Audius
A Decentralized Protocol for Audio Content

Roneil Rumburg Sid Sethi Hareesh Nagaraj Chirath Neranjena

Version: 1.0
Last Update: September 21, 2018*

Abstract

The music industry generates $43 billion in revenue but only 12% goes to content creators. Furthermore, creators have minimal control over how their music is distributed and little visibility into who is listening to it. To address these and other problems faced by creators, we introduce Audius, a fully decentralized music streaming protocol built with public blockchain infrastructure and other decentralized technologies. Audius allows creators to distribute to and get paid directly from their fans, and is comprised of the following components:

1. An efficient dual token economy that incentivizes active participation from all stakeholders
2. A decentralized storage solution for sharing audio and metadata
3. A unique track encryption scheme paired with a payment mechanism to unlock user-specific proxy re-encryption keys for content
4. A discovery protocol for users to efficiently query metadata
5. A community arbitration protocol to fairly and efficiently resolve disputes filed by protocol participants
6. A decentralized governance protocol, whereby artists, service providers, and listeners are individually and collectively enfranchised in decision making about protocol changes and upgrades

We also discuss a path to building this protocol over time, starting with a functional subset of these components and working with the community towards a complete implementation.

*Audius is a work in progress and the contents of this paper are subject to change. The most current version can be found at https://whitepaper.audius.co. For feedback and comments, please contact whitepaper@audius.co.

Contents

List of Figures

List of Tables

1 Introduction

Music creation and distribution have been dramatically changed by technology in the last two decades. Creating music no longer requires a team of producers and audio engineers; anyone in their bedroom can start with inexpensive software. Similarly, distributing music no longer requires factories that produce physical records and retail relationships for getting those records into stores; music platforms have enabled creators to distribute their own music.

Despite these changes in how artists create and disseminate music, the mechanics of value transfer and accrual are still largely obfuscated [1]. In 2017, the music industry generated $43 billion in revenue but only 12% of that made its way to artists [2]. As points of comparison, NFL players capture at least 47% of the revenue generated by the entire NFL [3], and NBA players capture between 49 and 51% [4]. Centralized user-generated music distribution platforms have succumbed to the influence of legacy institutions, struggling to find sustainable business models [5, 6] as existing institutions reap the rewards of their (and artists') labor.

1.1 Current problems

We see a number of specific challenges faced by creators and listeners today:

1. There is little to no transparency around the origins of creator payouts (e.g. number of plays, location, original gross payment before fees)
2. Incomplete rights ownership data often prevents content creators from getting paid; instead, earnings accumulate in digital service providers (DSPs) and rights societies
3. There are layers of middlemen and significant time delay involved in payments to creators
4. Publishing rights are complicated and opaque, with no incentives for the industry to make rights data public and accurate
5. Remixes, covers, and other derivative content are largely censored due to rights management issues
6. Licensing issues prevent DSPs and content from being accessible worldwide

1.2 The Audius project

We propose the Audius project as a solution to these problems. The mission of the Audius project is to create a fully decentralized community of artists, service providers, and listeners collaborating to share and defend the world's music. The Audius project will build a decentralized audio protocol guided by the foundational beliefs that:

1. Protocol participants should be compensated in proportion to how much value they create
2. Governance power should be earned by creating value in Audius, and shared consistently between user groups contributing to the protocol
3. Prices and earnings for participants should be consistent, predictable, and transparent
4. Access should be democratized; anyone can contribute to Audius if they follow the protocol rules, and all information is publicly accessible
5. Intermediaries should be removed when possible; when necessary, they should be algorithmic, transparent, and verifiably accurate

The Audius protocol allows creators, listeners, and service providers to collectively provide a high-quality end-user music streaming experience without centralized infrastructure. The protocol is comprised of the following 5 components working in conjunction:

1. **Audius and Loud tokens**: common systems for value accrual and value transfer between stakeholders (Section 2)
2. **AudSP**: A decentralized storage protocol built atop existing decentralized storage projects, for creators to share content through the protocol, listeners to share content they have cached, and content services to monetize bandwidth by serving files through the protocol (Section 3)
3. **Track upload and management**: A protocol for creators to share and manage their content (Section 4)
4. **Payments and revenue sharing**: A protocol for listeners to stream content, dividing payment among rightsholders, service providers, and content curators (who can earn a share of revenue generated by their reposts and playlists) (Section 5)
5. **Discovery**: A protocol for a network of discovery services that index the Audius blockchain and are paid by other network participants to query this dataset (Section 6)

In addition to the above components, there are two meta-protocols that govern Audius:

1. **Arbitration**: A protocol for a decentralized community of arbitrators who are paid to resolve disputes arising in other areas (Section 7)
2. **Governance**: A protocol for modifications and improvements to Audius, which shares governance power among those who have created and are creating value in the Audius protocol (Section 8)

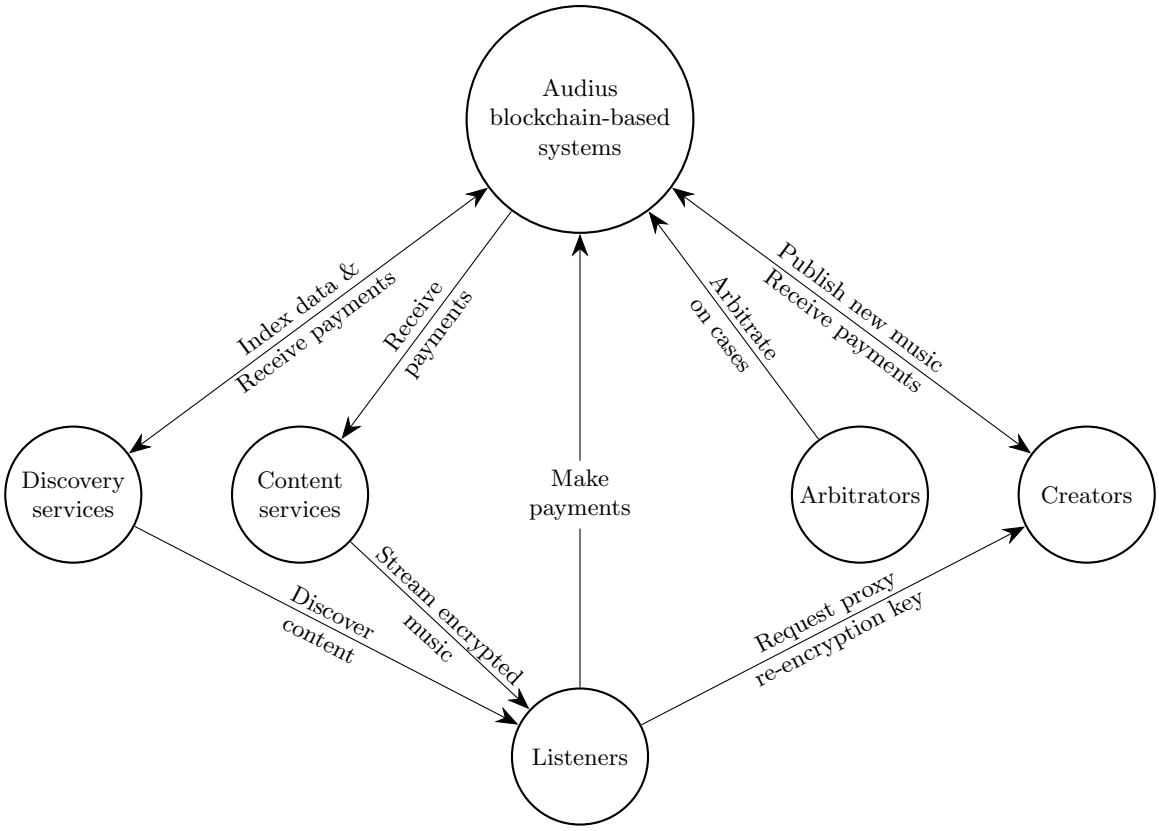

Figure 1: Audius protocol overview

The protocol will also require end-user facing clients—in the context of this paper, clients are software through which users interact with the Audius protocol. Audius will produce open-source reference client implementations, including an end-user listener client, creator client, and service provider implementations, but any developer may produce and distribute their own clients too.

The Audius blockchain, referred to as such throughout the paper, is the collection of smart contracts and blockchain-based code that is used for decentralized coordination within the Audius protocol. We do not currently intend to build our own blockchain, but instead plan to build atop existing blockchain-based platforms. Different parts of the Audius protocol could run on different blockchain-based platforms, or utilize off-chain scalability solutions, where scalability trilemma tradeoffs [7] can be made on a module and subprotocol-specific basis.

2 Audius and Loud tokens

The Audius protocol will be used by a number of different stakeholders with different goals. In order for these diffuse stakeholders to effectively work together toward common network goals, there needs to be a unified incentive structure that aligns the interests of the individual with the interests of the protocol.

The Audius protocol includes two different tokens: the protocol token (Loud), a price-stable medium of exchange used by creators and listeners to interact with the protocol, and the governance token (Audius), used by service providers to participate in staking protocols and earn proceeds from the minting of Loud tokens. This separates the mechanism for price-stable value transfer (Loud) from the mechanism for value capture and accrual (Audius), better serving the needs of users of each token.

Based on the mission and philosophies of Audius, token transfers should be:

- Trustless
- Transparent
- Uncensorable
- Fast
- Inexpensive
- Direct between users, when possible

We also plan to launch a few classes of non-

Table 1: Token types

Token type	Description	Used by	Used for
Loud	Price-stable medium of exchange (value transfer)	Creators, listeners, service providers	Track and service payments
Audius	Value capture and accrual	Service providers	Participate in staking dividends to provide services and earn minting proceeds

fungible tokens in future, representing creators, album/ compilation ownership, and limited-edition creator experiences. These are explained further in Section 10.

2.1 Loud (ticker LOUD): Token for price stable value transfer

Loud tokens are used by listeners and creators to pay for services and content, and earned by content creators and service providers. Loud tokens should be price-stable, ensuring that creators, listeners and service providers can participate in the Audius economy without concern for price volatility.

Loud tokens will be divisible and freely transferable. Relative price stability is maintained through a burn-and-mint mechanic:

- A blockchain-based system mints and sells new Loud tokens at a fixed price, creating a price ceiling
- Loud is burned and bonded over time, taking tokens out of circulation
- The minting system will maintain a reserve to buy back and burn tokens if the price declines

2.1.1 Mint

A blockchain-based system will mint and sell new Loud tokens at a fixed real-world price, denoted F_M. New tokens can be bought with intermediate cryptocurrencies such as Ethereum, with the intermediary's price being determined by a price oracle. Minting will have a rate limit, which is adjustable by the Audius governance protocol (Section 8). Most mint proceeds will be distributed to Audius token stakers on a recurring basis, but a percentage of the proceeds will be held in a reserve by the minting system to assist with price stability. If the Loud token price declines, the minting system will spend its reserve to buy back and destroy Loud to create positive price pressure.

2.1.2 Burn

A percentage, P_B, of every Loud payment is burned when listeners consume network services. This includes payments made (1) to creators to unlock track segments, (2) for track content retrieval, and (3) to request discovery results. P_B is tuned algorithmically, and correlated inversely with the Loud market price. This process can be modified by the governance protocol (Section 8). Users also permanently bond Loud tokens with each track uploaded or social action taken (detailed in Section 11.3), removing them from circulation. The combination of permanent bonding and burning will deflate supply over time, creating upward price pressure.

2.1.3 Price stability

This combination of a fixed price ceiling (new tokens can always be minted for F_M per Loud token) with the removal of tokens from circulation over time should maintain loose real-world price consistency in Loud tokens. If the market price of Loud tokens goes above the price set by the minting system, market participants recognizing this arbitrage opportunity will mint new tokens and sell them, bringing the price back down. Conversely, if the market price of Loud goes below F_M, the token supply in circulation will dwindle as tokens are burned and bonded by users, creating positive price pressure. This dwindling will be algorithmically managed by P_B. If the minting system has reserve funds available, it will also use those funds to buy back Loud tokens as the price declines, creating further positive price pressure.

2.2 Audius (ticker AUDS): Token for protocol governance

Audius tokens are used by service providers to help run the Audius protocol, and should allow transparent and direct value capture and accrual.

Audius tokens will be divisible and transferable, with a fixed supply of 1 billion tokens from the moment of inception. Service providers must bond these tokens in order to operate a discovery service (with a larger bond correlating to a higher probability of being chosen by listener clients), create a new discovery service API version, be an arbitrator (with a larger bond correlating to a higher probability of being chosen for cases), file certain

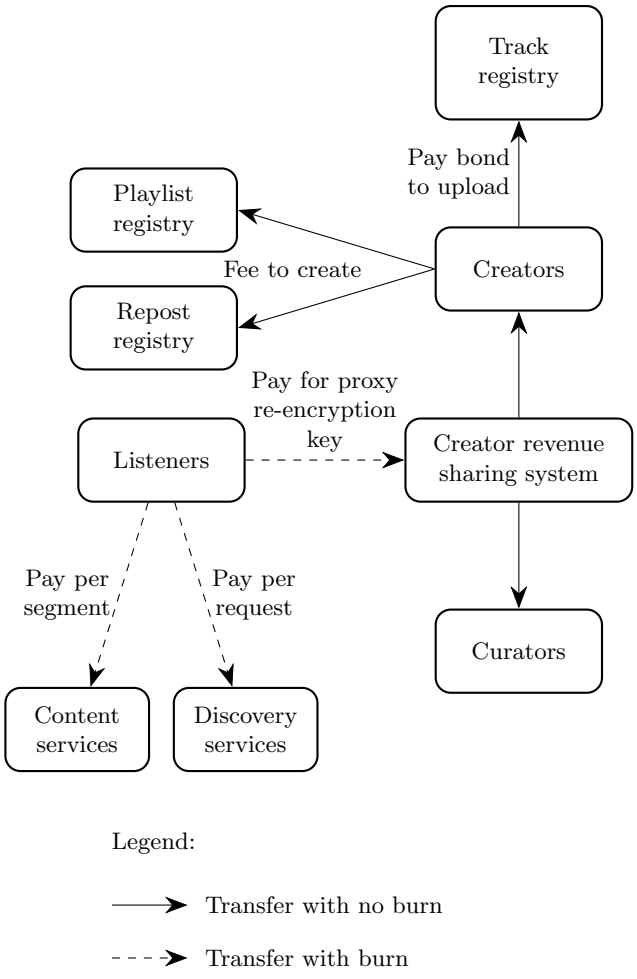

Figure 2: Listener and creator Loud payments

arbitration claims, and vote with their tokens on governance proposals. Fees for certain arbitration types and rewards for reporting invalid discovery service results are also paid in Audius tokens.

Audius token holders who have staked Audius tokens to be a service provider, including those who staked to vote in the governance system, will earn a share of the Loud minting proceeds. Service providers must stake Audius tokens to register their services; this requires one to have a stake in the protocol's long-term success in order to operate on the network, aligning their incentives with long-term network value creation.

The proceeds of minting collected in intermediate currencies will be used to purchase Audius tokens at market price, and these tokens are distributed to stakers of Audius tokens proportionally to how much they have staked as part of the service provider protocols. These minting proceeds partially compensate Audius stakers for the services they provide to the protocol, and give

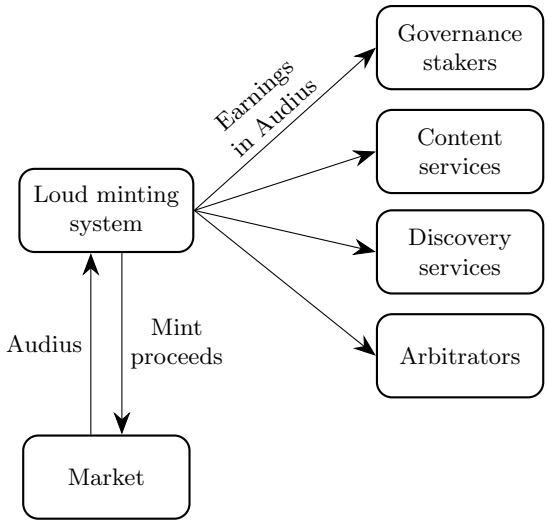

Figure 3: Earnings for stakers of Audius tokens

them a financial incentive to increase usage of Audius (more usage should lead to more minting).

Stakers of Audius tokens also have unique governance voting rights.

2.3 Why create new tokens?

This structure poses an interesting question: why create the new tokens for Audius and Loud? Other cryptoassets already exist that can serve some of the functions of our tokens.

There are many projects building price-stable currencies that may meet some of the necessary requirements of Loud. Putting aside the issues faced by many of these currencies today, including the untested nature of most options, the Loud token mint and burn mechanic gives Audius token holders an incentive to increase protocol usage, as increased usage should cause more Loud to be minted. Loud's design also allows Loud token earners to earn voting power in the protocol over time; earnings could be difficult to measure when using currencies not controlled by the protocol. However, as alternative price-stable token projects become more viable, the protocol could switch to using one or more of them or add them alongside Loud as a payment option, replacing P_B with a similarly-sized percentage paid to Audius token stakers. This fee would encourage similar protocol incentives and compensate stakers for services provided to the network.

Audius tokens exist to align governance and service provider incentives with financial incentives to increase protocol usage and create long-term protocol value. Participation in governance, as well as operating a service

provider, allows an Audius token holder to earn a share of the revenue generated by the minting of Loud tokens, incentivizing governance token holders to increase protocol usage.

3 AudSP: A decentralized storage protocol

Files shared through the Audius protocol must be highly available, independently verifiable, and decentralized. These principles are key to ensuring democratic participation and accessibility for all users of the Audius protocol. Artists sharing their tracks and metadata, listeners retrieving content, and service providers will all share longer-form information via this protocol, while references to files in this protocol will reside in the Audius blockchain. Additionally, the storage protocol must provide an equivalent user experience to existing centralized solutions and scale effectively as network demand increases.

To that end, we propose AudSP: a decentralized storage solution for the Audius network built on IPFS (InterPlanetary File System. IPFS enables modular object-level encryption, global distribution capability, secure content addressing, and object immutability [8, § 3.5.4]. In order to encourage high availability for files stored through the Audius protocol, AudSP will provide an incentive structure for users to host network content.

File references and associated metadata stored in the Audius blockchain will be IPLD links [9]. As the decentralized storage market matures, the Audius protocol may be extended to include other storage solutions such as FileCoin [10], Sia [11], or Swarm [12].

3.1 Content services

Content services are nodes in the Audius network which serve files in response to listener requests. The price of fetching a segment of a given quality will be defined at the protocol level, and can be modified via the governance protocol.

In order to be paid via the Audius network, a content service's IPFS Node ID multihash and wallet address must be registered on the Audius blockchain. To request a given segment, a listener client must first identify an appropriate content service by querying which IPFS nodes have pinned the segment and cross-referencing this set of nodes with the list of known content services. The client would then submit a request to the IPFS node for the segment along with the required payment.

3.2 Availability

The creator's node, detailed further in Section 11.1, will ensure that one copy of the creator's content and metadata is always available by permanently pinning [8, § 3.5.3] it. Content services have an incentive to pin files that are being listened to frequently, as they earn revenue for streaming that content. Once a listener has streamed a given track segment, their client can also pin and stream that content to others to earn Loud tokens, acting as a content service itself. These mechanisms should mean that as a track becomes popular, its replication factor will increase, commensurately increasing its availability.

4 Track upload and management

The track upload and management protocol for Audius comprises:

- A consistent audio content and metadata format specification to ensure accessibility (similar to the OMI metadata spec [13, § 3.7.1-3.7.2])
- A decentralized process for creators to control:
  - Track content
  - Revenue splits
  - Content ownership structure

To share a track on Audius, creators must (1) agree to the Audius open license (this license will be published in a separate brief), and (2) bond a fixed Loud token amount, denoted B_U. This bond is used to incentivize verifiers to analyze and arbitrate on the track, and remains bonded unless a track is delisted (either by a successful arbitration claim or at the creator's election). If the creator delists their content, their upload bond is burned.

The creator's client will then (1) slice the track into fixed-length segments, (2) encrypt them locally, and (3) upload these encrypted segments, the encryption key, and required metadata to their creator node, an always-on service operated by the creator (see Section 11.1 for more details). The creator node will then share the content and metadata on AudSP, producing an IPLD link for the metadata which the creator client will add to the Audius blockchain via a new transaction.

```
Track {
   owner_address
   map(creatorId => ownership)
   metadataIPLD
   ... other metadata ...
}
```

Where the linked metadata could be a JSON file structured along these lines:

```
{
  "trackTitle": "...",
  "segmentIpldLinks": ["...", "...", ...],
  ... other metadata ...
}
```

The creator can then modify track content/metadata by sharing the modified content to IPFS and updating the metadata IPLD link in the Audius blockchain.

4.1 Proxy re-encryption

The cryptosystem used to encrypt tracks will allow the issuance of listener-specific proxy re-encryption keys derived from the track encryption key and the listener's public key. The creator's node will handle key requests and issue new keys when valid payment for a track is made, issuing a new key by mixing the track encryption key with the listener's wallet's public key.

After fetching encrypted content and a re-encryption key, the listener client would locally decrypt the content using their wallet private key as follows:

```
proxied = reencrypt(encrypted_content,
                    reencryption_key)
plaintext = decrypt(proxied,
                    wallet_privkey)
```

This decrypts a given piece of content by locally re-encrypting it using the aforementioned key and subsequently decrypting it with the user's own private key. There is no 3rd-party proxy, but proxy re-encryption applied in this way allows everyone to share the same encrypted content while users can only decrypt the content on a case-by-case basis. Potential cryptosystems, including AFGH [14], are still being evaluated at this time. More details on the specific cryptosystem chosen will be published at a later date.

4.2 Revenue escrow

Track earnings will be escrowed for a grace period of T_{UE} days after upload, after which they will be disbursed to creators in real-time as payments are made by listeners. T_{UE} will be set at a later date, and will be modifiable using the Audius governance protocol.

This grace period gives potential claimants time to file arbitration claims to challenge ownership or establish a revenue sharing structure in the case of derivative content, with escrowed earnings being split or redirected according to the outcomes of these cases. This disincentivizes fraudulent upload behavior. If no arbitration case has been filed, after T_{UE} days have elapsed the escrowed earnings will be disbursed to the original uploader and their designated rightsholders. If a claim has been filed, earnings will remain escrowed until the dispute is resolved. At any point after the first T_{UE} days have elapsed, if an arbitration case is opened to dispute rights associated with a track, earnings for that track will stop being paid to existing rightsholders and be escrowed until the resolution of the case.

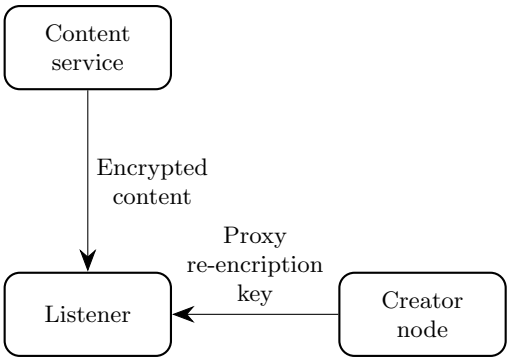

Figure 4: Listener track unlock process

5 Payments and revenue sharing

In accordance with the Audius project philosophies, the system for listener payments and revenue sharing should be:

1. Transparent for all parties involved
2. Cost- and time- efficient for all transactions
3. Flexible, accounting for multiple listening models and different listener behavior
4. Granular, with users paying each other directly and immediately for services rendered when possible

Listener clients accessing Audius can pay creators via two methods: pay-per-stream and subscription. A percentage of each payment, denoted P_B, is burned by the network, as detailed in Section 2.1. In addition, listener clients are responsible for paying content services and discovery services directly for services they consume on a per-request basis; these fees are not included in a subscription.

5.1 Pay-per-stream, ad-supported, and subscription models

The initial pay-per-stream model will have a fixed cost per track listen. Pay-per-stream listening could be paid

for by the listener or be ad-supported, with the client making payments on behalf of the listener as the listener views ads. This would make usage free for the listener if they choose.

The subscription model will have a fixed cost per month, and listens would be logged in a similar fashion to the pay per stream model, but the listener would not make a payment for each listen. On a recurring basis, subscription listens would be tallied and payouts would be made to artists by a transparent, auditable subscription system running on the Audius blockchain.

Payment is enforced by the encryption scheme used on shared content. A listener can retrieve an encrypted segment from decentralized storage, but must make a request and payment to the creator's node to generate a proxy re-encryption key to unlock the content. Further detail on creator nodes is available in Section 11.1.

This model will likely be modified over time as more protocol usage data is gathered.

5.2 Revenue sharing

All creator revenue is earned in Loud tokens and paid via the creator revenue sharing system on the Audius blockchain. This system executes any required revenue escrowing and splits revenue between the content creators and content curators. Curators earn revenue on a track listen if the track was discovered through the curator's repost or playlist. This attribution process would work similarly to online referral codes—a listener client would self-report the curators that facilitated a track listen.

6 Discovery

In order for a listener to discover content on the network, Audius needs a mechanism for indexing metadata that is efficiently queryable by users. Based on the philosophies of the Audius project, this index must be:

- Decentralized
- Efficient and straightforward for user clients to consume (promoting accessibility)
- Provably correct and transparent, eliminating profit incentives to manipulate the results returned to users
- Extensible, so that the Audius community can explore different ranking and searching methodologies.

These requirements rule out the most decentralized options due to usability and efficiency issues, e.g. users replicating the Audius blockchain locally and querying their local dataset. This section outlines a protocol for a class of discovery services to form, serving this function in a way that meets the above requirements.

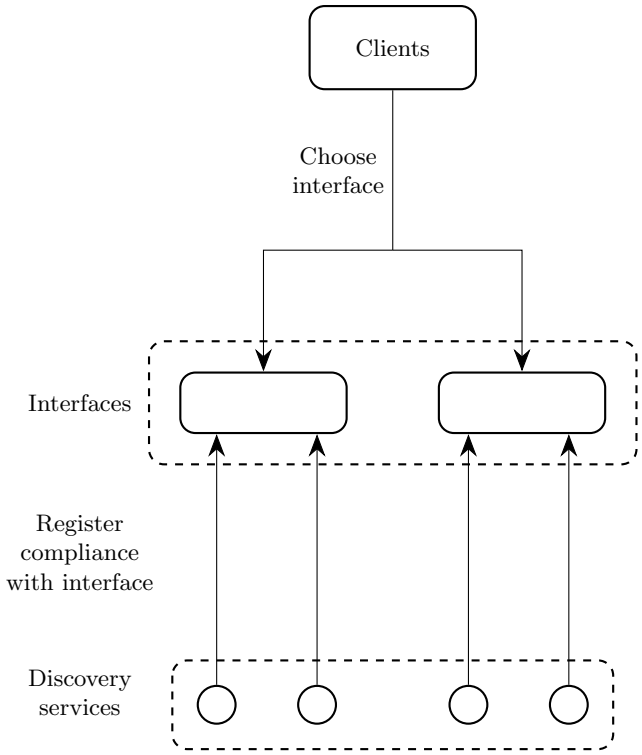

Figure 5: Discovery API interface registration and usage

Discovery service providers earn revenue by:

- Designing new discovery API interfaces that others use
- Providing a discovery service for clients to consume, indexing the Audius blockchain in compliance with a specific discovery API interface

Discovery services are read-only. Clients can use them to fetch a listener's feed, a playlist, song and creator metadata, search the corpus of Audius entities, and execute other queries about the network. Anyone can register a discovery service if they meet the requirements outlined in this section.

Clients pay discovery services a flat rate of Loud tokens per request, denoted F_D, and a percentage of this payment is burned by the network (P_B) by the mechanism described in Section 2.

6.1 Discovery API interfaces

Audius will produce a first-party discovery API interface, but other community members are encouraged to author their own interfaces that extend or modify the core API. The protocol allows listeners to select any discovery API

interface.

A service provider can register a new interface by providing the required metadata and staking a set number of tokens, denoted B_{DV}. They then earn a flat percentage of all revenue earned by discovery services using their interface (denoted P_{DV}), creating an incentive for API interface authors to lobby clients and users to use their interface. Providers can delist their interface at any time to withdraw their initially bonded tokens.

An API interface must index new blocks from the Audius blockchain atomically (i.e. all-or-nothing), and all API methods must be deterministic. Because of these requirements, for a given block hash, all discovery services running a given API interface will produce identical results for the same query. This consistency guarantee is essential for the penalty mechanism described in Section 7.3.

6.2 Discovery services

To operate a discovery service, a service provider must bond at least a set number of Audius tokens, denoted B_{DP}. In addition, they must register which discovery API interface they are operating, geographic coordinates, and one or more endpoints for reaching their discovery service. A service provider can request to delist their discovery service to unlock the initially bonded tokens. If a service provider's bonded token balance declines below B_{DP}, they are automatically delisted and their remaining balance of tokens is refunded.

Listener clients are expected to prefer discovery services with the fastest response times for their queries, incentivizing discovery service providers to provision infrastructure in high population density areas. Clients could automatically discover which discovery service provides them the lowest latency, using physical proximity and size of stake as hints. There is little incentive for service providers to misreport their service's location as listener clients will de-prioritize services with poor response latencies.

6.3 Enforcing accurate results

Every response a discovery service returns is signed with the private key that was used to bond the original tokens, the block hash of the block they have incorporated up to that point, the wallet address of the querier, and the API interface they are using to generate results. Blocks are indexed atomically and API methods are deterministic, meaning that every discovery service should produce identical results for the same query, block hash, API interface, and wallet address. If a discovery service produces invalid or inaccurate results, the signed result document returned by the service is a self-contained proof that the given service produced the given set of results.

Anyone can open an arbitration case with a discovery service's invalid signed result document by filing a claim and bonding a set number of Audius tokens for the duration of the case (denoted B_{AD}). A valid claim confirming invalid discovery results earns the claimant B_{AD} tokens in addition to a refund of their initial bond. Arbitrators in the case earn and split B_{AD} tokens, and the service provider loses $2B_{AD}$ tokens from their bonded balance to fund these rewards. If the claim is invalid, the claimant loses their original bond which is used to compensate arbitrators. Over time, we foresee a class of users emerging who use automated tools to query and find discovery API services who are producing inaccurate results, earning revenue.

7 Arbitration

Disputes may arise in Audius around who owns what content, whether a revenue split should be modified (for derivative content or other reasons), and enforcing honest behavior of service providers and track uploaders. Audius will have a network of neutral third party arbitrators voting on the outcomes of these cases to efficiently resolve disputes within the community. This protocol is designed to find consensus around disputes, resolving them in an efficient and decentralized manner.

7.1 Case types

For each case type (Table 2), a very clear and objective set of decision making guidelines will be published for all arbitrators to follow as a guide. A copy of these guidelines will be included in a contract on the network, and updates to these guidelines flow through the Audius governance protocol. A full fee and bond schedule for arbitration will be published closer to the time of the Audius main network launch, and these fees and bonds can be modified in the Audius governance protocol.

7.2 Arbitrators

Anyone can register themselves as an arbitrator by bonding at least B_A Audius tokens in the arbitration system. If their balance ever falls below this minimum amount, they are automatically delisted and their remaining balance of tokens is refunded.

7.3 Arbitration process

7.3.1 Case initiation

An arbitration case would be started by filling a claim with the arbitration system, including which type of case

Table 2: Arbitration case types

Case type	Claimant	Outcome type	Supporting data	Arbitration fee
Requesting outright ownership of percentage (up to 100%) of revenue generated by track due to real-world ownership of rights	Creator / rightsholder	Binary	Link to copyright or other proof of ownership in some jurisdiction, sybil-resistant proof of identity (i.e. claimant is person in copyright filing)	F_{AO} Loud tokens (fee)
Requesting addition of a revenue split for a track that is derived from a track owned by the claimant	Creator	% claim to revenue of derivative track	ID of original song in Audius (which claimant must own rights to), ID of derivative song	F_{AR} Loud tokens (fee)
Accuracy of discovery service results	Service provider	Binary	Inaccurate response (including signature of service provider)	B_{AD} Audius tokens (bonded for case duration, returned if claim successful)
Track content is invalid or does not comply with protocol	Service provider	Binary	Track ID	B_U Loud tokens (bonded for case duration, returned if claim successful)

it is, all required supporting information (different types of cases require different data), and a fee or bond of Audius or Loud tokens depending on the type of case.

7.3.2 Arbitration committee

The system will randomly choose N_{Ai} initial arbitrators for each case; the odds of a given arbitrator being chosen are directly proportional to the number of Audius tokens they bonded in the arbitration system. The current number of arbitrators evaluating a case is denoted N_A. The chosen arbitrators would then have 48 hours to submit a cryptographic hash of their response to the case. If they do not respond, a penalty is deducted from their staked balance and added to the arbitration fee for the case. The responses are hashed to prevent other arbitrators from changing their response based on preexisting responses.

Creator nodes are required to supply proxy re-encryption keys to arbitrators chosen to participate in arbitration of a case that includes their tracks for free. If they do not provide these keys in a timely fashion, the arbitration case is automatically resolved in favor of the claimant.

Each arbitration case will have its own decentralized environment to facilitate conversation and enable consensus-driven decision-making. We envision this comprising a chat thread, timestamped comments, and ownership attribution or sample annotation for relevant case types. There is already a global community of music enthusiasts annotating tracks for their own enjoyment [15], and we hope to empower them to earn revenue and notoriety for doing this on Audius. This unpermissioned forum will help arbitrators make better decisions and also provide an onramp for all others to register as formal arbitrators.

7.3.3 Vote tallying

After the 48 hours has elapsed, if fewer than 60% of arbitrators respond, a second round of arbitration for another 48 hours begins with an additional $N_A - n$ randomly selected arbitrators, where n is the number who replied in the first round. Subsequent rounds of arbitration will continue until at least $0.6N_A$ replies are aggregated. A penalty is levied against the bonded balance of chosen arbitrators who fail to respond in the 48-hour window.

After the 1 or more rounds of arbitration are completed, there is a 12 hour "reveal" phase where each arbitrator reveals the preimage response data used to generate their earlier published hash. If they do not reveal a valid preimage, a penalty is levied against the arbitrator's staked tokens to be added to the arbitration fee for the case, and the arbitrator is removed from the results.

The arbitration system then tallies the validated re-

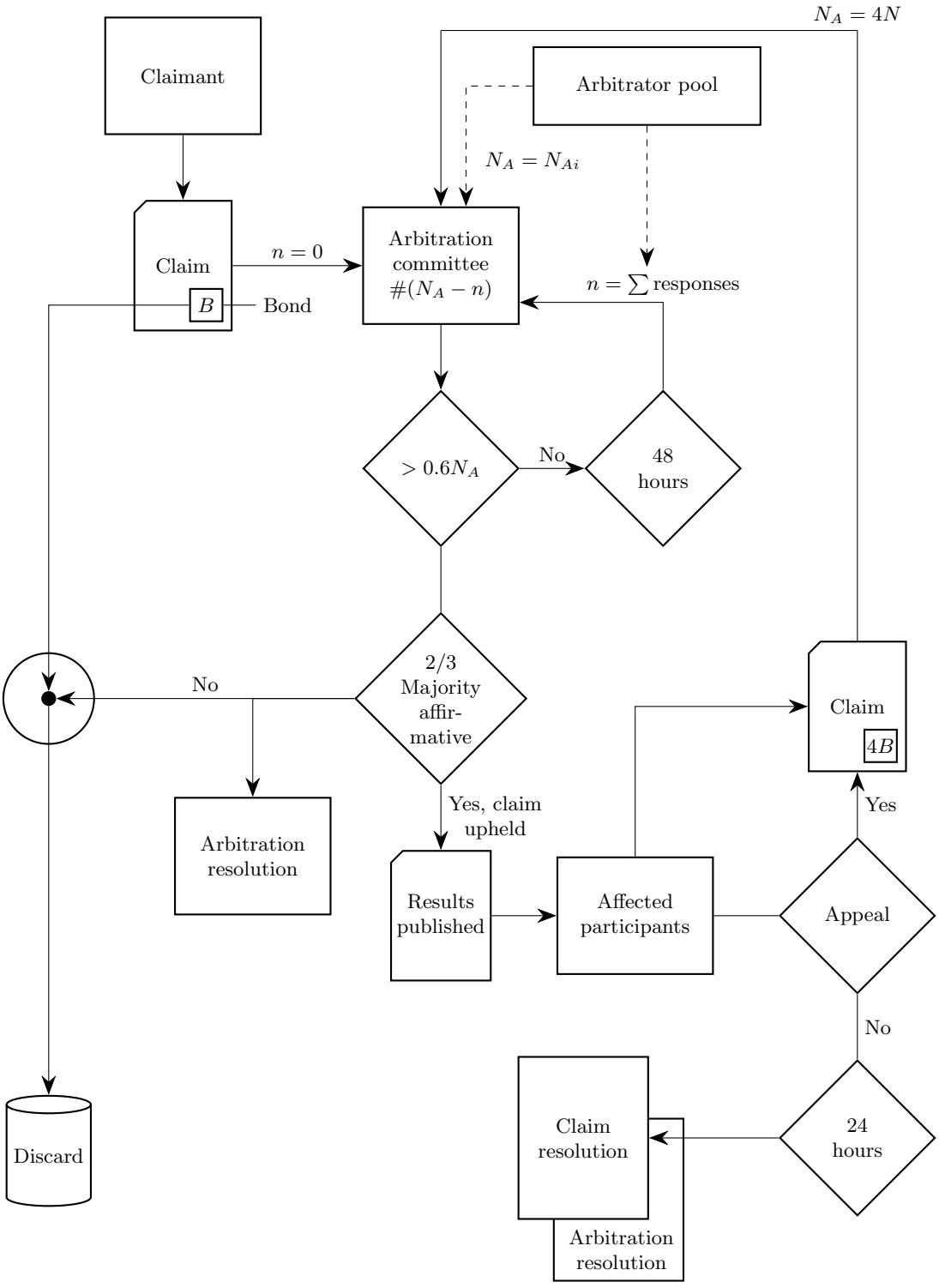

Figure 6: Arbitration protocol

sponses. There is a 24-hour grace period following the computation of results during which any network participant affected by the decision can appeal it; this process is described further below. After this grace period expires, any necessary actions are automatically carried out (ex. changing ownership amounts in accordance with the outcome).

7.3.4 Case outcomes

For binary outcome case types, a supermajority (2/3) voting affirmatively wins the case for the claimant. If an affirmative supermajority is not reached, the case is dismissed and the claimant's bond is lost.

If any supermajority (affirmative or negative) is found, arbitrators in the minority have a small fee levied against their bonded balance which is added to the arbitration fee. Those in the majority split the arbitration fee equally, i.e. if there are n arbitrators in the majority each gets $1/n$ of the arbitration fee as compensation for their services.

For the % claim to revenue case type, the mean and standard deviation of the arbitrator responses are computed. If the mean is below a minimum threshold percentage, the assigned revenue share split is 0 (no change). Otherwise, the mean is the final percentage split of revenue, and that split is added for the claimant, diluting down the revenue share of all existing participants. More concretely:

Let M be the number of rightsholders. Let P_m be the percentage revenue share assigned to rightsholder m. If a new rightsholder is assigned percentage P_{M+1}, each P_m is updated as follows:

$$P_{m'} = (1 - P_{M+1})P_m$$

with P_{M+1} being added to the distribution. The resulting distributions of percentages will be consistent, because:

Proof. Assuming the initial distribution is consistent (the percentages add to 100%), $\sum_m(P_m) = 1$. We must prove that $\sum_m(P_{m'}) + P_{M+1} = 1$ to prove that the new distribution is consistent. Expanding the summation with the above definition of $P_{m'}$, we arrive at:

$$\sum_m((1 - P_{M+1})P_m) + P_{M+1} = 1$$

We can simplify the summation as follows:

$$(1 - P_{M+1})\sum_m(P_m) + P_{M+1} = 1$$

And substitute the value of $\sum_m(P_m)$ given our assumption above:

$$(1 - P_{M+1}) + P_{M+1} = 1$$

Simplifying:

$$1 = 1$$

□

If an arbitrator is more than a certain number (N_{SD}) of standard deviations away from the mean, a small fee is levied against their bonded balance which is added to the arbitration fee. Those within one standard deviation of the mean split the arbitration fee equally: if there are n arbitrators near the mean each gets $1/n$ of arbitration fee as compensation for their services.

7.3.5 Appeal process

During the aforementioned 24-hour grace period after a decision has been made on a given case, any network participant affected by the outcome of that decision can file an appeal. The filer of this appeal must pay a fee that is quadruple the fee paid for the decision they are appealing, and the number of arbitrators elected to participate in that decision (N_A) is commensurately quadrupled, making their expected earnings equal. The same case can be appealed multiple times, by the same or differing parties, but the cost of an appeal will eventually become too great to continue.

7.4 Future improvements

As arbitration of certain tasks is automated over time, such as automated determination of revenue splits for derivative content, this same mechanism can work with arbitrators running software to make decisions on their behalf instead of submitting decisions manually. The arbitration fees would decrease, time provided for an arbitrator to respond to a given case would decrease, and the tolerances for deviation from the network mean would be lower, but the same consensus mechanism will be used to ensure service providers are actually doing the necessary computations.

8 Governance

As stated in the introduction, the mission of Audius is to create a fully decentralized community of artists, service providers, and listeners collaborating to share and defend the world's music. Integral to achieving this mission is a decentralized governance protocol, whereby artists, service providers, and listeners are individually and collectively enfranchised in decision making about protocol changes and upgrades. The Audius governance protocol, in accordance with this mission, is guided by a few high-level philosophies:

- Stakeholders should be enfranchised in decisions that affect them
- Voting power should be earned by creating value in Audius, not bought
- Participation in governance creates value in Audius, and should be rewarded
- High-stakes decisions should require more consensus, while lower-stakes decisions should be more efficient to make
- Power should be shared equitably among Audius stakeholder groups

Protocol upgrades are submitted as structured proposals, including deployed code when necessary, to an overarching governance system that has custody over all other systems that make up the Audius blockchain. Proposals also include a block count at which point they go into effect; this effectiveness date must be at least 1 week in the future at time of proposal submission to give users ample time to review and vote on the proposal.

To submit a proposal, a user must bond a set number of Audius tokens (denoted B_{GP}) in the governance system, which remain bonded for the duration of their proposal. Before a proposal's effective date, the original submitter can also choose to withdraw the proposal if they so choose, returning their bonded tokens. This bond is required as an anti-spam measure and to ensure that proposers to have a sufficient stake in the Audius protocol to make changes to it. At the proposal's resolution (successful, failed, or withdrawn), the bond is returned to proposal submitter.

8.1 Direct voting

Before the effectiveness date of a proposal, Audius users can submit a binary yes or no vote on it. The magnitude of a vote on any given proposal is determined based on the magnitude of the voter's membership in governance decision-making classes and the voting power assigned to those classes, in line with the philosophy that voting power should be earned (Table 3).

These user classes are not mutually exclusive. Therefore, if a user has earnings and/or holdings that fall into multiple classes, their vote can be counted in multiple classes.

At the moment the proposal is scheduled to go into effect, votes are tallied in each decision maker class, scaled based on the respective governance power and voter participation for a given class, and summed to produce aggregated vote counts for the affirmative or negative options.

8.2 Tallying the results of a vote

The distribution of governance power, defined G, is divided into percentages G_L, G_C, and G_P for listeners, creators, and service providers respectively, and must add to 100%. Votes in each class are tallied per Table 3 for affirmative (yes), negative (no), or abstain (no vote), with every vote being assigned to one of these categories. Consistent with the philosophy of equal power division between user groups, G_L, G_C, and G_P will all be 1/3.

Given affirmative vote counts Y_L, Y_C, and Y_P, negative vote counts N_L, N_C, and N_P, and abstaining vote counts A_L, A_C, and A_P for listeners, creators, and service providers respectively, we calculate the final vote outcome as follows:

Let V_Q be the percentage affirmative vote in a given class Q, such that:

$$V_Q = \frac{Y_Q}{Y_Q + N_Q}$$

Let R_Q be the percentage participation of a given class Q; this is defined as total votes placed divided by total potential votes, or in mathematical terms:

$$R_Q = \frac{Y_Q + N_Q}{Y_Q + N_Q + A_Q}$$

The percentage affirmative vote for an entire proposal, denoted V, is defined as follows:

$$V = \frac{G_L R_L V_L + G_P R_P V_P + G_C R_C V_C}{G_L R_L + G_P R_P + G_C R_C}$$

A mathematical proof of the consistency of this formula, as well as some example voting scenarios, is provided in Section 11.2.

The proposal is automatically carried out in the case the affirmative vote wins or withdrawn in the negative case. V must exceed either a simple majority threshold or a supermajority threshold depending on the decision type for a decision to pass (see Table 4). There is no way to reverse a proposal; a new proposal must be submitted and approved undoing the results of the original.

At the time of the Audius main network launch, there will be no quorum requirement for a governance proposal to be accepted. However, soon after network launch (and successful acceptance of some number of governance proposals), a proposal will be submitted to add quorum requirements to each decision type, with higher percentages for higher-stakes decisions.

8.3 Vote delegation

To make governance more accessible to users, voting can be delegated by anyone to other users or groups of users, such that if a user places no vote on a specific proposal,

Table 3: Governance decision-making classes

Decision maker class	How voting power is assigned
Content creators and curators	Sum of Loud tokens bonded to upload tracks, spent to curate content (create reposts or playlists), and earned via listens of listed[a] owned[b] content or curated content
Listeners	Loud tokens spent on listening activity (per-stream payments, subscriptions, etc.)
Service providers	Number of Audius tokens staked for governance[c] or to be a service provider

[a]If a piece of content is delisted by the creator, they are no longer conferred voting power for earnings of that piece of content.
[b]If ownership of a piece of content is transferred voluntarily or via the arbitration system, voting power for its historical revenue stream is transferred with it.
[c]An Audius token holder can stake their tokens in the governance system to count them towards votes on proposals. Once staked, these tokens will remain locked for a minimum period of time before they can be withdrawn. For tokens staked for governance, participation in governance is required to earn a share of Loud minting proceeds.

Table 4: Governance decision types

Decision type	Consensus type
Modify fee (denoted F_*) and bond (denoted B_*) amounts	Simple majority ($> 1/2$)
Change content upload format and metadata structure	Simple majority ($> 1/2$)
Modify revenue sharing structure between creators and curators	Supermajority ($> 2/3$)
Adjust Loud minting rate limit or burn percentage	Supermajority ($> 2/3$)
Add new governance decision type	Supermajority ($> 3/4$)
Modify required minimum vote for quorum in governance structure	Supermajority ($> 3/4$)

their designated delegate's vote will be used in place of their own. There will be two groups created at the time of main network launch: Audius DAO (Decentralized Autonomous Organization) and Artist Advisory DAO.

Audius DAO will be controlled by a small group of geographically distributed users chosen by Audius Inc., and decisions will be made by supermajority consensus of DAO members. This DAO will be delegated voting power by default on service provider signup for staked Audius tokens. However, a service provider could still delegate voting power to another user or group of their choosing.

The Artist Advisory DAO would be made up of artists who support the protocol. Members would be required to bond at least a given number of Audius tokens in an Artist Advisory DAO contract, which early in the network would be funded by artist advisory grants. The initial members of the DAO would be chosen by Audius, Inc. Current members of the DAO would vote on the admission of new members, and members of the DAO would vote on each proposal with voting power uniformly distributed across members. On listener or creator signup, by default their voting power will be delegated to the Artist Advisory DAO, but this delegation can be changed or removed at the user's election.

Any other group of users could federate to form their own voting DAO as well. We expect other groups to emerge to represent the interests of various stakeholders within the ecosystem.

8.4 Bootstrapping protocol governance

Early in the life of the Audius network, the Audius DAO will control governance. During this bootstrapping phase, the Audius DAO will also have the ability to intervene in catastrophic circumstances to fix critical issues in the Audius blockchain code, such as issues enabling fraud or resulting in unintended loss of Audius or Loud tokens. Over time, governance will be decentralized in phases, eventually reaching the fully decentralized model described above.

The Audius DAO will only exist early in the life of the network; at some point, the Audius DAO will be disintegrated, with users who have delegated voting power to Audius DAO having the option to choose a new delegate or stop delegating voting power.

9 Roadmap

Audius development will be broken into four milestones: alpha, beta, main-network launch, and decentralization.

9.1 Alpha

The first testing release of Audius will allow creators to share content and listeners to discover and consume content, using IPFS for storage and a set of smart contracts on a test network implementing track and creator registration. This will include open-source alpha implementations of the following:

- discovery service
- creator client
- creator node
- listener client

The alpha will not include payments, Audius/Loud tokens, AudSP, arbitration, or governance.

9.2 Beta

The Audius beta will remain on a testnet blockchain network, and add initial open-source implementations of the following features:

- AudSP (using test tokens and a beta content service)
- Arbitration and arbitrator client (using test tokens for staking)
- Social features

Releases after this milestone will be on a rolling basis until main-network launch, releasing new features as they are built.

9.3 Main-network launch

The Audius main-network will be the first non-testing release of Audius. This milestone will be met when the Audius blockchain moves from a testnet implementation to operating on a live main network of the blockchain platform it operates on. This milestone will add the following to the Alpha and Beta features above:

- Audius and Loud token functionality, including staking of Audius in service provider protocols
- Payments for service providers, creators, and curators

9.4 Full decentralization

Audius will reach its final milestone of full decentralization when the complete governance protocol has been released.

10 Future work

In addition to the above, we envision potentially adding the following features to the Audius protocol in future. All are in accordance with Audius' philosophies of stakeholder empowerment and transparent, democratized, and unmediated access.

1. A decentralized bounty economy protocol, enabling participants to request and complete specific tasks for a reward
2. A decentralized and transparent ad network
3. A class of non-fungible tokens (NFTs) to enable unique, personalized, and customizable experiences to maximize community engagement:
   (a) Creator token: creators could create their own tokens with unique token economics, exclusive merchandising and events, crowdfunding of content creation and tokenization of content listens
   (b) Track token: a tokenized representation of audio content, allowing transfer of ownership and exclusive or early access, among other customizations
   (c) Event token: a tokenized representation of an event like a tour or festival, consisting of merchandising and collectibles, community access, or event tickets

11 Supplemental specifications

11.1 Audius creator node specification

To share content on Audius, creators are required to run an always-online service, referred to as the creator node, to 1) service requests for proxy re-encryption keys and 2) ensure availability of their content and metadata (guaranteeing at least 1 copy is available through AudSP). In line with the project's mission to create a decentralized autonomous community, this structure gives creators autonomy over the dissemination of their content without external dependencies or points of failure.

11.1.1 Proxy re-encryption keys

At registration, a creator must log one or more IP addresses and/or fully-qualified domain names where their creator node can be reached to provide proxy re-encryption keys. When beginning to listen to a track, a listener's client will make a request to the creator's node, including a payment, for a proxy re-encryption key specific to the segment. If the creator node fails to reply with a valid key, the payment is revoked.

To service this request, the creator node derives a proxy re-encryption key using the listener's public wallet key and the private key used to encrypt the requested track and returns it to the listener. Because the re-encryption key is specific to the creator, listener, and segment, it can be transmitted insecurely or published without revealing the track contents to the greater network. More detail on the cryptosystem enabling this can be found in Section 4.1.

11.1.2 Availability

The creator node is responsible for ensuring availability of the creator's own content and metadata, but not for providing significant bandwidth to service requests. Optionally, the node can be configured to provide greater bandwidth and earn content request fees as a content service (see Section 3.1). Metadata must be shared permanently by the creator node, but should only be fetched by discovery services when indexing the Audius blockchain. If the replication factor (number of copies on AudSP) of a creator's given encrypted content file is above a threshold, the creator node could stop sharing the file without deleting the locally saved copy. The creator node would continue to monitor the Audius storage network, and if the replication factor drops below the required threshold it would re-share the file.

11.1.3 Creator node uptime

Failure to keep a creator node online can result in loss of track ownership, tracks being marked as "unverified" (making them undiscoverable), and loss of revenue during downtime.

11.1.4 Delegation of creator node responsibility

The creator node service will be straightforward for a creator to set up and operate themselves (for example, Mediachain created a simple tool to help users set up their Mediachain nodes [16]), but requires high availability. We foresee a class of service providers emerging to run nodes on behalf of creators for a small recurring fee. Many creators will likely use these services to avoid having to run their own creator node, which may be cumbersome to operate for a non-technical user.

11.2 Governance vote tallying

11.2.1 Proof of vote calculation consistency

Here we provide more rigor on the consistency of the governance vote tallying system.

From 8.2, we defined V_Q to be the percentage affirmative vote in a given class Q, such that: $V_Q = Y_Q/(Y_Q + N_Q)$.

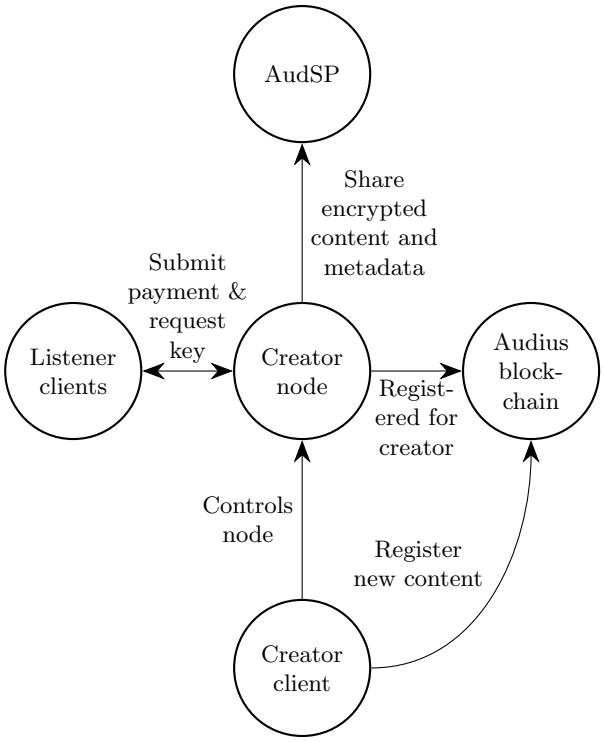

Figure 7: Creator node interactions with protocol participants

The negative vote count is calculated as $N_Q/(Y_Q + N_Q)$. We can prove the affirmative and negative vote in a category will always add to 100%, maintaining a consistent distribution:

Proof.

$$\frac{Y_Q}{Y_Q + N_Q} + \frac{N_Q}{Y_Q + N_Q} = 1$$
$$\frac{Y_Q + N_Q}{Y_Q + N_Q} = 1$$
$$1 = 1.$$

□

Given this, it logically follows that the negative vote in a category is equal to $1 - V_Q$.

As defined in Section 8.2, R_Q is the percentage participation of a given class Q:

$$R_Q = \frac{Y_Q + N_Q}{Y_Q + N_Q + A_Q}$$

The percentage affirmative vote for an entire proposal, denoted V, was defined as follows in Section 8.2:

$$V = \frac{G_L R_L V_L + G_P R_P V_P + G_C R_C V_C}{G_L R_L + G_P R_P + G_C R_C}$$

Where the vote for each class is scaled by the participation rate and governance power of the class. The normalization factor $(1/(G_L R_L + G_P R_P + G_C R_C))$ ensures the resulting distribution of votes remains consistent (the sum of yes and no voting percentages on a given proposal must equal 100%).

The percentage negative vote substitutes $(1 - V_Q)$ for each vote variable above:

$$\big[G_L R_L(1 - V_L) + G_P R_P(1 - V_P) + G_C R_C(1 - V_C)\big] \times \frac{1}{G_L R_L + G_P R_P + G_C R_C}$$

We can prove that the normalization factor $(1/(G_L R_L + G_P R_P + G_C R_C))$ leads to a consistent vote distribution in the final tally (i.e. the votes tally to 100%) by adding the yes and no votes and showing they add to 100%:

Proof.

$$\frac{G_L R_L V_L + G_P R_P V_P + G_C R_C V_C}{G_L R_L + G_P R_P + G_C R_C} + \frac{G_L R_L(1 - V_L) + G_P R_P(1 - V_P)}{G_L R_L + G_P R_P + G_C R_C} + \frac{G_C R_C(1 - V_C)}{G_L R_L + G_P R_P + G_C R_C} = 1$$

Factoring out the common term:

$$\big[(G_L R_L V_L + G_P R_P V_P + G_C R_C V_C) + (G_L R_L(1 - V_L) + G_P R_P(1 - V_P) + G_C R_C(1 - V_C)\big] \times \frac{1}{G_L R_L + G_P R_P + G_C R_C} = 1$$

Factoring out the common terms again:

$$\big[G_L R_L(V_L + (1 - V_L)) + G_P R_P(V_P + (1 - V_P)) + G_C R_C(V_C + (1 - V_C))\big] \times \frac{1}{G_L R_L + G_P R_P + G_C R_C}$$

And simplifying further, we get:

$$\frac{G_L R_L + G_P R_P + G_C R_C}{G_L R_L + G_P R_P + G_C R_C} = 1$$
$$1 = 1.$$

□

Based on this, it logically follows that the percentage negative vote on an entire proposal is $(1 - V)$.

11.2.2 Example decision result calculation

To illustrate the way votes will be tallied, consider a decision with the example votes and magnitude of user membership in given decision making classes in Table 5.

With an example power distribution of G being:

$$G_L = G_C = G_P = \frac{1}{3}$$

The affirmative vote can be calculated using the formula above:

$$V = \frac{G_L R_L V_L + G_P R_P V_P + G_C R_C V_C}{G_L R_L + G_P R_P + G_C R_C}$$

$$R_L = \frac{Y_L + N_L}{Y_L + N_L + A_L} = \frac{(600 + 100) + 300}{(600 + 100) + 300 + 50} = 0.95238$$

$$R_C = \frac{Y_C + N_C}{Y_C + N_C + A_C} = \frac{(10,000 + 50,000) + 0}{(10,000 + 50,000) + 0 + 500} = 0.99178$$

$$R_P = \frac{Y_P + N_P}{Y_P + N_P + A_P} = \frac{0 + 1000}{0 + 1000 + 0} = 1$$
$$V_L = \frac{Y_L}{Y_L + N_L} = \frac{600 + 100}{600 + 100 + 300} = 0.7$$
$$V_C = \frac{Y_C}{Y_C + N_C} = \frac{10,000 + 50,000}{10,000 + 50,000 + 0} = 1$$
$$V_P = \frac{Y_P}{Y_P + N_P} = \frac{0}{1000} = 0$$

$$V = \Big(\frac{1}{3} \times 0.95238 \times 0.7 + \frac{1}{3} \times 1 \times 0 + \frac{1}{3} \times 0.99178 \times 1\Big) \times \Big(\frac{1}{3} \times 0.95238 + \frac{1}{3} \times 1 + \frac{1}{3} \times 0.99178\Big)^{-1}$$
$$V \approx 56\%$$

Depending on the consensus mechanism employed, this proposal may or may not pass (56% yes is a majority but may not be a supermajority). Because user 5 placed no vote in Table 5, their vote reduced the participation rate of creators and listeners slightly.

Table 5: Distribution of voters and voting power for example decision

User	Vote	Creation/curation earnings	Listening spend	Audius tokens staked
1	Yes	10 000	600	0
2	Yes	50 000	100	0
3	No	0	0	1000
4	No	0	300	0
5	None	500	50	0

11.3 Social features and listener feed

Listeners can take the following actions within Audius:

- Listen to a track
- Like a track, adding it to the listeners' own library
- Follow other listeners and creators, and receive notifications when new original content, reposts, playlists, or comments are created by them
- Create a private playlist

In addition to the above, listeners and creators will burn Loud tokens in order to take the following actions that consume network resources:

- Create a publicly indexed and discoverable playlist (burn F_{SP} Loud tokens)
- Repost tracks to followers (burn F_{SR} Loud tokens)
- Comment on tracks, albums, reposts, playlists (burn F_{SC} Loud tokens)

All social actions within Audius are represented on the Audius blockchain, meaning users can use any client to connect to Audius and see the same social graph. Listeners can also view what other listeners have been listening to, as can service providers building third-party clients. This opens up many possibilities around content recommendation systems and alternative client experiences built by members of the Audius developer community.

12 Glossary

Arbitrator: someone registered to vote on arbitration of disputes within Audius.

Audius blockchain: the collection of smart contracts and blockchain-based code that is used for decentralized coordination within the Audius protocol.

Audius protocol: the amalgamation of clients interacting with each other and with the Audius blockchain.

Audius token: the token used for value accrual in Audius. Staked by some types of service providers to operate within the network, and earned by arbitrators in some cases. Staking this token to provide services allows one to earn a share of proceeds generated by the minting of Loud tokens.

AudSP: a protocol for serving and fetching data in the Audius protocol. Built on IPFS.

Content service: a service that serves encrypted audio segments to users of AudSP.

Creator: someone who creates content and shares it on Audius.

Creator client: the UI used by creators to control their creator node, view and manage earnings, and interface with the track upload and management protocol.

Creator node: an always-on API service operated by creators to issue proxy re-encryption keys for their content.

Creator revenue sharing system: a system that executes any required escrowing of earnings and splits track revenue among creators, rightsholders, and curators.

Curator: someone who creates reposts or playlists on Audius. They capture a share of the revenue generated by listens of content facilitated by their playlists and reposts.

Discovery interface: a version or type of the discovery service that indexes and responds to queries in a specified way.

Discovery service: an API service that indexes the Audius blockchain and responds to queries of that index, earning a fee for each request. Must be compliant with its specified discovery interface.

Listener client: the UI used by listeners to consume content in Audius.

Listener: someone who consumes content in Audius.

Loud minting system: a system that creates new Loud tokens on demand for a fixed real-world price.

Loud token: the token used for value transfer within Audius. Used by listeners to pay for content. Earned by creators, content services, discovery services, and arbitrators in some cases.

Oracle: an external service that provides real-world data (in our case price data) to a blockchain-based system.

Proxy re-encryption: An encryption scheme by which a derivative key can be used to transform data encrypted by one key into a version that can be decrypted by another key.

Rightsholder: someone who earns a share of revenue generated by a given piece of content.

Service provider: Someone or something that does work on behalf of users and the network. Operators of discovery services, operators of content services, arbitrators, arbitrageurs, operators of outsourced creator node services, stakers in governance, and those querying discovery services and filing claims for invalid results are all service providers.

References

[1] Micah Singleton. *This was Sony Music's contract with Spotify*. May 9, 2015. URL: https://bit.ly/audius-sony-spotify (visited on 08/28/2018).

[2] Jason B Bazinet, Mark May, Kota Ezawa, Thomas A Singlehurst, Jim Suva, and Alicia Yap. *Putting the Band Back Together. Remastering the World of Music*. White paper. Citi GPS: Global Perspectives & Solutions, Aug. 2018. URL: https://bit.ly/audius-remastering-music (visited on 08/28/2018).

[3] Sam Mamudi. *NFL lockout ends as players OK deal with owners*. July 25, 2011. URL: https://bit.ly/audius-nfl-lockout (visited on 08/28/2018).

[4] David Aldridge. *NBA, NBPA reach tentative seven-year CBA agreement*. Dec. 14, 2018. URL: https://bit.ly/audius-nba (visited on 08/28/2018).

[5] Dani Deahl and Casey Newton. *How SoundCloud's broken business model drove artists away*. July 21, 2017. URL: https://bit.ly/audius-soundcloud-model (visited on 08/28/2018).

[6] Josh Constine. *SoundCloud sinks as leaks say layoffs buy little time*. July 12, 2017. URL: https://bit.ly/audius-soundcloud-layoffs (visited on 08/28/2018).

[7] *Sharding FAQs*. Ethereum. 2018. URL: https://bit.ly/audius-ethereum-sharding (visited on 09/14/2018).

[8] Juan Benet. *IPFS—Content Addressed, Versioned, P2P File System (Draft 3)*. 2014. URL: https://bit.ly/audius-ipfs (visited on 08/28/2018).

[9] *IPLD Specifications*. Protocol Labs. 2017. URL: https://bit.ly/audius-ipld (visited on 08/28/2018).

[10] *Filecoin: A Decentralized Storage Network*. Protocol Labs. 2017. URL: https://bit.ly/audius-filecoin (visited on 08/28/2018).

[11] David Vorick. *How to Put Data on the Sia Network*. Nebulous, Inc. 2017. URL: https://bit.ly/audius-sia (visited on 08/28/2018).

[12] *What is swarm?* Ethersphere. 2017. URL: https://bit.ly/audius-swarm (visited on 08/28/2018).

[13] *OMI Requirements Documentation Minimum Viable Interoperability 1.0*. Specifications documentation. July 26, 2017. URL: https://bit.ly/audius-open-music-initiative (visited on 08/28/2018).

[14] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage". In: *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS)*. 2005. URL: https://bit.ly/audius-proxy-re-encryption (visited on 08/28/2018).

[15] *The WhoSampled App*. WhoSampled.com Limited. 2018. URL: https://bit.ly/audius-whosampled (visited on 09/02/2018).

[16] *Using Mediachain Deploy*. Mediachain. 2018. URL: https://bit.ly/audius-mediachain (visited on 09/02/2018).