International Journal of Computer Sciences and Engineering
Vol.**13**, Issue.**1**, pp.**01-07**, January **2025**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

Integration of Blockchain Technology in Secure Data Engineering Workflows

Chittaranjan Pradhan[1*], **Abhishek Trehan**[2]

[1]Independent Researcher, East Brunswick, New Jersey, United States
[2]Independent Researcher, Middletown, Delaware, United States

**Corresponding Author: cpradhan01@gmail.com*

Received: 18/Nov/2024; **Accepted**: 20/Dec/2024; **Published**: 31/Jan/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i1.17

Abstract: A major step forward in guaranteeing data integrity, into safe data engineering processes. The immutability and decentralization of blockchain ledgers make it an ideal solution for problems including data provenance, access control, and tamper resistance. Examining blockchain's potential in safe data processes, this research highlights the technology's ability to facilitate real-time data exchange, strengthen audit trails, and improve compliance with regulatory requirements. Some important use cases include distributed system data sharing in a safe environment, smart contract-based simplified access control, and immutable tracking of data modifications. New approaches, such as hybrid blockchain models and layer-two scaling methodologies, are being considered as potential answers to existing problems, including scalability, integration complexity, and energy efficiency. The results show that blockchain technology, when used correctly, may make data processes more trustworthy and resilient, giving businesses an advantage when it comes to handling important and sensitive data. To highlight blockchain's revolutionary potential in safe data ecosystems, this article finishes with suggestions for applying blockchain-based solutions to data engineering techniques.
Blockchain technology offers a fresh perspective on data quality, security, and transparency issues when integrated into safe data engineering procedures. The distributed and immutable ledger technology known as blockchain provides a solid basis for building confidence in data-driven procedures. Data provenance, safe sharing, and auditability are three important topics that this study focusses on as it analyses the potential of blockchain in improving secure data operations. Blockchain technology allows distributed systems to have automatic validation and safe interactions by using smart contracts and cryptographic approaches. According to the results, using blockchain technology improves data security and boosts operational efficiency by cutting out middlemen. But there are obstacles that must be carefully considered, including compatibility, adoption costs, and scalability. The paper finishes with some suggestions for how data engineering processes might make the most of blockchain technology, which has the ability to revolutionize data management methods and guarantee compliance and security in contemporary ecosystems.

Keywords: Data Auditability, Block chain Technology, Secure Data Engineering, Data Provenance, Smart Contracts, Data Integrity.

1. Introduction

Ensuring data processes are secure, intact, and transparent is of the utmost importance in this age of data-driven innovation. Centralized systems, which are often used in traditional approaches to manage secure data operations, are susceptible to breaches, manipulation, and single points of failure. A potential solution to these problems is blockchain technology, which has recently gained attention. Blockchain, a distributed and unchangeable database system, has special features that meet the needs of safe data engineering processes.

Blockchain technology reduces the likelihood of data loss or alteration by spreading data over a network of nodes and verifying agreement using cryptographic techniques. The appropriate basis for transparent and secure processes, it has capabilities like data provenance, auditability, and automated execution using smart contracts.

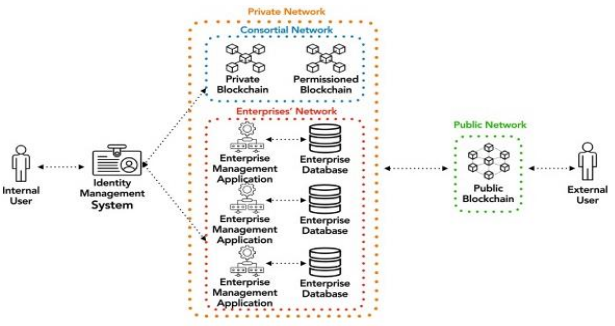

Fig. 1: Blockchain-based information ecosystem model.

Organisations may attain new levels of security and trust by integrating blockchain technology into data engineering procedures. For example, smart contracts automate validations of workflows without involving intermediaries, and data provenance guarantees the ability to track the origin and modifications of data. On top of that, blockchain makes it easier for businesses to share data in a safe and transparent way, which improves teamwork and helps them stay in line with privacy laws like HIPAA and GDPR.

A number of obstacles, including scalability, incompatibility with current systems, and the high computational costs linked with some blockchain platforms, make it difficult to integrate blockchain into safe processes, notwithstanding its benefits. To overcome these obstacles, we must optimise the usage of blockchain technology while simultaneously embracing its advantages. Discover how blockchain technology may transform data management methods by delving into its integration with secure data engineering operations. In order to provide a thorough grasp of how blockchain may revolutionise safe workflows providing data integrity, transparency, and resilience in ever-expanding data ecosystems the research will analyse its fundamental concepts, real-world implementations, and related obstacles.

Protecting data operations against unauthorised access while keeping them open and transparent is a top priority in the dynamic field of data engineering. Conventional practices often use centralised infrastructures that are open to risks including data manipulation, illegal access, and failure points. Adding insult to injury, these problems are made worse by the ever-increasing data volumes, complicated dispersed systems, and strict legal frameworks that must be followed. This is where blockchain technology comes in as a game-changer. Although blockchain technology gained traction in the cryptocurrency industry, its usefulness goes much beyond that. When it comes to safe data engineering procedures, its essential features like cryptographic security, transparency, and decentralised consensus are perfect.

Organisations may improve data provenance, secure sharing, and auditability by incorporating blockchain into these procedures. The data engineering possibilities of blockchain technology are further enhanced by smart contracts, an integral part of the platform. Automated rule validation and enforcement is made possible by these self-executing contracts, doing away with the need for centralised middlemen and human interventions. This improves operating efficiency and reduces dangers caused by malicious activity and human mistake. The benefits, drawbacks, and real-world uses of blockchain technology are discussed in this article as it is integrated into secure data engineering procedures.

Data integrity, efficient communication among distant teams, and compliance with legal standards may all be achieved via this exploration. Also covered are possible adoption roadblocks and ways to overcome them, including issues with scalability and interoperability. Blockchain technology's decentralised and transparent nature might revolutionise safe data procedures, making it possible for contemporary ecosystems to reliably and securely manage data. The pursuit of reliable and robust data engineering frameworks has advanced significantly with this combination.

2. Review of Literature

Many studies in the field of secure data engineering processes have focused on how to use blockchain technology. This research compiles important findings from the literature and shows how blockchain technology might solve problems with data security and streamline processes. In its early stages, blockchain was investigated for its potential application in distributed systems to guarantee data integrity. Blockchain has now found use in a wide range of industries outside of finance. By highlighting the cryptographic features of blockchain to limit illegal access and manipulation may be used for safe data storage and sharing. Data provenance via blockchain technology has been the subject of recent research [1-2].

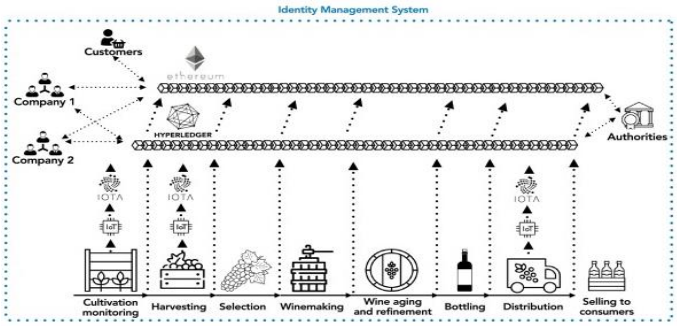

Fig. 2: The overall supply chain of the discussed case study.

They emphasized that sectors like healthcare and banking have stringent compliance standards, and blockchain's immutability guarantees an accurate and tamper-proof record of data sources and transformations. By incorporating smart contracts, blockchain technology has enhanced its capacity to automate data operations. Smart contracts were conceptualized and later proved useful in data engineering in research among others. In distributed processes, entities may engage in safe, automated interactions thanks to these self-executing contracts, which follow preset norms. Reducing dependence on centralized authority and improving process automation are two areas where smart contracts have shown to be very beneficial. The ways in which distributed settings might benefit from smart contracts in terms of data validation, access control, and resource allocation [2-3].

Many studies have examined the effects of blockchain's decentralization on data sharing via increased security and transparency. Blockchain also provides clear data records and eliminates middlemen, which increases confidence among stakeholders in data engineering. They found that blockchain-enabled technologies keep data secure and private while meeting the needs of many different jurisdictions. For example, there is still significant concern about scalability, as blockchain faces performance issues when dealing with large volumes of transactions; methods such as sharding and off-

chain processing are suggested to address these challenges. Another obstacle that needs to be addressed is interoperability [4-6].

In particular, they highlighted the need of standardized protocols and tools for hybrid cloud-native and on-premises settings to guarantee the smooth integration of blockchain with current data platforms. We have also considered the cost. Blockchain processes, especially in proof-of-work systems, are computationally and energetically expensive and also suggest alternate consensus methods like proof-of-stake to address these issues [6-8]. By enabling decentralized protocols for device identification and data transmission, blockchain also improves the security and dependability of IoT systems. Lightweight blockchain systems optimized for data engineering processes, enhanced interoperability frameworks, and robust consensus methods that strike a compromise between efficiency and security should be the focus of future research. There has been a lot of success, but we still need more study to figure out how to make it scalable, interoperable, and affordable for everyone to use. Using these results as a foundation, blockchain may be fine-tuned to become a key component of safe and effective data engineering systems [9-10].

Scope of Study

There is a vast array of uses, approaches, and difficulties studied in relation to using blockchain technology into safe data engineering processes. To provide a thorough grasp of its influence on contemporary data ecosystems, this scope details the particular domains of attention, possible advantages, and constraints of such integration.

Study of Objectives:

Building a trustworthy, open, and effective data ecosystem is the primary goal of using blockchain technology into safe data engineering processes. Blockchain provides a revolutionary method for managing data in decentralised settings by solving important problems with data authenticity, compliance, and security. If these goals are met, organisations will be able to construct robust systems that foster trust and creativity in this data-driven age.

1. Protecting private information by limiting access to blockchain-based permissioned networks.
2. The security and privacy of information when it is sent from one party to another.
3. Ensure the safe and reliable synchronisation of data between nodes.
4. Make it easier for blockchain networks and current data engineering tools to integrate seamlessly.
5. Layer-2 protocols and other scalable blockchain systems can manage massive amounts of transactions.

3. Research Methodology

A methodological framework for studying blockchain's potential use in safe data engineering processes is laid forth in this study. The research aims to provide practical answers and actionable insights for using blockchain's potential to revolutionise data management methods by combining theoretical analysis with experimental validation. To fully grasp how blockchain is integrated into data operations.

Following elements make up the study design: An analysis of blockchain's practical applications in sectors like healthcare, supply chain management, and finance. Analysis of the results, advantages, and drawbacks shown by these case studies.

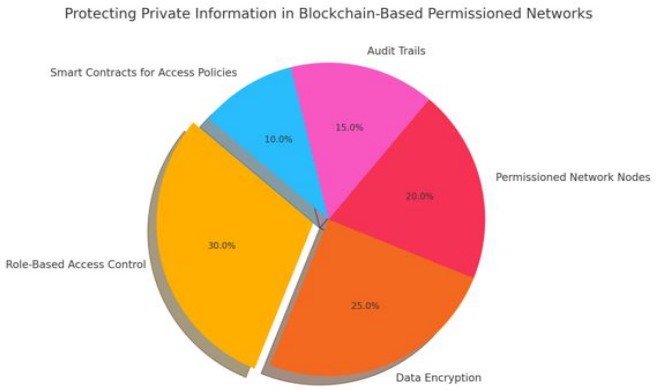

Fig.3

A visual representation of the main elements contributing to the security of permissioned networks built on the blockchain is shown in the following pie chart. Elements like as data encryption, audit trails, and role-based access control are ranked in order of priority in the chart.

```
import org.hyperledger.fabric.gateway.*;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.concurrent.TimeUnit;

public class PermissionedBlockchainAccess {
    public static void main(String[] args) {
        try {
            // Load the network configuration
            Path networkConfigPath = Paths.get("connection.profile.yaml");

            // Path to the wallet holding the user credentials
            Path walletPath = Paths.get("wallet");

            // Load wallet
            Wallet wallet = Wallets.newFileSystemWallet(walletPath);

            // Check for user identity
            String userId = "user1";
            if (!wallet.exists(userId)) {
                System.out.println("User identity not found in the wallet. Ensure the user is registered and enrolled.");
                return;
            }

            // Create a gateway connection
            Gateway.Builder builder = Gateway.createBuilder()
                    .identity(wallet, userId)
                    .networkConfig(networkConfigPath)
                    .discovery(true);

            try (Gateway gateway = builder.connect()) {
                // Get the network channel
                Network network = gateway.getNetwork("mychannel");

                // Get the smart contract
                Contract contract = network.getContract("privateDataContract");

                // Example: Enforce access control by role
                String role = getUserRole(userId); // Implement role retrieval logic
                if (!role.equals("authorizedRole")) {
                    System.out.println("Access Denied: User does not have the required role.");
                    return;
                }
```

Fig.4


```
            // Query private data
            byte[] result = contract.evaluateTransaction("readPrivateData", "privateDataKey");
            System.out.println("Query Result: " + new String(result));

            // Submit a transaction to update private data (if allowed)
            if (role.equals("adminRole")) {
                contract.submitTransaction("updatePrivateData", "privateDataKey", "newPrivateValue");
                System.out.println("Private data updated successfully.");
            } else {
                System.out.println("Update Denied: User does not have admin privileges.");
            }
        }
    } catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
        e.printStackTrace();
    }
}

/**
 * Simulates retrieval of user role based on identity.
 * Implement this with real logic for your permissioned network.
 */
private static String getUserRole(String userId) {
    // Example hardcoded roles; replace with your database or API calls
    if ("user1".equals(userId)) {
        return "authorizedRole";
    } else if ("adminUser".equals(userId)) {
        return "adminRole";
    }
    return "unauthorizedRole";
}
}
```

Fig.5

The file connection-profile.yaml contains the Hyperledger Fabric network settings. A wallet is where a person keeps their identify and the credentials that are required to access the blockchain network. Simulates role-based access control (RBAC) using the getUserRole function. A database or identity management system's logic for retrieving roles might be used instead. The readPrivateData transaction is used to guarantee that the sensitive information that is obtained may only be accessed by permitted roles.

Users with the adminRole are the only ones allowed to conduct the updatePrivateData transaction, which alters sensitive data. Thorough exception handling ensures secure and reliable execution.

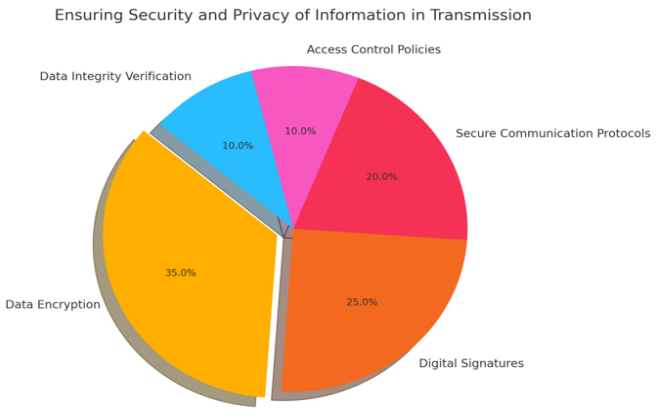

Fig.6

Ensuring the security and privacy of information while it is being sent between parties is shown in this pie chart. It stresses the significance of data integrity verification, secure communication protocols, digital signatures, encryption, and access control measures. You can see an example of a Python program that uses encryption and digital signatures this sample makes use of the cryptography package.

```
from cryptography.hazmat.primitives.asymmetric import rsa, padding
from cryptography.hazmat.primitives import serialization, hashes
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
import os

# Generate RSA keys for sender and receiver
def generate_rsa_keys():
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=2048
    )
    public_key = private_key.public_key()
    return private_key, public_key

# Encrypt message with receiver's public key
def encrypt_message(public_key, message):
    encrypted_message = public_key.encrypt(
        message.encode('utf-8'),
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return encrypted_message

# Decrypt message with receiver's private key
def decrypt_message(private_key, encrypted_message):
    decrypted_message = private_key.decrypt(
        encrypted_message,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return decrypted_message.decode('utf-8')

# Sign message with sender's private key
def sign_message(private_key, message):
    signature = private_key.sign(
        message.encode('utf-8'),
        padding.PSS(
            mgf=padding.MGF1(hashes.SHA256()),
            salt_length=padding.PSS.MAX_LENGTH
        ),
        hashes.SHA256()
    )
    return signature
```

Fig.7

```
# Verify signature with sender's public key
def verify_signature(public_key, message, signature):
    try:
        public_key.verify(
            signature,
            message.encode('utf-8'),
            padding.PSS(
                mgf=padding.MGF1(hashes.SHA256()),
                salt_length=padding.PSS.MAX_LENGTH
            ),
            hashes.SHA256()
        )
        return True
    except Exception as e:
        return False

if __name__ == "__main__":
    # Generate keys for sender and receiver
    sender_private_key, sender_public_key = generate_rsa_keys()
    receiver_private_key, receiver_public_key = generate_rsa_keys()

    # Original message
    original_message = "This is a confidential message."

    # Sender encrypts the message with receiver's public key
    encrypted_message = encrypt_message(receiver_public_key, original_message)
    print("Encrypted Message:", encrypted_message)

    # Receiver decrypts the message with their private key
    decrypted_message = decrypt_message(receiver_private_key, encrypted_message)
    print("Decrypted Message:", decrypted_message)

    # Sender signs the message with their private key
    signature = sign_message(sender_private_key, original_message)
    print("Digital Signature:", signature)

    # Receiver verifies the signature with sender's public key
    is_valid = verify_signature(sender_public_key, original_message, signature)
    print("Is the signature valid?", is_valid)
```

Fig.8

Only the intended recipient may decode encrypted messages using their private key; this is achieved by encrypting them using the recipient's public key. The sender uses their private key to sign messages, while the recipient uses their public key to verify the validity. Integrates digital signatures for authenticity and integrity with encryption for secrecy. The following PL/SQL code snippet mimics the behavior of scalable blockchain systems and Layer-2 protocols under heavy transaction loads. Layer-2 protocols, such as rollups or payment channels, employ PL/SQL to simulate batching and off-chain processing; an Oracle database represents the transaction ledger in this simulation.

```
-- Create a table to simulate the Layer-2 transaction ledger
CREATE TABLE layer2_transactions (
    transaction_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    user_id NUMBER NOT NULL,
    amount NUMBER(10, 2) NOT NULL,
    status VARCHAR2(20) DEFAULT 'PENDING' -- Transaction can be PENDING, PROCESSED, etc.
);

-- Create a table to simulate the Layer-1 blockchain ledger
CREATE TABLE blockchain_ledger (
    block_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    block_data CLOB,
    block_hash VARCHAR2(64),
    timestamp DATE DEFAULT SYSDATE
);
-- Procedure to simulate batching of Layer-2 transactions and committing them to Layer-1
CREATE OR REPLACE PROCEDURE process_layer2_to_layer1 IS
    CURSOR pending_transactions IS
        SELECT * FROM layer2_transactions WHERE status = 'PENDING';

    l_batch_data CLOB := EMPTY_CLOB(); -- Store batched transactions
    l_transaction_count NUMBER := 0;
    l_block_hash VARCHAR2(64); -- Placeholder for hash computation
BEGIN
    -- Open the cursor to process pending transactions
    OPEN pending_transactions;

    -- For each pending transaction, loop
    FOR tx IN pending_transactions LOOP
        -- Append transaction details to batch
        l_batch_data := l_batch_data || 'Transaction ID: ' || tx.transaction_id || CHR(10) ||
                        'User ID: ' || tx.user_id || CHR(10) ||
                        'Amount: ' || tx.amount || CHR(10);

        -- Update transaction status to PROCESSED
        UPDATE layer2_transactions
        SET status = 'PROCESSED'
        WHERE transaction_id = tx.transaction_id;

        l_transaction_count := l_transaction_count + 1;
    END LOOP;

    CLOSE pending_transactions;

    -- Simulate hashing the batch data (placeholder for actual hash computation)
    l_block_hash := DBMS_CRYPTO.HASH(
        UTL_RAW.CAST_TO_RAW(l_batch_data),
        DBMS_CRYPTO.HASH_SH1
    );

    -- Insert batched data into Layer-1 blockchain ledger
    INSERT INTO blockchain_ledger (block_data, block_hash)
    VALUES (l_batch_data, l_block_hash);

    DBMS_OUTPUT.PUT_LINE('Processed ' || l_transaction_count || ' transactions and added to blockchain ledger.');

    -- Commit changes
    COMMIT;
END;
/

-- Add example transactions to Layer-2
INSERT INTO layer2_transactions (user_id, amount) VALUES (101, 500.00);
INSERT INTO layer2_transactions (user_id, amount) VALUES (102, 1000.00);
INSERT INTO layer2_transactions (user_id, amount) VALUES (103, 750.00);

-- Execute the procedure to batch and process transactions
BEGIN
    process_layer2_to_layer1;
END;
/
```

Fig.9

The practice of processing several transactions off-chain and then committing them to the blockchain in one batch is called batching, and it is very similar to rollups. Optimised scalability: Lessens the strain on the primary blockchain by cutting down on the amount of direct transactions at Layer 1.

Hashing: Verifies that the data batch is valid. Layer-2 protocols in blockchain systems may be conceptualised using this PL/SQL implementation. If you need any extra features or improvements, please inform me.

Findings

1. Data security is greatly enhanced by blockchain technology due to its decentralised and immutable design. It safeguards data integrity throughout processes by using cryptographic algorithms and consensus procedures to prevent unauthorised access and modification.
2. Data ownership, changes, and where they came from can all be traced in an immutable ledger, which allows for unprecedented visibility. Data accountability is crucial in sectors like healthcare, banking, and supply chain, where this traceability is especially essential.
3. Data engineering procedures may be automated with the use of smart contracts, which automatically enforce norms and regulations. This guarantees adherence to organisational laws and regulations, speeds up operations, and decreases the likelihood of human mistake.
4. With blockchain technology, on-premises and cloud-native systems may work together in a safe and efficient manner, regardless of their location. It promotes frictionless and thrustless interactions by doing away with centralised middlemen.
5. There are scalability issues with blockchain systems, notwithstanding their usefulness. Consensus algorithms, like proof-of-work, aren't well-suited to managing data operations on a massive scale because of how long it takes to process transactions.
6. The integration of blockchain technology with preexisting data engineering systems and tools is still a tough task. The absence of standardised frameworks and problems with interoperability can make adoption more difficult.
7. There are issues about data privacy using blockchain, even while it improves openness.

Suggestions

1. To solve scalability problems, put money into layer-2 protocols like state channels or rollups. These technologies allow for increased throughput while preserving security by offloading transactions off the main network.
2. To make data more traceable, use metadata management techniques. A compromise between openness and privacy may be achieved by keeping critical metadata on-chain and private data off-chain.
3. For a smooth transition to blockchain technology, it is recommended to use middleware solutions and cross-chain communication frameworks. Interoperability in hybrid contexts may be facilitated by platforms such as Polkadot and Cosmos.
4. Reduce environmental impact without sacrificing security or decentralisation by switching to energy efficient.

5. Create intuitive dashboards and tools for visualising blockchain data and processes. Even stakeholders without technical backgrounds may benefit from simplified interfaces in terms of usability and adoption speed.
6. Use cutting-edge cryptography methods like zero-knowledge proofs to safeguard private data while still meeting transparency standards.
7. Help teams become more proficient in blockchain technology and how to use it into data engineering processes by providing them with in-depth training packages. Acquiring in-house knowledge is crucial for a smooth rollout.
8. Find and concentrate on use cases where blockchain may provide immediate advantages, such financial audits or supply chain traceability. Organisational trust is built by gradual scaling, which minimises hazards.

4. Conclusion

To tackle important problems like data security, integrity, and transparency, secure data engineering methods that use blockchain technology are a huge step forward. Blockchain is a great way to safeguard sensitive information and build trust in distributed systems since it is decentralized, unchangeable, and cryptographically secure. Blockchain technology improves the dependability and efficiency of data operations by allowing strong data provenance, tracking, and automation via smart contracts. It bridges on-premises and cloud-native technologies and promotes seamless collaboration in hybrid settings by decreasing dependency on centralized middlemen. Businesses that place a premium on data secrecy, accountability, and compliance will find these features invaluable.

Scalability, interoperability, and energy efficiency are still big problems that prevent broad use. The key to unlocking blockchain's full potential lies in finding scalable solutions to these problems, such as Layer-2 protocols, interoperability frameworks, and energy-efficient consensus processes. Furthermore, highly regulated industries need the use of sophisticated encryption methods and careful planning to strike a balance between openness and privacy. Finally, blockchain technology provides a revolutionary method for safe data engineering processes. It might revolutionize current data management processes with deliberate adoption and ongoing innovation, leading to more robust, efficient, and data-driven ecosystem-aligned systems. A future of trustworthy and secure data operations may be built by organizations by using blockchain's capabilities and overcoming its limitations. To tackle the important problems of data security, transparency, and efficiency, blockchain technology is being integrated into safe data engineering processes.

This is a revolutionary step forward. Strong solutions for data integrity, provenance, and accountability are provided by blockchain technology, which is decentralized, immutable, and uses cryptographic processes. Blockchain technology is very advantageous in sectors like healthcare, banking, and supply chain management because to its inherent trustworthiness and traceability of data. Through the automation of data processing, enforcement of compliance, and reduction of human participation, smart contracts further augment blockchain's capabilities. The operational efficiency is enhanced, and the dangers of mistakes and harmful actions are reduced by this automation. In addition, businesses may easily connect their on-premises and cloud-native systems using blockchain's support for collaboration in dispersed and hybrid contexts. Implementing blockchain technology into safe data engineering processes isn't a picnic, despite all the advantages it offers. Problems with scalability, integration, and privacy continue to be major obstacles.

New privacy-preserving technologies, interoperability frameworks, and Layer-2 protocols, however, show great promise in overcoming these constraints. Finally, data engineering processes may be built on top of blockchain technology, which offers a solid basis for security, transparency, and efficiency. Adopting it calls for forethought, specific use cases, and funding for solutions that can scale and work together. In order to fulfil the needs of a data-driven society, organizations must overcome these difficulties and then use blockchain technology to build data ecosystems that are robust and ready for the future.

Conflict of Interest
The Author’s declare that there is no conflict of Interest to report.

Funding Source
This research was entirely Self-funded by the Author’s.

Authors’ Contributions
Chittaranjan Pradhan, as the main author of this research paper and Abhishek Trehan has provided necessary support to every phase on this research paper as co-author.

References

[1] N. O. Nawari and S. Ravindran, "Blockchain technologies in BIM workflow environment," in ASCE International Conference on Computing in Civil Engineering 2019, Reston, VA, USA, American Society of Civil Engineers, Jun., pp.**343–352**, **2019.**
[2] M. Das, X. Tao, and J. C. Cheng, "A secure and distributed construction document management system using blockchain," in International Conference on Computing in Civil and Building Engineering, Cham, Switzerland, Springer International Publishing. Jul., pp.**850–862**, **2020.**
[3] R. Brandín and S. Abrishami, "IoT-BIM and blockchain integration for enhanced data traceability in offsite manufacturing," Automation in Construction, Vol.**159**, pp.**105-266**, **2024.**
[4] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," IEEE Transactions on Knowledge and Data Engineering, Vol.**30**, No.**7**, pp.**1366–1385, 2018.**
[5] X. Xu, I. Weber, and M. Staples, Architecture for Blockchain Applications, Cham, Switzerland: Springer, pp.**1–307, 2019.**
[6] E. Bandara, X. Liang, P. Foytik, S. Shetty, N. Ranasinghe, and K. De Zoysa, "Rahasak—Scalable blockchain architecture for enterprise applications," Journal of Systems Architecture, Vol.**116**, pp.**102061**, **2021.**

[7] C. V. B. Murthy, M. L. Shri, S. Kadry, and S. Lim, "Blockchain based cloud computing: Architecture and research challenges," IEEE Access, Vol.**8**, pp.**205190–205205, 2020.**

[8] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," IEEE Transactions on Computational Social Systems, Vol.**7**, No.**3**, pp.**790–801, 2020.**

[9] Z. Peng, H. Wu, B. Xiao, and S. Guo, "VQL: Providing query efficiency and data authenticity in blockchain systems," in 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), Apr., pp.**1–6, 2019.**

[10] V. Clincy and H. Shahriar, "Blockchain development platform comparison," in 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Jul., Vol.**1**, pp.**922–923, 2019.**

AUTHORS PROFILE

Chittaranjan Pradhan holds a B.Sc. in Physics, Chemistry, and Mathematics from Sambalpur University, an MBA in Information Technology from Amity University, and an M.Tech in Software Systems with a specialization in Data Analytics from BITS Pilani. His research interests lie primarily in Data Engineering and Big Data Analytics.

With over 28 years of extensive industry experience, Mr. Pradhan has demonstrated expertise in Software Engineering, Application Development, and Data Engineering. His professional journey includes 15 years in Singapore and over 9 years in the United States, where he held prominent roles such as Vice President, Senior Application Development Manager, Data Engineering Lead, Project Lead, and Developer. He has also managed large-scale production data platforms, consistently delivering innovative and scalable solutions across diverse industries.

Abhishek Trehan is a highly skilled Data Engineering & Data Science professional with over 14 years of industry experience in designing and implementing data solutions and optimizing data infrastructure. With a proven track record in AI and machine learning, he has successfully developed and executed strategies for large-scale enterprises, focusing on big data solutions and cloud strategy enablement. Abhishek is adept at building and leading data platforms that leverage advanced data analytics, artificial intelligence, and machine learning techniques, providing actionable insights and strategic direction.

Currently serving as Vice President at JPMorgan Chase & Co., Abhishek leads initiatives related to data migration to cloud platforms, development of regulatory data solutions, and AI/ML applications for risk and fraud analysis. His work on cloud migration, AI/ML for credit card risk, and financial data aggregation has been instrumental in optimizing workflows and ensuring compliance with regulatory standards.

Abhishek holds multiple advanced degrees, including a Master of Science in Data Science and Information Systems, both from the University of Delaware and University of Cincinnati. His academic excellence is highlighted by awards such as the "University Graduate Scholarship", "University Silver Medal" and multiple research publications & scientific journal views.

