International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
http://ijcjournal.org/

Design and Implementation of Decentralized Voting System on the Ethereum Blockchain

S.U. Nnebe[a]*, C.S. Okafor[b], T.I. Onyeyili[c], G. Nathaniel[d]

[a,b,c,d]*Department of Electronic & Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria*

[a]*Email: su.nnebe@unizik.edu.ng*

[b]*Email: cs.okafor@unizik.edu.ng*

[c]*Email: ti.onyeyili@unizik.edu.ng*

[d]*Email: godspowernathaniel25@gmail.com*

Abstract

This work involves the design and implementation of a decentralized voting system on the Ethereum blockchain, which is a peer-to-peer network. The system is helpful in carrying out free and fair elections as information stored on the blockchain is immutable. This voting application uses solidity as the backend language and the web3 library for reading and interacting with the blockchain. JavaScript, Hyper Text Markup Language (HTML), and Cascading Style Sheets (CSS) are used to design the front end and the control logic for the website. The voting system works with the locally installed Ethereum node. The user visits the website and registers his details which are then uploaded to the blockchain in the cryptographically hashed pattern. After registering, the user is directed to the voting page, which reads the intelligent contract data and allows the user to cast his vote and at the same time update the blockchain. This system can be deployed in schools, organizations, countries, anywhere there is a need for governance and democratic voting. The prototype built was tested and found to be working perfectly.

Keywords: E-voting; blockchain; Ethereum; Democracy; Decentralized voting.

1. Introduction

The most fundamental part of democracy is its availability to citizens. Democracy allows the citizens to share their opinions and beliefs and creates a platform that gives everyone a say and the right to control the progress of things. Any society practicing democracy, must offer a process of decision making, this process of deciding and making decisions is known as voting. There is more than one alternative to voting, it could be by head-count, ballot boxes, or employing an electronic system. Any method of voting selected must be transparent for the vote to be accepted. The whole process should not pose any misconduct or malpractice and the privacy of individual identity must be protected.

---

* Corresponding author.

With many of the available voting processes today, the voting results can be easily manipulated, with the main challenge being, providing transparency and security, while protecting individual privacy and thus promoting democracy. In today's world, people need to be connected, and they are willing to have access to information quickly, especially through the internet or IoT-based devices and as such people are desirous to be in touch with the current affairs happening around the world [1].

In the world today, taking the paper-based voting process as an example, the voters must trust both the public authority for publishing the results and their local vote-collecting groups for collecting the votes. In the paper-based voting process, a lot of factors that can truncate a free fair election while counting and deciding votes include self-interest, human mistakes or a dishonest public authority. To solve this problem and get better election results, an electronic approach can be employed, where the process of collecting results is automated and stored. With the recent Covid 19 pandemic, there is a need for more automated and online-based applications. One of such is the use of online voting in the 2020 United States Presidential elections [2]. Countries like Estonia have since long practiced internet voting [3]. In Nnamdi Azikiwe University, Awka today, the use of internet voting is no longer a mystery as its Student Union Government elections are held online, recording more student participation and promoting democracy. The issue with this is that the results and data stored by the third party can easily be hacked as it has a single entry point and is hosted on centralized servers. There are a lot of issues that can arise from the blockchain of the stored data such as server damage, data manipulation or even hacking. Therefore, blockchain alone cannot provide the optimal solution for an electronic voting system. The best way to gain good ground on the problem is to break down the activities and control of the data. The control and decision-making of the data would not be centralized. With one of the proposed use case of blockchains being voting because of its security and peer-to-peer networks [4], they have been a surge in a lot of DAOs (Decentralized Autonomous Organizations) which carry out voting on their indigenous blockchains, an example of such DAOs are Choice Coin, Invictus DAO.

To solve some of the mentioned problems, decentralized voting should be adopted. In decentralized voting, the decisions are carried out online rather than in traditional voting. The use of blockchain technology in decentralized voting solves the problem of privacy, correctness and integrity by protecting individual identity using cryptographic properties, hashed data, and Biometrics which are physical or behavioural human characteristics that can be used to digitally identify a person, for him/her to be granted access to systems, devices, or data [5]. A decentralized voting system was developed by creating a smart contract on the Ethereum network that allows voters to vote and read election data ensuring a free, fair and trustworthy approach to voting for good governance and decision making. This becomes very important considering the current state of elections in the world and Nigeria in particular, with lots of misconduct and malpractices. This blockchain-based e-voting system would remove all discrepancies involved with voting and would promote democracy.

2. Literature Review

A blockchain is a distributed data structure that is replicated and shared among the members of a network [6]. It was introduced in 2008 with the publishing of Bitcoin's white paper [7]. A blockchain is a decentralized, distributed, public ledger that holds cryptographically stored data of old and current transactions. The data

stored on ledgers are organized in blocks where each block is a digital piece of information about transactions [8]. Each block contains a cryptographic hash of the previous block to assure there is a standard order to the blocks. This, therefore, links the blocks and builds symbolically a chain and gives the blockchain two essential properties; any modification of the earlier block would be made public to all the blocks on the chains and would invalidate all the previous blocks and anyone can verify the whole chain looking at the first block. Blocks created based on the latest block of the most current chain are processed by nodes in a peer-to-peer network. When a change occurs on the blockchain, all the nodes and blocks must be updated to reflect that change which creates a need for a consensus mechanism. The consensus mechanism ensures that all blocks are recording the same transaction at the same time, in the same order [9]. Every creation follows a consensus mechanism, which is a protocol in order to agree on which transactions are legitimate and added to the blockchain. This protocol ensures everyone has the same pieces of information about the transactions. Information on the blockchain is transparent and anyone can view the content of the blockchain. It is important to note that transactions are not completely anonymous. However, information about the users is limited to their digital signature

A set of instructions used for modifying the state of the blockchain is referred to as a transaction. Transaction fees are a part of public blockchain networks in order to have the transactions processed by specialized nodes, which may either be by proof of work, or by proof of state.

These specialized nodes are called miners. In a blockchain like Ethereum that uses proof of work [10], the miner nodes supply computing power and carry out transactions, while blockchains like Algorand that uses a transaction method known as pure proof of state [11], authorize transactions by randomly picking an account to serve as the third party and verify blockchain transactions. The Ethereum blockchain uses a proof of work that follows the proof of work protocol that trusts the block with the most computational work put into it. In a proof of work protocol, each block stores a nonce (number used once). It differentiates the block from other blocks and makes each block unique when it is put through some hashing algorithm. Therefore, due to the nonce, the hash output of the contents of the block will change. This is where computational work comes into play. Miners must find a nonce value that when plugged into a specific hashing algorithm, generates an output that meets certain requirements [12]. In a broader sense, miners create new blocks on the blockchain as they validate and authorize transactions [13].

For each transaction to be made on the blockchain, a user must provide an asymmetric key pair, a private key and a public key. The public key is associated with a digital wallet which serves as an address to the user. The public key is cryptographically hashed to form a random string of letters and numbers but is made public on the blockchain and can be viewed by anyone. Any transaction carried out on a public blockchain is expected to be included in the ledger if they are valid and take part in consensus. Besides public blockchains, there are consortium blockchains, where the consensus process is handled by a pre-selected set of nodes. This kind of blockchain is known as a private blockchain where write permissions are kept to one organization and permissions may be made public or restricted [14]. The private key is kept secret to ensure that only the owner of the wallet can sign transactions.

The Ethereum blockchain is one of the widely accepted and acknowledged blockchains in the web3 world

today. Ethereum is a decentralized peer-to-peer network. It is a general-purpose blockchain that has a good understanding of a higher programming language. Other blockchains integrate well with Ethereum such as Polygon, Avalanche, and Fantom Opera. Ethereum sets a standard for a good number of blockchains. One of the major drawbacks of the Ethereum network is its high gas fees, but according to Vitalik, the Ethereum founder, a new version of Ethereum 2.0 [15] is in the making, with the question being only a matter of when. The brain behind a smart contract is a set of logic, that if met would always return true. It implements the conditions of an agreement if all necessary parameters are met. Smart contracts allow applications to communicate with the blockchain, Ethereum uses Solidity, Solana- Rust and Algorand- TEAL (Transaction Execution Approval Language). Solidity is the language used to write smart contracts on the Ethereum blockchain, it is a high-level object-oriented programming language with syntax and semantics like all other programming languages [16]. It allows for the declaration of variables and creation of functions. Ethereum smart contracts are deployed at specific addresses, the code is visible to everyone on the blockchain with the smart contract address. Smart contracts are immutable, and do not change. To modify a smart contract, the old one has to be destroyed from the blockchain and a new smart contract uploaded.

2.1. Nigerian Elections

The 2019 elections held a lot of discrepancies, and accusations, with videos and photos spreading online of various electoral misconduct and malpractices going on[17], it can be said that the Nigerian elections have never been completely free and fair, with post electoral differences and court cases always being the order of the day[18]. In earlier times, it was the use of hoodlums and thugs to disturb election processes, but recent moves by the government and electoral body have placed a curb to such activities. Another recent technological trend that has further made elections more free and fair is the vast use and acceptance of social media as a tool for information dissemination. At the snap of a finger, information can easily be uploaded and accessed online, this in a turn has led to decentralization of information, because not only few selected bodies like the government installed bodies or private owned media sources, but anybody today can be a distributor of information and news. With this in place, nefarious activities carried out by thugs and individuals can easily be exposed to the masses before, after and even during election. Videos of strange electoral conducts happening live filmed at the epicenter of the incident had been seen. With all this in view, one must confess that a more secure way of voting and promoting democracy is needed. Decentralization puts the system in the hands of the users. The more nodes in a blockchain, the more decentralized it becomes.

3. Methodology

The Ethereum blockchain as one of the widely accepted and acknowledged blockchains in the web3 world today was used. It is a decentralized peer-to-peer network and a general-purpose blockchain that has a good understanding of a higher programming language. Other blockchains integrate well with Ethereum such as Polygon, Avalanche, and Fantom Opera. Ethereum sets a standard for a good number of blockchains.

The smart contract was written using remix, an online Integrated Development Environment (IDE) for writing and testing solidity smart contracts. The Remix is a powerful, open -source tool that helps one to write solidity

contracts straight from the browser or locally. Remix supports testing, debugging and deploying smart contracts. Therefore Remix boasts a good virtual environment for simulating the states of smart contracts.

The steps involved in writing the smart contract are:

(i) Creation of a new solidity file on remix. (ii) Declaring solidity version, classes and object structs. (iii) Compiling the smart contract on remix. (iv) Launching the virtual environment to test the smart contract. (v) Testing each state, variable and function.

After testing the smart contract on the remix, the next phase was integrating it with a solidity framework. In this work truffle framework, an open-source software developed by MIT, which integrates Ganache that creates a local blockchain, for testing, bundling and building dApps before deployment was used. Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), which aims at making life easy for a developer. When compared to Hardhat, truffle and Ganache are more graphics based while Hardhat is more command line based.

Truffle allows for the compilation, and testing of the live results of the developed codes. It also allows the building and integration of the deployed smart contract to the developed codes. The smart contract serves as the backend app while the frontend app was built with Web3.js which supports Client Side Rendering. Web3.js is an Ethereum library that allows interaction with the blockchain like reading the state of the blockchain, and getting transaction hash and history.

3.1 Truffle Installation

To install truffle, launch the command prompt under normal privileges and run:

npm install -g truffle, which installs truffle globally on the personal computer. After installation, run the truffle command to make sure it has been added to environment path variables. On running truffle, it would bring out a list of commands to execute, like the build, compile, and console.

After truffle installation, a local directory which would be used to navigate through the command line, installing a truffle instance to download a Truffle Box that is a pre-built Truffle project was created. In this work, it was the pet-shop project, a basic truffle template using lite server for development, then delete the old files, place the already developed smart contract from remix and run truffle migrate to deploy the smart contract to the blockchain.

Command for truffle-

npm install -g truffle

Truffle

truffle unbox pet-shop

3.2 Testing with Truffle

To run truffle tests, Ganache was installed for running the local node. Ganache provides us with ten accounts and funds them each with 100eth. Ganache allows truffle to test, and migrate contracts to the Ethereum blockchain. Start by creating a test file (election.js). Write all tests in JavaScript inside the file with the Mocha testing framework and the Chai assertion library. The Mocha testing framework and the Chai assertion library are bundled with the Truffle framework. All these tests are written in JavaScript to simulate client-side interaction with the smart contract, much like what is done in the console. The JavaScript test codes are written. Truffle tests are run to see the outcome of the code.

The first test checks that the contract was initialized with the correct number of candidates by checking the candidate's count is equal to 2. The next test inspects the values of each candidate in the election, ensuring that each candidate has the correct id, name, and vote count.

3.3 Application Development

The next step of development is to navigate to the project folder and open it with any code editor of your choice. In this work, Visual studio code was used, as it's free software, with a lot of plug-ins available for development. The code would open with the already unboxed truffle files, with a truffle-config.json file. Create a js file, for handling different events on the server, such as loading when initializing metamask. Initialize web3, the JavaScript library that allows our client-side application to talk to the blockchain. Web3 was configured inside the "initWeb3" function.

Initializing the smart contracts, would fetch the deployed instance of the smart contract inside the initWeb3 function and assign some values that would allow interaction with it.

Render function: The render function lays out all the content on the page with data from the smart contract. The candidates created were listed inside the smart contract and this was done by looping through each candidate in the mapping, and rendering it to the table. The current account that is connected to the blockchain was fetched inside this function and displayed on the page.

The building of the client-side application, which would talk to the smart contract, was done by modifying the HTML and JavaScript files that came with the Truffle Pet Shop box already installed. The existing code was used to get started. It is important to note other things that came with the Truffle Pet Shop box like the Bootstrap framework that reduces the CSS and lite-server program to be written and serves as assets for development purposes.

After writing the client-side application code, run truffle migrate --reset to redeploy the smart contract to the blockchain.

To deploy the development server on the command line, run npm and then run dev to start the lite server.

4. System Implementation and Result Analysis

The system modeled a closed-off election process, with eligible voters being given access to the smart-contract data, it had a function that checked to make sure that users can only vote once. It contains the smart contract logic, the web application, and JavaScript codes. It protects privacy and anonymity, while also carrying out credible election processes.

The decentralized voting system developed was built in two parts. The smart contract is written in solidity, the JavaScript codes are used to validate a voter, and the web application communicates with the contracts and represents the voting process. The purpose of smart contracts is to act as the back-end of the system. It contains different variables, functions, and rules of how the voting process works. Due to the limitations of Ethereum, such as private variables that are not truly private and all transactions being public, ways of keeping data inaccessible were adopted. Specialized tools were used to support the development of smart contracts. On successful compilation of the smart contract, it returns an address which is the location it is stored on the blockchain.

For users to vote and be given access to the voting page, they must enter a valid address that tallies with their Metamask address. If the wallet address matches, the user is directed to the voting page, else, the user is prompted to register again. After registration, a function reloads the page and redirects the user to the voting page. The web3 gallery is used to fetch data from the blockchain. The web app creates an interface and serves as a GUI for the voting process. The web application was built using truffle's pet-shop project that used a lite server to load the webpage's content. Due to constraints in Solidity, there is no easy way to fetch a whole map or array from the smart contract. This had to be done manually by getting each vote one by one from the contract and decrypting them in the process. The developed Web app functions as follows:

- Reload the content on windows.event change.
- Render smart contract data.
- Process and update vote count from the blockchain.
- Register users to vote

4.1 Flowchart

Pseudo-Code of the System: The pseudo-code for the system development:

- Load Homepage.
- Check Registration Status.
- If registration status is false, show the register button.
- Else show launch app and log out button.
- If the page is registered, load registration content.
- On register form submissions, save NIN, save wallet address.
- Check if wallet address matches, Metamask address.

- If a match exists, redirect to Vote.html.
- Else issue error code: credentials do not match.
- If page equals vote page, load contract data.
- On vote submission, register vote, and reload the page.
- If the user already voted, show vote polls, prevent voting

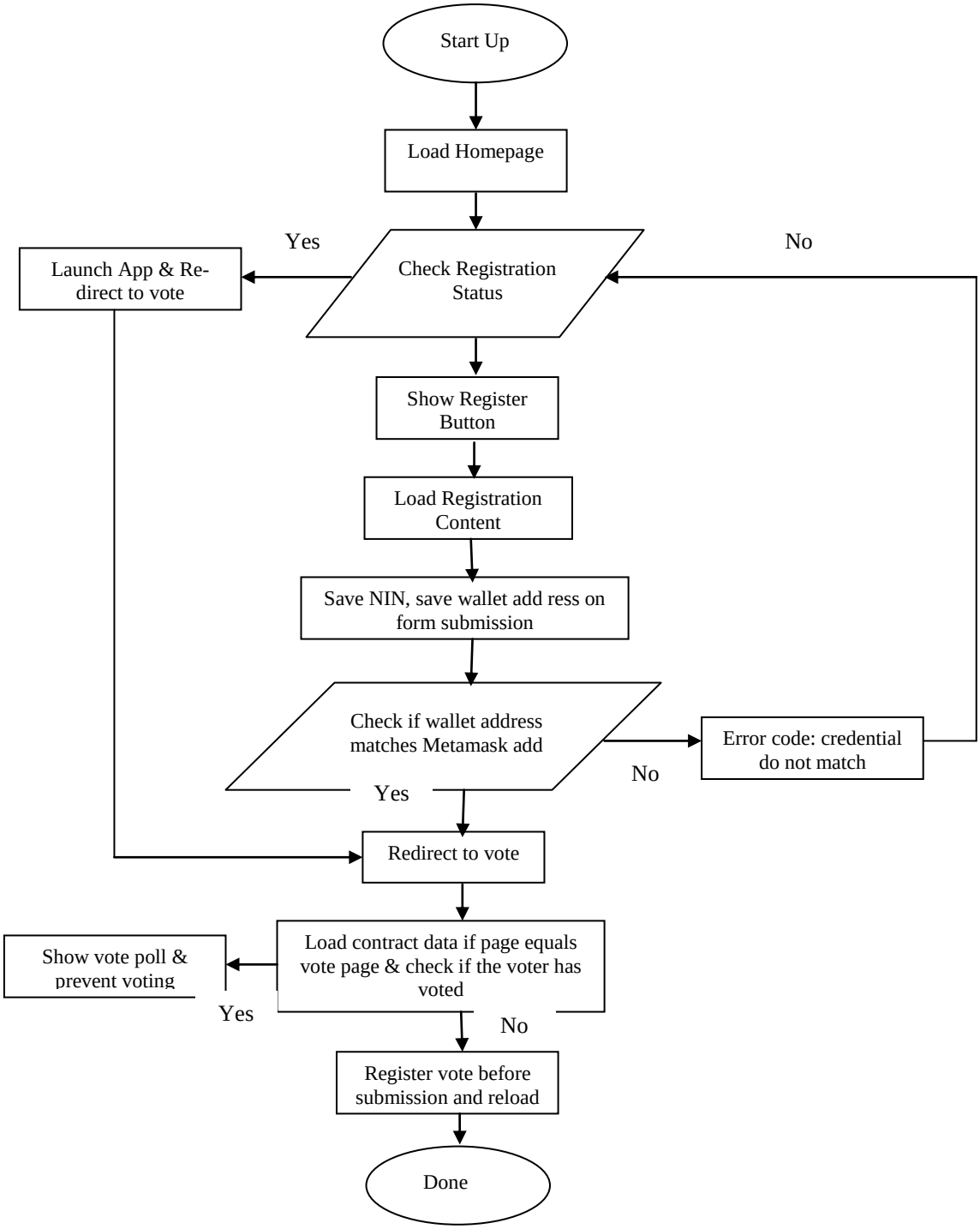

Figure 1: Flowchart showing the pseudo code for the system development

5. Conclusion

The goal of this work "decentralized voting" is to provide an effective and reliable means of carrying out a more secure and democratic process of voting for organizations.

The work uses real-time updated data from the blockchain, to decide the vote counts and candidates. The changes made are immutable and cannot be hacked as there is no central entry point. This work can be deployed to schools, countries, states, and bodies practicing democracy, although much change would have to be made to the source code and voting method as needed. This project is a prototype of an actual voting system, as the necessary tools to create and implement a voting system for a country, or organization are large in size. The prototype built was tested several times and found to be working normally. However the system has a limited scope as it covers two candidates, proper data mapping in terms of geographical location should be done. It has been established that blockchain could be used to develop a good voting system though it involves adequate funding and also lower gas fees should be implemented on the Ethereum blockchain

Finally, the in-depth research done on this work gave an insight into the limitations associated with the software and necessary changes that could be made for better improvement in this technological era. It was discovered that for the system to be more accurate it requires much more resources such as libraries, public blockchain ledgers, and accurate data manipulation from the blockchain.

References

[1] C.S. Nwokoye, A.N. Aniedu, C.S. Okafor , A.C. Nzemalu A.C, "Design Of Interactive Smart Mirror System for Digital Information Display Based on Multitasking Approach Using Raspberry Pi", *Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, Vol.7 No.2, PP. 143 – 147, July 2022.

[2] Husayn Kassai, "The U.S. Election And The Pandemic: Is E-Voting The Way Forward?" https://www.forbes.com/sites/forbestechcouncil/2020/09/10/the-us-election-and-the-pandemic-is-e-voting-the-way-forward/?sh=6e66e59de650. Sept. 10, 2020. [Online; accessed 6-February-2022].

[3] Valimised, "Statistics about internet voting in Estonia," https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia, [Online; accessed 6-February-2022].

[4] Joe Liebkind, "How Blockchain Technology Can Prevent Voter Fraud" https://www.investopedia.com/news/how-blockchain-technology-can-prevent-voter-fraud, Dec. 9, 2020, [Online: Accessed 10th February-2022].

[5] C.S. Okafor, S.U. Nnebe, T.L. Alumona, V.C. Onuzuluike, U.C. Jideofor, "Door Access Control Using RFID and Voice Recognition System, *International Journal for Research in Applied Science & Engineering Technology*, Vol. 10, Issue 3, PP. 157 – 163, Mar 2022.

[6] J¨org Bremer, Sebastian Lehnhoff, (2017) "Decentralized Coalition Formation with Agent-based

Combinatorial Heuristics. ADCAIJ", Advances in Distributed Computing and Artificial Intelligence Journal, Salamanca, 6(3).

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf, 2008.

[8] Jack Ahlkvist, Anton Gustafsson, Carl Lundborg, Joakim Mattsson Thorell, Aron Sandstedt, Sanjin Slavnic, "A Decentralized Voting System", B.Sc. Thesis, Chalmers University of Technology, Gothenburg Sweden, 2019.

[9] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in 2017 IEEE international congress on big data (BigData congress). IEEE, 2017, pp. 557–564.

[10] POW- Ethereum docs: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/

[11] The Algorand Protocol: https://algorand.foundation/algorand-protocol/about-algorand-protocol

[12] C. Dannen, Jan. 2017 "Introducing Ethereum and Solidity", pp. 111–137. [Online]. Available: https://doi.org/10.1007/978-1-4842-2535-6 6.

[13] RebeccaYanga, Ron Wakefield, Sainan Lyua, Sajani Jayasuriya, Fengling Han, Xun Yi, Xuechao Yang Gayashan Amarasinghe, Shiping Chenc, May 2020, "Public and Private Blockchain in Construction Business Process and Information Integration", Automation in Construction Vol.118; https://doi.org/10.1016/j.autcon.2020.103276.

[14] Endgame. Dec. 6, 2021, Available:S https://vitalik.ca/general/2021/12/06/endgame.html, [Online; accessed 04-February-2022].

[15] Solidity 0.8.15 documentation. [Accessed 5-February-2020]. [Online]. Available:https://docs.soliditylang.org/en/v0.8.15/

[16] Electoral Fraud And Democratic Election: A Comparison Of Nigeria 2019 Elections And United States 2020 Elections

[17] Atiku vs Buhari: Nigeria Presidential Election Petitions Tribunal go pass judgement today

[18]Truffle [Online] Available: https://trufflesuite.com/ sourced October 2021