Community Update

1

April 5. 2021

joystream

Antioch Network

1

Sumer Network

2

Olympia Network

3

Hydra v3

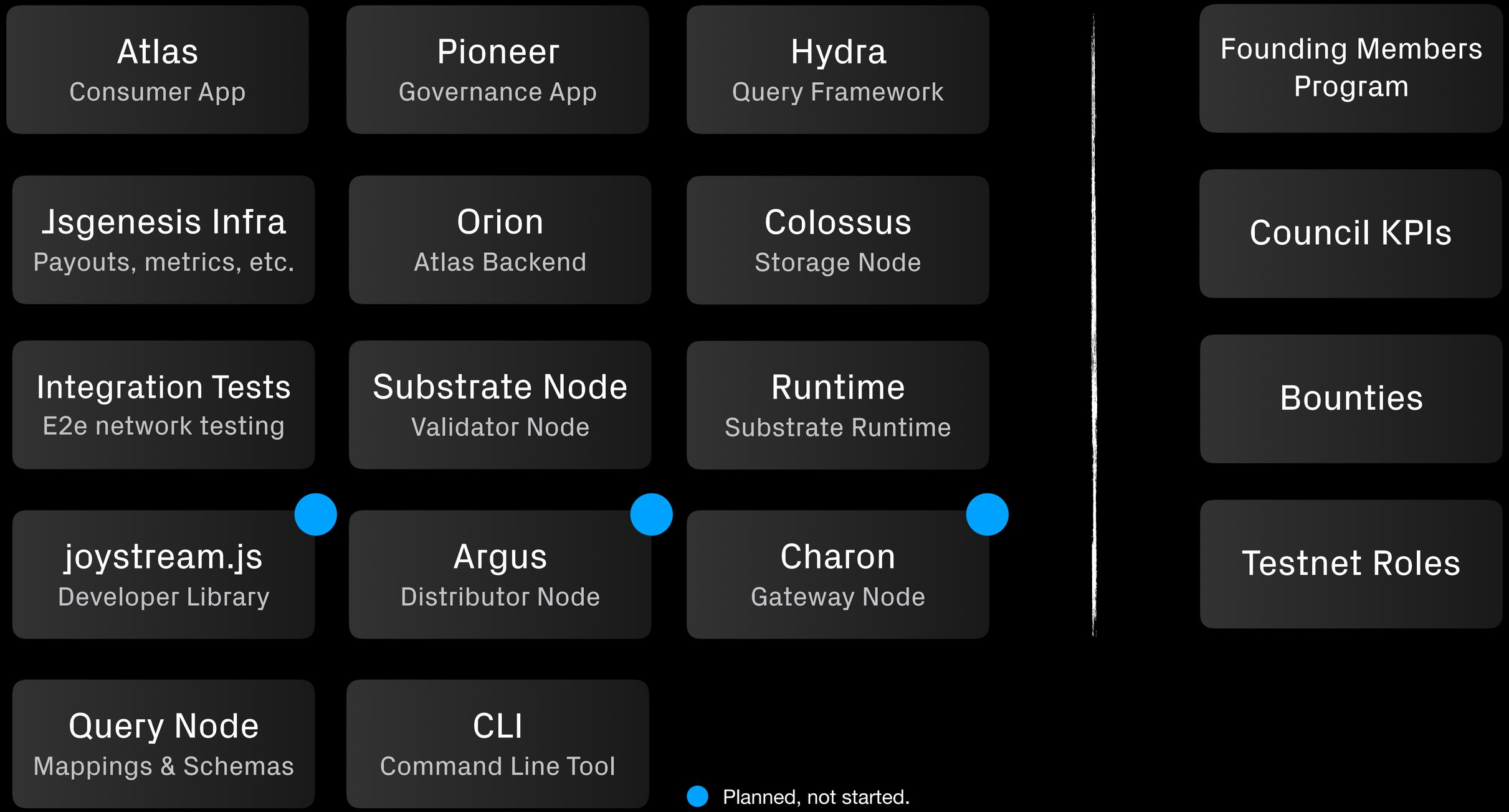
4

Community

5

New Specifications

6


Atlas
Consumer App
Pioneer
Governance App
Hydra
Query Framework
Jsgenesis Infra
Payouts, metrics, etc.
Orion
Atlas Backend
Colossus
Storage Node
Integration Tests
E2e network testing
Substrate Node
Validator Node
Runtime
Substrate Runtime
joystream.js
Developer Library
Argus
Distributor Node
Charon
Gateway Node
Query Node
Mappings & Schemas
CLI
Command Line Tool
Planned, not started.
Founding Members Program
Council KPIs
Bounties
Testnet Roles

Shamil Gadelshin
Blockchain Engineer

Arsen Kondratiev
Blockchain Engineer

Ondrej Raska
Blockchain Engineer

Gabriel Steinberg
Blockchain Engineer

Dmitrii Zhelezov
Blockchain Engineer

Martin Wessel-Berg
COO & Growth

Metin Demir
Blockchain Engineer

Maciej Gołaszewski
Front-end Engineer

Piotr Sadlik
Front-End Engineer

Vladyslav Mazuren
Front-End Engineer

Edvin Dzidic
Front-End Engineer

Bartosz Dryl
Front-End Engineer

Mikołaj Jędrzejewski
Front-End Engineer

Klaudiusz Dembler
Front-End Engineer

Diego Cardenas
Front-End Engineer

Ben Holden-Crowther
Growth

Natalia Kirejczyk
Designer

Leszek Wiesner
Front-End Engineer

Tomasz Nadratowski
Designer

Kuba Mikołajczyk
Designer

Miłosz Klimek
Designer

Bedeho Mender
CEO

Mokhtar Namaani
CTO

Antioch Network

Antioch Network Launching

Rescuing the Babylon network.

ANTIOCH

Announcing Antioch

After encountering a chain split bug on the now unrecoverable Babylon testnet, we have decided to transition over to the patched and improved Antioch network.

Babylon Network

3 months

Runtime Upgrade

~1 week ago

Chain split

New Runtime

Old Runtime

Bug in Substrate

Our best hypothesis.

Polkadot.

paritytech / substrate

Watch 345 Star

Code Issues 631 Pull requests 67 Discussions Actions Projects 8 Wiki Security

Invalid signatures passed to a pallet's method stop block import #6585

Open nahuseyoum opened this issue on Jul 6, 2020 · 0 comments

nahuseyoum commented on Jul 6, 2020 • edited Contributor

We have a pallet with a function that accepts an SR25199 signature and verifies it. Our implementation looks like this:

```
pub trait Trait: system::Trait {
   ...
   type Public: IdentifyAccount<AccountId = Self::AccountId>;
   type Signature: Verify<Signer = Self::Public> + Member + Decode + Encode;
}
```

And a verify method which verifies the signature

```
pub struct Proof {
   pub signer: T::AccountId,
   pub signature: T::Signature,
}
```

```
fn verify_signature(
       proof: &Proof<T::Signature, T::AccountId>,
       signed_payload: &[u8]
   ) -> Result<(), Error<T>>
{
    match proof.signature.verify(signed_payload,&proof.signer)
    {
        true => Ok(()),
        false => Err(<Error<T>>::UnauthorizedTransaction.into()),
    }
}
```

This setup worked fine for us until recently where we discovered that our network was stopping to finalise with an error about Signature verification.

```
====================

Version: 2.0.0-rc2-5eb246fb6-x86_64-linux-gnu

0: sp_panic_handler::set::{{closure}}
1: std::panicking::rust_panic_with_hook
at /rustc/b8cedc00407a4c56a3bda1ed605c6fc166655447/src/libstd/panicking.rs:476
2: std::panicking::begin_panic
```

Assignees
bkchr

Labels
I2-bug

Projects
None yet

Milestone
No milestone

Linked pull req
Successfully m
this issue.
None yet

Notifications
You're not rece
thread.

2 participants

v2rc4

v2.0.1

Antioch Network

<1w from today

Sumer Network

Goals

1. New Content Directory.
2. New Atlas Studio.
3. New operations working group.

Live ~late April

1

New Content Directory

- Simple data model & permissions
- Extensible metadata standard
- Finalized

Content Directory

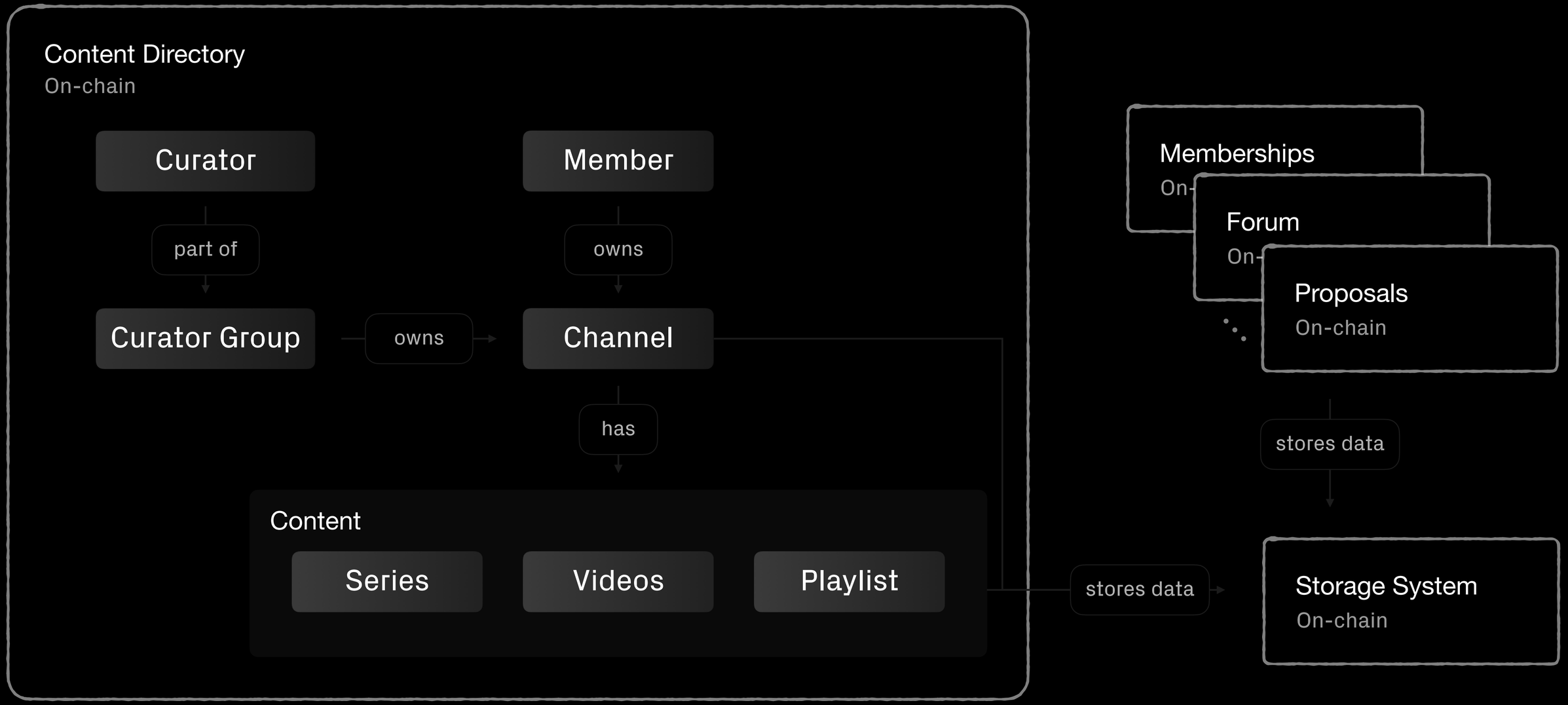
Introduction

The content directory is an on-chain index of all content and metadata, and related conce
channels and playlists. The data model was conceived to facilitate publishing and curati
inherent to the platform within the constraints of the blockchain environment, as a result
following four major design traits:

- **Versioned**: Entities can have multiple simultaneous representations, known as *sche*
flexibility of multiple representations per entity allows us to avoid having to migrate
directory whenever we want to alter the representation of some category. For examp
metadata field to videos. This is extremely valuable, because such a migration - ever
not only require freezing substantial portions of on-chain state across multiple block
incur the same high-security risk as doing any runtime upgrade at all. As a result, it
infrequently, and only after substantial community coordination. This is extremely co
platform because the content directory has to be able to evolve quickly to allow new
features.
- **Structured**: Representations are structured. This structuring not only allows for integ
on the content but also is the foundation for having permissions in the context of a
scenario. Without structure, you cannot selectively give different actors to write acce
properties.
- **Linked**: Representations are linked allowing us to build realistic domain models whe
reused in different relationships.
- **Owned**: Entities are owned, initially be the actor responsible for creating them, and t
permission model is aware of this ownership status.
- **Bespoke write access model**: There is a write access model that attempts to captur
access control rules one would want to enforce over this type of data model. The m
accommodate an evolving set of subject matter domain concepts without being upo
very assumption mentioned prior. As a result, it has a bespoke structure that attemp
balance between simplicity and expressivity.

Working Group

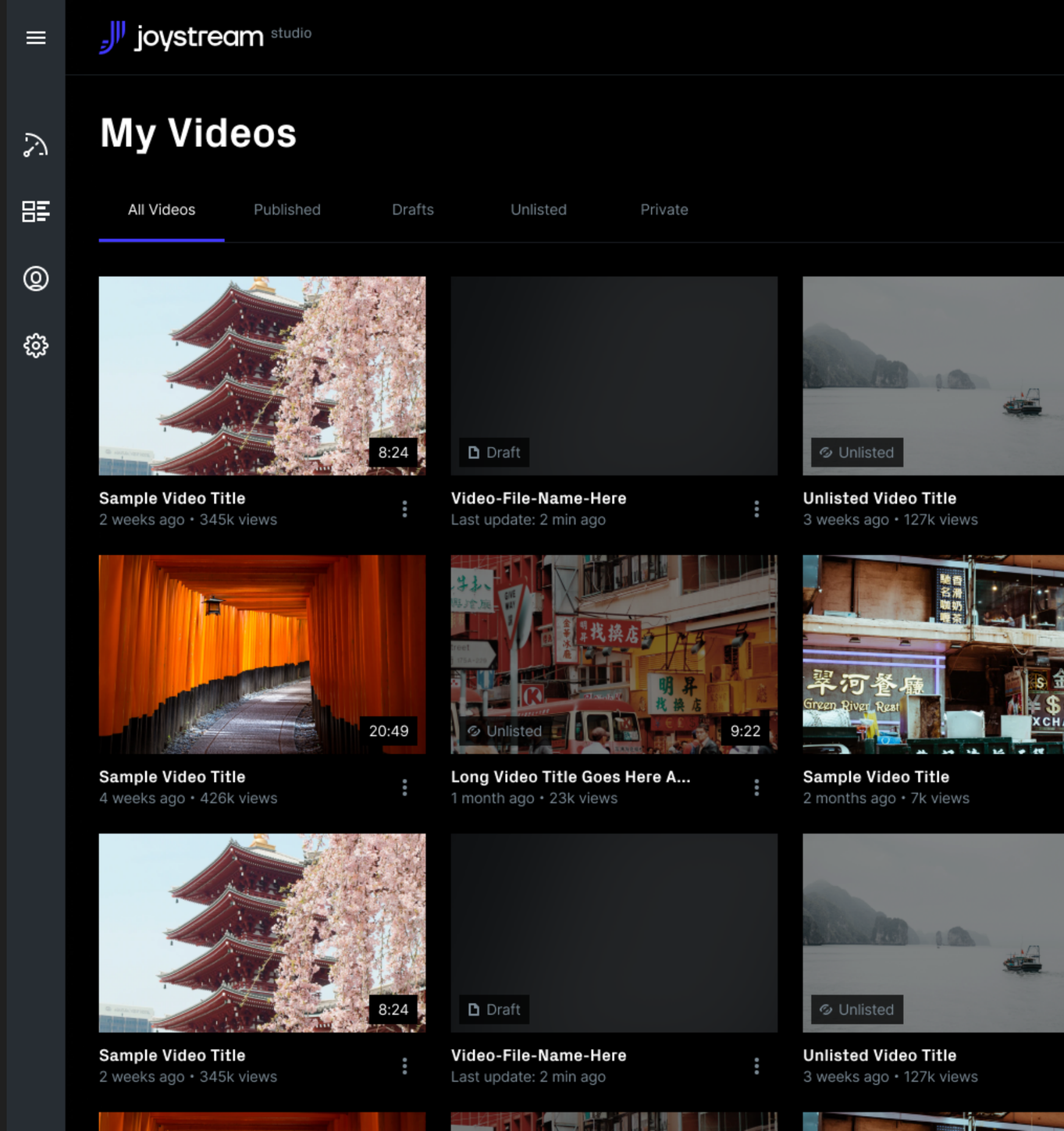
The content directory subsystem has a working group. The purpose of the group is to all


Content Directory
On-chain
Curator
Member
part of
owns
Curator Group
owns
Channel
has
Content
Series
Videos
Playlist
stores data
Storage System
On-chain
Memberships
On-
Forum
On-
Proposals
On-chain
stores data

2

Atlas Studio

Sign-up, create and manage channels, publish and manage videos.



joystream studio

Wild Crypto Fan16
Garry Covin

Videos
Channel
Uploads

Wild Crypto Fan16

420 Followers

Description

Write channel description here

0 / 1200

Channel Language

Channel language is the main language the content you publish on your channel.

Choose language

Publicness

Channel language is the main language the content you publish on your channel. We use it to provide users feed they look for. This

joystream studio

Wild Crypto Fan16
Member name <150px

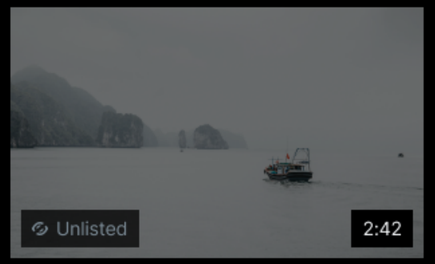
Videos
Channel
Uploads

My Videos

All Videos | Published | Drafts | Unlisted

Default Video Title
2 weeks ago • 345k views

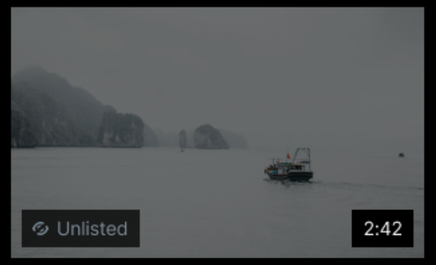
Default Video Title
2 weeks ago • 345k views



Unlisted Video Title
3 weeks ago • 127k views

Default Video Title
2 weeks ago • 345k views

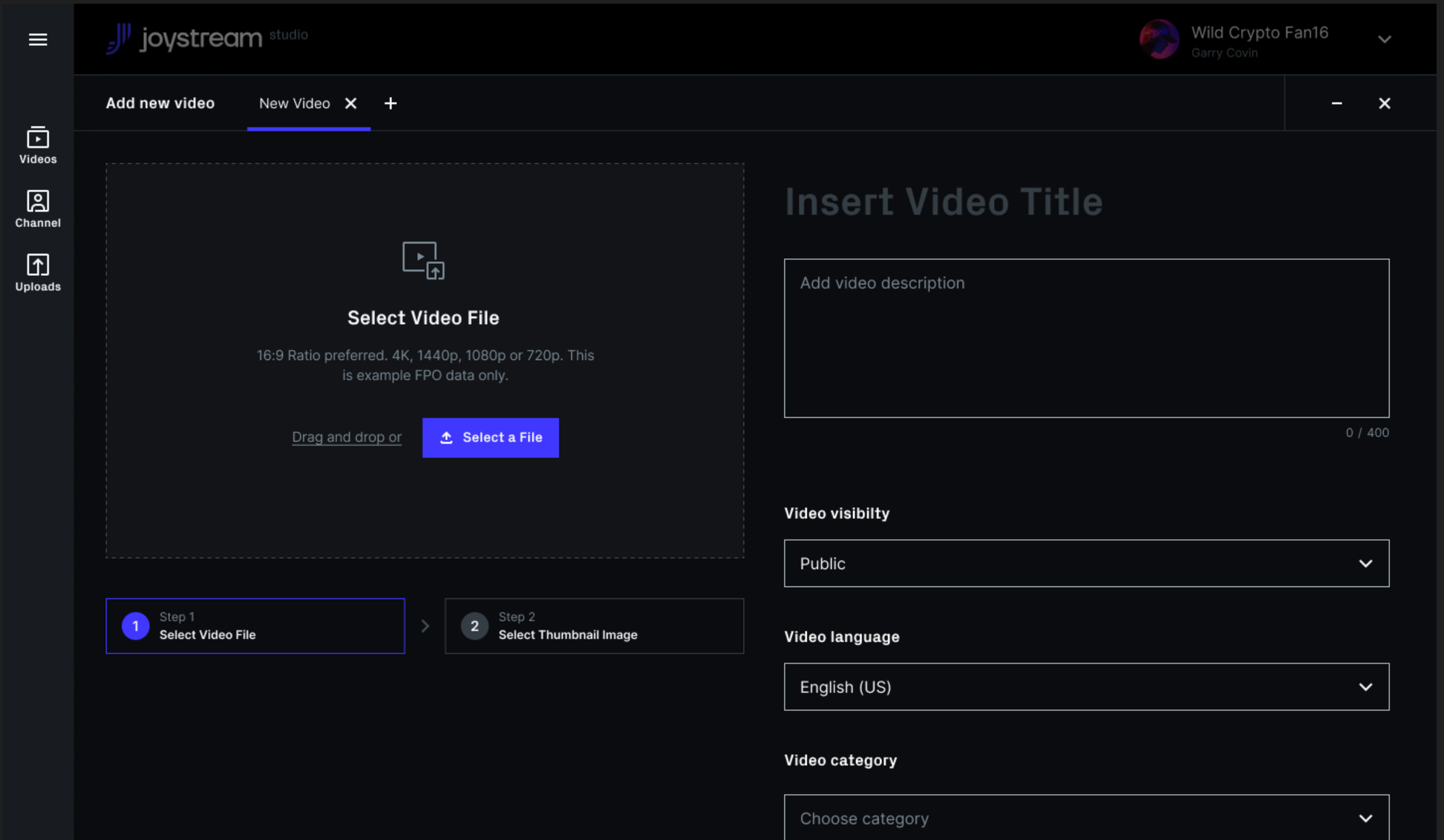
Default Video Title
2 weeks ago • 345k views

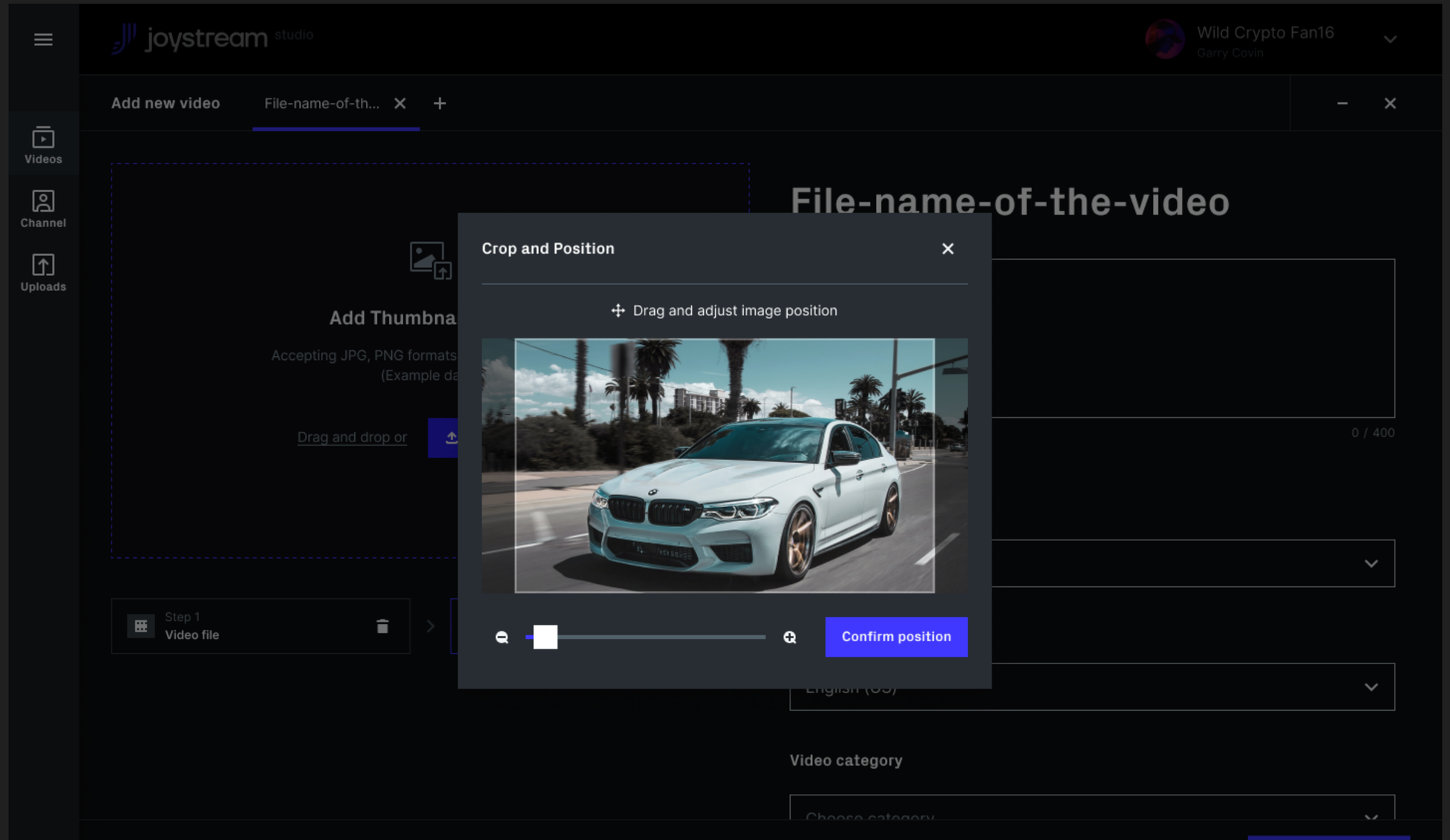


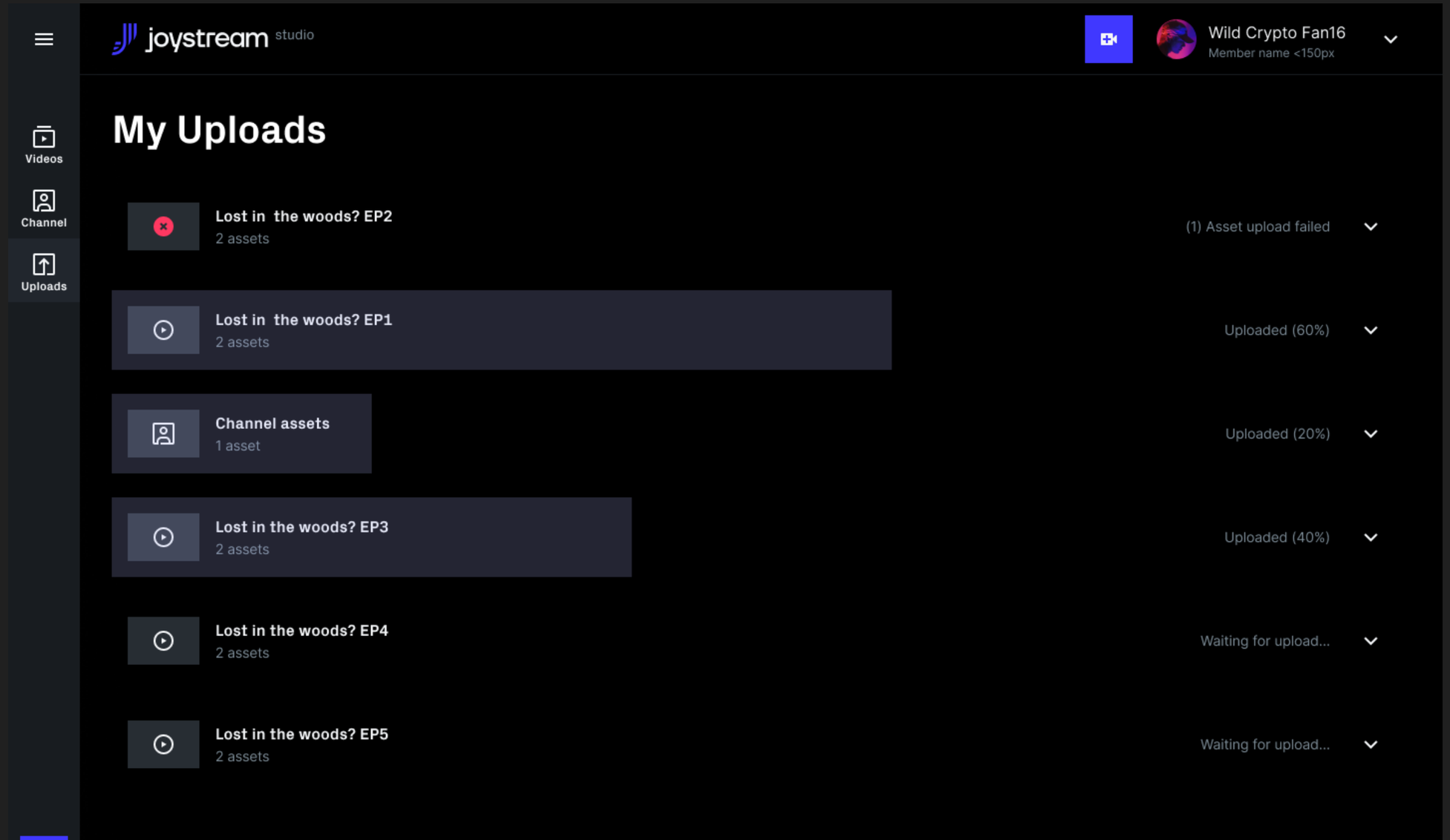
Unlisted Video Title
3 weeks ago • 127k views

Default Video Title
2 weeks ago • 345k views

Default Video Title
2 weeks ago • 345k views


joystream studio
Wild Crypto Fan16
Garry Covin
Add new video
New Video
Videos
Channel
Uploads
Select Video File
16:9 Ratio preferred. 4K, 1440p, 1080p or 720p. This is example FPO data only.
Drag and drop or
Select a File
1
Step 1
Select Video File
2
Step 2
Select Thumbnail Image
Insert Video Title
Add video description
0 / 400
Video visibilty
Public
Video language
English (US)
Video category
Choose category


joystream studio
Wild Crypto Fan16
Garry Covin
Add new video
File-name-of-th...
Videos
Channel
Uploads
File-name-of-the-video
Add Thumbna
Accepting JPG, PNG formats
(Example da
Drag and drop or
0 / 400
Step 1
Video file
Crop and Position
Drag and adjust image position
Confirm position
Video category

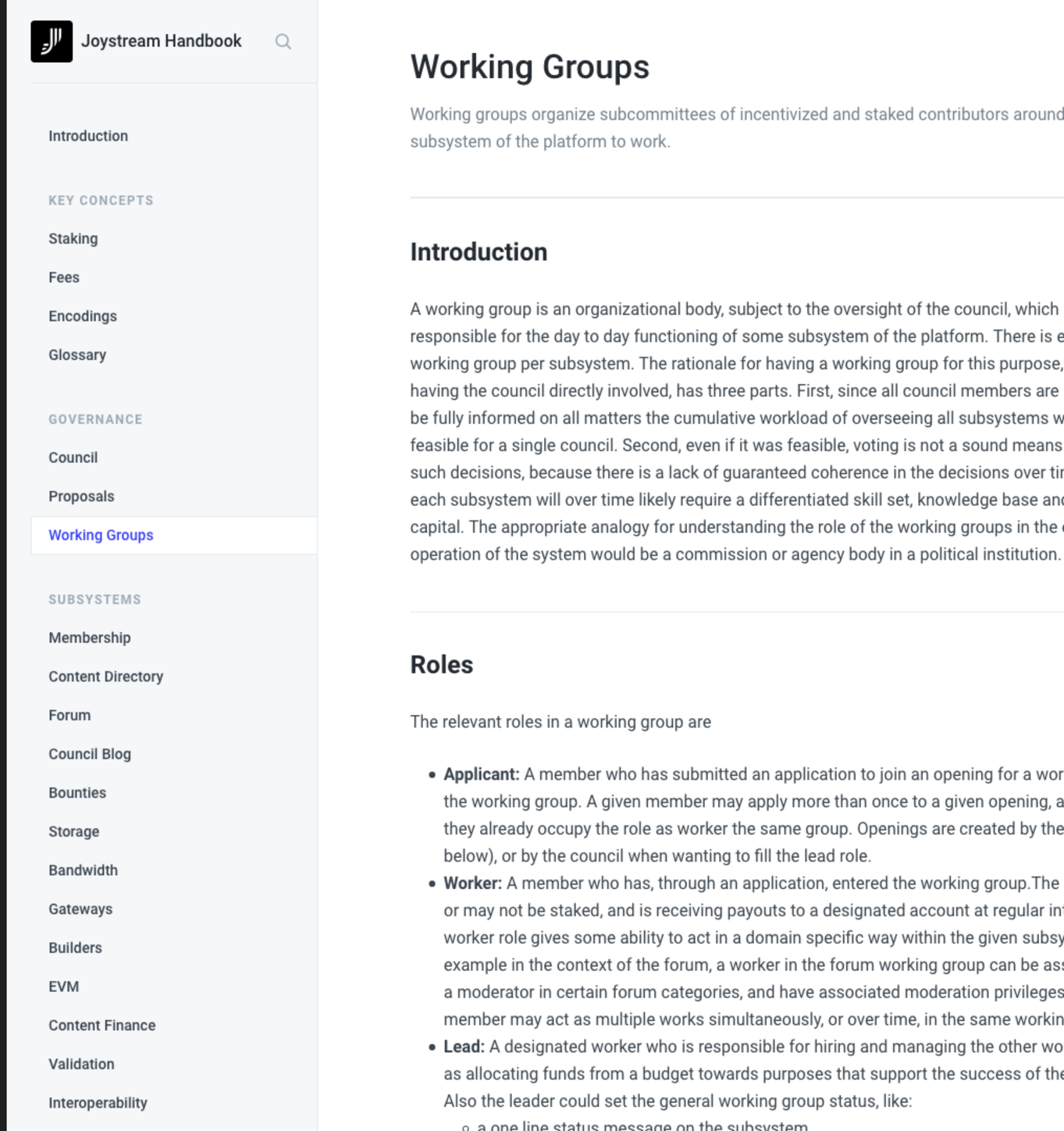

joystream studio
Wild Crypto Fan16
Member name <150px
Videos
Channel
Uploads
My Uploads
Lost in the woods? EP2
2 assets
(1) Asset upload failed
Lost in the woods? EP1
2 assets
Uploaded (60%)
Channel assets
1 asset
Uploaded (20%)
Lost in the woods? EP3
2 assets
Uploaded (40%)
Lost in the woods? EP4
2 assets
Waiting for upload...
Lost in the woods? EP5
2 assets
Waiting for upload...

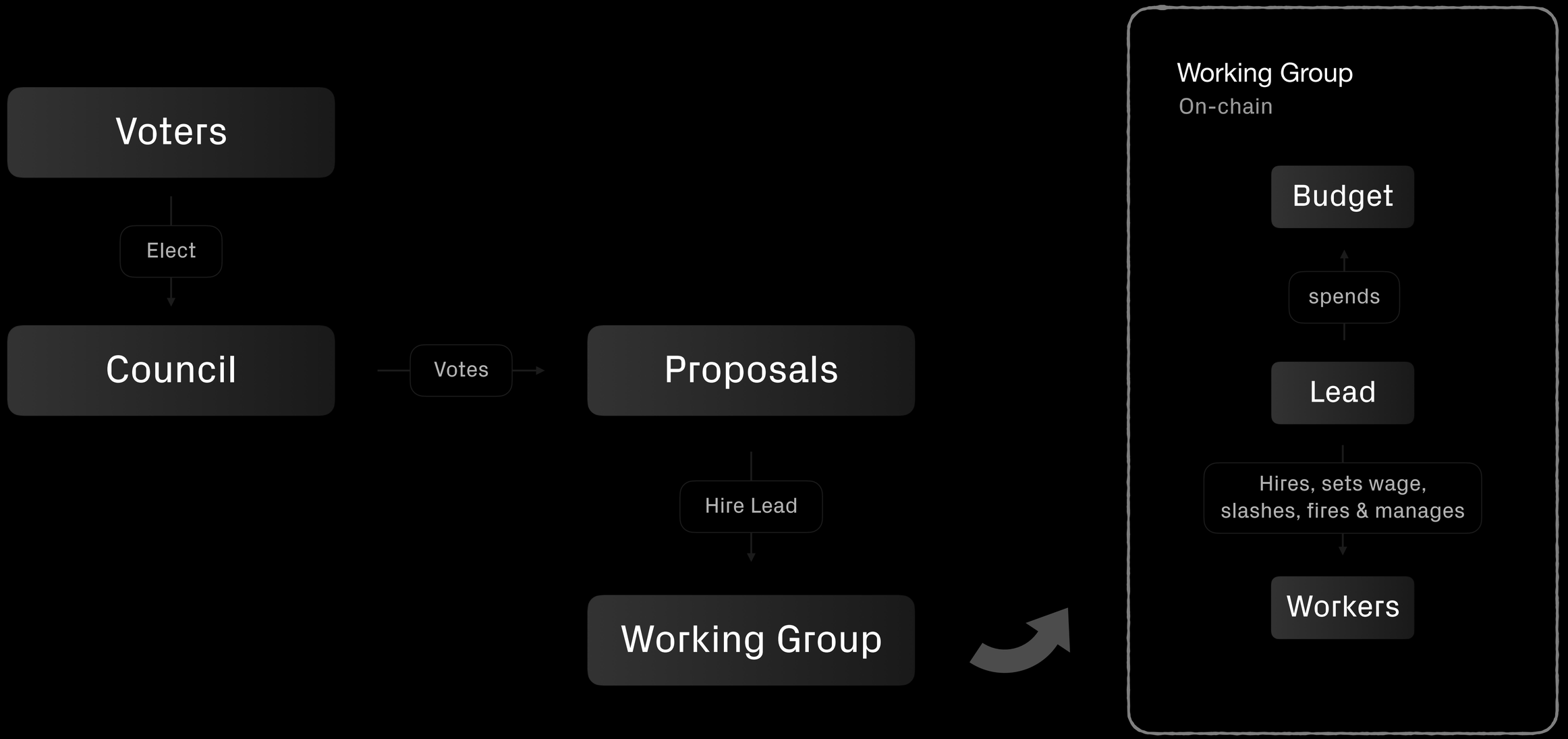
3

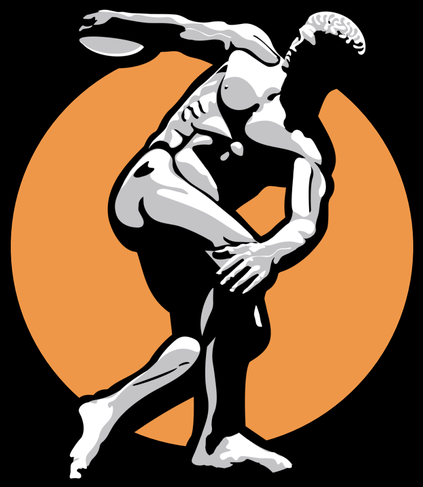
Operations Working Group

Formalising entry, periodic rewards and staking for off-chain roles such as

- Developers
- Managers
- Marketers




Voters
Elect
Council
Votes
Proposals
Hire Lead
Working Group
Working Group
On-chain
Budget
spends
Lead
Hires, sets wage,
slashes, fires & manages
Workers



Olympia Network

Goals

1 **New benchmarked & audited runtime.**

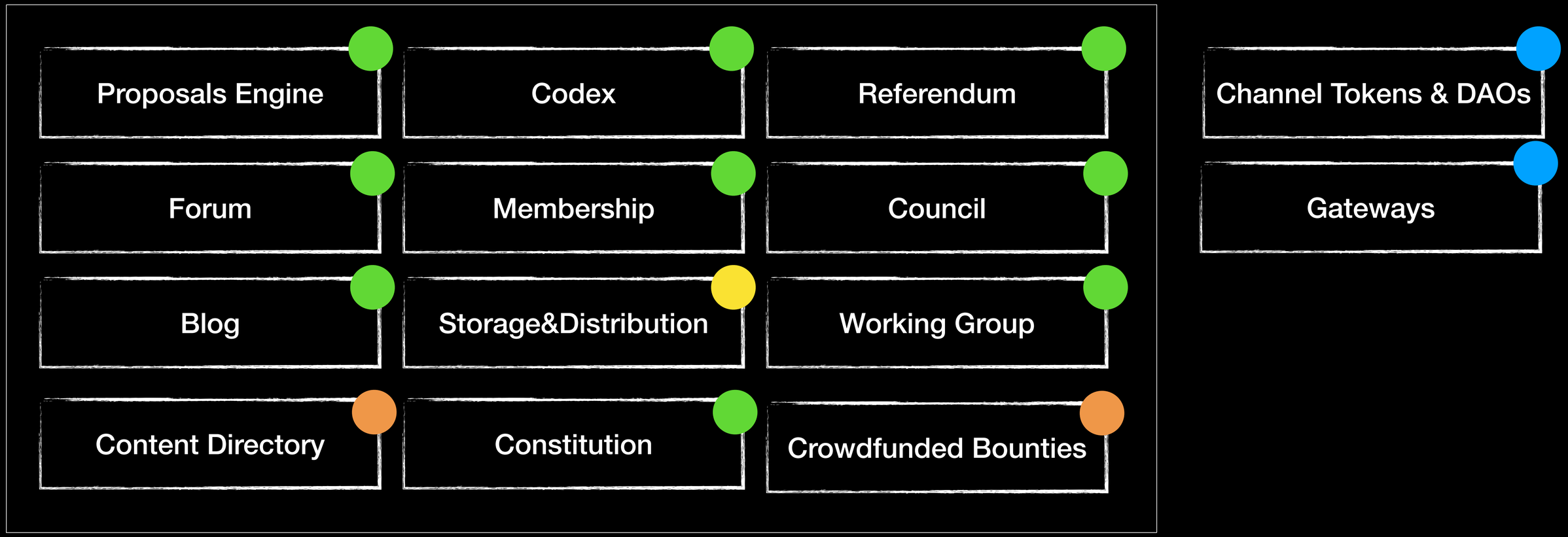
Working groups, council, elections, staking, council, forum, constitution, proposals, membership system.

2 **Pioneer v2.**

Redesigned and built from scratch, using Hydra.

Live ~Q2/Q3 2021

1 Mainnet Runtime



Benchmarked & audited.

Implemented, not audited.

Under development.

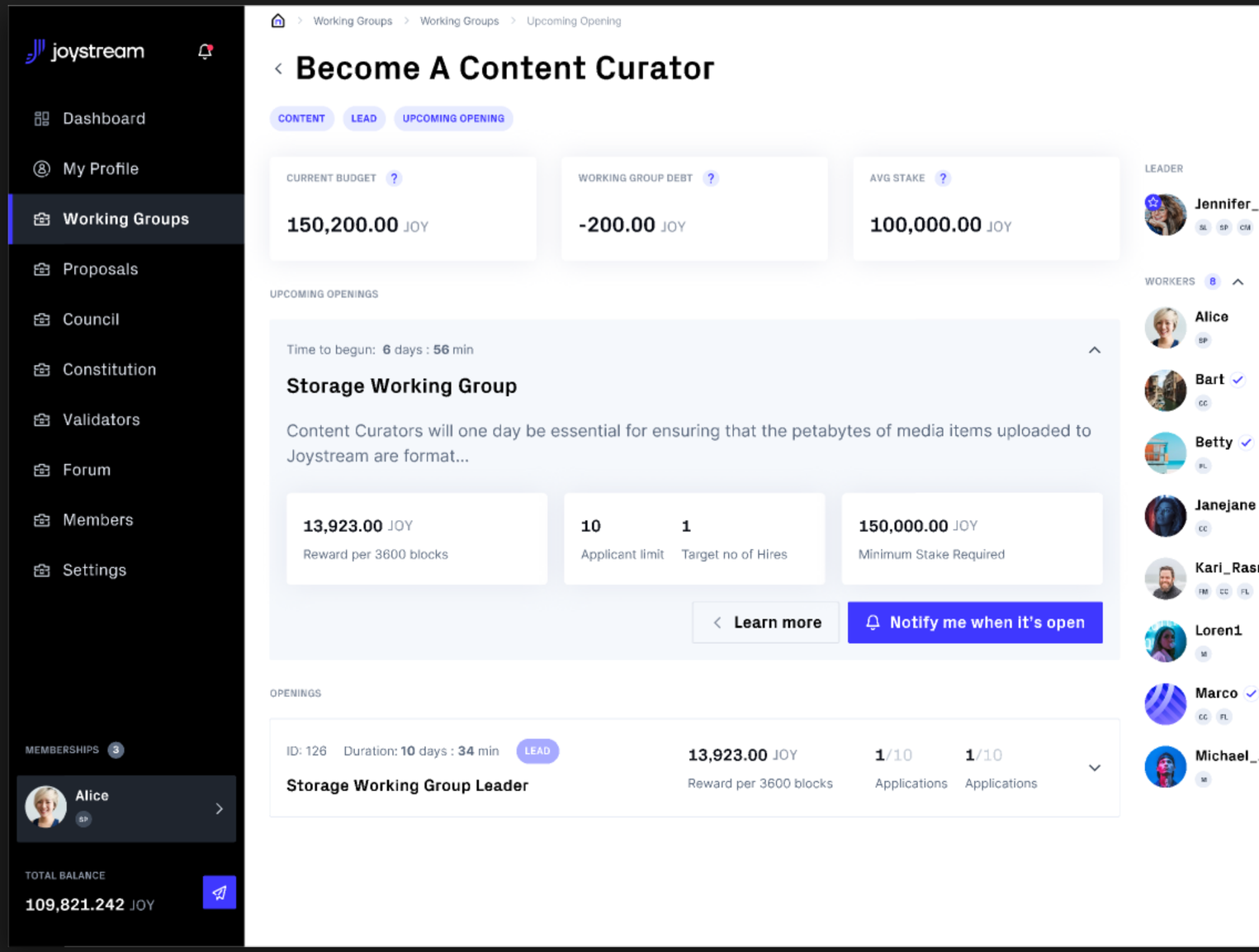
Not started development.

Security Research Labs

2

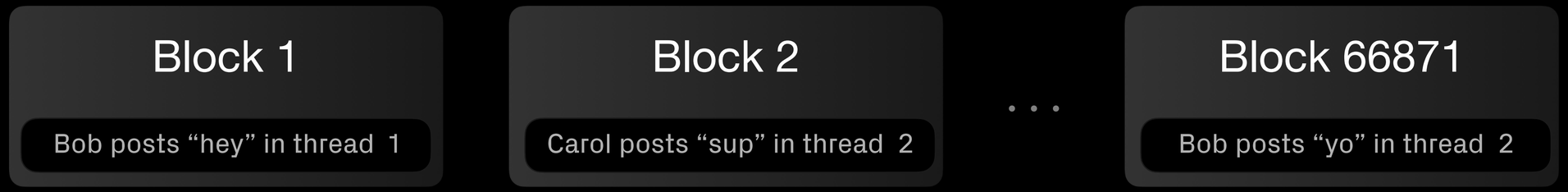
Pioneer v2

Pioneer is the app where token holders vote, stake, work, communicate and collaborate on the Joystream blockchain.



Hydra v3

Example: Blogging Blockchain



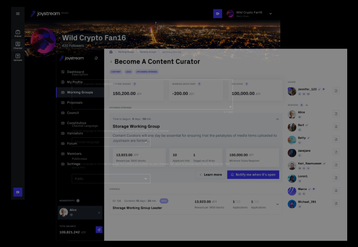
subsocial

Naive Architecture

Application

Queries

Full Node



Infeasible Queries

- How many posts did Carol post in May?
- What are all of Bobs posts, order by date, where he talks about “Trump”?
- What are posts 400-600 in thread nr. 2 ?

joystream

Product Manifesto Roles Founding Members Community

Hydra - A Substrate query node framework

Inspired by The Graph, it gives a smooth way to provide powerful GraphQL queries to app developers over your Substrate blockchain state and history.

kusama
@kusamanetwork

The judges have finished evaluating and the winners o
#Hackusama are:

For the Open Hack category:

🥇 In first place: Hydra, a GraphQL query node builder
@substrate_io chains built by Dmitry ZHELEZOV and
@metmirr.

Oversett tweeten

Hydra Hackusama Demo
This is a video tutorial for Hydra CLI, a Substrate query no
framework inspired by TheGraph.Install: ...
youtube.com

Explorer Blog Docs Security Jobs Network Join our Discord Sign In

APIs for a vibrant decentralized future

The Graph is an indexing protocol for querying networks like Ethereum and IPFS. Anyone can build and publish open APIs, called subgraphs, making data easily accessible.

Explore Subgraphs


Data Model
Event/Tx Mappings

Hydra Node

API Server

Query Database

Processor

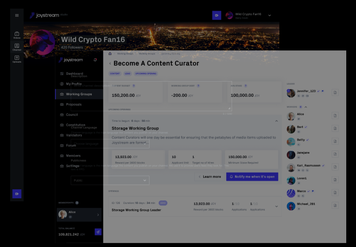
Indexer Database

Queries

Reads

Application

Full Node



New Specifications

5

v2 Storage & Distribution System

Applications with a business model.

- Separate storage & distributor roles.
- Council, DAOs and members can store.
- Partial replication.
- Reclaiming space.
- Flexible distribution policy.

v2 Storage and Distribution System Specification #2224

Open **bedeho** opened this issue 27 days ago · 1 comment

bedeho commented 27 days ago — Member

Background

The current Joystream network, as of the `Sumer` release, has an extremely limited system for storing and distributing data, both in terms of functionality and ability of the system to accommodate any kind of even limited scale of utilisation. This specification is intended to substantially improve upon these limitation by settling the organisation and function of the system at a level suitable for mainnet purposes. Importantly, this specification should be read in the context of the Gateway specification #2089, which is complementary in that it outlines an incentive model for the vast majority of expected load on distributors.

Major Changes

The overall design philosophy of the system remains the same as before in the following respects

- Permissioned entry of of staked service providers.
- Discretionary slashing, not bound by cryptographic evidence.
- Chain holds index of data, service provider obligations and rations utilisation of the system.
- Publishers are not charged for the distribution cost of data.
- Single chain selected upload host for each data object.

However the following major changes are introduced

- Distinct roles for storage and distributing data.
- Storage system with redundancy and only partial replication in nodes.
- Distribution system with flexible policy space, allowing for CDN like organisation.
- Efficient deletion and ownership transfer of groups of data objects.
- On-chain host resolution metadata.
- Distributors are incentivised in the Gateway system: #2089

Architecture

Working Groups and Roles

There are two working groups, the *storage working group* and the *distribution working groups*, each with its own set of workers and two separate leads, called the *storage lead* and *distribution lead* respectively. The workers in the storage working group are called *storage providers*, and operate dedicated nodes for this purpose, called *storage nodes*. Likewise for the distribution group there is *distribution providers* and *distribution nodes*.

Gateways

Applications with a business model.

- Serve viewers on the web & app stores.
- Monetise by adds or charging your users.
- Pay for the bandwidth & content royalties for your users.

Gateway Specification #2089

Open **bedeho** opened this issue 8 days ago · 0 comments

bedeho commented 8 days ago — Member

Document

This is a **very rough** specification of how gateways will work in Joystream, in particular the APIs, which hopefully should be conceptually clear enough to warrant implementation. It should be noted that this is a specification only because of its relatively high level of specificity, this should however not be mistaken to imply that these are immutable design and implementation decisions. This should should be taken as a concrete starting point for an overall approach, where many details may need to change.

Background

1. **Advertising is indispensable near term:** Reaching a broad audience of content consumers in the near term, in a way which is economically sustainable, means that the system must have ways of capturing value that does not depend on consumers having a wallet and tokens. This value capture is required to cover the cost of the computing infrastructure, such as storage and queries, but also to entice content creators to publish and create content on the system. The only mechanism that satisfies this goal is some kind of advertising system.
2. **Advertising requires Sybil prevention:** The most important technical and policy requirement of any advertising system is the ability to correctly distinguish genuine from fraudulent impressions.
3. **Sybil prevention depends on app delivery:** Sybil prevention is only really practical when the platform controls the delivery of the user facing experience. Such control enables detection and prevention of abusive user practices, for example by using techniques such as Captchas or email registration. As an example of the most blatant attack possible, anyone running a Joystream application could simply replace all the advertising endogenous to the Joystream system with their own integrations. A milder attack is just normal client side ad-blocking in browsers.
4. **Blockchains are isolated from web 2.0 app delivery:** Asserting control over web 2.0 assets for distributing apps, such as
   - app store entries (phones, TVs, etc.)
   - ICANN domain names
   - desktop app certificates
     is a major barrier for Blockchain systems. Alternative systems are being deployed, their adoption is likely to take a long time.
5. **Conclusion:** Delivery of apps must be a role for a sustainable way for Joystream to reach a broad consumer audience in a sustainable way in the near term.

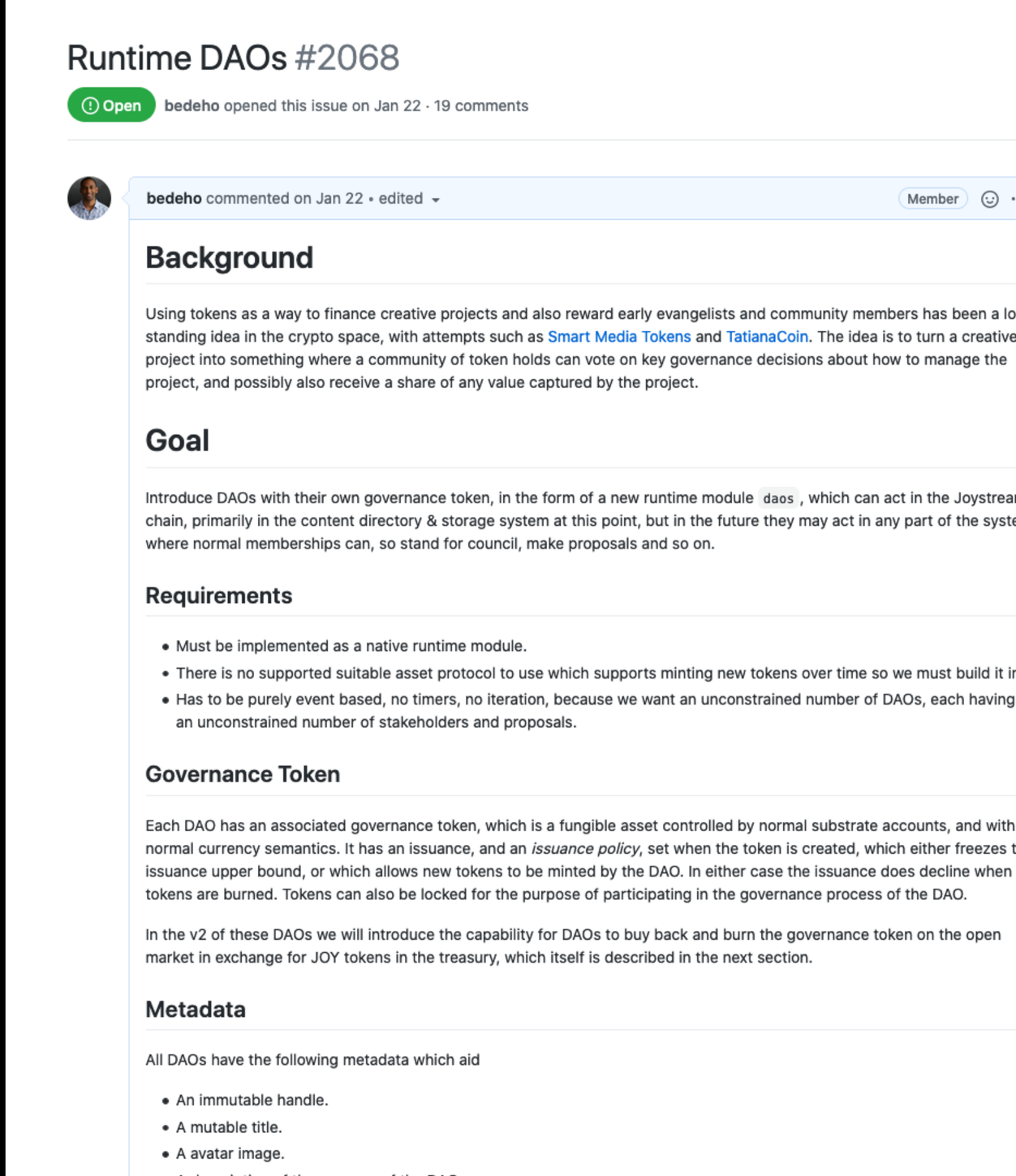
Idea

The idea for how to solve problem described is to couple advertising and app-delivery at the same layer, called a gateway, exogenous to the core Joystream protocol. This layer sits on top of the core services provided by the protocol, including content creation, and pays for utilisation of these services on behalf of its screened users. This incentivises gateway operator to properly internalise the cost of Sybils, and thus invest in screening them properly, because the gateway has to pay for the

Channel Tokens & DAOs

Creator tokens for fundraising & revenue splits.

- Issue a token for your channel.
- Raise funds for your channel.
- Give governance and revenue split to token holders.



Runtime DAOs #2068

Open bedeho opened this issue on Jan 22 · 19 comments

bedeho commented on Jan 22 • edited Member

Background

Using tokens as a way to finance creative projects and also reward early evangelists and community members has been a lo
standing idea in the crypto space, with attempts such as Smart Media Tokens and TatianaCoin. The idea is to turn a creative
project into something where a community of token holds can vote on key governance decisions about how to manage the
project, and possibly also receive a share of any value captured by the project.

Goal

Introduce DAOs with their own governance token, in the form of a new runtime module `daos`, which can act in the Joystrea
chain, primarily in the content directory & storage system at this point, but in the future they may act in any part of the syste
where normal memberships can, so stand for council, make proposals and so on.

Requirements

- Must be implemented as a native runtime module.
- There is no supported suitable asset protocol to use which supports minting new tokens over time so we must build it ir
- Has to be purely event based, no timers, no iteration, because we want an unconstrained number of DAOs, each having
  an unconstrained number of stakeholders and proposals.

Governance Token

Each DAO has an associated governance token, which is a fungible asset controlled by normal substrate accounts, and with
normal currency semantics. It has an issuance, and an *issuance policy*, set when the token is created, which either freezes t
issuance upper bound, or which allows new tokens to be minted by the DAO. In either case the issuance does decline when
tokens are burned. Tokens can also be locked for the purpose of participating in the governance process of the DAO.

In the v2 of these DAOs we will introduce the capability for DAOs to buy back and burn the governance token on the open
market in exchange for JOY tokens in the treasury, which itself is described in the next section.

Metadata

All DAOs have the following metadata which aid

- An immutable handle.
- A mutable title.
- A avatar image.

Crowdfunded Bounties

Incentivised independent funding for community goods.

- Anyone can create and fund.
- Creating & contributing is incentivised.
- Creator oracle judges outcome.

Crowdfunded Bounties #1998

Open **bedeho** opened this issue 24 days ago · 1 comment

bedeho commented 24 days ago — Member

Background

Currently, the only way to fund the production of goods that create benefits to a broad set of platform participants is through a financing proposal or discretionary spending by a working group lead out of the group budget. These processes incur the transaction costs of beneficiaries having to convince a number of external decision-makers, such as a council financing quorum, that this is a good idea. For smaller initiatives that ideally should start and finish sooner, or where they depend on knowledge or insight that is not as broadly shared, these processes become too costly.

Proposal

Crowd funded bounties which allows a member, or the council, to crowd fund work on projects with a public benefit. The funding mechanism for the bounties attempts to facliate two forms of crowd funding:

- **Assurance Contract**: It only triggers if some minimal quantity is raised, otherwise all funds are returned. Described here https://en.wikipedia.org/wiki/Assurance_contract.
- **Dominant Assurance Contract**: The proposer is the only person who can submit a bounty solution - and thus claim the raised funds, however, the proposer must put up a pool of funds which will be distributed among all third parties that fund the pool in the event that the minimum quantity is not raised. Described further here http://mason.gmu.edu/~atabarro/PrivateProvision.pdf.

In either case, an oracle is required to judge how much of the collected funds should be paid to any given contributor of work on the bounty, and this will either be a pre-specified member, or the council iself.

Bounty Creation

Any member or the council can create a bounty by providing the following information.

- **Metadata:** A standardised structure document describing user facing information, for example a title, amount requested, deliverable, discovery metadata, link to forum etc. Is not stored in storage, chain only sees raw extrinsic payload blob, like rationales before.
- **Oracle:** Origin that will select winner(s), is either a given member or the council.
- **Cherry**: An mount of funding, possibly 0, provided by the creator which will be split among all other contributors should the min funding bound not be reached. If reached, cherry is returned to the creator. When council is creating bounty, this comes out of their budget, when a member does it, it comes from an account.
- **Screened Entrants:** The set of members who are allowed to submit their work, if not set, then it is open. Main use case for this is to model dominant assurance contract where member sets contribution cherry and him/herself sa only elidable worker.
- **Minimum Amount:** The minimum total quantity of funds, possibly 0, required for the bounty to become available for people to work on

Community

6

Founding Members Inducted

Current founding members 5

	Total score	Tokens allocated / projected
tomato	10000	1.00% / 9.52%
nexusfallout	0	0.20% / 0.00%
enjoythefood	0	0.20% / 0.00%
freakstatic	5750	0.30% / 5.48%
l1dev	0	0.20% / 0.00%

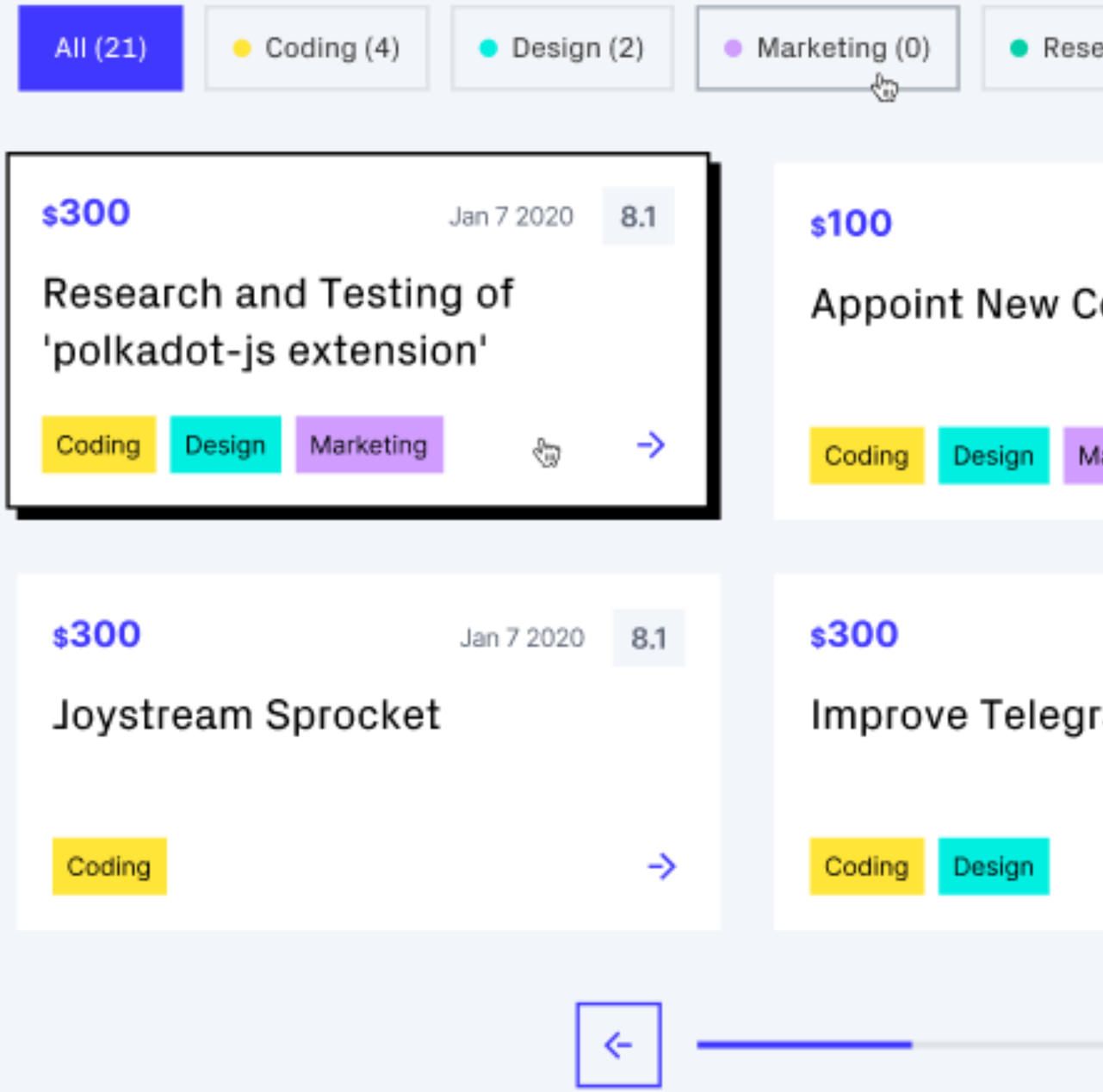
Leaderboard

Founding Members | Non-Founding Members

	Direct Score	Referral Score	Total Score
@cheomsk Member: #552	5900	0	5900
@lopegor Member: #1369	5100	358	5458
@narniec Member: #585	3000	1523	4523
@seainvestor Member: #684	2150	2238	4388
@sasha Member: #1015	3300	0	3300
@joystreamenthusiast Member: #555	2350	125	2475
@fiery Member: #1676	2300	0	2300
@xandrell Member: #867	2150	0	2150

Bounty Program

A way for community members to earn cash and tokens solving valuable community tasks.



Roles

Content Creator

19 run this role on Antioch

Content Curator

1 run this role on Antioch

Validator

18 run this role on Antioch

Council Member

12 run this role on Antioch

Storage Provider

2 run this role on Antioch