Provenance Guard: Leveraging IPFS And ScalableBlockchain For Secure And Transparent Data Provenance

Prof. Pooja Shettar[1*], Prof. Manjula Pawar[2], Supreeth S Kadappanavar[3], Prajakta Kulkarni[4]

[1*]Department of Computer Science, KLE Technological University, Hubballi, India,
Email: poojashettar@kletech.ac.in
[2]Department of Computer Science, KLE Technological University, Hubballi, India,
Email: manjulap@kletech.ac.in
[3]Department of Computer Science, KLE Technological University, Hubballi, India,
Email: supreeth.chintzz@gmail.com
[4]Department of Computer Science, KLE Technological University, Hubballi, India,
Email: prajaktakulkarni962@email.com

***Corresponding Author:** Prof. Pooja Shettar

*Department of Computer Science, KLE Technological University, Hubballi, India,
Email: poojashettar@kletech.ac.in

Abstract:
Data provenance in the cloud environment faces significant challenges due to concerns over storage and processing management. In this paper, we propose a scalable data provenance solution employing Blockchain andthe InterPlanetary File System (IPFS). The system leverages the distributed Ethereum ledger for storing provenance data and associated hashes, while IPFS manages entry record hashes. This separation enhances mutual trust between cloud service providers and users, improving data management transparency and security. We implemented our solution as a full-stack web application using the MERN stack (MongoDB, Express, React, and Node.js), integratedwith a local Geth instance running a modified proof-of-work algorithm. The application communicates with the Geth instance via the Web3.js library, using Solidity smart contracts to link the application and the blockchain.Our performance analysis results demonstrate improved transaction latency and overall performance, indicating the system's effectiveness in addressing data provenance challenges in a cloud environment. Future work involves securing the mapping file, extending the service to all user-uploaded files, and investigating scalability and performance in a larger blockchain network. We also plan to examine the potential of Hyperledger for managing data provenance andthe feasibility of implementing Blockchain as a Service (BaaS) provided by cloud service providers.

Index terms: Data Provenance, Scalable Blockchain, Data Integrity, Transparency, Auditability

I. INTRODUCTION

Blockchain technology has experienced significant growth since its conception in 2009, evolving from a platform for cryptocurrencies to a versatile solution with applications across various industries. As a decentralized, immutable, and public ledger, blockchain offers numerous benefits, including increased security, transparency, and reliability. The decentral- ized nature of the blockchain network eliminates the need for a central authority, raising questions about verifying user trans- actions. Consensus protocols help validate transactions, pre-vent fraud, and maintain network security, making blockchain a practical choice for businesses and real-world applications. Data provenance plays a crucial role in ensuring the reliability of information as the volume of data being collected grows exponentially. By providing a historical record of data from its origin, data provenance helps users and cloud service providersestablish trust. As a result, designing secure data provenancemodels is essential for cloud storage and detecting malicious activities.

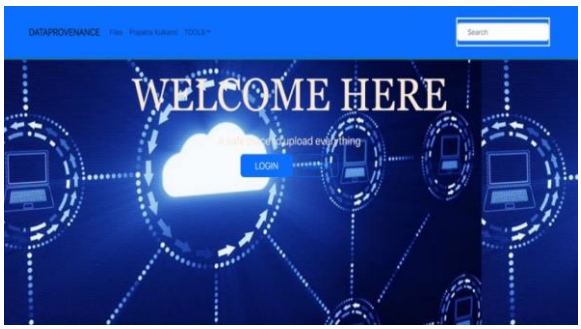

Fig. 1: Homepage of Our Application

Blockchain technology offers a tamper-proof environment for implementing data provenance, enhancing data account- ability, transparency, privacy, and accessibility in the cloud. However, data storage on the blockchain can be expensive, and the Interplanetary File System (IPFS) emerges as an alternative solution. IPFS is a secure, decentralized, and cost-effective peer-to-peer storage network that requires no servers, allowingusers to distribute their work without incurring additional costs[6,7,8,9,10].

In this research, we propose a novel approach to data provenance using scalable blockchain technology combined with the benefits of IPFS for efficient storage. By employing an effective proof-of-work consensus mechanism, our solution can be applied to a wide range of real-life use cases. The con- sensus contributes to data security through decentralized and immutable storage, requires fewer resources for transaction processing, and improves transaction latency.T

II. LITERATURE SURVEY

Blockchain technology has been applied to various aspects of data provenance, aiming to enhance security, transparency, and efficiency. This section presents a review of some notable works in the field, highlighting their contributions and identifying research gaps.

A. Data Provenance

Smart Provenance introduced a decentralized, blockchain- driven data provenance system that harnessed smart con- tracts and an open provenance model (OPM) for recording unchangeable data trails. Although the framework securely captured and verified provenance data, it depended on Google Drive for data storage rather than utilizing IPFS. Additionally, no specific consensus mechanism was implemented.

In another study, the authors examined the challenges and possibilities of applying consensus protocols to blockchain- based data provenance. They introduced BlockCloud, a frame- work incorporating a Proof-of-Stake (PoS) consensus en- gine for blockchain-based data provenance within a federated cloud environment. Nonetheless, incorp- orating PoS distributedledger technology into the cloud computing domain posed difficulties.

B. Blockchain for Data Provenance

Smart Provenance [2] introduced a distributed, blockchain- based data provenance system that leveraged smart contracts and an open provenance model (OPM) to record immutable data trails. While the framework securely captured and val- idated provenance data, it relied on Google Drive for data storage and did not utilize IPFS. Furthermore, no explicit consensus mechanism was employed.

In [3], the authors discussed the challenges and oppor- tunities of using consensus protocols for blockchain-based data provenance. They presented BlockCloud, a framework that incorporated a Proof-of-Stake (PoS) consensus engine for blockchain-based data provenance in a federated cloud environment. However, integrating PoS distributed ledger tech- nology in the cloud computing domain proved challenging.

C. IPFS and Data Provenance

The work in [4] proposed an IoT data provenance frame- work based on blockchain and smart contracts. The frameworkstored IoT device data off-chain, and cryptographic hashes of device metadata were stored in the blockchain, enhancing scalability. The authors utilized the MD5 hashing function, which is cryptographically broken and no longer secure.

Cloud PoS [5] presented a blockchain-based data prove- nance framework for the Blockchain-as-a-Service (BaaS) cloud

computing environment using a novel PoS consensus model. The process allowed a fair chance for each cloud user to lead the blockchain based on their staked resources. However, the stake allocation and verification process requireddynamic adjustment of actual resources, which could leadto extra delays before validators could verify the staked resources.

D. Scalable Blockchain

The literature reveals the potential for further research in thearea of data provenance using scalable blockchain technology. The integration of IPFS as a cost-effective and decentralized storage solution, along with more secure and efficient consen- sus mechanisms, can contribute to the development of novel approaches in the field.

III. PROPOSED METHOD

A. Architecture Design

The proposed architecture design takes into account user interactions within the system, encompassing file tasks such as uploading, downloading, and removing files. When a user signs up for an account and logs in, metadata is produced for any file activities they undertake. This metadata is connected to the Blockchain network and IPFS. The provenance databaseoversees IPFS, and upon a block's completion of the validation procedure on the blockchain, a receipt is generated and sent back to the provenance database. The provenance entry in IPFS is merged with the receipt. Although the provenance database monitors user actions, it only retains the hashed identities of users, thus preserving privacy. A high-level architecture designis illustrated in Figure 2.

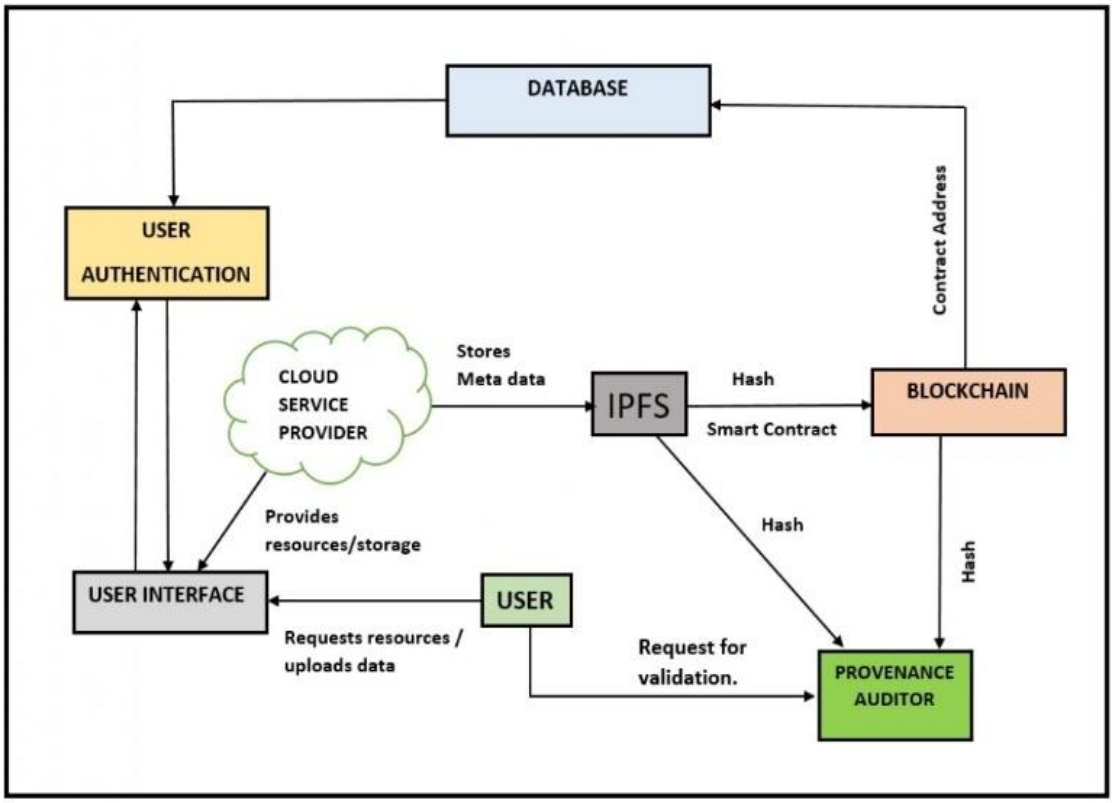

Fig. 2: High-Level Architecture Design

B. Sequence Diagram

The sequence diagram demonstrates the process followed when a user performs file operations, such as uploads, downloads, and deletions. Metadata is generated and stored in IPFS, while hashes of the metadata file are saved in the blockchain. The user logs in using their credentials, and when they log out and request validation, the auditor retrieves the hash from IPFS and the blockchain, sending the user the status. The sequence diagram is shown in Figure 3.

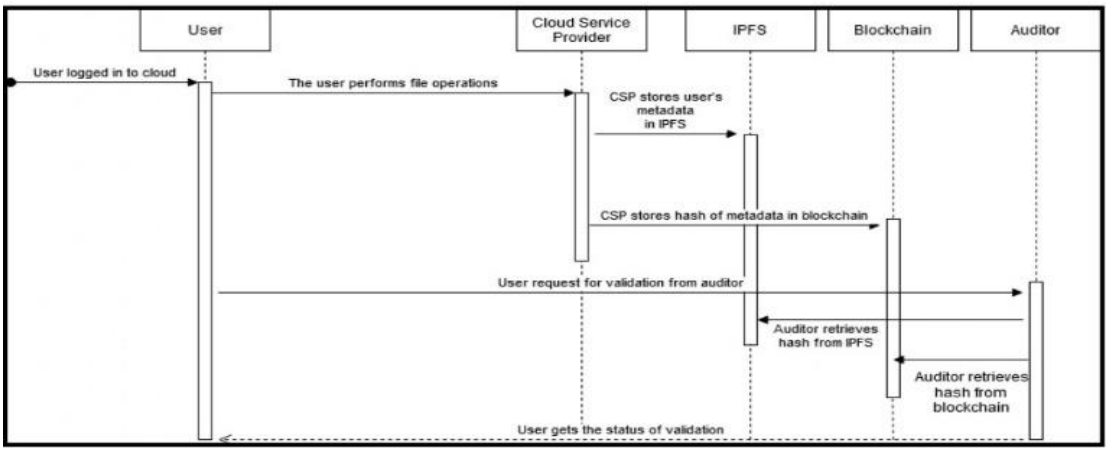

Fig. 3: Sequence Followed by the System

C. Activity Diagram

The activity diagram represents the actions that take place when a user performs operations on files. The user's actions (uploads, downloads, and deletions) are considered input, while the metadata log generation is the output. The activity diagram is depicted in Figure 4.

B. Software Requirements

The proposed system requires the following software com-ponents:

- Ubuntu 20.04 LTS or higher, or any C-based operatingsystem
- NodeJS 16.14.2 or higher
- IPFS 0.13 or higher
- React
- Ethereum Blockchain (Geth 1.10.17 or higher)

V. IMPLEMENTATION

In this section, we describe the implementation of our proposed system using three algorithms. These algorithms facilitate the interaction between IPFS and blockchain, vali- date file integrity, and implement a modified proof-of-work algorithm.

A. Algorithm 1: IPFS and Blockchain Interaction

Algorithm 1 IPFS and Blockchain Interaction	
1.	**procedure** INTERACTION(*file, action, user*)
2.	*Metadata*←generate Metadata (*file, action, user*)
3.	*IPFS Hash*← add To IPFS(*metadata*)
4.	*Blockc hain Hash* ← add To Block chain (*ipfs Hash*)
5.	**return** *blockchainHash*
6.	**end procedure**

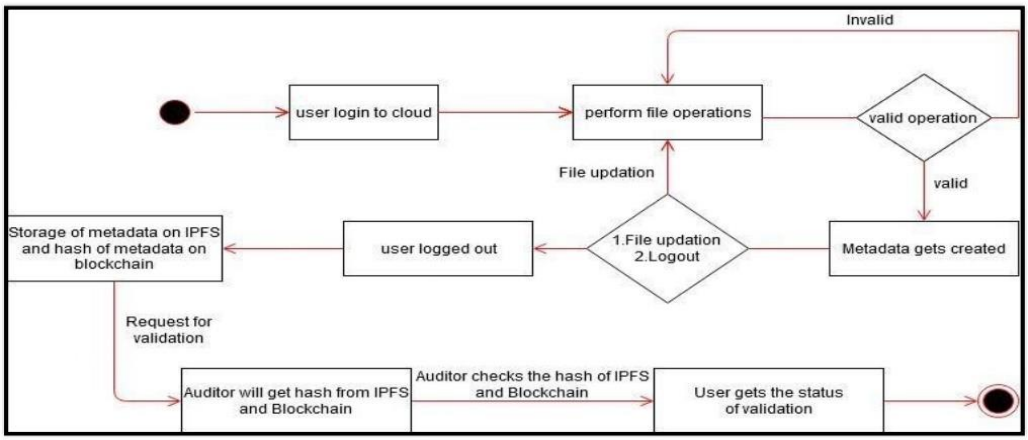

Fig. 4: Activity Diagram

In the proposed method, we take into account the user's interactions with the system and their file operations, as well asthe architecture and sequence diagrams. This approach ensures the privacy and security of user data by using IPFS for de- centralized storage and blockchain technology for immutable record-keeping.

IV. HARDWARE AND SOFTWARE REQUIREMENTS

A. Hardware Requirements

The following hardware requirements are necessary for the proposed system:

- A working PC/Laptop
- Minimum of 8GB RAM
- Minimum of 512GB Hard Disk
- Intel Core i5 or higher processor

The first algorithm describes the interaction between IPFS and blockchain for the proposed system. This interaction enables the storage of metadata in IPFS and the creation of immutable records in the blockchain.

B. Algorithm 2: Validate File Integrity

The second algorithm is responsible for validating the integrity of a file by comparing its hash with the one stored in the blockchain. This ensures that the file has not been tampered with.

Algorithm 2 Validate File Integrity	
1.	**Procedure** VALIDATEIN- TEGRITY (*file, block chain Hash*)
2.	*Computed Hash* ← compute File Hash (*file*)
3.	*Stored Hash*← get Hash From Block chain (*block chain Hash*)
4.	**if** *computed Hash == stored Hash* **then**
5.	**return** True
6.	**else**
7.	**return** False
8.	**end if**
9.	**end procedure**

C. Algorithm 3: Modified Proof-of-Work Algorithm

The third algorithm presents a modified proof-of-work algorithm for the proposed system. This modification aims to enhance the efficiency and security of the system while maintaining its decentralized nature.

Algorithm 3 Modified Proof-of-Work Algorithm	
1.	**Procedure** MODIFIEDPOW (*block, difficulty*)
2.	*Nonce* 0
3.	*New Difficulty*← calculate New Difficulty (*difficulty*)
4.	**while** True **do**
5.	*Hash*← compute Block Hash (*block, nonce*)
6.	**if** is Hash Valid (*hash, new Difficulty*) **then**
7.	*block.* Set Nonce (*nonce*)
8.	*block.* Set Hash (*hash*)
9.	**return** *block*
10.	**end if**
11.	*Nonce* ← *nonce* + 1
12.	**end while**
13.	**end procedure**

VI. SETUP AND EVALUATION

In this section, we outline the configuration of our system and the assessment of its performance. We have developed a full-stack web application employing the MERN (MongoDB, Express, React, and Node.js) stack, which is connected to a local Geth instance executing the modified proof-of-work

algorithm. The web application communicates with the local Geth instance via the Web3.js library and leverages Solidity smart contracts to establish a connection between the web application and the local Geth blockchain.

A. Setup

We have set up a full-stack web application using the MERN stack, which consists of MongoDB as the database, Express asthe web application framework, React as the front-end library,and Node.js as the runtime environment. The web application is connected to a local Geth instance, which runs our modified proof-of-work algorithm.

The local Geth instance is configured to work with the Ethereum blockchain using the modified proof-of-work con- sensus algorithm. Web3.js library is used to interact with the local Geth instance and to manage the communication between the web application and the blockchain. Solidity smart contracts are employed to ensure the secure and decentralized nature of the system.

The web application provides users with an intuitive inter- face to interact with the blockchain, allowing them to perform operations such as uploading, downloading, and deleting files while maintaining the data provenance. The application also features an authentication system to ensure that only autho- rized users can access and interact with the stored data.

To ensure the security of the data, the application utilizes IPFS to store files in a decentralized manner. This ensures that no single point of failure exists within the system, makingit more resilient to attacks or data loss. The system also implements a modified proof-of-work consensus algorithm, which reduces the difficulty of the mining process, allowing for faster transaction processing and increased throughput.

B. Evaluation

Blockchain's primary function is to execute transactions, which involve reading or writing data onto the chain. In our assessment, we utilized the Proof of Work consensus algorithm with a difficulty level of 0x00. Our experiments with 5, 10, 15,and 20 transactions revealed that transaction latency increased linearly with the load, indicating that the work completed is directly proportional to the time required.

Employing the modified Proof-of-Work consensus algo-rithm, we observed a decrease in the time needed for trans- actions, resulting in enhanced transaction latency speed. In comparison to the original PoW consensus graph shown in Figure 5, the modified PoW algorithm executes transactions 16.3 percentage faster. Consequently, the system can achieve more work in less time.

VII. RESULTS

In this segment, we showcase the outcomes of our system assessment, emphasizing the influence of diverse elements on the system's performance, including factors like the quantityof nodes, latency, levels of difficulty, and IPFS-related latency.

A. Latency vs. Number of Nodes

The growing number of nodes in the Blockchain network contributes to its scalability over time. However, this also leads to an increase in network transaction fees and latency, while the read latency remains relatively constant. In our evaluation,we added three nodes to the system and observed that the proposed mechanism for data provenance suffers from poor performance due to the increased number of nodes.

With the help of the modified PoW consensus algorithm, the latency is comparatively lower than the original Proof-of- Work, even when the number of nodes is increased. Whena single node is taken into consideration, the latency is reduced by 0.28 seconds. Although this reduction may seem insignificant, increasing the number of nodes results in a latency decrease of over 1 second, contributing to a significant increase in the performance of the nodes.

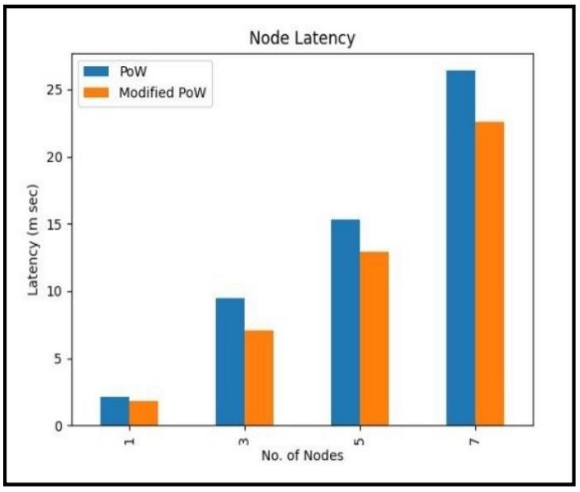

Fig. 5: Latency vs. Number of nodes

B. Latency vs. Puzzle Difficulty

Difficulty levels indicate how many zeros must be appended to a hash, making miners work harder to solve a more complex puzzle. As the difficulty level increases, so does the transaction latency. In our evaluation, we configured two nodes for this scenario, as shown in Figure 5.

As the difficulty level increases, the latency remains low, as seen in the graph in Figure 5, in comparison to Figure 4.

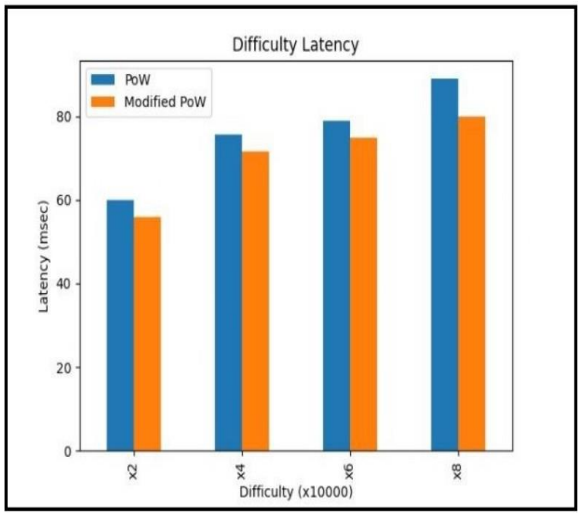

Fig. 6: Latency vs. Puzzle Difficulty

C. IPFS Latency

IPFS latency refers to the total time taken by IPFS nodes to store a file over a P2P network. In our evaluation, we varied the size of the files and calculated the latency of the file operations, as shown in Figure 9.

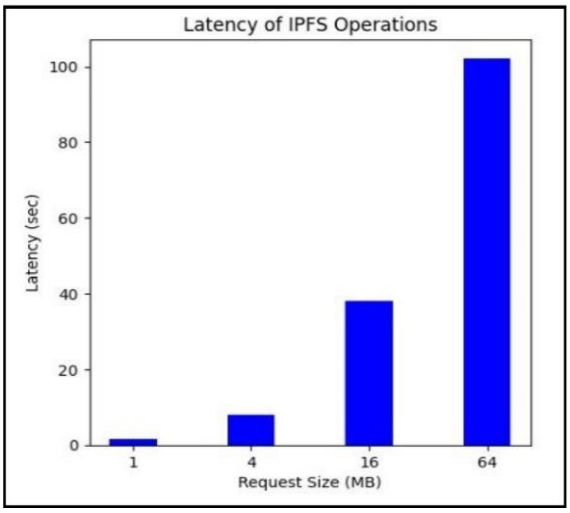

Fig. 7: IPFS Latency

D. Web Application Results

We built a full-stack web application using the MERN stack (MongoDB, Express, React, and Node.js) to provide users with an intuitive interface to interact with the system. The web application integrates with a local Geth instance that employs the modified PoW consensus algorithm. We utilized Web3.js and Solidity smart contracts to establish a connection between the web application and the local Geth instance.

1) *Node.js and Web3.js Integration:* Node.js served as the backbone of our server-side implementation, enabling seamless communication between the web application and the Ethereum Blockchain. We leveraged the Web3.js library to facilitate interactions with the Ethereum smart contracts, allowing users to perform various file operations, such as uploading, downloading, and deleting, while maintaining a record of the corresponding metadata on the blockchain.
2) *React-based User Interface:* The web application's user interface was built using the React.js library, providing a responsive and user-friendly experience. The application al- lowed users to perform file operations with ease, while the system simultaneously managed the data provenance aspects, including the IPFS and Blockchain interactions.

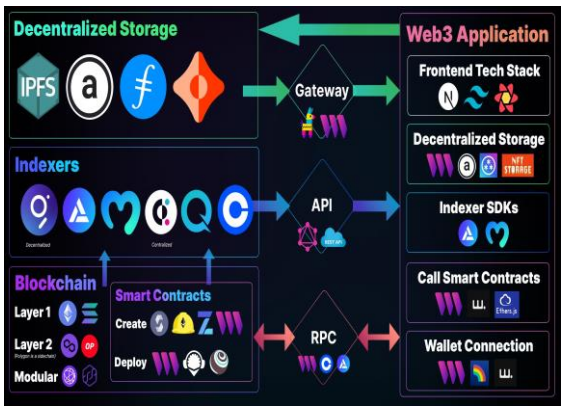

Fig. 8: Working of The Web3 Apps. Image source: https://blog.jarrodwatts.com/

3) Performance and User Experience: The web applica- tion's performance and user experience were evaluated based on the efficiency and responsiveness of the interface, as well as the latency of file operations in conjunction with the modified PoW consensus algorithm. Our evaluation indicated that the web application performed efficiently, with minimal latency in file operations, even when the number of nodes and difficulty levels increased.

In conclusion, the integration of Node.js, Web3.js, and the MERN stack, along with the modified PoW consensus algorithm and IPFS, contributed to the overall performance and user experience of our data provenance system in a cloud environment.

By analyzing the results, we can conclude that our system performs efficiently, even when the number of nodes and diffi- culty levels increase. The modified PoW consensus algorithm and the use of IPFS contribute to the overall performance and scalability of our data provenance system in a cloud environment.

VIII. Conclusion And Future Scope

In this paper, we introduced the design and execution of a data access mechanism that employs Blockchain and IPFS technologies. Our cloud-based solution utilizes a distributed Ethereum ledger for storing provenance data and its cor- responding hash, while IPFS handles the hashes for entry records. This division of responsibilities fosters trust between CSPs and users. We performed performance analysis under a variety of scenarios to assess the efficacy of our proposed system.

For future work, we aim to improve the security of the mapping file and broaden the provenance service to include all files uploaded by users. Moreover, we plan to conduct performance analysis on a more extensive blockchain networkto evaluate the scalability and resilience of our approach.

Additionally, we will explore the potential of Hyperledger as a more efficient method for managing data provenance, given its permissioned blockchain infrastructure and the associated advantages of enhanced privacy and scalability. We also intendto examine the viability of adopting Blockchain as a Service (BaaS) offered by cloud service providers, which could lead to greater efficiency and simplified implementation of our proposed system. By integrating these advancements, we hopeto strengthen the overall performance, scalability, and security of our data provenance solution in a cloud setting.

REFERENCES

1. X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, ProvChain: A Blockchain- Based Data Provenance Architecture in Cloud Environment with En- hanced Privacy and Availability," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2017, pp. 468–477.
2. S. Singh and S. Kim, Smart Provenance: A Distributed, Blockchain Based Data Provenance System," *arXiv preprint arXiv:1802.06893*, 2018.
3. J. Jang-Jaccard and S. Nepal, "Consensus protocols for blockchain- based data provenance: Challenges and opportunities," *IEEE Cloud Computing*, vol.5, no. 4, pp. 32–39, 2018.
4. A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts," *IEEE Access*, vol. 6, pp. 50718–50734, 2018.
5. S. K. Datta, R. P. Pateriya, and S. K. Rathi, Cloud PoS: A Proof-of-Stake Consensus Design for Blockchain Integrated Cloud,"

Future Generation Computer Systems, vol. 110, pp. 1004–1022, 2020.

6. Pawar M.K., Patil P., Hiremath P.S. (2021) A Study on Blockchain Scalability. In: Tuba M., Akashe S., Joshi A. (eds) ICT Systems and Sustainability. Advances in Intelligent Systems and Computing, vol 1270. Springer, Singapore.
7. Pawar M.K., Patil P., Hiremath P.S., Hegde V.S., Agarwal S., Naveenku- mar P.B. (2021) "Scalable Blockchain Framework for a Food Supply Chain". In: Thampi S.M., Gelenbe E., Atiquzzaman M., Chaudhary V., Li KC. (eds) Advances in Computing and Network Communications. Lecture Notes in Electrical Engineering, vol 735. Springer, Singapore. https://doi.org/10.1007/978-981-33-6977-1 35.
8. M. K. Pawar, P. Patil, M. Sharma and M. Chalageri, "Secure and Scalable Decentralized Supply Chain Management Using Ethereum and IPFS Platform," 2021 International Conference on Intelligent Technolo- gies (CONIT), 2021, pp. 1-5, doi:10.1109/CONIT51480.2021.9498537 .
9. Pawar, M.K., Patil, P., Patel, A.S. (2023). Distributed and Scalable Healthcare Data Storage Using Blockchain and KNN Classification.*I*n: Zhang, YD., Senjyu, T., So-In, C., Joshi, A. (eds) Smart Trends in Computing and Communications. Lecture Notes in Networks and Systems, vol 396. Springer, Singapore. https://doi.org/10.1007/978-981- 16-9967-2 70.
10. M. K. Pawar, P. Patil, R. Sawhney, P. Gumathanavar, S. Hegde and K. Maremmagol, "Performance Analysis of E-Certificate Generation and Verification using Blockchain and IPFS," 2022 International Conference on Inventive Computation Technologies (ICICT), 2022, pp. 345-350, doi:10.1109/ICICT54344.2022.9850830.