Cheesewheel.xyz

Benjamin Boule, Eugene Njinkeu,
Ian Melhorn, Will Pape

AUDIUS

Overview

Cheesewheel.xyz is where Web3 meets music, social tokens, and NFTs.

A launchpad for emerging artists to get discovered and build a community using Web3 tools, powered by the CWD governance token (CheesewheelDAO).

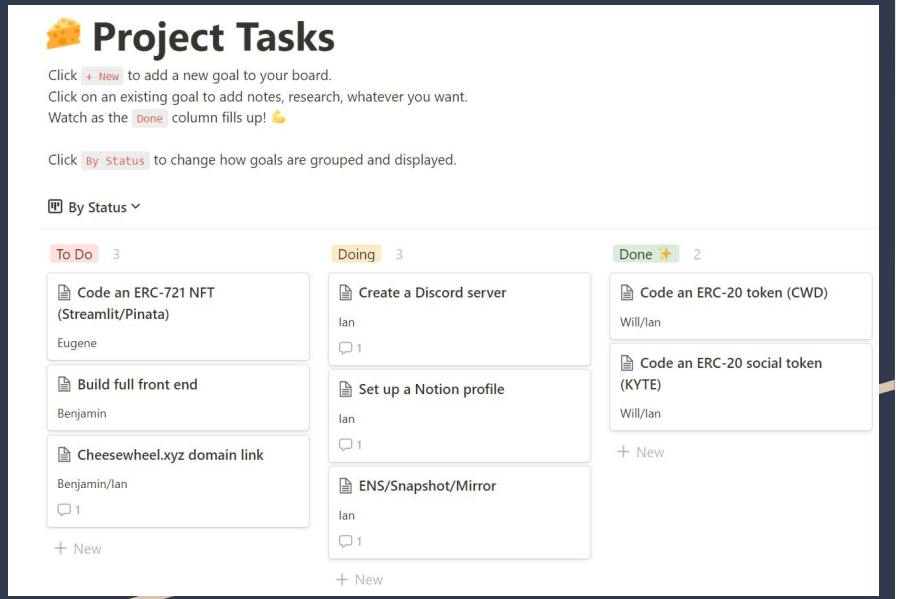
An intuitive user interface aggregating and enabling the use cases below:

- **Music:**
  - Stream artist music on Audius.
- **Social tokens:**
  - Launch ERC-20 social tokens for artists.
- **NFTs:**
  - Launch “access” NFTs for artists.
  - Sell artist merchandise in the form of NFTs.
  - Physical delivery of items after NFT purchase.

Onboard First Artist

- **Name:** Chris Kyte
- **Location:** Myrtle Beach, SC
- **Sponsors:**
  - Wise Dreamers (clothing)
  - The Private Shop (apparel, sneakers)
  - AwalWandering (jewelry)
- **Label affiliations:**
  - BackGood / 10k Projects / Dreamville
    - Trippie Redd
    - Internet Money
    - 6IX9INE
    - J.Cole
- **Audius music:** https://audius.co/cheesewheelCWR/free-bibby-prod-by-king-kevin

Project Deliverables



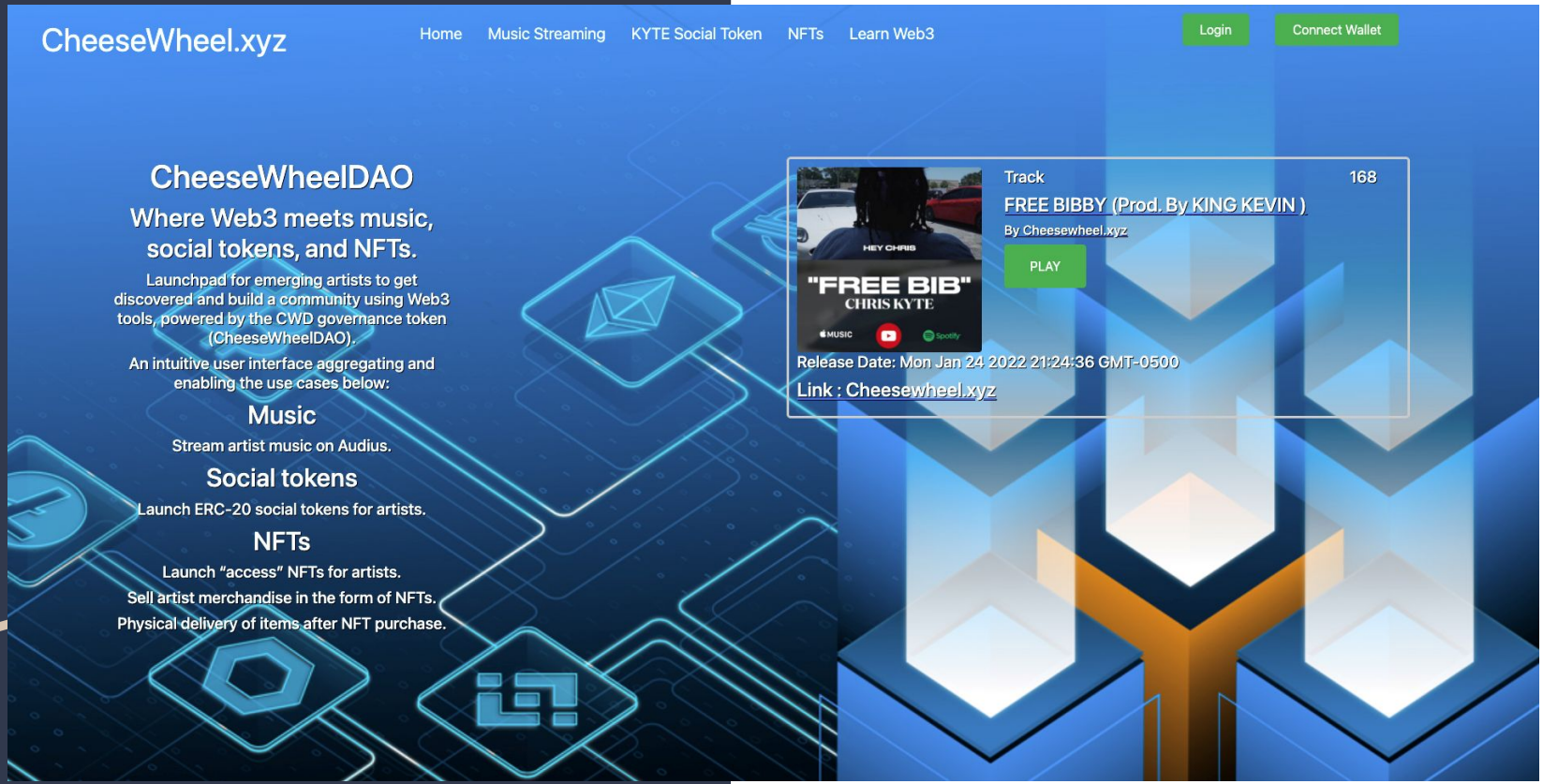
Notion

@cheesewheelDAO and cheesewheel.xyz secured

1. Create the CheesewheelDAO in Discord.
2. Set up an ENS and Snapshot profile for member governance/voting.
3. Link a Mirror blog account for project updates.
4. **Code an ERC-20 token (CWD) for platform governance.**
5. **Code an ERC-20 social token (KYTE) smart contract with crowdfunding functions for artist.**
6. **Code an ERC-721 NFT contract with the CheesewheelDAO logo to showcase on Streamlit.**
7. **Build a front-end dApp interface.**

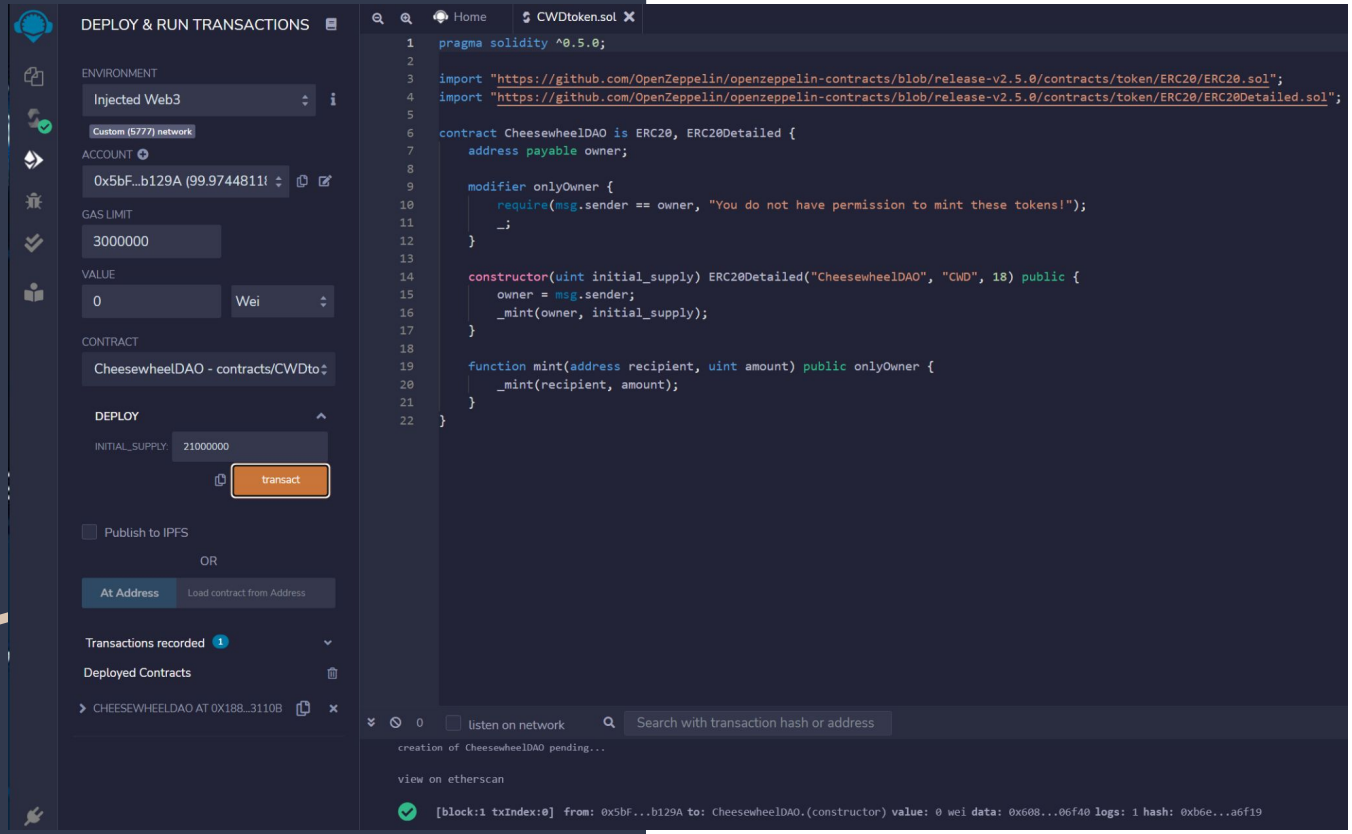
dApp Front End

- Front end website



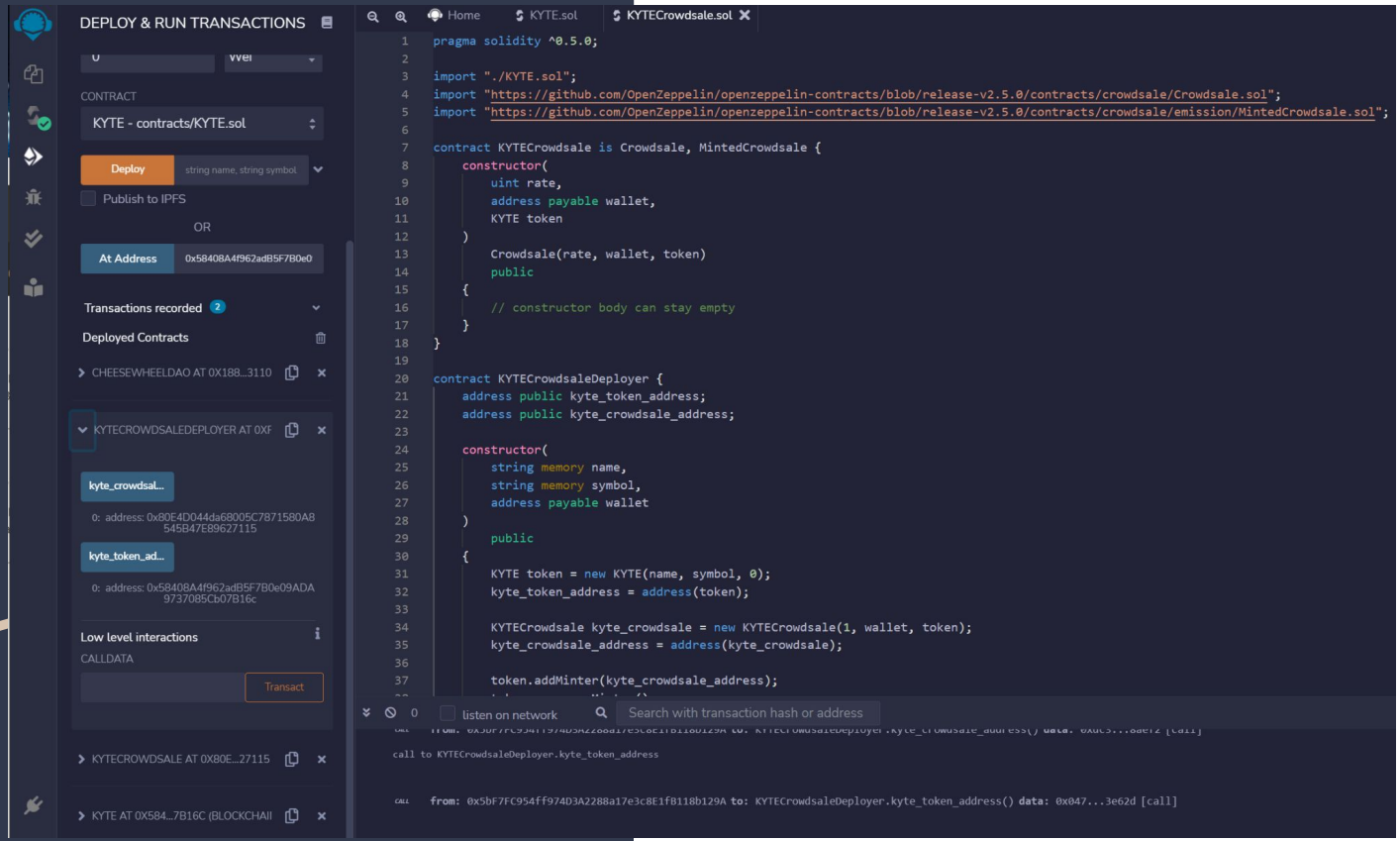
dApp Back End

- CWD ERC-20 governance token contract



dApp Back End

- KYTE ERC-20 social token crowdfunding contract



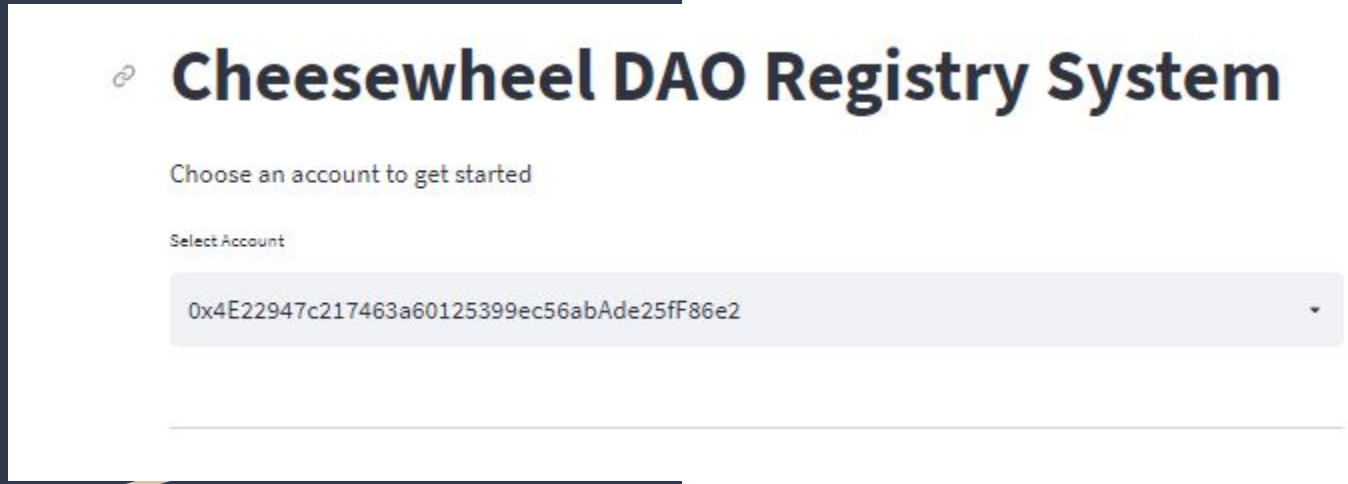
dapp Back End

- Artist ERC-721 NFT Contract
- Streamlit registry

```
pragma solidity ^0.5.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/release-v2.5.0/contracts/token/ERC721/ERC721Full.sol";

contract KYTENFT is ERC721Full {
    constructor() public ERC721Full("cheesewheel", "KYTE") {}

    struct Artwork {
        string name;
        string artist;
        uint256 appraisalValue;
    }

    mapping(uint256 => Artwork) public artCollection;

    event Appraisal(uint256 tokenId, uint256 appraisalValue, string reportURI);

    function registerArtwork(
        address owner,  string memory name,      string memory artist,       uint256 initialAppraisalValue,       string memory tokenURI    )
        public returns (uint256) {
        uint256 tokenId = totalSupply();

        _mint(owner, tokenId);
        _setTokenURI(tokenId, tokenURI);

        artCollection[tokenId] = Artwork(name, artist, initialAppraisalValue);

        return tokenId;
    }

    function newAppraisal(
        uint256 tokenId,
        uint256 newAppraisalValue,
        string memory reportURI
    ) public returns (uint256) {
        artCollection[tokenId].appraisalValue = newAppraisalValue;

        emit Appraisal(tokenId, newAppraisalValue, reportURI);
```

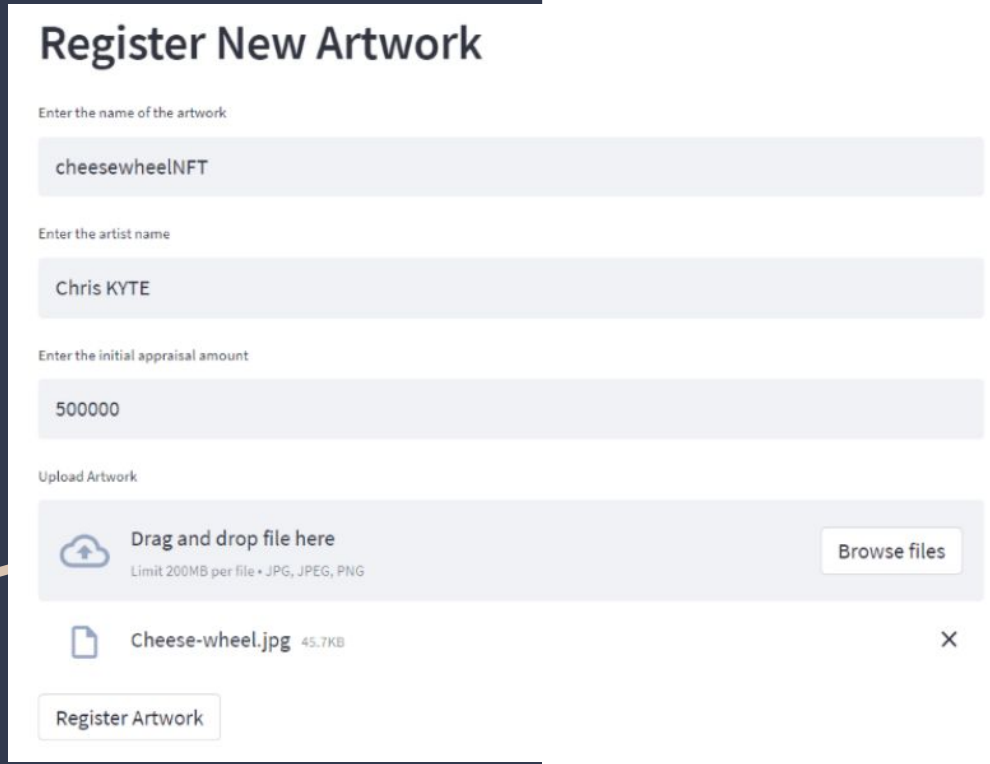
dapp Back End

- Artist ERC-721 NFT Contract
- Streamlit registry



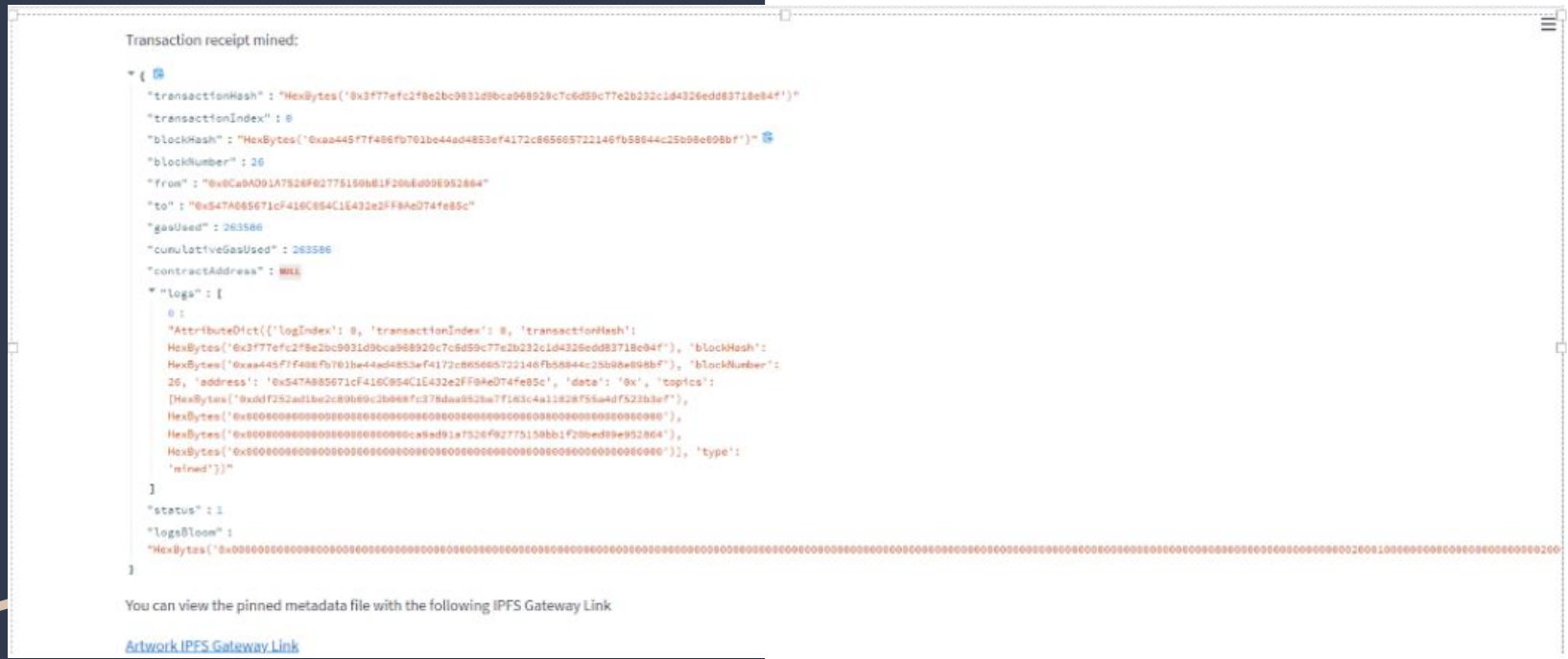
dapp Back End

- Artist ERC-721 NFT Contract
- Streamlit registry



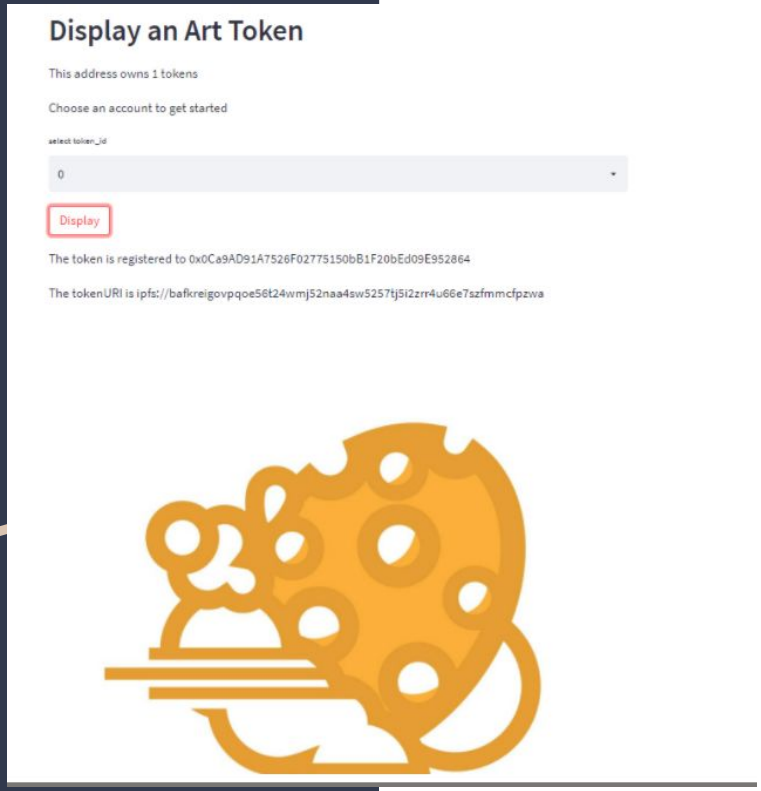
dApp Back End

- Artist ERC-721 NFT Contract
- Streamlit registry



dApp Back End

- Artist ERC-721 NFT Contract
- Streamlit registry



Questions?