International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Tax Crow- tax transparency system using blockchain

Priyam Kanojia[1], Sarvesh Gandhi[2], Sardeep Singh[3]

[1]*Student, Department of Computer Science and Engineering*
Acropolis Institute of Technology and Research, Indore
[2] *Student, Department of Computer Science and Engineering*
Acropolis Institute of Technology and Research, Indore
[3] *Student, Department of Computer Science and Engineering*
Acropolis Institute of Technology and Research, Indore

Abstract - In recent years, the rapid development of blockchain technology and cryptocurrencies has influenced the financial industry by creating a new crypto-economy. Then, next-generation decentralized applications without involving a trusted third party have emerged thanks to the appearance of smart contracts, which are computer protocols designed to facilitate, verify, and enforce automatically the negotiation and agreement among multiple untrustworthy parties. In this paper, we present a tax transparency system using blockchain.

***Key Words*:** blockchain, smart contracts, smart contract verification, Ethereum, distributed ledger, hyper ledger.

1.INTRODUCTION

The future of society is digital; how to transfer the physical society's relationships to money, law, and even lifestyle and culture into digital relationships in the virtual world is a big challenge in IT technologies.

For more than a decade, the blockchain is established as a technology where a distributed database records all the transactions that have happened in a peer-to-peer network. It is regarded as a distributed computing paradigm that successfully overcomes the issue related to the trust of a centralized party. Thus, in a blockchain network, several nodes collaborate among them to secure and maintain a set of shared transaction records in a distributed way without relying on any trusted party. In 2008, Satoshi Nakamoto introduced Bitcoin that was the first proposed cryptocurrency introducing the blockchain as a distributed infrastructural technology. It allowed users to transfer securely crypto-currencies, known as "bitcoins" without a centralized regulator.

Besides, Ethereum, NXT, and Hyperledger Fabric were also proposed as blockchain-based systems used for cryptocurrency. Unlike Bitcoin, they can use smart contracts (SC). Blockchain technology overlaps traditional contracts by including the terms of agreements between two or more parties but surpasses them thanks to smart contracts by automating the execution of agreements in a distributed environment when conditions are met.

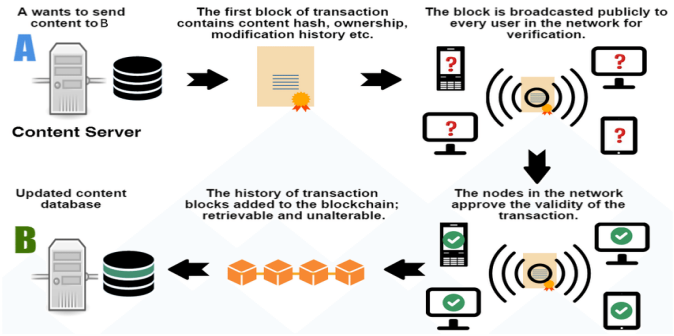

Figure 1: Architecture of Blockchain

2.SMART CONTRACT

Smart contracts are executable codes that run on top of the blockchain to facilitate, execute, and enforce an agreement between untrustworthy parties without the involvement of a trusted third party. Smart contracts gave network automation and the ability to convert paper contracts into digital contracts. Compared to traditional contracts, smart contracts enabled users to codify their agreements and trust relations by providing automated transactions without the supervision of a central authority. In order to prevent contract tampering, smart contracts are copied to each node of the blockchain network. By enabling the execution of the operations by computers and services provided by blockchain platforms, human error could be reduced to avoid disputes regarding such contracts.

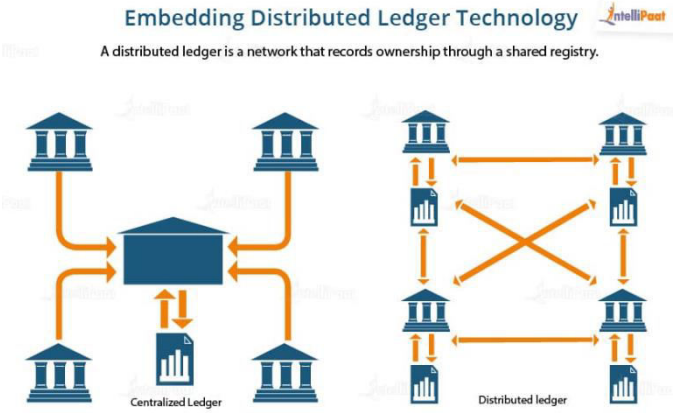

Figure 2: Central Ledger Vs Distributed Ledger

- TaxCrow is a platform where people can see that where their tax-paid money is being used.
- This platform is expected to increase the transparency in civil works.

- People can pay their taxes entitled to a particular work if they want and if they don't want to pay to a particular work the taxes will be automatically paid to the government.
- Bidding system for the contractors.

The main features are as follows:

Legality: The code complies with legal regulations. The controlled assets have ownership, and they are valid.

Probativeness: Process data and scenarios must be securely stored, and they can be used for legal evidence.

Consistency: Smart contracts should be consistent with the existing law text. Before publishing the smart contract, it should be reviewed by professional law persons to assure that the contract will not contradict the existing laws.

Customizability: Smart contracts are customizable. Multiple basic contracts can be combined into complex or complicated contracts.

Observability: Smart contracts require interfaces to observe the state of contracts, including the contract itself, its performance, and everything about the contract.

Verifiability: The records about the smart contracts can be verified. The working logic and the correctness of the execution of smart contracts are verifiable.

Self-enforceability: It needs enforcement to protect against breach and third parties which do not rely on law enforcement. Cryptographic contracts would give control by the cryptographic keys to operate the property for persons who rightfully own the property in terms of the contract.

Access-controlling: The information of contracts, such as knowledge, control, and performance, must be only accessible for contract-related persons. Unless when a conflict occurs, these properties of the contract will be exposed to the third parties.

To satisfy the above properties and solve the problems of smart contracts, we can learn from business contract architecture to design the role-based architecture of blockchain-based smart contracts system (as shown in Figure 3) In this architecture, there are a variety of roles played in the contract establishing, implementing, and trust mechanism.

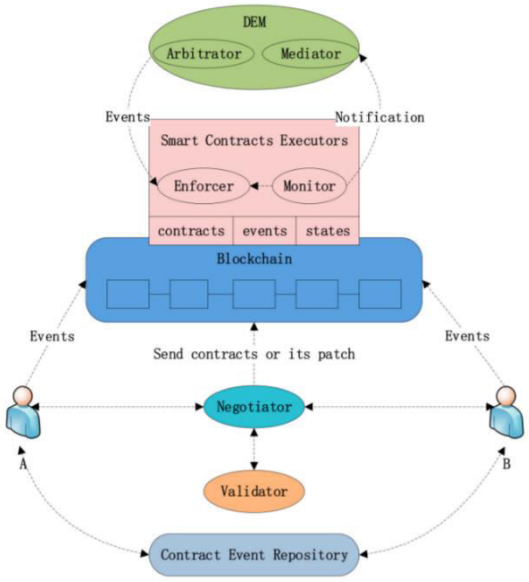

Figure 3: Smart Contract System

3. PROJECT SIGNIFICANCE

- This project has its significance in the area of open governance.
- People usually think that their tax paid money is being misused and in many cases, they are not false.
- This platform makes the taxpayers indirect boss of civil work.
- This platform can also help the government's knowledge about which type of work people want to be done and where.

3.OPERATIONAL PROCESS OF SMART CONTRACT

A smart contract is a common agreement between two or more parties. It stores information, processes inputs, and writes outputs thanks to its pre-defined functions. For instance, the smart contract can define the constructor function that enables the smart contract creation. Hosting a new smart contract in the blockchain is enabled by invoking the constructor function through a transaction, whose sender becomes the smart contract owner. A self-destruct function is another example of the functions that can be defined in a smart contract. Usually, only the smart contract owner can destruct the contract by invoking this function.

A smart contract is likely to be a class that includes state variables, functions, function modifiers, events, and structures [16] which is intended to execute and control relevant events and actions according to the contract terms. Besides, it can even call other smart contracts. Each smart contract includes states and functions. The former are variables that hold some data or the owner's wallet address (i.e., the address in which the smart contract is deployed). We can distinguish between two state types, namely *constant states*, which can never be changed, and *writable states*, which save states in the blockchain. The latter are pieces of code that can read or

modify states. We can distinguish between two function types, namely *read-only functions*, which do not require *gas* to run and *write functions* that require *gas* because the state transitions must be encoded in a new block of the blockchain. Furthermore, paying currency is required to avoid infinitely smart contract runs.

As aforementioned, a smart contract is hosted in the blockchain by invoking its constructor function through a transaction submitted to the blockchain network, then the constructor function is executed, and the final code of the smart contract is stored on the blockchain. Once deployed, the creator of the smart contract got the returned parameters (e.g., contract address), then users can invoke any available smart contract's function by sending a transaction.

ACKNOWLEDGEMENT

This paper summarizes the features of smart contracts completely and proposes the concept and framework of smart contract engineering (SCE) to meet the requirements of large-scale smart contract software production and verification in the future. The roadmap of formal methods for smart contracts is described in detail, including its main steps: from modelling to model transformation, to verification, to automatic code generation, and to conformance testing. In future work, we will improve the combination of smart contracts and computational law by designing a legal-oriented smart contract language, which can strengthen the legal supervision of smart contracts. Moreover, we also plan to develop a tool that automatically detects the conformance between the contract code and models, including the natural language context. It will speed up the extension and development of smart contract engineering

REFERENCES

1. Alharby M, Aldweesh A, van Moorsel A (2018) Blockchain-based smart contracts: A systematic mapping study of academic research (2018). In: 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), IEEE, pp 1–6

2. Amani S, Bégel M, Bortin M, Staples M (2018) Towards verifying ethereum smart contract bytecode in isabelle/hol. In: Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs. Association for Computing Machinery, New York, pp 66–77

3. Szabo, N. Formalizing and securing relationships on public networks. First Monday 1997, 2, 9. [CrossRef] Blockchain. Available online: https://en.wikipedia.org/wiki/Block_chain_ (accessed on 27 March 2020).

4. Blockchain and CSR. Available online: https://medium.com/@jeremilepetit/blockchain-and-csr-emergenceof-a-social-smart-contract-smart-contract-dfce4d5f064f (accessed on 15 November 2020).

5. IEEE Standards Coordinating Committee. Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990); IEEE Computer Society: Washington, DC, USA, 1990.

6. Sanghavi, A. What is Formal Verification. Available online: https://archive.eetasia.com/www.eetasia.com/STATIC/PDF/201005/EEOL_2010MAY21_EDA_TA_01.pdf?SOURCES=DOWNLOAD (accessed on 25 November 2020).

7. Solidity documentation @- https://solidity.readthedocs.io/en/v0.4.24/
8. Nodejs documentation @ - https://nodejs.org/en/get-involved/
9. Dapp Community- https://www.dappuniversity.com