CR Community Proposal Voting App

Ben - Shijun - 2021.10.21

2021.10.22 - **v2**: added project submission flow, financial terms
2021.10.21 - **v1**: initial content

Objective

- Request a pool of ELA from CR to vote for projects (ex: 100k ELA), by the community, not by CR members
- Proposals can be made and voted by the community
- Max 10k ELA allocated for a project
- Community votes for 1, or maybe max 2 projects per month, depending on projects
- 1 community member = 1 vote

Considerations

- This specification is for a Proof of Concept only, and the product will be improved after a first launch, in case the CR council agrees with it.

- In case of bug (vote cheats, etc), the centralized owners of the ELA can decide to not distribute project funds)

Definitions

- **DID**: Decentralized identity
- **Credential**: piece of data inside a user’s DID
- **EE**: Elastos Essentials
- **TG**: Telegram
- **DB**: Database

Use case: become a voter

- User sends his DID on TG manually to an admin

- Admin uses vote backend to save the TG/DID map -> saved to voting DB

- When the admin fills the TG+DID in the backend app, this generates a random 4 digits ID, used to verify that the TG user really owns the DID.

- Admin gives the ID to the user on TG.

- User enters the voting app, signs in with DID. The first time, the web app will ask him to confirm the confirmation number.

- User should type the 4 digits. The backend can activate the DID for voting only after a matching 4 digits is entered.

Use case: submit a project

- Project owner opens the web apps and clicks a “manage projects” button
- On the manage project screen, he can see a list of previously created proposals
- He can click ‘submit new project’, or delete an on going project
- For new projects, he provides a title, a link to an existing CR suggestion page. Submitter must sign in with DID and must provide his name + email.
- After submission, an admin reviews the request, and can approve or reject from the back office.
- Once validated, a submitted project is open for voting until the next 15th of a month. Every 15th of each month, the project with the most votes gets the funds.
- If a submitted suggestion is transformed into a proposal on the CR website, it gets cancelled on the community voting app because there can’t be double funding.

Use case: vote

- User opens the web app in EE or in the desktop browser. Sign in with wallet connect or injected DID connector

- User sees the proposals and selects one

- User vote yes or no. and calls backend api

- Backend api checks that the DID is bound to a TG user and sends a DID sign challenge to EE

- User uses the DID sign screen in EE to sign server data, and data is returned

- Voting backend verifies the signed data and. saves the proposal id + user DID into DB

Financial items

- This project is for now a PoC. A proposal will be submitted to CR to get 100k ELA.
- If the proposals passes, the 100k ELA are managed by a multi-signature wallet held by the 12 CR council members. The 100k ELA are send to this wallet at once once the proposal is passed.
- When projects are elected on this community voting app, a manual process initiated by the voting project admin starts, and 8 of 12 CR council members must send the funds.
- CR council members should not sign the payment transaction according to whether they support the project or not. They must sign because the decision is not theirs.

Back-office features

- Technology: React (front end) + NodeJS, MongoDB, DID JS SDK (back end)
- Authentication
  - Harcoded list of admin DIDs in the backend
  - Admins sign in with DID (connectivity SDK) and get an access token (JWT)
  - Access token is used for all further API calls
- Screens - **all screens and related apis protected by admin authentication token**
  - Members list
    - Community members list (list of DID / TG name) + button to add a new member + button to delete each member
  - Projects list
    - List of all projects with status (to be review (by admin), active (votes open), finished (votes closed)) + ability to view project details + accept or reject a project that was just submitted.
  - New member
    - Input fields: TG ID + DID -> validate -> show a 4 digits confirmation code (pre-registration saved to DB)

Public features

- Technology: React
- Authentication
    - DID, connectivity SDK
    - After sign in, user gets a session authentication token (JWT), used for other screens/apis
- Screens - **all apis protected by user authentication token**
    - List of projectsthat can be voted for + start/end date, (number of votes yes/no) + “approved|rejected” status + buttons to vote yes or no
        - When a user clicks a vote, if not signed in, this prompts the DID sign in first
    - TG verification screen
        - The first time after sign in, if the 4 digits TG confirmation code has not been validated, user must input the 4 digits given by the TG admin. If the code matches the database code, then the DID becomes active and allowed to vote.

Testing

- All operations should work both from a desktop browser (through Wallet connect) and from inside EE (injected).

Ownership & Deployment

- As Trinity Tech is mandated to implement this first version of the system, the web app will be deployed to the trinity-tech.io server.