Block Chain Implementation for Storing and Monitoring Data

Naziya Shahjahan Khan, Priya Ramsingar Yadav, Khusleen Kaur Puri, Prathamesh Manoj Deodkar, Harshal Pramod Pagare

Department of Computer Science and Engineering, Sandip University,
Nashik, Maharashtra, India.

Published In IJIRMPS (E-ISSN: 2349-7300), Volume 12, Issue 3, (May-June 2024)

Abstract

Blockchain implementation for storing and monitoring data involves leveraging decentralized, secure, and transparent ledgers to enhance data integrity and accessibility. This technology ensures tamper-proof records through cryptographic hashing and consensus mechanisms, reducing the risk of data manipulation. Real-time monitoring becomes efficient as participants across the network can access and verify data, fostering trust in the information's accuracy. Smart contracts, self-executing code on the blockchain, automate monitoring processes, enabling predefined conditions to trigger actions. Overall, blockchain implementation provides a robust foundation for secure, transparent, and decentralized data storage and monitoring solutions.

Introduction

Background

Conventional centralized data storage systems often encounter issues such as data tampering, lack of transparency, and single points of failure. Blockchain, however, operates on a decentralized network of nodes, each maintaining an identical copy of the ledger. This inherently distributed nature makes it resilient to unauthorized alterations and enhances the overall security of stored data.

Core Principles of Blockchain

At the heart of blockchain implementation are cryptographic techniques and consensus mechanisms. Each block in the chain contains a unique cryptographic hash of the previous block, creating a continuous and unbroken chain. Consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) ensure agreement among participants, validating transactions and maintaining the integrity of the ledger.

Secure Data Storage

Blockchain employs cryptographic hashes to secure data, making it practically impossible for malicious actors to alter information without detection. Once data is added to a block and the block is added to the chain, changing any information in a previous block requires changing all subsequent blocks — a computationally infeasible task.

Transparency and Accessibility

Unlike traditional databases where access permissions are managed by a central authority, blockchain allows for transparent and permissioned access. Participants in the network can access the entire history of the ledger, providing a high level of transparency. This transparency fosters trust among participants and stakeholders.

Real-time Monitoring

Blockchain facilitates real-time monitoring through the use of smart contracts. These self-executing contracts contain predefined rules and conditions. When these conditions are met, the contract is automatically executed, triggering actions such as updating the ledger, sending notifications, or executing other business logic. This automation streamlines monitoring processes and reduces the need for intermediaries.

Applications Across Industries

Blockchain implementation for storing and monitoring data has found applications in various industries, including finance, supply chain, healthcare, and more. It is particularly valuable in scenarios where data integrity, transparency, and security are paramount.

Literature Survey

Blockchain Fundamentals

The review begins by establishing the fundamental principles of blockchain technology, elucidating concepts such as decentralized ledgers, cryptographic hashing, and consensus mechanisms. This foundation is crucial for understanding the subsequent sections that delve into specific applications and implications.

Data Security and Integrity

Several studies have highlighted the cryptographic mechanisms inherent in blockchain that ensure the security and integrity of stored data. The immutability of blockchain, achieved through cryptographic hashing and the decentralized nature of the network, significantly reduces the risk of data tampering.

Transparency and Accessibility

Transparency is a critical aspect of blockchain implementation. This section explores literature that discusses how blockchain's transparent and permission access features contribute to increased trust among participants. Case studies from various industries demonstrate the positive impact of transparency on data reliability.

Real-time Monitoring with Smart Contracts

The integration of smart contracts in blockchain systems enables real-time monitoring and automation of predefined processes. This section reviews studies that showcase the versatility of smart contracts in facilitating instant data updates, triggering actions based on predefined conditions, and reducing the need for intermediaries in monitoring processes.

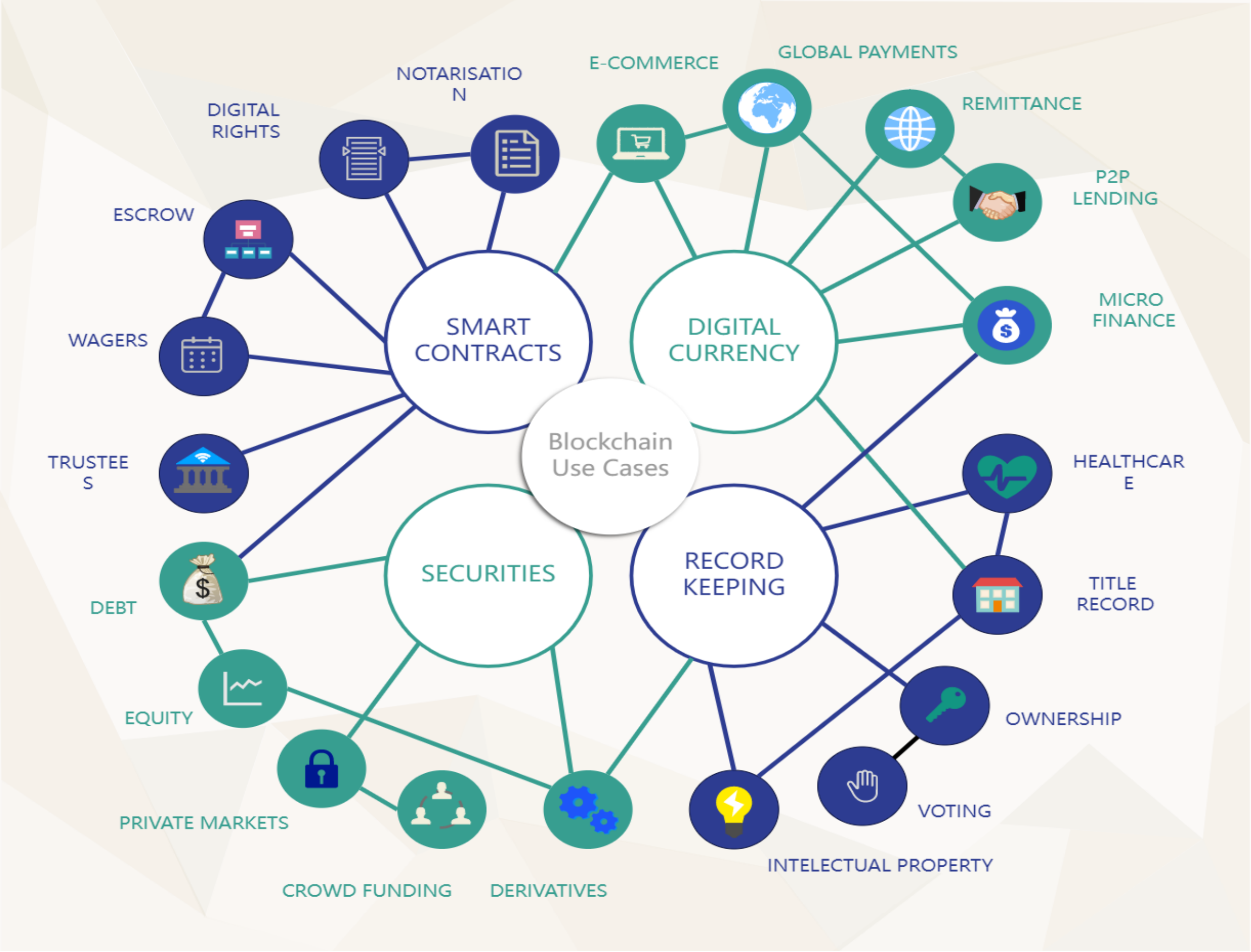
Industry-specific Applications

Drawing from literature across industries such as finance, healthcare, supply chain, and more, this section highlights specific use cases where blockchain implementation has demonstrated significant advantages in terms of data security, transparency, and monitoring efficiency.

Challenges and Future Directions

While the benefits of blockchain implementation are evident, challenges such as scalability, energy consumption, and regulatory considerations are discussed in this section. The review also outlines potential future directions for research, including hybrid solutions, interoperability, and the integration of emerging technologies

Use Case Diagram of Blockchain

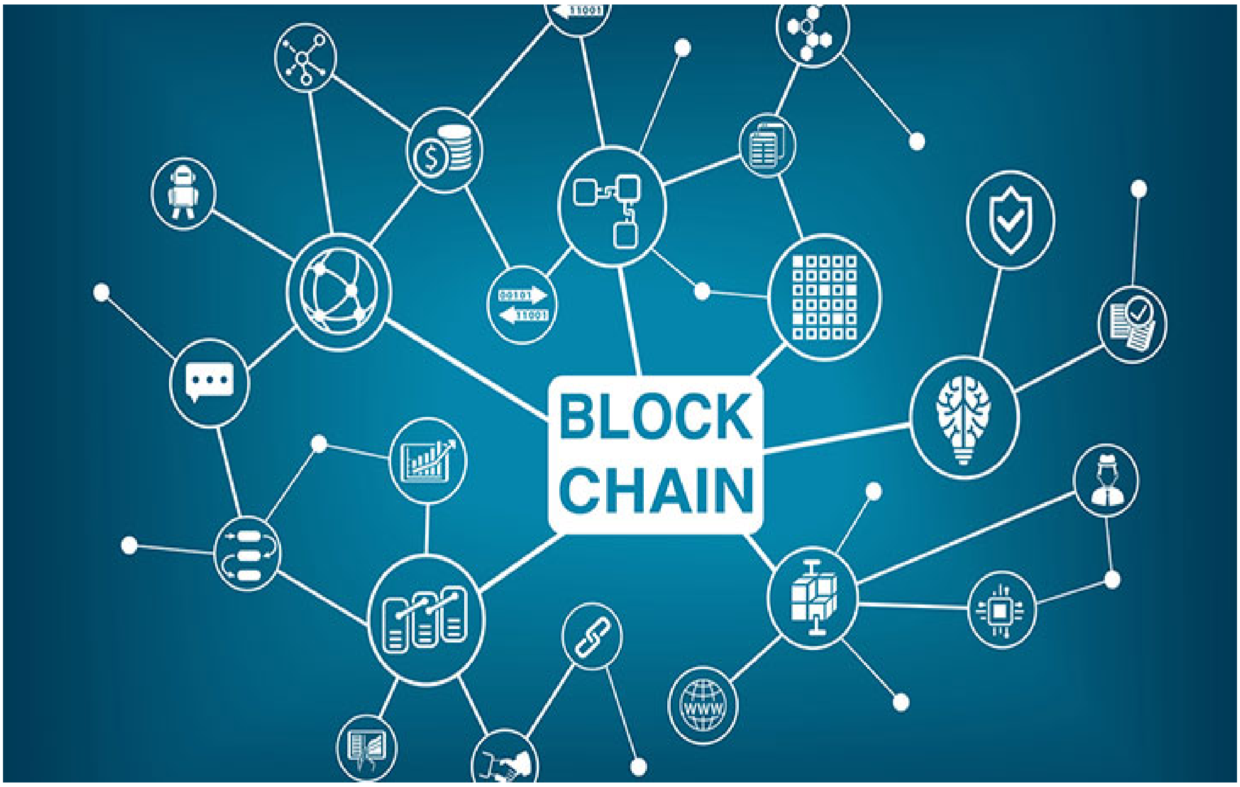


A use case diagram is a user-facing diagram that summarizes a system's users and their interactions with it. It's a Unified Modeling Language (UML) diagram that uses a set of symbols and connectors to help teams represent scenarios, goals, and system scope.

A use case diagram for blockchain can help analyze problem statements and discuss solution designs with customers. It can also help explain system behavior to users by describing all externally visible system behavior.

A use case diagram is usually simple and only summarizes some relationships between use cases, actors, and systems. It doesn't show the order in which steps are performed to achieve goals, and it shouldn't contain more than 20 use cases.

Methodology



It works under the principle of decentralized distributed digital ledger. This technology enables cryptographically secure and anonymous financial transactions among the user nodes of the network enabling the transactions to be validated and approved by all the users in a transparent environment.

Software Requirements and Specification

Software Requirements

Windows 10
HTML
CSS
NPM
JavaScript
NodeJS
ReactJS

Hardware Requirements

RAM: 4 GB or above
Storage: 8 GB or above
Processor: Intel Core i3 or above

System Design

System Design for Blockchain Implementation for Storing Monitoring Data:

1. **Blockchain Architecture Selection:** Choose an appropriate blockchain architecture (public, private, or consortium) based on factors such as data privacy requirements, scalability needs, and the level of decentralization desired.
2. **Data Schema Definition:** Define a standardized data schema for monitoring data to ensure

interoperability and consistency across the blockchain network. This schema should include essential attributes such as timestamp, data type, source, and metadata.

3. **Smart Contract Development:** Develop smart contracts to govern data storage, access control, and validation rules. Smart contracts automate data validation processes, ensuring that only authorized and valid data entries are added to the blockchain.
4. **Consensus Mechanism Selection:** Select a consensus mechanism suitable for the use case, such as Proof of Work (PoW), Proof of Stake (PoS), or Practical Byzantine Fault Tolerance (PBFT), balancing factors like scalability, energy efficiency, and security.
5. **Identity and Access Management:** Implement identity and access management solutions to manage user identities, permissions, and access control policies. This ensures that only authenticated users can submit, access, or modify monitoring data
6. **Data Encryption and Privacy:** Employ encryption techniques to ensure the privacy and confidentiality of sensitive monitoring data. Encryption mechanisms protect data both at rest and in transit, safeguarding it from unauthorized disclosure or tampering.
7. **Decentralized Storage Integration:** Integrate decentralized storage solutions, such as InterPlanetary File System (IPFS) or distributed file systems, to store large data files or off-chain data while maintaining references on the blockchain.
8. **Scalability Solutions:** Implement scalability solutions, such as sharding, sidechains, or offchain scaling techniques, to accommodate the growing volume of monitoring data and ensure efficient transaction processing
9. **Interoperability with External Systems:** Establish interoperability with existing monitoring systems, data sources, and external databases through standardized APIs, data formats, or middleware layers. This enables seamless data exchange and integration with external stakeholders or systems.
10. **Monitoring and Analytics Tools:** Develop monitoring and analytics tools to track blockchain performance, monitor data quality, and derive insights from monitoring data stored on the blockchain. These tools facilitate real-time monitoring, analysis, and visualization of monitoring data-trends and patterns.

System Architecture

System Architecture for Blockchain Implementation for Storing Monitoring Data:

1. **Data Sources:** The system architecture begins with various data sources that generate monitoring data, such as IoT devices, sensors, databases, or external data feeds. These sources continuously generate data that needs to be securely stored and managed.
2. **Data Ingestion Layer:** The data ingestion layer is responsible for collecting monitoring data from diverse sources and preparing it for storage on the blockchain. This layer may involve data preprocessing, normalization, and encryption to ensure compatibility and security.
3. **Blockchain Network:** At the core of the architecture is the blockchain network, consisting of a decentralized network of nodes that collectively maintain the ledger of monitoring data. The blockchain employs consensus mechanisms such as Proof of Work (PoW), Proof of Stake (PoS), or other consensus algorithms to validate and append transactions to the blockchain in a secure and immutable manner.
4. **Smart Contracts:** Smart contracts are self-executing contracts with predefined rules and logic deployed on the blockchain. In the context of storing monitoring data, smart contracts can enforce data validation rules, access control policies, and automate certain actions based on predefined

conditions. For example, smart contracts can automatically trigger alerts or notifications when predefined thresholds are exceeded

5. **Data Storage Layer:** The data storage layer comprises the distributed ledger technology (DLT) of the blockchain itself, where monitoring data is stored in encrypted and tamper-proof blocks. Each block contains a batch of transactions representing monitored data along with cryptographic hashes linking it to the previous block, ensuring the integrity and immutability of the data
6. **Consensus Mechanism:** The consensus mechanism governs how agreement is reached among network nodes regarding the validity of transactions and the order in which they are added to the blockchain. Depending on the specific requirements of the application, consensus mechanisms such as PoW, PoS, Delegated Proof of Stake (DPoS), or Practical Byzantine Fault Tolerance (PBFT) can be employed to ensure the security and integrity of the network
7. **Data Access and Query Layer:** This layer provides interfaces and APIs for authorized users and applications to query and access monitoring data stored on the blockchain. Access control mechanisms ensure that only authenticated and authorized users can retrieve specific data based on their permissions and roles.
8. **User Interfaces:** User interfaces, such as web portals, mobile applications, or command-line interfaces, enable users to interact with the system, visualize monitoring data, configure alerts, and manage access permissions. These interfaces provide a user-friendly experience for stakeholders to monitor and analyze data in real-time.
9. **External Integrations:** The system architecture may include integrations with external systems, databases, or analytics platforms to enable data exchange, analysis, and integration with existing monitoring infrastructure. These integrations facilitate interoperability and enhance the functionality of the overall system.

By integrating these components into a cohesive architecture, the blockchain implementation for storing monitoring data can provide a secure, transparent, and scalable solution that ensures data integrity, security, and accessibility for various stakeholders.

ER Diagram

Entity-Relationship (ER) diagrams are typically used to illustrate the logical structure of a database system, including entities, their attributes, and the relationships between them. For a blockchain implementation for storing monitoring data, the ER diagram may look something like this:

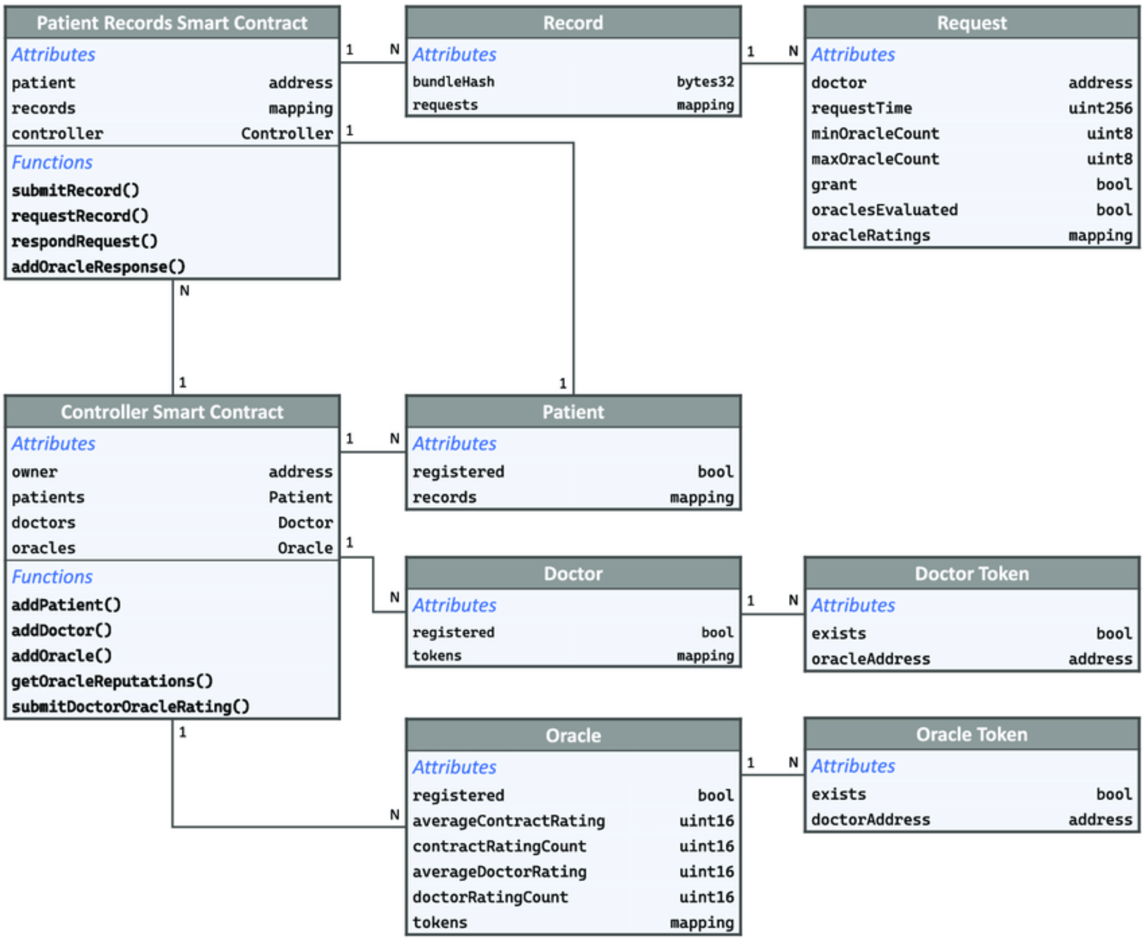
Entities

1. **Block:** Represents a block in the blockchain containing a batch of monitoring data transactions.
   Attributes: Block Id, Timestamp, Hash, Previous Block Hash, Nonce, Merkle Root, etc.
2. **Transaction:** Represents an individual transaction containing monitoring data
   Attributes: Transaction Id, Timestamp, Sender Address, Receiver Address, Data Payload, etc.
3. **Node:** Represents a node in the blockchain network.
   Attributes: Node Id, IP Address, Wallet Address, etc.
4. **Smart Contract:** Represents a self-executing contract with the terms of the agreement between parties
5. **User:** Represents a user or participant in the blockchain network.
   Attributes: User Id, Name, Role, Public Key, etc.

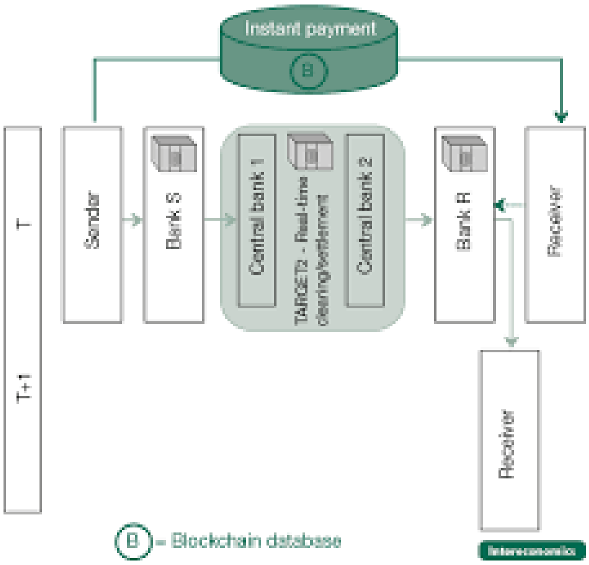
Relationships

1. **Blocks-Transactions:** One-to-Many relationship indicating that a block can contain multiple transactions
   Foreign Key: Block Id in Transaction entity
2. **Transactions-Sender:** Many-to-One relationship indicating that a transaction can have one sender.
   Foreign Key: Sender Addressing Transaction entity
3. **Transactions-Receiver:** Many-to-One relationship indicating that a transaction can have one receiver.
   Foreign Key: Receiver Address in Transaction entity
4. **Node-Blocks:** One-to-Many relationship indicating that a node can create or store multiple blocks
   Foreign Key: Node Id in Block entity
5. **Smart Contract-Transactions:** One-to-Many relationship indicating that a smart contract can be involved in multiple transactions
   Foreign Key: Contract Id in Transaction entity
6. **User-Transactions:** One-to-Many relationship indicating that a user can be involved in multiple transactions.
   Foreign Key: User Id in Transaction entity.

This ER diagram captures the basic structure of a blockchain system for storing monitoring data, including the entities involved (blocks, transactions, nodes, smart contracts, and users) and their relationships. Depending on the specific requirements and functionalities of the system, additional entities and relationships may be included in the diagram.



Data-flow Diagram (DFD)



Advantages

Advantages of Blockchain Implementation for Storing and Monitoring Data:

1. **Data Integrity and Immutability:** Blockchain technology ensures that monitoring data remains unchanged and tamper-proof once recorded, providing a verifiable and transparent record of all transactions. This guarantees the integrity of the data, enhancing trust among stakeholders.
2. **Enhanced Security:** Utilizing cryptographic techniques and decentralized storage, blockchain enhances the security of monitoring data, protecting it against unauthorized access, cyberattacks, and data breaches. Each transaction is cryptographically secured, reducing the risk of data manipulation or unauthorized tampering.
3. **Transparency and Auditability:** Blockchain provides a transparent and auditable record of all transactions, accessible to all participants in the network. This transparency fosters trust among stakeholders and enables real-time monitoring and verification of data integrity, contributing to accountability and compliance with regulatory standards
4. **Decentralization and Resilience:** By distributing data across a network of nodes, blockchain eliminates single points of failure and reduces the risk of data loss or system downtime. This decentralization enhances the resilience of the system against cyberattacks, network failures, and natural disasters, ensuring continuous access to monitoring data
5. **Efficient Data Sharing and Collaboration:** Blockchain facilitates secure and efficient data sharing and collaboration among stakeholders, eliminating the need for intermediaries or complex data-sharing agreements. Smart contracts automate the execution of predefined rules, ensuring transparent and consistent data access permissions, fostering collaboration, innovation, and information exchange.
6. **Scalability and Performance:** Blockchain solutions can be designed to scale efficiently to accommodate the growing volume and variety of monitoring data without compromising performance or efficiency. New blocks can be added to the blockchain in a decentralized manner, allowing the system to handle increasing data demands seamlessly.
7. **Regulatory Compliance:** Blockchain technology enables organizations to embed regulatory rules directly into smart contracts, automating compliance with industry standards and regulatory requirements. This reduces the risk of non-compliance penalties and ensures adherence to regulatory frameworks governing monitoring data management.

8. **Cost-effectiveness:** Blockchain eliminates the need for intermediaries and reduces administrative overhead associated with data management, resulting in cost savings for organizations. Additionally, the decentralized nature of blockchain reduces infrastructure costs by eliminating the need for centralized servers or data centers.

By leveraging these advantages, blockchain implementation for storing monitoring data offers a robust and trusted infrastructure for data management, ensuring data integrity, security, transparency, and compliance while fostering collaboration, innovation, and efficiency across various industries.

Applications

Applications for Blockchain Implementation for Storing Monitoring Data:

1. **Supply Chain Management:** Blockchain can be applied to store monitoring data related to the movement and condition of goods throughout the supply chain. This includes tracking temperature sensitive products, verifying product authenticity, and ensuring compliance with quality standards. By storing this data on a blockchain, stakeholders can ensure transparency, traceability, and integrity across the entire supply chain.
2. **IoT (Internet of Things) Monitoring:** In IoT applications, sensors collect vast amounts of data from various devices and sensors. Blockchain can provide a secure and decentralized platform for storing this data, ensuring its integrity and enabling real-time monitoring of devices and environmental conditions. This is particularly valuable in sectors such as smart cities, agriculture, andhealthcare, where continuous monitoring and data accuracy are essential.
3. **Environmental Monitoring:** Blockchain can be utilized to store monitoring data related to environmental parameters such as air quality, water quality, and climate conditions. By recording this data on a transparent and immutable ledger, stakeholders can track environmental changes, identify pollution sources, and enforce environmental regulations effectively. This promotes sustainable practices and facilitates data-driven decision-making in environmental management.
4. **Healthcare Data Management:** In the healthcare sector, blockchain can be employed to securely store and manage patient monitoring data, medical records, and clinical trial data. By ensuring the integrity and confidentiality of sensitive healthcare information, blockchain enhances patient privacy, facilitates interoperability among healthcare providers, and enables secure data sharing for research and treatment purposes.
5. **Financial Monitoring and Compliance:** Blockchain can be applied to store monitoring data related to financial transactions, compliance activities, and regulatory reporting. By recording financial data on a tamper-proof and auditable blockchain ledger, financial institutions can streamline compliance processes, mitigate the risk of fraud and money laundering, and enhance transparency in financial transactions.
6. **Energy Monitoring and Management:** In the energy sector, blockchain can be utilized to store monitoring data from smart meters, renewable energy sources, and energy consumption patterns. By leveraging blockchain for data storage and management, energy companies can optimize energy distribution, incentivize renewable energy production, and enable peer-to-peer energy trading while ensuring data integrity and transparency.

These applications demonstrate the versatility and potential impact of blockchain technology in storing monitoring data across various sectors. By providing secure, transparent, and tamper-proof data storage solutions, blockchain facilitates data-driven decision-making, enhances trust among stakeholders, and drives innovation in monitoring and management practices.

Software Testing

Software testing involves a series of steps to ensure that a software application meets its specified requirements, functions correctly, and delivers a quality user experience. Here are the typical steps involved in software testing:

1. **Requirement Analysis:** Understand and analyze the software requirements specified in the software requirement specification (SRS) document. This step involves identifying the functional and non-functional requirements of the software.
2. **Test Planning:** Develop a test plan that outlines the testing approach, objectives, scope, resources, and timelines. The test plan serves as a roadmap for the testing process and helps in organizing and prioritizing testing activities.
3. **Test Case Development:** Based on the requirements and test plan, create test cases that define the input data, expected outcomes, and test conditions for each test scenario. Test cases should cover both positive and negative test scenarios to validate the behavior of the software under different conditions.
4. **Test Environment Setup:** Set up the testing environment, including hardware, software, test tools, and test data. Ensure that the testing environment accurately represents the production environment to simulate real-world conditions.
5. **Test Execution:** Execute the test cases according to the test plan. This involves running the software with different input values and test scenarios to verify its functionality, performance, and reliability. Record the test results, including any defects or issues encountered during testing.
6. **Defect Reporting:** Report any defects or issues identified during the testing process using a defect tracking system. Provide detailed information about each defect, including steps to reproduce, severity, priority, and any other relevant information.
7. **Defect Management:** Manage the reported defects by prioritizing them based on severity and impact on the software. Assign defects to development teams for resolution and track the status of each defect until it is fixed and verified.
8. **Regression Testing:** Perform regression testing to ensure that changes or fixes made to the 32 software do not introduce new defects or regressions. Re-run previously executed test cases to validate the stability and integrity of the software after changes are made.
9. **Test Reporting:** Generate test reports summarizing the testing activities, including test coverage, test results, defect metrics, and any other relevant information. These reports provide stakeholders with insights into the quality and readiness of the software for release.
10. **Test Closure:** Evaluate the completion criteria specified in the test plan to determine if the testing objectives have been achieved. Conduct a test closure meeting to review the testing activities, lessons learned, and recommendations for future improvements.

By following these steps, software testing teams can systematically verify and validate the software to ensure that it meets quality standards, complies with requirements, and delivers value to end-user

Test Cases

Description of the Monitoring Data Source

The monitoring data source in this case study pertains to a supply chain management system. Various IoT (Internet of Things) devices, sensors, and smart contracts are employed to monitor and track the movement, condition, and authenticity of goods throughout the supply chain. These devices generate data related to location, temperature, humidity, and other relevant parameters. 5.2. Data Transactions on the Blockchain Blockchain technology is utilized to record and manage the monitoring data transactions. Each relevant event, such as the departure of goods from a warehouse, transit between locations, or arrival at the destination, triggers a transaction on the blockchain. Smart contracts automatically execute predefined rules, ensuring that the data recorded is accurate and tamper-proof. Transaction Structure: Each transaction includes a timestamp, a hash of the data, and the digital signatures of the involved parties. The hash ensures the integrity of the data, while digital signatures authenticate the participants.

Decentralization

The blockchain network is decentralized, meaning that multiple nodes across the supply chain have a copy of the entire transaction history. This decentralization enhances transparency and reduces the risk of a single point of failure. Immutability: Once a block is added to the blockchain, it is extremely difficult to alter or erase the information. This immutability ensures the integrity and trustworthiness of the monitoring data.

Algorithm

Here's a concise rundown:

1. **Security Protectors:** Blockchain algorithms safeguard digital data.
2. **Data Containers:** They help create and secure blocks of information.
3. **Agreement Makers:** Algorithms ensure everyone agrees on valid transactions.
4. **No Bosses:** They enable decentralization, meaning no single authority controls the network.
5. **Puzzle Solvers (Proof of Work):** Miners solve puzzles to validate transactions.
6. **Stakeholders' Influence (Proof of Stake):** Validators are chosen based on their coins and willingness to stake them.
7. **Energy and Speed Focus:** Some algorithms aim to reduce energy use and increase transaction speed.
8. **Cryptographic Guards:** They employ encryption techniques to secure data.
9. **Automated Contracts:** Algorithms execute self-executing contracts automatically.
10. **Adaptive Innovators:** They evolve to meet challenges and improve performance over time.

Conclusion

The study investigated the implementation of a blockchain system for monitoring data storage, addressing scalability, regulatory compliance, and energy consumption challenges. Key findings include the identification of scalability issues related to transaction throughput and consensus mechanisms. Regulatory compliance considerations emphasized the importance of data protection and privacy, with specific challenges such as the right to be forgotten. Energy consumption concerns were highlighted, especially in systems using proof-of-work consensus. The proposed system incorporated potential enhancements such as sharding, smart contract upgrades, and advanced privacy features. The use of blockchain for monitoring data storage has several implications. The decentralized and immutable nature of blockchain ensures data integrity and transparency. Enhanced privacy features contribute to secure and compliant data storage, addressing regulatory concerns. The implementation of smart contracts facilitates automated and secure data monitoring processes. The integration with emerging technologies, such as AI and IoT, adds a layer of sophistication to real-time data tracking. Blockchain's potential to establish trust in data storage processes can significantly impact industries relying on secure and auditable data management.

Recommendations

Future Research Scalability Solutions

Further research should explore and develop advanced scalability solutions, including sharing techniques, to address the increasing demands of transaction throughput in blockchain systems.

Energy-Efficient Consensus Mechanisms

Research efforts should continue to investigate and implement energy-efficient consensus mechanisms, such as proof-of-stake, to mitigate the environmental impact of blockchain systems.

Integration with Emerging Technologies

Explore deeper integration with emerging technologies, such as AI, machine learning, and IoT, to enhance the capabilities of the blockchain system in monitoring data storage.

Industry-Specific Use Cases

Investigate industry-specific use cases and conduct case studies to understand the practical applications and benefits of blockchain in monitoring data storage across diverse sectors.

In conclusion, ongoing research and development efforts in these areas will contribute to the evolution of blockchain systems for monitoring data storage, ensuring their relevance, efficiency, and compliance with regulatory standards in an ever-changing technological landscape.

References

[1] Salah K., Rehman M.H.U., Nizamuddin N., Al-Fuqaha A. Blockchain for AI: Review and openresearch challenges. IEEE Access, 2019, 7, 10127–10149. https://doi.org/10.1109/ACCESS.2018.2890507

[2] Lahami M., Maâlej A.J., Krichen M., Hammami M.A. A Comprehensive Review of Testing Blockchain Oriented Software, Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022). 25–26 April 2022, 355–362.

[3] Litke A., Anagnostopoulos D., Varvarigou T. Blockchains for deliver chain management: Architectural factors and challenges toward an international scale deployment. Logistics, 2019, 3, 5. https://doi.org/10.3390/logistics3010005

[4] Kouhizadeh M., SarkisJ. Blockchain practices, potentials, and viewsin greening deliver chains. Sustainability, 2018, 10, 3652. https://doi.org/10.3390/su10103652

[5] Schilling L., Uhlig H. Some simple bitcoin economics. J. Monet. Econ., 2019, 106, 16–26. https://doi.org/10.1016/j.Jmoneco.2019.07.002

[6] Ravishankar C.V., Kavitha K.S. Blockchain Applications that are Transforming the Society. In: Gururaj H.L., Ravi Kumar V., Goundar S., Elngar A.A., Swathi B.H., editors. Convergence of Internet of Things and Blockchain Technologies. Springer: Cham, Switzerland, 2022, 23–39.

[7] Zaabar B., Cheikhrouhou O., Jamil F., Ammi M., Abid M. HealthBlock: A stable blockchainbased healthcare records management system. Comput. Netw., 2021, 200, 108500. https://doi.org/10.1016/j.Comnet.2021.108500

[8] Jamil F., Cheikhrouhou O., Jamil H., Koubaa A., Derhab A., Ferrag M.A. PetroBlock: A blockchain-primarily based price mechanism for fueling clever motors. Appl. Sci., 2021, 11, 3055.

https://doi.org/10.3390/app11073055

[9] Frikha T., Chaabane F., Aouinti N., Cheikhrouhou O., Ben Amor N., Kerrouche A. Implementation of Blockchain Consensus Algorithm on Embedded Architecture. Secur. Commun. Netw., 2021, 9918697. https://doi.org/10.1155/2021/9918697

[10] Al-Jaroodi J., Mohamed N. Blockchain in industries: A survey. IEEE Access, 2019, 7, 36500–36515. https://doi.org/10.1109/ACCESS.2019.2903554

[11] Pal A., Tiwari C.K., Haldar N. Blockchain for business management: Applications, demanding situations and potentials.The Journal of High Technology Management Research, 23(2), 2021, 100414. https://doi.org/10.1016/j.Hitech.2021.100414

[12] Zhang L., Xie Y., Zheng Y., Xue W., Zheng X., Xu X. The challenges and countermeasures of blockchain in finance and economics. Syst. Res. Behav. Sci., 2020, 37, 691–698. https://doi.org/10.1002/sres.2710

[13] Tapscott A., Tapscott D. How blockchain is changing finance. Harv. Bus. Rev., 2017, 1, 2–5

[14] Prybutok V.R., Sauser B. Theoretical and realistic applications of blockchain in healthcare records control. Inf. Manag., 2022, 59, 103649.

[15] Adere E.M. Blockchain in healthcare and IoT: A systematic literature review. Array, 2022, 14, 100139.

[16] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” IEEE Communications Surveys Tutorials, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016

[17] M. D. Pierro, “What is the blockchain?” Computing in Science Engineering, vol. 19, no. 5, pp. 92–95, 2017

[18] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Tech. Rep., 2008. https://bitcoin.org/bitcoin.pdf

[19] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum Project Yellow Paper, vol. 151, pp. 1–32, 2014.

[20] D. T. T. Anh, M. Zhang, B. C. Ooi, and G. Chen, “Untangling blockchain: A data processing view of blockchain systems,” IEEE Transactions on Knowledge and Data Engineering, vol. PP, no. 99, pp. 1– 1, 2018

[21] T. Aste, P. Tasca, and T. D. Matteo, “Blockchain technologies: The foreseeable impact on society and industry,” Computer, vol. 50, no. 9, pp. 18–28, 2017.