



A Blockchain Framework with Smart Contract Mechanism

Amrita Jain¹, Savi Jain², Shruti Lashkari³, Sweta Gupta⁴, Ashwinee Gadwal⁵
^{1,3,4,5}Acropolis Institute of Technology and Research, Indore, India

²Chameli Devi Group of Institutions, Indore, India

¹amritajain@acropolis.in, ²savijainn27@gmail.com, ³shrutilashkari@acropolis.in

⁴swetagupta@acropolis.in, ⁵ashwineegadwal@acropolis.in

Abstract. The overall objective of the study is to understand the current status of blockchain applications and evaluate their ability to satisfy the growing demand for blockchain knowledge in the applications industry. In order to determine which would be the superior option in each situation, it also evaluated the respective advantages of Ethereum and Hyperledger. The study's extensive data set allowed it to offer priceless insights into the intricate workings of a blockchain application. The materials used included reports, journals, and periodicals. The "smart contract" refers to a digital transaction that runs on its own, logs the pertinent dynamic activity on a distributed ledger, and uses predefined criteria to demonstrate its legitimacy. The key component of a blockchain that enables its use as a platform for use cases beyond currency is a smart contract. Voting, education, entertainment, real estate, the Internet of Things (IoT), The development of blockchain technology has advanced significantly in recent years, with a particular emphasis on smart contracts; yet, little research has been done on the idea. Notwithstanding the many advantages of smart contracts, a number of obstacles have prevented their widespread use, including as security holes, coverage gaps, and the difficulties of lawfully enforcing contracts.

Keyword: Blockchain, Smart Contract, Solidity, Legal Issues, Ethereum, Hyperledger, Fabric.

1. Introduction

In 2008, an anonymous person or group using the alias Satoshi Nakamoto released paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [3]. Satoshi Nakamoto did not use the term "blockchain" in this article [1], but he did introduce the world to Bitcoin, the first decentralized cryptocurrency. blockchain is the current terminology for this decentralized system of exchanging data amongst peers. To address the problem of double spending, Satoshi proposed a point to point network in that paper. The possibility of double-spending describes the scenario in which digital money is used twice. Blockchains are a kind of distributed database that record and verify all of the transactions that have ever taken place on a particular network [4]. Existing monetary systems have a degree of centralization in how transactions are processed between parties. There must be an impartial third party present for this to work. The need for a secure and automated means of drafting business agreements is growing as more and more business processes are moved to the digital realm. There's a chance that using a centralized system may raise transaction fees and compromise data privacy. Because of advancements in blockchain technology [2], formerly untrustable parties may now conduct financial transactions without the need for a trusted third party.

The basis of smart contracts is one of the most well-known and talked-about applications of blockchain technology. A smart contract is a piece of executable code that can be included on a blockchain to help facilitate, execute, and enforce the terms of an agreement between unreliable parties [5]. You may conceive of it as a mechanism that, once the agreements have been met, will release the digital assets. Smart contracts [6] are a great alternative to the conventional contracts that are more difficult to read, take more time to execute, and

cost more money. Smart contracts, made possible by the efforts of blockchain consortiums like Hyperledger [4], have enabled the simplification of many business and financial processes in the modern world.

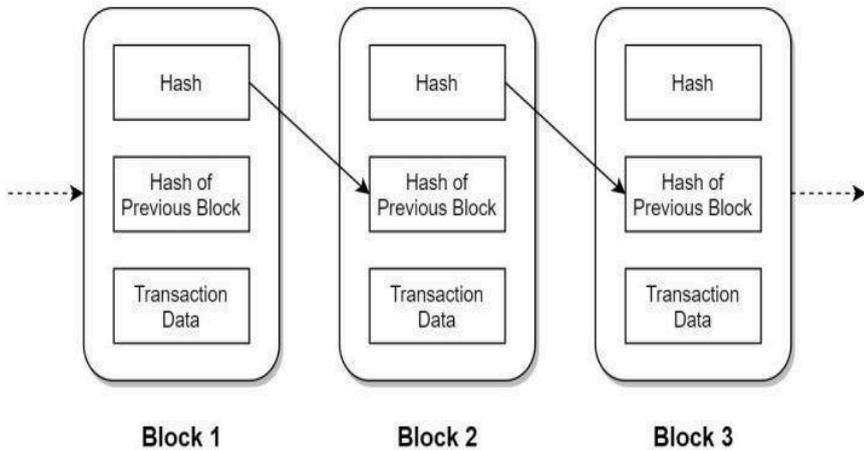


Fig.1. Representation of a block in Blockchain Technology

Many industries are using blockchain's features to meet certain business needs as more and more businesses rely on it for transaction and data security. Blockchain is being used in packaging and production to improve product traceability and guarantee compliance. By enabling stakeholders to follow products from point of origin to point of destination, this improves supply chain visibility and guarantees that they adhere to rules and industry norms. Blockchain technology is revolutionising accounting procedures in the banking, financial services, and insurance (BFSI) industries by increasing operational effectiveness and transparency. Real-time, unchangeable transaction recording made possible by blockchain technology can increase process efficiency, lower fraud, and boost financial reporting accuracy. Fundamentally, blockchain technology is a distributed ledger that uses hashing algorithms to encrypt each block of transactions. Because of this encryption, which makes the data safe and impenetrable, blockchain is a desirable choice for sectors that demand a high degree of security and data integrity. Because blockchain technology can provide both increased security and consistency, it is becoming more and more popular among both public and private organizations. Because distributed ledgers are available online, stakeholders from a variety of industries may instantly confirm transactions. By preventing inconsistencies, enhancing coordination, and guaranteeing that all stakeholders have access to the most recent data, this transparent, shared data architecture increases confidence in blockchain applications.

The following are the key characteristics of blockchain: Improved Security Model, Immutability, Timely Settlement, Consensus, Decentralized, Distributed Ledger Technology.

The following are just some of the many uses that have been found for blockchain technology: Health Data Repository and Business Exchanges, The Administration of Electronic Assets, Processing of Payments, IoT-based ecosystems, Forensics and identity administration, Tracking of Financial Transactions, Administration of Stock and Supply Chain Operations, Registry of Residents, Managing Public Relations Efforts, Managing Content, To Trade in Virtual Currencies, Real Estate and Property Administration

Some of the advantages of the blockchain are as follows : Credibility and openness to feedback , We may follow the path of the data , Benefits: Enhanced Productivity , Save Money, Increasing efficiency via process automation , Supply chain resilience , Enhancing Professional Connections , Delays and obstacles in the supply chain will be eliminated , Permanent Records of Activity to Ensure Compliance , Methods for finding and stopping fraud.

2. Background Study

The issue of trusting a centralised 1Ether (ETH) system—the digital currency used by Ethereum apps as a worldwide currency—can be resolved with distributed ledger technology, or blockchain. Therefore, to maintain and update a distributed ledger of all transactions, a blockchain network depends on a decentralised network of computers, or nodes.

To put it another way, the miners are a subset of the network nodes that are in charge of updating the blockchain—a distributed public ledger—on a regular basis. The first decentralised system to allow users to transmit and receive digital currency (bitcoins) without banking or other regulatory intervention was Bitcoin [7]. The tasks of miners include gathering transactions, resolving difficult computational problems (proof-of-work), and appending the finished transactions to the blockchain as blocks. Since then, a number of different blockchain-based development platforms have been put forth, such as NXT [9], Ethereum [10], Hyperledger Fabric [11], and others. All of these systems host or can be used with smart contracts to carry out events and actions automatically.

2.1 How smart contracts work in practice

A written agreement between two or more parties that is digitally signed is called a "smart contract." It has pre-defined functions that allow it to receive inputs, generate outputs, and store data. [10]. For example, the function `Object()`, which is used to create smart contracts in native code, may be specified in the contract itself. A new smart contract can be hosted on the blockchain by submitting a transaction that invokes the function `Object()` in native code; the sender of the transaction then becomes the owner of the new smart contract. Another example of the functionality that could be present in a smart contract is the self-destruct capability. Only the owner has the ability to destroy a smart contract.

A smart contract is likely to be a class [10] composed of state variables, functions, function modifiers, events, and structures in order to carry out and regulate significant events and activities in compliance with the contract requirements. The ability to communicate with other smart contracts is another feature. To put it simply, every smart contract has a state machine and is capable of performing a particular action. While the former are merely informational elements (i.e., the address where the smart contract is installed), the latter contains the owner's actual payment details. A distributed ledger may hold any of the two types of states: changeable and constant. The latter category includes programs that read or change system states. Operations can be classified as either read-only, which does not need gas 2, or write, which does, because the resulting state changes must be documented in a newly created block of the blockchain before they can be considered committed.

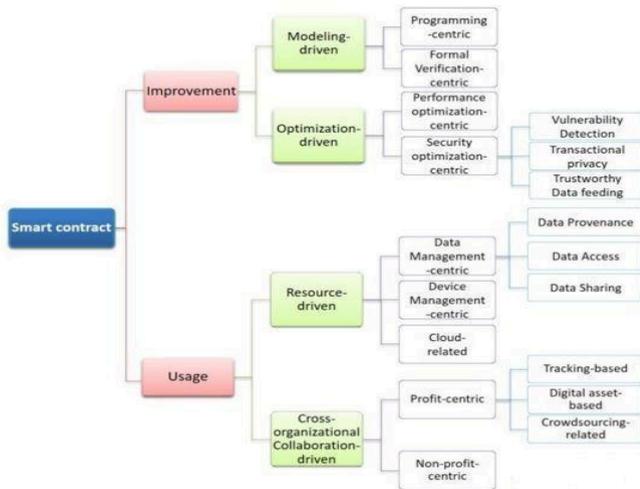


Fig.2. Classification of research on smart contracts enabled by blockchain Technology

The proposed taxonomy of blockchain-enabled smart contracts is depicted in Figure 2; it consists of four main categories: resource-driven smart contracts, smart contracts that are improved through modelling and optimization, cross-organizational collaboration, and finally, smart contracts that are improved through using blockchain technology to facilitate transactions between organisations.

2.2 Smart Contract Infrastructures

Nowadays, a wide range of blockchain systems, including NXT, Ethereum, and Hyperledger Fabric, provide strong support for the development and implementation of smart contracts. These platforms offer a variety of choices for creating smart contracts, each with unique benefits in terms of security, code execution, and programming languages.

For example, Bitcoin [7] is a public blockchain platform that may be used to handle cryptocurrency transactions, but with extremely little computational capabilities. Bitcoin's underlying programming language is a bytecode scripting language that operates on a stack. There is not much room for advanced reasoning in Bitcoin smart contracts. For smart contracts to work as intended on Bitcoin's blockchain, significant modifications to mining capabilities and mining incentive structures are required [8].

Using just a proof-of-stake consensus mechanism, NXT [9] is an open-source blockchain platform. Some examples of active smart contracts are included. But because it's not Turing-complete, you can't utilise your own custom smart contract templates; you're stuck with whatever the developers provide.

As of right now, Ethereum [10] is the only blockchain technology that can be used to create smart contracts. The Ethereum virtual machine is a Turing-complete virtual computer that can be used to build intricate and customised smart contracts. Every node in the Ethereum network runs a duplicate of the same application since the Ethereum Virtual Machine (EVM) provides the foundational framework upon which smart contracts are executed. Smart contracts are created using Solidity, a high-level programming language, and then converted into EVM bytecode for blockchain execution. Since Ethereum can be used to create DApps of various kinds and purposes, it is the most popular platform for smart contract development.

Hyperledger Fabric [11] is permissioned and only accessible by a limited number of companies through a membership service provider; in contrast to public blockchains such as Bitcoin and Ethereum, the network is composed of peers that are owned and maintained by those companies. IBM introduced Hyperledger Fabric, an open-source enterprise-grade distributed ledger technology platform that makes smart contract execution possible. Its versatility and adaptability allow it to be utilised for a wide range of business objectives. Hyperledger Fabric's modular design allows it to be easily and swiftly coupled with a wide variety of plug-and-play components to support a variety of commercial use cases.

Ethereum-based and Hyperledger Fabric-based smart contracts differ from one another in a number of ways. While Hyperledger Fabric supports a large number of computer languages, such as Go, Java, and Javascript, Ethereum smart contracts are written in the popular programming language Solidity [11]. Since Ethereum contract code is spread over the network, each miner that gets a transaction containing it is free to execute it in its own virtual machine [10]. Only the peers the application designates (endorsing peers) process and sign transactions generated by the application in Hyperledger Fabric. By contacting the appropriate chain-code, each of these endorsing peers independently carries out the transaction proposal from the application.

2. Literature Survey

In Figure 3, we see a flowchart depicting the study's methodology and the numerous activities that were performed. The electronic medical records (EMRs) are encrypted using the elliptic curve cryptography (ECC) and Edwards-curve digital signature method (EdDSA), and then stored in the cloud. The hashes of their connections are presently recorded on the distributed ledger. Both the block

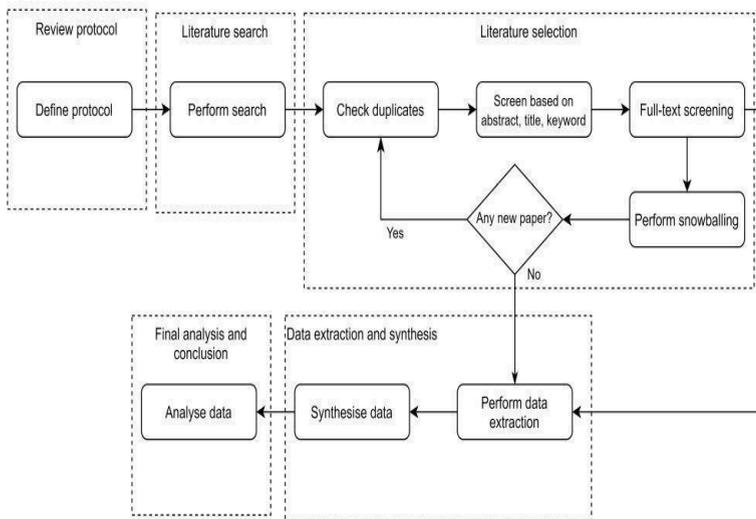


Fig.3. Process followed to conduct the review

size of the ledger and the amount of patient data are considered in this layout. To test the effectiveness of the proposed access control framework for the real-time smart healthcare system, we conduct a performance assessment using a private Ethereum system. [12]

Smart contracts require additional development and digital support to handle quality checks and the generation of delivery evidence during the delivery process. The ability of a blockchain to record off-chain transactions, such as the settlement of disputes, is one of its primary characteristics. By developing an open-source supply chain management blockchain, the authors advanced blockchain research. The use case, pilot design, and case study all made use of this blockchain[13].

This article's first paragraph gives a brief overview of blockchain technology. The writers then address the current state of research on smart contracts and blockchain 2.0. We shall look at the associated concepts of smart contracts in the following section. The smart contract's workings and the difficulties it faces are also explained. After a summary of the pertinent ideas and methods has been provided in response to these problems and situations, the future trends and challenges in the creation of smart contracts are finally assessed[14] provide an improved approach for AoI execution that bypasses the stack in favour of directly accessing storage locations through addressable storage locations. We deploy the ATOM protocol on a BC-IoT testbed built on a combination of Ethereum and Hyperledger Burrow. The studies show that ATOM performs better than state-of-the-art methods currently in use. Ledger size, gas consumption, and average update latency may all be decreased by as much as 62.7% thanks to ATOM's capabilities. Compared to the conventional approach to smart contracts, ATOM can improve execution performance by as much as 1.6 times and EVM Memory access efficiency by as much as 10 times. [15]

There is a wealth of current research on the feasibility of integrating smart contracts and blockchain technology into different business strategies. Expanding and enhancing the current capabilities of blockchain-based smart contracts requires an understanding of their technical aspects. The purpose of this study is to identify the key technological features of blockchain-based smart contracts and the most crucial areas for further research [16].

To guarantee the atomicity of data transfers throughout the compute result release and payment processes, a two-phase atomic delivery protocol has been developed. Under addition, contract theory suggests that, to maximise the advantage for the energy service provider and encourage user participation, the optimal contracts are formed in instances of knowledge asymmetry. This is done to increase the energy provider's profit. Extensive simulation results suggest that the suggested SPDS, when compared to standard schemes, may be useful in enhancing participant payoffs [17]. The fact that the SPDS is able to increase the payouts successfully demonstrates this.

This field of study is predicated on the idea that distributed ledger technology—more especially, blockchain technology—might offer solutions to these problems. A smart contract built on Solidity provides this solution. Even the most complicated situations that may occur in land administration systems, such as the division of properties, the transfer of partial ownership, the combination of numerous properties into one, and the limitation of real estate transactions, can be handled by this contract. A protocol implementation based on the ERC-20 and ERC-721 token standards is the recommended smart contract [18], which was designed to meet the unique requirements of land administration systems.

The article introduces a method for transforming a **Blockchain Platform Independent Model (PIM)** into a **Solidity Platform Specific Model (PSM)** to automate and streamline the creation of smart contract code. The research demonstrates how this transformation can

be used to generate Solidity code for smart contracts, focusing on the process and evaluating the generated code's effectiveness [19].

This article not only provides an in-depth analysis of smart contracts from many viewpoints, but it also identifies and dispels some of the most prevalent myths around them. This research also offers some recommendations and explanations about how to operate smart contracts efficiently. The findings of this research might be very useful in shaping the standards for smart contracts in the future. [20]

Encryption technologies and two-round protocols safeguard sensitive data, while corresponding ways of embedding and transmitting information are developed for various applications. It is more difficult to discern who is talking to whom since the same options, effective price ranges, and invalid bids have all been included into two separate contracts. The experimental findings validate the suggested model's simplicity and robustness to perturbations. This paradigm also has the potential for covert communication [21].

The study demonstrates that using smart contracts on a private Ethereum blockchain can effectively streamline emergency healthcare transactions, providing a **predictable** and **secure** method of transaction prioritization. The approach's reduced computational requirements and ability to handle different levels of trust make it highly applicable for **real-time intelligent healthcare systems**, paving the way for more efficient and transparent management of healthcare emergencies.[22].

In this research, we propose a novel formal verification method for ATL model checking to assess blockchain smart contracts. We do this by rethinking the interaction between the smart contract and the user as a two-player game. Next, we validate the characteristics that are important to us using MCMAS, a powerful ATL model checker designed for multi-agent systems. To demonstrate how our method for identifying smart contract defects may be used in real-world scenarios, we also provide three examples. We offer these case studies to illustrate the applicability of our suggestion [23].

The suggested solution shows that a safe, user-friendly, and effective system for processing social security applications and handling the related paperwork may be developed by integrating blockchain technology, IPFS, and smart contracts. The process is streamlined and made more transparent by the integration of these technologies, which also improves the security and auditability of social security services and lessens the strain for human reviewers. This strategy could greatly increase the effectiveness and reliability of online social security services [24]. It is through contrasting our proposed solution with the present one that these advantages become most clear.

While current counter measures for identifying vulnerabilities in smart contract code may face challenges, the research demonstrates the potential of tools that can more accurately locate and verify vulnerabilities. By fine-tuning these tools and considering different constraints, it may be possible to reduce false positives and improve the precision and accuracy of vulnerability detection. This, in turn, can lead to the creation of **more secure** and **trustworthy smart contract software**, laying the foundation for safer, more reliable blockchain applications in the future. [25].

The proposed decentralized method using **Merkle-Patricia Trie** and **parallel search** techniques offers a robust solution to the vulnerabilities associated with external oracles in smart contracts. By allowing smart contracts to directly access blockchain data, this approach enhances **data transparency**, **security**, and **efficiency**, making it a promising solution for improving the reliability of blockchain applications. The experimental results from the **Ethereum blockchain** validate the effectiveness of the method, suggesting that it could be a valuable tool for increasing the robustness of blockchain-based systems in the future [26].

End users of the trusted resource allocation method may tailor their purchasing decisions to their own priorities with respect to delay and cost by selecting from four distinct pricing structures. Moreover, smart contracts can instantly pair reliable edge servers with final users. As edge server activity is monitored, end users may also rate the trustworthiness of the corresponding smart contracts. The results of the simulation show that although the GBPM has the potential to provide differentiated pricing and optimise end-user utility accordingly, the REM is more attuned to edge servers exhibiting abnormal behaviour and quickly ruins their reputations to boost transaction success rates [27].

In simulation, the approach guarantees a trust probability of 0.38 even with 85% miner collaboration. Furthermore, because the method processes blocks on average in 1.3 seconds instead of 5.6 seconds for serial techniques, it shows enhanced scalability. The total computation and transmission costs are 101 bytes and 28.48 milliseconds, respectively. This demonstrates the effectiveness of the proposed approach in contrast to the existing approaches [28].

The organisations work together to create a channel within the network by cooperating with one another. Each of them has a form of identification that can be used to verify their identities and confirm their signatures on every transaction. We chose JavaScript as the language of choice for building our recommended smart contract for the sake of testing and research. We assess the smart contract using Hyperledger Caliper, and it provides us with an average throughput and latency of 10.4 tps and 0.7 seconds, respectively. Given the present control settings of the coffee beans, this indicates that the smart contract is quick enough to be deployed in a real scenario [29].

The **LM system** offers a promising solution for creating a collaborative, decentralized **AI market** that enables participants with limited mutual trust to work together in a **secure, transparent, and fair** manner. By leveraging the **Ethereum blockchain** and **IPFS**, LM enhances **auditability, traceability, and collaborative fairness**, which can help foster trust and enable distributed AI research and development. This system is particularly useful in situations where **data** and **computational resources** are fragmented or insufficient, allowing participants to pool their resources and tackle larger, more complex AI tasks. Overall, LM represents a significant step toward building a more **open and inclusive AI ecosystem**[30]. This assumption is based on our collaboration structure, which necessitates cooperation amongst dispersed AI contributors.

The registered participants are granted access permissions that are specific to their responsibilities in order to ensure that on-chain constraints are being followed. The creation of "smart contracts" preserves the provenance of data and offers reliable alerts and notifications. The presentation goes into the implementation details and testing of the algorithm. We describe, contrast, and rate the different security components that are part of our system in this section [31].

You must first create multi-phase smart contracts in order to create secure resource sharing and protect yourself from the malevolent actions of service requesters and self-serving cars. The consortium blockchain then implements a byzantine fault tolerance-based proof-of-stake (BFT-based PoS) consensus mechanism as a successful way to reach consensus. We develop a contract-based incentive mechanism to further encourage cars to share their computer capabilities with individuals who have submitted service requests. The finest contracts are those that are designed to enhance the expected utility of the service requesters as well as the overall welfare of society. The simulations' outcomes show that the suggested incentive mechanism functions more effectively and efficiently than the ones that are in place [32].

Table 1 : Comparative Study

References	Title of Paper	Name of Author	Technology	Methods	Parameters	Year
[33]	Automating Procurement Contracts in the Health Care Supply Chain using Blockchain Smart Contracts	Ilhaam A. Omar et al.	Ethereum Network,Blockchain, Decentralized Storage Technology	Registration Smart Contract, Price Negotiation, Purchase Order, Rebate Settlement Smart Contracts	Data provenance, Data Transparency, Data Immutability, Cost and Security Analysis	26 Feb 2021
[34]	Research on a Covert Communication Model Realized using Smart Contracts in Blockchain Environment	Lejung Zhang et al.	Steganography, Blockchain, Smart Contract	Covert Communication Method using voting Contract, Voting Contract, Information Embedding & Transmission Method, Improvement of Communication Security & Efficiency	Tamper Resistance & System Security, Contract Security, Scalability	26 Feb 2021
[35]	Defining Smart Contract Defects on Ethereum	Jiachi Chen et al.	Smart Contract and Ethereum	Byte Code Level Detection, Source Code Level Detection, Validation Survey	Availability, Performance, Reusability, Maintainability	26 Feb 2021
[36]	ContractGuard: Defend Ethereum Smart Contracts with Embedded	Xinming Wang et al.	Blockchain, Ethereum, and Smart Contract,	Protection Boundary of Contract Guard, Intra-procedural	Handle False Alarms, The Overhead of Contract Guard	14 May 2021

	Intrusion Detection			Path Indexing and Profiling, Calling-context Indexing and Profiling, Gas-efficient Adaptive Path Set Storage Mutation Strategy, Feedback mechanism, Contramaster, Mann Whitney U-test		
[37]	Oracle Supported Dynamic Exploit Generation for Smart Contracts	Haijun Wang et al.	Smart contract, test oracle, fuzzing		Static Analysis, Dynamic Analysis	14 May 2021
[38]	Blockchain based Distributed Framework for Automotive Industry in a Smart City	Pradip Kumar Sharma et al.	Blockchain, Supply Chain Management	Distributed Framework Model, Private Ethereum Blockchain, Algorithm to select Miner Nodes	Unparalleled security, Evidence integrity and secure storage, Mobility solution, Ability to audit records, Execution speed and cost reduction	29 April 2020
[39]	Blockchain Technologies and their Applications in Data Science & Cyber Security	Bhavani Thuraisingham et al.	Data Science, Cyber Security, Bitcoin, Blockchain, Smart Contracts	Asymmetric Key Cryptography, Proof of Work Model, Multilevel Secure database system	Heterogeneous Data, IoT Security, Distributed Ledger Operational Resilience	3 June 2021

4. Research challenges of Smart Contract

As a new technology, smart contracts are facing a number of issues that must be resolved to guarantee its broad use and dependability. These issues cover a wide range of topics,

such as consensus processes, off-chain dependencies, legality, and immutability and scalability.



Fig.4. Problems and unanswered questions

5. Existing Methods

Contract management systems and Layer 2 protocols provide two distinct perspectives on the future of smart contracts.

5.1 Protocols for Layer 2:

The performance and scalability of blockchain networks depend heavily on Layer 2 solutions, especially as the need for smart contracts and decentralised apps (dApps) grows. Blockchain networks can greatly increase transaction speed, lower costs, and solve the scalability issues of Layer 1 systems by utilising protocols like Ethereum Plasma, state channels, and rollups. These developments ensure that blockchain can manage the increasing volume of transactions and use cases without sacrificing security or decentralization, making it more useful for real-world applications [41] and the Bitcoin Lightning Network [40] are two well-known Layer 2 technologies. The MIT Media Lab's Digital Currency Initiative contributed to the development of a straightforward software application known as The Lightning Network. It is a means to expand public blockchains and facilitate interoperability across cryptocurrencies. Only big net transactions will need to be resolved directly in a blockchain with limited resources since small transactions will be relocated to an area that is cryptographically secure and "off-chain" [40]. The project's budget and timeline will be significantly impacted by this. Ethereum Plasma is a set of smart contracts that allow many blockchains to be created within a single root blockchain. Only with the assistance of the root blockchain can the Plasma chain continue to function as it does. The root chain is only identified and punished when there is proof of fraud, even though it is the general enforcer of all computers. It is possible for many Plasma blockchains with distinct business logic and smart contract terms to live simultaneously. Large-scale decentralised app operations can be made scalable and reliable with Plasma [41]. Blockchain networks can attain high throughput and scalability by utilising Layer 2 technologies, all while maintaining the security and decentralisation that the main-chain offers. Blockchain technology can be utilised for a greater variety of applications, such as large-scale decentralised finance, gaming, and other real-time systems, thanks to its Layer 2 processing capacity of hundreds or thousands of transactions per second. This development

is essential for getting past the scalability limitations of conventional Layer 1 systems and increasing the applicability and usability of blockchain technology globally.

5.2 Alternatives for contract management:

In some, **blockchain** technology plays an important role in the secure execution of smart contracts. By leveraging **cloud-based platforms**, **encrypted storage**, and systems designed for **flexibility** and **trust**, businesses can manage contracts effectively without being limited by blockchain's inherent constraints. These systems ensure that contract requirements are met, provide a foundation of trust among parties, and offer the **security** and **transparency** needed in today's digital environment. This approach enables businesses to realize the advantages of smart contracts while sidestepping some of the practical challenges posed by blockchain technology. Fabasoft Contracts [42] is one of the newest solutions for managing contracts. Cloud-based contract management software offers numerous advantages, including enhanced security, efficiency, and compliance, while also supporting transparency in supply chains and product authentication. By automating the verification, enforcement, and management of contracts, this software reduces manual effort, mitigates risks, and ensures that contracts are adhered to throughout their lifecycle. Whether used to track perishable goods or verify the authenticity of products, such software is becoming a critical tool in improving business operations, especially in industries where trust, security, and efficiency are paramount [43].

6. Blockchain-based Smart Contract Programming Languages

Programming languages for blockchain-based smart contracts come in a variety of forms, with platforms like as Ethereum providing a selection of alternatives (such as Solidity, Vyper, etc.) to meet various requirements. In contrast, Bitcoin makes use of the more limited Script language. Ethereum stands out for providing flexibility and more options for developers with a variety of skill sets, as evidenced by its support for about 25 languages. As blockchain technology advances, other languages and frameworks ought to emerge, enabling more intricate and interoperable decentralized apps [44]. Given how essential smart contract languages are to a blockchain's operation, security issues could occur if even one of these languages is defective [45].

Low-level languages are organised in a way that allows them to be implemented in a manner that is unique to the underlying execution environment. Humans often find low-level programming to be unintelligible and riddled with obscurities. [46] Some examples of these programming languages are Bitcoin-script and Michelson.

Both developers and the blockchain ecosystem gain a great deal from the adoption of high-level programming languages for blockchain-based smart contracts. A wider variety of developers may now more easily understand the development process thanks to these languages. Additionally, they lower the possibility of mistakes, increase the safety and clarity of smart contracts, and facilitate interoperability by enabling several languages to operate concurrently on the same ledger. All things considered, this facilitates the creation, administration, and analysis of blockchain contracts, encouraging safer and more effective contract administration in decentralized systems provides examples of such languages, including two that are called Solidity and Liquidity.

If such languages provide a balance between the complexity of high-level and low-level languages, they are regarded as being of an intermediate level. The details of this reduction rely on the computing model, type system, logic, and semantics, but they ease the task of doing verification or static analysis of systems. The term "Simplicity" [47], for example, may serve as an example of this linguistic style.

6. Conclusion

Smart contract technology is developing quickly and has the potential to revolutionize a wide range of industries. Smart contracts can improve productivity, transparency, and trust across a range of industries, including software testing, supply chains, e-government, the Internet of Things, and cyber security, by automating procedures that previously required human intervention. As smart contracts develop further, they could resolve important trust and legal concerns, opening the door to future systems that are more effective, safe, and open.

References

- 1 M. Muneeb, Z. Raza, I. U. Haq and O. Shafiq, "SmartCon: A Blockchain-Based Framework for Smart Contracts and Transaction Management," in *IEEE Access*, vol. 10, pp. 23687-23699, 2022, doi: 10.1109/ACCESS.2021.3135562.
- 2 J. Sun, S. Huang, C. Zheng, T. Wang, C. Zong and Z. Hui, "Mutation testing for integer overflow in ethereum smart contracts," in *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 27-40, Feb. 2022, doi: 10.26599/TST.2020.9010036.
- 3 S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system bitcoin: A peer-to-peer electronic cash system," Bitcoin. org. Disponible en <https://bitcoin.org/en/bitcoinpaper>, 2009.
- 4 V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent advances in smart contracts: A technical overview and state of the art," *IEEE Access*, vol. 8, pp. 117 782– 117 801, 2020.
- 5 W. Xiong and L. Xiong, "Data Trading Certification Based on Consortium Blockchain and Smart Contracts," in *IEEE Access*, vol. 9, pp. 3482-3496, 2021, doi: 10.1109/ACCESS.2020.3047398.
- 6 I. A. Omar, R. Jayaraman, M. S. Debe, K. Salah, I. Yaqoob and M. Omar, "Automating Procurement Contracts in the Healthcare Supply Chain Using Blockchain Smart Contracts," in *IEEE Access*, vol. 9, pp. 37397-37409, 2021, doi: 10.1109/ACCESS.2021.3062471.
- 7 Nakamoto S Bitcoin: A peer-to-peer electronic cash system. Available online at <https://bitcoin.org/bitcoin.pdf> (2008). Last accessed: 2020-10-20
- 8 Lewis A A gentle introduction to smart contracts. Available online at <https://bitsonblocks.net/2016/02/01/gentle-introduction-smartcontracts/> (2016). Last accessed: 2020-10-07
- 9 Nxt community: Nxt whitepaper. Available online at <https://nxtdocs.jelurida.com/NxtWhitepaper> (2016). Last accessed: 2020-10-07
- 10 Buterin V et al (2014) A next-generation smart contract and decentralized application platform. White paper
- 11 Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, Enyeart D, Ferris C, Laventman G, Manevich Y et al (2018) Hyperledger fabric: A distributed operating system for permissioned Blockchains. In: *Proceedings of the Thirteenth EuroSys Conference*, ACM, pp 30
- 12 A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao and Y. Zhang, "A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System," in *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5914-5925, 1 April, 2021, doi: 10.1109/JIOT.2020.3032997.
- 13 Y. Madhwal, Y. Borbon-Galvez, N. Etemadi, Y. Yanovich and A. Creazza, "Proof of Delivery Smart Contract for Performance Measurements," in *IEEE Access*, vol. 10, pp. 69147-69159, 2022, doi: 10.1109/ACCESS.2022.3185634.
- 14 C. Wu, J. Xiong, H. Xiong, Y. Zhao and W. Yi, "A Review on Recent Progress of Smart Contract in Blockchain," in *IEEE Access*, vol. 10, pp. 50839-50863, 2022, doi: 10.1109/ACCESS.2022.3174052.
- 15 T. Li, Y. Fang, Z. Jian, X. Xie, Y. Lu and G. Wang, "ATOM: Architectural Support and Optimization Mechanism for Smart Contract Fast Update and Execution in Blockchain-Based IoT," in *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 7959-7971, 1 June, 2022, doi: 10.1109/JIOT.2021.3106942.

- 16 T. M. Hewa, Y. Hu, M. Liyanage, S. S. Kanhare and M. Ylianttila, "Survey on Blockchain-Based Smart Contracts: Technical Aspects and Future Research," in *IEEE Access*, vol. 9, pp. 87643-87662, 2021, doi: 10.1109/ACCESS.2021.3068178.
- 17 Y. Wang et al., "SPDS: A Secure and Auditable Private Data Sharing Scheme for Smart Grid Based on Blockchain," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7688-7699, Nov. 2021, doi: 10.1109/TII.2020.3040171.
- 18 M. Stefanović, Đ. Pržulj, S. Ristić, D. Stefanović and D. Nikolić, "Smart Contract Application for Managing Land Administration System Transactions," in *IEEE Access*, vol. 10, pp. 39154-39176, 2022, doi: 10.1109/ACCESS.2022.3164444.
- 19 M. Jurgelaitis, L. čeponienė and R. Butkienė, "Solidity Code Generation From UML State Machines in Model-Driven Smart Contract Development," in *IEEE Access*, vol. 10, pp. 33465-33481, 2022, doi: 10.1109/ACCESS.2022.3162227.
- 20 V. Capocasale and G. Perboli, "Standardizing Smart Contracts," in *IEEE Access*, vol. 10, pp. 91203-91212, 2022, doi: 10.1109/ACCESS.2022.3202550.
- 21 L. Zhang, Z. Zhang, W. Wang, Z. Jin, Y. Su and H. Chen, "Research on a Covert Communication Model Realized by Using Smart Contracts in Blockchain Environment," in *IEEE Systems Journal*, vol. 16, no. 2, pp. 2822-2833, June 2022, doi: 10.1109/JSYST.2021.3057333.
- 22 A. Saini, D. Wijaya, N. Kaur, Y. Xiang and L. Gao, "LSP: Lightweight Smart-Contract-Based Transaction Prioritization Scheme for Smart Healthcare," in *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 14005-14017, 1 Aug.1, 2022, doi: 10.1109/JIOT.2022.3145406.
- 23 W. Nam and H. Kil, "Formal Verification of Blockchain Smart Contracts via ATL Model Checking," in *IEEE Access*, vol. 10, pp. 8151-8162, 2022, doi: 10.1109/ACCESS.2022.3143145.
- 24 S. Tang, Z. Wang, J. Dong and Y. Ma, "Blockchain-Enabled Social Security Services Using Smart Contracts," in *IEEE Access*, vol. 10, pp. 73857-73870, 2022, doi: 10.1109/ACCESS.2022.3190963.
- 25 S. Ji, D. Kim and H. Im, "Evaluating Countermeasures for Verifying the Integrity of Ethereum Smart Contract Applications," in *IEEE Access*, vol. 9, pp. 90029-90042, 2021, doi: 10.1109/ACCESS.2021.3091317.
- 26 M. S. Chishti, F. Sufyan and A. Banerjee, "Decentralized On-Chain Data Access via Smart Contracts in Ethereum Blockchain," in *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 174-187, March 2022, doi: 10.1109/TNSM.2021.3120912.
- 27 H. Cheng, Q. Hu, X. Zhang, Z. Yu, Y. Yang and N. Xiong, "Trusted Resource Allocation Based on Smart Contracts for Blockchain-Enabled Internet of Things," in *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 7904-7915, 1 June1, 2022, doi: 10.1109/JIOT.2021.3114438.
- 28 N. S. Patel, P. Bhattacharya, S. B. Patel, S. Tanwar, N. Kumar and H. Song, "Blockchain-Envisioned Trusted Random Oracles for IoT-Enabled Probabilistic Smart Contracts," in *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14797-14809, 1 Oct.1, 2021, doi: 10.1109/JIOT.2021.3072293.
- 29 C. Valencia-Payan, J. F. Grass-Ramírez, G. Ramirez-Gonzalez and J. C. Corrales, "A Smart Contract for Coffee Transport and Storage With Data Validation," in *IEEE Access*, vol. 10, pp. 37857-37869, 2022, doi: 10.1109/ACCESS.2022.3165087.
- 30 L. Ouyang, Y. Yuan and F. -Y. Wang, "Learning Markets: An AI Collaboration Framework Based on Blockchain and Smart Contracts," in *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14273-14286, 15 Aug.15, 2022, doi: 10.1109/JIOT.2020.3032706.
- 31 H. R. Hasan, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar and S. Ellahham, "Blockchain-Enabled Telehealth Services Using Smart Contracts," in *IEEE Access*, vol. 9, pp. 151944-151959, 2021, doi: 10.1109/ACCESS.2021.3126025.
- 32 S. Wang, D. Ye, X. Huang, R. Yu, Y. Wang and Y. Zhang, "Consortium Blockchain for Secure Resource Sharing in Vehicular Edge Computing: A Contract-Based Approach,"

- in *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1189-1201, 1 April-June 2021, doi: 10.1109/TNSE.2020.3004475.
- 33 I. A. Omar, R. Jayaraman, M. S. Debe, K. Salah, I. Yaqoob and M. Omar, "Automating Procurement Contracts in the Healthcare Supply Chain Using Blockchain Smart Contracts," in *IEEE Access*, vol. 9, pp. 37397-37409, 2021, doi: 10.1109/ACCESS.2021.3062471.
- 34 L. Zhang, Z. Zhang, W. Wang, Z. Jin, Y. Su and H. Chen, "Research on a Covert Communication Model Realized by Using Smart Contracts in Blockchain Environment," in *IEEE Systems Journal*, vol. 16, no. 2, pp. 2822-2833, June 2022, doi: 10.1109/JSYST.2021.3057333.
- 35 J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo and T. Chen, "Defining Smart Contract Defects on Ethereum," in *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 327-345, 1 Jan. 2022, doi: 10.1109/TSE.2020.2989002.
- 36 X. Wang, J. He, Z. Xie, G. Zhao and S. -C. Cheung, "ContractGuard: Defend Ethereum Smart Contracts with Embedded Intrusion Detection," in *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 314-328, 1 March-April 2020, doi: 10.1109/TSC.2019.2949561.
- 37 Haijun Wang, Yi Li, Shang-Wei Lin, Cyrille Artho, Lei Ma, Yang Liu , "Oracle-Supported Dynamic Exploit Generation for Smart Contracts" , *Computer Science , Cryptography and Security*, Submitted on 14 Sep 2019 (v1), last revised 18 Sep 2019 (this version, v2)
- 38 P. K. Sharma, N. Kumar and J. H. Park, "Blockchain-Based Distributed Framework for Automotive Industry in a Smart City," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4197-4205, July 2019, doi: 10.1109/TII.2018.2887101.
- 39 B. Thuraisingham, "Blockchain Technologies and Their Applications in Data Science and Cyber Security," 2020 3rd International Conference on Smart Blockchain (SmartBlock), 2020, pp. 1-4, doi: 10.1109/SmartBlock52591.2020.00008.
- 40 Dryja T, Glasbergen G-J, Lovejoy J Layer 2 - the lightning network. Available online at <https://dci.mit.edu/lightning-network/> (2019). Last accessed: 2020-10-20
- 41 Poon J, Buterin V (2017) Plasma: Scalable autonomous smart contracts, pp 283–295
- 42 Fabasoft: Fabasoft contracts. Available online at <https://www.fabasoft.com/en/products/fabasoft-contracts> (2020). Last accessed: 2020-10-07
- 43 Dangl A Top trends 2020: Hyperautomation and smart contracts. Available online at <https://www.fabasoft.com/en/news/blog/top-trends-2020-hyperautomation-and-smart-contracts> (2019). Last accessed: 2020-10-07
- 44 A. J. Varela-Vaca and A. M. R. Quintero, "Smart contract languages: A multivocal ´ mapping study," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–38, 2021.
- 45 N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *International conference on principles of security and trust*. Springer, 2017, pp. 164–186.
- 46 G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- 47 D. Harz and W. Knottenbelt, "Towards safer smart contracts: A survey of languages and verification methods," *arXiv preprint arXiv:1809.09805*, 2018.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

