IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 2, Ser. II (Mar. – Apr. 2021), PP 01-09
www.iosrjournals.org

An Improved Ethereum-Based Model for Smart Contract Implementation

Ahubele, B.[1], Eke, B. O.[2], Onuodu, F. E.[3]
[1,2,3] *Department of Computer Science, University of Port-Harcourt, Nigeria*

Abstract
This work discussed a way of improving smart contract implementation using Ethereumblockchain. This is achieved by the introduction of an improved verification and validation module, together with the Off-blockchain Checker that ensures that cold wallet to cold wallet transaction is checked. The system was modeled using Object-Oriented Analysis and Design methodology, with solidity as the language of implementation. The system showed improved performance over an existing system reaching Transaction Speed (TS), Scalability (S), Consensus and Incentive (CI), Availability and Usability (AU) and Security (S) of 86, 91, 88, 82 and 90 percents respectively against that of an existing system which had 67, 72, 74, 66, 71 respectively.
Keyword: *Blockchain, Ethereum, Validation, Verification, Off-Blockchain Checker*

Date of Submission: 08-03-2021 Date of Acceptance: 22-03-2021

I. Introduction

A Smart Contract could be defined as a piece of code written in Solidity (high-level language) and stored as bytecode in the blockchain to enable it run reliably in a stack-based ethereum virtual machine (EVM) once invoked.A Blockchain on the other hand is a distributed ledger technology which emerged in order to tackle issues of trust in ledger transactions through the provision of a platform for decentralized transactions without the need for a central authority or central control. Participants of this network, contributes to the establishment of trust among themselves by various consensus protocols associated with each blockchain.

An On-Blockchain is a decentralized and distributed digital ledger which records transactions across many computers making it difficult for records to be altered retroactively, without the alteration of the entire subsequent blocks. [1] stated that an On-blockchain allows participants to verify and audit transactions independently and relatively inexpensively. With the advent of distributed platforms, the possibility of securing record of information in a decentralized manner has become increasingly possible and practicable.

Many kinds of distributed ledger platforms have been introduced and used to carry out decentralized transactions. The first of these ledger platforms or blockchain was Bitcoin which was proposed by Satoshi Nakamoto[2].

Conducting transactions in centralized systems require the involvement of a middle-man for the establishment of trust with such systems usually trust-based. These trust-based systems have created numerous issues such as single point of failure, high transaction cost, lack of transparency, misconstrued information etc.

The Ethereumblockchain was launched in 2013 as an idea by VitalikButerin. However, the technology became operational in 2015, and has been largely used to extend the use of blockchain from a mere currency exchange to a decentralized applications (DApps).With Ethereum, users are encouraged to contribute to Ethereum service which is a programmable blockchain. Like other cryptocurrencies, Ethereum uses blockchain technology. The feature that separates Ethereum from other cryptocurrencies is that it allows the development and execution of "smart contracts" and "distributed autonomous applications-DApps.

According to [3], shared blockchain technology enables business collaborations which require high reliability, trust worthiness, privacy-preserving and immutable data repositories.Generally, Smart contracts and blockchain technology holds a great potential for the automation of business transactions reducing the need for paperwork, expenses, involvement of lawyers and court services to facilitate trust between parties. Blockchain and decentralized applications opens manifold opportunities to redesign collaborative business transactions such as supply chain, travel and tourism industries booking reservation, decentralized global market for computing power, logistics processes and a few others. Traditionally, such processes have been executed by relying on trusted third-party providers such as Electronic Data Interchange (EDI) hubs or escrows.

II. Distributed Ledger Technology

Blockchain is a domain of the distributed ledger technology field; and has been defined by different researchers from different perspectives as a trust layer that eliminates the need for a trusted third-party intermediary by combining peer-to-peer networks (P2P), cryptography and hash technology [4]. Similarly, [5] defined a distributed ledger system as a system in which hardware and software components located at networked computers communicate and coordinate their actions only by message passing.

[6]alsodefined DLT as an append-only chain of cryptographically-linked blocks of data maintained and updated by a decentralized network nodes. These networks of databases can operate smoothly and securely without the need for any central party or central administrator that every participant knows and trusts. [7] also defined a blockchain as a shared ledger that records all transactions that have ever occurred or exchanged since its creation. Various researchers have classified blockchain into public, federated or consortium and private [8], generic and specific [9]. Various researchers have classified blockchain into public, federated or consortium and private [8], generic and specific [9]. DLT operations are designed in such a way that information stored and communicated through the networks has a high level of trustworthiness, and every participant in the network can get simultaneous access to a common view of the information. Unauthorised changes to the information and its history are very difficult. However, there are various types of distributed systems today, such as Clusters, Grids, P2P (Peer-to-Peer) networks and distributed storage systems but each aimed at solving different kinds of problems.

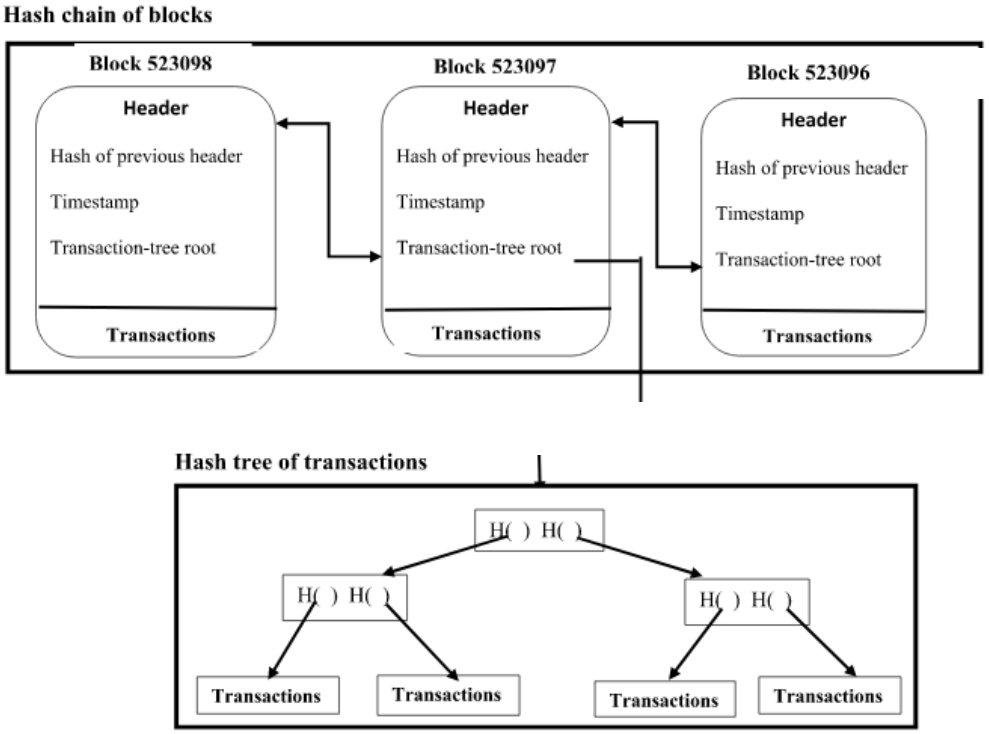

Figure 1: Illustration of how Blockchain works (**Source: [10]**)

2.1 EthereumBlockchain

Ethereum was the second blockchain introduced after Bitcoin by a young cryptographer called VitalikButerin in 2015 [8]. Being the second blockchain platform, its existence facilitated the need for developing applications using blockchain technology, other than the cash payment system which the initial blockchain was created for. Ethereum is described as an open-source cryptocurrency and smart contract computing platform that enable developers to build decentralized applications (DApps) that run on blockchain technology. The major difference between Bitcoin and Ethereum is that Bitcoin is a deflationary currency with a theoretical limit of 21 million bitcoins ever produced and its total amount in circulation shrinks over time as people lose access to their wallet, stockpile it, etc. While Ether has no limit, but does have a fixed ratio of production. Currently, about 15.5 million Ether is mined every year, which comes down to 5 ETH every second, far more than bitcoin's current rate of 25 BTC every 10 minutes or so. Furthermore, ethereum platform currency is called Ether (ETH) which powers the Ethereum network by paying for transaction fees and computational services.

There are two types of accounts: externally owned accounts and contracts accounts. Externally owned accounts are controlled by people. Thus, similarly to Bitcoin, each person has his own private key, which is used in order to make transactions in the Ethereumblockchain. Conversely, contract accounts are controlled by some smart contract code. In other words, such accounts are some sort of cyber-entities,having their own balance that can be triggered through some transactions coming from an external account (or some other contracts). Once triggered, the code specified in the contract is executed. This code can in turn generate some other transactions. The presence of these smart contracts allows developers to use Ethereum as a general purpose framework to create DApps.

2.2 Smart Contract

A smart contract is a computer protocol that run on Ethereum Virtual Machine (EVM) to digitally facilitate, verify and enforce the negotiation or performance of a transaction or an agreement. Smart contracts are self-executing, self-enforcing computer codes that contain rules that determine how the involved parties can interact with each other. Once these predefined rules are met, automatically the agreement is enforced. The EVM provides the operating system in which decentralized applications are built, deployed and executed on ethereumblockchain platform.

Furthermore, the idea of smart contract was first conceived by Nick Szabo in 1996 where he described it as a kind of vending machine, where users could input value or data and in turn get a finite item from the machine like a snack or soft drink. In addition, smart contract offers numerous advantages such as its ability to track performance in real-time consequently bringing tremendous cost saving. Smart contract offer the potential to disrupt many industries such as banking sector, insurance, supply chain, telecommunication, art world, music and film, education and many more.

The benefits of smart contracts include:

1. Direct customer's relationship

Smart contracts remove the need for trusted third party intermediaries and allow for transparent and direct relationships with customers.

2. Resistance to failure.

In the event of failure of any participating node in the blockchain network, the network will continue to function with no loss of data or integrity. Moreso, in executing applications using blockchain platform, businesses aren't dependent on a third party i.e no single person or entity is in control of data or money.

3. Trust.

Trust enable business agreements to be automatically enforced and executed. Unlike traditional business process that requires a third party control for trust enforcement.

3. Fraud reduction.

Since smart contracts are stored in a distributed blockchain network, their outcome is validated by everyone in that network. Therefore, no one can force control to release other people's funds or data, as all other blockchain participants would spot this and mark such an attempt as invalid.

4. Cost efficiency.

Eliminating trusted third party intermediary eliminates additional fees, allowing businesses and their customers not only to interact and transact directly but also to do so with low or no transaction fees.

5. Record keeping.

All contract transactions are stored in chronological order in the blockchain and can be accessed along with the complete audit trail.

6. Security

Smart contracts provide adequate security by employing the highest level of available data encryption mechanism. However, this level of protection provided by smart contract is among the best and the most secure on the World Wide Web.

7. Speed

Smart contracts are transactions that run on blockchain platform, which executes faster than traditional programs. This speed can save many hours when compared to traditional business processes execution off-blockchain.

2.3 Related Works

[11] designed Hawk, a blockchain cryptography and privacy-preserving smart contracts tool. The authors developed a framework for private smart contracts using zkSNARKs with strong privacy goals that include hiding the amounts of monetary transfers and contract state from non-participants and supporting private inputs that are hidden even from the other participants in the contract. However, their major drawbacks includes the following:

1. lack of scalability as each contract requires kilobytes of data to be put on-chain
2. absence of absolute privacy-preserving guarantee with Hawk tool since it relies on trusting a third-party manager who gets to see all the private data
3. SNARKS require a peer-circuit trusted setup which means that every distinct program that a contract implements, a new trusted set up is required. Multi-party computation can be included to reduce trust in the setup, which is infeasible to perform on a per-circuit basis as required by Hawk.

[12] presented design and implementation of a smart contract application. The author leverage an android application which uses the concept of smart contract to enable two parties determine legally valid electronic purchase and rent contracts without being dependent on any trusted-third party(TTP). Furthermore, the design was scalable and handles huge number of parallel contracts, detect network errors and prevents users from losing money (Ether).

[13] designed scripting smart contracts for distributed ledger technology. The authors used technologies such as zero-knowledge proof, proof-carrying code, static analysis, Merkelized Abstract Syntax Trees (MAST) in scripting smart contracts to ensure a more efficient system. Although, they focused on existing smart contracts scripting problems, strengths, weaknesses and solutions for known problems but they could not include universal solutions to smart contract open questions on major vulnerability attacks.

III. New System Design and Implementation

The proposed model generates correct-by-design systems of Smart contracts on Ethereum Virtual Machine (EVM). Verification tool ensures creating secure multiple smart contracts. There are three main stages of the model which include the design time stage, the On Blockchain Run time and the Off Blockchain Run time.

The adopted Methodology for the Proposed System Design is Object Oriented Analysis and Design, which brought out the different modules of the system and how they all interact. To implement the system, the solidity language was used.

3.1 Design Time BPM of the Proposed System

The Business Process is modelled to spell out clearly the business logic of the organizations and stakeholders of the contract; and is where the contract term is clearly specified. The specification of the Contract can be represented in simple English or using other tools that can be understood by the contracting parties. Linking the BPM model is the verification which checks if the contract terms are clearly modelled. Once the verification is completed, the verifiedcontract content and its logics can be translated to the level that it can be handled by the blockchain. Thisis handled using Solidity language that is required for the compilation of the contract. Sometimes a state machine parser or a parse tree is used to represent the logic at design time. The next stage of the system is the process implementation using solidity or any programming language of choice of the blockchain.

3.2Run Time Model of the System

The runtime model of the proposed system has two parts namely - the ON Blockchain part and the OFF Blockchain part.

3.2.1 ON-Blockchain Part of the Run Time Model of the Proposed System

The components that are used during the run time include the Contract which is defined in Blockchain, the Trigger which determines when the contract is fulfilled so that the reward can be released. The modified mediator manager (process instant contract) which serves as the middle man between the two contracting parties on the Blockchain. The mediator manager contains the Escrow, the DepositCollector, the validator, the Business logic of the contract and partial data payload. Other core components of the On-Blockchain part of the run time model are the participants account on the blockchain and the Execution State which have remained unchanged. The Escrow is also unchanged in the On-Blockchain part of the system development. Contract validation is rather used instead of verification which has already been done on the design time of the system development.

3.2.2 Off-Blockchain Part of the Run Time Model of the System

Unlike the existing system, the proposed system Off-Blockchain contains the main Data Payload, the Key Distributor, the Interface to the Internal Process and Off-Blockchain checker. The Off-Blockchain checker was introduced to make sure that cold wallet to cold wallet transaction is checked. The Off-Blockchain checker contains the participants or Stakeholders Off-Blockchain Account, the Contract Counterfoil and the Escrow Confirmer. The content of the Off-Blockchain checker is like the synch or backup of core activities carried out

On-Blockchain so that within a second node contract can be successfully checked before execution on the Blockchain. This is introduced to mitigate against 51% attack by malicious contract stakeholder. The super nodes that allow the core confirmation to be done and the main execution to occur is rewarded too withthe Ether. But the nodes on the chain are no longer fully relied upon to confirm the contract execution since the Off-Blockchain checker may have been empowered to take care of part of it.

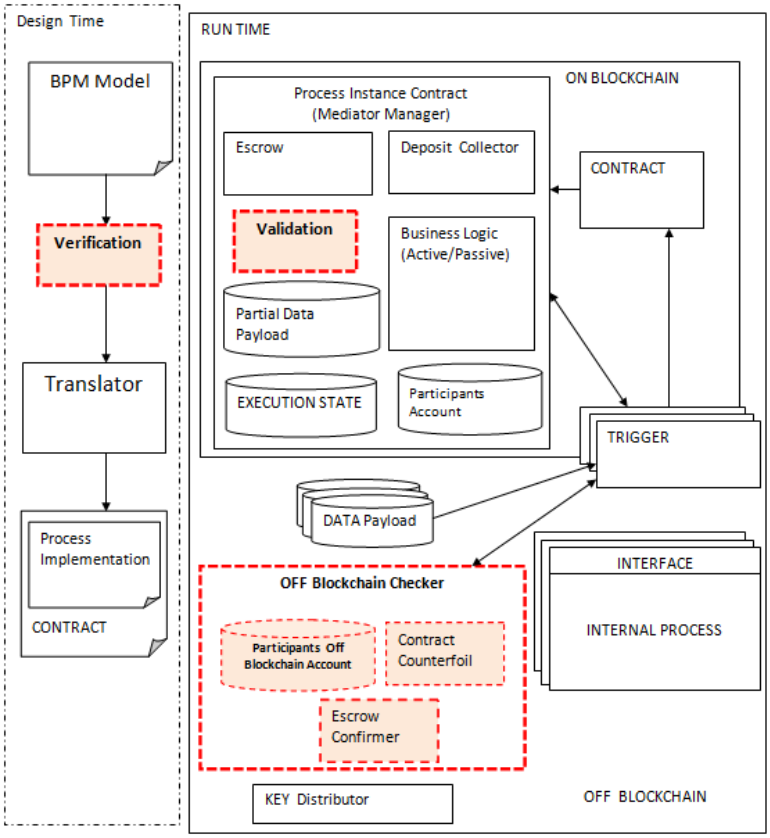

Figure 2: The Architecture of the Proposed Smart Contract Model

3.3 Descriptive Analysis of the Proposed System Architecture

The following were added to the proposed system

i. The verification tool
ii. The validation tool
iii. The off-blockchain checker

The verification tool used in the proposed system is etherscan; and include the following tools:

1. ChainStack: Enables scrutiny and launch of Ethereum nodes within minutes.

2. Coinage Solidifier: Accepts solidity codes for flattening into a concatenated version, before further making it ready for the etherscan verification process.

3. Ether Address Lookup: Enables the addition of links to strings that look like Ethereum addresses, so as to look them up on the blockchain explorer.

The validation tool is used for proving the accuracy of the inputted smart contracts. The validation tool for the proposed system is process mining. The process mining validation enables extraction of meaningful event logs froma blockchain. The event log can be imported in any process mining tool.

The off-blockchain checker on the other hand enables transactions that occur outside of the blockchain itself, and can be contrasted with on-chain transactions. It is also referred to as a centralized blockchain. It's features encompass a coupon-based mechanism and crypto-tokens. Furthermore, it also runs on an Ethereumblockchain interfaced with Remix IDE.

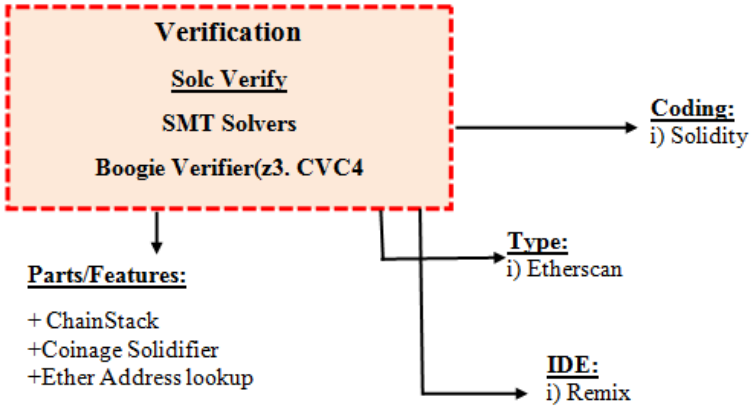

Figure 3: The Verification component of the Proposed Smart Contract Model

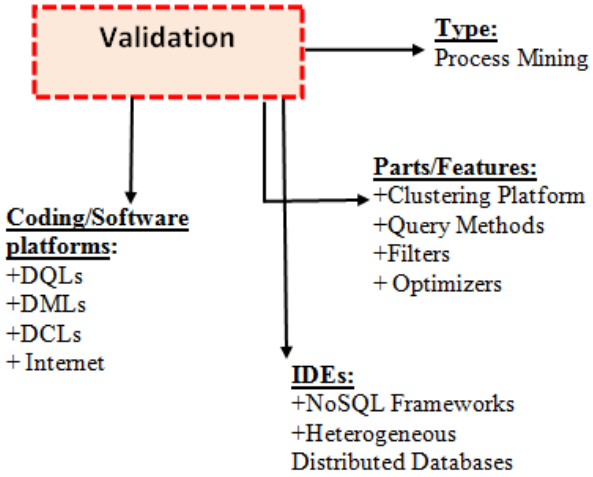

Figure 4: The Validation component of the Proposed Smart Contract Model

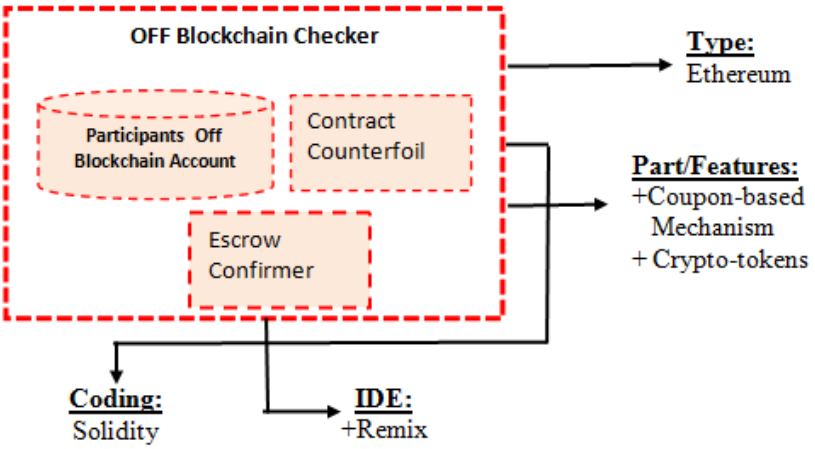

Figure 5: The Architecture of the Proposed Smart Contract Model

3.4 The New System Algorithm

The parameters used in the algorithm are defined as follows;

DT = Design Time, BPM = Business Process Modeling, T = Translator, PI = Process Implementation, FC = Factory Contract, RT = Runtime, OB = On-Blockchain, OFB = Off-Blockchain, T = Trigger, DP = Data

Payload, FC = Factory Contract, DAPPs = Decentralized Applications, VF = Verification, V = Validation, OFBC = Off-Blockchain Checker

Table 1: The New System Algorithm

System Algorithm
{User input settings}
Input BPM, DP
Step 1 Run T
Step 2 Run PI
Step 3 Run VF
Step 4 Initialize DT
Step 4.1 DT ←BPM+T+PI+VF
Step 5 FC ←PI
Step 6 Initialize RT
Step 6.1RT ←OFB+OB+T+FC+DP
Step 7 Run V
Step 8 UPDATE OB
Step 8.1OB ← OB + V
Step 9 Run OFB
Step 10 Run OFBC
Step 11 UPDATE OFB
Step 11.1OFB ←OFB + OFBC
Step 12 DISPLAY DAPPs AS OUTPUT
End
Quit System

IV. System Output and Result

Figure 6 shows how the contract was deployed by the user on the Remix IDE. Once a contract is successfully compiled, the contract solidity source code has been translated to the Ethereum bytecode. For a transaction to be recorded permanently on the Ethereumblockchain, it must be deployed. Similarly, in figure 7, once a user initiates deployment of the transaction, a confirmation page will be displayed where the user confirms the transaction before it can be deployed on the Ethereumblockchain. Upon a successful deployment, the user contract is permanently recorded on the blockchain. The user may also reject a transaction in order to allow any correction or addition in the source code.

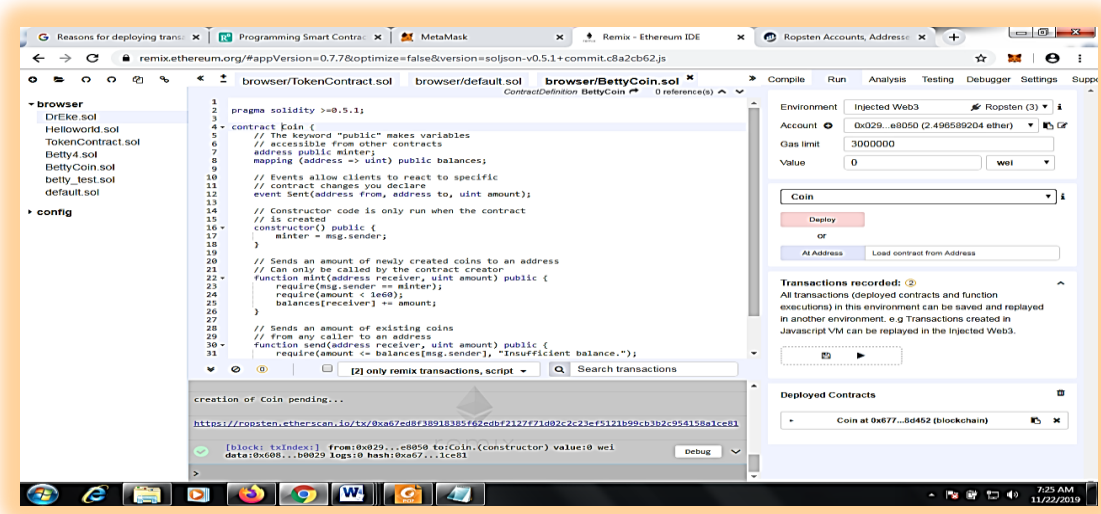

Figure 6: SSCDLS - Smart Contract Scripting and Deployment on Remix IDE

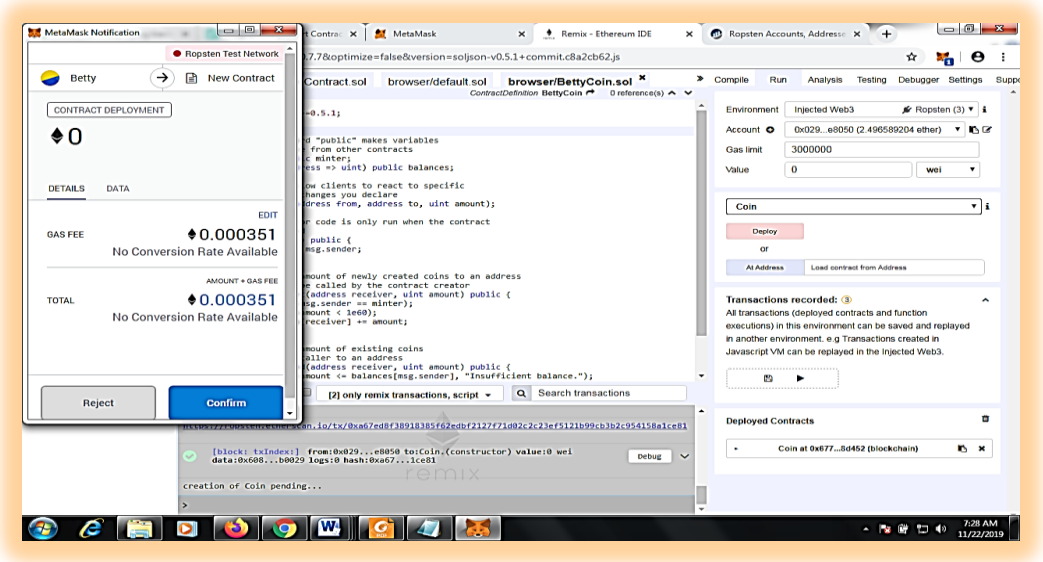

Figure 7: Smart contract confirmation and transaction details screenshot.

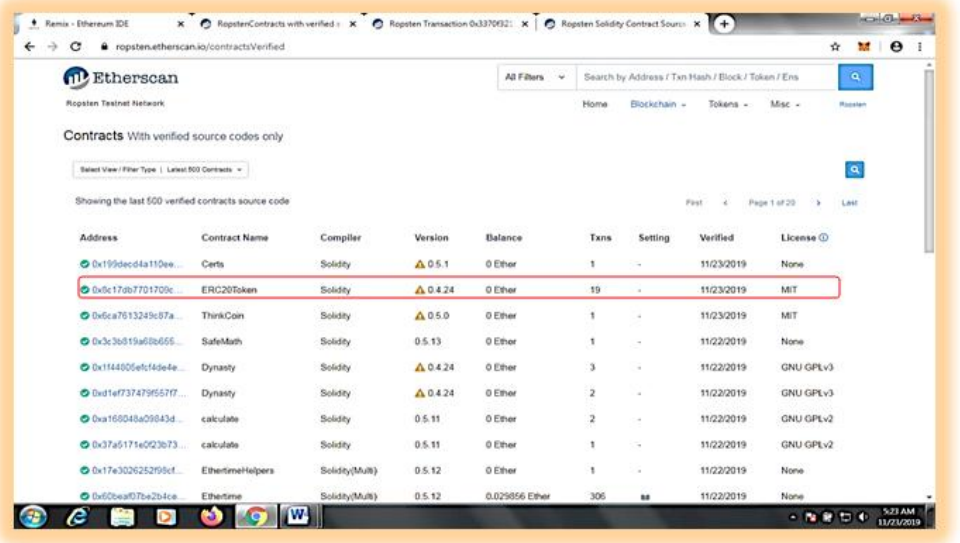

Figure 8: SSCDLS - Viewing verified contracts on Etherscan.

Similarly, to test the system performance, the system was evaluated against the [14] system. The system showed considerable improvement over the [14]. Table 2 shows the system performance in Processes/Confirmation, while figure 9 shows performance evaluation chart of the systems. It is seen that the new system performed better than the existing system. Similarly, table 3 shows the performance of the new system compared against the [14] system.

Table 2: System Performance in Processes/Confirmation

Process	**Tasks**	**Trace Type**	**Traces**	**Confirmation (Weber et al, 2016)**	**Confirmation (New System)**
Supply Chain Process	10	Conforming	15	2	5
		Not Conforming	57	3	7
Incident Management	10	Conforming	4	5	10
		Not Conforming	120	3	7
Incident Management with Payment	9	Conforming	4	4	8
		Not Conforming	19	5	12
Incident Management with Data Transformation	9			10	25

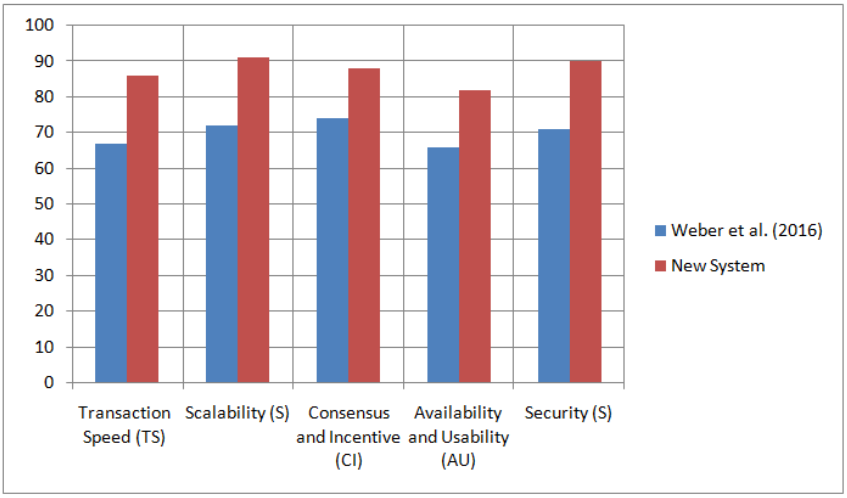

Figure 9: Performance EvaluationChart of the system compared against the Weber et al. (2016) system

Table 3: Performance Evaluation of the Systems

SN	Performance parameter (%)	Weber et al. (2016)	New system
1.	Transaction Speed (TS)	67	86
2.	Scalability (S)	72	91
3.	Consensus and Incentive (CI)	74	88
4.	Availability and Usability (AU)	66	82
5.	Security (S)	71	90

V. Conclusion

The implementation of smart contract through the use of blockchain technology promises to be the next major tidal wave of innovation and a driving force for promoting equitable societies through efficient and trusted business execution, solving complex economic issues and capable of transforming how we live and work daily. Furthermore, developers, shareholders and investors worldwide are harnessing Ethereum's open source, decentralized economy to build innovative projects and create inspiring solutions to solve a host of global issues on healthcare, insurance, finance, Air-ticket booking, land ownership/registration, supply chains, businesses and beyond.

References

[1]. Carlos, M.E., I. Solaima, I. Sfyrakis, J. Ng-Crowcroft (2018). Implementation of Smart Contracts Using Hybrid Architectures with On- and Off-Blockchain Components.23-29. https://arxiv.org/pdf/1808.00093.pdf. Retrieved09/10/2018.
[2]. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. 100-121. https://www.blockchain.org
[3]. Hull, R., V.S. Batra, Y.M. Chen, A. Deutsch, F.F.T. Heath III and V. Vianu (2016). Towards a Shared Ledger Business Collaboration Language Based on Data- Aware Processes. Springer International Publishing, Cham.18-36.
[4]. Mercy Corps (2017). A Revolution In Trust: Distributed Ledger Technology in relief and Development. https://www.mercycorps.org/sites/default/files/
[5]. Coulouris, G., J. Dollimore and T. Kinberg. (2001). Distributed Systems Concepts and Design. Addison-Wesley.UK.45-63
[6]. Michel R., G. Andrew, G. Brian, C. Gina (2018). Distributed Ledger Systems: A conceptual framework. SSRN Electronic Journal. DOI: 10.2139/ssrn.3230013.
[7]. Alharby, M., and A.Van-Moorsel (2017). Blockchain-based Smart Contracts: A Systematic Mapping Study, International Conference in Programming Languages with Applications to Biology and Security. Springer.142-161. https://doi.org/10.5121/csit.2017.71011. Retrieved 22/09/2018.
[8]. Buterin, V. (2017). A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper/. Accessed 11/10/2018.
[9]. Mattila, J.(2018). The blockchain phenomenon. Berkeley roundtable of the International Economy.International Journal of Software Engineering Research (IJSER), 9(5), 12 - 17
[10]. Narayanan, A., J. Bonneau, E. Felten, A. Miller and S. Goldfeder (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.
[11]. Kosba, A. E. Miller, Z. Shi, Wen and C. Papamanthou (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. IEEE Symposium on Security and Privacy.839-858.
[12]. Florian, S. (2017). Design and Implementation of a smart contract application. University of Zurich, Department of Informatics (IFI).Switzerland.1-143.
[13]. Pablo, L.S., T. Simon and M. Darryl (2017). Scripting smart contracts for distributed ledger technology. University of Kent, UK.
[14]. Weber, I., X. Xu, R. Riveret, G. Governatori, A. Ponomarev and J. Mendling (2016).Untrusted Business Process Monitoring and Execution Using Blockchain. In BPM 2016, LNCS 9850. Springer.329-347.