International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2020/195718 Volume 7, Issue 2, March – April (2020)

REVIEW ARTICLE

An Empirical Review on Blockchain Smart Contracts: Application and Challenges in Implementation

Jasvant Mandloi
Department of Information Technology, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore, India.
jasvant28284@gmail.com

Pratosh Bansal
Department of Information Technology, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore, India.
pratosh@hotmail.com

Published online: 30 April 2020

Abstract – This paper focuses on an extensive review of Blockchain Smart contract applications in real-world scenarios and challenges in it. As in today's world, Blockchain has got tremendous importance in different types of services, so it will be beneficial to analyze the loopholes in the cybersecurity aspect as well as other technical issues in privacy and governance. A smart contract is the software code that comprises a self-executed set of rules and regulations that runs on Blockchain. To implement various real-world services over different decentralized platforms, a lot of research work is going on. In this paper, a detailed review is done on more than 100 latest published documents based on the Blockchain Smart contract and applications. By following the systematic mapping research methodology first, we have identified paper relevant to our research domain, and then by applying various filters, we extracted relevant information. At last, the article highlighted the challenges and research gaps that have to be addressed in the future.

Index Terms – Smart Contracts, Blockchain, Consensus, IoT, Bitcoins, Cryptocurrency.

1. INTRODUCTION

1.1. About Blockchain Technology

In the modern era of Information technology, we are gradually shifting towards a more digital outlook. In the last decade, the concept of cryptocurrency (Bitcoin) on the distributed ledger called Blockchain has opened opportunities for different types of services [1] [2]. It also unlocked an exciting research area in this field as primarily from the perspective of security, privacy, and other technical issues in implementing it [3]. The impact of Blockchain can be understood by the fact that $339.5 million global spending on the Blockchain solution is done in the year 2017, and the global Blockchain market is predicted to be worth in 2021 [4]. Bitcoin is the brainchild of "Satoshi Nakamoto" first cryptocurrency based on Blockchain and considered to be the first real-world application on this concept in the year 2008[5]. A Blockchain is a form of distributed ledger which stores all completed transactions. The Blockchain technology has some key features that gave an edge over the present centralized techniques like decentralization, persistence, anonymity, auditable, and provenance [6].

Cryptocurrencies have occurred to be considered as the 1st generation in Blockchain technology. In this, Bitcoin is the first popular currency developed using the Blockchain concept. Blockchain architecture developed after Bitcoin, such as Ethereum, which supports complex distributed applications other than cryptocurrencies [7] [8]. It is the second generation as it can help complex services and allows you to write a self-executed code called Smart contracts.

1.2. About Smart Contracts

A smart contract is the software program that comprises a self-executed set of rules and regulations that runs on Blockchain [9] [10]. To implement various real-world services on the different decentralized platform uses cases are already proposed, i.e., supply chain, IoT, and a lot of research work are going on [10]. It is not a new concept; Nick Szabo first introduced it in 1994, has basic properties of Self-confirming, auto-executing, and tamper-resistant [11].

It gets attraction after 2008 when Blockchain technology comes into the real integration with it to allow the peer-to-peer transaction and available publicly with security and trust. By Figure 1 (a) we get the basic idea about the smart contracts where it resides in the Blockchain and essential elements of it [12]. A Smart Contract is a set of programs that exist on the Blockchain uploaded by a participating node in the network.

Functions recognized in the smart contracts with the unique address assigned to them.

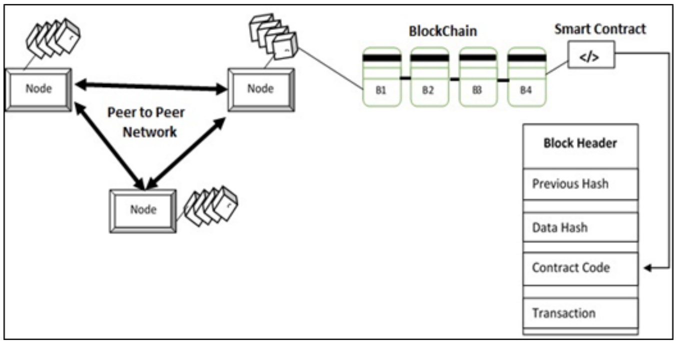

Figure 1 (a) Smart Contract Architecture

It has a unique address of block through which a smart contract is recognized, and it has the Unique Blockchain address by which it is identified. In smart contracts, there is a set of executable functions and a separate set of variables. The function in the smart contracts gets executed whenever a transaction is completed to these functions. The transaction contains input parameters that are needed for these functions. After the execution of the Smart Contract condition parameter in the contract gets changed based on the criteria used in the method [13].

Figure 1 (b) illustrates the smart contract primary components. The smart contracts are developed using coding languages like Solidity, Python, and Formality.

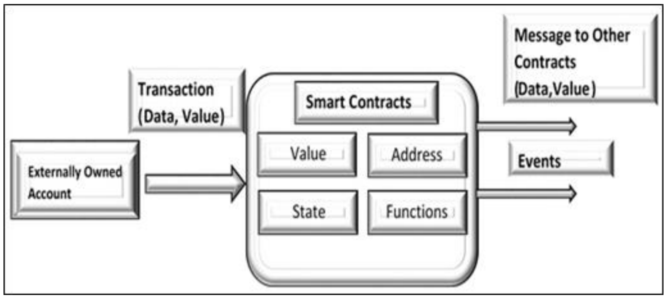

Figure 1 (b) Main Elements in the Smart Contract

For compiling smart contracts, language-based compilers are used [14]. It gets converted to bytecode and then uploaded on the Blockchain network with a unique address. Nowadays, Blockchain smart contract has become a mainstream research area, and Much work is done in this area to develop apps like IoT, music rights, E-voting, and implementing it for banking applications [7, 15]. More efforts are made to establish services using smart contracts that can replace the traditional services with the trusted decentralized Blockchain. It still has challenges when it comes to the implementation in the real world scenario. For each problem, Blockchain or smart contracts cannot be the solution for that, and we have to analyze the applicability and usability of it. It has security, privacy, and governance or legal issues. In [16], the author raises security concerns like the majority attack (51%) because the mining pool created it takes control of the Blockchain. Another security issue raised is a fork problem in which when there is an update in the Blockchain software because of that, there may be chances that inconsistency occurs. In the development of Smart contracts, language selection is also a big challenge, presently what programming languages are used in Ethereum and Hyper ledger are prone to faults and bugs [17]. In the present scenario, languages used for the development are in the early stage, and minor bugs or errors in the code will invite unfortunate incidents. In the year 2016, because of the vulnerabilities in the code of DOA, it put the loss of $40 M in the ether network. [18]. Some authors worked on the programming loopholes after performing the case study and conclude that unsafe programming will permit an attack wallet, which results in a loss of $150 M in November 2017. Another challenge with smart contracts is a myth that they are immutable.

As soon as a Smart contract is developed and deployed on the Blockchain, no more alterations are allowed. [19] the solution to overcome the above problem is by developing software engineering standards, especially for Blockchain and smart contracts. The smart contract for the public domain is first introduced on the Ethereum platform. On this platform, research work is done to enhance the security and privacy of the smart contract, and issues identified are a coding issue, timestamp dependency, criminal contracts, and untrusted data feed [7,20]. The smart contracts are implemented on the Blockchain network, so facing data privacy issues because of transactional transparency and the absence of data feed privacy. To reduce the corruption and middleman charges, it is an overwhelming response to adopt the Blockchain technology for the public and private services. For the successful implementation of it with the above challenges governance, legal issues are also there, and plenty of analysis is required before adopting the Blockchain. The success of any Blockchain implementation in the public sector depends on how regulatory authority provides the data access, and it is under different jurisdictions [21]. For identifying the research gap in the current smart contract development and implementing real-world applications using it. We have decided to perform the systematic survey on it following the methodology suggested in [22].

2. RESEARCH METHODOLOGY

To conduct a review on the smart contract, we used a systematic mapping technique. The purpose of a systematic mapping analysis is to present a survey of the research field, to find out if research data exists and to understand the extent of work done and the evidence available [22]. In our paper, we have followed the recommendations suggested for a proper literature review described by Kitchenham and

Charters [22]. To explore related documents, we have adopted a systematic mapping strategy as our study methodology as our focus was to review current studies related to Smart Contracts and distributed Technologies. The study's findings will make it possible to let us recognize research areas related to Smart Contracts and Blockchain technology and probable research shortfalls. Figure 2 shows the various steps followed to perform the review.

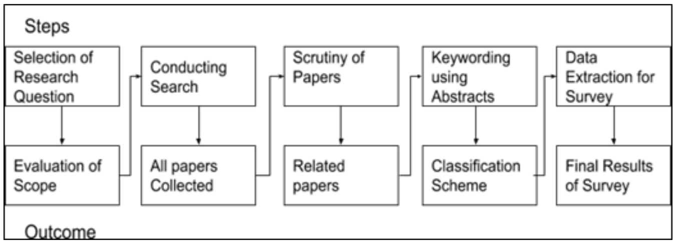

Figure 2 Steps Followed to Perform Review

To perform the Systematic Survey, we have covered the following research questions in the paper.

2.1. Research Questions

Q.1 Which publication sources is selected to identify the research work done in the domain of Blockchain smart contracts.

Motivation: - To get an idea about the Blockchain smart contracts publication details, it will also help at the time of our publication to select a good platform.

RQ.2 What are the different areas in the area of Blockchain smart contracts that are touched.

Motivation: - To identify the information available in the current research channels about the Blockchain smart contracts.

RQ3.What are the different types of services offered or proposed using the Blockchain Smart contracts?

Motivations: - To explore the different areas in which BlockChain Smart contracts are in use. Also, to get information based on the available research literature where future real-world services are proposed using BlockChain Smart contracts.

RQ4.What are the issues in the Blockchain Smart contracts for implementing real-world services over it?

Motivation: - To identify the issues that occurred in the Blockchain smart contracts and the impact of it on the applications. It also helps to determine the cause of the problems and categorization of it.

RQ5.What is the different Blockchain Environment used to develop and implement smart contracts?

Motivation: - To explore different Blockchain environment used to create and implement smart contracts on it. We also get information about the type of infrastructure available and technical details related to it. With this also searched for the available simulators and frameworks to test it before implementation.

- Blockchain adoption.
- The programming language used to develop Smart contracts
- Consensus mechanism used

RQ6. What are the available possible solutions to the problems in the existing scenario?

Motivation: - To detail the current status of the convenient solution to the issues identified and to provide details about the performance of the proposed or used solution.

2.2. Conducting the Research

In this stage, the study is done to find out all the related research work on this domain. The search methodology specifies the process that will be used to conduct a detailed, systematic search for literature. We created a search methodology that is used to extract research content relevant to the topic from the different databases through web resources. The term used for initial searching is challenges in Smart Contract implementation. Afterward, we decided to use a common phrase "Smart contract" & "Blockchain Smart contracts" even though we can add an application of Smart contracts but get a massive number of papers, and they are more related to the business and financial aspects. However, our goal is to do the mapping study for the technical aspects related to the Blockchain Smart contracts and implementation issues.

After determining the search parameter, we selected scientific databases for the search. We finalized the following resources to collect research papers and content related to the topic.

1. IEEE-Xplore
2. ACM_Digital_library
3. Springer
4. Elsevier
5. Science-Direct
6. Research Gate
7. aRxiv
8. Others

Also, get some excellent papers from journals like PLOS but are very less, so we had put all such documents under the tag ‘others.’

2.3. Scrutiny of Relevant Papers

All the papers searched are not related to the research questions addressed in the paper. It needs to be checked all the examined documents to find out their actual relevance [22]. After using a method to search for research papers in the research database available on the Internet, the next step is to scrutinize the papers. In the first step, research papers are selected on the basis of titles that are relevant to the research question, and material found irrelevant is excluded. To narrow down our research papers in the next step, we have read out the abstract of all the documents selected in stage one. With this, we have used specific criteria for selection and rejection, and the following points are considered to include the research paper.

- Papers should not be in the poster category.
- Papers should be in English.
- Papers should be related to computer science, not theoretical or any business idea.
- All the papers that are emphasizing on Smart contracts with Blockchain.

2.3.1. Key Phrasing Based on the Abstract

The next step in the mapping process following the initial abstract recognition of the targeted papers is critical phrasing. In the first step, the abstract can be read in two stages and main phrase can be generated that is applicable to the research work. In the second phase, with a higher degree of understanding with the key phrases and creating a cluster based on the categories useful for the study. After reading all the selected paper abstracts, some updates are done in the categories, or new categories are formed.

2.3.2. Data Extraction and Mapping Process

The data retrieval pattern is designed to collect the information required to respond to the quaternary finalized for the research study. The Figure 3 depicts the grouping scheme.

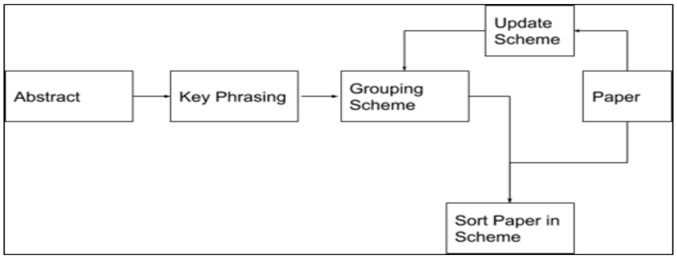

Figure 3 Creating Grouping Scheme for Papers

A data extraction parameter is selected to address the information required to address the research questions chosen for the study. Basic parameters are like a year of publication name of the author, publication channel. Some more information is also retrieved after reading the papers. The information collected from the research paper based on the below parameters (Table 1) is stored in the excel file to analyze the data.

SL.No	Parameter	Description
P-01	Paper Id	Unique Id to the papers selected for
P-02	Title	Title of the paper selected for the
P-03	Authors	Author Details
P-04	Publishers	Details of the publication where it is
P-05	Publication	Conference /Workshop/Journal
P-06	Abstract	Abstract in the paper
P-07	Study	The outcome of the paper
P-08	Research	Information related to research

Table 1 Parameters for Information Extraction

2.3.3. Basic Information of the Papers

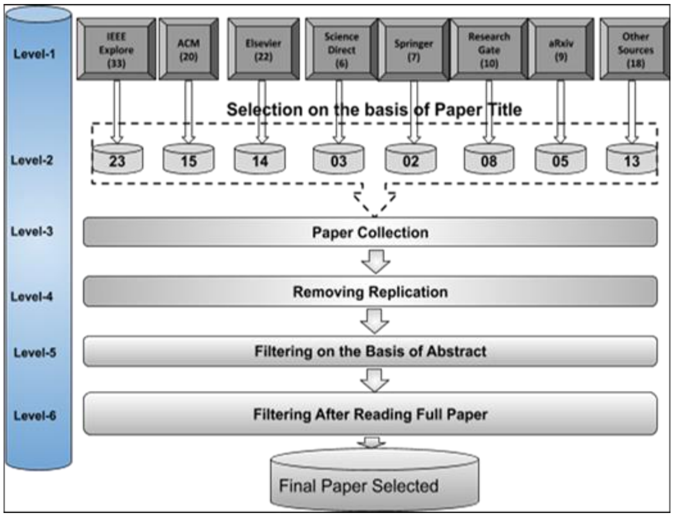

Figure 4 Search & Selection Process

2.3.4. Publication Year, Source & Publication Type with Channel

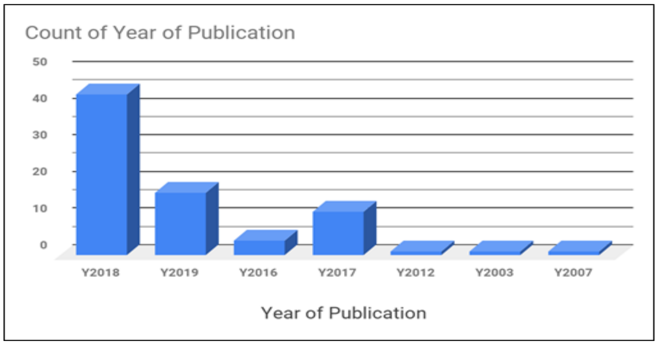

Figure 5 (a) Year of Publication

Figure 4Shows the search and selection process of the paper. Figure 5 (a) shows the year of publication and Figure 5 (b) depicts the publication sources.

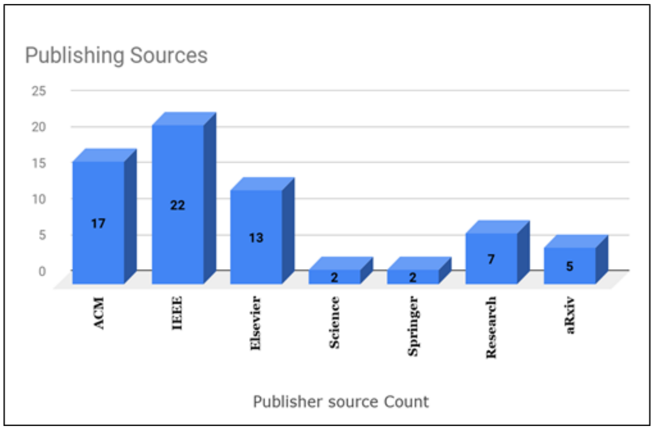

Figure 5 (b) Publication Sources

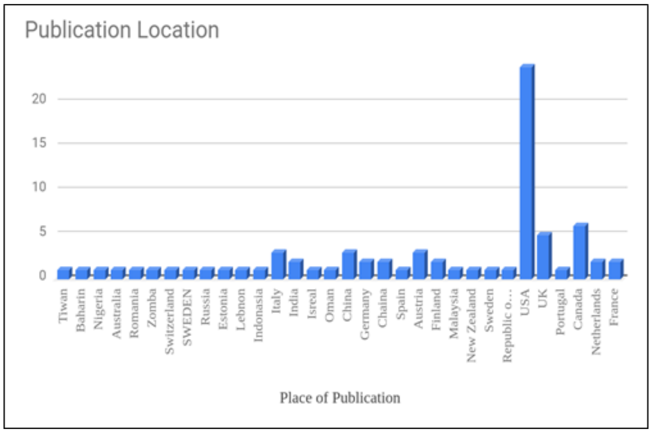

Figure 6 (a) Publication Location

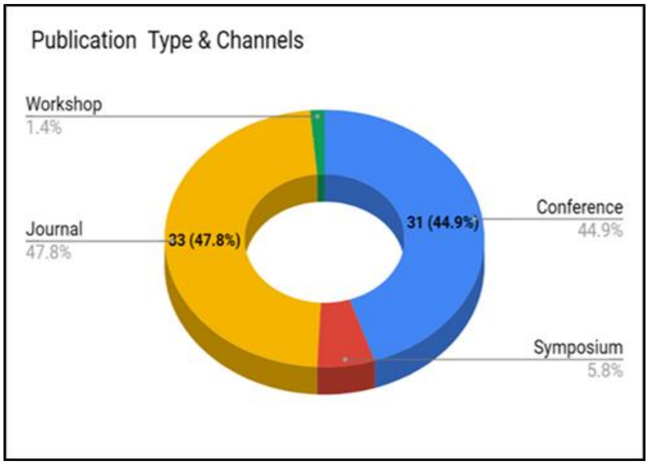

Figure 6 (b) Publication Source Type & Channels

Figure 6 (a) shows the publication location and Figure 6 (b) depicts the publication source type and channels.

2.4. Classification of the Relevant Papers

Grouping of the papers made on the basis of P-06 & P-07 after review of all the papers identified and the final grouping on the basis of the findings. We have identified that major papers are focusing on the Blockchain based smart contract security, on the basis of the consensus algorithm, Blockchain Platform, and application of the smart contracts. We have classified paper based on the three research questions (3,4,5) and identified the problem and existing solutions for that and also highlighted areas where work is required to be done.

In these three main types of papers are collected that contains the application of Blockchain smart contracts, privacy, and security flaws in implementing real-world services, Blockchain Platforms used to develop smart contracts.

3. APPLICATION OF BLOCKCHAIN SMART CONTRACTS

Scope for Blockchain is now seen in each and every field of day to day life like applications in various fields include cryptocurrencies, education and healthcare, advertisement, public liability, intellectual property security, energy, and community applications. [23]. Blockchain technology's ability to register transactions on distributed ledgers gives governments a new way to enhance transparency, avoid corruption, and strengthen confidence in the public sector. The results show that there is still minimal acceptance of Blockchain-based applications in e-government, and there is a shortage of empirical evidence. The key barriers to the adoption of Blockchain are primarily defined as technical aspects such as safety, scalability, and versatility. In the meantime, legislative and regulatory support for the need for new governance models and the use of this technology are key operational challenges [24].

3.1. Internet of Things

In recent days, the significant increase in IoT applications has given rise to access to big insecure data. IoT devises data security, and privacy is a big challenge in this area. Many researchers already discussed the use of Blockchain technology to address these issues [25]. Rouhani and Deters have done a systematic survey on the application of Blockchain smart contracts in IoT and collected information from various resources that highlight the use of a Blockchain-based control mechanism for access control. Blockchain smart contract-based Semantic Thing Web Framework (SWoT) explores the design of smart contracts for resource registration, discovery, and selection.

3.2. Healthcare

A Blockchain-based electronic health record uses a searchable encryption mechanism. The index for Electronic health record is designed by applying complicated logic expressions and saved in the Blockchain network to facilitate a data user to browse the index using the expressions. Since only the database is imported to the Blockchain to facilitate propagation, the owners of the data have full control over who can view their electronic health record information [26]. Based on Blockchain, the author has proposed a new mechanism called the origin chain. It is used for protection having features like high-availability, flexibility, traceability of data. It also allows for automatic regulatory compliance tracking and adaptation in drug traceability scenarios [27].

3.3. Supply Chain Management

The data immutability feature of Blockchain is the key reason for using it in supply-chain management. The benefit of using distributed ledger technology in the supply-chain is tracking and controlling the steps from product development to distribution, retaining quality control, offering an integrable and reliable process. In the paper, the author collected data from the sources that were gathered by the Finnish enterprise consortium's Blockchain experts by interviewing 30 companies and operating in 36 countries to know the requirement of the industry. Also, the author explored the Quality Function Design (QFD) process, in that smart contract ledger is more acceptable for Blockchain apps used to integrate production chain instead of transactions and hash functionalities better. Chen et al. [28] introduce an insurance framework for the supply chain quality, consisting of four levels, IoT tools, decentralized ledger, smart contract, and business level. IoT devices produce the data required to track the output of products and services, resources, and transactions. Blockchain level stores the generated data in a transparent and secure ledger.

3.4. Business Cycle Management

The major issue of concern in the cooperative business cycle of companies is a lack of confidence, and like many other technologies, Blockchain can be seen as a solution to this problem. In addition, the author has reviewed three cases for the incorporation of Blockchain in BCM, which have been developed by creating a structure for turning the cooperative business process into an industry contract and then performing its cases as a smart contract. In the paper, a scenario that implemented a workflow case study of order processing using Blockchain and Hyper-Ledger Composer authorized by Hyperledger Fabric was also addressed. Implementation details include workflow asset retrieval, access control component, and specification [25].

3.5. Maintaining Records

One of the basic features of Blockchain systems, such as tamper-proof and trustable databases makes it possible to use for record storage. In the paper [25] author discussed one case for investigating the capacity of Blockchain for record-management applications. By addressing things about the conservation and reliability of records due to the absence of a reliable digital archive and benefits such as low-cost processing.

In another case, the author has mentioned that there are three attributes, precision, durability, and credibility of a verified document. Such attributes can be guaranteed by a Blockchain encrypt-proof framework that stores documents as a chain of hashes. Reliability needs three prerequisites, creation point completeness, conformity with proper rules, and naturalness [25]. One of the frameworks described is "Blockchain-based contract recording. “Smart contracts act as a mediator in this architecture. Smart contracts are liable for authentication of activities, violations in Service Level Agreements (SLA), and fines estimates. Hence, disagreements are instantly addressed based on smart contract regulations.

3.6. E-Voting

In Hanifa Tunisia [42], Blockchain can help build a secure voting mechanism that preserves privacy. Electoral processes are typically plagued by confidence problems, and the risk of bribery and fraud in centralized voting systems is exceptional. Blockchain technology is one of the solutions for the voting system because it comprises a decentralized network, and many users own the entire database copy. Blockchain itself was used as a distributed banking system in the Bitcoin system. One of the cheating reasons for database tampering can be downgraded by introducing Blockchain in the allocation of datasets on e-voting systems. Using hash values to record the voting results of each polling booth linked to each other makes the reporting system more secure, and the use of digital signatures allows the system more reliable.

McCorry et al. [29] propose a secret voting scheme for the meeting room using smart contacts and a zero-knowledge proof algorithm. Deployment is done on smart contracts with Solidity and distributed Blockchain from Ethereum. The "voting contract" and "cryptography contract" are the two key smart contracts. The voting contract defines the code to vote and verify documentation of zero-knowledge. The cryptography contract will provide the voters with two kinds of zero-knowledge proof: Schnorr and one out of two proofs.

3.7. Digital Identity

Centralized authority for digital identity protection causes various problems like the privacy of user data and untrusted third parties. In addition to that, they do not provide a single identity that can be adopted by all. As a result, many third parties who maintain digital identity may have access to private information of users. In identity management systems, Blockchain and smart contracts can contribute to addressing this matter [25, 30]. In another proposed work related to Self-sovereign identity, which has four main parts that include identification, authentication, verification, and storage. To implement, it requires platforms like Ethereum and Bitcoin that generate identity identifiers and how they are used by decentralized identity systems. The element for authentication uses the (PKI) Public Key Infrastructure with the Zero-Knowledge algorithm to confirm the identity of the user. The verification element signifies the process of obtaining a claim that may have different certificates. The proposed study also focussed on two key methods that are used to connect claims and certificates are "Identity Register" and the "Claim Registry."

3.8. Industrial Application

Tiago. M [31] proposed Industry 4.0, which is a framework designed to facilitate the way modern industries work by using state of the art technologies, which are often used to build IoT for industry applications automation or Big-Data. For the modern industry, requirements, different types of Blockchain Public, Private, federated architecture are proposed, and smart contract applications are also intended to implement the industry 4.0 framework. After offering a general framework to decide how using a Blockchain is a suitable option for implementing an Industry 4.0 program. The most suitable industrial Blockchain-based technologies for each Industrial 4.0 platform were studied, and also their crucial challenges.

Jameela Al-Jaroodi [32] various industrial technology areas are discussed in this paper, and the use of Blockchain technology has been proposed. Besides, it discusses the possibilities, costs, and obstacles of integrating Blockchain into various applications for the industry. The main aim of the author is to define the specifications that enable Blockchain deployment for various applications in the industry. In the paper, up to a certain extent, they are able to identify the key criteria and main challenges facing the effective use of Blockchain technologies in the industrial domains. Some of the challenges are to be addressed by introducing new features and with the current techniques in other applications. Research has been done to increase the performance of Blockchain technology in these applications. Smart contracts may be used for all government pay systems as a way to improve the fairness of deals as well as try to prevent overbilling, as long as contracts and bids are traditional ways of money laundering and misappropriation of funds [33].

C Udokwu[34] in the review examined various kinds of businesses where smart contract implementations are going on and its concerns to minimize their use. Contributing to the main research issue of working there, how to implement smart contracts within present enterprises effectively?. To find out the answers related to the above question, they have surveyed the major sections where smart contracts are going to be used, and identified trust and transparency are the two main features that attract enterprises to shift over smart contracts. Herewith this author also identified limitations of the technology like compatibility and implementation, Standardization, lack of study, and practical experience, as well as worries about architectural design.

3.9. The Banking Sector

The author has proposed a loan based Blockchain (LoC), a new banking loan management system focused on smart contracts over the approved Blockchain Hyper Ledger Fabric. By taken as a case study, the Chinese poverty alleviation loan. The proposed work is to establish a digital account framework for the exchange of assets among central and decentralized directories and to introduce locking and unlocking algorithms for smart contracts.

F. Casino [35] has done a systematic literature review on the Blockchain-based application on multiple domains. Based on the content survey related to the application of Blockchain in various platforms, the author has identified Blockchain-enabled technologies across a range of fields like supply chain, business, healthcare, IoT, privacy, and data management. It also suggested some research topics, trends, and emerging areas and highlighted the limitation of Blockchain also.

A Ramchandran [36] suggested Blockchain to build a stable and unalterable system for data provenance management that instantly verifies provenance information. They used Blockchain capability as a medium to encourage trusted compilation, authentication, and management of data provenance. Smart contracts and the (OPM) Open Provenance Model are used to build a framework to store permanent data trails. They put forth a framework that can collect and verify provenance data easily and safely, and avoid any fraudulent alteration of the data collected as long since most participants are genuine.

4. ISSUES IN IMPLEMENTING REAL-WORLD SERVICES OVER BLOCKCHAIN SMART CONTRACTS

4.1. Privacy Issues

The Blockchain is the breakthrough technology, but an organization needs it or not has to be decided based on the understanding of the technology [37]. The author addressed the benefits and drawbacks of Distributed ledger technology for the insurance industry and can extend it to anyone. In the report, they discussed crucial questions to answer before adopting Blockchain technology. The questions are is it required to have a shared database? ; is it mandatory to have several parties involved in writing the data? , are the potential writers untrusted? ; is it required to remove trusted intermediaries?. After that, they apply it to the insurance sector and conclude that not all insurance scenarios will be profited by it. The author suggested that acceptance should be sensibly evaluated depending on an organization’s profile and business plans. In the last decade, Smart Contracts on

Blockchain technology emerged as the solution for the regulatory and technical challenges in the centralized or third-party system. It has an advantage over the existing workflow but has some implementation challenges. [38] The Blockchain system is considered as immutable, it means that if the Blockchain technology gets enough level of validation, in that cryptography assures that it cannot be replaced or reversed. It gives Blockchain edge over the conventional file system or database, immutability can achieve through the consensus algorithm, and the same applies to Smart Contracts. In the case of private Blockchain, validators are selected by the Blockchain community itself. However, concerning public Blockchain, it is entirely dependent on the consensus algorithm and anyone on the internet who can solve the cryptographic puzzles and validate it. This problem occurs when the fork happens at different nodes in the network, will see the different blocks, and have a different opinion about the chain's history. In the present scenario, public Blockchain immutability is not guaranteed, and it depends more on economics rather than trusted parties. Even if it is introduced by the government or any big organization to bring down the network, they can support the fact that this would be a delicate and expensive operation. In the case of Cryptocurrency will only get more secure as their value and mining capacity grow continuously. However, in the case of the other applications of the enterprise and other institutions, proof-of-work immutability makes no meaning at all.

The government is adopting new technologies like Blockchain Smart contracts and deploying public services over it to provide better public administration, i.e., recordkeeping, land registry [39]. At present, the noticeable concern is not the transparency of the software, but the confidence in the Distributed ledger on the part of Blockchain developers, policymakers, government agencies, and the public at large. This faith relies entirely on technology, rather than managing authority credibility to assure that system and data reliability. The possibility of technology failure is always there, and if government services like user identity management are implemented over it, a single mistake may cause disastrous results.

Most of the Blockchain handlers are prone to privacy attacks; various experts recommend using an unidentified communication network, namely TOR, to guarantee access privacy. [40] In this author raises the issue for the mechanism required through which unidentified users can publish and execute transactions and does not allow anyone to connect such transactions to their network addresses.

A protocol is proposed that tries isolating the transactions performed by pseudonymous identities and not allowed to link or retrieve details from the data that appears in the Blockchain. As in the present scenario, none of the proposed protocols are there that can hide details of the users from network-level attackers as the user submit or retrieve the data from the Blockchain network. Emerging privacy-centric cryptocurrency, namely Zcash and Monero, deploy cryptographic primitives such as Zero-knowledge succinct. It has features like non-interactive information, traceable ring signatures, secret transactions, and stealth addresses to provide better privacy than Bitcoin. The TOR is used to maintain privacy, but the limitation is designed for low latency applications like communicating using web surfing and real-time quick messaging. To adopt the Blockchain smart contracts for the real world applications other than the cryptocurrency, require privacy protocols that delink users network-level information from their transaction. By using the infrastructure of a Blockchain network, where the connection takes place for the announced transactions. Some approaches are also proposed to improve privacy for route-based transactions. It is an interesting open challenge to develop a mechanism for performing multi-hop transactions privately against the network-level adversary.

Smart contract’s popularity increased in the last few years, to the extent that millions of dollars are now exchanged every day over it. The Blockchain platform that presently is popular for smart contracts is Ethereum, which is designed to support smart contracts. The programming language used in it is solidity developed under the banner of Ethereum. In the early day's several attacks are performed by taking advantage of bugs in the code [19]. In the paper, the author has suggested a separate software engineering approach for the Blockchain smart contracts that focus on decentralized programming on the Blockchain platform. In the research work weakness of the library is because of negligent programming practice either due to Solidity. The vulnerability has been exploited anonymously by two measures. In the first step, an attacker takes over the smart contract library and calls the initialization function. Consequently, suicide functions were called, which destroyed the library, creating a situation where it was not feasible to execute functions on smart contracts in the library. As all the delegate calls ended up in a dead smart contract library, and finally, the author suggests that the training of software engineering is expected for smart contracts. In Dec 2017, “CryptoKitties,” a game on Ethereum Blockchain, got immediate success just after its launch. It involved around 1.8 million users with over $20 Million of spending in Ether and was at point lead, taking 12% of all Ethereum transactions. Smart contracts reside over the Blockchain network with storage capability and prove the popularity of the Smart contracts.

Nonetheless, Smart Contracts are still limited in their ability to satisfy all expectations; they require more development in a variety of ways. Most platforms and services over Smart Contracts are in their early stages and come with problems, ranging from a deficiency of regulation to the pseudonymous process of illegal activities [41], in the paper author highlights four main topics are Technology, Legalization, Acceptance, and Usability. Safety is one of the first worries of any distributed ledger system and associated trials. As most of the technical errors are related to the semantic of scripting languages, it is an urgency to improve existing contract languages and develop substitutes. A very effective way to conduct testing and find bugs and errors is to make things open-source and collect feedback from users.

The category of attacks varies from environment to environment. It is required to understand the working and security and privacy breaches of a respective Blockchain platform before adopting it. The Smart contracts for the public domain services are visible to all the users, and because of it always chances of privacy leakage. However, the purpose of keeping secrets is, to some extent, inconsistent with the nature of public Blockchains and hence requires additional operations. To implement it, acceptance of encryption algorithms may cause extra load on the system. There is still no systematic way of using smart contracts for several design purposes, particularly where legal content is involved. From a legal perspective, parameters and policies are absent on smart contracts. The scripting languages used for the Smart Contract are designed in a way that is more comprehensive and simple-to-use for both professionals and newcomers.

Some privacy challenges that may hinder Blockchain wide application. In [42], the author explained that certain types of attacks are possible, i.e., Network analysis, address clustering, Transaction fingerprint, DOS attack, Syble attack. In addition to that, there are chances of transaction pattern exposure through transaction graph analysis, AS-level deployment analysis. With this, there are also chances of mixing in centralized services. When a Blockchain smart contract developed in which human or sensor input data are required, it increases performance and security issues of the system called Oracle problem [43].

4.2. Security Challenges in Implementing Blockchain Smart Contract

To include Blockchain in the mainstream for a typical application like banking, education, and e-voting by using smart contracts still have challenges. In [3, 16], the author did serve to identify the current research scenario in Blockchain, and it performs mainly on the bitcoin network. In their survey, the main findings are security incidents and types of attacks like DDoS, 51% attack, data malleability problem to challenge the integrity of it. They also highlighted the issue of authentication and cited a hardware-based solution for it. Another issue they have raised in their report is about the privacy issue in that the public can visualize the transaction without linking information to the source identity. However, the authors included some facts to prove that anonymity can be breached by doing traffic analysis.

Smart contracts are an enticing aspect of second-generation Blockchain. It takes the attention of the research community working in this area. It is an executable code that runs upon that can enable, implementation of a contract between two private parties without the involvement of a third party. However, it is at the initial stage, and usage of it has just started, and the author [7] has done survey work to identify the current research area in this field. In these four significant areas are identified, i.e., Codifying, security, privacy, and performance issues. Security issues mean bugs in the code or the weaknesses that a challenger might use to launch an attack. The security attacks discussed in the research work are transaction ordering dependency vulnerability [44]. In this, order of execution will depend on the minor as the Blockchain states are changing, a malicious user may take financial advantage by it.

The next security issue raised is the timestamp dependence as smart contracts are using block timestamps as an activating condition to perform some essential operations. Security attacks conducted using the above dependence are like a miner can set a block timestamp to be the exact value that affects the value of the timestamp-dependent condition and favors the miner. Another security concern highlighted is reentrancy vulnerability with the above The Dao hack attacker stealing $ 60 million US dollar through it. In the case of Ethereum, when another contract is called, the current call waits for the call to be completed. It can lead to a problem when the receiver of the call makes use of the intermediate state the caller is waiting. It is a matter of concern when the recipient of the call makes use of in-between state the caller is waiting.

Next to a security issue identified is the criminal activities the possibility of creating three different kinds of illegal activities in a smart contract system, i.e., leakage/sale of an important document, theft if the private key, and a full range of physical world crime [29]. To perform it possible techniques are using Serpent scripting language for leakage of a secret document, stealing of private keys can be achieved using SNARKS cryptographic primitives. After that, the author should shed light on the authenticated data streams, which are data from an external source that may result in the crime of calling the token.

In the Blockchain anonymity of the user, identity plays a crucial role, and it is one of the most highlighted features of this technology. However, challenges related to security and privacy challenges are open, and it is the present scenario requirement that should be addressed [46]. The security breaches and attacks have been identified through a study done by Loi et al. and found that 8833 out of 19366 existing Ethereum contracts are vulnerable. They have served on the security and privacy issues and suggested future treads that are required to be explored, like the cleaning of the unemployed contracts and mechanism to identify it.

S.no	Vulnerability	Reason	Location
1	Call to the unidentified Function	The function that is being called may not occur	Contract source code
2	Out-of-gas send	The alternative of the caller is executed	
3	Exception Handling	Exception handling not properly done	
4	Typecasting	Error due to type checking in contract execution	
5	Reentrancy vulnerability	The function is re-entered before exit	
6	Field disclosure	The miner publishes a private value	
7	Immutable bug	Modify a contract after implementation	Ethereum virtual machine bytecode
8	Ether vanished	Submit Ether to an unknown address	
9	Stack overflow	Stack value range crosses the value of 1024	
10	Unpredictable state	Contract state gets changed before it is invoked	Blockchain mechanism, i.e., Consensus algorithms.
11	Randomness bug	A malicious miner biases seed	
12	Timestamp dependence on the Block	Malicious minor changes timestamp of the block	

Table 2. Details of the Vulnerabilities Occurred in the Smart Contracts Ethereum Platform [20]

Details of the Vulnerabilities Occurred in the Smart Contracts Ethereum Platform [20] is listed in Table 2. The nearest research work is performed in [20], which only includes the smart contracts of Ethereum rather than any other network used in the current scenario. In that, they have listed out security vulnerabilities that occur due to the programming language used for the Smart contracts and point out programming pitfalls that lead to vulnerabilities [20]. In the Blockchain system, a user's private key is regarded as the identity and one of the security credentials. In the process, it is generated by the user and not by the third party. Hartwig et al. [47] find out a flaw in the Electric Curve Digital Signature Algorithm as randomness is not necessary during the signature process; an attacker can quickly recover it. Once the private key is stolen, and it reaches the hands of the criminal, the chances that the user account gets tampered are increased. In the Blockchain environment, it is not possible to track the criminal and information about the modification. The criminal gets control of the smart contracts for a different number of malicious events, and that may have an impact on the life of the regular user. Criminal Smart contracts will help leakage of personal information, theft of cryptographic keys, and various crimes committed in the real world [19]. The code is running on the Blockchain; chances of security vulnerabilities are possible due to semantic defects. Author [19] did a proper examination of Ethereum smart contracts and a list of twelve types of vulnerabilities they may occur.

When a user interacts with the Ethereum through smart contracts, then a certain amount is charged called gas. Gas is just like a wallet balance that can be transformed into an ether, which is also the currency of the Ethereum network. Some Smart contracts expansion and distribution are not effectively [48] highlighting seven gas costly patterns and categorize it into two parts, the first useless code that is available in the code but not going to be used anymore. The second is a loop-related code that is not properly designed, using an analysis suggested by the GASPER method. Which is capable of automatically discovering three kinds of expensive gas patterns in smart contracts, namely dead code, opaque predicate, and expensive loop operation.

Attacks on the Blockchain Smart contracts are also possible through the network level. [49]. Maria et al. rigorously analyze the effect of the BGP hijacking, for performing this type of attack, which requires control over the network parameter. They also carried out a thorough study to test the effect of routing attacks on both the network and the node level on bitcoin. They conclude that the number of active Internet prefaces to be hijacked rests on the sharing of mining resources. This attack will put more impact when the bitcoin mining pools are centralized, and the results of it may be the splitting of the network and delay in the block propagation.

Research work is also conducted to strengthen the security of Blockchain and Smart contracts that run on it. Loi et [50] propose an innovative mining pool system called SMART POOL. The basic working of the system is SMART POOL gets a transaction from the Ethereum node clients that contains mining information. After that, the minor conduct hashing calculations based on the assigned responsibilities and returns the finalized shares to the SMART POOL application. When the number of signed shares exceeds the threshold, they will be added to the SMART POOL contract on the Ethereum. In the next step, the SMART POOL will verify the shares of credit rewards to the client. It has some advantages over the normal peer to peer pool that are decentralized, improved efficiency, more secure.

In the [44] by Loi et al. also suggested a solution to secure the Ethereum smart contracts called OYENTE for detecting bugs in the code. OYENTE has the feature of symbolic execution to examine the bytecode of the Smart contracts and follow the implementation model of Ethereum EVM. The basic workflow of the OYENTE in which the input parameter is Smart contract bytecode and Ethereum global states. In that Input data must pass through the various steps from CFG Builder to the validator and result on the visualizer, it will help to carry out the debugging and program analysis.

In the Blockchain Smart contracts, Privacy leakage is also the area of concern and considered a severe threat. In [51] the author proposes a framework called HAWK, a new technique for developing privacy-preserving Smart contacts also removed the dependency of any code encryption or complication techniques. The basic idea behind the HAWK is a contract that is divided into two portions, private part, and the public part. The data considered to be private or of economic importance are kept in the private portion, and the rest of the other data is written in the public portion. The HAWK compilation of the code is done in three types. In the first, the code is compiled in the virtual machines just like the EVM. In the second category, services are limited to users of smart contracts. In the third category, the manager is an honest stakeholder in the HAWK system. The HAWK manager execution is done in the Intel SGX enclave (hardware-level security). It will monitor the privacy of the contract but will not reveal it. It also has a feature to maintain privacy between the different HAWK contracts. If its manager breaks the rules, then it will be routinely punished financially, and the user will get reimbursement.

In [52], For the safety of smart contracts, programmers who accept the ability to write contracts must be able to test their code in order to identify security flaws before uploading them on irreversible Blockchain environments. However, for smart contracts, there is only a range of security testing tools. Current research on automatic vulnerability scanning of smart contracts is not enough, and it is at an early stage. With the clear goal of incorporating Blockchain Smart Contracts more efficiently into protection and privacy, we must first consider their limitations before adopting them uniformly.

Smart contracts can hardly get rid of glitches as they are special computer program types. Despite worse, an exploitable security bug may result in serious outcomes, like cryptocurrency/money loss. In the proposed review, the author focuses on the most common types of security bugs in smart contracts. i.e. the re-entry bug, which caused a famous DAO attack with a loss of US$ 60 million. ReGuard, a fuzzing-based analyzer to dynamically diagnose reentrancy bugs in Ethereum smart contracts [53]. Another tool is to address the security issues of smart contracts, Securify, a security analysis framework for Ethereum based smart contracts that is scalable, fully automated, and capable of proving that contract activities are safe/unsafe in respect of a given property. It analyses the contract in two phases in the first; it analyzes the dependence graph of the contract symbolically to extract precise semantic details from the code. Afterward, check patterns of compliance and violation that capture quite enough conditions to prove or disprove whether the property holds or not. All patterns are explained in a given domain-specific language to enable extensibility better. In [56], the author has highlighted challenges related to one of the famous permission Blockchain platforms to develop and run smart contract applications. It has a basic feature called chain code to implement smart contracts by using common programming languages, such as Go, Node.js, and Java. All languages have some risk factors since security challenges may arise here. The author highlighted the risk associated with the Go language and identified 14 types of risk. The introduction of the Ethereum Network has resulted in many instances where the implementation of Ether Coins Smart Contracts has led to problems or disputes. The Smart Contract and Blockchain programming discipline, with structured best practices that can help solve the problems and disputes described above, is not yet adequately advanced compared to conventional software engineering. In the current scenario, Smart Contracts depend on a non-standard life-cycle of software, whereby, for example, applications distributed can hardly be upgraded or bugs fixed by launching a new version of the software [56].

4.3. Governance and Legal Challenges in Implementing Blockchain Smart Contracts

The idea of Smart contracts is closely related to Blockchain technology. It is not necessary to have such a linkage. However, the combination of both will provide an alternative solution to the traditional system managed by the government or the third party. Due to the evolution of cryptocurrency over Blockchain, Opens the opportunity for new technologies like a smart contract. It has legal obstacles, namely identification and authorization, privacy and data protection, fraud, errors, and errors in the (not always smart) code, false Oracles, consumer rights issues, competition law. If the customer or the user do not correctly understand the smart contract, later customer may claim that he is misleading or adequately informed about the terms and condition and validity of the contract is challenged. To improve its provision is needed to check the false promise and provision to backtrack from the contractor’s compensation for misleading. Another issue is that in the smart contracts, there is any bug or leakage it may occur because of the programmer, but from the false judgment or due to error, the well-known DOA attack on Ethereum Smart contracts is an example of it. This incident leads to rescind the Blockchain and break the concept of the distributed ledger. The inputs to the Smart contracts are fed from the real-world interfaces that depend locally like time, status delivery, interest rate, and if the interface gets tempered or wrong information is fed, then it will have an impact on the system. Besides the issue, the privacy issue will put limits on the smart contracts, like information of the participants are anonymized or pseudonymized, but the distributed ledger is public. Due to the public availability of Smart Contracts like legal and commercial terms may lead to taking attention to the competition authority. At last applicability of the criminal laws and sections will put a restriction on disclosure of specific information in public.

In [55], D.L Hoffman has proposed to have a lawful layer to support the Blockchain Smart contracts for the real-world application. It will face some initial challenges like implementation; the necessity to implement jurisdiction-specific legal ontologies also required some legal language to draft contract by including the distributed ledger technology. Nevertheless, it is not yet believed that smart contracts would replace the long-standing foundations of contract law. However, Smart contracts are software written code, and the authorized agency will require additional challenges in applying contract, whether a party has performed responsibility or the party has broken laws and other associated issues. The semantic legal layer for supporting Smart Contracts may be the solution for the above issues.

A. Savelyev [56] highlighted the issue of legal constraints in the Blockchain smart contracts and applications developed on it. It has open challenges to distribute the copyrighted work in the digital environment, and Blockchain smart contracts can be considered as the solution to such problems. But many questions to be answered before implementing it are

- where to hold copyrighted content and the related need to change the legal status of internet mediators.
- How to find the right balance between the static existence of Blockchain documents and the need to modify them because of the very essence of copyright law.
- Who grants rights based on a set of confidential details that are not publicly.

These questions have to be answered before the implementation of the application over it.

To explore the RQ5, we have done a detailed survey by considering the following points that will cover the overall development and structure of the Blockchain smart contracts.

5. THE BASIC ARCHITECTURE OF BLOCKCHAIN AND SMART CONTRACTS

A Blockchain is a form of distributed ledger which stores all completed transactions. In this, if the current block gets filled new block is added in the chain. Asymmetric cryptographic algorithms with the consensus mechanism are used to maintain the authenticity, privacy, and consistency of the data record. The Blockchain technology has some key features that gave an edge over the present centralized techniques like decentralization, persistence, anonymity, auditable, and provenance [6]. The main structure of the Blockchain has basic components like block and consensus algorithm. The basic structure is already discussed in the introduction part of the paper. In [57], the author studied software development activities, including the study of specifications, task assignment, testing and evaluation of Blockchain software projects. This is conducted through an online survey of active Blockchain software developers found from prominent Blockchain projects through the mining of Github repositories. With the feedback they received, they realized that Code Review and Unit Testing are the two most viable application development practices among Blockchain software developers. Finally, they concluded that most of the Blockchain applications chosen by the group discussion and project owners vary from the usual software project selection specifications.

5.1. Stages of Blockchain Development

Originally, as with Bitcoin, the platform was not programmable, but Blockchain systems have arisen that incorporate such features. The following four stages characterize the different types of Blockchain technologies and use (the first three are recognized, and the fourth is the developing level of AI-based Blockchain).

Blockchain 1.0 focuses on transactions, mainly for implementations in cash-based applications such as money transfer, financial transfers, and electronic payment systems. Blockchain 2.0 is an extension of Blockchain 1.0; it includes anonymity, smart contracts, and the beginning of tokens and functionality of non-native Blockchain properties. Blockchain 3.0 further extends the emphasis on Blockchain to incorporate decentralized applications.

A decentralized application is a backend technology operating on a shared peer-to-peer network that directly connects consumers and providers. This open-source software framework utilizes cryptographic tokens to operate on shared Blockchain. Such three Blockchain phases are not established simply by modifying or incorporating features. In addition to that, it influences the capabilities of the given Blockchain. The added features also enable new opportunities to be developed that would otherwise not be feasible and raise the long term value of using Blockchain as a whole.

Similarly, the latest and growing Blockchain iteration — Blockchain 4.0—offers major value opportunities. This involves the integration of artificial intelligence (AI) into Blockchain systems on two separate sides of the technology.

The AI is based on a predictive model that describes uncertainty. This is constantly evolving, and algorithms are supposed to guess or presume truth. Blockchain, on the other hand, uses a deterministic hashing algorithm that generates the same results when the inputs remain unchanged. The findings are permanent, and reality recording algorithms and cryptography are expected. While technologies are different, their common use helps solve complex problems [58].

As smart contracts were developed using a Blockchain platform. Those platforms provide simple architectures for developers to build smart contract applications. Many of this newly emerged Blockchain will support smart contracts.

In this part, we have evaluated seven mostly used Blockchain platforms for the smart contract implementation are Ethereum, [31], Neo[59], Hyperledger Fabric [60] Corda [61], Stellar [62], Rootstock and EOSIO [69].

5.1.1. Ethereum

Ethereum is a decentralized platform able to execute smart contracts. Like Bitcoin's Turing-incomplete script code, Ethereum also developed Turing-complete languages such as Solidity, Serpent, Low-level Lisp-like Language (LLL), and Mutan to enable general user applications besides cryptocurrency apps [31].

Ethereum uses Proof of Work as a consensus algorithm that is also code-intensive. Here, Ether is used to measure the cost of solving miners' puzzles. Basically, gas serves as an internal price for a transaction that lacks the unpredictable demand of Ether. Inclusively, the cost of a purchase can be measured using the gas cap multiplied by gas prices, since the gas cap shows the maximum amount of gas required to construct a block price and the gas price is the cost of a gas unit.

5.1.2. Neo

Neo project started in the year 2017, It is a platform that uses the Blockchain and digital identity to manage the digital assets to make itself managed and to achieve a smarter economy with the decentralized network. NEO uses a dedicated Byzantine Fault Tolerance (dBFT) algorithm that gives a tolerance of f = any (f−1)/3 fault to a consensus network composed of n nodes. This system includes two kinds of nodes: the normal node, and the consensus node. Normal nodes vote for consensus nodes according to their percentage of NEO. When a majority is to be reached, a speaker is selected randomly to agree on the proposal, and then other consensus nodes vote according to the dBFT algorithm. The languages which are used in Neo for smart contracts are C #, VB.Net, F #, Java, and Kotlin. For such languages, NEO gives compilers and plug-ins that are used to compile high-level languages into instruction sets backed by virtual NEO machines [59].

5.1.3. Hyper Ledger

It's also a decentralized network for smart contracts. Like Ethereum, which runs smart connections on virtual machines (e.g. EVM), Hyperledger uses Docker containers for code execution. Unlike virtual machines (VMs), containers that support smart contract applications with lower operating costs while losing isolation. The Fabric Blockchain-network permission (private or consortium) is allowed because Fabric is intended to support general business applications [60].

5.1.4. Corda

Corda is designed for digital-currency purposes, as compared to other Ethereum applications. To function as a global network for storing and distributing historical records of digital properties. Corda uses coding languages such as Java and Kotlin6, which run at the top of the Java Virtual Machine (JVM). Usually, Corda promotes private networks where businesses set up an authored network for private exchange of digital assets. Consensus can be easily achieved in private Blockchain networks. Corda adopts Raft as a consensus algorithm. It is easy to achieve a consensus in Raft by choosing a member, duplicating logs, and guaranteeing security [29].

5.1.5. Stellar

In contrast to Corda, stellar is a professional digital currency site. Stellar is faster and more accessible than Ethereum. In the meantime, Stellar will support language richness such as Python, JavaScript, Golang, and PHP. Stellar run application codes on top of Docker containers close to Fabric, thereby reducing the overhead. Steller is also a permission Blockchain platform, and the consensus is easily achieved. It has its own consensus algorithm known as the stellar consensus protocol [64,62].

5.1.6. Rootstock

Rootstock runs on the Bitcoin platform while enabling speedier transaction execution, i.e., it validates the transaction within 20 secs, and contracts are turing complete. Also, Rootstock has its virtual machine to execute contracts, and it is a public Blockchain platform, and data models are account-based. It uses the POW consensus algorithm and by adopting lightweight implementation reduces overhead. The main purpose of introducing Rootstock is to support digital currency [65].

5.1.7. EOSIO

The EROSION software presents a new Blockchain architecture that enables decentralized systems to be scaled vertically and horizontally. To be achieved by creating an operating system as a platform on which applications can be created. The platform offers identity, authentication, databases, distributed communication, and task scheduling through multiple CPU cores or clusters. The resulting infrastructure is a Blockchain system that can potentially scale up to thousands of transactions per second, avoid user fees, and empower decentralized applications to be distributed and managed quickly and easily, within a controlled network.

EOSIO software uses the Delegated Proof of Stake can fulfill the performance requirement of Blockchain applications. The basic concept of the DPOS algorithm, which holds the token in the Blockchain system, will appoint the block producer with the continued approval through the voting system. It supports the public as well as private Blockchain. It is supporting various types of applications finance, Education, Supply Chain other than digital currency [63] [65].

6. DISCUSSION

In this portion, we are going to discuss the result of the research questionaries' selected in the previous section, identified challenges, and proposed solutions for it. The result of this systematic mapping shows that major work is done in the application development of Blockchain smart contracts, and through the research, untouched domains are identified. After that, a major portion of the study is on the specific Blockchain architecture and mechanism and how to use it to implement specific services or tasks. Our main focus is to restrict our mapping to the Blockchain smart contracts and their implementation challenges. In that, a lot of work has been done in the application of the smart contracts in the different new domains like supply chain and digital asset management.

Parameter Platform	Execution Environment	Language for Coding	Turing Completeness	Data Model	Consensus Algorithm	Permission	Usage
Ethereum	EVM	Solidity, Serpent, LLL, Mutan	Turing complete	Account-based	PoW	Public	General
Hyper ledger Fabric	Docker	Java, Golang	Turing complete	Key-value pair	PBFT	Private	General
Corda	JVM	Java, Kotlin	Turing Incomplete	Key-value pair	Raft	Private	Digital Currency
Stellar	Docker	Python, JavaScript, Golang and PHP,	Turing Incomplete	Account-based	SCP	Consortium	Digital Currency
Rootstack	VM	Solidity	Turing complete	Account-based	PoW	Public	Digital Currency
EOSIO	Web	C++	Turing complete	Account-based	BFT-DPOS	Public	General
NEO	NeoVM	Java Script, C#	Turing complete	Account-based	DBFT	Public	Digital Assets

Table 3.Comparison of Smart Contract Platform

Table 3 shows the comparison of smart contract platform. In that, we come to know that a lot of work is done to develop applications like medical records and E-Voting, and it is also accepted in the banking industries. As development is going on in the future, we get a new version of the Blockchain smart contract application having AI capabilities. In the next step, we have done work to identify the security and privacy issue in the implementation of the Application in Blockchain smart contracts. Most of the work is done on the Ethereum and language used to develop Solidity, and most of the papers are on the security and privacy issues due to language pitfall. Some papers are on reward distribution like Gas in Ethereum, and few are on the security and privacy issues due to the public network or the untrusted network used to transfer the information between the blocks.

In a few papers, content related to the security and privacy issues occurred due to the consensus mechanism used in different Blockchain platforms, and most of the Blockchain platforms are emphasizing the account-based system. In that, there will be proper control over the whole network, and there are fewer chances of anonymous users in the network to perform malicious activity. In a few cases, cryptographic algorithms like electric curve cryptography used are proved to be not secure enough for the confidential information application.

Next in the category, we get papers related to the legal issues in implementing the block contacts that are related to copyright and the mapping of the smart contract with the real world contracts based on the local land laws.

6.1. Proposed Solutions & Open Challenges

In this subsection, we put light on the proposed solutions for the identified problems in implementing applications Over the Blockchain smart contracts and, in the last part, discussed some of the open challenges.

In [66] T.h. Lee suggested a solution for the termination of smart contracts as the conclusion of smart contracts is required for the protection and stability of any Blockchain system, especially those following Turing-complete smart contract languages. Resource-constrained Blockchain networks like Ethereum and Hyperledger Fabric might prevent proper closure of smart contracts when the pre-allocated resources are not adequate. Although smart contract execution is usually based on the current status of Blockchain and application inputs, this strategy is not always effective. In their work, they suggested a lazy technique by statistically confirming a smart contract's conditional closure and non-closure to determine input conditions in which the contract finishes or not. Once a smart contract is concluded, the evidence-based Blockchain network must check whether the current state and the validity of the contract satisfy the terms of termination in order to decide if the contract is qualified (i.e. eventually suspending) to function on the chain.

Security issues due to the bugs as smart contracts can hardly get rid of glitches as they are special computer program types. In [53] C.Liu proposes ReGuard, a fuzzing-based analyzer to diagnose reentrancy bugs in Ethereum smart contracts dynamically. As an exploitable security bug may result in serious outcomes, like cryptocurrency/money loss. It is initiated for the Ethereum web services, and such initiatives are required for the other platforms and services. ReGuard performs trace analysis of bugs through runtime. They instantiated ReGuard as a web service for Ethereum contracts.

Another security issue due to the bugs in the smart contract is due to integer bugs, which is particularly difficult to avoid in the case of Ethereum Virtual machine and especially Solidity programing language. To overcome such a problem, C. Torres [67] introduced a framework OSIRIS that integrates symbolic execution with taint analysis to correctly and incorporate vulnerabilities throughout smart contracts from Ethereum. They have tested it on the 1,2 million smart contracts and found approximately 42 thousand out of which have integer bug problems.

In the case of Ethereum, there is another problem of re-entry in which the adversary has the power to exploit and frequently call the intermediate state of the caller contract. To address problem Y. Harari[68] introduces a language called Bamboo, and its syntax is very much similar to Erlang and mainly for the polymorphic contracts of the Ethereum.

After Ethereum Hyper ledger fabric in another platform that is popularly used to develop the smart contract in a permission Blockchain environment. One of the benefits of it is that common programming languages are used to develop smart contracts. But a major drawback associated with the languages is that they are not intended for the development of smart contracts. K.Yamashita[54], in their work, proposed a tool considering the risk in the Go language and finding out risk using the static analysis. However, it is only related to the Go language, and it is required to be updated as new changes are incorporated in the language or the applications developed using it.

To address the problems of security in the smart contract, In [69] t.sankov presents an Ethereum Smart Contract Security Analyzer, which is scalable, completely automated, and competent in providing contractual behaviors as secure/insecure in respect of the given property. Analysis of the Security consists of two steps in step one it symbolically analyzes the connection graph of the contract to derive correct semantic information from the written code. In the next step, it tests cycles of conformity and infringement that catch ample conditions to prove whether or not a property holds. Besides, to achieve extendibility, all patterns are specified in the domain-specific language. It has certain benefits like being able to analyze every aspect of the contract behavior to avoid a false negative, able to categorize warnings and support new domain languages. But, it is only restricted to the Ethereum based smart contracts and not for other platforms.

Double spending is another security concern in the cryptography-based digital currency. In [70] w.wirachantika has selected Fawkescoin and found security and integrity issues in it. To avoid double spending in it, they have suggested the use of DSA on the Merkle tree for data verification without knowing the data contents.

Criminal Smart contract is another very fiery issue in the security concern of the smart contracts. A. Jules in [49] has raised the issues related to it and presented three types are secret leakage, key theft, and calling cards crime. Above all, issues related to the leakages are possible through the present cryptography setup.

In the existing system, privacy is also concerned as parties involved in the transaction are pseudonyms and there are chances that information gets leaked on the platform. To overcome this issue, A.Kosba in [51] presented a mechanism called Hawk, which is mainly for financial transactions that cannot store the transaction information and hide it from public access. They implemented this protocol through the formal model, and it is on top of a decentralized platform.

6.2. Open Challenges

6.2.1. Mixed Model

The requirement of Public Blockchain smart contract is having the feature of private Blockchain. D.Huang[71] is proposing an attribute-based encryption protection solution built on the private Blockchain. In that, they have suggested it for the IoT based and banking application. But still, it is an open challenge when it comes to the different types of other applications and requires some concrete solution for it.

6.2.2. Compatibility

In the adoption of Blockchain in E-Government applications, our findings show that very little academic work has been done in this area. More intensive work in this area is required to advance the level of sophistication in this field. In addition to the technical aspects, it also requires a new governance model to adopt it, and new reference architectures are required.

6.2.3. Suitable Consensus Mechanism

Selecting Consensus algorithm in implementing smart contracts on the public Blockchain architecture. The proof of work algorithm is the first algorithm used in bitcoin to achieve distributed consensus in a large scale untrusted environment. A low-cost trustworthy algorithm is required that can fill the requirement to implement smart contracts on all possible forms of the Blockchain environment public, private, consortium.

6.2.4. Testing Framework

Once smart contracts are implemented on the Blockchain platform, it is not going to be reverted, and this may result in a loss of money and confidential information. So, before deploying it, a framework of the analyzer is required that can check the smart contracts in all the aspects, and it helps in removing bugs and security threats in it. In our findings, most of the academic research papers are based on the Ethereum and the bugs in the programming languages used to develop it. Very less work is done on the other platforms with other dimensions of the security and privacy threats that may occur due to the architecture and the network.

6.2.5. Standards to Develop Applications

In recent years after the success of the bitcoin, Blockchain rapidly grows, and it is going to be used in all types of applications, from financial to education. In that Blockchain version, two smart contracts play an important role in Blockchain and smart contract development. The same software engineering practices are followed that can be used for regular software development. But regular software engineering will not complete the requirement. It is also an open challenge area where a lot of work is required to standardize it. In our findings, we found only one paper on it.

A lot of work has been done to identify the bugs in the various smart contracts platforms and their applications. In our findings, we come to know about various frameworks and tools that are detecting bugs and security glitches in the smart contracts before deployment. Such tools are not available in all the languages, platforms presently used for smart contracts. With this, another issue to be addressed in the future is common tools and frameworks for all.

6.2.6. Security & Privacy Issues

Improvements in the safety and performance of smart contracts are required to tackle practical and competitive decentralized applications. Most proposed applications needed the on-chain and off-chain combination. Recognized two main approaches to improving the efficiency of applications based on distributed ledger technology, using a lighter consensus mechanism and performing concurrent transactions. Nonetheless, Blockchain-based solution efficiency only compares to other Blockchain solutions, and yet there is a wide gap between Blockchain-based solution performance and established implementations. Research to boost the efficiency of smart contract implementation and the overall Blockchain-based applications is in its early stages.

6.2.7. Scalability

With a growing amount of Blockchain use, and an increase in the huge number of regular transactions, the scale of the Blockchain is increasingly rising. All transactions are preserved to get validated in each node. The origins of the current transaction must first be checked before the validity of the transaction. The block size restricted and the time duration used to produce a new Block, this plays a part in failing to meet the demand that millions of transactions be performed concurrently in real-time situations. The size of transactions meanwhile, as miners would prefer to validate transactions with higher transactional fees.

7. CONCLUSION

In this systematic survey, we have analyzed Blockchain, and smart contract platforms and applications developed and currently using it. In our findings, a lot of research work has been found in academic research related to security, privacy, and governing issues. Also, remedial measures are suggested by many authors/researchers, but still, few domains related to security are unrevealed. The promenading issues are that there is no standardization of pre simulation tools of the smart contract before deploying. Also, for identifying bugs in the programming language used to develop smart contracts is available for very few platforms and programming languages.

Moreover, there is no proper coordination or standard communication between different smart contract platforms, as different programming languages are used. In contrast to the traditional consensus mechanism used in traditional, Blockchain is not very much compatible with the smart contract applications, but it is in its developing stage. As per this paper investigation, this is predicted that Artificial Intelligence will be used in the future for various applications using the smart contract. A lot of platforms are available to develop the smart contracts applications out of which Ethereum and Hyper-Ledger are mostly used, but still, some more customized platforms having features of both (public and private) are required.

REFERENCES

[1] T. ODEJOBI, "CSC626-LectureNote," 13 April 2012. [Online]. Available: ifecisrg.org/sites/default/files/csc626-LectureNote.pdf. [Accessed Oct,12, 2018].

[2] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," NIST Interagency/Internal Rep., p. 57, 2018, doi: 10.6028/NIST.IR.8202.

[3] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on Blockchain technology? - A systematic review," PLoS One, vol. 11, no. 10, pp. 1–27, 2016, doi: 10.1371/journal.pone.0163477.

[4] "Global market for blockchain technology 2018-2023 | Statista." https://www.statista.com/statistics/647231/worldwide-blockchain-technology-market-size/ (accessed Oct. 15, 2018).

[5] N. . Satoshi, "Bitcoin: A Peer-to-Peer Electronic Cash System," , . [Online]. Available: http://bitcoin.org/bitcoin.pdf. [accessed 28 12 2018].

[6] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, pp. 557–564, 2017, doi: 10.1109/BigDataCongress.2017.85.

[7] M. Alharby and A. Van Moorsel, "BlockSim: A simulation framework for blockchain systems," Perform. Eval. Rev., vol. 46, no. 3, pp. 135–138, 2019, doi: 10.1145/3308897.3308956.

[8] A. Bahga and V. K. Madisetti, "Blockchain Platform for Industrial Internet of Things," J. Softw. Eng. Appl., vol. 09, no. 10, pp. 533–546, 2016, doi: 10.4236/jsea.2016.910036.

[9] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea, and B. Hamida, "Consortium Blockchains: Overview, Applications and Challenges," Int. J. Adv. Telecommun., vol. 11, no. 1&2, pp. 51–64, 2018.

[10] B. K. Mohanta and D. Jena, "An Overview of Smart Contract and Use Cases in Blockchain Technology: 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)," 2018 9th Int. Conf. Comput. Commun. Netw. Technol., pp. 1–4, 2018, doi: 10.1109/ICCCNT.2018.8494045.

[11] M. H. Miraz and M. Ali, "Applications of Blockchain Technology beyond Cryptocurrency,," 2018. accessed: Oct-20-2019. [Online]. Available: www.aetic.theiaer.org.

[12] N. Szabo, "Formalizing and securing relationships on public networks," First Monday, 1997. https://firstmonday.org/ojs/index.php/fm/article/view/548 (accessed Dec. 27, 2018).

[13] E. Regnath and S. Steinhorst, "SmaCoNat: Smart Contracts in Natural Language," Forum Specif. Des. Lang., vol. 2018-Septe, no. September, 2018, doi: 10.1109/FDL.2018.8524068.

[14] S. Tikhomirov, "s-tikhomirov/smart-contract-languages", GitHub, 2018. [Online]. Available: https://github.com/s-tikhomirov/smart-contract-languages. [Accessed: 10- Dec- 2018].

[15] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," Proceeding 2017 11th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/TSSA.2017.8272896.

[16] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," Proceeding 2017 11th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/TSSA.2017.8272896.

[17] J. Moubarak, E. Filiol, and M. Chamoun, "On blockchain security and relevant attacks," in IEEE Middle East and North Africa Communications Conference, MENACOMM, 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/MENACOMM.2018.8371010

[18] R. M. Parizi, Amritraj, and A. Dehghantanha, "Smart contract programming languages on blockchains: An empirical evaluation of usability and security," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10974 LNCS, no. June, pp. 75–91, 2018, doi: 10.1007/978-3-319-94478-4_6.

[19] G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, and R. Hierons, "Smart contracts vulnerabilities: A call for blockchain software engineering?," 2018 IEEE 1st Int. Work. Blockchain Oriented Softw. Eng. IWBOSE 2018 - Proc., vol. 2018-Janua, pp. 19–25, 2018, doi: 10.1109/IWBOSE.2018.8327567.

[20] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SoK)," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10204 LNCS, no. July, pp. 164–186, 2017, doi: 10.1007/978-3-662-54455-6_8.

[21] A. Murali, "Most government blockchain projects in India are stuck at the starting blocks," 2018. https://factordaily.com/india-government-blockchain-projects/ (accessed Dec. 17, 2018).

[22] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007. https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering (accessed Apr. 20, 2019).

[23] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, "A survey of blockchain applications in different domains," ACM Int. Conf. Proceeding Ser., pp. 17–21, 2018, doi: 10.1145/3301403.3301407.

[24] F. R. Batubara, J. Ubacht, and M. Janssen, "Challenges of blockchain technology adoption for e-government," Proc. 19th Annu. Int. Conf. Digit. Gov. Res. Gov. Data Age - dgo '18, pp. 1–9, 2018, doi: 10.1145/3209281.3209317.

[25] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," IEEE Access, vol. 7, no. c, pp. 50759–50779, 2019, doi: 10.1109/ACCESS.2019.2911031.

[26] J. Al-Jaroodi and N. Mohamed, "Blockchain in Industries: A Survey," IEEE Access, vol. 7, no. c, pp. 36500–36515, 2019, doi: 10.1109/ACCESS.2019.2903554.

[27] A. Prashanth Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," Math. Found. Comput., vol. 1, no. 2, pp. 121–147, 2018, doi: 10.3934/mfc.2018007.

[28] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, and J. Zhang, "A Blockchain-Based Supply Chain Quality Management Framework," Proc. - 14th IEEE Int. Conf. E-bus. Eng. ICEBE 2017 - Incl. 13th Work. Serv. Appl. Integr. Collab. SOAIC 207, pp. 172–176, 2017, doi: 10.1109/ICEBE.2017.34.

[29] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10322 LNCS, pp. 357–375, 2017, doi: 10.1007/978-3-319-70972-7_20.

[30] A. Hughes, A. Park, J. Kietzmann, and C. Archer-Brown, "Beyond Bitcoin: What blockchain and distributed ledger technologies mean for

firms,” Bus. Horiz., vol. 62, no. 3, pp. 273–281, 2019, doi: 10.1016/j.bushor.2019.01.002.

[31] T. M. Fernandez-Carames and P. Fraga-Lamas, “A Review on the Application of Blockchain to the Next Generation of Cybersecure Industry 4.0 Smart Factories,” IEEE Access, vol. 7, no. c, pp. 45201–45218, 2019, doi: 10.1109/ACCESS.2019.2908780.

[32] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, “Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends,” IEEE Trans. Syst. Man, Cybern. Syst., vol. 49, no. 11, pp. 2266–2277, 2019, doi: 10.1109/TSMC.2019.2895123.

[33] R. C. De Souza, E. M. Luciano, and G. C. Wiedenhöft, “The uses of the Blockchain Smart Contracts reduce the levels of corruption: Some preliminary thoughts,” ACM Int. Conf. Proceeding Ser., pp. 5–6, 2018, doi: 10.1145/3209281.3209408.

[34] C. Udokwu, A. Kormiltsyn, K. Thangalimodzi, and A. Norta, “An Exploration of Blockchain enabled Smart-Contracts Application in the Enterprise,” no. June, pp. 1–28, 2018, doi: 10.13140/RG.2.2.36464.97287.

[35] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues,” Telemat. Informatics, vol. 36, pp. 55–81, 2019, doi: 10.1016/j.tele.2018.11.006.

[36] A. Ramachandran and D. M. Kantarcioglu, “Using Blockchain and smart contracts for secure data provenance management,” 2017, [Online]. Available: http://arxiv.org/abs/1709.10000.

[37] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaria, “To Blockchain or Not to Blockchain: That Is the Question,” IT Prof., vol. 20, no. 2, pp. 62–74, 2018, doi: 10.1109/MITP.2018.021921652.

[38] G. Greenspan, “The Blockchain Immutability Myth - CoinDesk.” https://www.coindesk.com/blockchain-immutability-myth/ (accessed Dec. 16, 2018).

[39] V. Lemieux, “Blockchain for Recordkeeping: Help or Hype?,” 2016. https://www.researchgate.net/publication/309414363_Blockchain_for_Recordkeeping_Help_or_Hype (accessed Dec. 27, 2019).

[40] R. Henry, A. Herzberg, and A. Kate, “Blockchain access privacy: Challenges and directions,” IEEE Secur. Priv., vol. 16, no. 4, pp. 38–45, 2018, doi: 10.1109/MSP.2018.3111245.

[41] Y. Hu, M. Liyanage, A. Mansoor, K. Thilakarathna, G. Jourjon, and A. Seneviratne, “Blockchain-based Smart Contracts - Applications and Challenges,” vol. 1, no. 1, pp. 1–12, 2018, [Online]. Available: http://arxiv.org/abs/1810.04699.

[42] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” J. Netw. Comput. Appl., vol. 126, pp. 45–58, 2019, doi: 10.1016/j.jnca.2018.10.020.

[43] J. Buck, “Blockchain Oracles, Explained | Cointelegraph,” Cointelegraph, 2017. https://cointelegraph.com/explained/blockchain-oracles-explained (accessed Dec. 15, 2019).

[44] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” in Proceedings of the ACM Conference on Computer and Communications Security, 2016, vol. 24-28-October-2016, pp. 254–269, doi: 10.1145/2976749.2978309..

[45] A. Juels, A. Kosba, and E. Shi, “The ring of gyges: Investigating the future of criminal smart contracts,” Proc. ACM Conf. Comput. Commun. Secur., vol. 24-28-Octo, pp. 283–295, 2016, doi: 10.1145/2976749.2978362.

[46] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” Futur. Gener. Comput. Syst., vol. 107, no. Xiaoqi Li, pp. 841–853, 2020, doi: 10.1016/j.future.2017.08.020.

[47] H. Mayer, “ECDSA Security in Bitcoin and Ethereum: a Research Survey,”2016.https://pdfs.semanticscholar.org/1785/6bad4335c8ca7419aab2c715ea25ce5e0621.pdf(accessed Dec, 18,2018)

[48] T. Chen, X. Li, X. Luo, and X. Zhang, “Under-optimized smart contracts devour your money,” SANER 2017 - 24th IEEE Int. Conf. Softw. Anal. Evol. Reengineering, pp. 442–446, 2017, doi: 10.1109/SANER.2017.7884650.

[49] M. Apostolaki, A. Zohar, and L. Vanbever, “Hijacking Bitcoin: Routing Attacks on Cryptocurrencies,” Proc. - IEEE Symp. Secur. Priv., pp. 375–392, 2017, doi: 10.1109/SP.2017.29.

[50] L. Luu et al., “SmartPool : Practical Decentralized Pooled Mining This paper is included in the Proceedings of the,” Proc. 26Th Usenix Secur. Symp. (Usenix Secur. ’17), pp. 1409–1426, 2017.

[51] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts,” Proc. - 2016 IEEE Symp. Secur. Privacy, SP 2016, pp. 839–858, 2016, doi: 10.1109/SP.2016.55.

[52] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, and K. K. R. Choo, “A systematic literature review of blockchain cyber security,” Digit. Commun. Networks, 2019, doi: 10.1016/j.dcan.2019.01.005.

[53] C. Liu, H. Liu, Z. Cao, Z. Chen, B. Chen, and B. Roscoe, “ReGuard: Finding reentrancy bugs in smart contracts,” Proc. - Int. Conf. Softw. Eng., pp. 65–68, 2018, doi: 10.1145/3183440.3183495.

[54] K. Yamashita, Y. Nomura, E. Zhou, B. Pi, and S. Jun, “Potential Risks of Hyperledger Fabric Smart Contracts,” IWBOSE 2019 - 2019 IEEE 2nd Int. Work. Blockchain Oriented Softw. Eng., pp. 1–10, 2019, doi: 10.1109/IWBOSE.2019.8666486.

[55] D. L. Hofman, “Legally speaking: Smart contracts, archival bonds, and linked data in the blockchain,” 2017 26th Int. Conf. Comput. Commun. Networks, ICCCN 2017, pp. 0–3, 2017, doi: 10.1109/ICCCN.2017.8038515.

[56] A. Savelyev, “Copyright in the blockchain era: Promises and challenges,” Comput. Law Secur. Rev., vol. 34, no. 3, pp. 550–561, 2018, doi: 10.1016/j.clsr.2017.11.008.

[57] P. Chakraborty, R. Shahriyar, A. Iqbal, and A. Bosu, “Understanding the software development practices of blockchain projects: A survey,” Int. Symp. Empir. Softw. Eng. Meas., 2018, doi: 10.1145/3239235.3240298.

[58] J. Angelis and E. Ribeiro da Silva, “Blockchain adoption: A value driver perspective,” Bus. Horiz., vol. 62, no. 3, pp. 307–314, 2019, doi: 10.1016/j.bushor.2018.12.001.

[59] NEO, “NEO Smart Economy,” Www.Neo.Org, 2014. https://neo.org/ (accessed Dec. 19, 2019).

[60] E. Androulaki et al., “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” Proc. 13th EuroSys Conf. EuroSys 2018, vol. 2018-January, pp. 1–15, 2018, doi: 10.1145/3190508.3190538.

[61] R. G. Brown, “The Corda Platform: An Introduction,” 2018. Accessed: 16-Nov-2019. [Online]. Available: https://www.corda.net/wp-content/uploads/2018/05/corda-platform-whitepaper.pdf.

[62] D. Mazi`eres “Stellar Consensus Protocol - Stellar.” https://www.stellar.org/papers/stellar-consensus-protocol (accessed Dec. 27, 2019).

[63] EOS.IO, “Documentation/TechnicalWhitePaper.md at master · EOSIO/Documentation · GitHub,” Github.com, 2018. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md (accessed Nov 27, 2019).

[64] T. Bocek and B. Stiller, “Smart contracts - Blockchains in thewings,” in Digital Marketplaces Unleashed, Springer Berlin Heidelberg, 2017, pp. 169–184.

[65] Z. Zheng et al., “An overview on smart contracts: Challenges, advances and platforms,” Futur. Gener. Comput. Syst., vol. 105, pp. 475–491, 2020, doi: 10.1016/j.future.2019.12.019.

[66] T. C. Le, L. Xu, L. Chen, and W. Shi, “Proving conditional termination for smart contracts,” in BCC 2018 - Proceedings of the 2nd ACM Workshop on Blockchains, Cryptocurrencies, and Contracts, Co-located with ASIA CCS 2018, 2018, pp. 57–59, doi: 10.1145/3205230.3205239.

[67] C. F. Torres, J. Schütte, and R. State, “Osiris: Hunting for integer bugs in ethereum smart contracts,” in ACM International Conference Proceeding Series, 2018, pp. 664–676, doi: 10.1145/3274694.3274737.

[68] Y. Hirai, “Bamboo: a language for morphing smart contracts„” 2018, https://github.com/pirapira/bamboo, (accessed June 30, 2018).

[69] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," Proc. ACM Conf. Comput. Commun. Secur., no. June, pp. 67–82, 2018, doi: 10.1145/3243734.3243780.

[70] W. Wirachantika, A. M. Barmawi, and B. A. Wahyudi, "Strengthening fawkescoin against double spending attack using Merkle tree," ACM Int. Conf. Proceeding Ser., pp. 49–54, 2019, doi: 10.1145/3309074.3309105.

[71] D. Huang, C. J. Chung, Q. Dong, J. Luo, and M. Kang, "Building private blockchains over public blockchains (POP): An attribute-based access control approach," in Proceedings of the ACM Symposium on Applied Computing, 2019, vol. Part F147772, pp. 355–363, doi: 10.1145/3297280.3297317.

Authors

Jasvant Mandloi is research scholar at Information Technology Department of Institute of Engineering &Technology (IET), a UTD of Devi Ahilya University, Indore, Madhya Pradesh, India and the author of 6 refereed publications. He was Head of the department at one of the prestigious Engineering college of Central India. He was secretory of CSI Indore chapter for year 2016-17. He served as National Co-Coordinator for Vidyarthi Vigyan Mantahn (A Project for School Level Students by Vigyan Bharati, an NGO working for National Science Movement). He has done his graduation in Information Technology from JIT Borawan Madhya Pradesh in year 2006. He has received Master's degree Mtech IT from UTD RGPV Bhopal in year 2008.He has membership of technical association ISTE and CSE and received awards at National level from IBM and NEN India. He has recently cleared the UPSC interview and get selected as T&P officer in the Daman Technical Department. His areas of research interest are Blockchain, Smart Contract, Cloud computing, Cyber security and Machine learning.

Dr. Pratosh Bansal is Professor in Information Technology Department of Institute of Engineering & Technology (IET), a UTD of Devi Ahilya University, Indore, Madhya Pradesh, India and the author of 33 refereed publications. He was Director, IQAC of the University, Professor In-charge of CSI Student Chapter (IET) and Professor In-charge of IET Incubation Centre. He was Vice-President of CSI Indore Chapter. He served as National Coordinator for Vidyarthi Vigyan Mantahn (A Project for School Level Students by Vigyan Bharati, an NGO working for National Science Movement). He has done his graduation in Mechanical Engineering from Govt. Engineering College, Jabalpur, Madhya Pradesh, India, in 1995. He received his M.Tech. (Energy Management) in 1999, M. Tech. (Computer Science) in 2003 and PhD in 2011 in Computer Engineering from IET, DAVV, Indore, MP, India. His areas of research interest include enterprise resource planning, knowledge management, e-commerce, digital forensics, cloud computing, green IT and energy management.