DAO To fund raise PhD Thesis

Thomas Suau, Alexandro Fulmini, Thomas Forget , Maxime Desalle

June 25, 2022

Abstract

Our goal for this Hackathon is to help students to raise funds for their PhD thesis's project. To do so, we are purposing a platform where we create a crowdfunding smart contracts, a DAO to manage funds and an Ownership NFT of research outcomes.

Contents

1 Introduction

Context
On a budget of €80 bn., €27.5 bn. (34.37%) are going on scientific excellence to four programs [1]. Many students don't find funds for their PhD thesis'.
Moreover, time lost by researchers to find funds is considered by one of the main actual problem in academic research[2].
Diminishing weight of public funds in fundamental research and disinterest of private investors on theoretic research prevents development of new abstract key features for the applied research [3].

Our problem
Our problem is to finance PhD Thesis' in fields which are not financed by traditional way.

Our solution
We are purposing a platform where student with requirements [4] can fund his project.

2 Keys concepts

2.1 Requirements

Basicaly, we want the student name, advisor name, name of the sponsor's institution (with a confirmation by institution), theme of the PhD Thesis', description and funds necessary.

By using a .json file we can easily reused this structure for all fundraising.

Example :

```
{
        "student_name" : "FirstNameS LastNameS",
        "advisor_name" : "FirstNameA LastNameA",
        "institutionnal_support" : "NameInstitution",
        "proof_by_institution" : "Hash/Letter/...",

        "theme" : "ThemeOfThesis",
        "description" : "Description of the Project",
```

[1] cf. [2] Le financement de la politique européenne de la recherche
[2] cf. [3]
[3] Coronavirus stop financing in 2018 due to non-return profit
[4] A PhD Thesis' project, a support advisor and an institution (like university or laboratory)

```
        "fund_required" : xx
}
```

Firstly, when the student with all his informations is logged, he can create a fundraise backed by a smart contract[5].

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/draft-ERC721Votes.sol"
import "@openzeppelin/contracts/utils/Counters.sol";

contract NFTCrowdsale {
    // Create an `address public` variable called `kasei_token_address`.
    address public NFT_token_address;
    // Create an `address public` variable called `kasei_crowdsale_address`.
    address public NFT_crowdsale_address;

    // Value to keep track of funding progress
    uint256 public totFunded = 0;
    //
    address[] public listInvestors;

     mapping(address => uint256) public amountShares;
    // Add the constructor.
    constructor(
        string memory name,
        string memory symbol,
        address payable wallet,
        uint goal
    )
        public
        {
        // Create a new instance of the NFTvote contract, with 0 initial suppl
        NFTvote token = new NFTvote(name, symbol);
        // Assign the token contracts address to the `nft_token_address` va
        NFT_token_address = address(token);
        }

        // Buy shares of NFT
        // Put Eth in deposit
        //map msg.sender to amount
```

[5][5]NFTvote.sol

```
    function buyShare() payable public{
        uint256 balance = address(this).balance;
        address msgSender = msg.sender;
        uint256 msgValue = msg.value;
        listInvestors.push(msgSender);
        totFunded += msgValue;
        amountShares[msgSender] = msgValue;
        if (totFunded > goal){
        //     mintAll()
        //}
    }

    function mintAll() {
        // for all address in listInvestors
        for (uint i = 0; i < listInvestors.length; i++){
            token.safeMint(listInvestors[i]);

        }
    }
}
```

If funds required are reached :

- a NFT will be minted[6]. It will contain all informations submitted in the .json file.

- a DAO will be created. It will own the NFTContract, manage fund distribution for the student and allow stakeholders to vote with their VoteTokenVote Token is in ERC721 Standard.

The goal is to manage every requests and funds movements by smart contracts. Moreover, patents and commercial output from thesis will generate revenue and can be redistributed easily through DAO and shared NFT ownership.

For scientific the goal is to finance their research without the need to go every time look for new investors and partners.
Maintain as much freedom as possible for the researcher, and as much security as possible for the investor.

[6]Which represents ownership of future outputs of this Thesis, NFTContract

3 Technical Implementations

3.0.1 Scientific

Basically, it will be frontend implementation for fundraising with associated smart contracts[7] and unlock voting token[8] when the limit is reached.
The student will receive funds every months (or other distributions is possible), and upgrade NFTContract with submitted paper during the progress of the research.
At the end of the contract all patents, property and profits must be returned to the DAO through the NFT ownership. We can imagine that DAO can decided to sell ownership at someone who wants to buy it. In this case, profits will be redistributed at members of DAO and royalties can be in part return to the platform for the service.

3.0.2 Investors

3.1 Business Model

Initially, the Business Model was based on the return of selling NFT property of a DAO. But, it's at least for 3 to 4 years. So in the first times a grant model can be imagine with the incentive to sustain academic research which is not financed by traditional way.

4 Conclusion

We can't implement every options during this Hackathon

4.1 Openning

To move forward we have been created a DAO who own an ownership NFT. We can easily generalize this technical part and imagine many other use cases such that : record DAO for music, stakeholder in a startup or stakeholder in a patent.

More generally, a DAO allows to share property on a NFT. This can be part of future by allowing shared subscription, shared diploma and many other stuffs.

References

[1] *What is the Research DAO ?* https://docs.researchdao.io/

[2] *Le financement de la politique européenne de la recherche* https://www.touteleurope.eu/economie-et-social/le-financement-de-la-politique-europeenne-de-la-recherche/

[7] [5] The smart contract of FundRaising

[8] Before the existence of DAO voting tokens are only ERC721A

[3] *Lettre ouverte du 18 décembre 2019* Sauvons l'université http://www.sauvonsluniversite.com/IMG/pdf/lettre-ouverte-18-dc3a9cembre-relu.pdf

[4] *ANALYSIS OF NATIONAL PUBLIC RESEARCH FUNDING (PREF)* https://publications.jrc.ec.europa.eu/repository/bitstream/JRC107600/jrc107600$_p ref_h andbook_d ef_d ef_u$

[5] https://github.com/TheFulmini/ResearchDAO/blob/main/SmartContract/NFTvote.sol