e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:05/May-2021 Impact Factor- 5.354 www.irjmets.com

BLOCKCHAIN-BASED SMART CONTRACTS: COMPHRENSIVE STUDY

Divik Juneja*[1], Dr. Suman Madan*[2]

*[1] Research Student, Department of Computer Science, Jagan Institute of Management Studies, Sector-5, Rohini, Delhi, India.

*[2] Associate Professor, Department of IT, Jagan Institute of Management Studies, Sector-5 Rohini, Delhi, India.

ABSTRACT

Smart contract is one of the interesting feature of Blockchain Technology. Blockchain-based smart contract is like a legal bond deployed on a blockchain network as an asset record which has a power to automatically change the ownership of an asset if specific conditions are met, Example: - Contract owner gets money for his asset. Smart contracts set up a contract between two untrustworthy parties without the involvement of a third party. The aim of this paper is to explain the meaning of smart contracts and how they can be used. It also discusses the future potential of smart contracts as well as research challenges in the field.

Keywords: - Blockchain Technology, Smart contracts, Applications of smart contracts.

I. INTRODUCTION

1.1 Blockchain Technology

Blockchain technology has aroused a lot of interest in many people around the world in recent years, with emerging applications in core domains, including finance, energy, insurance, logistics and mobility. A blockchain is a peer-to-peer, decentralized, distributed, and ever-growing immutable ledger that is made up of a connected chain of records. These groups of records are referred to as blocks, and the records themselves are referred to as events or transactions. The ledger accepts transactions after they have been checked by the blockchain network's participants [1]. Each block contains the previous block's cryptographic hash value, a timestamp, and a collection of data transactions. When new block gets validated by consensus and written to the database(ledger), it is not possible to alter transactions without the agreement by the network majority.

1.1.1 Features of Blockchain

1) **Immutable: -** Immutability in blockchain means data stored in ledger can’t be easily changed. Secure transactions are only allowed after data that has been added to the block is validated by participants in the blockchain network. Miners are the participants in the network that verify the transactions and approve them to include in blocks.
2) **Decentralized: -** Blockchain is decentralized network which means there is no central authority to supervise over the database. Every member of the blockchain network has a similar ledger copy in their own machines. Centralized [Fig.1] and decentralized network [Fig.2] can be seen in the below figures.

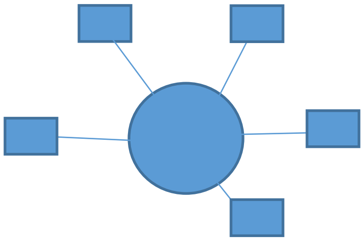

Fig 1: - Centralized Network

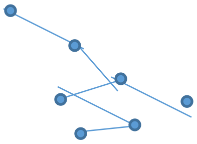

Fig 2: - Decentralized Network

3) **Peer-to-Peer network: -** Blockchain network allows only two parties, i.e. the sender and receiver achieving peer to peer network. This enables everyone in the blockchain network to authorize their transactions on their own, removing the involvement of third party authorization.

4) **Security is Better: -** There are number of computers called nodes participating in the blockchain network which confirm the transaction on the network, so that's why blockchain technology has a better security. Moreover, there is not even a single chance of shutting down of the system.

1.2 Significance of Blockchain-Based Smart Contracts

A smart contract is self-executing and self-enforcing code that runs on the blockchain to facilitate and execute the terms of an agreement between two untrusted parties. It is a contract that gets transfer to another person who completes the pre-defined conditions of a smart contract and all digital assets would be named on his name [2]. With the first implementation in the financial industry, Buterin et al. [3] pioneered the idea of smarts contract, stressing the main features. Smart contracts are an extension of the use of distributed ledger technology. On the blockchain network, smart contracts function as decentralized programs. The program is immutable, and its immutability has been cryptographically validated to ensure the program's trustworthiness. The execution of smart contracts in peer-to-peer mode without the presence of a centralized third party and service availability without any centralized dependence are two of their most important features. Contracts are smarter than paper contracts because of their autonomous execution matched to predefined conditions.

The features of smart contracts make them applicable to a wide range of domains. Many of these characteristics are inherited from the blockchain technology. Following are the main features of smart contracts: -

1.2.1 Trusted Third Party Elimination

Blockchain technology is well capable of working in accordance with decentralized nodes. Autonomous execution is provided under certain conditions. Continuous service availability is ensured by eliminating single points of failure with the help of decentralization. Decentralization helps to reduce data usage and latency in operations when compared to centralized systems. Decentralization ensures that the computational logic is transparent, removing the centralized "Black Box" concept and transferring accountability to all participants.

1.2.2 Forge Resistance

The digital signatures verify the credibility of each transaction and block in the distributed ledger. The forge resistance is a key distinguishing feature of blockchain that adds to its worth. The record of transaction and the execution of computational logic are validated cryptographically and keep going through the network.

1.2.3 Transaction is Transparent

Transaction transparency is one of the important feature of smart contracts. Both stakeholders in the blockchain ecosystem can see the blockchain ledger and smart contract logic. The blockchain's openness is a distinguishing attribute that makes it valuable in comparison to centralized databases.

1.2.4 Autonomous Execution

If the blockchain system reaches the triggering state, the programmed condition and flow of events specified to be completed will be executed. With the agreement of all parties in the blockchain network, the triggering state can be specified in the smart contract. It may be anything from a lack of funds, node reaching a specific geographic position or the system receiving a payment. There is no need for centralized third party as the execution is automated and activated on peer condition which ensures service availability.

1.2.5 Accuracy

The conditions in smart contracts are already fixed by an owner and tested before deployed in network nodes. Once the condition is fulfilled, the execution is performed automatically. The execution is expected to be accurate without any human or other error. Autonomous accurate execution reduces biased operations and increases confidence.

1.3 Smart Contracts Vs Traditional Banking

Transactions between two parties is likely to occur in a centralized manner (e.g. usually banks) which is supervised by a third party. Traditional banking has many disadvantages such as bankers tend to work slow during customer service as banks always requires human resources to process customer transactions, transaction fees is high, you literally have to go to bank when it comes to making a transfer or any type of big financial transaction at one particular time. While in the case of smart contracts, it is completely online and it can be carried out at any time or place. Ethereum is the most common blockchain platform to develop smart contracts. This is because Turing-completeness feature is there in Ethereum that allows creating more customized and advanced contracts.

1.4 Smart Contracts Example

An example to illustrate what a smart contract does. You would already know that purchasing a car from a car dealer involves many steps which can be tedious to some extent. Some people can’t afford a car as he might be out of finance, so he would need to get financing. This will necessitate a credit check, as well as the completion of multiple forms containing personal details in order to verify your identity. You'll have to deal with a variety of people along the way, including the salesperson, finance broker, and lender. Not to forget, commissions by middleman also adds up to the base price of car.

Smart contracts on the blockchain will help to simplify this complex method, which currently involves many intermediaries due to a lack of confidence among transaction participants. If your identity is stored on blockchain network, lenders can easily give you credit based on that. Then, after the funds have been released to the dealer, a smart contract will be established between the bank, the dealer, and the lender, with the lender holding the car's title and repayment starting on the agreed terms. Since the transaction is registered on a blockchain and it is shared among the members of blockchain network and can be reviewed at any time, so the transfer of ownership will be automatic.

1.4 Benefits of Smart Contracts

The benefits of smart contracts: -

- **Speed and accuracy: -** Since smart contracts are digital and automated, you won't have to waste time reviewing paperwork or reconciling and fixing mistakes that are often written into manual documents. Computer code is often more precise than the legalese used in conventional contracts.
- **Trust: -** Smart contracts conduct transactions according to predefined rules and share the encrypted records of such transactions with other participants. As a result, no one has to wonder if knowledge has been tampered with for personal gain.
- **Security: -** Since blockchain transaction records are encrypted, they are extremely difficult to hack. Since each record on a distributed ledger is linked to previous and subsequent records, changing a single record would require changing the entire chain.
- **Savings: -** Since participants can trust the visible data and the technology to properly conduct the transaction, smart contracts remove the need for intermediaries. Since it is incorporated into the code, there is no need for an extra individual to check and verify the terms of an agreement.

1.5 Platforms for Smart Contracts

Different blockchain platforms (e.g., Ethereum, Bitcoin and NXT) can be used to create and deploy smart contracts. For designing smart contracts, different platforms have different features. Some platforms allow smart contracts to be developed using high-level programming languages. This segment will only cover three public platforms.

- **Bitcoin: -** It [4] is a public blockchain network that can be used to process cryptocurrency transactions, but it only has a small compute capacity. Bitcoin makes use of a bytecode scripting language that is built on stacks. Using the Bitcoin scripting language, the ability to create a smart contract with complex logic is severely restricted [5]. A simple logic that requires several signatures to sign a single transaction before the payment is confirmed is possible in Bitcoin. However, due to the limitations of the Using Bitcoin scripting language, writing contracts with complex logic is not feasible. For example, the Bitcoin scripting language does not support loops or withdrawal limits [2]. The only way to execute a loop is to repeat the code several times, which is inefficient.

- **NXT: -** NXT is a public blockchain platform with smart contract models built in [5]. NXT only permits the development of smart contracts based on those models. Due to the lack of Turing-completeness in its scripting language, it does not allow personalized smart contracts.

- **Ethereum: -** Ethereum [2,6] is a public blockchain platform that uses a Turing-complete programming language to support advanced and customizable smart contracts. Withdrawal caps, loops, financial contracts, and gambling markets are all possible on the Ethereum network. Ethereum smart contracts use a stack-based bytecode language to write their code, which is then executed by the Ethereum Virtual Machine (EVM). Ethereum smart contracts can be written in a variety of high-level languages, including Solidity, Serpent, and LLL. These languages' code can then be compiled into EVM bytecodes and executed. Ethereum is currently the most popular platform for smart contract creation.

1.6 Smart Contracts Applications

Smart contracts can be used in a number of different ways. Some of these applications are as follows:

- **Internet of things and smart property [7]: -**Allowing certain nodes to exchange or access various digital resources without a trusted third party is one possible use case for blockchain-based smart contracts. This use case is being investigated by a number of organizations. Slock.it, for example, is a German company that uses Ethereum-based smart contracts to rent, sell, or share something (for example, selling a car) without the need for a middleman.
- **Music rights management [8]: -** One possible use is to store the ownership rights to a piece of music on the blockchain. As music is used for commercial purposes, a smart contract will enforce payment to music owners. It also ensures that the payment is split evenly among the music's owners. Ujo is a startup that is looking at how blockchain-based smart contracts can be used in the music industry.
- **E-commerce: -** Facilitating exchange between untrustworthy parties (e.g., seller and buyer) without the use of a trusted third party is one possible use case. Trading expenses will be reduced as a result of this. Only after the buyer is happy with the product or service, smart contracts will release payment to the seller [9].

II. RELATED WORK

Smart contracts have built-in accountability and forge resistance, making it easier to execute contractual agreements. Smart contracts are useful in a variety of applications due to their distinguishing characteristics. There is a plethora of smart contract solutions on the market, each with its own set of distinguishing features that are best suited to particular applications.

- Wang et al.[10] presented a detailed overview of blockchain-powered smart contracts, highlighting the smart contracts' unique problems as well as future developments.
- Wright et al. [11] discussed the advantages and disadvantages of emerging decentralized technology, as well as the need for the expansion of a new subset of law known as Lex Cryptographia, to regulate blockchain-based smart contract-based entities under legal theory.
- Aggrawal et al. [12] provided a detailed in-depth study in the sense of smart communities, as well as a comparison to previous surveys.
- Wust et al. [13] critically studied the applicability of blockchain for a specific application situation, suggesting a formal framework for determining the appropriate technological solutions, and evaluating it with some real-world examples.

- Clack et al. [14] investigated the design landscape of possible formats for storing and transmitting smart legal agreements in conjunction with blockchain technology, with a focus on the financial services sector.
- Chen et al. [15] proposed an agent model for contract execution over a network of decentralized nodes and public ledger, to prevent users from manipulating smart contract execution by applying principles of game theory and agent based analysis.
- Sousa et al. [16] designed, implemented, and evaluated a BFT ordering service for HLF on top of the BFT-Smart state machine replication/consensus library, implementing also optimizations for wide-area deployment with good results.
- Xu et al.[17] proposed how to classify and compare blockchains and blockchain-based systems to assist with the design and assessment of their impact on software architectures. Moreover, taxonomy is intended to help with important architectural considerations about the performance and quality attributes of blockchain-based systems.
- Marino et al. [18] created a whole new collection of standards that allowed smart contracts to change or undo contracts, and they explained how to use them on the Ethereum smart contracts platform.
- Norta et al. [19] outlined the issues with non-machine readable classical contracts that are solely dependent on trust.

III. RESEARCH METHODOLOGY

To investigate studies related to smart contracts, I chose the systematic mapping analysis discussed in [20] as the research methodology. The findings of this systematic mapping study will help us to identify and map smart contract research areas. It would also allow us to recognize research gaps that should be addressed in future studies.

3.1 Definition of Research Questions

Following research questions are:

RQ1. What are the latest smart contract research topics?

RQ2: What are some of the latest smart contract uses?

RQ3: What are the research gaps that need to be filled in future research?

3.2 Doing the search

This move entails searching for and locating all scientific papers relevant to the research topic of smart contracts. For my research, I chose the phrase "smart contract" as the primary keyword for finding papers. I chose this word because I wanted to limit the scope of our research to only smart contract-related work. I chose scientific databases to perform my search after determining the keyword for the search process. Only high-quality papers from conferences, journals, seminars, symposiums, and books were considered.

3.3 Relevant Papers Screening

This move entails looking for related papers to my research questions. To find important articles, I used the same method as in [21]. I began by excluding papers that were unrelated to research based on their names. If I couldn't make up my mind about a paper, I'd look at the abstract. Each paper was also screened using exclusion criteria. Non-English papers, papers without full text, papers that used smart contracts in fields other than computer science, redundant papers, journals, newsletters, and grey literature were all removed.

3.4 Key-Wording using Abstracts

This step involves using the key-wording technique mentioned in [21] to identify all related articles. I began by reading each paper's abstract to determine the most relevant keywords and main contribution. The keywords were then used to categorize the papers into different groups. I read all of the papers after classifying them and made any possible improvements to the classification.

3.5 Data Extraction and Mapping Process

This procedure is used to collect all of the details needed to answer the study's research questions. Each paper yielded a different set of data. The key goals and achievements of papers are encapsulated in these data items.

IV. STUDY RESULTS

The findings of my systematic mapping analysis on smart contracts are discussed in this section. First, we'll go over the findings of my search and screening for related documents. The findings of the classification process are then discussed.

4.1 Searching and Screening Results

The systematic mapping analysis, which we discussed in Section 2, includes two steps: searching and screening for relevant papers. The following are the outcomes of these measures. During the search process, I searched for all papers in various scientific databases that used the word "smart contract." In sum, I collected 50 articles. During the screening process, I first ruled out papers that were meaningless based on their titles and/or abstracts. There were two reasons for the large number of papers that were rejected. First, several articles were unrelated to my research because I wanted to look at smart contracts from a theoretical perspective. Many journals, for example, looked at the issue from an economic or legal perspective. Another explanation for the exclusion was that some of the papers were about cryptocurrency or blockchain in general, which had nothing to do with our research questions. Following that, some duplicate papers were excluded. As a result, we only chose 30 articles for our systematic mapping analysis.

4.2 Classification Results

I divided the papers into two sections, smart contract problems and other smart contract related subjects, using the key-wording technique we discussed in Section 3. I discovered that about two-thirds of the papers deal with smart contract problems. I divided the problems into four categories: **codification, stability, privacy, and efficiency.**

Coding problems are associated with the growth of smart contracts and pose challenges. Bugs or vulnerabilities that an attacker might exploit to initiate an attack are referred to as security issues. Concerns over releasing contract details to the public are referred to as privacy concerns. Performance concerns refer to problems with blockchain systems' ability to scale.

Table 1. Classified issues and its solutions

Smart Contract Issues		Proposed Solutions
Codifying Issues	Proper and correct smart contracts coding is a difficult task.	• Use partial-automation of smart contracts creation. • Education (e.g., online tutorials).
	Unable to change or stop smart contracts.	• A collection of new standards can be adopted for changing/stopping smart contracts.
	Under-optimized smart contracts.	• ‘GASPER’ tool can be used which locates gas-costly pattern automatically.
Security Issues	Transaction-ordering dependency vulnerability.	• ‘SendIfReceived’ function can be used. • ‘OYENTE’ tool works directly with Ethereum virtual machine (EVM) byte code without access to the high level representation.

	Timestamp dependency vulnerability.	• Use block number as a random seed instead of using timestamp. • OYENTE' tool works directly with Ethereum virtual machine (EVM) byte code without access to the high level representation.
Privacy Issues	Lack of transactional privacy.	• 'Hawk' tool [26] can be used that does not store financial transactions in the clear on the blockchain, thus retaining transactional privacy from the public's view.
	Lack of data feeds privacy.	• 'Town Crier (TC)' tool can be used which enables private data requests with encrypted parameters.
Performance Issues	Sequential execution of smart contracts	• Execute smart contracts in parallel fashion.

V. DISCUSSION

This section is about the study results and explains the research questions answers that we defined in Section 3.

RQ1. What are the latest smart contract research topics?

The findings of this comprehensive mapping study revealed that the majority of current smart contract research is focused on identifying and resolving smart contract issues. Four separate issues were identified: coding, security, privacy, and efficiency. Two of the most discussed topics were coding and security. This is due to the fact that smart contracts hold valuable currency units, and any security violation or coding mistake could result in money being lost. Writing correct codes is difficult, the incapability to modify or cancel contracts and the lack of help to detect under-optimized contracts have all been described as codifying problems. Some of the security problems discovered include transaction-ordering dependency, timestamp dependency, mishandled exceptions, re-entrancy, untrustworthy data streams, and criminal activities. The lack of transactional privacy and the lack of data feeds privacy have been described as privacy concerns. The consecutive execution of smart contracts has been described as a performance problem. While some solutions to these problems have been suggested, some of them are just abstract concepts without any clear evaluation. A few others have yet to be addressed. For instance, [22]'s solution is merely a recommendation to use alternative programming languages without any implementation. [23] has identified criminal activities that have yet to be eradicated.

Other research focused on smart contract applications or other smart contract-related issues. Among the suggested applications are trading and fair exchange, identity management, the Internet of Things, and the formation of agreements. Smart contracts are being combined with the Internet of Things and licensing management, as well as exploring smart contract scripting languages, introducing new consensus methods, and suggesting an indexing method to look for useful information in blockchain systems.

RQ2: What are some of the most recent applications of smart contracts?

Applications developed on top of the blockchain technology are known as smart contract applications. On the Ethereum blockchain, we discovered a variety of smart contract applications. These applications will enable untrustworthy participants to share everyday objects, form contracts, manage their identities, and track the actions of Internet of Things devices.

RQ3: What are the research gaps that need to be filled in future research?

The first is a dearth of research into scalability and efficiency problems. The capacity of blockchain systems to scale is affected by the sequential execution of smart contracts, as we discussed in Section 4.2. As the number of smart contracts increases in the future, this issue will escalate. The author of [24] described a very high-level approach, which is simultaneous contract execution, without any concrete evaluation. Executing contracts that are based on each other at the same time is the problem of concurrent contract execution. It is important to conduct research on recognizing and resolving performance issues in order to ensure that blockchain can scale.

The second gap is that almost all current research is focused on Ethereum smart contracts, despite the fact that smart contracts can be built on other blockchains. Different blockchains have their own set of features and benefits. As a result, future research might look at various blockchain implementations for deploying and running smart contracts.

The third gap is the small number of smart contract implementations. Despite the fact that smart contracts have attracted a lot of coverage, there have only been a few implementations in the literature. This is because the smart contract concept is still in its beginnings.

The fourth gap is a scarcity of studies on how to deal with illegal activity in smart contracts. [25] only listed three forms of illegal activities that can be carried out using smart contracts but did not provide any solutions. As a result, future studies should concentrate on discovering additional forms of criminal activity and developing ways to combat them.

The last gap is lack of high-quality peer-reviewed literature on smart contracts. The majority of the study is done as blog posts or grey literature and does not make a significant contribution. As a result, high-quality publications on smart contracts are needed.

VI. CONCLUSION

Blockchain technology is a distributed database that records all network transactions. The main advantage of blockchain is that it allows untrustworthy parties to interact without the need for a third party. Other than cryptocurrencies, blockchain can be used to operate a number of distributed applications. One of these applications is smart contracts, which are executable codes that facilitate, execute, and enforce an agreement between untrustworthy parties. Although there are other frameworks for developing smart contracts, Ethereum is probably the most common blockchain system.

To better understand current smart contract topics, I conducted a thorough mapping study. The primary goal of this systematic mapping study was to classify and map smart contract research areas. I was able to recognize study gaps that would need to be resolved in future studies as a result of this. From a technological standpoint, the focus of this research was on smart contracts. As a result, research from various viewpoints were omitted (e.g., papers with an economic perspective).

Some study gaps in smart contract research need to be considered in future studies in this paper. The discovered gaps are: - a lack of research on scalability and efficiency problems, a lack of research on implementing smart contracts on blockchain platforms other than Ethereum, a limited number of suggested smart contract implementations, a lack of research on illegal activity in smart contracts, and a lack of high-quality smart contract research.

VII. REFERENCES

[1] E. Ben Hamida, K. L. Brousmiche, H. Levard, and E. Thea, "Blockchain for Enterprise: Overview, Opportunities and Challenges", Jul. 2017.

[2] V. Buterin, "A next-generation smart contract and decentralized application platform", 2017.

[3] V. Buterin, et al., A Next-generation Smart Contract and Decentralized Application Platform", 2014.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", 2008

[5] A. Lewis," A gentle introduction to smart contracts", 2016

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things", 2016.

[8] W. Egbertsen, G. Hardeman, M. van den Hoven, G. van der Kolk, and A. van Rijsewijk, "Replacing paper contracts with ethereum smart contracts," 2016.

[9] W. Banasik, S. Dziembowski, and D. Malinowski, "Efficient zero-knowledge contingent payments in cryptocurrencies without scripts", 2016.

[10] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, F.-Y. Wang, "An Overview of Smart Contract: Architecture, Applications, and Future Trends", 2018.

[11] A. Wright, P. De Filippi, "Decentralized Blockchain Technology and the Rise of Lex Cryptographia", 2015.

[12] S. Aggarwal, R. Chaudhary, G. S. Aujla, N. Kumar, K.-K. R. Choo, A. Y. Zomaya, "Blockchain for Smart Communities: Applications, Challenges and Opportunities", 2019.

[13] K. Wust, A. Gervais, "Do You Need a Blockchain?", 2018.

[14] C. D. Clack, V. A. Bakshi, L. Braine, "Smart Contract Templates: Essential Requirements and Design Options", 2016.

[15] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, W. Shi, "Decentralized Execution of Smart Contracts: Agent Model Perspective and its Implications", 2017.

[16] J. Sousa, A. Bessani, M. Vukolic, "A Byzantine Fault-tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform", 2018.

[17] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, "A taxonomy of Blockchain-based Systems for Architecture Design", 2017.

[18] B. Marino, A. Juels, "Setting Standards for Altering and Undoing Smart Contracts", 2016.

[19] A. Norta, "Designing a Smart-contract Application Layer for Transacting Decentralized Autonomous Organizations", 2016.

[20] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering,", 2008.

[21] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology? |a systematic review", 2016.

[22] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems", 2016.

[23] A. Juels, A. Kosba, and E. Shi, "The ring of gyges: Investigating the future of criminal smart contracts", 2016.

[24] M. Vukolić, "Rethinking permissioned blockchains", 2017.

[25] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter", 2016.