ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 11 , November 2021

An implementation of Blockchain based smart contract model for Tendering

Denis Kiyeng

P.G. Student, Department of Computer Science, Kabarak University, Nakuru, Kenya

ABSTRACT: Blockchain technology is the foundation for secure and non-immutable transactions. Today, every organization is investing in security and trust ecosystem powered by blockchain. Going into the future, the blockchain technology will permeate almost every sector, procurement included. It is possible for tendering process to be disrupted by BYOE (Bring Your Own Encryption) concept while ensuring bidding can be done with minimum interference through corruption. In this paper we have come up with a blockchain based smart contract model using Ethereum. The model was hosted locally running on 127.0.0.1:9545 and MetaMask installation, were configured for the running environment in order to provide a functional prototype. In addition, Ganache running on localhost 127.0.0.1:7545 for graphical interface was realized for model testing and validation. In testing the functional model, two organizations were involved and upon using the system, they were able to carry out their tendering process with successive chaining and token generation thus ensuring secure participant data protection. The running hosted model is available at https://gov.chimera-iot.co.ke/index.php .The model was able to automatically initiate the tender, place bid and perform bid evaluation & relay bid results with relevant bid report.

KEY WORDS: Bid, Evaluation, Ethereum, network, blockchain

I.INTRODUCTION

The world businesses order is changing at a fast pace, business entities are reengineering their business processes and improving the efficiency of operations. The essential concept about having non-immutable transactions will redefine the distributed ledger landscape determining who controls or leverages the opportunities brought by the new technology infrastructure. More so, with ever increasing space of emerging technologies in cyber space realm. Soon, the competitiveness on how businesses are connected is going to depend on the type or speed at which they integrate new technology. Organizations will be gaining a differentiating factor from each other depending on how they invest on the blockchain and related emerging technologies.

The decentralized ecosystem offered by blockchain is based on P2P networks. Which has attracted wide attention in distributed application systems in recent years (Guo, *et al*., 2021). It provides a tamper proof resistant digital platform by applying the chain-block structure and establishing a trusted consensus mechanism to synchronize data changes. At the same time, the decentralization, traceability and non-immutability of on-chain information storage makes blockchain a trusted machine with high reliability and security (Li, *et al*., 2021).

Already working use cases have been realized in various fields, such as the Internet of Things, supply chain management and voting system. It has been proved that applications based on Blockchain can improve the availability of data and reduce costs, while maintaining the openness and transparency of the application (Liao, *et al*., 2021).

The proposed use in procurement is poised to ensure accountability of public officials (Shin, and Ibahrine, 2020). Therefore organizations, using these technology will raise success, reduce risks in complex contracts, strengthen procurement and contracting practice while holding officials accountable and in general, strengthening the tendering process. Additionally, it provides a procurement infrastructure which is more open to competition through public examination and review, thus making it a transparent process (Erdmenger, 2017).

In recent years, many governments emphasize organization and public entities to adopt Electronic bidding (E-bidding), with core aim of ensuring open and safe bidding environment (li, *et al*., 2021). In 2020, the government of Kenya published the public procurement and asset disposal regulation providing detailed guidance and elaborate procedures for establishment and use of e-procurement systems.

It is therefore, primetime for organizations as it re-engineers its process to integrate blockchain technology. Not long, it will be supporting bigger number of business use cases in diverse industry wide operations. As blockchain is expanding more customers are able to realize its flexibility and impact as it powers up many enterprises.

II. SIGNIFICANCE OF THE MODEL

Tendering process is vital for stakeholders, especially when it can be traced later after bid closure. Tendering inefficiencies cost organizations a huge chunk of cash in delayed purchases, missed discounts and transaction disputes (Åkers, 2018). The working procurement ecosystem currently is based on paper and tender box. It depends on the goodwill of tendering department because the outsiders are locked out from participating during the entire process of receiving bids. There is no working mechanism to establish a secure digital infrastructure that deters the tendering organization from trapping the bids inside the business process. Hutchinson (2015) also agrees that regardless of more elaborate tendering rules, still corruption and manipulation continue to degrade the entire tendering process due to lack of a tamper proof model and framework in place. On the other hand, modern business integrate blockchain to establish trust based technology mechanism (Korpela, *et al.,* 2017). Nevertheless tendering still lack a model based on Blockchain based smart contract has a tool to establish trust.

III. LITERATURE SURVEY

A. The Blockchain technology

A blockchain, originally block chain (Makupi, 2020), is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data, generally represented as a Markel tree root hash (Iyidogan, 2018).

By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way". For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks, which requires consensus of the network majority (li., *et al*., 2018). Although blockchain records are not unalterable, blockchain may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance. Decentralized consensus has therefore been claimed with a blockchain (KAFOL *et al*., 2018).

Constantly growing as ‘completed’ blocks are recorded and added to it in chronological order, it allows market participants to keep track of digital currency transactions without central recordkeeping. Each node gets a copy of the blockchain, which is downloaded automatically. Within the blockchain is a block that is the ‘current’ part of a blockchain, which records some or all of the recent transactions (Makupi, 2020). Once completed, a block goes into the blockchain as a permanent database. Each time a block gets completed, a new one is generated. There is a countless number of such blocks in the blockchain, connected to each other in proper linear, chronological order. Every block contains a hash of the previous block.

B. Requirements for Model Development and Implementation

In the implementation of the tendering process, the study used Ethereum blockchain API, because Ethereum is an open-source platform that is publicly available and a well-known choice for developing distributed applications. With Ethereum, each transaction has a ‘gas’ usage and ‘gas cost’ (Dannen, 2017). The ‘gas’ usage is determined by how computationally expensive the transaction is. The purpose of working out the ‘gas’ used in each transaction comes down to the fact that every node in the blockchain verifies the transaction.

If transactions were allowed to be arbitrarily complex, verification on the network would be slow and result in a processing bottleneck (Peters &Panayi, 2016). Citing ‘gas’ needed for each transaction allows miners to determine whether or not it is worth including the transaction in the block that they are mining and they will use the cost of ‘gas’, as set on each transaction by the node that pushed it, to determine this. To mine one of these blocks, a node must satisfy

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 11 , November 2021

the proof-of-work constraint (Ethereum choice for the consensus protocol). This constraint necessitates a certain degree of work to be undertaken by the node that wishes to push a block to the wider chain. This stops a node from pushing an arbitrary number of blocks to the chain that would be computationally unfeasible (Thampi, 2010). Table1below outlines the requirements for model development.

Requirement	Function
Truffle JS	Is used to create our contracts and write them in the Solidity language.
Private blockchain	It runs on the TestRPC Ethereum client, is used for the unit tests during development.
Ropsten testing network	Used for operational performance tests. This network most closely resembled the Ethereum production environment.
Infura.io node	Used to connect to Ropsten.
Blockchain explorer	Used to connect to Ropsten
Blockchain explorer-	Used to interact with the contracts in **My Ether Wallet** using an account made via the **Meta Mask** chrome extension.

Overall, the model is categorized according to the subsections discussed below. The model subsections sections are organized according to achievable functions;

i. **The smart Contract -** To write an Ethereum smart contract Solidity programming language is used. It's a general-purpose programming language that uses a class (*contract*) and methods that define it. Solidity main purpose is to send and receive digital tokens as well as storing states.

ii. **Ethereum wallets-** To fully demonstrate the research objectives, we will create three types of wallets on Ethereum: for the bidder, for the tendering organization, and for the contract respectively.

iii. **Dapp** - The front end users will use Dapp to connect with blockchain via the Smart Contract network (Front End → Smart Contract → Blockchain). The ullustration on how the front-end users connect with the Dapp applications is shown below in figure 12.

IV. METHODOLOGY

Towards realizing the concept at viable level of usability and security, the model was developed with web3 JSON RPC, metamask, ganache, truffle framework, and node.JS and geth client. The model functional overview entails;
First, creating a bare Truffle project temple using Truffle commands, run them against an existing Truffle project. Once this operation is completed, a new directory for your Truffle project is created. Truffle requires that you have a running Ethereum client which supports the standard JSON, RPC and API. MetaCoin box is used to create a token that can be transferred between accounts. When this process is done, you'll now have a project structure with the following items: `Contracts` directory for Solidity contracts, `migrations` directory for scriptable deployment files, `test` directory for test files for testing the application and contracts and `truffle.js` for truffle configuration file.

Each platform has an associated `npm run` configuration to help build on each platform. Because each platform has different (but similar) build processes, they require different configuration. The detailed outline steps to demonstrate the transaction in process is as shown in the next section.

V. MODEL IMPLEMENTATION

To successfully demonstrate this transaction process the following steps were followed

Step 1: Truffle with MetaMask installation

For development with Truffle this means Dapp is used the same way users will interact with it on a live network. When interacting with smart contracts in a browser, make sure they are compiled, deployed, and interacting with them via **`web3`** in client-side JavaScript. Truffle-contract library is used, as it makes interacting with contracts easier and more robust.

MetaMask is the way to interact with Dapps in a browser. It is an extension for browser (Chrome or Firefox) that connects to an Ethereum network without running a full node on the browser's machine. It connects to the main Ethereum network, any of the testnets (Ropsten, Kovan, and Rinkeby), or a local blockchain such as the one created by Ganache or Truffle Develop. Once the MetaMask is installed, the front-end is ready to be used and to see the dapp running.

i. Using MetaMask with Ganache

Ganache is a graphical application that runs a blockchain that is used for testing purposes. It runs on `127.0.0.1` instead of `localhost` because the address does not require a network connection and so is more suitable for development.

ii. Detecting MetaMasks's web3 injection

Dapp is needed to checking for MetaMask's **web3** instance and that the extension itself is configured properly with Ganache. MetaMask injects its own **web3** instance, after the window has loaded, the following checks are performed:

```
// Is there is an injected web3 instance?
if (typeof web3 !== 'undefined') {
  App.web3Provider = web3.currentProvider;
  web3 = new Web3(web3.currentProvider);
} else {
  // If no injected web3 instance is detected, fallback to Ganache.
  App.web3Provider = new web3.providers.HttpProvider ('http://127.0.0.1:7545');
  web3 = new Web3(App.web3Provider);
}
```

List 1: injected web3 instance

iii. Setting up MetaMask

To use Ganache with MetaMask, MetaMask icon in your browser is selected and this screen will appear:

Create Password

New Password (min 8 chars)

Confirm Password

CREATE

Fig :MetaMask initial screen

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 11 , November 2021

MetaMask is then connected to the blockchain created by Ganache. At the menu that shows "Main Network", **Custom RPC** is selected.

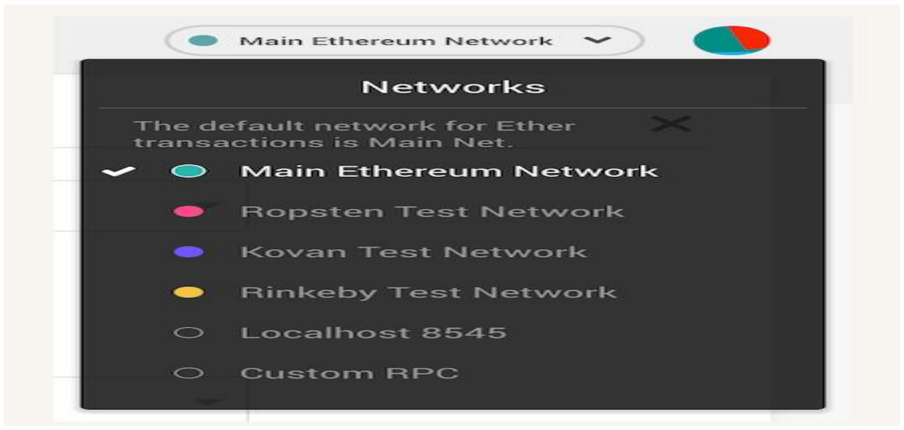


Fig 2: MetaMask network menu

Once MetaMask is connected to Ganache, it is taken to the accounts screen. Each account created by Ganache is given 100 ether. The first account should have less than the others because that account supplies the gas for smart contract deployment. Since you've deployed your smart contract to the network, this account paid for it.

iv. Using MetaMask with Truffle Develop

Truffle Develop is a command-line application that runs a temporary blockchain that is also used for testing purposes. It runs on `127.0.0.1:9545`. Using MetaMask with Truffle Develop is very similar to that of Ganache. The only

```
// Is there is an injected web3 instance?
if (typeof web3 !== 'undefined') {
  App.web3Provider = web3.currentProvider;
  web3 = new Web3(web3.currentProvider);
} else {
  // If no injected web3 instance is detected, fallback to Truffle Develop.
  App.web3Provider = new web3.providers.HttpProvider('http://127.0.0.1:9545');
  web3 = new Web3(App.web3Provider);
}
```

List 2: web3 code instance

difference is that Truffle Develop runs by default on `127.0.0.1:9545`, so the web3 code is as shown below:

Step II: Development of the tender process Module

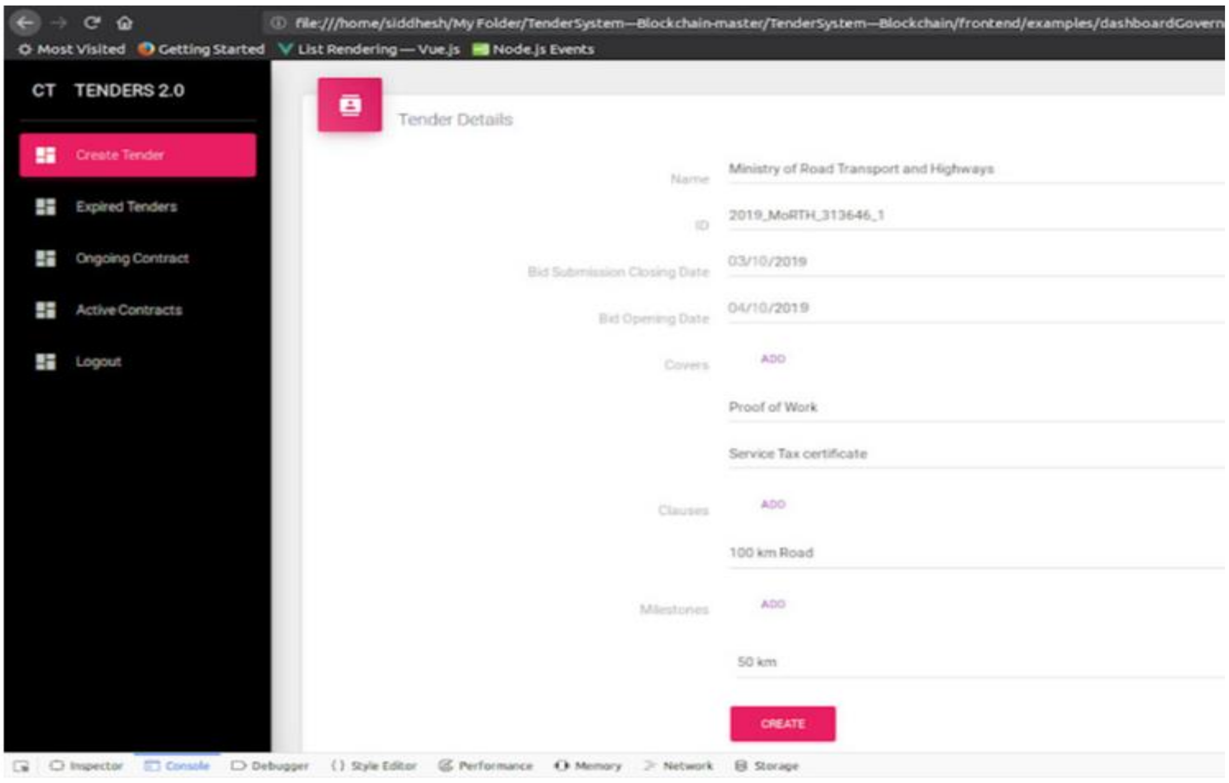
The module is sectioned in processes has outlined below;

Process I: Initiating a Tender

The Request for Tender (Process 1) is initiated by a contract placed on the blockchain. This contract is created with a length, given in milliseconds, that determines how long the contractors (also referred to as Tendering Organization: TO) have to place their bids. This time is calculated using the Unix Epoch time at the time of creation. An upper limit is also given, which is used to control the number of tenders a contractor can place for this auction. The entity that created the auction is also required to pass in a public key that is specific to this request for tender contract. These three attributes are held in the contract's state.

```
1: procedure REQFORTENDER (_length,_pubk,_limit)
2: biddingEnd ← TimeNow() +_length
3: limit ← limit
4: pubk ← pubk
```

List 3: Key authentication


file:///home/siddhesh/My Folder/TenderSystem—Blockchain-master/TenderSystem—Blockchain/frontend/examples/dashboardGovern

Tender Details

Name: Ministry of Road Transport and Highways
ID: 2019_MoRTH_313646_1
Bid Submission Closing Date: 03/10/2019
Bid Opening Date: 04/10/2019
Covers: Proof of Work; Service Tax certificate
Clauses: 100 km Road
Milestones: 50 km

Fig 3: Initializing a tender

Process II: Placing a Bid
When a contractor places a tender (Process 2), the first step is to place a smart contract elsewhere on the blockchain that contains only the tender data. Experiments on the Ropsten test blockchain showed that 5000 bits could be stored on the blockchain with a gas consumption that didn't get immediately rejected for being too expensive. This roughly translates to 700 words. This study suggested that the usually more verbose tender contracts are adapted for this format, perhaps by extracting the key information (PTO) necessary for the tender. The data on this contract is available to anyone maintaining the blockchain, and so is protected through the encryption performed using the bidders bid-specific symmetric key sealed by PTO (SKBidder).
Placing the tender for the auction involves using the auction smart-contract. The contractor passes in their ID, which should have been pre-agreed between the auction creator and the contractor, the address of their tender data on the blockchain, and then the components of the certificate that will have been given to them by the auction creator. This certificate should be signed using the private key that corresponds to the request for tender's public key that is put into the contract upon creation. To offload some of the computational complexity of the contract, the parts of this certificate which is the hashed message, and the v, r, and s values are extracted on the client side. The request for tender will take the address of all placed tenders and store it in an array.

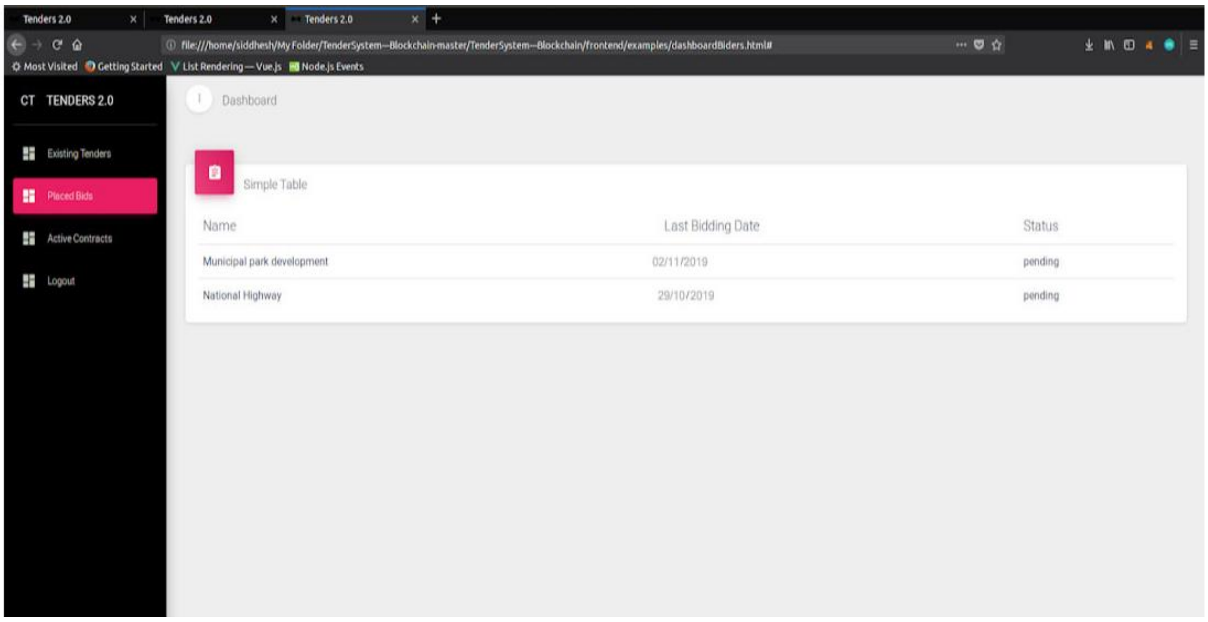
All tenders left on the contract are recorded, but the validity is determined using the certificate, the time that the tender was placed, and the record of bids placed. The time when the tender was placed is retrieved using the now () function available on an Ethereum contract. The now () function uses the time from epoch registered on the node executing the transaction.

This timing can be considered trusted because of the timing consensus of nodes on the blockchain. Nodes are unable to mine blocks that have timestamps earlier than the parent blocks time stamp. Nodes are discouraged from placing blocks at an arbitrary amount of time ahead for the same principle, few nodes on the chain is willing to put a block on the chain far ahead of their current time because it is unable to place new blocks on top of it, thus stopping the block from getting picked up by a majority of nodes. Nodes are also discouraged from moving the time back because the challenge issued in the proof-of-work is orders of magnitude

The bid is considered valid if the following hold;
i. The contract is placed within time
ii. The certificate is verified to match the public key held by request for tender contract
iii. The contractor is allowed to place more bids.

If it fails any of these checks, it is still placed but flagged as an invalid bid. The number of bids placed by the contractor is only incremented if it is being placed using a valid certificate, protecting the contractor from being locked out of the auction by a malicious party placing arbitrary numbers of bids using their identifier.

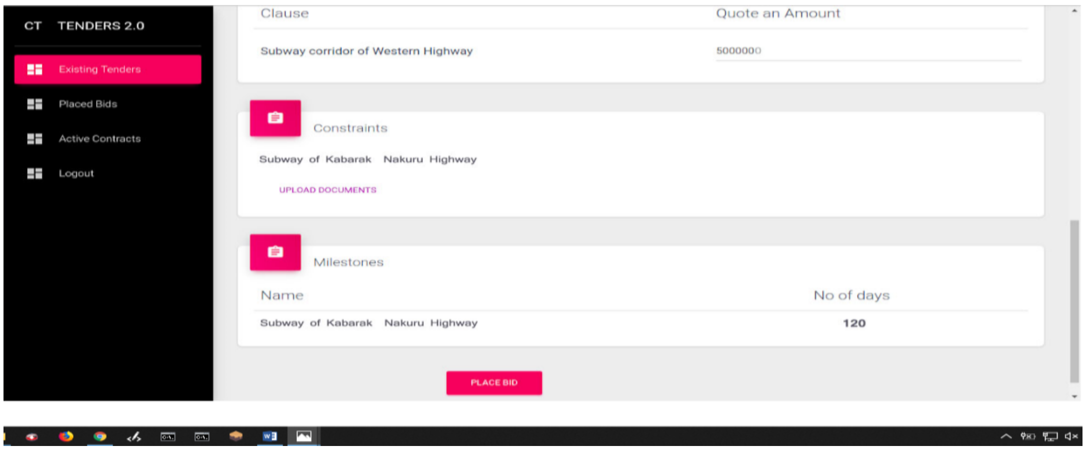
ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 8, Issue 11 , November 2021



Name	Last Bidding Date	Status
Municipal park development	02/11/2019	pending
National Highway	29/10/2019	pending

Fig 4: Bid verification

The tender reference smart contract is then created and passed the id of the contractor, the address of the tender data, the validity of the contract, a copy of the array holding all of the addresses of the tenders placed before this one, and the auction end time. The address table is required to ensure the integrity of the array revealed by the auction creator – no bids could be intentionally erased from the array because the record of its existence will exist in the tender reference contracts.



Clause	Quote an Amount
Subway corridor of Western Highway	5000000

Subway of Kabarak Nakuru Highway

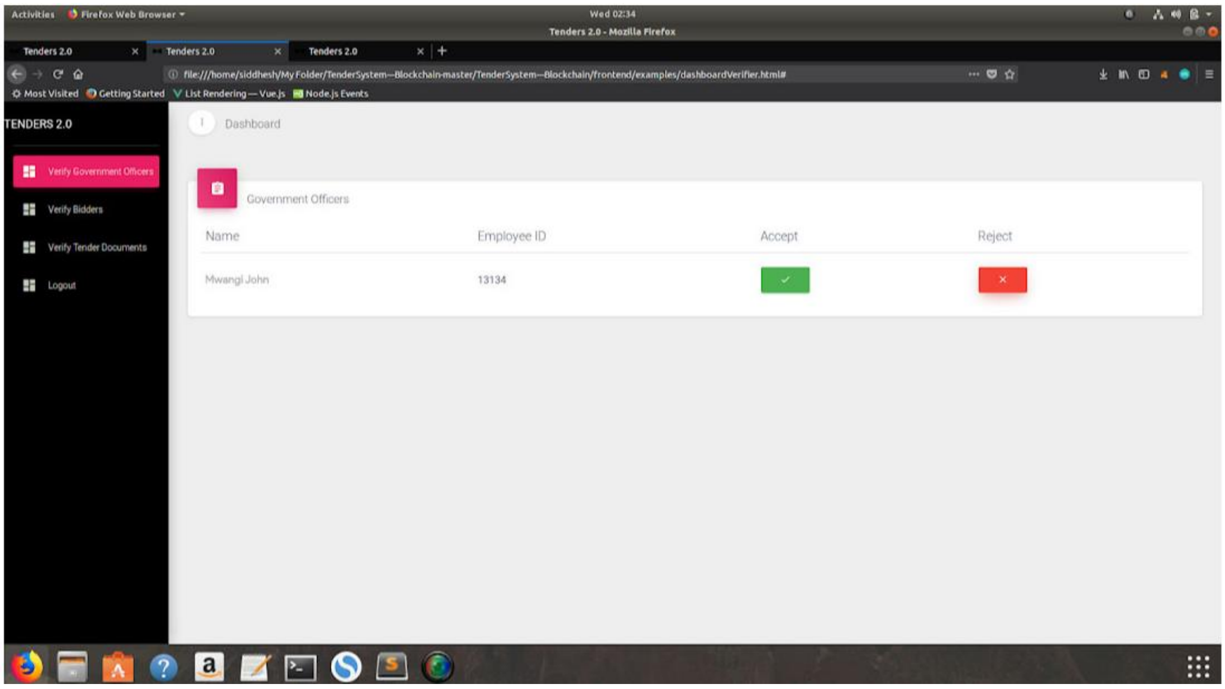
Name	No of days
Subway of Kabarak Nakuru Highway	120

Fig 5: Bid verification

Once the auction time has elapsed, the addresses can be requested from both the request for tender contract and the tender reference contracts. Process 2 performance indicates that altering the contract's state to hold the addresses increases the gas usage, and thus the cost, on each subsequent transaction.

Process III: Bid Evaluation

Process 3 is a retrieval algorithm that is only applicable to the schemas where the addresses are held in the state of the Request for Tender contract. The addresses would be retrieved from the contract after the request for the tender period has elapsed. The client application is used to interact with the blockchain will request the contract on the blockchain. Retrieving the information does not require any transactions and thus incurs no transaction cost.



Name	Employee ID	Accept	Reject
Mwangi John	13134	✓	×

Figure 6: Bid evaluation

The client application will receive a list of bids which is the addresses of the bids placed. These bids can then be queried for their validity and, once ascertained, the address of the actual tender data can be requested.

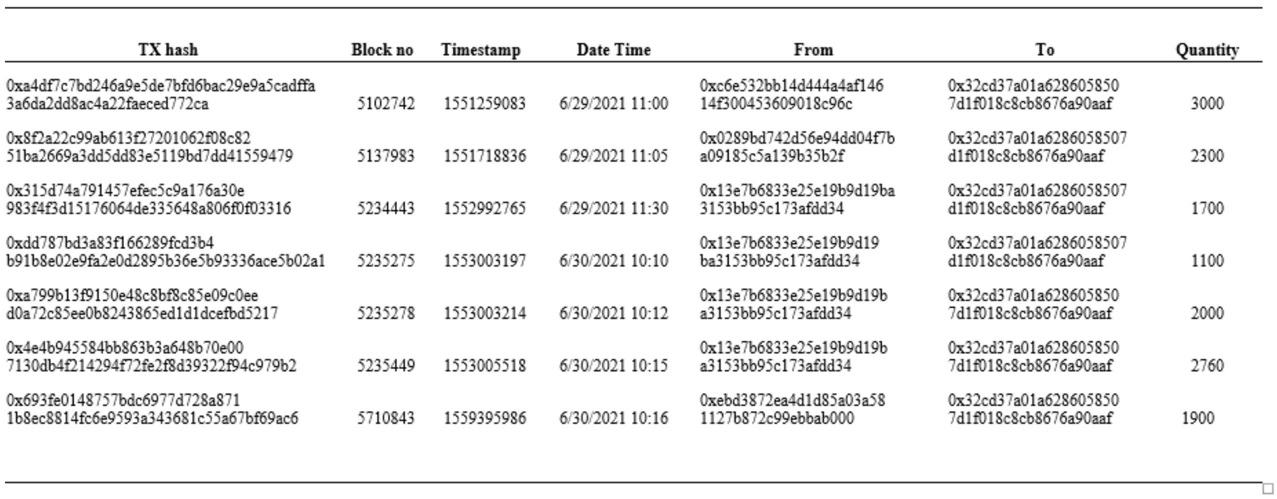
Organizational Sample Report

As shown in the table below, it can be noticed that Organization A registered 6 bidders into the platform and all of them submitted their bids using ETokens equal to the value of their item quotation. The blocks of transactions can be trailed from when the biding organization opened the dander for bidding to start. Other transactions captured including registration, signing agreements and information updates is captured in the blockchain.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 8, Issue 11 , November 2021



TX hash	Block no	Timestamp	Date Time	From	To	Quantity
0xa4df7c7bd246a9e5de7bfd6bac29e9a5cadffa 3a6da2dd8ac4a22faeced772ca	5102742	1551259083	6/29/2021 11:00	0xc6e532bb14d444a4af146 14f300453609018c96c	0x32cd37a01a628605850 7d1f018c8cb8676a90aaf	3000
0x8f2a22c99ab613f27201062f08c82 51ba2669a3dd5dd83e5119bd7dd41559479	5137983	1551718836	6/29/2021 11:05	0x0289bd742d56e94dd04f7b a09185c5a139b35b2f	0x32cd37a01a6286058507 d1f018c8cb8676a90aaf	2300
0x315d74a791457efec5c9a176a30e 983f4f3d15176064de335648a806f0f03316	5234443	1552992765	6/29/2021 11:30	0x13e7b6833e25e19b9d19ba 3153bb95c173afdd34	0x32cd37a01a6286058507 d1f018c8cb8676a90aaf	1700
0xdd787bd3a83f166289fcd3b4 b91b8e02e9fa2e0d2895b36e5b93336ace5b02a1	5235275	1553003197	6/30/2021 10:10	0x13e7b6833e25e19b9d19 ba3153bb95c173afdd34	0x32cd37a01a6286058507 d1f018c8cb8676a90aaf	1100
0xa799b13f9150e48c8bf8c85e09c0ee d0a72c85ee0b8243865ed1d1dcefbd5217	5235278	1553003214	6/30/2021 10:12	0x13e7b6833e25e19b9d19b a3153bb95c173afdd34	0x32cd37a01a628605850 7d1f018c8cb8676a90aaf	2000
0x4e4b945584bb863b3a648b70e00 7130db4f214294f72fe2f8d39322f94c979b2	5235449	1553005518	6/30/2021 10:15	0x13e7b6833e25e19b9d19b a3153bb95c173afdd34	0x32cd37a01a628605850 7d1f018c8cb8676a90aaf	2760
0x693fe0148757bdc6977d728a871 1b8ec8814fc6e9593a343681c55a67bf69ac6	5710843	1559395986	6/30/2021 10:16	0xebd3872ea4d1d85a03a58 1127b872c99ebbab000	0x32cd37a01a628605850 7d1f018c8cb8676a90aaf	1900

Table 2: Organization Sample Report

VI. CONCLUSION AND FUTURE WORK

For parties to be involved in monitoring the organization activities, they need efficient tools and intuitive assessment that gives clear results. To build such an environment, blockchain and smart contracts show great potential. In this study, the tendering process is implemented in the blockchain environment to provide an open and fair tendering scheme.

REFERENCES

[1] Guo, Y. (2010, August). E-government: Definition, goals, benefits and risks. In *2010 International Conference on Management and Service Science* (pp. 1-4). Ieee.
[2] Guo, Y. M., Huang, Z. L., Guo, J., Guo, X. R., Li, H., Liu, M. Y., ... & Nkeli, M. J. (2021). A bibliometric analysis and visualization of blockchain. *Future Generation Computer Systems*, *116*, 316-332.
[3] Li, X., Wang, Z., Leung, V. C., Ji, H., Liu, Y., & Zhang, H. (2021). Blockchain-empowered Data-driven Networks: A Survey and Outlook. *ACM Computing Surveys (CSUR)*, *54*(3), 1-38.
[4] Liao, T. S., Wang, M. T., & Tserng, H. P. (2002). A framework of electronic tendering for government procurement: a lesson learned in Taiwan. *Automation in construction*, *11*(6), 731-742.
[5] Erdmenger, C. (Ed.). (2017). *buying into the environment: experiences, opportunities and potential for eco-procurement*. Routledge.
[6] Åkers, J. (2018). Driving fashion with data: A qualitative study of how buying firms in the buyer-driven fashion supply chain can benefit from a digitized supply chain reconfiguration.
[7] Hutchinson, A. C. (2015). *Fighting Fair*. Cambridge University Press.
Korpela, K., Hallikas, J., & Dahlberg, T. (2017, January). *Digital supply chain transformation toward blockchain integration.* In *proceedings of the 50th Hawaii international conference on system sciences*.
[8] Makupi, D. (2020). 5G Propelled AI, IoT and Blockchain. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*
[9] Iyidogan, E. (2018). Incentive Mechanism and Economic Model of Blockchain Based Cryptocurrencies.
[10] KAFOL, C., BREGAR, A., & TRILAR, J. (2018). BLOCKCHAIN FOR ENERGY UTILITIES. *DAAAM International Scientific Book*.
[11] Dannen, C. (2017). *Introducing Ethereum and Solidity*. Berkeley: Apress.
[12] Peters, G. W., & Panayi, E. (2016). Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. In *Banking beyond banks and money* (pp. 239-278). Springer, Cham.
[13] Thampi, S. M. (2010). Survey of search and replication schemes in unstructured p2p networks. *arXiv preprint arXiv:1008.1629*.