DOTOKEN

Decentralized Event Application

By LAMSSAOUI Siham

TABLE OF CONTENTS

INTRODUCTION

In the Blockchain area everything can be monetized

Monetizing events

Get sponsors more involved.

Receives direct funds

Bring together sponsors and participants

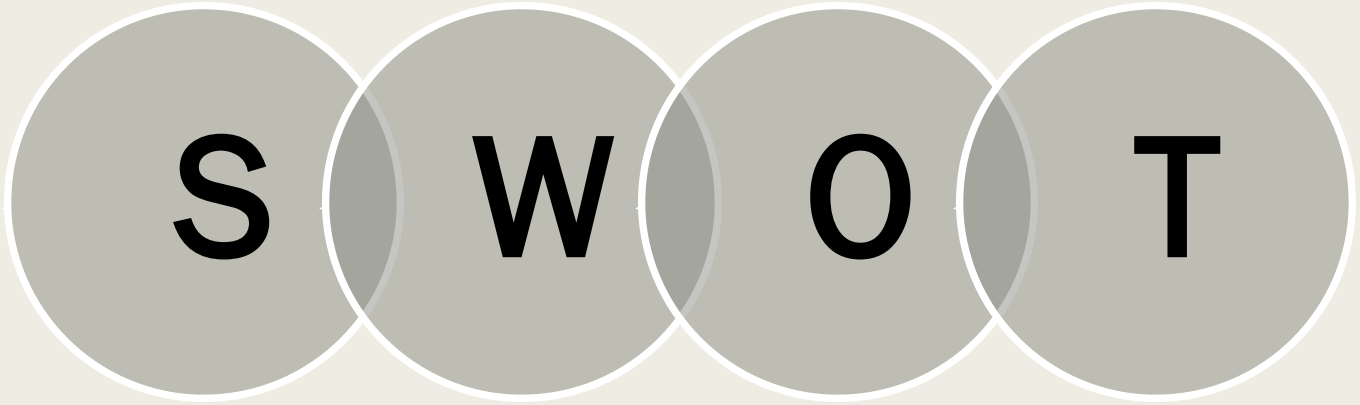
PROBLEM

Event hosts and sponsors usually do not get direct benefits. This is especially remarkable in the case of online Events.

SOLUTION

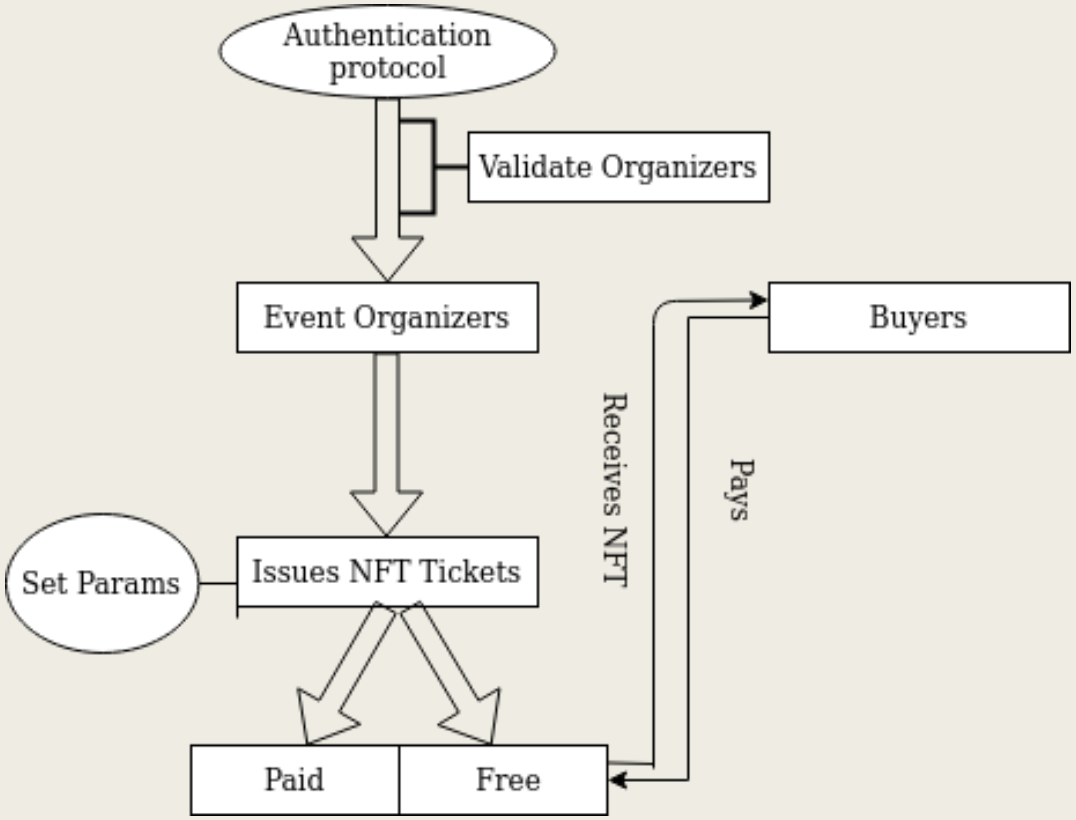
To build a Decentralized application based on Polkadot Blockchain, in order to monetize online events so both Sponsors and hosts can get a share of the benefits

SWOT ANALYSIS



STRENGTHS	WEAKNESSES	OPPORTUNITIES	THREATS
To get Sponsors involved in the project since they can get a direct benefit, a	The culture distance between the potential traditional Sponsors and the blockchain industry	Sponsors can integrate the project participants in their respective communities by the tokenization	Traditional sponsorship

PROJECT SCOPE



- Development of the #DOTOKEN, application for the monetization of online events and introducing blockchain into people's daily lives
- Develop an Solidity smart contract that cover most cases.
- Web page and mobile application to interact with the front users
- Application contains mainly two volet one for token buyers wishing to attend a specific event or course, and service sellers wishing to sell tickets for event or course,etc.
- Fully testable application is finished in more or less 6 months.

TARGET GROUP

- athletes

- Media

- Photographers

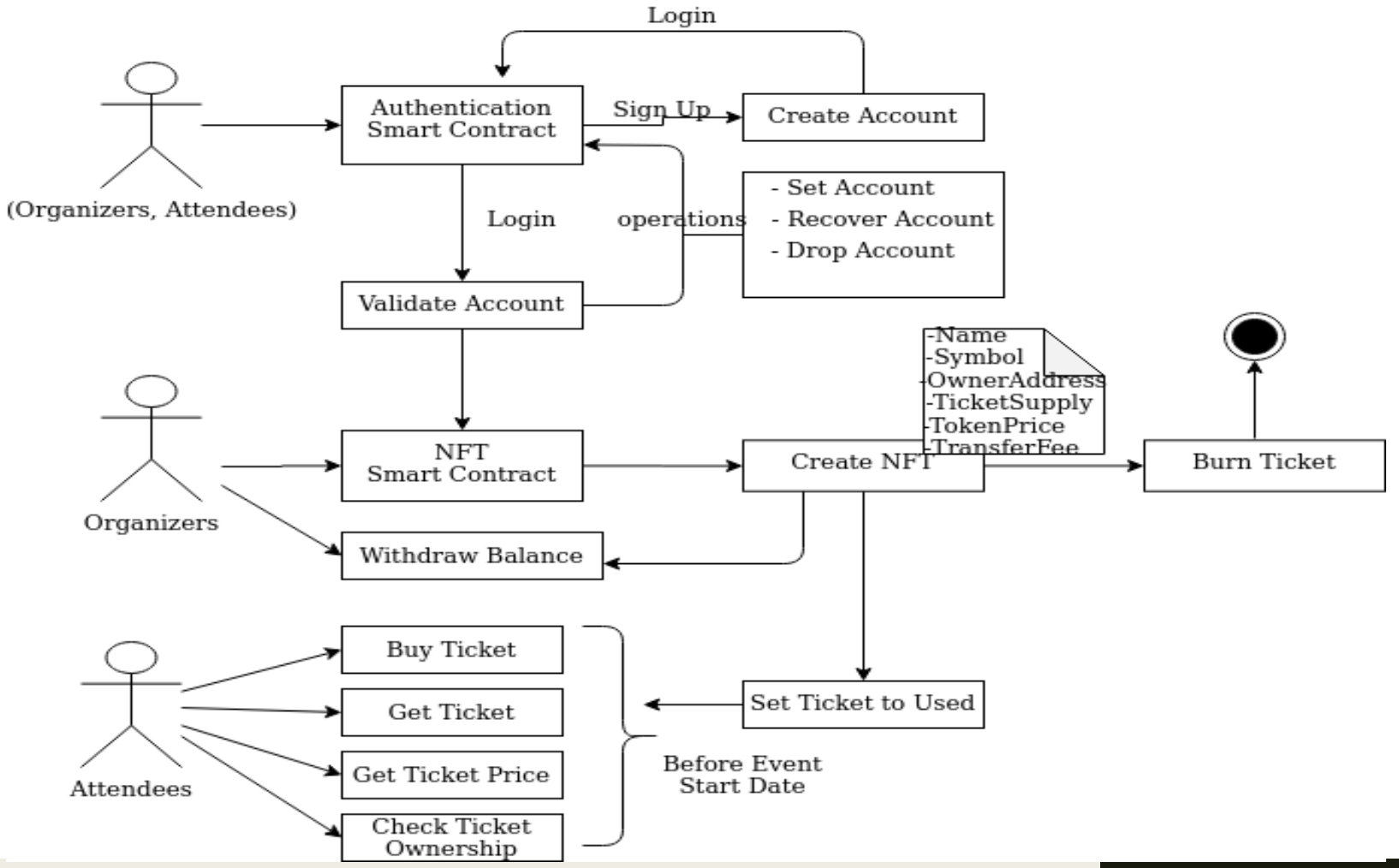
- Musicians

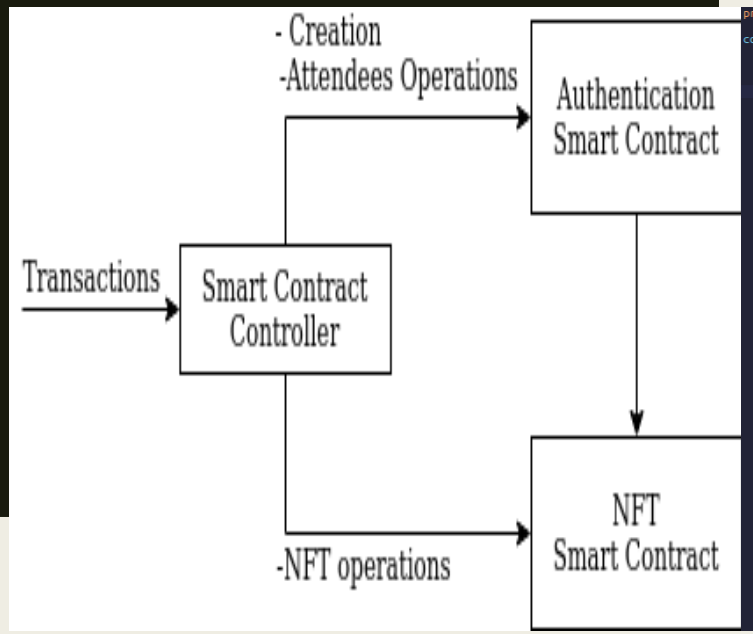
- influencers

- Journalists (finance investigative reports of interest to the community)

- Writers (Publication of works)

- Public administrations (For example, to reward good neighbors who recycle rubbish well, welcome dogs from municipal kennels or perform community support services)


Login
(Organizers, Attendees)
Authentication
Smart Contract
Sign Up
Create Account
Login
operations
- Set Account
- Recover Account
- Drop Account
Validate Account
-Name
-Symbol
-OwnerAddress
-TicketSupply
-TokenPrice
-TransferFee
Organizers
NFT
Smart Contract
Create NFT
Burn Ticket
Withdraw Balance
Buy Ticket
Get Ticket
Get Ticket Price
Check Ticket
Ownership
Attendees
Set Ticket to Used
Before Event
Start Date




```
pragma solidity ^0.5.5;

contract EtherAuth {
    mapping (string => address) authAddr ;
    mapping (string => address) recoveryAddr ;

    event Create(string login);
    event AuthChange(string login, address from, address to);
    event RecoveryChange(string login, address from, address to);
    event Drop(string login, address by);

    function createAccount(string memory _login) public {
        require(bytes(_login).length <= 32);
        require(bytes(_login).length > 2);
        require(authAddr[_login] == address(0));
        authAddr[_login] = msg.sender;
        recoveryAddr[_login] = msg.sender;
        //emit Create(bytes32ToString(_login));
        emit Create(_login);
    }

    function authAddress(string memory _login) view public returns (address){
        return authAddr[_login];
    }

    function setAuthAddress(string memory _login, address _addr) public {
        require(authAddr[_login] == msg.sender || recoveryAddr[_login] == msg.sender);
        emit AuthChange(_login, authAddr[_login], _addr);
        authAddr[_login] = _addr;
    }

    function recoveryAddress(string memory _login) view public returns (address){
        return recoveryAddr[_login];
    }

    function setRecoveryAddress(string memory _login, address _addr) public {
        require(recoveryAddr[_login] == msg.sender);
        emit RecoveryChange(_login, authAddr[_login], _addr);
        recoveryAddr[_login] = _addr;
    }

    function dropAccount(string memory _login) public {
        require(recoveryAddr[_login] == msg.sender);
        delete authAddr[_login];
        delete recoveryAddr[_login];
        emit Drop(_login, msg.sender);
    }
```

```
pragma solidity ^0.5.5;


import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v2.4.0/contracts/token/ERC721/ERC721MetadataMintable.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v2.4.0/contracts/token/ERC721/ERC721Burnable.sol";
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v2.4.0/contracts/token/ERC721/ERC721Metadata.sol";


contract DoToken is ERC721MetadataMintable, ERC721Burnable {


    string public name;
    string public symbol;
    address payable OwnerAddress;
    uint64 public TokenSupply;
    uint64 public eventStartDate;
    uint256 public TokenPrice;
    uint256 public initialTokenPrice;
    uint64 public transferFee;


  constructor(
        string memory _name,
        string memory _symbol,
        uint64 _eventStartDate,
        uint64 _TokenSupply,
        uint256 _TokenPrice,
        uint256 _initialTokenPrice,
        uint64 _transferFee ) ERC721Metadata(_name, _symbol) public {
        name = _name;
        symbol = _symbol;
        OwnerAddress = msg.sender;
        eventStartDate = uint64(_eventStartDate);
        TokenSupply = uint64(_TokenSupply);
        initialTokenPrice = uint256(_initialTokenPrice);
        TokenPrice = uint256(_TokenPrice);
        transferFee = uint64(_transferFee);
  }


  struct doToken  {
        uint256 price;
        bool forSale;
        bool used;
    }

  doToken[] doTokens;

    event doTokenCreation(address _by, uint256 _TokenId);
    event doTokenDestruction(address _by, uint256 _TokenId);
    event BalanceWithdrawn(address _by, address _to, uint256 _amount);

    modifier EventNotStarted() {
        require((uint64(now) < eventStartDate),"event has already started");
        _;
    }
```

PROJECT IMPLEMENTATION

WHY WE NEED THIS PROJECT

- ownership:

  NFTs cannot. Blockchain technology helps enshrine your ownership rights — and make digital assets a heck of a lot easier to move around.

- Transferable:

  NFTs can be freely traded on specialist markets. NFTs also solve the annoying problem about "walled gardens" in games — meaning coveted assets within a popular game could be used in a totally different title… or exchanged for items in a different game, even with a completely different publisher.

- Authentic:

  Fraud's a big problem — affecting everything from art to tickets and collectibles. The blockchains powering NFTs clamp down on counterfeiting — and give buyers confidence that they'll get what they pay for

COMMUNICATION PLAN

- App concept document & App specification – beginning of the project (Analyst)
- App prototype & App NFTS creation – end the 2nd month(Developer)
- Social media newsfeed – once per week
- App webpage and mobile – one month
- Client participatory design session - one per month
- Team meetings – one per week
- Once the website is finished, we will carry out a communication plan in generalist media, in order to introduce into the territory of blockchain and NFTs the people who operate daily in traditional finance.
- The project will have its own social channels: TW, linkedin, instagram
- We will carry out online meetups in English and Frensh to publicize the product