Vote Validation using Blockchain Technology: A Conceptual Use Case

Frankline Makokha

Abstract—Blockchain Technology is one of the computing technologies touted to likely bring about disruption in ways people conduct their transactions. By design, blockchains are decentralized, peer to peer, distributed consensus, and have anonymity property thus eliminating the need for a central Authority. Blockchain has been widely used in crypto currencies, with other uses lip frogging at slower paces. This paper explores the various uses cases that have been advanced for blockchain highlighting the shortcomings of the listed cases. The paper expounds more on usage of blockchain in electoral processes, analyzing existing voting use cases and identifying the shortcomings of the listed blockchain voting use cases. A more elaborate voting use case is conceptualized with clear description on how to generate Digital Votes, linked to a NONCE and previous Digital Votes. The process of vote validation is explained with the main component of the blockchain voting being highlighted as the Smart Contract.

Index Terms—Blockchain, Digital Vote, NONCE, Smart Contract

I. INTRODUCTION

The value of advances in computing power and computing technologies is realized when the computing power and technologies are integrated in the various facets of human life. Immediately after a technology is proven to be commercially viable, various use cases are developed to help spur its integration and application in various tasks.

One such new technology is the blockchain technology.

According to [1], block chain is a set of digital transactions cryptographically linked by hashed content of previous transaction and timestamps. Other definitions of blockchain are: a series of hashed timestamps, each cryptographically linked to the previous, using a hash digest [2]; and, a distributed database of records, or public ledger of all transactions or digital events that have been executed and shared among participating parties [3].

A use case is a description of a proposed functionality of a new system [4]. Other definitions include: a description of how an entity may use a system under development [5], and a description of the possible sequences of interactions between the system under discussion and its external actors, related to a particular goal [6].

Blockchain being a fairly new technology, it is yet to be used on a large scale in computing and non-computing tasks. To speed up the integration of blockchain in various human tasks, development of handy use cases is very critical. According to [7] while blockchain technology is commonly considered potentially disruptive in various regards, there is a lack of understanding where and how

blockchain technology is effectively applicable and where it has mentionable practical effects.

Use cases are key as they are a powerful technique for capturing and communicating functional requirements for software development [8].

II. RELATED WORK

Numerous attempts of using blockchain are happening in the financial sector, due to bitcoin and other crypto currencies [9]. According to [9] Uses cases of blockchain can be categorized based on the application software in which blockchain is implemented. This categories are: Data Management where use cases have ben listed as content and resource distribution, cloud storage, identity management, tamper proof event log and audit trail, digital content publishing and selling and IoT sensor data purchasing.

The other category by [9] is Data verification where the listed use cases are Photo and video proofing, document notarization, work history verification, product quality verification and proof of origin. The other category by the same author is Financial, where the use cases have been listed as currency exchange and remittance, P2P payments, crowd funding, value transfer and lending, among others.

The author also has a final category dubbed others where other listed use cases are lottery, property rights registration, court proceedings, electronic locks, gaming among others.

Whereas a list of companies where the named uses are in use has been provided by [9], it is not clear cut to a reader the exact role blockchain plays in those applications and how it implemented.

Blockchain use cases have been categorized based on global challenge areas identified by earth system scientists [10]. These are areas that need urgent action in the 21st century, namely climate change, biodiversity and conservation, healthy oceans, water security, clean air and water & disaster resilience.

The listed blockchain use cases for climate change are Peer-to-peer renewable energy-trading systems, Blockchain-based land, corporate, civil and asset registries, Secure paperless transactions and peer-to-peer vehicle.

Cryptocurrency for investment in habitat restoration and species conservation, Recording of pesticide use on agricultural land, Timber and other natural resources provenance tracking and Digital data platform for species tracking and disease control are listed as uses for Biodiversity and conservation challenge.

Healthy oceans has Tracking fish provenance, Real-time monitoring of ocean temperature and pH and Decentralized & open-source ledger of ocean data as some of the use cases.

Water monitoring and management, decentralized,

Published on August 14, 2020.
Frankline Makokha, University of Nairobi, Kenya.
(corresponding e-mail: goldmedalist321@gmail.com)

catchment-based approach to improving water quality and Hyperlocal water data for monitoring water quality, among others have been listed as water security challenge use cases.

Clear air challenge has Air pollutant data collation from distributed sources, Automated air-quality monitoring system and Early detection of toxic chemical leaks as usecases.

The final global challenge area, Weather and disaster

Resilience has Extreme weather impact analysis, Decentralized weather sensors generating automated alerts and automatic rerouting of power to prevent blackouts as usecases among others.

Whereas [10] lists several use cases, the listing does not conceptualize how the listed items can be put into actualization and linkage to blockchains.

Several blockchain uses in real estate have been listed by [11]. Among the use cases listed by [11] are: Token Securitization, which involves subdividing the value of a real estate property into very small values which are sold as tokens to those interested. The advantage advanced by [11] is that since tokens exist on the blockchain, the investor is able to make transfers without the delays, high costs, middleman and other sources of friction that typically belabor real-estate transactions. Tokenization enables complete partitioning of the property and enable one or more counterparties to purchase tokens and receive rental cash flow distribution rights.

With regards to the land registries use case, blockchain registries have the potential to shorten a normally lengthy process of recording and transferring titles while offering transparency and eliminating the need to trust a centralized party [11].

Blockchains can also be used to create token enabled market places [11]. This is done by creating a regulated marketplace for secondary offering that allows accredited investors to more easily trade their shares (i.e. tokens) amongst each other.

Blockchains could also be used to provide standardized property data which would be crucial for buyers of property and sellers during marketing of the property. According to [11], blockchain networks and technologies provide a foundation to process, exchange, and control the downstream flow of data in a way that allows independent, unaligned, and even competing stakeholders to share with one another, without ever giving up ownership or control of their data to a central operator and its interests.

Whereas the use cases raised by [11] are valid, they are not conceptualized to enable progression to real usage.

This paper seeks to address the identified gap of lack of conceptualization on how to put blockchain technology into practical use. Further this paper adopts an industry approach, where usecases are developed that would be generic enough for use across the various applications applicable to the whole industry.

This paper does not propose new use cases but advances the select existing one to near implementation level via high level design concepts and interfaces. The selected use case is the voting system. This is selected because the working flow of a voting process is common knowledge.

III. CONCEPTUAL USE CASE

In order to facilitate faster adoption of blockchain, it is critical to develop use cases that are programmatically viable. Conceptual use cases come in handy to this realization. The selected use in this paper is the voting system, which is discussed in detail in the subsequent subsections.

A. Existing Blockchain Voting Solutions

Voting is one of the areas likely to benefit most from advances in blockchain technology. This is because elections are fundamental pillar of a democratic system enabling the general public to express their views in the form of a vote [12].

Various solutions have been proposed, that implement blockchain technology, case in point [12]. The solution however does not clearly state on how consensus on vote validity is arrived at. The solution is also not clear on how a previous valid vote is linked to the next vote cryptographically, a key principal in blockchain.

The other mobile voting framework utilizing blockchain technology and multifactor authentication was proposed by [13]. The framework introduces the use of a one time authentication password to verify the user's eligibility to vote and after the verification, one casts a vote, which is stored in a blockchain database. The vote is then picked up by a node, processed and distributed to other nodes for validation.

Whereas [13] mentions a blockchain database, there is no clear description of how the blockchain is created. Further, the framework gives the responsibility of picking the vote from the database to one node, which processes the vote and sends it to other nodes for validation. This defeats the concept peer to peer in blockchain where there is no central node to execute tasks for other nodes.

An elaborate blockchain integrated voting platform is proposed by [14] which clearly states how the blockchain transaction executed and linked to the others by use of nonce and smart contracts. However, [14] proposes each polling station as a node on the network. This means for voting where there are several polling stations running into hundreds or thousands, each of the polling station will be required to have a copy of the distributed database and participate in consensus.

This will be a tall order since it means all the polling stations must be linked with high speed Internet connectivity and power to enable realtime participation in consensus during the vote process. Setting up and maintaining such a huge network would be an expensive affair.

Blockchain voting as a service is introduced by [15], which is anchored on three blockchain frameworks, namely Exonum, Quorum and Geth.

The solution proposed by [15] uses a proof of Authority (PoA) consensus algorithm, where a select set of nodes not participating in the transactions are selected as validators and are paid for validating transactions. The solution further uses smart contracts and non interactive zero knowledge proof.

Smart contract are applications that execute on decentralized infrastructure, such as a blockchain in a tamperproof manner that no party including their creator can

alter their code or interfere with their execution [16].

Zero knowledge proof is a validation technique where a protocol needs to convince the verifier of the veracity of a statement without revealing the content supporting that statement [17].

Whereas the solution proposed by [15] is comprehensive, the solution requires a voter to sign their vote by keying in a valid ID. This could lead to potential identification of who voted for which particular candidate.

Further, the solution relies on a consensus by select validators, this introduces central authorities, negating the principle of no central authority in blockchains. In addition, consensus is based on majority and not all of the nodes. This is a potential cause of conflict and lack of trust in the system.

A comprehensive blockchain based voting solution is elaborated by [18]. It is a four layered architecture consisting of the client layer made of devices that interact with the user, the application layer made of the e-voting services, the Blockchain layer made of the hyperledger fabric 2.0 for ensuring that only ordered blocks of an endorsed transaction are made available to the committing peer nodes before they are added to the blockchain, and the data storage layer.

Whereas the solution by [18] is comprehensive, it is composed of various nodes, assigned different tasks, namely E-Voting nodes, administrator nodes, public nodes, vote validation nodes and committing nodes.

This kind of set up basically means the nodes are at different hierarchy and thus are not peer to peer as envisioned by the blockchain architecture when it comes to decision-making.

Further, a compromise of a higher level node, like the administrator node means the entire system can be compromised.

It is on these listed shortcomings that this paper proposes a conceptual architecture that uses blockchain peer to peer consensus decision making in voting process and vote validation.

IV. PROPOSED BLOCK CHAIN VOTING SOLUTION

The proposed blockchain voting solution uses a predetermined number of blockchain databases, each hosted by different stakeholders as agreed on within the country or among voting stakeholders. The databases are under the administration of the different stakeholders, but running the same database of voters, same smart contract software, and any security measures required including connection to high speed Internet access for synchronization purposes. The high level architectural diagram is shown in Fig. 1.

From Fig. 1, a voter after undergoing preliminary mandatory checks at the voting station is allowed to access the voting terminal. The voting terminal can be an insert card based or enter voting number based device.

The voting terminal is linked by highspeed redundant Internet connectivity to all the blockchain databases. All the databases have the same applications running on them in terms of voter registers and the smart contract to be used in validating the voter credentials and arriving at a consensus on the vote cast.

Fig. 1. High Level conceptual diagram of the Proposed Blockchain Voting Platform.

The databases are linked to each other via redundant internet connection to ensure always on network connectivity amongst all databases.

The cross database connection is as shown in Fig. 2.

Fig. 2. High Level Conceptual Cross Blockchain Databases Linkage.

V. THE VOTING PROCESS

The blockchain voting process starts after the user inserts or keys in their voting registration details and proceeds to vote after verification and the vote is cast and a block generated and appended to the block chain database. The specific details are explained in the succeeding subsections.

A. Verification Terminal

Upon user inserting or keying in their voting details, the voting terminal forwards the details to all the linked databases. Using the smart contract in the database, a NONCE is computed based on the agreed on protocol in the smart contract, and also on the authenticity and eligibility to vote of the submitted details.

The computed NONCE value is compared amongst all the blockchain databases for concurrence. In case of non-

concurrency the user is denied access to the next stage, otherwise they proceed. The NONCE is updated simultaneously in all the blockchain databases awaiting further use. The update of the NONCE in the databases signifies successful identification of voter to the voting terminal.

B. Voting Stage

Upon positive verification of the voter, the voting terminal is notified of success and presents the user with a list of candidates to vote for based on the nature of the elections.

After a vote is cast for a particular candidate and submitted simultaneously to all the blockchain databases, the existing NONCE related to the particular voter alongside the vote for a particular candidate is used to compute a unique value called a Digital Vote based the smart contract. The computed value is compared amongst all the block chain databases for concurrence.

If there is no concurrence amongst all the databases, the user is notified on the screen to repeat, else the computed value is stored concurrently with atomicity in the block chain databases.

The number of votes for the particular candidate is updated and the voter is notified on the voting terminal of the vote having been successfully accounted for. The flag on that particular voter is updated as having voted to prevent repeat voting.

The process is repeated with the next NONCE computation taking as input the immediate preceding NONCE already saved in the database. Further, the computation for the next unique value (digital vote) after voting takes as input the immediate preceding unique value during the computation of the new unique value.

C. Proposed Blockchain Voting Algorithm

```

Insert/Enter voter card details
{
    While validated == True
    {
        Do
        Generate Nonce
        Subject generated NONCE to smart contract
        for consensus by all blockchain
        Databases
    }
    While NONCE consensus == True
    {
        Save the NONCE with atomicity in all
        blockchain databases
        Using saved NONCE as primary key
        display list of election candidates
        For each vote cast for a candidate, using
        smart contract algorithm, generate Digital
        Vote
        Digital Vote = Smart Contract {Current
        Database Digital Vote + NONCE
        + Vote for Particular Candidate}
        Compare generated Digital Vote for
        consensus
    }
    While Digital Vote Consensus == True
    {

```

```

        Update the candidate's vote with atomicity in
        all block chain databases

```

```

        Save the Digital Vote with atomicity to all
        block chain databases database
    }
}
End

```

D. Nonce Generation

According to [19], a nonce is a cryptographic input value which must never repeat within a given context. Given cryptographic algorithms require as input a key and a NONCE, with the key being constant, the NONCE must never be repeated [19].

The nonce generation process should ensure that the generated nonce adheres to the attributes of a NONCE, namely: Uniqueness (the value will never be repeated); Unpredictable; Timeliness (encoded as a unique timestamp) [20].

To ensure the generated NONCE is unique, the proposed NONCE generation algorithm must include the user voting identification number, the timestamp on when all block chain databases simultaneously did the verification and a secret seed, encrypted with an atleast 2048 bits secret key, decrypted only at run time and used without the value being seen.

A default NONCE is used where the NONCE is to be used for the first time, as defined in the smart contract.

NONCE = Smart contract {Previous NONCE + Unique
Voter Identifier + timestamp +
secret seed}

E. Digital Vote Generation

The generated NONCE, alongside the current digital vote successfully saved in the database, and the particular vote for a given candidate are subject to a smart contract, by each blockchain database. The result, called the digital vote is compared for each blockchain database for universal consensus.

For this to be actualized, a vote for each candidate will have been pre assigned a value, so that a voter selecting that candidate triggers the value to be used in the smart contract.

A default initial value of the digital vote is also pre defined in all blockchain databases to be used during the first time a vote is cast.

Digital Vote = Smart Contract {Current Database Digital
Vote + NONCE + Vote for
Particular Candidate}

F. Blockchain Formation

Since blockchains are digital transactions cryptographically linked to previous transactions, the digital vote generated is linked to the previous digital votes already saved in the database.

The NONCE used in the digital vote is also linked to previous NONCEs already generated in the block chain databases.

This linkage makes it impossible for one to alter any

digital vote already saved in the databases, as it would affect the subsequent digital vote values. Further, the smart contract would not allow the new value to be saved until a consensus has been arrived at by all participating blockchain databases.

The current NONCE is linked to the previous NONCE, while the current Digital Vote is linked to the current NONCE and the previous Digital Vote.

G. Vote Validation

Once a vote has been cast for a particular candidate and the generated digital vote has been confirmed valid through consensus, it is saved with atomicity against the name of the candidate whom the vote was cast for.

Attempting to alter the votes will mean the person altering will have to generate an associated digital vote.

Further attempting to save the altered vote will encounter challenges since all the blockchain databases have to concur to the value being saved through the defined smart contract.

The saved digital vote validates the vote cast for a particular candidate.

H. Smart Contract

A smart contract is a set of programs, which are self-verifying, self-executing and tamper resistant [21]. According to [22], a smart contract takes transaction as an input, executes the corresponding code and triggers the output events.

To develop a smart contract, one of the tools that could be used is the Remix, a browser based tool with an integrated IDE.

Smart contracts functions based on common "If" statements coded on blockchain where the systems on the network perform or execute the events [23].

Based on the explanation on smart contract functionality by [23], the voting smart contract will be embedded in the proposed voting algorithm.

The smart contract will be invoked when a NONCE is generated by any of the distributed blockchain databases. Once it is invoked, the smart contract will check with all the other blockchain databases for consistency in the generated NONCE, once confirmed, the smartcontract will return control back to the voting algorithm with a predefined value indicating consensus or otherwise.

Once a digital vote has been generated by any blockchain database, the smart contract function is invoked once again to compare from all the other databases for consistency on their generated digital vote. Once more the smartcontract will return a given value indicating consensus or otherwise and hand over control back to the voting algorithm.

The smart contract compares the various parameters used by each blockchain database in computing the NONCEs and the Digital Vote, apart from just comparing the end results of the NONCE and the Digital Vote.

VI. CONCLUSION

This publication has highlighted how blockchain can be used to enhance electoral processes through the use of a digital vote mapped and associated with the actual vote cast.

The work in this publication is at a conceptual stage and

future work will involve progressing the concept into a working prototype.

REFERENCES

- [1] F. Makokha, "Cryptocurrencies: Computational Building Blocks", *International Journal of Research and Innovation in Applied Science (IJRIAS)* | Volume IV, Issue VII, July 2019|ISSN 2454-6194.
- [2] S. Arim, R. Umar and L. Rubina, "Conceptualizing Blockchains: Characteristics & Applications ", *11th IADIS International Conference Information Systems 2018*.
- [3] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "BlockChain Technology: Beyond Bitcoin", *Applied Innovation Review (AIR)*, Issue 2, 2016.
- [4] Sparx Systems (2004) Sparx Systems UML Tutorials: The Use Case Model[online]. Available: https://sparxsystems.com.au/downloads/whitepapers/The_Use_Case_Model.pdf
- [5] R. M. Dijkman and S. M. M. Joosten, "An Algorithm To Derive Use Cases From Business Processes", *American Journal of Physics*, 2002.
- [6] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley 2001.
- [7] M. Risius and K. Spohrer , "A Blockchain Research Framework", *Business Information System Engineering*, (Bus Inf Syst Eng)59, 385–409 .
- [8] D. Elenberg (2005), "Use Cases: Background, Best Practices, and Benefits"[online]. Available: http://www.itk.ilstu.edu/faculty/bllim/itk353/NotesSlidesDocs/MKS_Use_Cases.pdf
- [9] K. Zile and R. Strazdiņa, "Blockchain Use Cases and Their Feasibility", *Applied Computer Systems*, May 2018, vol. 23, no. 1, pp. 12–20.
- [10] World Economic Forum (September 2018), "Building Block(chain)s for a Better Planet", *World Economic Forum System Initiative on Shaping the Future of Environment and Natural Resource Security* [online] available http://www3.weforum.org/docs/WEF_Building-Blockchains.pdf.
- [11] B. Don, D. Rajah, S. Ott and K. Fromm, 'Real Estate Use Cases for Blockchain Technology', Enterprise Ethereum Alliance – Real Estate Special Interest Group, Volume 1, March 2019.
- [12] K. M. Khan, J. Arshad and M. M. Khan, "Secure digital voting system based on blockchain technology", *International Journal of Electronic Government Re- search (IJEGR)*, 14(1):53–62 .
- [13] T. P. Abayomi-Zannu, I. A. Odun-Ayo and T. F. Barka, "A Proposed Mobile Voting Framework Utilizing Blockchain Technology and Multi-Factor Authentication ", *Journal of Physics: Conference Series*, Volume 1378, Issue 3.
- [14] M. Alam, M. O. Yusuf and N. A. Sani, "Blockchain technology for electoral process in Africa: a short review", *International Journal of Information Technology*. 10.1007/s41870-020-00440-w.February 2020.
- [15] F. Þ. Hjalmarsson, G. K. Hreiðarsson, M. Hamdaqa and G. Hjalmtýsson, "Blockchain-Based E-Voting System," *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, 2018, pp. 983-986, doi: 10.1109/CLOUD.2018.00151.
- [16] S. Ellis, A. Juels and S. Nazarov(2017) . ChainLink: A Decentralized Oracle Network [online]. Available: <https://crushcrypto.com/wp-content/uploads/2017/09/LINK-Whitepaper.pdf>.
- [17] Y. Cui, "Application Of Zero-Knowledge Proof In Resolving Disputes Of Privileged Documents In E-Discovery", *Harvard Journal of Law & Technology*, Volume 32, Number 2 Spring 2019.
- [18] O. Daramola and D. Thebus, "Architecture-Centric Evaluation of Blockchain-Based Smart Contract E-Voting for National Elections", *Informatics 2020*, 7, 16; doi:10.3390/informatics7020016.
- [19] E. Zenner, "Nonce Generators and the Nonce Reset Problem", In: Samarati P., Yung M., Martinelli F., Ardagna C.A. (eds) *Information Security. ISC 2009. Lecture Notes in Computer Science*, Vol 5735. Springer, Berlin, Heidelberg, 2009.
- [20] G. M. Køien , " A Brief Survey of Nonces and Nonce Usage", *SECURWARE 2015 : The Ninth International Conference on Emerging Security Information, Systems and Technologies*, 2015.
- [21] B. K. Mohanta, S. S Panda and D. Jena "An Overview of Smart Contract and Use cases in Blockchain Technology", *9th ICCCNT 2018 July 10-12, 2018, IISC, Bengaluru Bengaluru, India*.
- [22] B. Arshdeep, and V. K. Madiseti. "Blockchain platform for industrial Internet of Things." *Journal of Software Engineering and Applications* 9.10 (2016): 533.
- [23] H. Sheikh, R. M. Azmathullah, and F. Rizwan, "Smart Contract Development, Adoption and Challenges: The Powered Blockchain,"

International Research Journal of Advanced Engineering and Science, Volume 4, Issue 2, pp. 321-324, 2019.