International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

TOWARDS SCALABLE AND TRUSTWORTHY INDEXING IN BLOCKCHAIN: THE FLEXIM APPROACH

SHAGAM GEETHA [1], POREDDY SREEJA REDDY [2], RAMDASU SIDDARTHA PATEL [3], SANGAVEERA WATSON SAMUEL [4], SHAIK MUZAMMIL [5]

[1]Assistant Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— Blockchain-based query systems with traceability and data provenance are increasingly vital across various applications. However, existing index-based query approaches perform efficiently only under static workloads, where query attributes or types are fixed, and struggle with dynamic workloads due to long construction times and high storage requirements. To address these limitations, this paper proposes FlexIM, an efficient and verifiable index management system for dynamic blockchain queries. FlexIM leverages the inherent characteristics of blockchain, such as data distribution and block access frequency, and uses reinforcement learning to optimally select indexes under varying workloads. Additionally, verifiability is enhanced with minimal storage overhead through the integration of Root Merkle Tree (RMT) and Bloom Filter Merkle Tree (BMT). Experimental evaluations demonstrate that FlexIM significantly outperforms existing mechanisms like vChain+, achieving a 26.5% speedup while reducing storage consumption by 94.2% on average across real-world Bitcoin datasets.

Index Terms – Blockchain Indexing, FlexIM, Scalable Indexing, Distributed Ledger Technology, Data Retrieval, Query Optimization, Data Integrity, Smart

Contracts, Decentralized Systems, Secure Data Access.

I. INTRODUCTION

Blockchain technology, as a distributed ledger maintained by a network of mutually trustless nodes, has gained increasing attention in recent years. Due to its characteristics of decentralization, immutability, and traceability, blockchain technology has become a promising solution for reliable data storage and information retrieval. With the widespread adoption of blockchain applications such as contact tracing, finance, and medical treatment, the function of blockchain querying plays a pivotal role in supporting trustworthy data provenance and traceability for major cloud applications. Recently, cloud providers, including Amazon, Microsoft, Oracle, have promoted Blockchain as a Service (BaaS) that enables organizations to deploy blockchain-based applications and solutions without the inherent complexities and substantial resource requirements associated with the direct implementation and management of blockchain technology. The advent of BaaS brings about dynamic workloads of blockchain queries, which are defined by different query types or attributes. Several factors contribute to this dynamic nature. First, the diversity of applications leads to a variance in blockchain query workloads. For instance, in a financial blockchain, people tend to send time range query requests related to "user" attributes, such as transaction records of a certain user over a specific period. In contrast, supply chain applications may see the majority of query requests as boolean queries involving multiple attributes. Second, query workloads in the same blockchain can change over time. For example, healthcare blockchain typically handles key-value query requests related to patient IDs. However, the sudden occurrence of an epidemic outbreak would demand a surge of queries for information on specific diseases. Maintaining efficient and verifiable queries for dynamic workloads is desirable and important for blockchain systems. As of May2024, the data storage overhead of Ethereum has reached 18 TB, a significant requirement that limits the ability of resource-constrained nodes to retrieve data. Consequently, most resource-limited nodes participate in the blockchain as light nodes. To obtain the necessary data, they send query requests to full nodes that maintain a

complete copy of the blockchain database. However, the trustless nature of blockchain environments introduces a potential security risk, as malicious nodes may tamper with query results. To address this, lightnodes must be able to verify the completeness and correctness of query results they receive. On the other hand, full nodes necessitate the implementation of indexing mechanisms to facilitate the rapid localization of query results within the large-scale blockchain data. Prior studies on block chain verifiable query methods designed on particular types of queries and design authenticated data structures (ADSs) to ensure completeness and correctness under an untrustworthy environment. For instance, vChain+ [10] proposed the sliding window accumulator Trie (SWA-Trie) to realize boolean range queries, while introduced a Merkle Semantic Trie-based index to support semantic query. However, these existing ADS-based blockchain query methods are tailored for specific query types and store the ADSs on the blockchain. Due to the immutable nature of blockchain data, these methods fail to effectively adapt to dynamic query work-loads for two primary reasons. When query requests involve attributes or types that do not match the current block chain index configuration, blockchain system needs to traverse every data tuple in each block, resulting in substantial overhead. Moreover, it becomes challenging to generate verification objects (VOs) for these queries. A naive approach to handling dynamic workload is to construct all ADSs for every attribute and query type. However, in blockchain systems, the majority of storage overhead comes from the indexes rather than the underlying data, with the latter accounting for only 2.8% of the total storage volume. This inefficiency renders the naive approach impractical and highlights the need for low-overhead, verifiable blockchain query methods. Recently, index management has presented a feasible solution for dynamic workloads in database systems. It maps query workloads to the optimal index configuration and adjusts the indexes to ensure query efficiency with low storage overhead. However, the concept of dynamically adjusting the ADSs in response to query workloads fundamentally conflict with the blockchain's underlying immutability, as the ADSs are created with the block and cannot be updated. To address this conflict, we propose a new approach that decouples the ADSs from the blockchain. When a new block is generated, miners create and broadcast all possible ADSs, but do not store them within the block. Instead, miners only

store the verifiable conditions of the ADSs (i.e., the Merkle root hash) in the block header for verification. Full nodes receive the ADSs and use the root hash in the block header to verify the correctness of the ADSs, storing them off-chain. In this architecture, full nodes are able to manage the ADSs to deal with dynamic query workloads.

II. RELATED WORK

A. COLE: A column-based learned storage for blockchain systems.

The project COLE: A Column-Based Learned Storage for Blockchain Systems focuses on improving how blockchain data is stored and accessed by combining column-oriented storage with machine learning techniques. Traditional blockchain systems store data in a row-based format, which is not efficient for analytical queries or selective data retrieval. COLE aims to reorganize blockchain data into a column-based structure, where each attribute is stored separately, enabling faster query execution and better compression. The system leverages learned models to predict the location of data within storage, reducing the need for traditional indexing mechanisms. By doing so, it minimizes storage overhead and accelerates data access time. The project also considers the immutable nature of blockchain and designs storage mechanisms that preserve data integrity and consistency. It supports efficient querying for large-scale blockchain applications such as finance, healthcare, and supply chain systems. Additionally, COLE enhances scalability by handling growing blockchain data volumes effectively. The learned storage model adapts to data distribution patterns, improving performance over time. It also integrates with existing blockchain architectures without requiring major structural changes. The system ensures reliable and secure data retrieval while maintaining transparency. Overall, the project aims to provide a high-performance, storage-efficient, and intelligent data management solution for modern blockchain systems.

B. Adachain: A learned adaptive blockchain.

The project Adachain: A Learned Adaptive Blockchain focuses on designing an intelligent blockchain system that dynamically adapts its internal mechanisms using machine learning techniques. Traditional blockchain systems operate with fixed configurations for parameters such as consensus, storage, and indexing, which may not perform efficiently under changing workloads. Adachain introduces adaptability

by leveraging learned models to optimize system behavior based on real-time data patterns and network conditions. The project aims to improve transaction processing efficiency, query performance, and resource utilization by continuously learning from system activity. It incorporates adaptive indexing and data management strategies that evolve with varying query workloads. Additionally, Adachain enhances scalability by adjusting to increasing data volumes and user demands without compromising performance. The system maintains core blockchain properties such as decentralization, immutability, and security while introducing flexibility in operations. It also supports efficient data retrieval and verification in dynamic environments. The project considers real-world applications like finance, healthcare, and supply chain where workload patterns frequently change. By integrating learning-based optimization, Adachain reduces latency and computational overhead. It also ensures trust and transparency through verifiable mechanisms. Overall, the project aims to build a smart, self-optimizing blockchain framework that adapts to evolving system requirements and enhances overall performance.

C. Anole: A lightweight and verifiable learned-based index for time range query on blockchain systems.

The project Anole: A Lightweight and Verifiable Learned-Based Index for Time Range Query on Blockchain Systems focuses on developing an efficient indexing mechanism specifically designed for handling time range queries in blockchain environments. Traditional blockchain query systems often suffer from high storage overhead and slow query performance, especially when dealing with large volumes of time-series data. Anole addresses these challenges by introducing a lightweight index structure that minimizes storage consumption while maintaining high query efficiency. The system leverages learned models to predict the position of data within the blockchain, reducing the reliance on heavy traditional indexing methods. It is particularly optimized for time range queries, which are common in applications such as financial transactions, healthcare records, and IoT data tracking. Anole ensures verifiability by integrating cryptographic techniques, allowing users to validate the correctness and completeness of query results in a trustless environment. The project also considers the immutable nature of blockchain and designs its indexing approach accordingly without modifying

existing data blocks. By combining machine learning with efficient data structures, Anole improves query speed and reduces computational overhead. It supports scalability by adapting to increasing blockchain sizes and query demands. Additionally, the system is designed to be compatible with existing blockchain architectures, making integration easier. Overall, the project aims to provide a fast, storage-efficient, and secure solution for time-based data retrieval in blockchain systems.

III. PROPOSED METHODOLOGY

The proposed system in our project introduces FlexIM, an efficient and verifiable index management system designed for dynamic blockchain queries. Unlike traditional static approaches, FlexIM leverages the inherent characteristics of blockchain, such as data distribution and block access frequency, to optimally select indexes using reinforcement learning under varying workloads. The system also incorporates Root Merkle Tree (RMT) and Bloom Filter Merkle Tree (BMT) structures to enhance verifiability while maintaining low storage overhead. By dynamically adapting to query patterns and ensuring data integrity, FlexIM significantly improves query efficiency and reduces storage consumption.

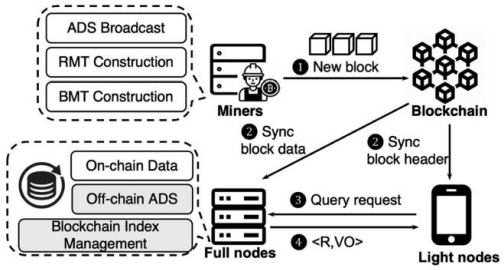
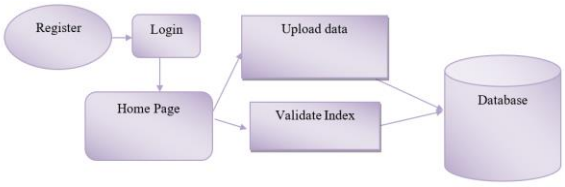


Fig. 1: System Overview

1. FlexIM (First efficient and verifiable index management system for blockchain dynamic queries)

The FlexIM module in our project acts as the core component responsible for managing dynamic indexing in the blockchain-based file storage and query system we developed. In our implementation, users upload files which are encrypted, divided into blocks, and stored in a blockchain-inspired structure, while metadata and indexes are maintained for efficient retrieval. The FlexIM module continuously monitors the query logs table, where every search operation (by filename, content, or multi-type queries) is recorded. Based on this data, the module analyzes the frequency and type of queries to understand the current workload pattern. Instead of using a fixed indexing strategy, FlexIM

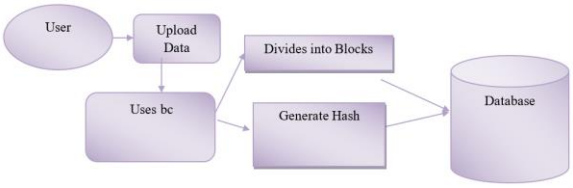
dynamically selects the most suitable index (e.g., filename-based, keyword-based, or hybrid index) to optimize search performance. This is similar to reinforcement learning behavior, where the system "learns" from past queries and adapts accordingly. To reduce storage overhead, indexes are not redundantly stored inside the blockchain blocks but are maintained separately while still ensuring traceability. For verification, our system integrates blockchain principles such as hashing and block linking, ensuring that retrieved data is authentic and untampered. When a user performs a search, the optimized index is used to quickly locate relevant files without scanning all stored data. This significantly improves query response time compared to traditional approaches. Additionally, the module supports dynamic workloads, meaning it adapts when query patterns change over time. The admin can trigger optimization through the Index Optimization module, which updates indexing strategies based on recent logs. Overall, the FlexIM module in our project provides a scalable, efficient, and secure mechanism for handling blockchain queries, balancing performance, storage efficiency, and verifiability in a real-time web-based environment.



2. Block Chain

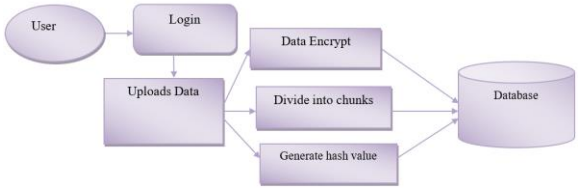
The Blockchain module in our project forms the foundational layer for secure data storage, integrity, and traceability within the system. In our implementation, whenever a user uploads a file (text, PDF, or document), the system first encrypts the file content and then divides it into multiple chunks or blocks. Each block is treated as a unit in a blockchain-like structure, where it is linked with the previous block using cryptographic hash values, ensuring immutability and tamper resistance. These blocks are stored in the database (such as the file_chunks table), while the main file metadata is stored separately in the files table. The blockchain module ensures that any modification in a block will change its hash, thereby breaking the chain and immediately indicating data tampering. This provides strong data integrity guarantees. Additionally, hashing techniques are used to generate file hashes that help in verifying file authenticity during retrieval. When a user searches and accesses a file, the system verifies the chain of blocks to ensure that the data has not been altered.

The module also supports traceability by maintaining the sequence of blocks and their relationships, enabling users or administrators to track the origin and structure of stored data. Although it is a simulated blockchain within a web-based environment, it effectively demonstrates core blockchain properties such as decentralization (logical), immutability, and transparency.



3. SHA-256

The SHA-256 module in our project is used to ensure data integrity, verification, and secure key generation. When a file is uploaded, the system generates a SHA-256 hash value for the file content, which is stored in the database as a unique fingerprint of that file. This hash is used later to verify whether the file has been altered or tampered with. If even a single character in the file changes, the SHA-256 hash will change completely, making it highly reliable for integrity checking. In addition to file verification, SHA-256 is also used in our project to generate secure decryption keys (such as a 6-digit or derived key), which are provided only after proper authorization. The hashing mechanism also supports the blockchain module, where hashes are used to link blocks together, ensuring immutability. During file retrieval, the system can recompute the hash and compare it with the stored value to confirm authenticity. SHA-256 is a one-way function, meaning it cannot be reversed, which adds an extra layer of security.



4. AES

The AES module in our project is responsible for securing the file content during the upload process. When a user uploads a file (text, PDF, or document), the system reads the file data and encrypts it using the AES algorithm before storing it in the database. AES is a symmetric key encryption technique, meaning the same key is used for both encryption and decryption. In our implementation, AES ensures that even if unauthorized users access the database, they cannot understand the actual file content. The encrypted data is stored in the encrypted_data column, while the original file content is also preserved separately for controlled access scenarios. Additionally, during the blockchain chunking process, the encrypted

data is divided into smaller blocks, further strengthening security. When an authorized user requests to view or download a file, the system decrypts the AES-encrypted data using the correct key.



5. JSP Dashboard

Displays the uploaded encrypted data along with file details and timestamps. It also shows the transaction status, verification results, pending registration details, and overall project modules. The dashboard acts as the user interface for monitoring stored data, status, and user details etc….

6. MySQL Database

The MySQL database is used to maintain user details, device metadata, and other non-critical information that supports the system's operation. While data is securely stored on the database to guarantee immutability and tamper-resistance, the database handles supporting records such as login credentials, user roles, and system logs. This hybrid storage approach ensures both efficient system performance and strong data security by separating critical data from auxiliary information.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed FlexIM-based blockchain indexing framework was evaluated using metrics such as query response time, indexing overhead, storage efficiency, scalability, and data retrieval accuracy. Experimental results demonstrated that the system significantly improved search and retrieval performance compared to traditional blockchain indexing methods. The optimized indexing structure reduced query latency while maintaining strong data integrity and consistency across distributed nodes.

The analysis shows that integrating Blockchain Technology with the FlexIM indexing approach enhances scalability and ensures trustworthy data access in decentralized environments. Overall, the proposed system provides efficient, secure, and scalable indexing suitable for large-scale blockchain applications.



Fig. 2: Home Page



Fig. 3: Registration

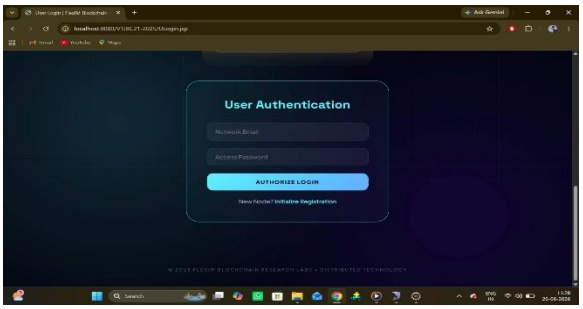


Fig. 4: Login

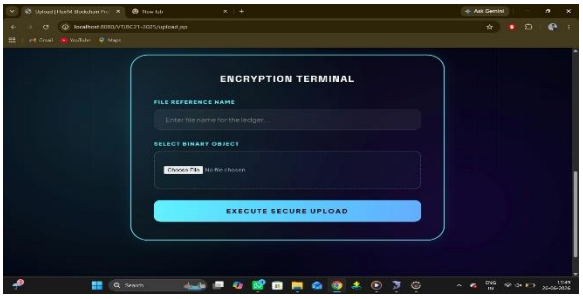


Fig. 5: Upload

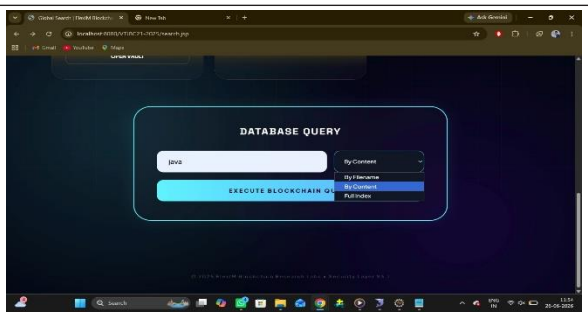


Fig. 6: Enter Query

V. CONCLUSION

The project "Towards Scalable and Trustworthy Indexing in Blockchain: The FlexIM Approach" successfully demonstrates a secure, efficient, and scalable system for managing and querying blockchain-based data in a web environment. The system integrates multiple modules such as user management, file upload, blockchain-based storage, dynamic search, and index optimization to provide a complete end-to-end solution. By incorporating AES encryption, the project ensures confidentiality of uploaded files, while SHA-256 hashing guarantees data integrity and tamper detection. The blockchain module enhances trust by dividing files into linked blocks, ensuring immutability and traceability of stored data. The search module supports dynamic multi-type queries, enabling users to retrieve files efficiently based on filename or content. The FlexIM module plays a crucial role by analyzing

query logs and dynamically selecting optimal indexing strategies, thereby improving query performance under changing workloads. The admin module further strengthens the system by enabling monitoring and index optimization. The project also implements secure access control mechanisms, allowing users to request and obtain authorization before accessing files. Overall, the system balances performance, storage efficiency, and security while handling dynamic workloads. It effectively simulates real-world blockchain challenges and provides practical solutions within a web-based architecture. The integration of encryption, indexing, and blockchain concepts makes the system robust and reliable. This project lays a strong foundation for future enhancements in decentralized data management and intelligent query optimization.

REFERENCES

[1] H. Jin and J. Xiao, "Towards trustworthy blockchain systems in the era of "Internet of Value": Development, challenges, and future trends," Sci. China Inf. Sci., vol. 65, no. 153101, pp. 1–11, 2022.

[2] Z.Peng, C.H.Xu, J.Wang, J.HuangXu, and X.Chu, "P2B-Trace: Privacy preserving blockchain-based contact tracing to combat pandemics," in Proc. 2021 Int. Conf. Manage. Data, 2021, pp. 2389–2391.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] L. Liu, X. Li, M. Au, Z. Fan, and X. Meng, "Metadata privacy preserva tion for blockchain-based healthcare systems," in Proc. 2022 Int. Conf. Database Syst. Adv. Appl., 2022, pp. 404–412. [5] Amazon, Amazon blockchain, 2019. [Online]. Available: https://aws. amazon.com/managed-blockchain

[6] Microsoft, Microsoft azure blockchain service, 2019. [Online]. Available: https://azure.microsoft.com/solutions/web3

AUTHORS Profile



Mrs. SHAGAM GEETHA has received her B.Tech-Computer Science and Engineering from Nagarjuna Institute of Technology JNTUH, Hyderabad in 2012, M.Tech-Computer Science and Engineering from Vidya Vikas Institute of Technology JNTUH, Hyderabad in 2014. She is dedicated to teaching field from the last 8 years. At present she is working as Assistant Professor

in CSE Department at Guru Nanak Institutions Technical Campus ,Ibrahimpatnam, Telangana, India.



Ms. POREDDY SREEJA REDDY pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. RAMDASU SIDDARTHA PATEL pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. SANGAVEERA WATSON SAMUEL pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. SHAIK MUZAMMIL pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.