23/05/2018

OFFER - SMART CONTRACT IMPLEMENTATION.

This document describes costs of Viddo token smart contract implementation. After analyzing Requirement specification document delivered by Edith Kalocsai I concluded that following functional and non-functional requirements must be implemented. Where it is not specifically described then description from the VIDDO requirements specification document will be taken into account.

Non-functional requirements:

1. Implemented token must follow ERC20 requirements.
   Standard is described here:
   https://theethereum.wiki/w/index.php/ERC20_Token_Standard

   All ERC20 functions as described below will be implemented:

```
contract ERC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint balance);
    function allowance(address tokenOwner, address spender) public constant returns (uint
                    remaining);
    function transfer(address to, uint tokens) public returns (bool success);
    function approve(address spender, uint tokens) public returns (bool success);
    function transferFrom(address from, address to, uint tokens) public returns (bool
                    success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
```

2. TokeName: VIDDOtoken
3. Token symbol: VDT
4. Number of decimals: 0
5. Total amount: 100 000 000
6. Tokens not sold/used in one phase can used in another phase.
7. Smart Contract will be implemented using Truffle Network
8. At least three automatic tests will be implemented for each function.
9. At least one automatic test will be implemented for each Functional Requirement.
10. Implement following phases: pre-ICO, ICO, and after ICO:

   - pre-ICO - no sale available. User get their tokens manually via airdrop or direct manual bonus functionality. Not possible to buy tokens with ETH.
   - ICO - clients can buy tokens via sale contract, manual bonus functionality available. Maximum number of tokens for sale - 40 000 000.
   - after ICO - no sale. Clients can buy tokens only to exchange them for pro account.

Manual bonus transfers possible.

Functional requirements:

1. Crowd sale smart contract must be implemented to allow clients to buy and sell tokens
2. Burn functionality must be implemented. Tokens can be burned in two situations:
   - token owner decides to burn his tokens. Amount is then deducted from his balance and token's totalSupply.
   - * Token owner can also exchange his tokens for PRO registration. In such situation tokens will be also burned, but we are going to inform users that token has been exchanged. Solidity event will be emitted with information that token was exchanged.

3. Seperate function will be implemented to Burn Tokens and exchange them for PRO account. Information about number of tokens exchange for pro account will be stored separately for each client.
4. Implement Buy PRO account function. This function will charge user current price of token and will exchange this token for pro account. No token will be delivered to the buyer. Event informing about such action will be emitted.
   x
5. Implement function informing about total amount of tokens. This is a standard ERC20 token function called total Supply.

6. Implement functionality to stop/pause token after all tokens has been burned. This means that when total Supply is zero then it should not be possible to use token anymore.

7. Use extern service (http://www.oraclize.it/) to query current ETH/USD and ETH/BTC price and use it for token price calculation.

8. Implement function to return current exchange rate used by token to be used by external website.

9. Implement function to airdrop tokens. This means that token owner (Edith) will collect set of ETH addresses and amount to be transferred at certain point. So, we need following functions here:

   a. add airdrop - make sure that current amount booked tokens is not bigger than total available amount
   b. remove airdrop
   c. get list of airdrops
   d. edit airdrop - make sure that current amount booked tokens is not bigger than total available amount
   e. execute airdrop - this function will be executed by the owner and will transfer tokens to all current registered users.
10. Make function to define total amount of tokens which can be sold/transferred in each phase.

11. Implement function to show total amount of tokens available for sale and total amount of tokens already in circulation. Take into account burned tokens.

Open questions:

1. Do investors (PRE-ICO) buy tokens or tokens are assigned manually? Answer was yes. Yes - they buy or Yes - it is assigned manually?
2. We are going to have a list of airdrops to be executed. Do we also need list of verified addresses to be able to buy tokens during ICO?

Difference is between transferring tokens for free and allowing only verified customers to buy tokens? Is it for ICO or for Pre-ICO?

3. Are we going to implement functionality to allow people to buy and sell their tokens or we are going to simply put token on the exchange market?

Offer:

I estimate total working time for implementation as follows:

- ERC20 Smart Contract implementation - 5 hours.
- Crowd sale contract, manual bonus functionality, exchange ratio calculation, burning tokens and buying PRO account – 20 hours.
- Test Implementation and testing - 20 hours.

Total: 45 hours.

Price: 45 hours * 60 USD/h = 2700 USD

Schedule: First working version will be delivered within 14 days of project start.
Final version will be delivered depending on the amount of found defects and clients availability, but not later than 28 days from the project start.

Functionality not included in this offer:

- Web page or any external interface is excluded from this offer. Only standard web3.js API will be available for interaction with this smart contract.
- No additional API will be implemented.