

A Scalable Blockchain Implementation Model for Nation-Wide Electronic Voting System

Apeh Jonathan Apeh[1(✉)], Charles K. Ayo[2], and Ayodele Adebiyi[1]

[1] Covenant University, Ota, Ogun State, Nigeria
[2] Trinity University, Yaba, Lagos State, Nigeria

Abstract. Blockchain technology adoption rate is fast growing as seen in cryptocurrency and distributed finance (DiFi) domains. It is also getting lots of attention in many other application areas including electronic voting(e-voting) systems. The electronic voting system is an interesting application use case for blockchain because it helps to solve critical problems within that space- the integrity of voting data, the secrecy of the ballot, and single point of failure. This is because of the characteristics that blockchain technology embodies. One of the challenges, however, is with the scalability of the blockchain network, how the blockchain technology can power the scalability of systems built on it. The aim of this paper, therefore, is to present a Blockchain Implementation Model that tackles scalability concerns for E-Voting System. This model can be adaptable in any national election, specifically, Nigeria's national elections. The resulting model would present a scalable electronic voting framework by leveraging the security and integrity infrastructures that blockchain technology brings to bear.

Keywords: Implementation model · Election · Ballot · Electronic voting system · Suffrage · Integrity infrastructure · Digital signature · Public key

1 Introduction

Security is considered the biggest concern with any electronic voting system [1]. This could be either the security of the ballot or even the election result. For an e-voting system to be trusted, people want to see that their ballot is not tempered with, nobody has control to manipulate the outcome of an election.

Blockchain technology has security infrastructures that make the electronic voting system an interesting application use case. These infrastructures include immutability, auditability, decentralization, and smart contract [2]. Leveraging on these properties of the blockchain helps to solve popular problems associated with most e-voting systems (i.e. compromise of the secrecy of the ballot and that of the election results). Despite the promises of blockchain technology, a major challenge has to do with its scalability [1, 3].

Scalability simply means the ability of a computing system to increase or decrease computing resources depending on the workload need of the system. There are two types of scalability in a computing system. These are vertical and horizontal scalability.

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12957, pp. 84–100, 2021.
https://doi.org/10.1007/978-3-030-87013-3_7

In vertical scalability, the system scales up or down, depending on the workload. For a scale-up, the system resources like the Random Access Memory (RAM), the processor can increase in size to be able to support the demand of the workload. In a scaled-down scenario, the compute resources will have to reduce in size. In horizontal scalability, however, the system scales out or in. In this situation, there is either an increase or a decrease in the number of system instances available to manage the workload.

In a blockchain network, scalability is simply the measure of the ability of the network to add new nodes. It is also a reflection of its throughput i.e. transactions per second (TPS) [1]. It is one of the performance challenges associated with blockchain. There are different approaches to improving the blockchain scalability available in the literature. Some of these approaches published in the literature are mentioned in Sect. 2.4.

The rest of this paper is organized as follows: Sect. 2 provides a review of the Electronic Voting System, and Blockchain Technology. In Sect. 3, the design of the proposed system is presented; Sect. 4 represents the methodology adopted; Sect. 5 contains the results and discussions.

2 Literature Review

2.1 Electronic Voting System

An electronic voting system is any computing system or electronic device that is used in the conduct of elections. Major components of a typical electronic voting system include haptic touchscreens, thermal printers, smartcard reader, external LCD, Uninterruptible Power Supply (UPS), and a standard computer [4, 5].

This description is typical of most electronic voting systems including the Direct Recording Electronic (DRE) machine used in the United States of America (USA), Netherlands, Brazil, Mexico, Australia, India [6]. These democracies have varying e-voting systems [7, 8] base on their peculiar situations.

For an e-voting system to be considered secured or credible, it must exhibit the following characteristics [9]: fairness, eligibility, privacy, verifiability [10], and coercion-resistance.

2.2 Blockchain Technology

Blockchain technology otherwise known as a Distributed Ledger Technology [2], is a fully distributed, peer-to-peer software network that rides on cryptography and distributed computing as underlying technologies. It makes use of cryptography to host applications and store data in a secured manner and to easily transfer digital instruments(value) that could represent real-world money [11–13].

It uses community validation (i.e. consensus) to keep committed transactions synchronized (this could be cast votes) and ledger replicated across multiple nodes or miners in a way that it is decentralized, transparent, immutable, auditable, persistent, and secured [2, 3, 14, 15]. Figure 1 shows how blockchain technology works [16].

Major steps in how the technology works include: **i**.Transaction Request **ii**.Transaction broadcast **iii**.validation **iv**.A verified transaction is combined with other

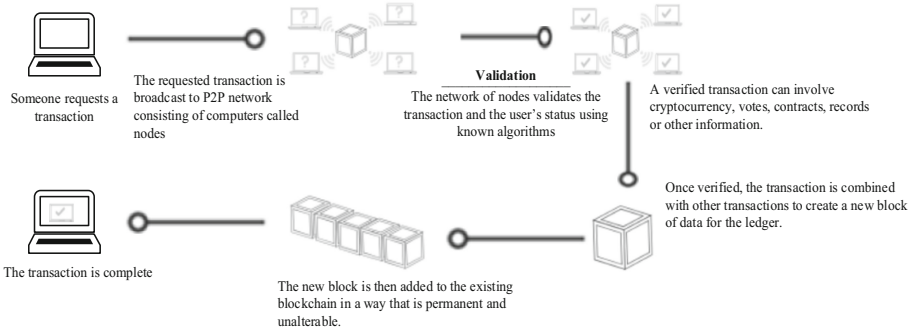

Fig. 1. How blockchain works [16]

Table 1. Blockchain structure

TXN ID	Timestamp	Transaction(txn) details (hash, size, status)	Last txn (current txn hash details, previous txn hash)
1	1585735315(Wed 1st April, 2020; 10:01:55 UTC)	0xfd36ebd112d79348246f 4e08f6a312d80060be7d044 ca7030417a0f6b3b02be6; 234MB, COMFIRMED	0xfd36ebd112d793482 46f4e08f6a312d80060be 7d044ca7030417a0f6b3b 02be6 + genesis hash
2	1586127600 (Sunday, 05-Apr-20 23:00:00 UTC)	0xd980b87683ea92c86edeb 5c2109ac681ebbd8f2f71d 4408b53123c09f782f01a; 230MB, COMFIRMED	0xfd36ebd112d79348246 f4e08f6a312d80060be7d044ca 7030417a0f6b3b02be6 + genesis hash + 0xd980b87683ea92c86edeb 5c2109ac681ebbd8f2f71d4408 b53123c09f782f01a
3	1590042600 (Thursday, 21-May-20 06:30:00 UTC)	0x485155c53688fc0061f7 a91ac211540b73aae020d0ee 9f7e84e7e45c756e6b6f; 301MB; COMFIRM	0xfd36ebd112d79348246 f4e08f6a312d80060be7d044ca 7030417a0f6b3b02be6 + genesis hash + 0xd980b87683ea92c86edeb 5c2109ac681ebbd8f2f71d4408 b53123c09f782f01a + 0x485155c53688fc0061f7a 91ac211540b73aae020d0ee 9f7e84e7e45c756e6b6f

transactions to create a new block of data for the ledger. **v.**The new block is then added to the existing blockchain in a way that is permanent and immutable.

Given that Table 1 typically represents a blockchain, when a new record is inserted into it, the last computed hash is broadcasted to every interested party. Every party doesn't need to keep a copy of the entire transaction history as only a few parties are sufficient. Given that everyone knows the last hash, anyone can verify that the data hasn't

been altered since it would be impossible without obtaining a different and thus invalid hash. The only way to tamper with the data while preserving the hash would be to find a collision in the data, and that's computationally impossible. It would require so much computing power that it's practically uneconomical.

Blockchain Deployment Types

There are different types of blockchains. The most common classifications are public, private, and consortium blockchains. Table 2 shows the main properties of these blockchain types.

Table 2. Blockchain types [3]

Properties	Public	Private	Consortium
Ownership	For the people by the people and of the people	Own by individuals or organizations	Own by a group of individuals or organizations
Read/write/audit permission	All	Restricted	Restricted
Consensus Process	Permission-less	Permissioned	Permissioned
Consensus determination	All miners	A selected set of peers	A selected set of peers
Examples	Bitcoin, Ethereum, Litecoin	Multichain	Energy Web

Blockchain has applications in some interesting areas such as Finance, Supply Chain Management, Digital Content Management, Public Services, Telecoms, Electronic Voting. Top four blockchain implementations: Bitcoin, Litecoin, Dogecoin, Ethereum.

Security Features of Blockchain

Blockchain technology is characterized by certain security features [2] that define its applicability in varying domains. These include:

i Decentralized, peer-to-peer network (i.e. no central controlling authority). This promotes fairness in an e-voting system as it does not allows any peer to have a controlling influence on any outcome.

ii Distributed Ledger- all peers on the blockchain network have a copy of the ledger. They validate transactions using a defined consensus algorithm. This feature promotes fairness in the e-voting system as no single peer/authority can decide acceptable votes. Fairness is about impartiality, equal treatment, and integrity.

iii Immutable history of transaction: each new block contains the hash of the previous one. It is impossible to change existing or committed transactions on a blockchain network without being detected. This feature promotes verifiability in an e-voting system.

iv Transparency: transaction on a blockchain network is only visible to authorized participants. This feature allows for the privacy of a voter in an e-voting system. It also helps enforce coercion-resistance.
v Smart Contracts: these are business logic deployed on the blockchain, they are shared and validated by participants or peers on the network. This can enforce eligibility of an e-voting system by ensuring failure-proof authentication into the e-voting system.

2.3 Blockchain and E-voting System

E-voting systems generally automate the electoral processes: e-registration, e-verification, e-voting, e-counting, e-transmission, and e-reporting. The ability of e-voting systems to implement these electoral processes shows their potential in enhancing the efficient and effective conduct of elections except for their vulnerabilities [6].

Hao and Ryan [6] posit that most fielded e-voting systems are found to have some vulnerabilities. Most common susceptibilities have to do with the compromise of the voting outcome of election and secrecy of the ballot. Abayomi-Zannu et al. [17] and Awotunde et al. [18] suggest blockchain technology has security infrastructures that the e-voting system can leverage to eliminate the inherent vulnerabilities points. For instance, the blockchain network is a decentralized, peer-to-peer network. This means, in a blockchain network, there is no central controlling authority for validating transactions. Instead, it utilizes the consensus between the peers to validate a transaction. Applying this in an e-voting system would mean that vote cast is a transaction. Also, rather than using the regular server-client deployment model which is prone to single-point-of-failure, a decentralized and consensus-driven approach is used to validate cast votes. Blockchain also ensures that a distributed ledger or record of cast votes exists on all authorized peers on the decentralized network. To ensure that a cast vote cannot be changed, blockchain ensures that transaction history (i.e. history of cast votes) is immutable and it is transparent enough that all authorized participants can view it.

2.4 Scalability Problem in Blockchain

Scalability has been identified as a performance and capacity concern in blockchain technology [1] hence a lot of efforts are on improving it. There have been different solutions proposed to improve it without compromising the security and decentralization of the blockchain. These include solutions that deal with adjusting some system parameters (e.g. the block size of the blockchain network, throughput, or TPS), sharding and consensus strategy of the blockchain (i.e. modifying the consensus mechanism).

i Block size approach

The block size of a blockchain network indicates the number of transactions a network can validate and add to the ledger per time. The higher the size of the block, the more transactions can be processed per time. Also, the block size of a blockchain network has storage implications on the nodes on the network hence, compressed blocks have less storage overhead. This increases linearly the bootstrap time of existing nodes with

growth in the blockchain history and delays the process for adding new nodes as well. The proposed solution to handle this capacity and performance concern is targeted in compressing the block size of the blockchain system [1]. Some of these projects are Segregated Witnesses (SegWit), Bitcoin-cash, block compression, and storage scheme optimization [19, 20].

ii Sharding

This is a traditional technology for the optimization of large commercial database systems. The whole idea is to divide and conquer. It is used in this category of database systems to divide database data into different fragments and get them shared across several servers. This allows for improved search performance and storage management. It is applied in the blockchain space by dividing the whole blockchain network into smaller networks called shards. The nodes are a part of each smaller network fragment.

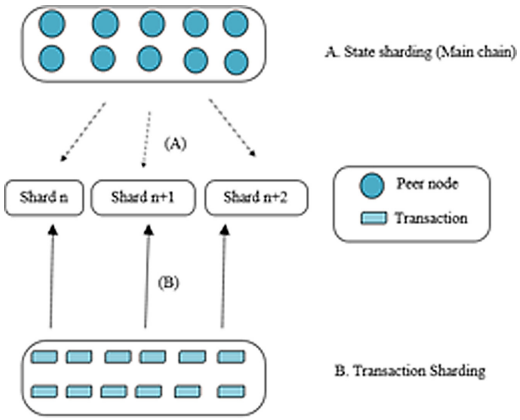

Fig. 2. Sharding architecture [1]

Sharding technology has been used in lots of blockchain projects including Elastico, OminiLedger, RapidoChain, and Monoxide [20–23]. An example of sharding architecture is shown in Fig. 2. The architecture shows that the blockchain network is divided into 3 shards using 3 procedures:

- firstly, each peer in the network (i.e. main net) is assigned to different shards. To reduce the storage overhead of each node, State sharding enables nodes in each shard to only store the state of their shard.
- Transaction sharding distributes transactions across the available shards and allows their parallel processing. Apart from transactions being executed within a single shard, cross-shard transactions are very common in a large system. Consequently, the system must be equipped with some protocols that would handle cross-shard transactions carefully and efficiently.

iii Consensus strategy

Consensus means agreement. So simply, consensus algorithms are those algorithms in a distributed or decentralized network that help them to unanimously take a decision when it is necessary without the need of a third party trusted authority [24]. Its major functions on the blockchain network include ensuring and maintaining decentralized governance, quorum structure, authentication, integrity, non-repudiation, byzantine fault tolerance, and performance [3].

They are critical to the way of working of the blockchain network as a result, the last category of proposed solutions to the scalability concern on the blockchain is the modification of the consensus mechanism. Popular consensus algorithms like Proof of Work(PoW), Proof of Stake and Distributed Denial of Service, Practical Byzantine Fault Tolerance, Hybrid Consensus, and others have gone through varying modifications and improvements geared towards optimizing their TPS and scalability [1].

The original PoW mechanism, for instance, has been improved through the Bitcoin-NG and GHOST projects. Bitcoin-NG improves on the original PoW by dividing time into epochs [1], a leader responsible for transaction serialization is assigned to each epoch. To support this improvement strategy, Bitcoin-NG introduced two types of blocks, namely: key block and microblock. The key block is generated by the miners through the PoW mechanism. It does not contain transaction data and is only used for the election of the leader. The leader is allowed to generate the microblock which contains the packaged transaction data. Consequently, transactions can be processed continually until the next leader that significantly reduces transaction confirmation time and improves the scalability is elected.

GHOST also builds upon PoW but in its case, it re-organizes Bitcoin's data structure to eliminate double-spending [25] attacks, a security concern caused by network delay which has to do with spending the same asset more than once. Another project which improves on PoW is SPECTRE [26]. It utilizes the structure called direct acyclic graph (DAG) to improve the transaction throughput and reduce the confirmation time of Bitcoin.

3 Design

This research proposed a scalable blockchain-based e-voting system (BEV) implementation. As indicated in the network view of our architecture in Fig. 5, our proposed system does not only achieve the end-to-end implementation of the electoral process but more importantly, showcases how scalability can be improved via the combination of lightweight peers and cache database. The e-voting system developed was built on the Ethereum Virtual Machine (EVM) which is the Ethereum blockchain network deployment environment.

The implementation model shown in Fig. 3 is made up of 4 layers interacting together. These include the presentation, business logic, data, and the Ethereum Virtual Machine tiers. The presentation layer which provides the user interface to the blockchain-based (DAPP) e-voting system, is the component via which voters can securely cast their votes. It has the Web3.js API which is a collection of JavaScript libraries that are used

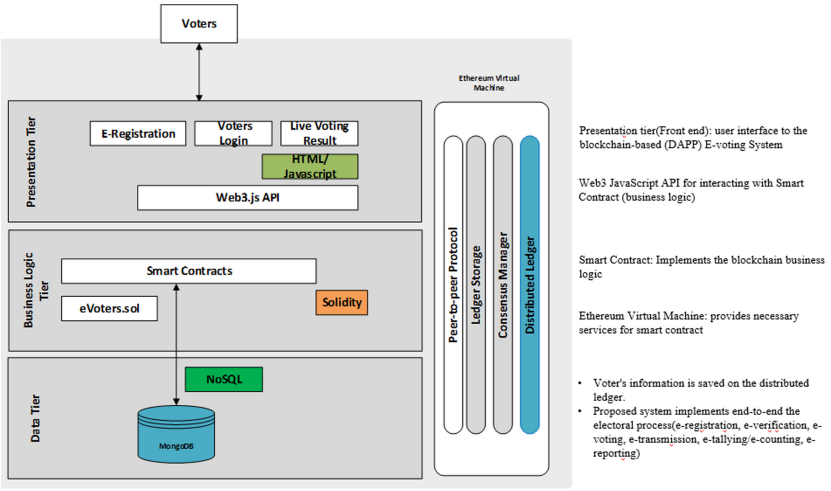

Fig. 3. Blockchain Implementation Framework for e-voting

for interacting with local or remote Ethereum nodes [27] through the smart contract. The business logic tier otherwise known as the application layer hosts the smart contract which implements the blockchain business logic. The data tier implements the state caching on the model's PU modules. The EVM provides critical services like peer-to-peer protocol, consensus manager for the management of the blockchain network.

Unlike the BEV model design [28] presented in Fig. 4, in which voter's registration and verification were done on a centralized system and only the cast votes are placed on the blockchain, the proposed design supports the end-to-end management of the entire electoral process (i.e. e-registration, e-verification, e-voting, e-transmission, e-counting, and e-reporting) on the blockchain. The proposed design recognizes how critical e-registration and voters' register are to the integrity of a voting result, hence, it implements the voter's database on the immutable, auditable blockchain ledger alongside the other steps in the electoral process. Figure 5 represents the architecture of the proposed design. It shows voters can interface with the proposed model through the voting device for registration and vote cast. Both the voter's details and their votes are signed, encrypted, and submitted on the blockchain network.

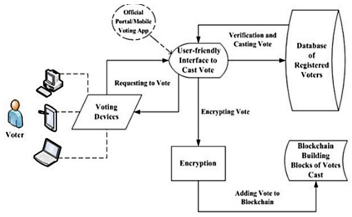

Fig. 4. Existing blockchain-based electronic voting system [28]

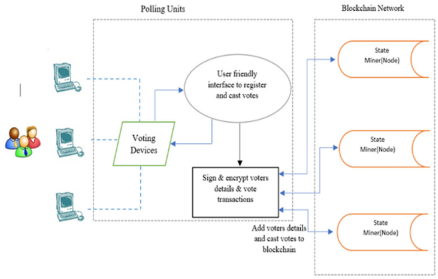

Fig. 5. Proposed blockchain-based e-voting model architecture

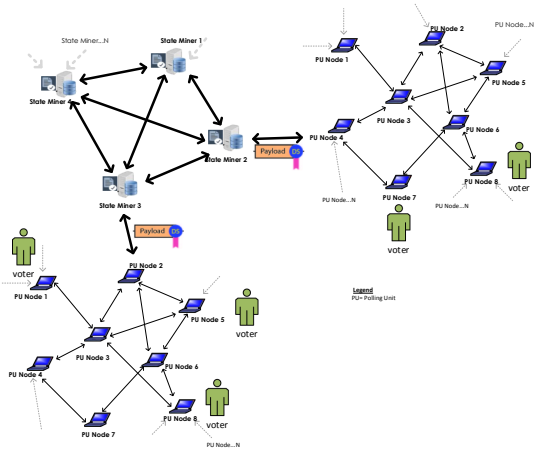

Fig. 6. Network view of the blockchain-enabled model for electronic voting system

The network view of the proposed model as shown in Fig. 6, is made up of two logical blockchains- Federal and State blockchains. This is to better handle latency issues related to querying the blockchain.

The Federal blockchain constitutes 36 state miners (representing the 36 states in Nigeria) and the Federal Capital Territory (FCT). While the State blockchain is made up of all the Polling Units (PU) in each state and the State Miner(node). Miners are nodes on the network with the capability to validate transactions.

The state miners are full nodes while the PU nodes are lightweight. Each full node has a complete copy of the distributed ledger (information about all registered voters and cast votes in the country), can validate new vote blocks, and verify transactions (i.e. cast votes).

For latency's purpose, PU nodes (i.e. lightweight nodes), don't store a full copy of the blockchain- they have a copy of the block header (i.e. metadata) to stay updated with the main blockchain and to verify transactions. Using the web3.js API, a copy of the blockchain with only eligible voters in every state is made available at the respective state PUs.

Communications between state miners and the PUs are digitally signed. The PU sends a payload that is signed with its public key. Hence during voting, voters get verified, then they cast vote. The cast votes are sent as a payload which is digitally signed with each PU public key. If for any reason, the PU loses connectivity to the blockchain, voting would be done offline, but verification gets done online later once connectivity is restored. A wrong actor engaging in double voting would be able to vote offline but when the PU connects, such votes would be invalidated because they had voted.

4 Methodology

The methodology includes the processes and methods followed to implement the proposed design above. The Nigerian Independent National Electoral Commission (INEC) was used as a case study for the blockchain-based model. Nigeria currently conducts her electronic verification(e-verification) using the Smart Card Reader (SCR) [29] and uses the INEC result viewing portal for collation of results as demonstrated in the 2020 Edo state gubernatorial election [30]. Electoral processes like e-voting, e-counting, and e-reporting are still not allowed under the Nigerian Electoral Law [31].

4.1 Setting up the Model

Ethereum network was used to set up the model. This is because, beyond the fact that Ethereum is open source and it provides blockchain developer the ability to build private blockchain networks on top of the ethereum environment when compared with other blockchain networks(Bitcoin, Litecoin, and Dogecoin), it outperforms them with respect to block-release time, transaction size, transaction rate, consensus model and consensus algorithm [32]. The following steps were taken to set up the proposed scalable blockchain-based electronic voting system for national election: Setting up the First Node on the Private Blockchain, Deploying the Smart Contract, Creating the Network, Deploying the Polling Units(PU) Module.

Setting up the First Node on the Private Blockchain
To set up our private blockchain (Dapp) called eVoters, we started with the first node among the three full nodes in our proposed model's proof of concept (PoC). The nodes which are called state miners are setup on the cloud:

eth-node1.codecounty.com, eth-node2.codecounty.com, eth-node3.codecounty.com. Each has Ubuntu Operating System (OS) installed on it. On top of the OS is installed the Ethereum "Geth" application. A secure shell protocol (SSH) is used to connect to each node from a Git bash tool:

$ ssh-i custom-eth.pem ubuntu@eth-node1.codecounty.com.

The "custom-eth.pem" file is the identity file with the public key of the eth-node1.codecounty.com node for a secured connection between the source machine and the blockchain node. Geth is installed to set up a custom/private ethereum node using the command: # sudo add-apt-repository -y ppa:ethereum/ethereum.

Below set of commands is used to update the repositories and install Ethereum alongside its command line (CLI) environment "Geth":

```
# sudo apt-get update.
# sudo apt-get -y install Ethereum.
```

Genesis Block

The genesis block holds the first/initial state of the entire network. In it, the network "blueprint" or "DNA" is determined. The genesis block is created using a genesis file. It is a JSON file with the required parameters to initialize the blockchain. Below is the genesis file used for eVoters:

```
{   "config": { "chainId": 19xx, "homesteadBlock": 0, "eip155Block": 0, "eip158Block": 0  },
    "difficulty": "10000", "gasLimit": "2100000",
    "alloc": {
        "e662…aa87": {"balance":"100000000000000000"},
        "e662…aac6": {"balance":"100000000000000000"},
    }  }
```

The chain is a unique identifier for the network and is used in the signing and validation of transactions. The difficulty is used to specify the hash iteration, this directly affects the rate at which new blocks are mined. Note that 10,000 is low difficulty number, "gasLimit" specifies the maximum number amount of gas (fees) allowed for a transaction, while "alloc" pre-funds specified accounts with "Ether".

Once the genesis file has been created the blockchain and genesis block is initialized using the command below: geth --datadir "./db" init genesis.json.

With the above command, a new chain is created and committed in the database. To run the newly created blockchain node use the following command:

```
geth --datadir "./db" --networkid 19xx
```

With this, the first blockchain node is deployed successfully.

Deploying the Smart Contract

An Ethereum based decentralized application isn't complete without the smart contract. A smart contract is a self-executing contract. It is built to enforce an agreement between multiple parties, in this case, an election.

To build and deploy a smart contract, remix and ethfiddle were used as recommended tool. EthFiddle is a browser-based solidity Integrated Development Environment (IDE) which allows users to write and test their solidity codes without deploying to a blockchain while "remix" (which is also a browser-based IDE) generates the bytes codes used to deploy the smart contract and Application Binary Interface (ABI) data which used to interact with the smart contract over the RPC protocol.

Upon generating the byte code of the contract on "Remix" a transaction is created and broadcasted to the network and in response, a transaction hash is generated, this hash acts as a transaction reference and can be used to get the transaction details.

The contract byte data is first read from a file and is used to create a transaction object. The transaction is then signed using a private key and sent to the blockchain network.

Upon propagation, a view into the transaction using the transaction hash generated above would reveal the new contract's address. This address is copied to the "app-config.json" file of the NodeJS application.

Creating the Network

To create a network, first, a new node has to be created, then a second node to communicate and synchronize with the first node. There are different ways to go about this. A simpler alternative is reusing the exact same genesis file and chain ID. Once the node has been created, run geth --datadir "./db" --networkid 1946 console to initialize the node and access its console.

Deploying the Polling Units (PU) Module

So far, we have set up and deployed the state miners on the blockchain network. This is important as the PU module will need the state miner blockchain network for connectivity, voter's verification, validation of cast votes and adding them to the blockchain.

As represented in Sect. 3, the PU is the module of our proposed model that is deployed at various polling units across the country. For our model to run optimally without feeling the full effect of network instability and latency, the PU module is set up with MongoDB to cache voting transactions on the voting device used at every PU. It also has a file called "app-config.json" which the PU module uses to communicate and sync with the federal blockchain network. The file contains critical configuration information like the provider's (parent node) RPC information (URL and port number), the private key, and address of the PU, alongside the application's configurations and smart contract address.

As a proof of concept, the PU module was installed on ten laptops operated by Service Integrators (SIs). These SIs are tech-savvy, well-educated and at different locations. Their laptops have the same specifications except one as indicated in Table 3. From records obtained from these SIs, it takes a maximum of three minutes to get it installed and to set up the PUs for voter verification and voting.

To set up the PU device, the SIs downloaded the PU module from the OneDrive repository, extract and navigate it to the folder where the packaged PU is stored, double click on the "run.bat" file. This installs and connects the PU device to the state miner that it has been configured to connect within the config.json file.

Once the setup is completed, the eVoter web module interface is loaded on port 5000 as shown in Fig. 7 below.

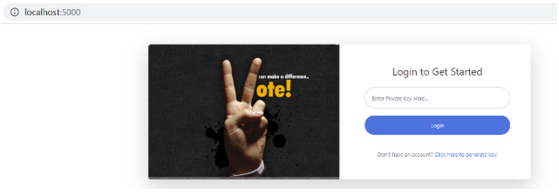

Fig. 7. eVoter web interface

The config.json does not just specify the address of the state miner that each PU should connect to, it also specifies other critical parameters including the private key and address of the state miner the PU is connecting to, contract details; the MongoDB configuration details; the provider parameter defines the state miner which the PU is securely communicating with. This variable also determines the state in which the PU is. MongoDB is a NoSQL (Structured Query Language) Database Management System.

Performance Impact of Proposed Model Design
The proposed design is meant to improve the scalability performance matrix of the blockchain-based e-voting system. The method for implementing this matrix is presented below.

Scalability Matrix
For the purposes of secured communication and scalability, the model design adopts the use of the Rivest, Shamir, Adleman (RSA) encryption scheme. All PUs are set up on-demand to communicate with the state miners on the blockchain network by utilizing the RSA scheme. This means that communication between state miners and the PUs is digitally signed. The PU sends a payload that is signed with its private key. During voting, voters get verified, before voting. The cast votes are sent as a payload which is digitally signed with each PU's private key. If for any reason, the PU loses connectivity to the blockchain network, voting could still be done offline, but verification and validation get done online later when connectivity is restored.

From the scalability standpoint of the proposed model performance, we considered how long it takes to deploy a new PU and the cost of the computing device including the cost of Uninterrupted Power Supply(UPS) and MODEM/subscription for internet connectivity as critical factors. As represented in Sect. 4.1.5, to set up PUs, pre-packaged PU modules are transferred, extracted, and then installed on the computer systems designated as PU nodes. The result from the scalability evaluation done with the ten volunteering SIs from a multinational telecommunications company, each setting up a PU is captured in Table 3.

Overall, the Keccak algorithm, MongoDB, and Web3.js are used to enhance the overall performance of our model by reducing response time between the PU nodes and the state miners on the blockchain network. Also, the more nodes that are added to the blockchain network, the more secure the network becomes since it relies on consensus to verify transactions and validate blocks.

5 Results and Discussions

Base on the aim of this research work which is to advance a scalable blockchain-enabled electronic voting model, we present below performance results from the experimentations conducted with respect to the scalability of our model.

5.1 Scalability

To evaluate the system scalability, we considered the time and cost of setting up the PUs. To set up PUs, we packaged the PU application module, port and install the packaged

file created on the computer system designated as PU node. To conduct the scalability evaluation, we engaged ten system integrators (SI) from a multinational telecommunications company that volunteered to participate in setting up a system each as PUs, making ten PUs in effect. Table 3 represents how long it took each of them to set up each PU.

Table 3. Evaluation of systems integrators' setup time

SI ID	Device type	Device RAM size	Processor spec	Estimated cost price (NGN)	PU name	Download time (min)	Setup time (min)
SI1	HP Laptop	16GB	CORE i5	300,000	LA-Ajah	3:00	2:37
SI2	HP Laptop	16GB	CORE i5	300,000	LA-Surulere	4.00	2:50
SI3	HP Laptop	16GB	CORE i5	300,000	LA-Alimosho	2:11	3:42
SI4	HP Laptop	16GB	CORE i5	300,000	R-Ikwerre	3:10	2:31
SI5	HP Laptop	16GB	CORE i5	300,000	R-Bonny	2:20	1:67
SI6	HP Laptop	16GB	CORE i5	300,000	R-Okrika	1:00	2:90
SI7	HP Laptop	16GB	CORE i5	300,000	K-Municipal	3:00	2.50
SI8	HP Laptop	16GB	CORE i5	300,000	K-Gwale	2:00	2.45
SI9	HP Laptop	4.0GB	Pentium	130,000	K-Dala	4:00	3:00
SI10	HP Laptop	16GB	CORE i5	300,000	A-Abaji	2:01	2:17
Average							2.64

From the recorded experiences of the ten SIs who volunteered as electoral technical officers-in-charge of the PUs, it takes an average of 2.64 min to get a PU ready once the module is on hand. Once the PU is set up, voter's accreditation and voting can start immediately provided it is time to start voting as scheduled by the election management body (i.e. the INEC). From experiences in elections in Nigeria, the INEC faces varying difficulties in starting polls at the stipulated time. Apart from logistics challenges like transportation of the INEC staff and sensitive materials to the PUs, at the PUs, it takes a lot of time and effort to get them ready to commence voting. This often leads to frustration and sometimes, the disenfranchisement of the electorate.

As seen in the result obtained in Table 3, we presented how the proposed system can be set up on an average of 2.64 min. This is a far cry from what is obtainable in the

existing system manned by the INEC where it takes an average of 10 min to set up or reconfigure the Smart Card Reader (SCR).

The significance of the proposed model setup time is not just in the ability to set up new PUs but also in the ease of deploying replacements for malfunctioning PU modules. This ability to easily scale out is very critical in ensuring that voters are not disenfranchised on account of delays in getting the PUs ready for suffrage.

5.2 Assumptions

i. All peers on the network have internet connectivity
ii. Miners do NOT have equal system specification- some miners could have higher compute capacity than those in polling units
iii. Miners and lightweight nodes are added to the private blockchain network on demand
iv. All miners have a copy of the blockchain
v. All payloads are digitally signed.

5.3 Conclusion

In this paper, we focus on how to improve the scalability of a blockchain-based e-voting system. We conducted a review of four major blockchain networks (Bitcoin, Litcoin, Dogecoin and Ethereum). The review showed the Ethereum network has better block-release time, transaction size, transaction rate, and consensus algorithm. As a Proof of Concept, a private blockchain-based e-voting system was set up using cloud infrastructure and lightweight nodes called PUs across. The rate of spinning up the ten PUs was recorded. The average PU spin-up time (scalability) was compared with INEC's existing system PU setup time. Our model's was found to be much better.

References

1. Zhou, Q., Huang, H., Zheng, Z., Bian, J.: Solutions to scalability of blockchain: a survey. IEEE Access **8**, 16440–16455 (2020)
2. Sarah, G.: Blockchain- What is the Business Value?. A paper presented at the Oracle Modern Business Talks, Webinar (2018)
3. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: architecture, consensus, and future trends. In: IEEE 6th International Congress on Big Data, pp. 557–564 (2017)
4. Vaibhav, et al.: literature survey- online voting: voting system using blockchain. Int. Res. J. Eng. Technol. **6**(6), 534–536 (2019)
5. Komminist, W., Adolfo. V., Andrea, M.: Experiments and data analysis of electronic voting system. IEEE, pp105–112 (2009)
6. Hao., F., Ryan, P.Y.A.: Practical attacks on real-world e-voting. In: Hao, F., Ryan, P.Y.A., (eds.) Real-World Electronic Voting: Design, Analysis and Deployment, pp. 45–196. Auerbach Publications, US (2016)
7. Ayo, C.K., Adebiyi, A.A., Sofoluwe, A.B.: E-voting implementation in nigeria: the success factors. Int. J. Comput. Sci. Appl. **15**(2), 91–105 (2008)

8. Sheriff, F.F., Ayo, C.K., Oni, A.A., Gberevbie, D.E.: Challenges and prospects of e-elections in Nigeria. In: Proceeding of 14th European Conference on E-Government, Brasov Romania, 13–14 June 2014
9. Mohammad, H.S., Essam, M.: A secure e-Government's e-Voting system. Sci. Inf. Conf. **44**(12), 1365–1373 (2015)
10. Gharadaghy, R., Volkamer, M.: Verifiability in electronic voting explanations for non security experts. 4th International Conference on Electronic Voting 2010 EVOTE 2010 **4**(4), 1–13 (2010)
11. Dannen, C.: Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress, Brooklyn, New York, USA (2017)
12. Deloittecom. Deloittecom. Retrieved from 20 Nov 2017. https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Technology/IE_C_BlockchainandCyberPOV_0417.pdf
13. Philip, B.: European Union Parliamentary Research Services; How blockchain technology could change our lives. European Union, Brussels (2017)
14. Ansif, A., Mohsin, R.: Electronic voting with biometric verification offline and hybrid EVMS solution. The Sixth International Conference on Innovative Computing Technology (INTECH 2016), pp. 332–337 (2016)
15. Cortie, V., Constantin, C.D., Francois, D., Benedikt, S., Pierre, Y.S., Bogdan, W.: Machine-checked proofs of privacy for electronic voting protocols. In: 2017 IEEE Symposium on Security and Privacy, pp. 993–1008 (2017)
16. Blockgeekscom. What is blockchain technology?. Retrieved from 17 July 2020. https://blockgeeks.com/guides/what-is-blockchain-technology/ (2020)
17. Abayomi-Zannu T.P., Odun-Ayo I., Tatama B.F., Misra S.: Implementing a mobile voting system utilizing blockchain technology and two-factor authentication in Nigeria. In: Singh, P., Pawłowski, W., Tanwar, S., Kumar, N., Rodrigues, J., Obaidat, M. (eds.) Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019). Lecture Notes in Networks and Systems, vol. 121. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3369-3_63
18. Awotunde, J.B., Ogundokun, R.O., Misra, S., Adeniyi, E.A., Sharma, M.M.: Blockchain-based framework for secure transaction in mobile banking platform. In: Abraham, A., Hanne, T., Castillo, O., Gandhi, N., Rios, T.N., Hong, T.-P. (eds.) Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020, pp. 525–534. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-73050-5_53
19. Lombrozo, E., Lau, J., Wuille, P.: Segregated witness (consensus layer) (2015). https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
20. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, L.P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 17–30. ACM (2016)
21. Kokoris-Kogias, E., et al.: IEEE symposium on security and privacy (SP). IEEE **2018**, 583–598 (2018)
22. Zamani, M., Movahedi, M., Raykova, M.: Rapidchain: scaling blockchain via full sharding. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 931–948. ACM (2018)
23. Wang, J., Wang, H.: Monoxide: Scale out blockchains with asynchronous consensus zones. In: 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19), pp. 95–112 (2019)
24. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Generation Computer System (2017)

25. Sompolinsky, Y., Zohar, A.: Secure high-rate transaction processing in bitcoin. In: Böhme, R., Okamoto, T. (eds.) Financial Cryptography and Data Security, pp. 507–527. Springer Berlin Heidelberg, Berlin, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47854-7_32s
26. Sompolinsky, Y., Lewenberg, Y., Zohar, A.: Spectre: a fast and scalable cryptocurrency protocol. IACR Cryptol. ePrint Arch. **2016**, 1159 (2016)
27. Web3js.readthedocs.io. 2021. web3.js - Ethereum JavaScript API — web3.js 1.0.0 documentation. https://web3js.readthedocs.io/en/v1.3.4/. Accessed on 15 June 2021
28. Dogo, et al.: Blockchain 30: Towards a secure ballotcoin democracy through a digitized public ledger in developing countries. In: 2nd International Conference on Information and Communication Technology and Its Applications (ICTA 2018), pp. 477–484 (2018)
29. INEC (2015). www.inecnigeria.org. www.inecnigeria.org/wp./FactSheet-on-PVC-and-Card-Readers.docx
30. Independent National Electoral Commission. INEC - Result viewing portal. Retrieved from 4 Oct 2020. https://inecelectionresults.com/
31. Independent national electoral commission. Regulations and Guidelines For The Conduct of Elections. Retrieved from 6 Oct 2020. https://www.inecnigeria.org/elections/regulations-and-guidelines-for-the-conduct-of-elections/
32. Hintzman, Z.: Comparing Blockchain Implementations. NCTA Technical Papers. Nctatechnicalpapers.com. https://www.nctatechnicalpapers.com/Paper/2017/2017-comparing-blockchain-implementations. Accessed on 16 June 2021