frontiers | Frontiers in Blockchain

TYPE Original Research
PUBLISHED 28 April 2025
DOI 10.3389/fbloc.2025.1567417

Mitigating cloud vulnerabilities using a blockchain platform

Dina Zoughbi[1]* and Kavitha Venkatachalam[2]

[1]Research Scholar at the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India, [2]Professor at the Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India

OPEN ACCESS

EDITED BY
Mubashir Husain Rehmani,
Munster Technological University, Ireland

REVIEWED BY
Chengzu Dong,
Lingnan University, Hong Kong SAR, China
Houaida Ghanmi,
Université Sorbonne Paris Nord, France

*CORRESPONDENCE
Dina Zoughbi,
✉ dinazoughbi8@gmail.com,
✉ dz0092@srmist.edu.in

RECEIVED 27 January 2025
ACCEPTED 11 March 2025
PUBLISHED 28 April 2025

CITATION
Zoughbi D and Venkatachalam K (2025) Mitigating cloud vulnerabilities using a blockchain platform.
Front. Blockchain 8:1567417.
doi: 10.3389/fbloc.2025.1567417

Content-Based Image Retrieval (CBIR) has become a critical technology for efficiently searching and retrieving images from large datasets based on their visual content. Traditional CBIR systems, which rely on low-level features like color, texture, and shape, often struggle with semantic gaps and scalability issues. With the rapid advancements in deep learning and cloud computing, there is a growing need to enhance CBIR performance for real-world applications. In order to tackle the issues, an enhanced CBIR process leveraging advanced neural networks, particularly Convolutional Neural Networks (CNNs), Siamese Networks, and attention mechanisms is proposed. It integrates multimodal data, including text and audio, to improve retrieval accuracy. Additionally, cloud-based infrastructure is employed to support large-scale image processing, enabling faster retrieval times and real-time performance. Edge computing techniques are also incorporated to reduce latency in applications requiring immediate responses. The proposed model demonstrates significant improvements in retrieval accuracy and efficiency compared to traditional CBIR methods. Deep learning models, particularly CNNs with transfer learning and attention mechanisms, effectively capture high-level semantic features. The integration of cloud infrastructure enhances scalability and real-time processing capabilities, while multimodal retrieval improves search relevance. The use of explainable AI techniques adds transparency to the decision-making process, increasing user trust. Hence, the advanced neural networks, coupled with cloud and edge computing, can significantly optimize CBIR systems, making them more robust, scalable, and applicable to a wide range of industries such as healthcare, security, and e-commerce.

KEYWORDS

cloud security, AWS EC2, DApp (decentralized application), AWSCLI and PuTTY, Metamask and Ganache

1 Introduction

In the peer-to-peer network, blockchain technology has been used for more than a decade as a distributed database record for all types of transactions. Centralized party trust is considered a distributed computing paradigm that effectively controls problems. Without the involvement of a trusted third party, smart contracts are implementable programs that run on top of blockchain to enforce, facilitate and execute agreements between untrustworthy parties. Commonly, the blockchain is another type of database that brings data together into separate groups called blocks (Yu et al., 2020). Within a specific capacity, each block contains groups of information. In a peer-to-peer fashion, the cryptocurrency exchange information available in the bitcoin blocks among nodes included in the network communication. When applied in the network, the blockchain begins an irreversible timetable of data. Once the block is completed, it becomes part of the chain, and it is proven to be correct (Singh et al., 2020). It is attached to the chain when the particular timestamp is given to each block of the chain. The numerical information is converted into a string of letters, and mathematical functions are used to generate the numbers via timestamp hash code generation. The hash code is changed when the information is altered (Taherdoost, 2023). To convert paper contracts into digital contracts, smart contracts provide network ability and automation. Without the supervision of a central authority, traditional contracts are compared with smart contracts to enable users to program their trust and agreement relations by offering automated transactions (Khan S. N. et al., 2021). To manage the data safely based on the P2P method, distributed computing ledger management technology refers to a blockchain that can organize data into blocks that are small sets of data connected in chain form. The data block is connected like a chain, so the blockchain can prevent data tampering (Kemmoe et al., 2020). Without the need for a third party to trust a centralized system, blockchain technologies are useful in different service applications since they enable the protection of P2P and secure data integrity. During the decryption and encryption of data with communication for the fog layer, in the internetwork layer, the secret sharing concept is used for privacy and security (Sookhak et al., 2021).

A new computing model is cloud computing; in the cloud, it offers users with omnipresence services and cost reductions for computing and user storage, and it enhances the comfort for individuals and business organizations in choosing their place to store the data. Cloud security issues are important factors that restrict cloud computing development, but with the development of intensification and cloud computing scales, research on edge computing and fog computing has also progressively increased (Gai et al., 2020). Jun Wu was the one who approved the publication and the associate editor coordinating the manuscript review of "Security" by CSA (Cloud Security Alliance) in 2017. One of the core technologies of cloud security is the cloud computing security focus area, which was identified by the guidance for Critical Areas of Focus in Cloud Computing v4.0. The current research hotspot is access control, and preventing stolen or accessed by illegal users from accessing data stored in the cloud is the main purpose of access control. Access control plays an important role in the cloud to protect related resources through access control from three service systems of cloud computing, such as SaaS (software as a services), IaaS (infrastructure as a service) and PaaS (platform as a service) (Yang et al., 2020). The difficulty of having a centralized trustworthy third party was not addressed, even though the technology of digitization solved the consistency, atomicity and storage problems with the use of a database management system (DBMS). Each with its own compliance mechanism, the number of centralized regulators increases when the cross-border transfers of goods occur. The members of the supply chain feel that their burden increases (Ghazal et al., 2023). The range varies between 8% and 14% according to the projection of the contribution of supply chain costs in developed and developing countries. It offers important value to the industry and consumers because of the global trading and shipping volume, which was estimated to be 2,300 parcels in 2020, with a value of $100 billion shipped globally every second. This will lead to high costs due to the maintenance of the centralized trusted third party (Xie et al., 2020).

Although several studies have discussed cloud computing, smart contracts and blockchain separately, few studies have integrated cloud computing, smart contracts and blocks. Therefore, the proposed study used the truffle configuration in the smart contract of blockchain, DApp (Decentralized Application) for solidity, for configuration puTTY, to perform data transfer (Metamask and Ganache) and deployment using AWS EC2. Even cloud computing is good for data management, but the security risk is high. To overcome this issue, blockchain is integrated into cloud computing, and after the process is completed, it will be displayed in the cloud.

1.1 Motivation

In confidential data transfer, decentralized decision making, anonymity and integrity protection, the application of blockchain is needed in logistics, financial sectors and healthcare. Cloud computing is also very popular for database management. However, many existing studies have shown that the main drawback of cloud computing is its security while storing or transferring user details. To overcome these issues, current research proposed a blockchain and cloud computing framework to demonstrate private and secured data transfer. For security purposes, the blockchain is integrated with cloud computing, and the user will feel safe and secure about data transfer without any risk.

1.2 Research contributions

The main objective of the proposed method is as follows:

- To configure the DApp solidity of smart contracts and AWS, EC2 was configured by AWSCLI and PuTTY.
- To test and compile the application after truffle configuration on a smart contract.
- To test and monitor the application after the deployment of the application container in AWS EC2.

1.3 Paper organization

The subsequent paper is divided into four sections. Section 2 contains the prevailing research. Section 3 includes the entire methodology of the complete study. Section 4 includes the overall results and analysis of the proposed model. Section 5 includes the conclusion, limitations and future work.

2 Literature review

Existing research (Awadallah et al., 2021) has implemented Byzantine fault tolerance to construct a distributed processing network that combines blockchain with CC (cloud computing). The procedure achieved average performance in terms of the client requirements. Similarly (Murthy et al., 2020), recommended integrating blockchain with CC for data management, usability, security, trustworthiness and scalability to achieve better performance. Correspondingly, previous studies (Awadallah and Samsudin, 2021) have implemented a secure BC-based RDB and an agile BC-based RDB to enhance the cloud RDB (relational database). The average performance in terms of user specifications was attained for the recommended approach. Similarly, the prevailing research (Sun and Yu, 2020) demonstrated the use of a formal verification framework to identify security issues in smart contracts. The existing research has achieved average performance in rectifying security issues in smart contracts. The existing research (Sharma et al., 2021) has implemented the architecture of blockchain-based decentralization to provide a highly secure environment. The average performance of the recommended architecture was attained. Correspondingly, the prevailing research (Chen et al., 2020) implemented the cascaded depth learning framework to improve the classification and feature selection in each layer of the model. The recommended research attained the average classification accuracy and data representation features. Similarly, the suggested research (Dolgui et al., 2020) implemented the model "virtual operation". The model has been developed with multiple logistics service providers. The design of smart contracts involves an event-driven dynamic approach to service and task composition. The developed model has been used for designing and controlling smart contracts. The model attained average performance. Correspondingly, the existing research (Sultana et al., 2020) recommended that blockchain-based access control and data sharing be developed for communication among IoT (Internet of Things) devices. The existing approach has been used to overcome the issues of data sharing in the IoT in terms of authorization, trust and authentication. To offer effective access control management, multiple smart contracts, such as the JC (Judge Contract), ACC (Access Control Contract) and RC (Register Contract), have been used. The recommended access control has achieved better performance. Literally, the prevailing research (Wang et al., 2020) has implemented blockchain-based smart contracts for fair payments for public cloud storage auditing. In the existing model, the CSP (cloud service provider) and data owner process the mentioned smart contract. The data possession proof has been regularly submitted by the CSP, and it has been ensured by the contract; otherwise, it has to pay the penalties without remuneration, and the model has achieved better performance. Similarly, the considered research (De Giovanni, 2020) demonstrated blockchain for a supply chain game with two firms (a retailer and a supplier). Firms have shifted to blockchain because of the high transaction costs and risk in terms of services and delivery. Then, the smart contract model is included for an effective process.

The prevailing research (Khatoon, 2020) has implemented blockchain-based smart contracts with the internet of Medical Things (IoMT) in e-healthcare. The existing research has achieved better performance in terms of the average energy, average latency and average packet delivery ratio. Similarly, the suggested research (Kumar et al., 2022) demonstrated a framework for secured cloud-based manufacturing operations with smart contracts based on the ERC20 interface in blockchain. The recommended research has achieved better performance by overcoming the critical loopholes in traditional supply chains. Correspondingly, the prevailing research (Wang et al., 2022) has implemented the DCEP (digital currency electronic payment), which is offered by a cloud service platform based on a blockchain smart contract for an incorporated energy (carbon) market. Decentralized intelligent dispatch has been implemented by utilizing blockchain IoT technology. The existing approach has achieved better efficacy and security in energy trading. Literally, the considered research (Tan et al., 2022) implemented the Service Level Agreement Model by Integrating Blockchain Smart Contract (SLABSC) to monitor the third party. Through blockchain technologies, trust issues between third parties, CSPs and CSCs have been addressed. The proposed method achieved better performance. Correspondingly, existing research (Saini et al., 2020) has implemented the smart contract framework to build access control in electronic medical records (EMRs). The recommended research approached smart contracts in four ways: access revocation, access authorization, user verification and misbehavior detection. The method has achieved average performance in smart healthcare systems. Similarly, the considered research (Kochovski et al., 2020) implemented the novel SC (smart contract)-based framework that has been offered SLA management between related actors and entities in an environment of decentralized computing, such as cloud providers, VMs (virtual machines) and cloud service consumers. The recommended framework has been implemented on a testnet (Ethereum ledger). The average performance of the framework was attained. Similarly, existing research (Muneeb et al., 2021) has implemented blockchain-based frameworks, such as the TBlockchain and SBlockchain. This framework has been used to store smart contracts, and data have been generated by smart contracts. This framework has achieved better performance. Similarly, the prevailing research (Uriarte et al., 2021) has recommended the use of an SLA management framework to improve trustworthiness. The framework depends on a two-level blockchain architecture in which smart SLA and permissioned blockchain have been used for dynamic service guidance and to generate objective measurements. The architecture achieved better performance (Chaganti et al., 2022). implemented a cloud-enabled smart farm security monitoring framework by analyzing behavioral patterns. The average performance of the framework was attained. Literally, the prevailing research (Khan A. A. et al., 2021) demonstrated the framework of secure blockchain awareness used for distributed data monitoring and management. The consortium P2P (peer-to-peer) and SH-256 (secure hash-encrypted) networks have been used to analyze changes in blockchain storage. The framework has achieved better performance.

Wei et al. (2020) implemented a distributed virtual agent model via mobile agent technology to improve trust computing and data security. The blockchain-based integrity verification scheme has been constructed by "block and response" to avoid data tampering. Similarly, the suggested research (Zhang et al., 2021) demonstrated a data auditing scheme of blockchain-based multicloud storage to secure accurate arbitrate service disputes and data integrity. The scheme has also been used with smart contracts to identify service disputes. The scheme achieved average performance. The existing research (Zghaibeh et al., 2020) has implemented a blockchain-based healthcare system called SHealth (Smart-Health). The recommendation system

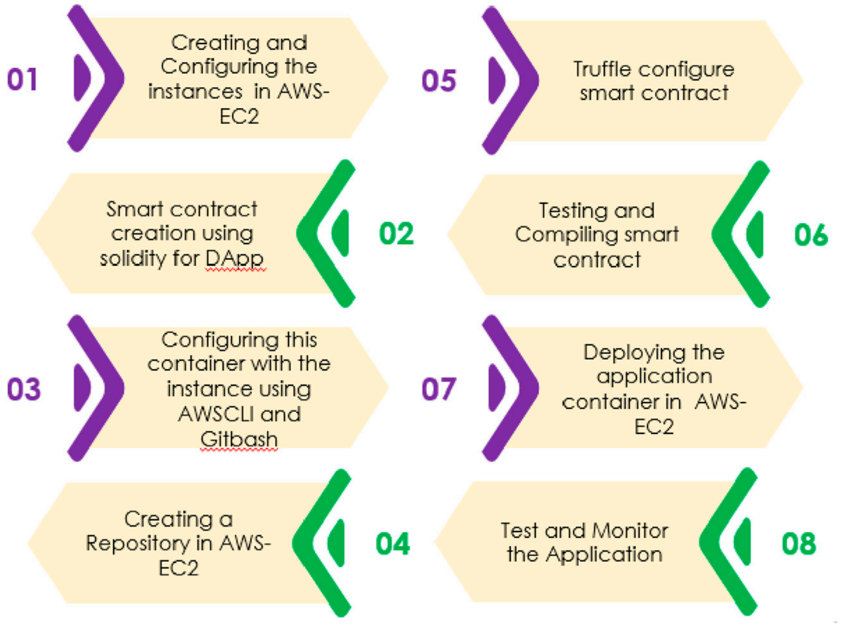

01
Creating and Configuring the instances in AWS-EC2
02
Smart contract creation using solidity for DApp
03
Configuring this container with the instance using AWSCLI and Gitbash
04
Creating a Repository in AWS-EC2
05
Truffle configure smart contract
06
Testing and Compiling smart contract
07
Deploying the application container in AWS-EC2
08
Test and Monitor the Application

FIGURE 1
Overall working process of the proposed method.

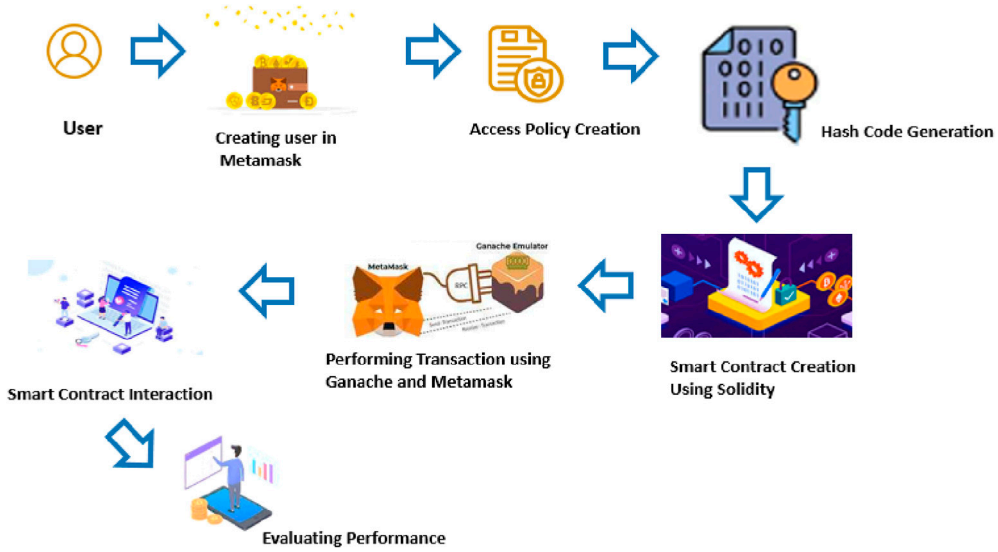

User
Creating user in Metamask
Access Policy Creation
Hash Code Generation
Smart Contract Creation Using Solidity
Performing Transaction using Ganache and Metamask
Smart Contract Interaction
Evaluating Performance

FIGURE 2
Architecture of the proposed model.

ensures resistance, security and availability from malicious attacks and tampering. Without compromising its authenticity, all stakeholders were able to access the distributed database. The system achieved better performance. Correspondingly (Sharma et al., 2020), recommended blockchain-based smart contracts in e-healthcare by the IoMT. The framework achieved average performance in terms of energy efficiency and latency. Similarly, the prevailing research (Sigalov et al., 2021) has implemented blockchain-based smart contracts BIM (building information modeling). The payment is made automatically through an authenticated financial institution once the contracted construction has been accepted by the client. The model attained average performance. Similarly, the existing research (Lin et al., 2021) implemented the OBEP (Optimized Blockchain-based Fair Payment) for secure payment transactions. Blockchain-based ZKP-free solutions can be combined with any protected accumulator,

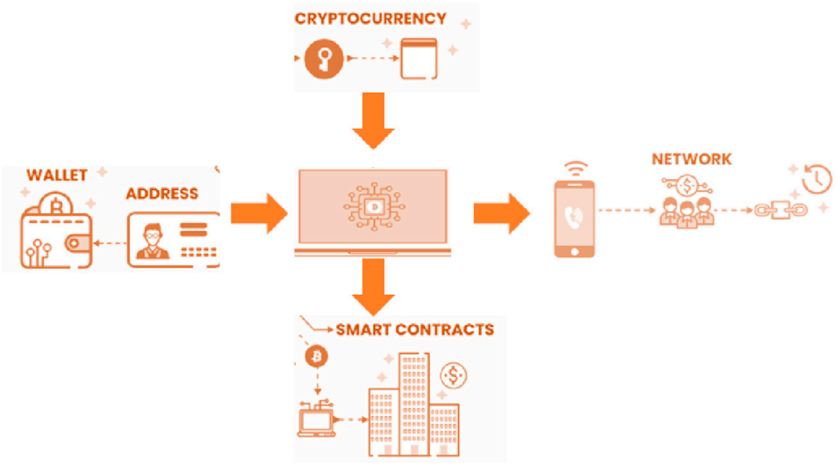


FIGURE 3
Testing and compiling on blockchain.

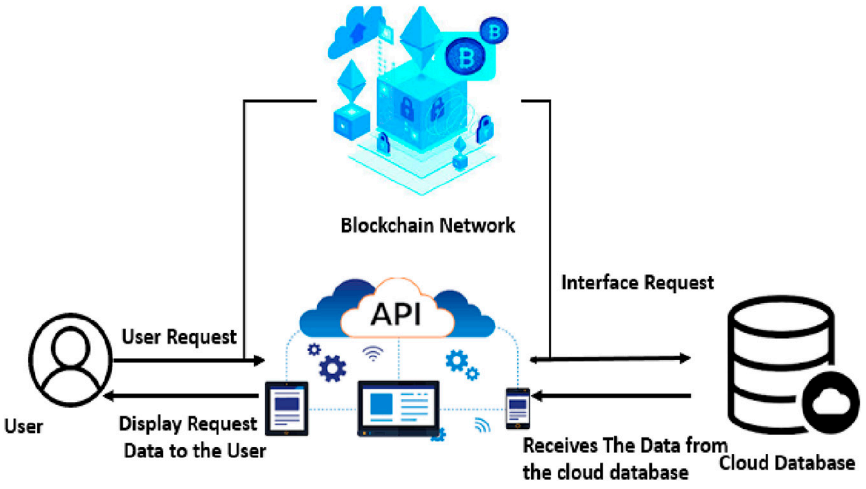


FIGURE 4
Testing and compiling of blockchain in the cloud.

symmetric encryption scheme or commitment. The model achieved better performance. Similarly, the suggested research (Fan et al., 2020) demonstrated the Ethereum of novel decentralized auditing smart contracts. The third party auditor (TPA) has been replaced with decentralized auditing system (Dredas) because anyone can obtain an auditing result without worrying about security. The average computational costs of the recommended model were determined. Literally, the prevailing research (Leduc et al., 2021) has implemented a novel blockchain farming marketplace platform to support trading between farmers and third-party stakeholders for agricultural goods. Correspondingly, the considered research (Qin et al., 2021) implemented the BMAC (blockchain-based multiauthority access control) for secured data transactions. To avoid single point failure, permissioned blockchain and the Shamir secret sharing scheme were used. The model attained average performance. Similarly, the prevailing research (Eltayieb et al., 2020) has implemented attribute-based signcryption with blockchain for secure data transactions. The recommended schemes satisfy the security needs of cloud computing, such as unforgeability and confidentiality. The scheme achieved average performance.

2.1 Problem identification

- The construction of Byzantine Fault Tolerance for the process of a distributed network with the integration of blockchain with CC. The performance in terms of client requirements should be improved (Awadallah et al., 2021).
- The secure BC-based RDB and the agile BC-based RDB enhance the cloud RDB (relational database), and they achieve average performance in terms of user specifications. However, the system was very stressful and difficult (Awadallah and Samsudin, 2021).

- The formal verification framework used to identify security issues in smart contracts achieved average performance. The framework of formal verification should improve the performance of smart contracts (Sun and Yu, 2020).

3 Proposed methodology

In every field, the data security and privacy of users are the main concerns of many companies or organizations. High-level organizations introduced their own cloud platform and asked their customers to store long-term data. Cloud platforms are helpful for managing large amounts of data, but security risks threaten users. To overcome the weakness of the cloud, blockchain technology is included to secure user data with the help of smart contracts. The overall flow of the proposed model is shown in Figure 1.

Figure 2 shows the Architecture of the proposed model. The initial step is to create Amazon Web Server (AWS) EC2. Then, inside the blockchain, the smart contract was created with the help of solidity for the DApp. The DApp solidity container was configured with the AWS EC2 account by PuTTY and AWSCLI, and within the AWS EC2, the repository was created. To configure the smart contract, the truffle is used, and then the testing and compiling stage of the smart contract is initiated. After the testing and compiling process, the container of the application is deployed in the AWS EC2. In the final stage, the deployed application is tested and monitored and then displayed in the cloud environment in a secure way because of the integration of the blockchain.

3.1 Architecture of the proposed model

First, with the help of Metamask, a user account is created. The user should acknowledge the terms and conditions of the access policy. The hash code is generated for storing the values of user information or their request. With the use of solidity, a smart contract is created, and the data transfer process is performed by Metamask and Ganache. To interact with the Ethereum blockchain, Metamask is used, which is a software package for cryptocurrency wallets. To set up a personal Ethereum Blockchain to organize contracts, run tests and develop the applications attainable by Ganache, it is a development tool in the truffle suite. After this process, the interaction of smart contracts occurs, and the performance of the smart contract is evaluated.

3.2 Testing and compiling on blockchain

Figure 3 shows the process of testing and compiling on blockchain. Initially, the Metamask stored the details of all types of data in the wallet and provided separate addresses for each one. Then, to address users' data transmitted to the server, the cryptocurrency data also interact with the web server. From the server, it is transmitted to smart contacts and users' mobile phones, and then it is spread through the network of users.

BOX 1 Pseudo Code 1: Pseudo Code for Blockchain.

```
Step 1: Inputs: $g_i, M1_{gK}$
Step 2: Output: $gi_{H2}$, Action, $Encypted_{data}$, bi
Step 3: Generate current Unix time stamp bi
Step 4: $bi_j$ = bi in decimal digit of precision
Step 5: $bi_d$ = bi in integer form
Step 6: $gibi = gi \parallel bi_j$
Step 7: $SYM_{KEYi} = SHA256\,(gibi)$
Step 8: $gi_{H1} = SHA256\,(gi)$
Step 9: $gi_{H2} = SHA244\,(gi)$
Step 10: Store the $SYM_{KEYi}$ with $gi_{H1}, gi_{H2}$ to the CD
Step 11: $Data = M1_{gK} + M1_{gK}$
Step 12: $Encypted_{data} = Encypted_{data\,(SYM_{KEYi},Data)}$
Step 13: Action= "Create"
Step 14: Return $gi_{H2}$, Action, $Encypted_{data}$, $bi_d$
Step 15: End
```

Box 1 defines the product creation transaction. It considers the public key of manufacture as MIPK and the product code as Pi. The present Unix timestamp and Pi are distinctive numbers. Hence, the unique string PiTi is produced by their concatenation. The SYM_KEYi is denoted by an acted symmetric key, and the PiTi SHA256 hash must be unique. Only the manufacturer should know the symmetric key generation, so it should be kept secret. The SHA244, SYM_KEYi and hash SHA256 are stored on the CDs. The information should comprise and encrypt the new owners and current public key, and in this situation, both are similar. Finally, the transactional table is formed by Ti, PiH2, Encypted_data and actions, the pseudo code 1 Encypt_data (), and encypt the data.

3.3 Testing and compiling of the blockchain in the cloud

Figure 4 shows the testing and compiling process of the blockchain on the cloud. The normal procedure of the cloud process is that the user sends the request through the server and saves in the cloud database and then the response for the user from the cloud database to the user through the web server. Due to the security risk in the cloud, the proposed model integrates blockchain to transmit user data in a secure way. The first step is for the user to provide requests to the cloud through the API; here, the blockchain acts as a bridge between the user and the cloud database. The work of the blockchain is to receive the user request and send it to cloud data and then obtain the response from the cloud database and send it to the user in a secure way. Therefore, the user does not need to worry about their data transfer through the web.

BOX 2 Pseudo Code 2: Pseudo Code for Blockchain in Cloud.

```
Step 1: Require: $seed_i\,(i = o, 1, 2 \ldots x)$
Step 2: Ensure: $qk_s^o, bk_s^o, qk_s^i, bk_s^i\ (i = 1, 2 \ldots x)$
Step 3: $(qk_s^o, bk_s^o) < -\, G_s\,(seed_o)$
Step 4: for $i = 1$ to m do
Step 5: $(qk_s^i, bk_s^i) < -\, G_s\,(seed_i)$
Step 5: End for
Step 6: return $qk_s^o, bk_s^o, qk_s^i, bk_s^i\ (i = o, 1, 2 \ldots x)$
```

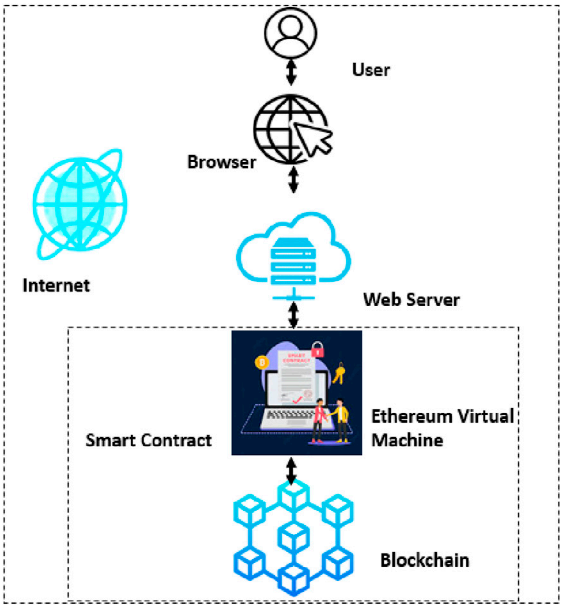
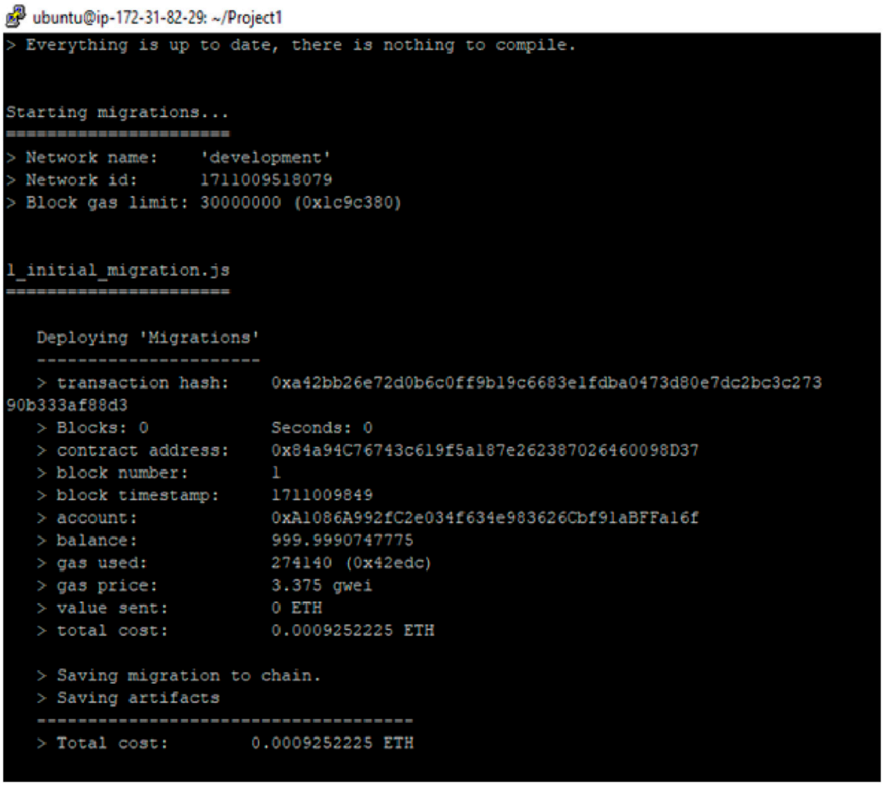


FIGURE 5
Testing and compiling of blockchain smart contracts.

Box 2 shows that the pseudo identity mechanism is used in permissionless blockchain because of its decentralized nature. For example, the user's identity is represented as an hashed public key by a comprised bitcoin address. Initially, both users and data owners must set a public key as their access control identity for each access control group. Protocol 1 (Ui, i ∈ 1, 2, ⋯, m) demonstrates the identity generation procedure for users and owners. In the meantime, as a requirement of data protection, the owner must produce an encryption key sk_{enc}^{0} before permitting access to other users. It shares each server s_i with one part; when the data owner partitions the user encryption key into n parts, the Shamir secret sharing scheme of t out n is employed here. If the user has any t of the parts, then only they can reconstruct the secret key.

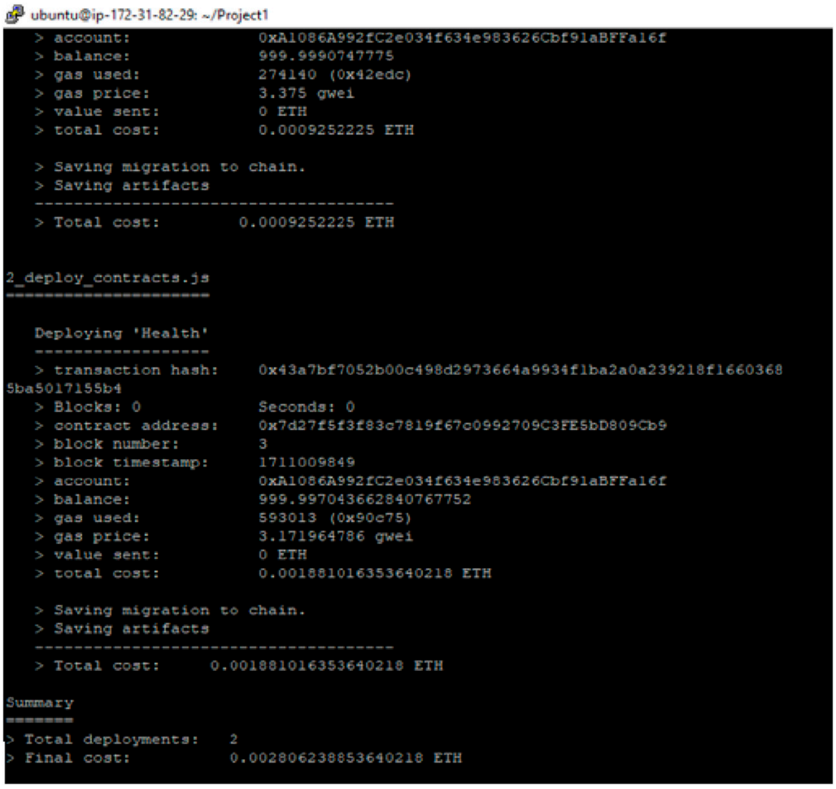
3.4 Testing and compiling the smart contract of the blockchain

Figure 5 shows the testing and compilation of blockchain smart contracts on the internet. Initially, the user gives their request; first, it will reach the browser, and then, it will reach the web server. Then, it will reach the EVM (ethereum virtual machine) of the smart contract; this is an application, and its function is from block to block to set the computing rules according to the state of the ethereum network. It executes the process of transferring the data, smart contracts and updates on account balance because it is the core part of Ethereum, after which it will finally reach the blockchain.



```
> Everything is up to date, there is nothing to compile.


Starting migrations...
======================
> Network name:    'development'
> Network id:      1711009518079
> Block gas limit: 30000000 (0x1c9c380)


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0xa42bb26e72d0b6c0ff9b19c6683e1fdba0473d80e7dc2bc3c273
90b333af88d3
   > Blocks: 0            Seconds: 0
   > contract address:    0x84a94C76743c619f5a187e262387026460098D37
   > block number:        1
   > block timestamp:     1711009849
   > account:             0xA1086A992fC2e034f634e983626Cbf91aBFFa16f
   > balance:             999.9990747775
   > gas used:            274140 (0x42edc)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.0009252225 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:        0.0009252225 ETH
```

FIGURE 6
Outcomes in puTTY Initialization.



ubuntu@ip-172-31-82-29: ~/Project1

```
   > account:             0xA1086A992fC2e034f634e983626Cbf91aBFFa16f
   > balance:             999.9990747775
   > gas used:            274140 (0x42edc)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.0009252225 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:        0.0009252225 ETH


2_deploy_contracts.js
=====================

   Deploying 'Health'
   ------------------
   > transaction hash:    0x43a7bf7052b00c498d2973664a9934f1ba2a0a239218f1660368
5ba5017155b4
   > Blocks: 0            Seconds: 0
   > contract address:    0x7d27f5f3f83c7819f67c0992709C3FE5bD809Cb9
   > block number:        3
   > block timestamp:     1711009849
   > account:             0xA1086A992fC2e034f634e983626Cbf91aBFFa16f
   > balance:             999.997043662840767752
   > gas used:            593013 (0x90c75)
   > gas price:           3.171964786 gwei
   > value sent:          0 ETH
   > total cost:          0.001881016353640218 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:     0.001881016353640218 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.002806238853640218 ETH
```

FIGURE 7
Outcomes in puTTY in Initialization.

BOX 3 Pseudo Code 3: Pseudo Code for Smart Contract on Blockchain.

```
Step 1: Result: Nonce Value Access the Smart Contracts from blockchain;
Step 2: Choose a blockchain to execute;
Step 3: Calculate and match the contract transaction;
Step 4: while All conditions do not meet do
Step 5: Get input fields from value or other source;
Step 6: Blockchain execute to generate hash;
Step 7: Store the hash on block.
```

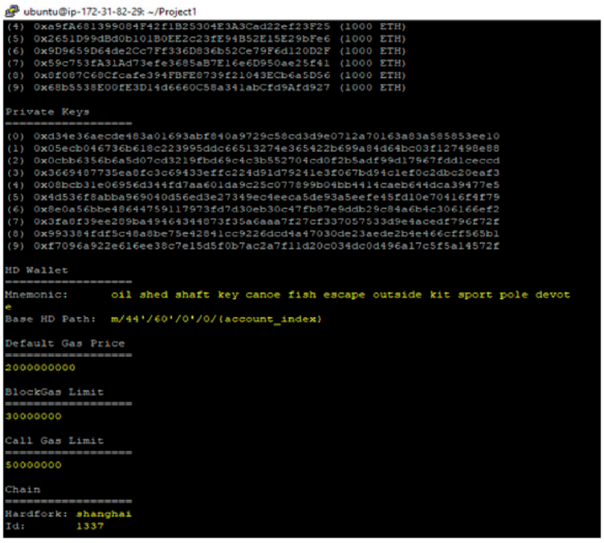
The pseudocode for a smart contract on a blockchain is shown; on the web, its input field and description are provided by the smart contract, and the user can access and deploy it in access mode (Box 3).

4 Results and discussion

The results attained by the proposed model are projected in this section, and the requirements and tools of the proposed model and the outcome of each stage are shown below.

4.1 Requirements and tools

- VScode
- AWS account
- AWS CLI
- puTTY



ubuntu@ip-172-31-82-29: ~/Project1

```
(4) 0xa9fA681399084F42f1B25304E3A3Cad22ef23F25 (1000 ETH)
(5) 0x2651D99dBd0b101B0EE2c23fE94B52E15E29bFe6 (1000 ETH)
(6) 0x9D9659D64de2Cc7Ff336D836b52Ce79F6d120D2F (1000 ETH)
(7) 0x59c753fA31Ad73efe3685aB7E16e6D950ae25f41 (1000 ETH)
(8) 0x8f087C68Cfcafe394FBFE8739f21043ECb6a5D56 (1000 ETH)
(9) 0x68b5538E00fE3D14d6660C58a341abCfd9Afd927 (1000 ETH)

Private Keys
==================
(0) 0xd34e36aecde483a01693abf840a9729c58cd3d9e0712a70163a83a585853ee10
(1) 0x05ecb046736b618c223995ddc66513274e365422b699a84d64bc03f127498e88
(2) 0x0cbb6356b6a5d07cd3219fbd69c4c3b552704cd0f2b5adf99d17967fdd1ceccd
(3) 0x3669487735ea8fc3c69433effc224d91d79241e3f067bd94c1ef0c2dbc20eaf3
(4) 0x08bcb31e06956d344fd7aa601da9c25c077899b04bb4414caeb644dca39477e5
(5) 0x4d536f8abba969040d56ed3e27349ec4eeca5de93a5eefe45fd10e70416f4f79
(6) 0x8e0a56bbe48644759117973fd7d30eb30c47fb87e9ddb29c84a6b4c306166ef2
(7) 0x3fa8f39ee289ba49464344873f35a6aaa7f27cf337057533d9e4acedf796f72f
(8) 0x993384fdf5c48a8be75e42841cc9226dcd4a47030de23aede2b4e466cff565b1
(9) 0xf7096a922e616ee38c7e15d5f0b7ac2a7f11d20c034dc0d496a17c5f5a14572f

HD Wallet
==================
Mnemonic:      oil shed shaft key canoe fish escape outside kit sport pole devot
e
Base HD Path:  m/44'/60'/0'/0/{account_index}

Default Gas Price
==================
2000000000

BlockGas Limit
==================
30000000

Call Gas Limit
==================
50000000

Chain
==================
Hardfork: shanghai
Id:       1337
```

FIGURE 8
The Connection of Ganache with puTTY.

4.2 Experimental outcomes

Figure 6 shows the initialization process of the hash transaction with deploying migrations. The following output contains the values of gas used: 274140, the gas price is 3.75 gwei, the block timestamp is 1,711,009,849 and the total cost is 0.0092.

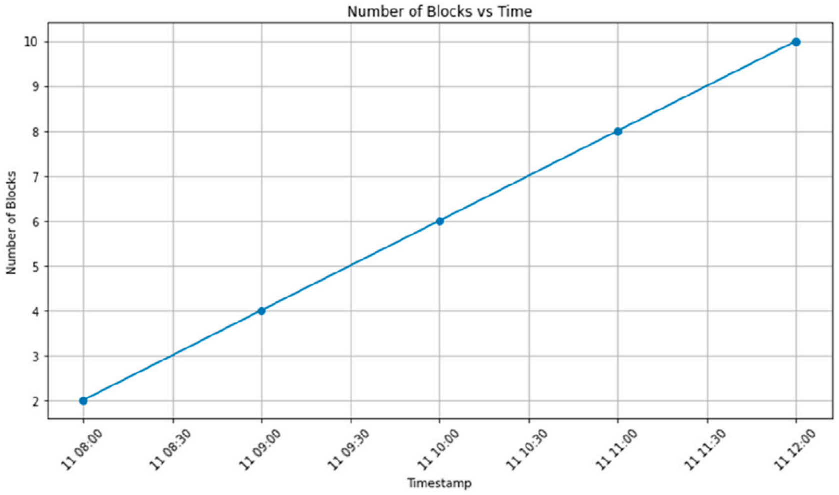


FIGURE 9
Number of Blocks and their Times.

Figure 6 shows the deployment process of hash transactions with health deployment. The following outputs are used: gas, 593013; gas, 3.171 gwei; block time, 1,711,009,849; and total cost, 0.001001.

Figure 7 shows the connection of Ganache with puTTY. Ganache is used for testing and development purposes by generating a local blockchain network. The figure shows the private keys of the user and the data stored in the wallet. The default gas price is 2,000,000,000, the blockgas limit is 30,000,000, and the call gas limit is 50,000,000.

Figure 8 shows the timestamp for the number of blocks generated in the system. Correspondingly, block numbers 2 to 4 are created at timestamps 11.08.00, 11.08.30 and 11.09.00. Similarly, block numbers 5 to 7 are created at timestamps 11.09.30, 11.10.00 and 11.10.30. Similarly, block numbers 8 to 10 are created at timestamps 11.11.00, 11.11.30 and 11.12.00. Figure 9 shows that the block is produced every 10 min during the process.

5 Conclusion and future recommendations

Cloud computing plays a vital role in all fields, even though some top-rated multinational companies (MCNs) have introduced their own cloud platform to store their own data and give payable access to the public to store their photos and documents on the cloud platform. However, the main risk is that cloud computing is a security risk, and data privacy is a threat to customers. To address this issue, the proposed model integrated the cloud and blockchain with smart contracts to overcome security risk issues. The user account has been created in the AWS EC2 of the cloud environment, and the data of users are processed on the smart contract, which has been solidified by the DApp. With the help of Ganache and Metamask, the final output is shown in the cloud, and an evaluation is performed. The proposed model has good efficacy for significant security measures. The future work of the proposed model is to develop middleware platforms to connect the gap among traditional systems and blockchain networks while facilitating smooth integration focused on API design.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

DZ: Conceptualization, Formal Analysis, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing. KV: Conceptualization, Formal Analysis, Methodology, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Awadallah, R., and Samsudin, A. (2021). Using blockchain in cloud computing to enhance relational database security. *IEEE Access* 9, 137353–137366. doi:10.1109/access.2021.3117733

Awadallah, R., Samsudin, A., Teh, J. S., and Almazrooie, M. (2021). An integrated architecture for maintaining security in cloud computing based on blockchain. *IEEE Access* 9, 69513–69526. doi:10.1109/access.2021.3077123

Chaganti, R., Varadarajan, V., Gorantla, V. S., Gadekallu, T. R., and Ravi, V. (2022). Blockchain-based cloud-enabled security monitoring using internet of things in smart agriculture. *Future Internet* 14 (9), 250. doi:10.3390/fi14090250

Chen, Y., Zhang, Y., Zhou, B., and Networking (2020). Research on the risk of block chain technology in internet finance supported by wireless network. *EURASIP J. Wirel. Commun. Netw.* 2020, 71–11. doi:10.1186/s13638-020-01685-6

De Giovanni, P. (2020). Blockchain and smart contracts in supply chain management: a game theoretic model. *Int. J. Prod. Econ.* 228, 107855. doi:10.1016/j.ijpe.2020.107855

Dolgui, A., Ivanov, D., Potryasaev, S., Sokolov, B., Ivanova, M., and Werner, F. (2020). Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. *Int. J. Prod. Res.* 58 (7), 2184–2199. doi:10.1080/00207543.2019.1627439

Eltayieb, N., Elhabob, R., Hassan, A., and Li, F. (2020). A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud. *J. Sys. Archit.* 102, 101653. doi:10.1016/j.sysarc.2019.101653

Fan, K., Bao, Z., Liu, M., Vasilakos, A. V., and Shi, W. (2020). Dredas: decentralized, reliable and efficient remote outsourced data auditing scheme with blockchain smart contract for industrial IoT. *Future Gene. Comput. Sys.* 110, 665–674. doi:10.1016/j.future.2019.10.014

Gai, K., Guo, J., Zhu, L., Yu, S., and Tutorials (2020). Blockchain meets cloud computing: a survey. *A Surv.* 22 (3), 2009–2030. doi:10.1109/comst.2020.2989392

Kemmoe, V. Y., Stone, W., Kim, J., Kim, D., and Son, J. (2020). Recent advances in smart contracts: a technical overview and state of the art. *IEEE Access* 8, 117782–117801. doi:10.1109/access.2020.3005020

Khan, A. A., Shaikh, Z. A., Laghari, A. A., Bourouis, S., Wagan, A. A., and Ali, G. (2021b). Blockchain-aware distributed dynamic monitoring: a smart contract for fog-based drone management in land surface changes. *Atmos. (Basel).* 12 (11), 1525. doi:10.3390/atmos12111525

Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., and N. Bani-Hani, and Applications (2021a). Blockchain smart contracts: applications, challenges, and future trends. *Peer. Netw. Appl.* 14, 2901–2925. doi:10.1007/s12083-021-01127-0

Khatoon, A. J. E. (2020). A blockchain-based smart contract system for healthcare management. *Electron. (Basel).* 9 (1), 94. doi:10.3390/electronics9010094

Kochovski, P., Stankovski, V., Gec, S., Faticanti, F., Savi, M., Siracusa, D., et al. (2020). Smart contracts for service-level agreements in edge-to-cloud computing. *J. Grid Comput.* 18, 673–690. doi:10.1007/s10723-020-09534-y

Kumar, A., Abhishek, K., Nerurkar, P., Ghalib, M. R., Shankar, A., and Cheng, X. (2022). Secure smart contracts for cloud-based manufacturing using Ethereum blockchain. *Trans. Emerg. Tel. Tech.* 33 (4), e4129. doi:10.1002/ett.4129

Leduc, G., Kubler, S., and Georges, J.-P. (2021). Innovative blockchain-based farming marketplace and smart contract performance evaluation. *J. Clean. Prod.* 306, 127055. doi:10.1016/j.jclepro.2021.127055

Lin, C., He, D., Huang, X., Choo, K.-K. R., and Security (2021). OBFP: optimized blockchain-based fair payment for outsourcing computations in cloud computing. *IEEE Trans. Inf. Forensics Secur.* 16, 3241–3253. doi:10.1109/tifs.2021.3073818

Muneeb, M., Raza, Z., Haq, I. U., and Shafiq, O. (2021). Smartcon: a blockchain-based framework for smart contracts and transaction management. *IEEE Access* 10, 23687–23699. doi:10.1109/access.2021.3135562

Murthy, C. V. B., Shri, M. L., Kadry, S., and Lim, S. (2020). Blockchain based cloud computing: architecture and research challenges. *IEEE Access* 8, 205190–205205. doi:10.1109/access.2020.3036812

Qin, X., Huang, Y., Yang, Z., and Li, X. (2021). A Blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. *J. Sys. Archit.* 112, 101854. doi:10.1016/j.sysarc.2020.101854

Saini, A., Zhu, Q., Singh, N., Xiang, Y., Gao, L., and Zhang, Y. (2020). A smart-contract-based access control framework for cloud smart healthcare system. *IEEE Internet Things J.* 8 (7), 5914–5925. doi:10.1109/jiot.2020.3032997

Sharma, A., Sarishma, Tomar, R., Chilamkurti, N., and Kim, B.-G. J. E. (2020). Blockchain based smart contracts for internet of medical things in e-healthcare. *Electron. (Basel).* 9 (10), 1609. doi:10.3390/electronics9101609

Sharma, P., Jindal, R., Borah, M., and Applications (2021). Blockchain-based decentralized architecture for cloud storage system. *J. Inf. Sec. Appl.* 62, 102970. doi:10.1016/j.jisa.2021.102970

Sigalov, K., Ye, X., König, M., Hagedorn, P., Blum, F., Severin, B., et al. (2021). Automated payment and contract management in the construction industry by integrating building information modeling and blockchain-based smart contracts. *Appl. Sci. (Basel).* 11 (16), 7653. doi:10.3390/app11167653

Singh, P. K., Singh, R., Nandi, S. K., Ghafoor, K. Z., Rawat, D. B., and Nandi, S. (2020). Blockchain-based adaptive trust management in internet of vehicles using smart contract. *IEEE Trans. Intell. Transp. Syst.* 22 (6), 3616–3630. doi:10.1109/tits.2020.3004041

Sookhak, M., Jabbarpour, M. R., Safa, N. S., and C. Applications (2021). Blockchain and smart contract for access control in healthcare: a survey, issues and challenges, and open issues. *J. Netw. Comput. Appl.* 178, 102950. doi:10.1016/j.jnca.2020.102950

Sultana, T., Almogren, A., Akbar, M., Zuair, M., Ullah, I., and Javaid, N. J. A. S. (2020). Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices. *Appl. Sci. (Basel).* 10 (2), 488. doi:10.3390/app10020488

Sun, T., and Yu, W. J. E. (2020). A formal verification framework for security issues of blockchain smart contracts. *Electron. (Basel).* 9 (2), 255. doi:10.3390/electronics9020255

Taher, M. Ghazal, Hasan, M. K., Abdullah, S. N. H. S., Bakar, K. A. A., Taleb, N., et al. (2023). "An integrated cloud and blockchain enabled platforms for biomedical research," in *The effect of information technology on business and marketing intelligence systems* (Springer), 2037–2053.

Taherdoost, H. J. I. (2023). Smart contracts in blockchain technology: a critical review. *Inf.* 14 (2), 117. doi:10.3390/info14020117

Tan, W., Zhu, H., Tan, J., Zhao, Y., Xu, L. D., and Guo, K. (2022). A novel service level agreement model using blockchain and smart contract for cloud manufacturing in industry 4.0. *Enterp. Inf. Syst.* 16 (12), 1939426. doi:10.1080/17517575.2021.1939426

Uriarte, R. B., Zhou, H., Kritikos, K., Shi, Z., Zhao, Z., and De Nicola, R. (2021). Distributed service-level agreement management with smart contracts and blockchain. *Concurrency Comput.* 33 (14), e5800. doi:10.1002/cpe.5800

Wang, H., Qin, H., Zhao, M., Wei, X., Shen, H., and Susilo, W. (2020). Blockchain-based fair payment smart contract for public cloud storage auditing. *Inf. Sci. (N. Y).* 519, 348–362. doi:10.1016/j.ins.2020.01.051

Wang, L., Zhu, L., Wang, X., Cong, H., and Shi, T. (2022). Design of integrated energy market cloud service platform based on blockchain smart contract *Int. J. Elec. Power Energ. Sys.* 135, 107515. doi:10.1016/j.ijepes.2021.107515

Wei, P., Wang, D., Zhao, Y., Tyagi, S. K. S., and Kumar, N. (2020). Blockchain data-based cloud data integrity protection mechanism. *Future Gene. Comput. Sys.* 102, 902–911. doi:10.1016/j.future.2019.09.028

Xie, S., Zheng, Z., Chen, W., Wu, J., Dai, H. N., and Imran, M. (2020). Blockchain for cloud exchange: a survey. *Comput. Electr. Eng.* 81, 106526. doi:10.1016/j.compeleceng.2019.106526

Yang, C., Tan, L., Shi, N., Xu, B., Cao, Y., and Yu, K. (2020). AuthPrivacyChain: a blockchain-based access control framework with privacy protection in cloud. *IEEE Access* 8, 70604–70615. doi:10.1109/access.2020.2985762

Yu, C., Zhang, L., Zhao, W., and Zhang, S. (2020). A blockchain-based service composition architecture in cloud manufacturing. *Int. J. Comput. Integr. Manuf.* 33 (7), 701–715. doi:10.1080/0951192x.2019.1571234

Zghaibeh, M., Farooq, U., Hasan, N. U., and Baig, I. (2020). Shealth: a blockchain-based health system with smart contracts capabilities. *IEEE Access* 8, 70030–70043. doi:10.1109/access.2020.2986789

Zhang, C., Xu, Y., Hu, Y., Wu, J., Ren, J., and Zhang, Y. (2021). A blockchain-based multicloud storage data auditing scheme to locate faults. *IEEE Trans. Cloud Comput.* 10 (4), 2252–2263. doi:10.1109/tcc.2021.3057771