Project1: Define a fungible token with a purpose

Bina Ramamurthy

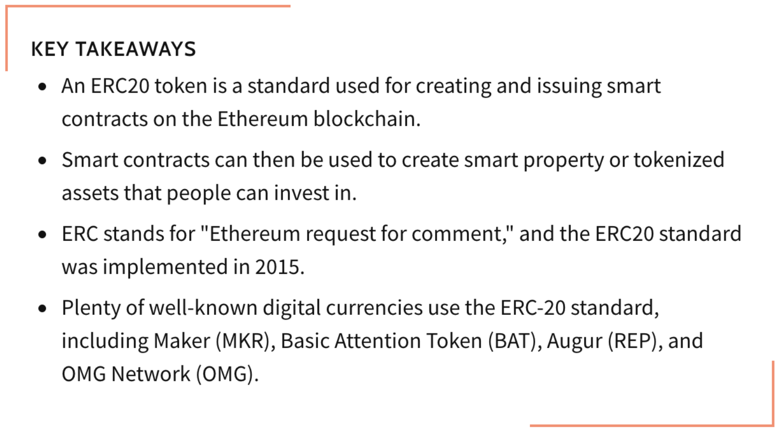
9/22/2021

Intoduction

Tokenization of assets is an important application of blockchain and cryptocurrency. In this project, you will design and implement a fungible token (FT) with a purpose: a service, a utility, or a product, or a process. The project idea has to be innovative and useful. There are standards for implementing fungible tokens. We will use ERC20 standard for Ethereum blockchain network.

`As of October 2019, more than 200,000 ERC-20-compatible tokens exist on Ethereum’s main network.[2]`

Review the key takeways[2] about ERC20 listed in Figure 1[2].



KEY TAKEAWAYS

- An ERC20 token is a standard used for creating and issuing smart contracts on the Ethereum blockchain.
- Smart contracts can then be used to create smart property or tokenized assets that people can invest in.
- ERC stands for "Ethereum request for comment," and the ERC20 standard was implemented in 2015.
- Plenty of well-known digital currencies use the ERC-20 standard, including Maker (MKR), Basic Attention Token (BAT), Augur (REP), and OMG Network (OMG).

Figure 1: What is ERC20

Description

The project involves design and development of two major components:

- The ERC20 token with a purpose, and
- The web application (or mobile) including a wallet such as MetaMask that delivers this token and lets participants use it for the designated purpose.

Project phases

Phase 1: Research and Identify problem to solve – 10% – Due Date: 10/2/2021 Saturday by 11.59PM on ublearns.

Our motto in system development is *Problem first.* The problem is to address an issue that is readily solvable by traditional systems. For example, global plastic cleanup. This problem is discussed in Chapter 7 in your text book [3]. Your assignment in phase 1 is to think of a problem where payments can benefit by tokenization. Write a short abstract about this problem you are solving with ERC20 tokens. Here is the format of the document for submission:

1. Title of the project
2. First name 1, Last name 1, email, person number 1; First name 2, Last name 2, email, person number 2
3. Issue(s) addressed:
4. Abstract: a 100 word description of the problem: what, why, when, who, where and how etc. How will you use ERC20 token to solve your problem?
5. Your ERC20 symbol and reasoning
6. The special functions besides the standard required functions of ERC20 standard.

Get the project idea approved by Bina or the TAs. You can see all the taken project ideas here. https://docs.google.com/spreadsheets/d/1Z7cd98xY6CDWJJAmcjiYsalTM8MFkowvb5qKUCTTazc/edit?usp=sharing After that create a pdf document and submit on ublearns.

Phase 2- Design and Implement the ERC20 token – 55% –Due date 10/16/21 Saturday by 11.59 PM

1. Understand the ERC20 token and its functions [4]. Determine the various parameters such as sysmbol, toatl supply, number of decimal places etc. for your ERC20 token and justify and record/document the reasons for your choice. Understand how these functiosn apply to your design of ERC20.

```
function totalSupply() public view returns (uint256);

function balanceOf(address tokenOwner) public view returns (uint);

function allowance(address tokenOwner, address spender) public view returns (uint);

function transfer(address to, uint tokens) public returns (bool);

function approve(address spender, uint tokens)  public returns (bool);

function transferFrom(address from, address to, uint tokens) public returns (bool);
```

2. Implement the ERC20 smart contract using the support smart contracts libraries [5][6].
3. Deploy your ERC20 on a test net such as Ropsten [7] and make sure you are able to view, and send your tokens using the MetaMask wallet. You mey need some inail Ropsten ETH balance to deploy any smart contract. You can obatin test Ether from these two sources [8][9].

Outcomes of Phase 2 On completion of Phase 2 you should have a custom ERC20 token smart contract ready and deployed to solve your problem. You should good understanding of what an ERC20 and how it is transacted.

What to submit?

1. Submit the complete code base, and the detailed instructions on how to deploy and test your ERC20 on Remix. (Optional: on a web interface). For examples of instruction of a interaction with a smart contract deployed on Remix see the chapter 4 and 5 of Blockchain in Action book [3].
2. Any **original** architectural diagram explaining your smart contracts and the relationship among them.
3. A screen shot of your ERC20 Token deployed on Remi IDE.

All these bundled together as a zip file and submitted on ublearns designed link for phase 2.

Phase 3: An application using your ERC20 token – 35% – Due date 11/6/2021 Saturday, by 11.59 PM

Now it is time to solve the problem you began with (phase 1) using the ERC20 token you created (in phase 2). Write the logic (code) to solve the problem in a smart contract (s). The (Solidity) code solution shoudl make appriate use of the ERC20 token you created.

Outcomes of phase 3 On completion of Phase 3 you should a smart contract **application logic** ready and deployed to solve your problem. This smart contract should make use of the ERC20 token of phase 2.

What to submit?

1. Submit the complete code base of the smart contract and ERC20 implementation, and the detailed instructions on how to deploy and test your smart contraxt application on Remix. (Optional: on a web interface). For examples of instruction of a interaction with a smart contract deployed on Remix see the chapter 4 and 5 of Blockchain in Action book [3].
2. Any **original** architectural diagram explaining your smart contracts and the relationship among them.
3. A screen shot of your smart contract that uses ERC20 deployed on Remi IDE.

All these bundled together as a zip file and submitted on ublearns designed link for phase 3.

References

1. ERC20 Token Standard. https://eips.ethereum.org/EIPS/eip-20, last viewed September 2021.
2. N. Reiff., What Is ERC-20 and What Does It Mean for Ethereum?, https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/, last viewed September 2021.
3. B. Ramamurthy. Blockchain in Action. Manning Publishers. 2020.
4. J. Dourlens, UNDERSTAND THE ERC-20 TOKEN SMART CONTRACT, https://ethereum.org/en/developers/tutorials/understand-the-erc-20-token-smart-contract/, last viewed Sept 2021.
5. Sahil Sen, How to create and deploy an ERC20 token, https://www.quicknode.com/guides/solidity/how-to-create-and-deploy-an-erc20-token, last viewed Sept., 2021.
6. Michal Zalecki, Create and distribute your ERC20 token with OpenZeppelin, https://www.tooploox.com/blog/create-and-distribute-your-erc20-token-with-openzeppelin, Last viewed Sept. 2021.
7. Gilad Haimov, How to Create an ERC20 Token the Simple Way, https://www.toptal.com/ethereum/create-erc20-token-tutorial, last viewed Sept. 2021.
8. Ropsten Faucet 1: https://faucet.dimensions.network/ (5 ethers), last viwed Sept. 2021.
9. Ropsten Faucet 2: https://faucet.ropsten.be/ (0.3 ethers), last viewed Sept. 2021.