Cluster Computing (2023) 26:395–421
https://doi.org/10.1007/s10586-021-03491-1

A review of smart contract-based platforms, applications, and challenges

Pratima Sharma[1] · Rajni Jindal[1] · Malaya Dutta Borah[2]

Received: 27 August 2021 / Revised: 17 November 2021 / Accepted: 24 November 2021 / Published online: 15 January 2022

Abstract
Blockchain is a modern technology that has gained enormous attention in scientific and practical applications. A smart contract is a digital transaction that runs, executes, and records the dynamic operation on the ledger automatically. A smart contract is the central aspect of a blockchain that facilitates blockchain as a platform outside the cryptocurrency spectrum. It applies to many applications such as education, voting, real estate, entertainment, IoT, supply chain, healthcare, and much more. While recent years have seen remarkable progress in developing blockchain technologies, emphasizing smart contracts, there is a lack of study of the smart contract concept. This paper extensively examines the core principles and guides recent research and advances in smart contracts. The study analyses are summarized in three key categories: (i) smart contract-based platforms and decentralized applications, (ii) risk problem identification, and (iii) potential solutions and future directions.

Keywords Blockchain · Smart contract · Distributed applications · Platforms

1 Introduction

Traditionally, through intermediaries, we have developed trust within our society. We often use third-party entities because we believe that they will store and protect our goods and send the correct amount to the right person when we require it. By diverting trust to decentralized systems, blockchain substitutes the need for intermediaries. Blockchain technology seeks to resolve this by allowing non-trusting participants without the intervention of a trustworthy third party to reach a consensus on their transactions and communications. It is possible to understand blockchain as a distributed ledger that preserves the record of all transactions that have ever taken place in the blockchain network. The foundational advancement behind the first decentralized e-payment system, Bitcoin, is blockchain. Beyond financial apps, blockchain has grown to support a range of decentralized applications. Many of these implementations depend on smart contracts being executed on top of the blockchain. A smart contract is a computer program that encodes a non-trusting party arrangement and is implemented based on some pre-defined rules [1]. As part of a network exchange, a smart contract is implemented or performed on blockchain networks. It is the miners' duty, specific forms of blockchain network users, to deploy new contracts and implement existing ones. Based on the computing costs needed to execute the contracts, miners get paid for this work. This study aims to describe and recognize the peer-reviewed research on smart contract technology that has been published. This work conducts a comprehensive analysis of recent developments in smart contracts. We plan to analyze applications related to blockchain-based smart contracts, find general challenges, and provide a possible way to solve them. Based on the above discussion, the following are the significant contributions of this paper.

- An overview of blockchain technology and smart contracts is briefly highlighted and analyzed.
- Existing blockchain and smart contract-based studies are explored, and their advantages, disadvantages are explained.

✉ Pratima Sharma
pratima.sharma1491@gmail.com

[1] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India

[2] Department of Computer Science and Engineering, National Institute of Technology Silchar, Assam, India

- Various smart contract decentralized platforms are analyzed based on multiple parameters, and basic building blocks are summarized.
- Present a systematic and comprehensive review of recent applications of smart contracts.
- Identify the challenges associated with smart contracts based on blockchain and find potential ways to resolve the identified problems.

Furthermore, we agreed to adopt the systematic research method to search for related papers in the significant science databases and create a classification map. The plan designed helps to understand the topics of interest better and to find holes for future work.

This paper's framework is as follows. The approach used to perform the systematic mapping analysis, including the description of the research issues, is presented in Sect. 2. Context details on blockchain and smart contract technology are given in Sect. 3. The comparative analysis and description of related papers/studies are illustrated in Sect. 4. We address and answer the study questions in Sects. 5, 6, and 7. The article ends in Sect. 8.

2 Research methodology

This section presents scientific and systematic literature to give a transparent and reproducible review of applications based on the smart contract. The systematic review is applied to explore the related issues to smart contacts within the blockchain technique. The review process involved pinpointing and exploring the research papers to answer the research questions based on smart contracts' current issues. The whole review process can be divided into various steps, shown below in Fig. 1, which illustrates all phases of our systematic review.

The presented methodological approach consists of the following essential steps:-

1. *Planning the review* A review protocol has been developed to frame the research questions answered by the systematic review. The research questions are the objectives that we keep in mind to address the systematic review's demands. Development of review protocol encompasses designing an appropriate strategy to search relevant studies, formulating criteria for inclusion–exclusion, deciding data extraction, and synthesis methods.
2. *Conducting the review* Identify various studies from the scientific journals by accessing their quality in terms of inclusion–exclusion criteria, data extraction, and synthesizing the data. Relevant search criteria help to identify appropriate candidate articles to cater to the need for review. Based on the suggested criteria, individual candidate studies are considered relevant or not relevant for the review.
3. *Reporting the results* The review results are summarized by mapping the research questions with the selected articles and categorizing them theme-wise. The developed review protocol results have been reported by identifying the smart contract's platforms, applications, challenges, and solutions from the selected papers. We address and answer the research questions in 5, 6, and 7 Sections.

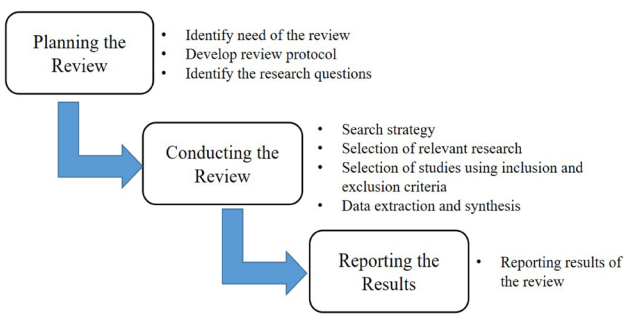

Fig. 1 Systematic review process

The first phase of the systematic review process focused on identifying the current research issues on blockchain-based smart contracts. As we found, smart contracts are one of the applications of blockchain adopted for the exchange of assets, shares, and monetary values without any middleman and conflict. But perhaps some issues are related to smart contracts, which need attention for making ready for use in the business world. Some critical issues we have identified from the research papers and articles are framed in research questions. Table 1 represents the research issues and motivation towards the exploration of research done to resolve issues.

2.1 Selection and analysis of relevant research work

To conduct our systematic review, various online journal databases are used, including Scopus, Science Direct, Web of Science, ACM Digital Library, SpringerLink, and Google Scholar. The term "smart contract," "smart contract applications," "smart contract tools," "smart contract platforms," "smart contract challenges," and "blockchain" are searched in all document titles. Additional searches based on including "issues/challenges related to the smart contract," "advances in the smart contract," "smart contract domains," and "applications of blockchain." The study analyses approx. 183 papers, which are mainly published between 2015 and May 2020. The exploration aims to find the answer to research issues framed in our research questions. Figure 2 describes the screening

Table 1 Research issues of smart contracts

RQ#	Research question	Motivation
RQ1	How can available platforms support smart contract development?	Investigating platforms used for smart contract development
RQ2	What are the presented decentralized applications, which mainly discuss smart contracts design and implementation?	Investigation of a diversity of applications of smart contracts already been used
RQ3	What are the challenges that relate to the applications of smart contracts?	Identifying the challenges and other hindrances related to applications of smart contracts
RQ4	What are the solutions to the identified challenges?	Outline the possible solutions to the identified challenges

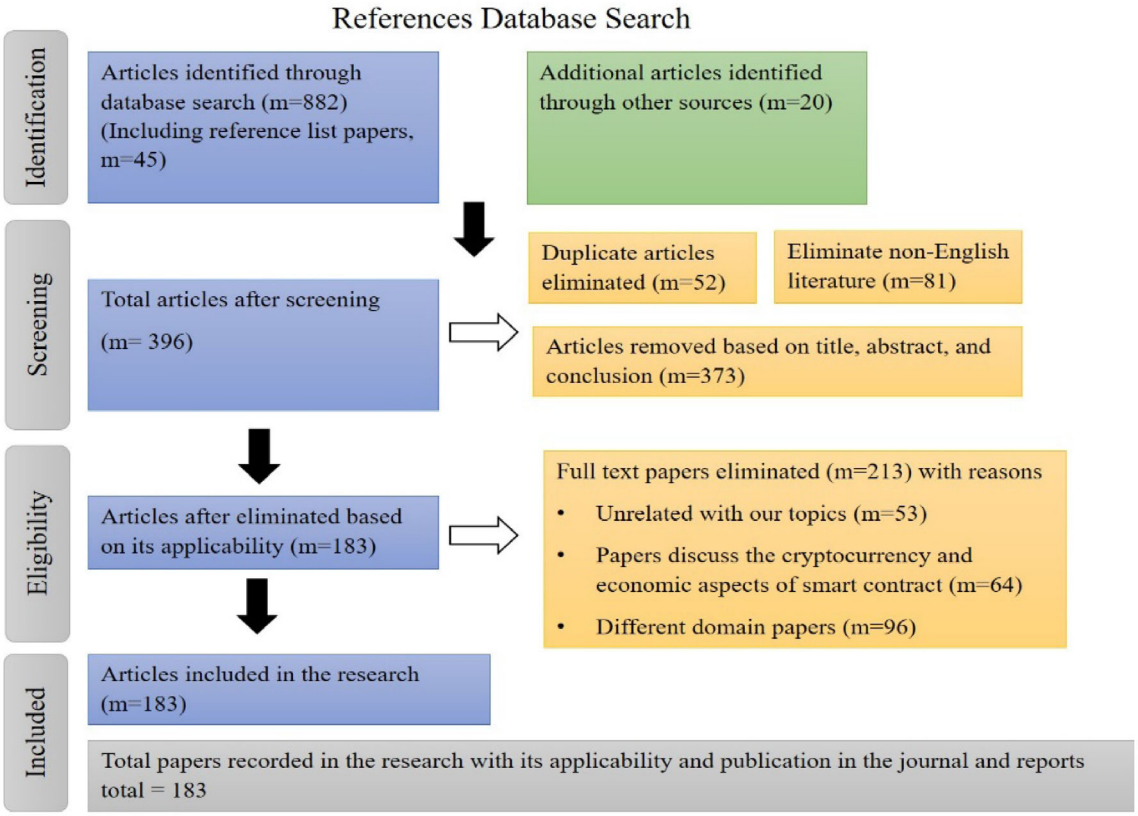

Fig. 2 Selection of relevant documents

process of extracted documents. The presented reference database search and selection approach consists of the following essential steps:

1. *Identification* This step is considered a keyword-based search strategy to find out the relevant studies. After forming search strings, the most essential and appropriate documents were selected for the proposed study. In total, 902 studies were extracted after exhaustively searching the mentioned databases and going through the reference sections of the studies extracted from the electronic libraries.
2. *Screening* After obtaining sufficient literature, refinement of the documents focused on the inclusion–exclusion criteria. The screening process firstly eliminated the duplicate and non-English articles. Then, extracted documents are analyzed based on title, abstract, and conclusion and removed that lie outside the proposed study's scope. In total, 506 articles were eliminated in the screening process.
3. *Eligibility* In this step, the full text of the extracted documents was considered to check the applicability based on the study's proposed theme. The documents that are relevant and could answer research questions are included. The documents focused on technical

details, not having abstracts, and relevance to our research objective are excluded. For instance, papers that discuss the ethical issues of the blockchain and smart contracts were eliminated.

4. *Inclusion* The proposed study considered approx. 183 articles, including journal/conference papers, book chapters, and excluding websites. The online website references are explicitly searched on the databases after getting details from the selected papers.

Table 2 depicts the analysis of the selected article based on publication sources with their respective numbers.

2.2 Review result

This section presents the systematic review process results by categorizing the selected studies with the identified questions. Table 3 presents the mapping of identified research questions with the selected studies. Table 4 depicts the distribution of academic papers category-wise selected for the review process. The detailed explanation and comparative analysis results of research questions are presented in 5, 6, and 7 Sections.

Figure 3 represents publication density year-wise for the smart contracts. After 2015, the research in smart contracts is tremendously increased, and it is maintained to date. Most of the papers are review papers and show the applications of smart contracts. Some documents also raise the issues and hurdles coming in the implementation of various applications in different domains.

3 Background studies

3.1 Blockchain technology

The developments in cryptography and distributed computing have introduced a modern computer technology called blockchain in the past decade. Blockchain is a distributed ledger that replicates and exchanges data through peer-to-peer networks. Blockchain was initially introduced by an unknown person, Satoshi Nakamoto, who created bitcoin to trade digital currencies without third parties [152] directly. Nakamoto developed the paradigm of a network of nodes working to maintain a decentralized and secure database. The blockchain platform is the methodology behind cryptocurrencies—a shared public database or a continuously updated registry of all transactions [181]. Blockchain, as the title suggests, is an ordered list of blocks. By referencing the previous block's hash, each block distinguishes by the hash sequence and ties to the preceding block. The only anomaly is the first block (called "Genesis block"), which does not have the previous block's hash value, known as the ancestor block. Blockchain can be regarded as both a technical breakthrough and financial advancement [12]. It provides a solution to any problem where a trustworthy ledger is required in a decentralized setting where it is impossible to trust actors, humans, and computers completely. The blockchain is a series of procedures and cryptographic mechanisms applied to a shared network to secure data storage within a distributed database composed of authenticated blocks encapsulating the data. The trust factor is a core feature of blockchain technology. Through blockchain, the cryptographically open-source code is used to manage the trust. Using encoding methods, each data block is safely handled in a secure layer. The data is passed to the miners, who verify it by solving mathematical puzzles and attaining consensus. The three key principles ensuring the system's functionality are (1) blocks and hashing, (2) mining, and (3) consensus [64, 120, 121]. The architecture of the blockchain is illustrated in Fig. 4.

3.2 Smart contract

The smart contract idea was developed a long time ago, but it was introduced recently. The idea of a smart contract was articulated about twenty years ago by a cryptographer scholar, Nick Szabo [153, 154]. The fundamental principle is to insert contractual concepts into computer components

Table 2 Publication sources of the selected papers

Publication source	Channel	References	No	%
Other conferences (ACM, Springer)	Conferences/Symposium/Workshop	[2–48]	48	25.1
IEEE	Conference/Symposium	[40, 49–78]	33	17.2
IEEE	Journals/Transactions	[79–108]	31	16.2
Other journals	Journal	[109–134]	30	15.7
Elsevier	Journals	[135–151]	18	9.4
Blockchain and smart contract	White paper	[1, 152–161]	11	5.7
Online	Online material	[162–169]	8	4.1
arXiv Preprint	e-Print Web archive	[170–175]	6	3.1
Books/Thesis	Book/Book chapter	[176–180]	5	2.6

Table 3 Mapping of research questions with the references

Research questions	References	
	Journal	Other sources
RQ1	[1, 70, 85, 109, 141]	[13, 43, 157–169, 181]
RQ2	[81, 90–107, 111–113, 119, 125, 129, 132, 137, 142–153]	[14–31, 51, 67–80, 126–128, 130, 131, 133–135]
RQ3	[85, 86, 88, 110, 140, 141, 153],	[32–36, 170, 171, 175, 179–181]
RQ4	[85, 86, 88, 108, 136, 140, 141, 176, 177]	[37–48]

such as liens, trusts, etc. He proposed four fundamental principles of contract design: evaluation, validation, privacy, and enforceability. Based on Szabo's concept, contracting parties should evaluate their success, check whether the contract was performed or violated. Also, the smart contract protects all parties' privacy and distributes the details as much as is required, and eventually, it would execute automatically. Nevertheless, the necessary architecture and specifications were not available at that time. The concept remained only an abstract term; today, smart contract deployment has become realistic by advancing blockchain technology. Smart contracts are a package of codes that encode and replicate real-world contractual agreements in the computer domain. A basic principle for contracts is to create a legal agreement between two or more parties that each party must meet its contractual obligations. The important consideration is that the contract will be governed by a legitimate administrative entity (organization). Smart contracts are eliminating the trustworthy third parties, that is, the mediators between contract members. They manage the mediators by using automatic execution of programs in a blockchain network, decentralized and evaluated by the network nodes. The smart contract also allows transactions between untrusted parties without (i) mediator commission fees, (ii) trusted-party dependency, and (iii) the counterparties' need for mutual interaction.

Smart contracts comprise a contract space, a balance, and code. It can be generated and granted access to any node in a network simply by publishing a transaction to the blockchain. When included in the blockchain, the smart contract code is fixed and cannot be changed. A network of miners who are accountable for managing the blockchain runs smart contracts. Miners achieve agreement on the smart contract's implementation result and upgrade the blockchain accordingly. Once implemented, each contract is assigned a 160-bit address and is executed using this address if a transaction is generated. The smart contract used various platforms for the development of applications. Ethereum is the largest and most important platform for developing decentralized applications, ranging from predicting markets and identity systems to other economic applications. Bitcoin is also a blockchain software that Nakamoto first created. Bitcoin offered a modern, improved exchange of money, emerging markets, and new independent decentralized organizations [65]. While Bitcoin's primary purpose is to transfer capital, its blockchain's immutability and openness have facilitated the creation of protocols that implement smart contracts. Many smart contracts store the blockchain data via the Bitcoin scripting language [87]. Apart from Bitcoin and Ethereum, there are continually evolving numbers of alternative systems derived from or are separate from the initial Bitcoin network and provide enhancements and innovative solutions for different impediments encountered in the former.

4 Related work

So far, several surveys and methodologies conducted by different authors across the globe, which considered different applications of smart contracts [8, 8, 9, 9, 115, 136, 172]. Considering the literature surveys, most of them explore blockchain technology applications, whereas others use smart contract techniques. Several general studies of blockchain technology, such as [10, 116] or survey articles, concentrate on a particular feature such as security [11] or decentralized applications [82, 137], or unique applications such as healthcare [117, 118] or IoT [119] are accessible. Nonetheless, none especially researched smart contract applications.

As per our knowledge, only a few surveys are there, which considered multiple smart contract domains. But there is no proper explanation and analysis of applications of a smart contract. Examples of these studies include the smart contract usage in many fields like medical [2, 49, 49, 50, 50, 51, 51, 52, 52, 53, 53, 79, 135, 176], supply-chain [3, 54, 54, 55, 55], IoT [4–6, 56, 80, 114, 170], identity mechanism [7, 57–59, 136, 171], data management [8, 60, 61, 115, 172], and more [9, 62, 81]. While there is yet another study on smart contracts [11], it is restricted to the network and security part of Ethereum. Although there are many studies on blockchain technologies [63], we contend that there is little focus on surveys

Table 4 Category-wise academic papers distribution

Category	Focus	Academic papers
Platforms	Ethereum	[11, 32, 85, 86, 109, 110, 141, 157, 181]
	Hyperledger Fabric	[13, 85, 86, 109, 141]
	Nem	[86, 109, 158]
	Corda	[109, 141, 159]
	Stellar	[109, 141, 164]
	Waves	[86]
	Cardano	[86, 109]
	Neo	[86, 160]
	EOS	[86, 109, 141, 161]
	RSK	[109, 141, 162]
	Tendermint	[86, 163]
	Quorum	[86]
Applications	Insurance	[14–16, 67, 90, 125, 126]
	Supply Chain	[17, 91, 92, 127, 142–145]
	Internet of Things	[18–20, 93–96, 146–150]
	Healthcare	[21–23, 51, 68–73, 81, 97, 111–113, 119, 128–130, 137, 151]
	Multimedia	[24, 74, 75, 98, 99, 152]
	Cloud Computing	[25–29, 76–79, 100–103, 131, 132]
	Identity Management	[30, 80, 133–135]
	Record Management	[31, 104–107]
Challenges	Readability	[141]
	Code correctness	[141]
	Dynamic control flow	[140, 141]
	Execution efficiency	[87]
	Lack of privacy	[32, 33]
	Unknown call issue	[11, 179, 180]
	Unorder of exception	[11, 34, 181]
	Gas exception and typecast issue	[11, 34, 170, 171, 175]
	Re-entrancy	[35, 88, 153, 175, 181]
	Programming smart contracts	[11, 36, 170, 182]
	Privacy issues	[85–88, 141]
Advances	Recover source code	[37, 141]
	Human readable execution	[108, 136, 141]
	Re-entrancy	[38–40, 140]
	Bytecode analysis	[41–43]
	Graph-based analysis	[44]
	System for execution	[176]
	Serialization of execution	[46, 47, 177]
	Contract inspection	[41, 48]

based on smart contract-enabled applications. Smart contract applications are not entirely explained in many review papers [138]. Indeed, some studies aimed at the smart contract application and include the applications' privacy and security challenges. These reviews consider only a few applications [83] and mainly focus on security [84–86]. Some surveys also consider blockchain's particular area and include the details of challenges and advancements [139]. Other reviews consider the smart contract's privacy problems [86]. Furthermore, other studies, such as [83], are based on the challenges and recent progress in the smart contract and the brief discussion of applications. Our paper presents the smart contract subject analysis, which examines comprehensive facets, requirements, applications,

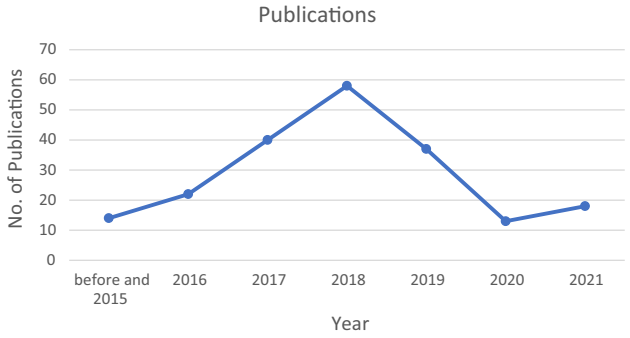

Fig. 3 Publications year wise

drawbacks, and solutions in this area. Table 5 presents a comparison of existing literature reviews and survey papers. The existing studies lack a detailed and comprehensive analysis of state-of-the-art applications allowed by smart contracts and their challenges.

All the earlier studies under consideration discuss the security challenges and plan to incorporate smart contracts in different environments. This article investigates the use of blockchain technology-based smart contracts as a whole without being specific to particular applications, thus addressing its current trends, classifications, and open issues that have not been discussed in the prior surveys. We are aiming to provide a detailed overview of the usage of smart contracts in different application areas. To the best of our understanding, our research surpasses all the current studies more systematically in terms of the core principle of a smart contract for IoT, healthcare, cloud computing, multimedia, supply chain management, insurance, and artificial intelligence.

5 Platform overviews and key concepts

Smart contracts have recently been developed on platforms based on blockchains. Such frameworks provide easy interfaces for developers to create smart contract applications. Many of these can endorse smart contracts across a variety of existing blockchain platforms as shown in Table 6. In this paper we discuss the most influential smart contract systems in the following sections: Ethereum [155], Hyperledger Fabric [13], Nem [156], Corda [157], Stellar [162], Waves [163], Cardano [164], Neo [158], EOS [159], Rootstock [160], Tendermint [161], and Quorum [165]. We chose them primarily because of the importance of growing social and technological maturities, as suggested in [122].

5.1 Ethereum

Ethereum is a distributed network designed to implement smart contracts [155]. Unlike Bitcoin's script system, which is incomplete turing, Ethereum created Turing complete languages like Mutan, Serpent, and Solidity [167] to offer generic use cases other than cryptocurrency applications. Ethereum compiles, runs, and loads software codes into Ethereum Virtual Machine (EVM) for smart contracts written in Solidity, Serpent, and Mutan languages. Also, Ethereum is based on an account-based data model that recognizes the users using a digital wallet. Similar to Bitcoin, Ethereum uses a proof of work algorithm that is also computing-intensive. Ether (ETH) is used instead of Bitcoins to account for the difficulty of solving puzzles performed by miners. Gas serves as an internal cost to execute a contract given ETH's volatile value. Informally, a transaction's overall cost can be calculated using gas limit/gas price. The gas limit determines the acceptable amount of Gas that must be used to generate a block, and gas price is the cost of a gas unit (in ETH). Users must spend different amounts of Gas to verify their sales sooner or later (i.e., large quantities of Gas helps in a fast verification). Since PoW is computationally costly, energy can be wasted for meaningless mining tasks by block. It is planned when the mining method is used for practical activities, such as solving mathematical problems and performing machine learning exercises.

5.2 Hyperledger fabric

Hyperledger Fabric is now a collaborative blockchain and smart contracts platform [13]. Like Ethereum, which operates a virtual machine (i.e., EVM) for smart connections, Hyperledger embraces Docker containers for

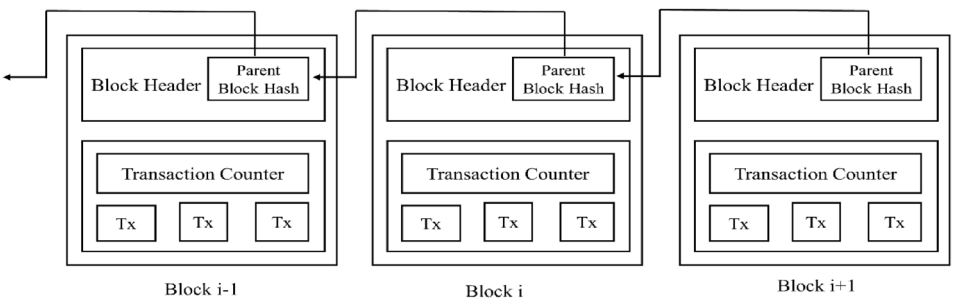

Fig. 4 Blockchain architecture

Table 5 Comparison of related work

Review papers	Year	Focus area	Remarks
[55]	2021	Aspects of blockchain adoption in supply chain management	Investigate the factors that facilitate blockchain adoption in supply chain management. It provides hypothesized acceptance models to identify the progressive literature on blockchain-based supply chain systems. The review also identifies the challenges and future directions in the field of the supply chain. Again, the study focuses on a particular domain that is the supply chain
[6]	2021	Privacy issues in blockchain-based IoT applications	Study the security challenges and privacy issues related to the IoT network using blockchain technology. It also highlights the new challenges imposed while utilizing blockchain technology in IoT systems. The study considers only the IoT domain, focusing on the security aspect, and lacks the details of smart contract applications and platforms
[9]	2021	Security issues in blockchain-based cloud computing applications	Perform in-depth and systematic reviews of cloud computing systems using blockchain technology. It examines the cloud security concerns such as trust, integration, and privacy using blockchain. This review covers only one application area: cloud computing, without any details of the smart contract concept
[62]	2021	Trends and challenges of decentralization model of internet	Identify current internet model challenges such as scalability, reliability, security, and trustability. Then, provide the opportunity of using blockchain technology in the internet model. Also, it explores the emerging internet technologies that benefit from using blockchain to provide better solutions. This review lacks the discussion of using smart contract techniques and platforms in the internet model
[138]	2018	Security, privacy, and scalability analysis of smart contracts	Present a discussion and analysis of smart contracts' problems and solutions regarding security, privacy, and scalability parameters. The study lacks a detailed analysis of smart contract applications
[83]	2019	Architecture of application areas related to smart contract	Review the blockchain-enabled smart contract platforms proposed a research framework and identify the smart contract's technical and legal challenges. The study covers only five application areas with a detailed structure of two platforms
[84]	2019	Security analysis of smart contract applications	The survey provides an analysis of six smart contract applications and discusses the associated security and privacy issues. The study discusses the eleven smart contract platforms specifically in terms of consensus methods only
[85]	2019	Security verification techniques related to smart contract	Survey the security verification topics of smart contract and presents the positive and negative aspect of each topic. The study considers only the security aspect of a smart contract without including applications and platforms
[86]	2020	Vulnerabilities of smart contracts	Survey vulnerabilities of smart contracts and analyze the various attack techniques to determine smart contract technology's impact. This study lacks the analysis of vulnerabilities in smart contract applications
[139]	2019	Challenges and advances of smart contract applications	Conducts a review of six smart contracts applications and platforms. It also presents challenges as well as technical advances. The study lacks an in-depth analysis of recent advances of smart contracts in the domain of multimedia, healthcare, cloud, and supply chain
Proposed work	2021	General study and comparative analysis	Conducts a systematic literature review of twelve platforms and eight smart contract applications. The proposed work performs an in-depth comparative analysis of each smart contract application, identifies the challenges, and presents the respective solutions

application deployment. Unlike virtual machines (VMs), Containers support smart contract implementations with lower latency while lacking isolation (i.e., programs executing on top of one operating system in one container). Hyperledger embraces standard, high-level programming languages such as Java and Go (aka Golang) rather than creating Ethereum's smart contract languages. Likewise, Hyperledger is now Turing complete. The data model is adopted as a key-value pair. The Hyperledger blockchain network (private or consortium) is allowed because Fabric serves general business applications. The users must be approved by Certificate Authorities (CAs) to access the network. Since the network includes various functions, several forms of CAs coexist. The Enrolment Credential Authority (ECA), for example, requires users to connect for blockchains. After the customer has authenticated, he/she will ask the Transaction Certificate Authority (TCA) for transaction certificates. Inside the authorized blockchain

Table 6 Smart contract platforms

Parameters	Native token	Consensus	Permission or permissionless	Smart contract language	Data model	Application	Turing completeness
Ethereum	ETH	PoW and PoS	Both	Solidity, Flint, SCILLA, Mutan	Account-based	General	Turing complete
Hyperledger Fabric	No	PBFT	Permissioned	Java, Golang	Key-value pair	General	Turing complete
Nem	XEM	Proof of authority	Both	Java	Account-based	Digital currency	Turing complete
Corda	No	Raft	Permission	Kotlin	Transaction-based	Digital currency	Turing incomplete
Stellar	Lumen (XLM)	Stellar Consensus Protocol	Both	Python, Javascript, Golang, and PHP	Account-based	Digital currency	Turing incomplete
Waves	Waves	LPoS	Public	RIDE	Account-based	Digital currency	Turing incomplete
Cardano	ADA	Ouroboros, PoS	Public	Plutus	Account-based	Digital currency	Turing incomplete
Neo	NEO and GAS	dBFT	Public	Java, Python, C#	Account-based	General	Turing complete
EOS	EOS	DPoS	Public	C++	Account-based	General	Turing complete
RSK	RBTC	BFT-DPOS	Public	Solidity	Account-based	General	Turing complete
Tendermint	No	BFT	Permissioned	Any language	Account-based	General	–
Quorum	No	Raft-based and Istanbul BFT	Permissioned	Solidity	Account-based	General	Turing complete

network, consensus can be achieved quickly. The Fabric takes advantage of Practical Byzantine Fault Tolerance (PBFT), which involves multi-round voting between authorized entities. PBFT depends on multi-round communication between nodes, which can lead to delays in time. To solve this problem, more powerful consensus algorithms should be created.

5.3 Nem

Nem was introduced on 31 March 2015 [156]. Nem is developed in Java, one of the world's most commonly used developing language. It makes it super open because programmers do not need to know a specific language like Solidity, Golang, etc. Also, Java is mature and has fewer security vulnerabilities than the latest languages, such as Solidity. Nem recently launched the Catapult or Mijin v.2 upgrades, which has made it the safest smart contract platform, according to various security experts. It is a technological breakthrough that unlocks fresh opportunities for the use of a blockchain network. Nem's greatest selling point is that it is extremely scalable. Also, while Ethereum can handle about 15 transactions a second, Nem can handle the 100's. This is because these developers are gradually jumping from other systems such as Ethereum.

5.4 Corda

Like numerous Ethereum applications, Corda [157] is specialized in digital-monetary applications, and it supports the preservation and sharing of historical records with the help of a ledger. Corda supports programming languages such as Java and Kotlin, which run on top of the Java Virtual Machine (JVM). Meanwhile, to help the verifiability, Corda is Turing incomplete. In comparison, the transaction-based model is utilized in Corda. Corda usually encourages shared networks where businesses set up a network to exchange digital assets privately. Consensus could be easily reached on private networks with blockchain. Corda embraces the Raft consensus mechanism [166]. Mutual agreement in Raft may be created by choosing a supervisor, log replication, and security protection. Corda uses the point-to-point exchange approach instead of communicating globally in blockchains. Consumers ought to present the message recipients and the essential details that are to be transferred.

5.5 Stellar

Stellar [162] is a developer forum for digital-currency use cases, similar to Corda. Stellar is faster relative to Ethereum and more available. Meanwhile, Stellar can serve a range of languages such as PHP, Golang, Python, and JavaScript. Stellar operates device instructions on top of Docker pods, similar to Fabric's, thereby decreasing overhead. For example, one stellar transaction's execution cost is just $0.0000002 and can be almost overlooked. In comparison, the execution period for a single transaction in stellar is average, around 5 s, compared to 3.5 min in Ethereum. Stellar is thus a perfect medium for financial applications. Similar to Ethereum, Stellar embraces the application architecture as the account-based platform. Stellar built its consensus protocol—Stellar Consensus Protocol (SCP) [162]. As Stellar is permissioned, it is easy to reach consensus via SCP.

5.6 Waves

The Waves Platform [163] is a public blockchain platform worldwide, founded in 2016. Waves Foundation aims to recreate the DNA of technology worldwide by offering a digital platform, delivering easy-to-use, fully usable resources to make blockchain accessible to anybody who can benefit from it. The Waves platform uses blockchain technology primarily to support the issuance, trade, and exchange of digital assets or tokens. Proof-of-Stake is to be used as a consensus algorithm. In June 2016, Waves Network completed its Initial Coin Offering, garnering over $16 million (BTC 30 000).

5.7 Cardano

Cardano is a layered architecture smart contract network that offers scalability and protection features [164]. Cardano is a blockchain of the third generation which aims to introduce scalability and interoperability into the blockchain network. Compared to other smart contract platforms, there is one exciting quality that makes Cardano unique. The majority of the other smart contract systems are coded through the imperial language of programming. Cardano uses Haskell for its source code and is a structured language for programming. Cardano uses Plutus for its smart contracts and is also a functional language. Plutus is a turing incomplete language. Cardano follows the account-based data model.

5.8 Neo

Neo, also known as Antshares, is also referred to as the Chinese Ethereum [158]. Neo is a non-profit community-based blockchain initiative that uses blockchain technologies and digital identity to digitize properties. It simplifies digital asset management through smart contracts and realizes a smart economy using a distributed network. Neo follows the account-based data model. Neo is developed in Java, Python, and C# programming languages. It supports the delegated Byzantine Fault Tolerance (dBFT) consensus algorithm. Neo's key goal is to be a distributed network for the smart economy. A smart economy is an integration of digital assets, digital identity, and smart contracts.

5.9 EOS

EOS [159] is planned to make decentralized applications scalable. Rather than supporting a single consensus algorithm, EOS incorporates Byzantine Fault Tolerance (BFT) and Delegated Stake Proof (DPOS) algorithms to benefit the consensus mechanisms. Delegates should be chosen at each round by stakeholders to create a new block, and BFT should decide to make the block permanent among those delegates chosen. Compared to Bitcoin, EOS follows the account-based model, but it also allows human-readable names to reference all accounts. Instead of customizing a program execution as per the virtual machine such as Ethereum, EOS supports Wasm to implement a smart contract.

5.10 Rootstock (RSK)

RSK [160] runs on Bitcoin while enabling quicker execution of transactions. For instance, RSK may ensure within 20 s that the transaction is being executed. Additionally, RSK is compliant with Ethereum (such as the use of Solidity to execute contracts). It supports Turing complete contracts. In reality, RSK built its virtual computers for smart contract management. RSK has built its PoW-based consensus method to support lightweight implementation while decreasing the overhead. Unlike Corda and Stellar, RSK has been primarily suggested to support currency applications. Since it runs on top of Bitcoin, RSK has a benefit, i.e., it is better than other systems independent of blockchains.

5.11 Tendermint

Tendermint is an open-source initiative that takes a strategy distinct from other blockchain technologies [161]. Tendermint, at its core, provides a blockchain consensus

platform and peer-to-peer networking features that can be accessed by an Application BlockChain Interface (ABCI). Therefore, the blockchain can be written in any programming language to make all "application-level" decisions, such as structuring to execute transactions independent of the coordination mechanisms underlying them. Distinguished approved nodes engage in the consensus protocol by taking turns proposing and voting on the next block in a two-step process. The framework will suggest a currency and denote each node's voting power in that currency, thereby effectively enforcing a consensus of Proof-of-Stake. However, the consensus protocol's dependence on timeouts renders it a "weakly synchronous" one defined by other sources as unsuitable for distributed applications that are publicly hosted. Using Tendermint for the content blockchain project allows the freedom to agree and build certain functionality such as smart deployment contracts even without dealing with low-level networking and consensus protocols.

5.12 Quorum

Quorum is a corporate version of Ethereum [165]. Quorum is suitable for any program involving high-speed and high-performance analysis of private transactions within a network of established participants. Quorum addresses common issues about the implementation of blockchain technologies within and outside the financial sector. Quorum is a shared ethereum application 'geth' fork with several protocol-level changes to meet market requirements. The Quorum project's primary purpose is to develop an ethereum enterprise client that empowers businesses to embrace blockchain technology and benefit from it. Since Quorum is an open-source project, its codebase is open to analysis from everyone, which encourages trust in the software. Further, open-sourcing improve acceptance and encourages developers from different sectors to participate in this platform's growth.

6 Smart contract applications

Smart contracts have a wide variety of uses, from the Internet of Things to multimedia. In specific, we generally categorize large smart contract implementations domain-wise. We explain them in more detail in the further sections.

6.1 Insurance

Applying smart contracts may decrease overhead processing and save costs in the insurance sector, especially claim handling [123]. The preparation and settlement of an insurance claim can take months because of a lack of automatic administration. It is troublesome for insurance providers and adds to compliance charges, gluts, and inefficiency. Smart contracts will automatically streamline the mechanism by automatically initiating a demand when such events occur, such as motor insurance [14]. The smart contract automates the motor insurance process by sharing legit information in the distributed network, improving the claim process and efficiency, and decreasing claims processing time and cost. As shown in Fig. 5, a smart contract defines code to set up the terms and conditions for all the participants. In this process, a smart contract acts as a central part for all policy stakeholders and automatically executes them whenever events trigger the involved stakeholder. Table 7 summarises the various insurance system applications.

6.2 Supply chain management

Management of the supply chain includes the passage of goods from raw material to complete products. Smart contracts may monitor ownership rights as products pass through the supply chain at any given moment, verifying who is responsible for the commodity. The monitoring of products from producers to warehouses, from warehouses to retailers, and retailers to suppliers, has been much simpler. Each delivery process stage verifies the finished product until it reaches the customer. When an object is delayed or misplaced, it is easy to search the smart contract to figure out exactly where it will be. When any stakeholder fails to comply with the contract terms, for example, if a manufacturer refuses to deliver a package on time, it will be transparent. Smart contracts make supply chains more transparent to smooth out the transfer of goods and restore trust in commerce. For example, Fig. 6 represents the blockchain-based supply chain management architecture for tracing product details. Blockchain technology is often used to record product status at each phase of its lifecycle. It helps to track products starting from the initial stage and strengthen supply chain operations.

Additionally, the blockchain participants are given unique identifiers, or digital signatures, to sign the blocks and add product details with RFID tags to the blockchain. This would eliminate execution errors and improve traceability. The output flow from smart contract to mobile application helps users track the finished product at each stage of the delivery process by using the smart contract's utility. If an item is delayed or lost, the smart contract can be consulted to find out exactly where it should be. If any stakeholder fails to meet the contract terms, for instance, if a supplier did not ship on time, it would be clear for every party to see. Making supply chains more transparent via smart contracts helps smooth out goods' movement and

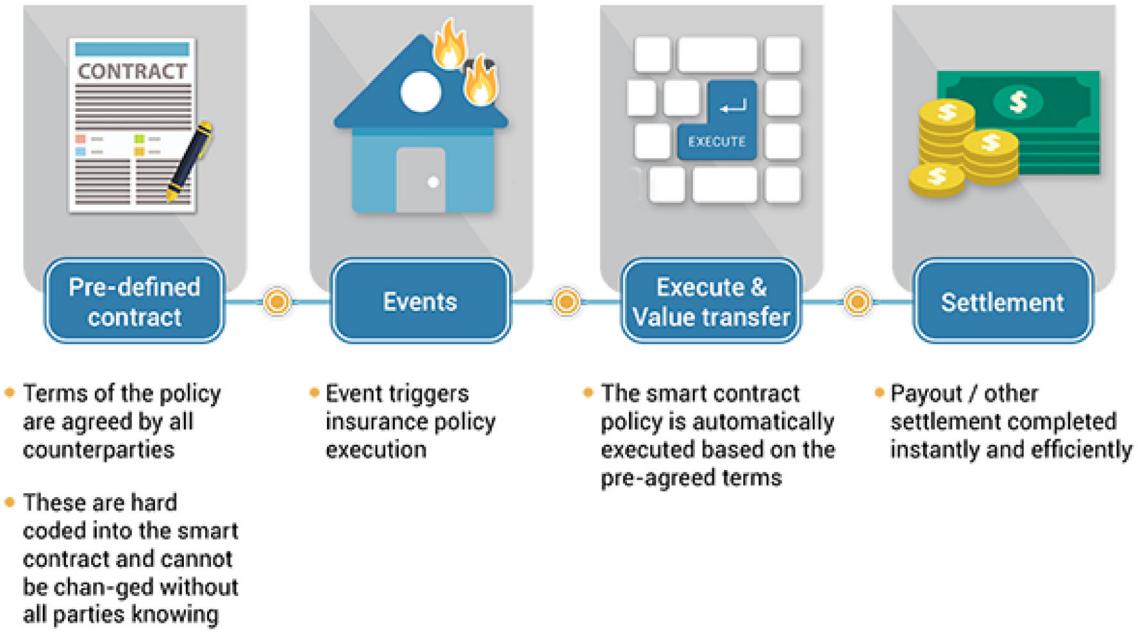

Fig. 5 Smart contract-based insurance system

Table 7 Insurance system applications

References	Year	Method	Contribution
[88]	2020	AI-based architecture for the automated insurance system	Presented an automated insurance system that secures insurance activities, detects fraudulent claims, reduces human interaction, and reduces monetary loss. It designed smart contracts to define rules for participants and enable the insurance sector's claims processing and refund process. The proposed architecture used artificial intelligence algorithms to predict the suspicious claims and percentage of the premium amount
[15]	2019	Smart contract based cybersecurity supervision in the insurance system	Presented analysis of cybersecurity issues in the insurance sector and identified the advantages of smart contracts to minimize security risks. It also outlined the existing regulatory framework against cyber risks and suggested the smart contract-based framework to deal with the associated security issues
[66]	2019	Decentralized insurance sector with IoT sensors	The proposed architecture supports the IoT devices to register the events triggered by the insurance contracts and offers privacy protection by using blockchain. It used an Ethereum platform for the implementation process and to evaluate the design of smart insurance contracts
[16]	2019	Smart contract application for the insurance services	Proposed smart contract application for the insurance sector with the help of the Ethereum platform. The objective of the proposed architecture is to automate claims operations and to manage payments with existing cryptocurrency
[124]	2021	Traceable online insurance claims using blockchain and smart contract technology	Proposed a smart contract-based system to solve the underwriting issue by storing insurance records and personal health records on the blockchain ledger. It also maintains the trust between policyholders and insurance institutions to provide effective claim traceability

restore trust in trade. Table 8 summarizes the analysis of supply chain applications that focus on smart contracts and blockchain technology.

6.3 Internet of things

The Internet of Things is the most exciting technique that serves various use cases, including supply chain systems, production tracking systems, stores, access controls,

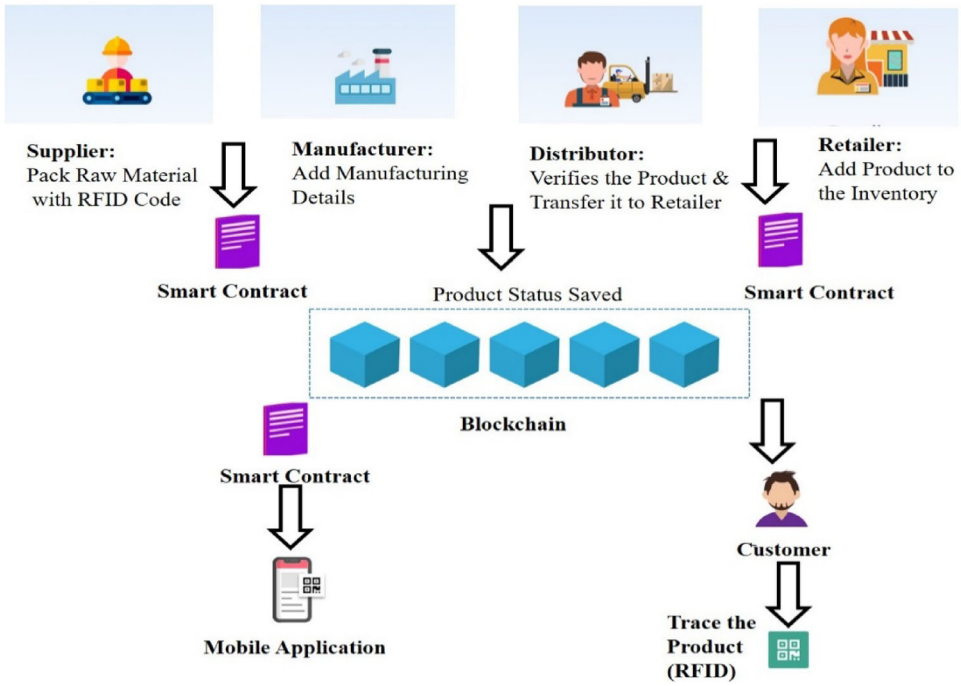

Fig. 6 Smart contract-based product tracking system

Table 8 Supply chain applications

References	Year	Method	Contribution
[140]	2018	Double chain based agriculture supply chain system	Proposed a double chain-based blockchain architecture for the agriculture supply chain system. It supports adaptive rent-seeking and matching between the supply and demand of resources
[141]	2019	Blockchain in operations and supply chain	Demonstrated the architecture of the blockchain-based logistics monitoring system using the Ethereum platform
[17]	2019	Blockchain architecture for the containerized food chain	Demonstrated the architecture using Hyperledger fabric to provide secure information sharing, enhance process access, and prevent risks in containerized food supply chain systems
[142]	2020	Blockchain-based Indian agricultural supply chain	Investigated the barriers to the adoption of blockchain in the Indian agriculture supply chain. Then, the identified barriers are modeled in the proposed architecture and evaluated the performance
[143]	2020	Blockchain-based supply chain management for information sharing	Suggested blockchain-based information sharing architecture that brings benefits to supply chain management. It also used a homomorphic encryption solution without a third party
[89]	2019	Blockchain-based construction supply chain system for key protection	Designed a blockchain-based private-key distribution framework to preserve key security and recovery. It provides payment security in the construction supply chain by using a key management scheme
[90]	2020	Blockchain-based safety management system in grain supply chain	Designed a multi-storage system for the grain supply chain. It supported data security, reliability, information interconnection, sharing, and whole process tracing features
[125]	2021	Smart contract-based agriculture food traceability	Proposed a consortium framework using smart contracts to trace the workflow in the agriculture food supply chain system and eliminate the need for central agencies. It ensures the security, integrity, and reliability of the food supply chain system

databases, e-health services [144–146]. IoT's key goal is to incorporate "digital" things into the internet and offer consumers specific services [147]. Implicitly, IoT was developed to automate the various business processes. The potentials of IoT can be increased with the incorporation of smart contracts. For example, industrial manufacturing.

Most of the existing manufacturers adopt a centralized approach to their IoT ecosystems. The firmware updates can only be obtained manually by different IoT devices on the central server by querying the server. Smart contracts give the problem an automated solution [91]. The smart contract stores the hash values of firmware updates and is executed on the distributed blockchain network. Resources are greatly saved as the smart contracts are used to access the firmware information directly. Also, smart contracts can give benefits to the IoT E-business model. The conventional business system also needs a trusted authority to complete the transfer, acting as an intermediary. This centralized payment, though, is expensive and can't completely leverage IoT benefits. In [18], it was suggested that Distributed Autonomous Corporations (DACs) manage purchases, in which there are no conventional roles such as governments or payment firms involved. Because of smart contract implementation, it works automatically without any intervention. Table 9 presents the analysis of IoT applications that focus on blockchain technology and smart contract.

6.4 Healthcare

Existing healthcare networks suffer from numerous issues such as fragmented data, data accuracy, interoperability, and privacy issues. By suggesting a framework [67, 126] or implementing a smart contract-enabled system using blockchain [2, 49, 49, 50, 50, 68, 79], many works aim to resolve these issues. These work focus on user identity management, access control, and sharing of medical data using various available blockchain platforms such as Ethereum [68, 111, 135], and Hyperledger [49, 50]. Kuo et al. [117] also explore key blockchain advantages for various healthcare applications such as record processing, insurance claim procedure, clinical testing, or Healthcare database development. As shown in Fig. 7, the blockchain-based healthcare model designs the smart contract to register the application users and allow them to share, access, and validate the healthcare data through the blockchain network.

Table 10 presented the selected studies emphasizing smart contracts, or at least the blockchain concept has been addressed in the architecture.

6.5 Multimedia

Violations of privacy and authorship are significant issues within the entertainment sector. Musicians, photographers, writers, and other artists are deprived of their royalties because their intellectual property is dishonestly exploited. Having a public authorship list on a blockchain is an innovative example of how smart contracts change affairs. Each time material is used for advertising activities, such

Table 9 IoT applications

References	Year	Method	Contribution
[19]	2018	IoT devices ownership management	Proposed a blockchain-based architecture for managing the ownership details of IoT devices without a trusted party. It defined the ownership rules with the help of smart contracts
[92]	2018	IoT-edge framework using blockchain and smart contract	Proposed a blockchain and smart contract-based edge-IoT framework to address security and scalability challenges. It uses a credit-based resource management method to control the resource requirement of IoT devices
[20]	2021	Smart contract-based policy for IoT	Designed smart contract policies to enforce the agreement for the IoT network entities. It proposed three different policies: hardware policy, authentication policy, and application policy for the IoT system
[93]	2018	Smart contract based monetization of IoT data	Proposed a smart contract-based monetization framework in which payment and token access issues automatically to the IoT owner without any third party
[148]	2019	Auditing scheme with a smart contract for IoT	Proposed a decentralized auditing scheme that supports dynamic auditing, public auditing, and batch auditing. Smart contracts are used to store deposits, which can pay for auditing and punishment
[94]	2019	Contract learning approach for resource sharing and task offloading	Introduced a two-layer architecture for resource sharing and task offloading in the IoT system. It combined the contract concept with computational intelligence. The first layer uses an incentive method, and the second layer uses a task offloading algorithm to minimize delay in the IoT system
[126]	2021	Blockchain and smart contract-based IoT smart agriculture system	Suggested a system using blockchain, smart contract, and IoT devices to automate the process in the agriculture sector. It uses blockchain as a backbone to manage data, IoT devices to collect data, and the smart contract provides the interaction between the contributing parties

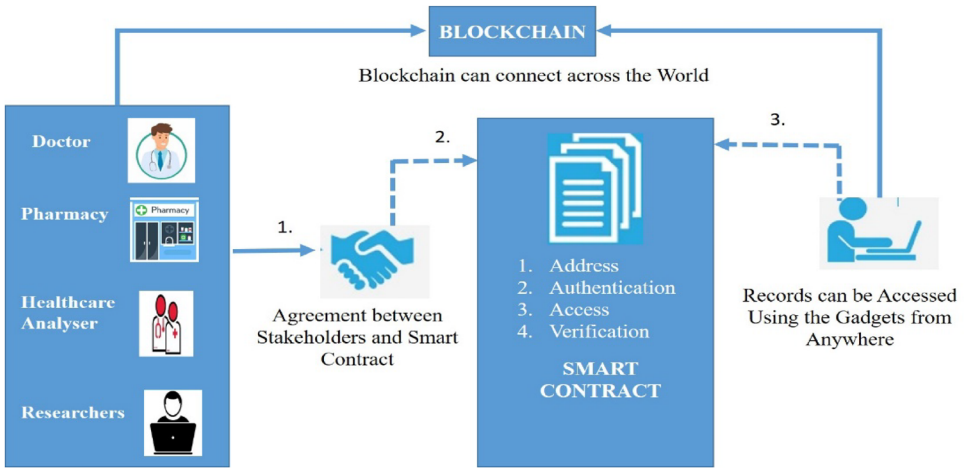

Fig. 7 Application model for blockchain-based healthcare application

Table 10 Healthcare applications

References	Year	Method	Contribution
[127]	2019	Blockchain-based smart healthcare system	Proposed a secured and smart healthcare system using blockchain to provide security and privacy to preserve the healthcare system
[149]	2019	Blockchain-based electronic health record	Proposed a Hyperledger based electronic healthcare record sharing system with the use of chain code. It proposed an access control policy algorithm to improve the data accessibility between healthcare providers
[21]	2020	Off-chain storage of patient diagnostic reports	Proposed a distributed off-chain storage system for patient record management with the help of IPFS storage. It preserves patient privacy and provides easy healthcare data
[95]	2019	Mobile cloud-based E-health system using blockchain	Presented the implementation of Ethereum blockchain in a real scenario with a mobile app and Amazon cloud computing. It preserves private health information with smart contracts, access control, security, and privacy features
[69]	2019	Secure sharing and accessing of medical data	Proposed a MedBloc architecture to allow patients and healthcare providers to access and share data securely. It used encryption and smart contract to regulate access to the records
[128]	2021	Smart contract-based healthcare system using Ethereum and Fog cloud	Designed a blockchain-enabled smart contract-based scheduling algorithm to collect patient data from different locations and distributes them to hospitals. It implemented task sequencing to schedule tasks in topological order
[22]	2021	Blockchain-based healthcare management system	Developed a blockchain-based healthcare architecture to enhance the security and privacy of electronic healthcare records. It utilized the orbitDB with IPFS database to store patient records
[70]	2018	Secure remote healthcare system for hospitals using smart contract	Proposed a remote healthcare system that provides privacy and security of health information with the help of a smart contract. It implemented a processing mechanism to filter the data from the sensors
[71]	2018	Blockchain and smart contract in decentralized health structure	Presented a decentralized model for the healthcare system. It used a scheme to distribute registered data for the creation of electronic medical records. It also proposed an algorithm for the use of a smart contract in the model

as an album, a copyright fee would be paid to that album's owner. A lot of people are involved in making an album. Therefore it can be hard to determine who has the copyright and who is entitled to payment because the current processes are not working well. This has led to confusion about entitlement, undoubtedly giving some contributors more than is due to others' detriment, while some don't get anything. Through tracking ownership rights in a shared blockchain network, smart contracts will guarantee that the payments go to the intended beneficiary. Smart contracts offer many benefits for many sectors, decreasing excessive risks and time consuming while increasing efficiency. By principle, they are more effective and trustworthy than conventional contract law, and since all acts are reported and checked, they are often considered to provide better protection. Also, the use of smart contracts can protect the

property rights of multimedia. For example, the smart contract can store the customer's wallet address and product id, and the same data is used to encode the digital product. In the case of unauthorized transfer of products without the creator's permission, the administrator utilizes the associated product and customer information to find its original creator. Table 11 presents a comparative analysis of various multimedia applications.

6.6 Cloud computing

Cloud computing is a popular platform for giving consumers access to a common processing and storage resource pool [74]. Customers may usually purchase services from trustworthy cloud service providers (CSPs). However, it is challenging to test CSPs' trustworthiness, as CSPs frequently partner with external enterprises to make more money. In [24], the authors suggested a smart-contract and game-theory approach. This approach's central concept lets two cloud servers query a database to do a single operation. In this approach, smart contracts are designed to induce uncertainty, deception, and mistrust amongst clouds. Then customers may conveniently decide the logical clouds which do not conspire and steal. The analysis was also carried out to validate this plan's feasibility. In the cloud computing environment, brokers are commonly used to provide services to users. In this process, the broker reviews the user requests to find out suitable services. This scheme required both the cloud service providers and users to trust the broker. If the broker has been hacked, both sides are insecure. In [75], the authors recently proposed a solution to prohibit brokers from using smart contracts. The core concept behind their strategy is to use distributed agreements [25] to identify users' needs. The various cloud storage applications are summarized in Table 12.

6.7 Identity management

Centralized approaches raise personal identity several issues, including the privacy of consumers and untrusted third parties. Also, they can generally not provide a single identity for different organizations to adopt. Accordingly, personal usage details may be open to other third parties grappling with the new identities. As shown in Fig. 8, a smart contract is designed to authenticate the IoT device's identity based on the stored information. The architecture utilized the blockchain structure to store and verify the IoT devices' details and improve the system's security and accuracy. Table 13 analyses the various smart contract-based identity management applications.

6.8 Record management

One of the key applications of blockchain is to provide a secure, tamper-proof, and trustable ledger for the management of records. Many applications, such as IoT, supply chain, medical, and cloud storage, have identified this blockchain feature. The use of blockchain technologies and smart contracts in academic studies was often considered for data management applications. Table 14 summarizes the various related applications.

Table 11 Multimedia applications

References	Year	Method	Contribution
[96]	2020	Smart contract based MPEG intellectual property rights	Analyze MPEG property rights requirements and present the evaluation of music and media's fair trade using blockchain technology. It creates an awareness of the MPEG ontology details and addresses the smart contract and blockchain challenges in the same domain
[73, 97]	2019	Detection of tempered images using blockchain	Proposed a blockchain-based architecture to register information about ownership and rights of the author and descriptive details of the image to detect a violation
[72, 97]	2019	Blockchain-based video integrity verification	Suggested a blockchain-based approach to combine hash-based message authentication and elliptic curve cryptography to verify the video's integrity
[72, 73]	2018	Multimedia privacy and provenance protection using blockchain	Proposed a blockchain-based multimedia protection framework that enables the users to take full control of data without trusted authority
[23]	2021	Blockchain-based protection of surveillance recordings	Designed a blockchain-based framework to secure real-time surveillance system recordings. It also proposes an algorithm to generate fingerprints of the frame and verify the frame consistency
[150]	2018	Global music industry using blockchain	Proposed a novel actor enabled the model to capture the unique properties of digital entrepreneurship

Table 12 Cloud applications

References	Year	Method	Contribution
[129]	2021	Decentralized cloud storage system using blockchain	Proposed a blockchain-based decentralized cloud storage system to ensure security features. It utilized the smart contract functionality to provide cloud users authentication, access control, and integrity checking features
[26]	2019	Privacy and auditing scheme	Developed protocol for cloud storage by utilizing blockchain and Rank-based Merkle AVL tree (RB-MHT) to preserve privacy and batch auditing to maintain the modified record's security in the scheme based on blockchain
[98]	2019	Encryption based searching	A secure technique of Public-key encryption to search for keywords referred to as SEPSE against Keyword Guessing Attacks (KGA) is developed. In this scheme, users can encrypt keywords through a threshold with the help of specific key servers
[130]	2019	Medical data storage using cloud and blockchain	Developed a framework for storing medical data using blockchain and cloud storage technologies to be stored and shared safely
[27]	2019	Decentralized storage	Developed an effective method to use blockchain and IPFS (InterPlanetary File System) to store provenance information. Users were also able to check the validity of their results
[99]	2019	Blockchain-based on-chain and off-chain storage	Suggested a performance-driven, auction-based reward system based on a blockchain consortium that ensures trust for both on-chain and off-chain information. The authors implemented a consortium that was used as a distributed hyperledger to tackle on-chain data protection. Using an information performance-driven auction system, the assessed data performance is used to maintain trust in off-chain data
[100]	2018	Role-based access control	An RBAC-SC was introduced to implement a trans-organizational framework for RBAC. Safe RBAC method (users cannot disguise roles and only allowed users must perform tasks), customer-oriented method (customers can report their duties to any agency), testable (everyone can verify the position of the user)
[101]	2019	Access control based decentralized method	Developed a decentralized scheme for securing cloud storage using access control. The conventional encryption algorithm based on attributes was modified by the introduction of Ethereum's smart contract technique. The distribution key is not reliant on the central authority, thus preventing the central authority's attacks
[76]	2018	Ethereum smart contract	Developed a model for controlling the access depending on blockchain, stored data in untrustworthy space, and implemented attribute encryption based on Ethereum smart contracts
[77]	2018	Decentralized auditing method	Developed a smart and decentralized public auditing plan for cloud storage, eliminating the TPA requirement for auditing
[28]	2018	Deduplication method using smart contract protocols	Suggested a deduplication scheme that allocates files to multiple servers and documents blockchain storage information. They describe smart contract-based protocols to ensure secure deduplication without central authority participants

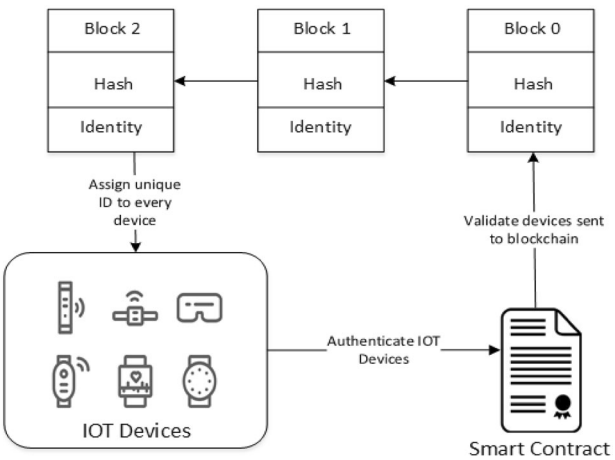

Fig. 8 Smart contract based identity management system

7 Challenges and future directions

While a smart contract is a revolutionary development, many challenges do remain to be addressed. We identified the smart contract's major challenges and summarized them in Table 15. The recent developments in overcoming these problems are presented in Sect. 7.2.

7.1 Challenges

Readability The smart contracts mainly used programming languages such as Solidity, Go, Kotlin, and Java. Then, they compile and execute source codes. Thus systems have different types of codes in different time intervals. Therefore it remains a major challenge to make programs readable in any form.

Table 13 Identity management applications

References	Year	Method	Contribution
[131]	2021	Blockchain-based IoT identity management system	Proposed a lightweight architecture using consortium blockchain technology to provide an identity management system. It addressed privacy, security, and scalability issues of centralized IoT systems
[78]	2019	Privacy-preserving identity management using smart contract	Proposed a novel identity management framework that manages the biometric data. It included the collection, storage, issuance, transformation, verification, and access control of identity data with a blockchain-based smart contract
[132]	2021	Smart contract-based self-sovereign identity management system	Recommended a cross-domain system using smart contract techniques to manage self-sovereign identity in the blockchain network. It preserves the privacy of the data using blockchain by providing complete ownership and management right of identity to the user
[29]	2019	Healthcare identity management using blockchain	Presented an insurance management framework in the healthcare sector to access the authorized person's identity and provide the insured amount after verification. It transformed the traditional industry to provide a more secure environment with the help of blockchain technology
[133]	2019	Cloud user identity management protocol	Designed an Ethereum based identity management protocol for cloud users. It enabled smart contracts to automatically provide the identity authentication process without trusted authority and the reputation system to prevent excessive cloud service providers

Table 14 Record management applications

References	Year	Method	Contribution
[102]	2021	Blockchain and smart contract techniques for record management	Proposed a framework using blockchain technology to maintain the record of registry papers. It also resolved the loan clearance tasks and provided transparency in the system
[30]	2019	Smart contract based electronic medical record management	Proposed a blockchain-based electronic medical record management architecture that utilized IPFS for the data storage and smart contract to save file hash values. It provided a protected environment against tampering attacks and ensured information transparency
[103]	2019	Blockchain-based educational record management	The proposed architecture uses blockchain technology to ensure the security and reliability of academic data and design smart contracts to regulate storage and sharing. It utilized off-chain servers for data storage and hash information stored on the blockchain. The cryptographic algorithms are used for the encryption and digital signature process to ensure security features
[104]	2020	Ledger based global human resource record management	Suggested a distributed ledger-based management scheme. It used a privacy-preserving framework for the management of human resource records. Smart contracts are designed to store the employee's information and provide security
[105]	2019	Data management analysis using blockchain	Analyzed the new interpretation of blockchain as an application of the data store. It identified and evaluated the best practices of blockchain in the domain of data operations. It also covered the insights of blockchain data analytics

Code correctness It is almost difficult to make changes after smart contracts have been implemented on blockchains. Therefore, assessing the validity of smart contracts before the formal release is vitally necessary. However, it is complicated to test smart contracts' validity due to the difficulty of designing smart contracts.

Performance expectancy refers to how much innovation in technology is considered superior to the technology currently in use. Because smart contracts exist on the blockchain, companies may understand the advantages inherent in the blockchain's decentralized existence. Consequently, transparency, immutability, and quick handling of transactions are potential benefits that can be realized compared to legacy technologies. To simplify manual operations, the prospect of embedding business logic in smart contracts is of utmost importance. Although all experts agreed that smart contracts could promote different domain solutions, a trade-off between scalability, cost, and security was noted. Given this trade-off, the anticipated result can be seen as an advantage for businesses. The use

Table 15 Summary of challenges and advances in smart contracts

Challenges	Advances
Readability	Recover source code [36, 139]
	Human Readable Execution [106, 134, 139]
Code correctness	Bytecode analysis [40–42]
Dynamic control flow	Graph-based analysis [43]
Execution efficiency	Serialization of execution [45, 46, 124]
	Contract inspection [40, 47]
Lack of privacy	System for execution [174]
Unknown call issue	Avoid mistyping the interfaces, correctly perform the type casting [11, 138]
	Avoid external calls [138]
Unorder of exception	Use Oyente to extract the control flow graph from the EVM bytecode of a contract [41]
	Propagating the exceptions through callers and reverting the state [138]
Gas exception and typecast issue	Develop functions that are not too gas-consuming [108]
	Use Ethereum wallets as well as Dapps to prevent erroneous sends [108]
Re-entrancy	Banning the nesting of calling within contract functions [138]
	Fuzz check on smart contracts [39]
Programming smart contracts	Use of an automated form method to detect errors and minimize checks [85, 138]
Privacy issues	Perform computations on encrypted data, and code obfuscation [11, 84, 85]

of smart contracts, in particular, can lead to a crucial competitive advantage over rivals [182].

Dynamic control flow While the distributed smart contracts are unchangeable, it is not assured that smart contracts' control flow would be immutable. A smart contract may communicate with other contracts in particular (e.g., movement of funds to the contract or forming a new contract). When designing the contract, the control flow of the smart contract has to be specifically planned. Over time, the presence of smart contracts will result in a larger number of interconnected contracts. Therefore, it is challenging to predict contract behaviors. Additionally, most current approaches focus on identifying possible dynamic flow control issues in systems while not necessarily guaranteeing the execution system's stability. So checking whether the execution environment is reliable is also essential.

Execution efficiency Miners execute smart contracts in sequence. In other terms, once the existing contract is ended, a miner does not execute another contract. The serialization implementation ultimately restricts the performance of the system. However, implementing smart contracts concurrently is difficult because of the mutual data between several smart contracts. Meanwhile, it is also essential to increase the reliability of smart contract execution. Also, checking the contract data without a proper medium is essential to improve the execution efficiency as it eliminates the need to redeploy a new contract.

Lack of privacy The privacy risk originated from concealing the business logic under the smart contract code. In [31], the authors identified that the Ethereum is designed to be safe only bytecode, and users are afraid to trust the contract about the unpublished source code. Moreover, due to software decoding the bytecode [32], contract protection has been put at risk.

Unknown call issue Many Solidity functions activate the callee/recipient's fallback function [11] when the signature function may not suit all of the features available in a Solidity contract. Hackers use primitives such as call, send, or delegate functions to transfer Ether to an address that does not exist by invoking the fraudulent fallback contract function. One of Ethereum's common assaults exploited the weakness [177]. The attack required participants to donate Ether in a simplified way to finance their choice of contracts and then utilize them to remove the capital [178].

Unorder of exception In [11], authors recorded two forms in which anomalies were handled. First, the execution was halted, and side effects were reversed along with all the Gas use. Second, the exception was passed through the chain restoring all the called contracts' adverse effects until it arrived at "call" with all Gas utilization. In [179], the authors reported that calling functions' return values were not checked in Solidity developers, as exceptions returned "0" when a call failed. In addition to this problem, [33] presented that around 27.9% of contracts calling other smart contracts by using the "send" function do not examine the return values.

Gas exception and typecast issue The "Gas" denotes the transaction execution costs. It is the measurement amount of Ether when a transaction is initiated [173]. If during the

transaction execution, Gas is exhausted, then the transaction will be revoked, and the sender has to bear all the miners' Gas costs [33]. In Ethereum, we can set the amount of Gas required for execution to not exceed the maximum limit (that is 2300) [11]. The problem is uncertainty [168] that arises due to the insufficient amount of Gas available when calling external functions [169].

Re-entrancy Re-entrancy was the most serious issue of smart contracts abused to render the DAO threat [179]. This revolved around the handing over power in an agreement between the two. In [173], the authors demonstrated the "withdraw" feature that collected money from an account by submitting money to a recipient and then changed the account balance. In [34] and [151], authors referred to this problem as critical and provided examples of attacks associated with money withdrawal.

Perceived technical capability This can be defined as leveraging technological capabilities to create a competitive edge. Companies need to comprehend blockchain and smart contracts to understand how they can be used for IoT, supply chain, and storage sectors. Additionally, smart contract programming can be complex, and organizations must trust developers to write secure code as per their requirements. Reaching experienced and respected developers may be challenging. Companies with good technical skills are more likely to overcome emerging technology challenges and achieve a competitive edge.

Programming smart contracts In programming smart contracts, [11] and [168] emphasized security as one of the key challenges. For example, in attempts to steal Ether from users, some documented attacks (Rubixi/GovernMental/DAO contracts) abused smart contracts' immutability. Security and usability issues were quite significant among the many problems listed in [35], as a blockchain guarantees data confidentiality and authenticity to ensure that blockchain-based applications are comfortable. There is also economic difficulty when programming smart contracts, in addition to security and reliability.

The most prestigious example is the expense of installing and running Ethereum's smart contracts, i.e., the Gas prices. The cost problem comes from the Ethereum system's architecture (the Solidity Compiler and the EVM), where programming directives written in Solidity and converted by the EVM into byte code are becoming too costly. The developers' most costly errors when programming smart contracts are writing and reading from permanent disk memory. That is understandable, as when data is placed in the network, it is kept in a permanent archive distributed across thousands of nodes. In [180], authors established that smart contracts in a functional language (Idris) and the use of an automated form method to detect compile-time errors minimize both the probability of programming errors and the need for checks. However, because they encode very detailed smart contract behavior properties, it brings a high Gas cost because programs become very large and therefore contain redundant operations in the resulting code.

Privacy issues The privacy issues increase when the organizations are not sure about risks associated with their digital industry assets. Network bottleneck, manipulation of data, Sybil attacks, or badly written code of smart contracts are vulnerabilities and threats to organizations. This may hamper the status of the organization if threats turn evident. Security and privacy actions should be managed and achieved. It is tedious for organizations to evaluate associated threats, carry out control strategies, and execute continuous monitoring to keep the risk minimal.

7.2 Recent advances

Recover source code It is seen in [36] that over 77 percent of smart contracts have not released available source codes, all of which include more than $3 billion in US dollars. The unavailability of source code makes the formal auditors of smart contracts elusive. In [36], the authors proposed a reverse engineering tool, "Erays" to process compiled smart contracts to address the problem. This reverse engineering method turns hex-encoded contracts into readable human pseudo-codes.

Human readable execution Although several projects aim to provide smart contract developers with high-level languages, these smart contracts are translated into other types, such as bytecode in EVM. In most cases, when stored on the blockchain and executed, two parties in the transaction have to accept the contracts. To overcome this problem, in [134], the authors suggested an intermediate-level language called IELE. IELE has a common Low-Level Virtual Machine (LLVM) syntax [106] that gives compilers high-level information during compilation time, communication time, run time, and idle time.

Re-entrancy Several plans have recently sought to overcome any of the above problems. Obsidian [37] has been recommended to fix re-entrancy attacks and money leakage issues. Obsidian exploits called states to permit continuity checking and state change inspection to mend re-entrancy vulnerability. An empirical method of data processing was suggested to discourage illegal digital currencies from stealing from the leak. In [38], the authors proposed to remove re-entrancy vulnerabilities by banning the nesting of calling within contract functions. Liu et al. [39] proposed that the Fuzz check on smart contracts be carried out by iteratively creating random but varied transactions to find bugs in re-entrancy.

Bytecode analysis Bytecode level analysis needs only the compiled smart contract bytecodes, which are much easier to access. How to use such bytecodes to identify

security threats has been an important topic of science. In [40], OYENTE was explicitly suggested to define possible security vulnerabilities, including mishandled exceptions and time-stamp-dependent issues. The rule-based representations for high-level bytecode analysis are produced based on the control graph generated by OYENTE [41]. In comparison, the smart contract delivery and management framework (SmartDEMAP) was introduced by Knecht and Stiller [42] to solve the trust issue during contract creation and implementation. Additionally, SmartDEMAP can also be fitted with other application quality assurance devices such as automatic bug-finders. Smart contracts can be applied in this manner only after the trustful requirements have been met.

Graph-based analysis Charlier et al. [43] suggested a multidimensional method to forecast smart contract interactions. This method especially incorporates stochastic processes and tensors to replicate current interactions and thus forecasts potential contract interactions. Additionally, the work in [44] presents a heuristic indicator of immutability in the control flow. This approach has, in particular, been evaluated on a call graph of all Ethereum smart contracts. Evaluating the call graph shows that two (out of five) smart contracts involve at least one-third party confidence.

System for execution EVMFuzz [174] was suggested to find bugs in the smart contract execution system. EVMFuzz continuously creates seed contracts for multiple executions of EVM to identify as many contradictions as possible among execution outcomes. With cross-referencing outputs, this approach will ultimately reveal vulnerabilities.

Serialization of execution To fill this void, Dickerson et al. [45] suggested a Software Transactional Memory-based solution that would require miners to execute contracts parallel. This strategy's core concept is to treat every smart contract's invocation as a speculative atomic action. Conflicts that happened during simultaneous executions will quickly be rolled back in this manner. Also, the study at [46] simultaneously explored smart contracts. In this article, market problems such as atomicity, interference, alignment, and capital control were well discussed. In [175], the authors suggested using constructive Computer Transactional Memory Systems to improve smart contract execution. The miner also saves a block graph of transactions into the database when performing contract transactions concurrently using multi-threading. The validators then re-execute the smart contract at the same time as the specified block graph. If the result is true otherwise, the block is appended to the blockchain.

Contract inspection The contents of the contract cannot be changed after deployment. What will designers do if they are asked to change certain values not outlined initially? One easy approach is to change and redeploy the smart contract. The redeployment of smart contracts can, however, result in more costs. In [40, 183], the authors proposed using the reification of memory architecture to recompile a compiled document's binary form. Meanwhile, [40, 183] work suggested the decompilation capabilities [47], in which the system would introspect without redeploying the smart contract.

8 Conclusion

This review outlines a discussion of smart contract enable applications. Specifically, we first present an outline of blockchain technology and smart contract. Then, we present a comparative analysis of various smart contract platforms. Next, we discuss the smart contract use cases, list various implementation areas of each application, and perform a comparative analysis. Meanwhile, we also identify the general challenges of the deployment of smart contract applications. Moreover, we also discuss the solutions for the identified challenges. This article would provide guidelines for designing safe and reliable smart contract applications and supporting blockchain technology growth. Smart contracts are growing rapidly with blockchain technologies, but a range of issues need to be tackled. Many emerging smart contract discussions focus on protection and safety issues, while the emergence of blockchain and smart contract implementations often poses new challenges. However, identifying and detecting these issues would need considerable effort in information development and data processing aspects. Filling this gap, therefore, is of great importance for the development of smart contracts.

Author contributions PS is the main author of this paper, who has conceived the idea and discussed it with all co-authors. PS has developed the architecture of the work and write up of this work. RJ and MD has supervised the entire work, evaluated the performance, and proofread the paper.

Funding None.

References

1. Buterin, V.: A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper/. Accessed 19 July 2021
2. Peterson, K., Deeduvanu, R., Kanjamala, P., Boles, K.: A blockchain-based approach to health information exchange networks. In: Proceedings of the NIST Workshop Blockchain Healthcare, vol. 1, pp. 1–10 (2016)
3. Korpela, K., Hallikas, J., Dahlberg, T.: Digital supply chain transformation toward blockchain integration In: Proceedings of

the 50th Hawaii International Conference on System Sciences (2017)
4. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In: Proceedings of the Europe and MENA Cooperation Advances in Information and Communication Technologies, pp. 523–533. Springer (2017)
5. Ruta, M., Scioscia, F., Ieva, S., Capurso, G., Pinto, A., Sciascio, E.D.: A blockchain infrastructure for the semantic web of things. In: Proceedings of the 26th Italian Symposium on Advanced Database Systems (SEBD 2018) (2018)
6. Liang, W., Ji, N.: Privacy challenges of IoT-based blockchain: a systematic review. Clust. Comput. (2021). https://doi.org/10.1007/s10586-021-03260-0
7. Al-Bassam, M.: Scpki: A smart contract-based pki and identity system. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, pp. 35–40. ACM (2017)
8. Lemieux, V.L.: Blockchain and distributed ledgers as trusted recordkeeping systems. In: Proceedings of the Future Technologies Conference (FTC), vol. 2017 (2017)
9. Gong, J., Navimipour, N.J.: An in-depth and systematic literature review on the blockchain-based approaches for cloud computing. Clust. Comput. (2021). https://doi.org/10.1007/s10586-021-03412-2
10. Sharma, P., Jindal, R., Borah, M.D.: A review of blockchain-based applications and challenges. Wirel. Pers. Commun. (2021). https://doi.org/10.1007/s11277-021-09176-7
11. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on ethereum smart contracts (sok). In: Principles of Security and Trust, pp. 164–186. Springer, New York (2017)
12. Liebenau, J., Elaluf-Calderwood, S.M.: Blockchain innovation beyond bitcoin and banking. In: Lindell, A.Y. (ed.) Legally-Enforceable Fairness in Secure Two-Party Computation Topics in Cryptology–CT-RSA, pp. 121–137. Springer, New York (2008)
13. Cachin, C.: Architecture of the Hyperledger blockchain Fabric. In: Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers (2016)
14. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications: a review and open problems. In: Proceedings of the 2001 workshop on New security paradigms, pp. 13–22 (2001)
15. Zgraggen, R.R.: Cyber Security Supervision in the insurance sector: smart contracts and chosen issues. In: Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). https://doi.org/10.1109/cybersecpods.2019.8885404
16. Aleksieva, V., Valchanov, H., Huliyan, A.: Application of smart contracts based on ethereum blockchain for the purpose of insurance services. In: Proceedings of the 2019 International Conference on Biomedical Innovations and Applications (BIA). https://doi.org/10.1109/bia48344.2019.8967468
17. Bechtsis, D., Tsolakis, N., Bizakis, A., Vlachos, D.: A blockchain framework for containerized food supply chains. In: Proceedings of the 29th European Symposium on Computer Aided Process Engineering, pp. 1369–1374 (2019). https://doi.org/10.1016/b978-0-12-818634-3.50229-0
18. Zhang, Y., Wen, J.: An IoT electric business model based on the protocol of bitcoin. In: Proceedings of the 18th International Conference on Intelligence in Next Generation Networks (ICIN), pp. 184–191 (2015)
19. Alblooshi, M., Salah, K., Alhammadi, Y.: Blockchain-based ownership management for medical IoT (MIoT) devices. Int. Conf. Innov. Inf. Technol. (IIT) (2018). https://doi.org/10.1109/innovations.2018.8606032
20. Puri, V., Priyadarshini, I., Kumar, R., Le, C.V.: Smart contract based policies for the Internet of Things. Clust. Comput. **24**, 1675–1694 (2021). https://doi.org/10.1007/s10586-020-03216-w
21. Kumar, R., Marchang, N., Tripathi, R.: Distributed off-chain storage of patient diagnostic reports in healthcare system using ipfs and blockchain. Int. Conf. Commun. Syst. Netw. (COMSNETS) (2020). https://doi.org/10.1109/comsnets48256.2020.9027313
22. Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M., Abid, M.: HealthBlock: a secure blockchain-based healthcare data management system. Comput. Netw. **200**, 108500 (2021). https://doi.org/10.1016/j.comnet.2021.108500
23. Ma, Z., Zhu, L., Yu, F.R., James, J.: Protection of surveillance recordings via blockchain-assisted multimedia security. Int. J. Sens. Netw. **32**(2), 69–80 (2021)
24. Dong, C., Wang, Y., Aldweesh, A., McCorry, P., van Moorsel, A.: Betrayal, distrust, and rationality: Smart counter-collusion contracts for verifiable cloud computing. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 211–227. ACM (2017)
25. Uriarte, R.B., Tiezzi, F., De Nicola, R.: Dynamic slas for clouds. In: Proceedings of the European Conference on Service-Oriented and Cloud Computing, pp. 34–49 (2016)
26. Wang, H., Wang, X.A., Xiao, S., Zhou, Z.: Blockchain-based public auditing scheme for shared data. In: Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 197–206. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22263-5_19
27. Hasan, S.S., Sultan, N.H., Barbhuiya, F.A.: Cloud data provenance using IPFS and blockchain technology. In: Proceedings of the 7th International Workshop on Security in Cloud Computing, pp. 5–12. (2019). https://doi.org/10.1145/3327962.3331457
28. Jingyi, L., Wu, J., Chen, L., Li, J.: Deduplication with blockchain for secure cloud storage. In: Proceedings of the CCF Conference on Big Data, pp. 558–570. Springer (2018). https://doi.org/10.1007/978-981-13-2922-7_36
29. Shobana, G., Suguna, M.: Block Chain technology towards identity management in health care application. In: Proceedings of the 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) (2019). https://doi.org/10.1109/i-smac47947.2019.9032472
30. Yang, W., Chen, J., Chen, Y.: An electronic medical record management system based on smart contracts. In: Proceedings of the 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media), pp. 220–223. Bali, Indonesia (2019). https://doi.org/10.1109/Ubi-Media.2019.00050
31. Tikhomirov, S.: Ethereum: state of knowledge and research perspectives. In: Proceedings of the 10th International Symposium on Foundations & Practice of Security (2017)
32. Pontiveros, B.B.F., Norvill, R., State, R.: Recycling smart contracts: compression of the ethereum blockchain. In: Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, pp. 1–5. (2018). https://doi.org/10.1109/NTMS.2018.8328742
33. Luu, L., Chu, D.-H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16 (2016). https://doi.org/10.1145/2976749.2978309
34. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 17–30 (2016)
35. Porru, S., Pinna, A., Marchesi, M., Tonelli, R.: Blockchain-oriented software engineering: challenges and new directions. In: Proceedings of the 2017 IEEE/ACM 39th International

Conference on Software Engineering (ICSE), Buenos Aires, pp. 169–171. (2017). https://doi.org/10.1109/ICSE-C.2017.142
36. Zhou, Y., Kumar, D., Bakshi, S., Mason, J., Miller, A., Bailey, M.: Erays: reverse engineering ethereum's opaque smart contracts. In: Proceedings of the27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 1371–1385 (2018)
37. Coblenz, M.: Obsidian: a safer blockchain programming language. In: Proceedings of the 39th International Conference on Software Engineering Companion, ICSE-C '17, pp. 97–99 (2017)
38. Mavridou, A., Laszka, A.: Tool demonstration: Fsolidm for designing secure ethereum smart contracts. In: Proceedings of the International Conference on Principles of Security and Trust, pp. 270–277. Springer (2018)
39. Liu, H., Liu, Z., Cao, Z., Chen, Z., Chen, B., Roscoe, B.: Reguard: finding reentrancy bugs in smart contracts. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, pp. 65–68. ACM (2018)
40. Luu, L., Chu, D.-H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 254–269 (2016)
41. Albert, E., Gordillo, P., Livshits, B., Rubio, A., Sergey, I.: Ethir: a framework for high-level analysis of ethereum bytecode. In: Proceedings of the International Symposium on Automated Technology for Verification and Analysis, Springer, pp. 513–520 (2018)
42. Knecht, M., Stiller, B.: Smartdemap: a smart contract deployment and management platform. In: Proceedings of the International Conference on Autonomous Infrastructure, Management and Security, p. 15 (2017)
43. Charlier, S., Lagraa, R., State, Francois, J.: Profiling smart contracts interactions tensor decomposition and graph mining. In: Proceedings of the Second Workshop on MIning DAta for financial applications (MIDAS 2017), pp. 31–42 (2017)
44. Frowis, M., Bohme, R.: In code we trust? In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 357–372. Springer, New York (2017)
45. Dickerson, T., Gazzillo, P., Herlihy, M., Koskinen, E.: Adding concurrency to smart contracts. In: Proceedings of the ACM Symposium on Principles of Distributed Computing, pp. 303–312. PODC (2017)
46. Sergey, I., Hobor, A.: A concurrent perspective on smart contracts. In: Proceedings of the International Conference on Financial Cryptography and Data Security (2017)
47. Bracha, G., Ungar, D.: Mirrors: design principles for meta-level facilities of object-oriented programming languages. ACM SIGPLAN Not. **39**, 331–344 (2004)
48. Parjuangan, S., Suhardi: Systematic literature review of blockchain based smart contracts platforms. In: Proceedings of the 2020 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 381–386. Bandung - Padang, Indonesia (2020). https://doi.org/10.1109/ICITSI50517.2020.9264908
49. Rouhani, S., Butterworth, L., Dimmond, A.D., Humphery, D.G., Deters, R.: Medichaintm: a secure decentralized medical data asset management system. In: Proceedings of the 2018 IEEE Conference on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics, pp. 14757–14767. IEEE (2018)
50. Mikula, T., Jacobsen, R.H.: Identity and access management with blockchain in electronic healthcare records. In: Proceedings of the 2018 21st Euromicro Conference on Digital System Design (DSD), pp. 699–706. IEEE (2018)
51. Hossein, K.M., Esmaeili, M.E., Dargahi, T., khonsari, A.: Blockchain-based privacy-preserving healthcare architecture. In: Proceedings of the IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), pp. 1–4. Edmonton, AB, Canada (2019). https://doi.org/10.1109/CCECE.2019.8861857
52. Sharma, P., Jindal, R., Borah, M.D.: Healthify: A blockchain-based distributed application for health care. In: Namasudra, S., Deka, G.C. (eds.) Applications of Blockchain in Healthcare Studies in Big Data. Springer, Singapore (2021)
53. Sharma, P., Jindal, R., Borah, M.D.: Improving security of medical big data by using Blockchain technology. Comput. Electr. Eng. **96**, 1–14 (2021)
54. Chen, S., Shi, R., Ren, Z., Yan, J., Shi, Y., Zhang, J.: A blockchain-based supply chain quality management framework In: Proceedings of the 14th International Conference on e-Business Engineering (ICEBE), pp. 172–176. IEEE (2017)
55. Alazab, M., Alhyari, S., Awajan, A., Abdallah, A.B.: Blockchain technology in supply chain management: an empirical study of the factors affecting user adoption/acceptance. Clust. Comput. **24**, 83–101 (2021). https://doi.org/10.1007/s10586-020-03200-4
56. Huh, S., Cho, S., Kim, S.: Managing IoT devices using blockchain platform. In: Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT), 2017, pp. 464–467. IEEE (2017)
57. Yasin, A., Liu, L.: An online identity and smart contract management system. In: Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 192–198. IEEE (2016)
58. DeCusatis, C., Zimmermann, M., Sager, A.: Identity-based network security for commercial blockchain services. In: Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 474–477. IEEE (2018)
59. Bendiab, K., Kolokotronis, N., Shiaeles, S., Boucherkha, S.: Wip: a novel blockchain-based trust model for cloud identity management. In: Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp. 724–729. IEEE (2018)
60. Lemieux, V.L.: A typology of blockchain recordkeeping solutions and some reflections on their implications for the future of archival preservation. In: Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), pp. 2271–2278. IEEE (2017)
61. Guo, H., Meamari, E., Shen, C.-C.: Blockchain inspired event recording system for autonomous vehicles. In: Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), pp. 218–222. IEEE (2018)
62. Zarrin, J., Phang, H.W., Saheer, L.B., Zarrin, B.: Blockchain for decentralization of internet: prospects, trends, and challenges. Clust. Comput. **24**, 2841–2866 (2021). https://doi.org/10.1007/s10586-021-03301-8
63. Tama, A., Kweka, B.J., Park, Y., Rhee, K.H.: A critical review of blockchain and its current applications. In: Proceedings of the 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), pp. 109–113. IEEE
64. Sankar, L.S., Sindhu, M., Sethumadhavan, M.: Survey of consensus protocols on blockchain applications. In: Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1–5. IEEE (2017)
65. Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W.: Sok: Research perspectives and challenges for

bitcoin and cryptocurrencies,. In:Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP), pp. 104–121. IEEE (2015)
66. Mahmoud, O., Kopp, H., Abdelhamid, A.T., Kargl, F.: Applications of smart-contracts: anonymous decentralized insurances with IoT sensors. In: Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). https://doi.org/10.1109/cybermatics_2018.2018.00254
67. Shae, Z., Tsai, J.J.: On the design of a blockchain platform for clinical trial and precision medicine. In: Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 1972–1980. IEEE (2017)
68. Kumar, T., Ramani, V., Ahmad, I., Braeken, A., Harjula, E., Ylianttila, M.: Blockchain utilization in healthcare: key requirements and challenges. In: Proceedings of the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom) (2018). https://doi.org/10.1109/healthcom.2018.85311
69. Huang, J., Qi, Y.W., Asghar, M.R., Meads, A., Tu, Y.-C.: MedBloc: a blockchain-based secure EHR system for sharing and accessing medical data. In: Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE) (2019). https://doi.org/10.1109/trustcom/bigdatase.2019.00085
70. Pham, H.L., Tran, T.H., Nakashima, Y.: A secure remote healthcare system for hospital using blockchain smart contract. In: Proceedings of the IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, pp. 1–6 (2018). https://doi.org/10.1109/GLOCOMW.2018.8644164
71. Novikov, S.P., Kazakov, O.D., Kulagina, N.A., Azarenko, N.Y.: Blockchain and smart contracts in a decentralized health infrastructure. In: Proceedings of the IEEE International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS) (2018). https://doi.org/10.1109/itmqis.2018.8524970
72. Vishwa, A., Hussain, F.K.: A Blockchain based approach for multimedia privacy protection and provenance. In: Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI) (2018). https://doi.org/10.1109/ssci.2018.8628636
73. Jnoub, N., Klas, W.: Detection of tampered images using blockchain technology. In: Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (2019). https://doi.org/10.1109/bloc.2019.8751300
74. Wang, H., Shi, P., Zhang, Y.: JointCloud: a cross-cloud cooperation architecture for integrated internet service customization. In: Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 1846–1855. (2017). https://doi.org/10.1109/ICDCS.2017.237
75. Scoca, V., Uriarte, R.B., De Nicola, R.: Smart contract negotiation in cloud computing. In: Proceedings of the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), pp. 592–599. IEEE (2017)
76. Sukhodolskiy, I., Zapechnikov, S.: A blockchain-based access control system for cloud storage. In: Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 1575–1578. Moscow (2018). https://doi.org/10.1109/EIConRus.2018.8317400
77. Yu, H., Yang, Z.: Decentralized and smart public auditing for cloud Storage. In: Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 491–494 (2018). https://doi.org/10.1109/ICSESS.2018.8663780
78. Liu, Y., Sun, G., Schuckers, S.: Enabling secure and privacy preserving identity management via smart contract. In: Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS) (2019). https://doi.org/10.1109/cns.2019.8802771
79. Xia, Q., Sifah, E.B., Asamoah, K.O., Gao, J., Du, X., Guizani, M.: Medshare: Trust-less medical data sharing among cloud service providers via blockchain. IEEE Access **5**, 14757–14767 (2017)
80. Cha, S.-C., Chen, J.-F., Su, C., Yeh, K.-H.: A blockchain connected gateway for BLE-based devices in the internet of things. IEEE Access **6**, 24639–24649 (2018)
81. Salah, K., Rehman, M., Nizamuddin, N., AlFuqaha, A.: Blockchain for AI: review and open research challenges. IEEE Access **7**, 10127–10149 (2019)
82. Sharma, P., Jindal, R., Borah, M.D.: A preventive intrusion detection architecture using adaptive blockchain method. In: Patgiri, R., Bandyopadhyay, S., Borah, M.D., Thounaojam, D.M. (eds.) Big Data, Machine Learning, and Applications, BigDML 2019, Communications in Computer and Information Science. Springer, Cham (2020)
83. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., Wang, F.: blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans. Syst. Man Cybernet. **49**(11), 2266–2277 (2019). https://doi.org/10.1109/TSMC.2019.2895123
84. Rouhani, S., Deters, R.: Security, performance, and applications of smart contracts: a Systematic Survey. IEEE Access **7**, 50759–50779 (2019). https://doi.org/10.1109/ACCESS.2019.2911031
85. Liu, J., Liu, Z.: A survey on security verification of blockchain smart contracts. IEEE Access **7**, 77894–77904 (2019). https://doi.org/10.1109/ACCESS.2019.2921624
86. Sayeed, S., Marco-Gisbert, H., Caira, T.: Smart contract: attacks and protections. IEEE Access **8**, 24416–24427 (2020). https://doi.org/10.1109/ACCESS.2020.2970495
87. Khalilov, M.C.K., Levi, A.: A survey on anonymity and privacy in bitcoin-like digital cash systems. IEEE Commun. Surv. Tutor. **20**(3), 2543–2585 (2018)
88. Dhieb, N., Ghazzai, H., Besbes, H., Massoud, Y.: A secure AI-driven architecture for automated insurance systems: fraud detection and risk measurement. IEEE Access **8**, 58546–58558 (2020). https://doi.org/10.1109/ACCESS.2020.2983300
89. Xiong, F., Xiao, R., Ren, W., Zheng, R., Jiang, J.: A key protection scheme based on secret sharing for blockchain-based construction supply chain system. IEEE Access **7**, 126773–126786 (2019). https://doi.org/10.1109/access.2019.2937917
90. Zhang, X., Sun, P., Xu, J., Wang, X., Yu, J., Zhao, Z., Dong, Y.: Blockchain-based safety management system for the grain supply chain. IEEE Access **8**, 36398–36410 (2020). https://doi.org/10.1109/access.2020.2975415
91. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016)
92. Pan, J., Wang, J., Hester, A., Alqerm, I., Liu, Y., Zhao, Y.: EdgeChain: an edge-iot framework and prototype based on blockchain and smart contracts. IEEE Internet Things J. **6**(3), 4719–4732 (2018). https://doi.org/10.1109/jiot.2018.2878154
93. Salah, K., Suliman, A., Husain, Z., Abououf, M., Alblooshi, M.: Monetization of IoT data using smart contracts. IET Netw. (2018). https://doi.org/10.1049/iet-net.2018.5026
94. Zhou, Z., Liao, H., Gu, B., Mumtaz, S., Rodriguez, J.: Resource sharing and task offloading in IoT fog computing: a contract-

learning approach. IEEE Trans. Emerg. Top. Comput. Intell. **4**, 1–14 (2019). https://doi.org/10.1109/tetci.2019.2902869
95. Nguyen, C., Pathirana, P.N., Ding, M., Seneviratne, A.: Blockchain for secure EHRs sharing of mobile cloud based e-health systems. IEEE Access **7**, 66792–66806 (2019). https://doi.org/10.1109/access.2019.2917555
96. Kudumakis, P., Wilmering, T., Sandler, M., Rodriguez-Doncel, V., Boch, L., Delgado, J.: The challenge: from MPEG intellectual property rights ontologies to smart contracts and blockchains [standards in a nutshell]. IEEE Signal Process. Mag. **37**(2), 89–95 (2020). https://doi.org/10.1109/msp.2019.2955207
97. Ghimire, S., Choi, J.Y., Lee, B.: Using blockchain for improved video integrity verification. IEEE Trans. Multimed. **22**(1), 108–121 (2019). https://doi.org/10.1109/tmm.2019.2925961
98. Zhang, Y., Xu, C., Ni, J., Li, H., Shen, X.S.: Blockchain-assisted public-key encryption with keyword search against keyword guessing attacks for cloud storage. IEEE Trans. Cloud Comput. (2019). https://doi.org/10.1109/TCC.2019.2923222
99. Chen, W., Chen, Y., Chen, X., Zheng, Z.: Toward secure data sharing for the IoV: a quality-driven incentive mechanism with on-chain and off-chain guarantees. IEEE Internet Things J. **7**(3), 1625–1640 (2019). https://doi.org/10.1109/JIOT.2019.2946611
100. Cruz, J.P., Kaji, Y., Yanai, N.: RBAC-SC: role-based access control using smart contract. IEEE Access **6**, 12240–12251 (2018). https://doi.org/10.1109/ACCESS.2018.2812844
101. Wang, S., Wang, X., Zhang, Y.: A secure cloud storage framework with access control based on blockchain. IEEE Access **7**, 112713–112725 (2019). https://doi.org/10.1109/ACCESS.2019.2929205
102. Soner, S., Litoriya, R., Pandey, P.: Exploring blockchain and smart contract technology for reliable and secure land registration and record management. Wirel. Person. Commun (2021)
103. Li, H., Han, D.: EduRSS: a blockchain-based educational records secure storage and sharing scheme. IEEE Access **7**, 179273–179289 (2019). https://doi.org/10.1109/ACCESS.2019.2956157
104. Kim, T., et al.: A privacy preserving distributed ledger framework for global human resource record management: the blockchain aspect. IEEE Access **8**, 96455–96467 (2020). https://doi.org/10.1109/ACCESS.2020.2995481
105. Paik, H., Xu, X., Bandara, H.M.N.D., Lee, S.U., Lo, S.K.: Analysis of data management in blockchain-based systems: from architecture to governance. IEEE Access **7**, 186091–186107 (2019). https://doi.org/10.1109/ACCESS.2019.2961404
106. Lattner, C., Adve, V.: LLVM: a compilation framework for lifelong program analysis & transformation. In: Proceedings of the International Symposium on Code generation and Optimization: Feedback-Directed and Runtime Optimization, IEEE Computer Society, p. 75 (2004)
107. Gupta, R., Tanwar, S., Al-Turjman, F., Italiya, P., Nauman, A., Kim, S.W.: Smart contract privacy protection using AI in cyber-physical systems: tools, techniques and challenges. IEEE Access **8**, 24746–24772 (2020). https://doi.org/10.1109/ACCESS.2020.2970576
108. Hartel, P., Homoliak, I., Reijsbergen, D.: An empirical study into the success of listed smart contracts in ethereum. IEEE Access **7**, 177539–177555 (2019). https://doi.org/10.1109/ACCESS.2019.2957284
109. Xia, Q., Sifah, E.B., Smahi, A., Amofa, S., Zhang, X.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments. Information **8**(2), 44 (2017)
110. Yue, X., Wang, H., Jin, D., Li, M., Jiang, W.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. **40**(10), 218 (2016)
111. Nugent, T., Upton, D., Cimpoesu, M.: Improving data transparency in clinical trials using blockchain smart contracts. F1000Res **5**, 2541 (2016)
112. Kim, H.M., Laskowski, M.: Toward an ontologydriven blockchain design for supply-chain provenance. Intell. Syst. Account. Financ. Manag. **25**(1), 18–27 (2018)
113. Figorilli, S., Antonucci, F., Costa, C., Pallottino, F., Raso, L., Castiglione, M., Pinci, E., Del Vecchio, D., Colle, G., Proto, A., et al.: A blockchain implementation prototype for the electronic open source traceability of wood along the whole supply chain. Sensors **18**(9), 3133 (2018)
114. Ouaddah, A., Abou-Elkalam, A., Ait-Ouahman, A.: Fairaccess: a new blockchain-based access control framework for the internet of things. Secur. Commun. Netw. **9**(18), 5943–5964 (2016)
115. Lemieux, V.L.: Trusting records: is blockchain technology the answer? Rec. Manag. J. **26**(2), 110–139 (2016)
116. Sharma, P., Jindal, R., Borah, M.D.: Blockchain technology for cloud storage: a systematic literature review. ACM Comput. Surv. **53**(4), 1–32 (2020)
117. Kuo, T.T., Kim, H.E., Ohno-Machado, L.: Blockchain distributed ledger technologies for biomedical and health care applications. J. Am. Med. Inform. Assoc. **24**(6), 1211–1220 (2017)
118. Hölbl, M., Kompara, M., Kamišalic, A., Nemec-Zlatolas, L.: A systematic review of the use of blockchain in healthcare. Symmetry **10**(10), 470 (2018)
119. Panarello, A., Tapas, N., Merlino, G., Longo, F., Puliafito, A.: Blockchain and IoT integration: a systematic survey. Sensors **18**(8), 2575 (2018)
120. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: beyond bitcoin. Appl. Innov. **2**, 6–10 (2016)
121. Zhao, J.L., Fan, S., Yan, J.: Overview of business innovations and research opportunities in blockchain and introduction to the special issue. Financ. Innov. **2**(1), 28 (2016)
122. Bocek, T., Stiller, B.: Smart Contracts: Blockchains in the Wings, pp. 169–184. Springer, Berlin (2018)
123. Tapscott, A., Tapscott, D.: How blockchain is changing finance. Harvard Bus. Rev. (2017)
124. Chen, L., Deng, Y.Y., Tsaur, W.J., Li, C.T., Lee, C.C., Wu, C.M.A.: A traceable online insurance claims system based on blockchain and smart contract technology. Sustainability (2021). https://doi.org/10.3390/su13169386
125. Wang, L., et al.: Smart contract-based agricultural food supply chain traceability. IEEE Access **9**, 9296–9307 (2021). https://doi.org/10.1109/ACCESS.2021.3050112
126. Pranto, T.H., Noman, A.A., Mahmud, A., Bahalul Haque, A.K.M.: Blockchain and smart contract for IoT enabled smart agriculture. PeerJ Comput. Sci. (2021). https://doi.org/10.7717/peerj-cs.407
127. Tripathi, G., Ahad, M.A., Paiva, S.: S2HS: a blockchain based approach for smart healthcare system. Healthcare **8**, 100391 (2019). https://doi.org/10.1016/j.hjdsi.2019.100391
128. Lakhan, A., Mohammed, M.A., Rashid, A.N., Kadry, S., Panityakul, T., Abdulkareem, K.H., Thinnukool, O.: Smart-Contract aware ethereum and client-fog-cloud healthcare system. Sensors **21**(12), 4093 (2021). https://doi.org/10.3390/s21124093
129. Sharma, P., Jindal, R., Borah, M.D.: Blockchain-based decentralized architecture for cloud storage system. J. Inf. Secur. Appl. **62**, 1–15 (2021)
130. Chen, Y., Ding, S., Xu, Z., Zheng, H., Yang, S.: Blockchain-based medical records secure storage and medical service framework. J. Med. Syst. **43**(1), 5 (2019). https://doi.org/10.1007/s10916-018-1121-4

131. Bouras, A., Lu, Q., Dhelim, S., Ning, H.: A lightweight blockchain-based IoT identity management approach. Future Internet **13**(2), 24 (2021). https://doi.org/10.3390/fi13020024
132. Niu, J., Ren, Z.: A self sovereign identity management scheme using smart contracts. In: Proceedings of the 2nd International Conference on Computer Science Communication and Network Security, vol. 336 (2021)
133. Wang, S., Pei, R., Zhang, Y.: EIDM: a ethereum-based cloud user identity management protocol. IEEE Access **7**, 115281–115291 (2019). https://doi.org/10.1109/ACCESS.2019.2933989
134. Kasampalis, T., Guth, D., Moore, B., Serbanuta, T., Serbanuta, V., Filaretti, D., Rosu, G., Johnson, R.: IELE: an intermediate-level blockchain language designed and implemented using formal semantics, Tech. Rep. (2018)
135. Zhang, P., White, J., Schmidt, D.C., Lenz, G., Rosenbloom, S.T.: Fhirchain: applying blockchain to securely and scalably share clinical data. Comput. Struct. Biotechnol. J. **16**, 267–278 (2018)
136. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Comput. Sci. Rev. **30**, 80–86 (2018)
137. Casino, F., Dasaklis, T.K., Patsakis, C.: A systematic literature review of blockchain-based applications: current status, classification and open issues. Telemat. Inform. **36**, 55–81 (2018)
138. Macrinici, A., Cartofeanu, C., Gao, S.: Smart contract applications within blockchain technology: a systematic mapping study. Telemat. Inform. **35**(8), 2337–2354 (2018)
139. Zheng, Z., Xie, S., Dai, H.-N., et al.: An overview on smart contracts: challenges, advances and platforms. Future Gener. Comput. Syst. **105**, 475–491 (2019)
140. Leng, K., Bi, Y., Jing, L., Fu, H.C., Van Nieuwenhuyse, I.: Research on agricultural supply chain system with double chain architecture based on blockchain technology. Futur. Gener. Comput. Syst. **86**, 641–649 (2018). https://doi.org/10.1016/j.future.2018.04.061
141. Helo, P., Hao, Y.: Blockchains in operations and supply chains: a model and reference implementation. Comput. Ind. Eng. **136**, 242–251 (2019). https://doi.org/10.1016/j.cie.2019.07.023
142. Yadav, V.S., Singh, A.R., Raut, R.D., Govindarajan, U.H.: Blockchain technology adoption barriers in the Indian agricultural supply chain: an integrated approach. Resour. Conserv. Recycl. **161**, 104877 (2020). https://doi.org/10.1016/j.resconrec.2020.104877
143. Dwivedi, S.K., Amin, R., Vollala, S.: Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism. J. Inf. Secur. Appl. **54**, 102554 (2020). https://doi.org/10.1016/j.jisa.2020.102554
144. Habeeb, R.A.A., Nasaruddin, F., Gani, A., Hashem, I.A.T., Ahmed, E., Imran, M.: Real-time big data processing for anomaly detection: a survey. Int. J. Inf. Manag. **45**, 289–307 (2018)
145. Khattak, A., Shah, M.A., Khan, S., Ali, I., Imran, M.: Perception layer security in internet of things. Future Gener. Comput. Syst. **100**, 144–164 (2019)
146. Yaqoob, I., Ahmed, E., Ur Rehman, M.H., Ahmed, A.I.A., Algaradi, M.A., Imran, M., Guizani, M.: The rise of ransomware and emerging security challenges in the internet of things. Comput. Netw. **129**, 444–458 (2017)
147. Miorandi, D., Sicari, S., Pellegrini, F.D., Chlamtac, I.: Internet of things: vision, applications and research challenges. Ad Hoc Netw. **10**(7), 1497–1516 (2012)
148. Fan, K., Bao, Z., Liu, M., Vasilakos, A.V., Shi, W.: Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial IoT. Future Gener. Comput. Syst. **110**, 665–674 (2019). https://doi.org/10.1016/j.future.2019.10.014
149. Tanwar, S., Parekh, K., Evans, R.: Blockchain-based electronic healthcare record system for healthcare 4.0 applications. J. Inf. Secur. Appl. **50**, 102407 (2020). https://doi.org/10.1016/j.jisa.2019.10240
150. Chalmersa, D., Matthewsb, R., Hyslopb, A.: Blockchain as an external enabler of new venture ideas: digital entrepreneurs and the disintermediation of the global music industry. J. Bus. Res. **125**, 577–591 (2018)
151. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Gener. Comput. Syst. **107**, 841–853 (2017)
152. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf (2008). Accessed 10 June 2019
153. Nick, S.: The idea of smart contracts. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html (1997). Accessed 10 Nov 2019
154. Nick, S.: Formalizing and securing relationship on public networks. http://firstmonday.org/ojs/index.php/fm/article/view/548/469 (1997). Accessed 15 Apr 2020
155. Buterin, V., et al.: Ethereum white paper. https://www.ethereum.org/ (2013)
156. Symbol from Nem: https://nemtech.github.io/catapult-whitepaper/main.pdf (2020)
157. Brown, R.G.: The corda platform: an introduction. https://www.corda.net/content/corda-platform-whitepaper.pdf (2018)
158. Neo white paper: https://docs.neo.org/docs/en-us/basic/whitepaper.html
159. EOS: an introduction. https://www.iang.org/papers/EOS_An_Introduction-BLACK-EDITION.pdf
160. Lerner, S.D.: Rootstock whitepaper. https://docs.rsk.co/RSK_White_Paper-Overview.pdf (2015)
161. Kwon, J., Buchman, E.: Cosmos: a network of distributed ledgers. https://cosmos.network/cosmos-whitepaper.pdf
162. Mazieres, D.: The stellar consensus protocol: a federated model for internet-level consensus. https://www.stellar.org/papers/stellar-consensus-protocol.pdf (2016)
163. Waves: https://docs.kolinplatform.com/development/waves-platform-smart-contracts
164. Cardano. https://cardano.org/why/
165. Quorum enterprise ethereum client. http://docs.goquorum.com/en/latest/
166. Howard, H., Schwarzkopf, M., Madhavapeddy, A., Crowcroft, J.: Raft refloated: do we have consensus?
167. Solidity introduction. https://solidity.readthedocs.io/en/v0.5.11/introduction-to-smart-contracts.html
168. Delmolino, K., Arnett, M., Kosba, A., Miller, A., Shi, E.: Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab (2016)
169. Bellomy, B.: Solidity pitfalls: typecasting and fallback functions: augmenting humanity (2017)
170. Xu, R., Chen, Y., Blasch, E., Chen, G.: Blendcac: a blockchain-enabled decentralized capability based access control for IoTs. https://arxiv.org/abs/1804.09267 (2018)
171. Grüner, A., Mühle, A., Meinel, C.: On the relevance of blockchain in identity management. https://arxiv.org/abs/1807.08136 (2018)
172. D'Angelo, G., Ferretti, S., Marzolla, M.: A blockchain-based flight data recorder for cloud accountability. https://arxiv.org/abs/1806.04544 (2018)
173. Seijas, L., Thompson, S.J., McAdams, D.: Scripting smart contracts for distributed ledger technology. IACR Cryptology ePrint Archive, p. 1156 (2016)

174. Fu, Y., Ren, M., Ma, F., Jiang, Y., Shi, H., Sun, J.: Evmfuzz: differential fuzz testing of ethereum virtual machine. https://arxiv.org/abs/1903.08483 (2019)
175. Anjana, S., Kumari, S., Peri, S., Rathor, S., Somani, A.: An efficient framework for concurrent execution of smart contracts. https://arxiv.org/abs/1809.01326 (2018)
176. Kormiltsyn, A., Udokwu, C., Karu, K., Thangalimodzi, K., Norta, A.: Improving healthcare processes with smart contracts. In: Abramowicz, W., Corchuelo, R. (eds.) Business Information Systems, Lecture Notes in Business Information Processing, vol. 353. Springer, Cham (2019)
177. DuPont, Q.: Experiments in Algorithmic Governance: A History and Ethnography of the DAO, A Failed Decentralized Autonomous Organization", Bitcoin and Beyond: Cryptocurrencies, Blockchains and Global Governance. Routledge, London (2017)
178. Massacci, F., Ngo, C.N., Nie, J., Venturi, D., Williams, J.: The seconomics (security economics) vulnerabilities of decentralized autonomous organizations. In: Stajano, F., Anderson, J., Christianson, B., Matyáš, V. (eds.) Security Protocols XXV Security Protocols 2017 Lecture Notes in Computer Science, vol. 10476. Springer, Cham (2017)
179. Dika, A.: Ethereum smart contracts: security vulnerabilities and security tools. (Masters thesis). NTNU (2017)
180. Pettersson, J., Edström, R.: Safer smart contracts through type-driven development. Master's thesis, Dept. of CS&E, Chalmers University of Technology & University of Gothenburg, Sweden (2015)
181. Morisse, M.: Cryptocurrencies and bitcoin: charting the research landscape. In: Proceedings of the 2015 Americas Conference on Information Systems (AMCIS2015) Puerto Rico
182. Sharma, P., Jindal, R., Borah, M.D.: A preventive intrusion detection architecture using adaptive blockchain method. In: Patgiri, R., Bandyopadhyay, S., Borah, M.D., Thounaojam, D.M. (eds.) Big Data, Machine Learning, and Applications. BigDML 2019. Communications in Computer and Information Science, vol 1317. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62625-9_3
183. Bragagnolo, S., Rocha, H., Denker, M., Ducasse, S.: Smartinspect: solidity smart contract inspector. In: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), pp. 9–18. IEEE (2018)

Pratima Sharma is a Research Scholar, Department of Computer Science & Engineering at Delhi Technological University, India. Prior to this she worked as an Assistant Professor, at Inderprastha Engineering College, Ghaziabad, for nearly three years. She received the M.Tech. and B.Tech. degrees in Computer Science and Engineering from Guru Gobind Singh Indraprastha University, Delhi, India, in 2013 and 2015, respectively. Her research and publication interests include Blockchain Technology, Honeypot, Network Security, Information Security and Data Mining. She has presented papers at International/ National conferences, published articles and papers in various journals.

Rajni Jindal is working as Professor and Head at Computer Engineering Department, Delhi Technological University, Delhi. She received her M.E. from Delhi College of Engineering. She completed her Ph.D. (Computer Engineering) from Faculty of Technology, Delhi University, Delhi. She also worked as Professor (IT), Dean (Research and Collaboration) at Indira Gandhi Delhi Technical University for women, Delhi for 3 years. She possesses a work experience of around 31 years in research and academics. Her major area of interests are Database Systems, Data Mining, Operating Systems and Compiler Design. She has authored around 145 research papers and articles for various national and international journals/conferences and also 5 books. She is a senior member of IEEE and life member of CSI.

Malaya Dutta Borah Dr. Malaya Dutta Borah is working as an Assistant Professor in the Department of Computer Science & Engineering at National Institute of Technology (NIT) Silchar, Assam. Her research areas are Blockchain Technology, Data Mining and Cloud Computing. She has authored/co-authored around 45 research papers in International/National Journals/Conferences/Books of repute. She has published 2 patents. She is the Editor of three upcoming edited books (publishers- CRC press, IGI Global and Springer) in the field of Blockchain technology. She is Treasure of IEEE Silchar subsection. She has experience of organising International conferences and workshops. She is a member of IEEE, CSI, ACM and MIR Lab. Her Online Official Profile is available at: http://cs.nits.ac.in/malaya/.