *The Review of Financial Studies*, 34(3):1191–1235, 2021.
[11] 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.
[12] Ethereum Foundation. Daos, dacs, das and more: An incomplete terminology guide.
[13] Ralph Hertwig. Tapping into the wisdom of the crowd—with confidence. *Science*, 336(6079):303–304, 2012.
[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] 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.
[20] 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.
[21] E Glen Weyl, Puja Ohlhaver, and Vitalik Buterin. Decentralized society: Finding web3's soul. *Available at SSRN 4105763*, 2022.
[22] Qiheng Zhou, Huawei Huang, Zibin Zheng, and Jing Bian. Solutions to Scalability of Blockchain: A Survey. *IEEE Access*, 8:16440–16455, 2020.