Implementation of Crowd Sale using ERC-20 Tokens

Ashish Sharma[1], Yogesh Sharma[2], Radhika Bansal[3], and Sushant Verma[4]

[1]Computer Science Department, Maharaja Agrasen Institute of Technology, India, ashish@mait.ac.in

[2]Computer Science Department, Maharaja Agrasen Institute of Technology, India, yogeshsharma@mait.ac.in

[3]Computer Science Department, Maharaja Agrasen Institute of Technology, India, radhikabansal5898@gmail.com

[4]Computer Science Department, Maharaja Agrasen Institute of Technology, India, sushantv7597@gmail.com

Abstract: The paper aims at creating ERC-20 Token and crowd sale step-by-step with Ethereum smart contracts. In the process, it focuses on testing the smart contracts, deploying the smart contracts to the Ethereum blockchain, and building an ICO website deployed to the web for the buying and selling of the tokens. The Ethereum blockchain provides a platform for creating our cryptocurrency, or tokens that can be purchased via Ether which is the native cryptocurrency of the Ethereum blockchain. ERC-20 is a standard document that specifies the behavior of the tokens so that they are compatible with other platforms like cryptocurrency exchanges. We used Ethereum which is a blockchain like Bitcoin. We created a token called "Token" and there are 100,000,000 such tokens. Firstly, the token smart contract keeps track of some token attributes which are basic. It also keeps track of who owns "My Token" and how much. ERC-20 tokens can be used as a payment just, just like any other cryptocurrency, from one account to another. They can also be purchased in a crowd sale, like an ICO.

Keywords: Ethereum, Blockchain, Smart Contracts, Crowd Sale, Solidity, Meta Mask

1. INTRODUCTION

In the past few years, blockchain and its applications have shown a potential growth in the field of computer science, resulting in strong demand for software applications. It has provided a path for a lot of new projects which are proving an easiness in the day to day life. In this project we created ERC-20 Token and crowd sale step-by-step with Ethereum smart contracts, test the smart contracts, deploy the smart contracts to the Ethereum blockchain, and build an ICO website deployed to the web. The Ethereum blockchain allows us to create our cryptocurrency or token, that can be purchased with Ether, the native cryptocurrency of the Ethereum blockchain. ERC-20 is simply a standard that specifies how these tokens behave so that they are compatible with other platforms like cryptocurrency exchanges. We used Ethereum which is a blockchain like Bitcoin. The major difference between ERC20 tokens and other cryptocurrencies is that ERC20 tokens are created and hosted on the Ethereum blockchain, whereas bitcoin and bitcoin cash is the native currencies of their respective blockchains.

The project aims at creating ERC-20 Token and crowd sale step-by-step with Ethereum smart contracts, test the smart contracts, deploy the smart contracts to the Ethereum blockchain, and build an ICO website deployed to the web. The Ethereum blockchain allows us to create our cryptocurrency or token, that can be purchased with Ether, the native cryptocurrency of the Ethereum blockchain. ERC-20 is simply a standard that specifies how these tokens behave so that they are compatible with other platforms like cryptocurrency exchanges. We used Ethereum which is a blockchain like Bitcoin. We created a token called "Token" and there are 100,000,000 of these tokens in existence. First, the token smart contract keeps track of some basic token attributes. It also keeps track of who owns "My Token" and how much. ERC-20 tokens can be

DOI: 10.5281/zenodo.3708402

transferred from one account to another as payment, just like any other cryptocurrency. They can also be purchased in a crowd sale, like an ICO.

1.1 *Types of Blockchain*

- Centralized network —this network, there is a central network owner. The central network owner is a single point of contact for information sharing. The biggest disadvantage of the centralized network is that the whole system is controlled by a single central node, which can also become a single point of failure that can crash the whole system [1].
- Decentralized network —In the case of a decentralized network, we have multiple central nodes that have a copy of all the resources. This solves the problem of a single point of failure with a centralized network as it has multiple owners, so if a particular central node fails, the information can still be accessed from the other nodes.
- Distributed network — the distributed network is the decentralized network taken to the extreme. They are difficult to maintain. But it avoids the centralization completely thus a single point of failure does not affect the whole system. This type of system has infinite scalability.

1.2 *Brief of Smart Contracts*

Szabo's conception defines the smart contracts as the digital set of rules for information transfer that use algorithms to automatically execute a transaction once the predefined conditions are successfully met and that control the process completely [2]. This definition, which was way ahead of its time, and is accurate to this day. The code of the smart contracts is used to enter all the terms and conditions of the contract concluded between all the participants to the transaction into the blockchain. The participant's obligations are given in the smart contract in the "if-then" form. There can be two or more parties involved, and they can be individuals or organizations. Once the given conditions are met, the smart contract independently executes the transaction and ensures that the agreement is adhered to.

1.3 *ERC20 and Other Tokens*

ERC-20 defines six different functions for the benefit of other tokens within the Ethereum system. These are generally basic functionality issues, including the method of how users can access data regarding a particular token and in which tokens are transferred. This set of functions ensures that Ethereum tokens of different types will uniformly perform in any place within the Ethereum system. Most of the digital wallets which support the ether currency also support ERC-20-compliant tokens. But since the ERC-20 standard is in initial, there will likely be bugs.

The rest of this paper is represented as follows: Section 2 represents the literature review, Section 3 represents terminology used, Section 4 represents methodology, Section 5 represents experimental results and discussions, Section 6 represents the conclusion, and Section 7 represents the scope for improvement.

2. LITERATURE REVIEW

Bitcoin was the first cryptocurrency to use blockchain and has been the market leader since the first bitcoin was mined in 2009. After the birth of Bitcoin with the genesis block, more than 1,000 altcoins and crypto-tokens have been created, with at least 919 tradings actively on unregulated or registered exchanges. This entire class of cryptocurrencies and tokens has been classified by some tax authorities as having the same status as commodities. [3] Crypto market-related factors such as market beta, trading volume, and volatility appear to be a significant determinant for many cryptocurrencies both in the short- and long-run. The attractiveness of cryptocurrencies also matters in terms of their price determination, but only in the long-run.

DOI: 10.5281/zenodo.3708402

This indicates that the formation (recognition) of the attractiveness of cryptocurrencies is subjected to the time factor. In other words, it travels slowly within the market. [4] Cryptocurrencies are not likely to replace traditional fiat currency, they could change the way Internet-connected global markets interact with each other, clearing away barriers surrounding normative national currencies and exchange rates. [5]

3. TERMINOLOGY USED

3.1 Ethereum

Ethereum is an open-source platform. It is a blockchain-based distributed computing technology as well as an operating system introducing smart contract functionality. Ethereum has a cryptocurrency known as Ether which is used to pay off mining nodes for computations performed. Each Ethereum account has its ether balance which can be transferred from one account to another [6].

3.2 Ethereum Blockchain

The structure of the Ethereum blockchain is a shared record of the complete transaction history. This is similar to that of a Bitcoin. Every time a bitcoin transaction is made, the network 'breaks' the total amount as if it was paper money, issuing back bitcoins in a way that makes the data behave similarly to a physical coin. In an Ethereum application, the network needs to keep track of the 'state', including each user's balance and all the smart contract code [7].

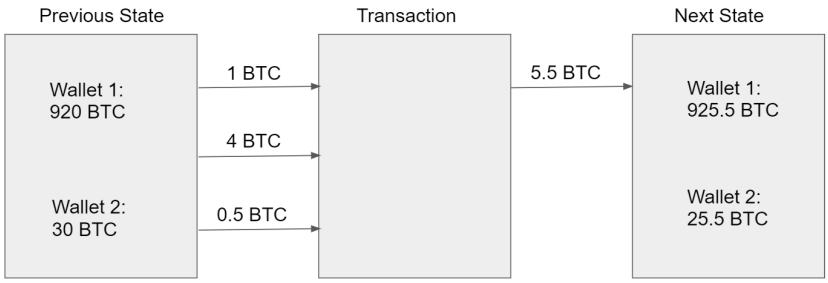

Figure 1: A Bitcoin Transaction

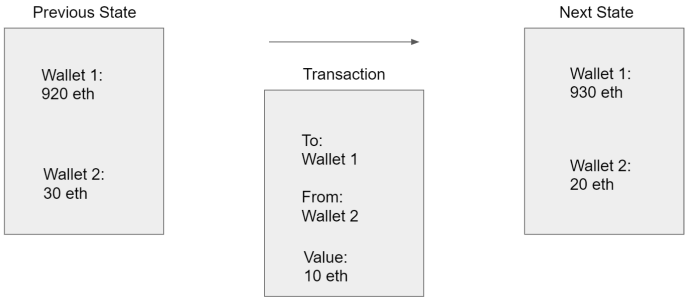

Figure 2: An Ethereum Transaction

DOI: 10.5281/zenodo.3708402

3.3 Crowd Sale (ICO)

An Initial Coin Offering also referred to as ICO, is a fundraising process that helps the new projects to sell their underlying crypto tokens in exchange for ether and bitcoin. Similar to that, Crowd sales are a way for a company to raise capital for their business by creating their ERC-20 token that can be purchased by investors. Whenever a crowd sale takes place, the company gets liquid capital in the form of Ether that was paid by the investors, as well as holding onto a reserved amount of the ERC-20 tokens that were sold in the crowd sale [8].

4. METHODOLOGY

The constituents of this application are:

- A web application that allows users to interact with the underlying technology by making requests to the application back-end to serve the purpose of buying and selling ERC20 tokens using Ethereum blockchain.
- A back-end that serves to deploy the smart contracts on the Blockchain, retrieve data or make a transaction.
- Various libraries are necessary for compilation of the code of the application and deal with the smart contracts.

Ethereum platform makes use of the blockchain technology and helps to make a successful project because blockchain makes the funding process safe and it is accessible from anywhere in the world and completely transparent.

After installing all the dependencies, we started building ERC20 token smart contract which implements ERC20 token standards and has the following attributes:

- Token name string as "Token".
- The token symbol for the cryptocurrency.
- Total supply of tokens as a uint256 public variable "totalsupply".
- Balance of every account that owns tokens as a Solidity mapping (address => uint256) public variable “balanceOf”.

This smart contract [9] implements the following functions:

- Transfer function - allows users to send tokens to another account.
- Approve function - allows another account to spend tokens, similar to a cryptocurrency exchange that updates the allowance mapping to see how much the account is allowed to spend.
- transfer from a function - allows another account to transfer tokens.

Along with the smart contract, the test was also generated to discover more about how it works. These tests ensure that the smart contract behaves the way it is supposed to. Following is the demonstration:

DOI: 10.5281/zenodo.3708402

```
PS F:\wetransfer-c6eb16\token_sale-5_front_end> truffle test
Warning: Both truffle-config.js and truffle.js were found. Using truffle-config.js.
Using network 'development'.


Compiling your contracts...
===========================
> Compiling .\contracts\Token.sol

TruffleConfig {
  _deepCopy: [ 'compilers' ],
  _values: {
    truffle_directory: 'C:\\Users\\Sush\\AppData\\Roaming\\npm\\node_modules\\truffle',
    working_directory: 'F:\\wetransfer-c6eb16\\token_sale-5_front_end',
    network: 'development',
    networks: { development: [Object] },
    verboseRpc: false,
    gas: null,
    gasPrice: null,
    from: null,
    confirmations: 0,
    timeoutBlocks: 0,
    production: false,
    skipDryRun: false,
    build: null,
    resolver: TestResolver {
      resolver: [Resolver],
      source: [TestSource],
      search_path: 'C:\\Users\\Sush\\AppData\\Local\\Temp\\test-119931-11212-1ours04.ddgj',
      seen: []
```

Figure 3: Truffle Tests

We build a crowd-sale smart contract that will allow investors to purchase tokens in an initial coin offering (ICO).

The smart contract functions as a crowd sale as follows:

- It holds an “admin” account for the crowd sale address admin.
- It references the ERC20 token smart contract - Token public “tokenContract”.
- It stores the token price - uint256 public “tokenPrice”.
- It stores the number of tokens sold - uint256 public “tokensSold”.
- It implements a “sell event” so that consumers can get notifications whenever a token has been sold.
- It implements a “buyTokens” function - allows users to purchase tokens in the crowd sale.
- It implements an "endSale" function - allows an admin to end the crowd sale and collect the Ether funds that were raised during the sale.

To check whether ERC20 smart contract and crowd sales smart contract works as they are supposed to, some tests were run which gave us the following results:

```
},
confirmations: undefined,
production: undefined,
timeoutBlocks: undefined,
_: [],
'show-events': false,
showEvents: false,
test_files: [
  'F:\\wetransfer-c6eb16\\token_sale-5_front_end\\test\\Token.js',
  'F:\\wetransfer-c6eb16\\token_sale-5_front_end\\test\\TokenSale.js'
]


Contract: Token
  √ initializes the contract with the correct values (182ms)
  √ allocates the initial supply upon deployment (86ms)
  √ transfers token ownership (266ms)
  √ approves tokens for delegated transfer (421ms)
  √ handles delegated token transfers (823ms)

Contract: TokenSale
  √ initializes the contract with the correct values (197ms)
  √ facilitates token buying (524ms)
  √ ends token sale (303ms)


8 passing (3s)
```


DOI: 10.5281/zenodo.3708402

Figure 4: Test results

Since the smart contracts passed all the test cases and work as expected, now the smart contract is deployed at the localhost server and the ganache workspace.

Now once the smart contracts are deployed, the next step is to migrate them using truffle. Truffle migrations allow us to "push" the smart contracts to the Ethereum blockchain (either on the localhost, testnet or mainnet) and it helps us to set up crucial steps for joining contracts with other smart contracts as well as populate them with the initial data. Where migrations shine is the management of contract addresses on the blockchain.

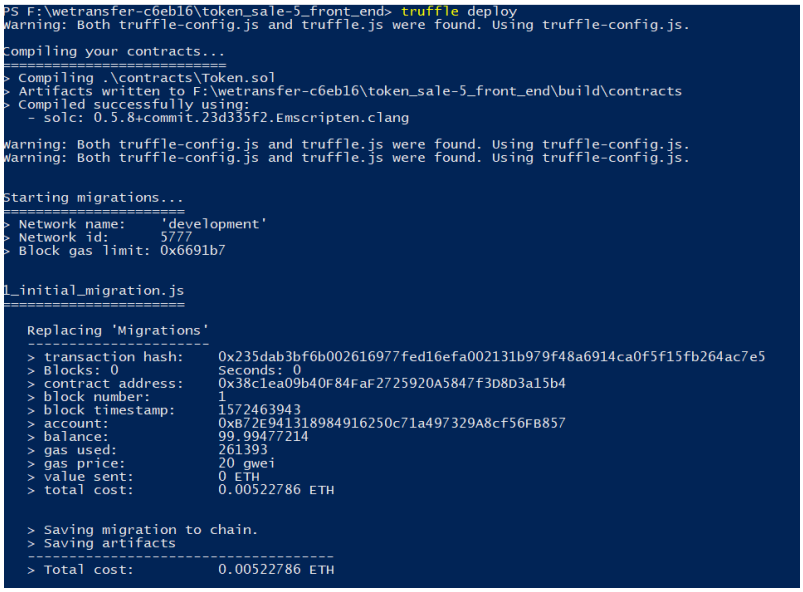

Figure 5: Deploying Contracts

An individual's information needs to be updated at the time he/she enters another country. Due to security reasons, only the information about successfully reaching the destination can be changed. More power given to the application will create security issues. Moreover, limited power means that there will not be a single point of failure. Even if the app fails, no one will be able to update other information of an individual due to the limited rights of the application.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The front end of the project, i.e. the crowd sale website was developed using HTML, JavaScript, and Bootstrap. To run the crowd sale website on localhost, NPM local host packages were used which were connected to the MetaMask [10] extension to work with Ethereum.

DOI: 10.5281/zenodo.3708402

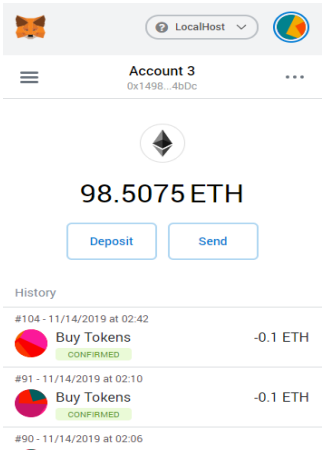

Figure 6: Buyers' account transactions after buying tokens.

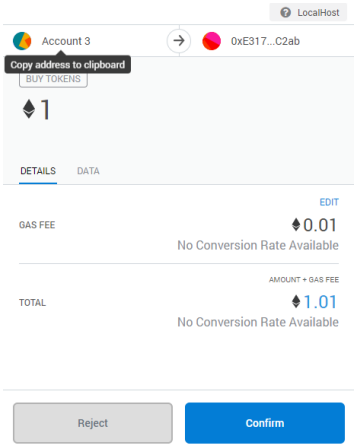

Figure 7: Meta Mask Pop-Up for confirmation of buying tokens.

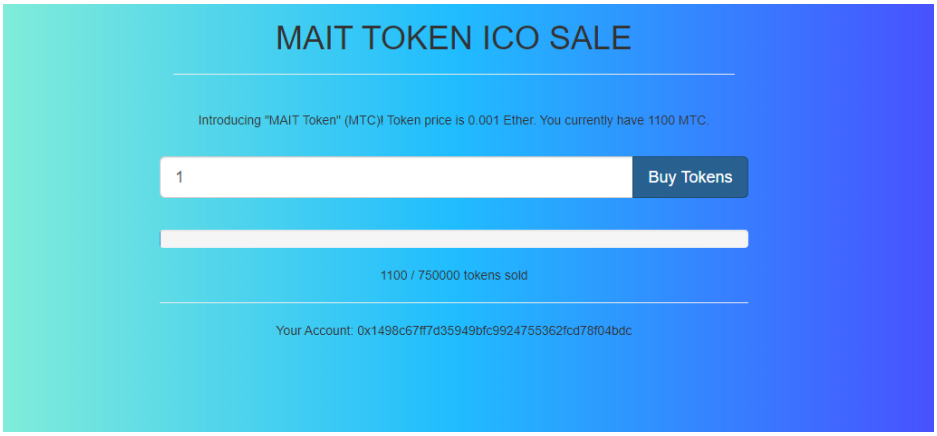

Figure 8: Crowd Sale Website

6. CONCLUSION

As seen in the previous section, the proposed system uses the Ethereum blockchain network for creating an Initial Coin Offering platform. This ICO – Crowd sale website, holds an admin account which is responsible for selling the ERC-20 tokens. This system uses two smart contracts to justify the behavior of the tokens and the transactions as well. These tokens could be purchased by the investors using the Ethers in their Ethereum accounts.

This project was made possible by exploiting the merits of Blockchain technology. The existence of Blockchain on Hyperledger [11] makes it possible for all the instances of this application to be on a unified decentralized platform. Using this platform, the companies could raise capital for their business by creating their ERC-20 token that can be purchased by investors in the form of Ether.

DOI: 10.5281/zenodo.3708402

7. SCOPE FOR IMPROVEMENT

Ethereum is a shared computer technology that belongs to everyone but is not controlled by anyone. It is available permanently, no one can stop it. It is extremely redundant as it consists of a multitude of computers

connected (in peer-to-peer mode), each running a public program (open source). The value of Ethereum will grow with the number of interesting applications developed and used. The bubbling of applications created (even before the finalization of Ethereum) gives us an indication that the movement is well launched. Opening the platform encourages its development by anyone interested in doing so. Another growth vector will be the Ethereum interconnection with everyday devices. A computer isolated from its environment has little value, whereas its networking increases tenfold its possibilities. Existing applications include applications such as Board Room which is a voting system (useful for managing an organization), event or market prediction applications (such as Augur or Gnosis). Ethereum is based on several programs developed in parallel by independent teams. This approach is much more robust than if there was only one version of the code that manages the network. Also, Ethereum has its established development plan and significant improvements are planned. Finally, Ethereum is a vast and ambitious ecosystem on which a digital revolution will be built. Many systems that we all use today can be rebuilt or upgraded to Ethereum, with the advantage of reducing or eliminating the centralization of these systems.

References

[1] Z. Zheng and S. Xie, "Blockchain challenges and opportunities: a survey," *Int. J. Web and Grid Services,* vol. 14, no. 4, pp. 1-24, 24 July 2018.

[2] M. Canul and S. Knight, "Programming Smart Contracts in Ethereum Blockchain using Solidity," *SIGCSE'19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education,* p. 1236, 2019.

[3] L. Guo, Y. Wang and D. L. K. Chuen, "Cryptocurrency: A New Investment Opportunity?," *The Journal of Alternative Investments,* vol. 20, no. 3, pp. 16-40, 2018.

[4] Y. Sovbetov, "Factors Influencing Cryptocurrency Prices," *Journal of Economics and Financial Analysis,* vol. 2, no. 2, pp. 1-27, 2018.

[5] P. D. DeVries, "An Analysis of Cryptocurrency, Bitcoin, and the Future," *International Journal of Business Management and Commerce,* vol. 1, no. 2, p. 9, 2016.

[6] S. Tikhomirov, "Ethereum: State of Knowledge and Research Perspectives," *Springer,* February 2018.

[7] A. Bogner and M. Chanson, "A Decentralised Sharing App running a Smart Contract on the Ethereum Blockchain," *Proceedings of the 6th International Conference on IoT,* p. 177–178, 2016.

[8] S. T. Howell, D. Yermack and M. Niessner, "Initial Coin Offerings: Financing Growth with Cryptocurrency Token Sales," *The National Bureau of Economic Research,* vol. 2, no. 1, pp. 1-66, 2018.

[9] K. Iyer and C. Dannen, "Building Games with Ethereum Smart Contracts," *Springer,* pp. 57-90, 2019.

[10] S. T. Muriki, "Online reviews immutability tool using Blockchain," *Department of Computer Science California State University,* vol. 1, no. 1, pp. 1-53, 2018.

[11] Q. Nasir and M. A. Talib, "Performance Analysis of Hyperledger Fabric Platforms," *Hindwai,* vol. 1, no. 1, pp. 1-14, 2018.

DOI: 10.5281/zenodo.3708402