International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Implementing a Block Chain-Based Secure File Storage System with Enhanced User Authentication

MRS P SWAROOPA [1], DEVIREDDY VARDHAN REDDY [2], ANNAREDDY ANUSHKA REDDY [3] , GORLAPALLY BHEEM RAO[4] , ADUNURI SURENDER [5]

[1] MRS P SWAROOPA, ASSISTANT PROFESSOR, COMPUTER SCIENCE AND ENGINEERING,ACE ENGINEERING COLLEGE
[2] DEVIREDDY VARDHAN REDDY, STUDENT COMPUTER SCIENCE AND ENGINEERING, ACE ENGINEERING COLLEGE
[3] ANNAREDDY ANUSHKA REDDY, STUDENT COMPUTER SCIENCE AND ENGINEERING, ACE ENGINEERING COLLEGE
[4] GORLAPALLY BHEEM RAO, STUDENT COMPUTER SCIENCE AND ENGINEERING, ACE ENGINEERING COLLEGE
[5] ADUNURI SURENDER ,STUDENT COMPUTER SCIENCE AND ENGINEERING , ACE ENGINEERING COLLEGE

***Abstract*—File storage platforms face inherent challenges such as censorship, limited transparency, vulnerability to single points of failure, and restricted user control over data. To address these limitations, this paper proposes a decentralized file-sharing system that integrates the Ethereum blockchain with the InterPlanetary File System (IPFS). Our design leverages smart contracts to securely manage file metadata and enforce access controls, providing an immutable and tamper-resistant record of data ownership and permissions. IPFS is utilized for efficient, distributed file storage, enhancing scalability and availability. User authentication is handled through wallet-based cryptographic verification, eliminating reliance on centralized identity providers. Additionally, the system supports micropayment-based monetization via smart contracts, enabling direct and transparent transactions between content creators and consumers. The proposed platform delivers a secure, censorship-resistant, and user-empowered file-sharing environment consistent with the principles of Web3.**

Keywords- Blockchain, IPFS, Smart Contracts, Decentralized Storage, Ethereum, File Sharing, Web3.

1. INTRODUCTION

With the increasing reliance on cloud storage solutions for digital data exchange, concerns regarding privacy, data ownership, and centralized control have grown significantly. Platforms such as Google Drive and Dropbox offer convenience but operate under centralized architectures that present inherent risks. These include susceptibility to data breaches, unauthorized access, single points of failure, and potential content censorship. Moreover, users typically relinquish control over how their data is managed once it is stored on these platforms.

Decentralized storage systems provide a promising alternative by distributing data across peer-to-peer networks and enabling user-centric access control. Blockchain technology further enhances this model by offering immutable record-keeping, transparent operations, and cryptographically secure transactions. When integrated with distributed file systems like the InterPlanetary File System (IPFS), such architectures allow for scalable, tamper- resistant storage that supports verifiable file ownership and retrieval integrity.

This paper presents a secure, decentralized file- sharing platform that utilizes Ethereum blockchain for metadata management and IPFS for content distribution. The system includes enhanced user authentication through MetaMask, enabling cryptographic identity verification without centralized servers. It also supports monetization and access control features using smart contracts. The proposed system not only strengthens data security and user privacy but also demonstrates how decentralized technologies can be effectively applied to real-world file-sharing use cases in a Web3 environment.

2. Literature Survey

Decentralized file storage has gained prominence as a solution to the inherent limitations of centralized systems such as censorship, single points of failure, and lack of user control. Foundational work like IPFS introduced content-

based addressing and peer-to-peer networking to enable distributed immutable storage. Ethereum's smart contracts extended blockchain capabilities to manage metadata and enforce access control on-chain, providing transparency and security. Subsequent projects like Filecoin and Storj have combined these approaches with cryptographic proofs and incentive mechanisms to encourage reliable data storage and retrieval. User-friendly tools such as MetaMask and Ethers.js facilitate seamless blockchain interaction and authentication. These developments form the basis for the decentralized file-sharing platform proposed in this paper.

S,NO	Paper Name	Algorithms / Technologies Used
1	IPFS - Content Addressed P2P File System (Benet, 2014)	Content-based addressing, Distributed Hash Tables (DHT), Peer-to-peer networking
2	Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform	Blockchain, Solidity smart contracts, Consensus algorithms (Proof of Work/Stake)
3	Decentralized Storage Networks: A Survey	On-chain vs. Off-chain storage, Cryptographic hashing, Token-based incentives
4	Filecoin: A Decentralized Storage Network	Proof of Replication, Proof of Spacetime, Token incentives
5	Storj: A Decentralized Cloud Storage Network	Erasure coding, Blockchain-based reputation system, Decentralized storage nodes, Encryptio

III. Existing System

Current file storage solutions primarily rely on centralized cloud service providers such as Google Drive, Dropbox, and Amazon S3. These platforms offer ease of use, high availability, and integration with other cloud services. However, their centralized nature introduces critical vulnerabilities. Data stored on centralized servers is subject to potential single points of failure, making it susceptible to outages, cyber-attacks, and unauthorized data access. Additionally, centralized control can lead to censorship, where providers may restrict or remove user content without transparency or recourse.

To address these issues, decentralized storage platforms have emerged, leveraging peer-to-peer networks to distribute files across multiple nodes. InterPlanetary File System (IPFS) is a widely adopted decentralized protocol that uses content-addressed storage to ensure data integrity and availability without relying on a central server. However, IPFS alone does not provide built-in mechanisms for access control, ownership verification, or payment enforcement.

Blockchain technology, notably platforms like Ethereum, complements decentralized storage by enabling smart contracts that manage metadata, ownership rights, and access permissions in a transparent and tamper-proof manner. Projects such as Filecoin and Storj integrate blockchain with decentralized storage to provide incentivized data hosting. Nonetheless, these solutions often face challenges related to scalability, usability, and transaction costs, limiting their widespread adoption.

In summary, while centralized systems dominate current file storage, their limitations have sparked the development of decentralized architectures combining distributed storage and blockchain verification. Our work builds upon these foundations by proposing an integrated system that enhances security, user authentication, and monetization features while maintaining usability.

IV. Proposed System

The proposed system is a decentralized file-sharing application designed to provide secure, transparent, and user-controlled data storage using a hybrid architecture that combines blockchain technology and distributed storage networks. At the core of the system, the InterPlanetary File System (IPFS) is used to store files in a decentralized, peer- to-peer environment, ensuring content immutability and eliminating reliance on centralized servers. The application uses Pinata to pin and manage IPFS content, ensuring file availability across the network. File metadata—such as IPFS hash (CID), file name, size, uploader address, and upload timestamp—is recorded on the Ethereum blockchain through custom Solidity smart contracts. This provides verifiable and tamper-proof ownership records. The frontend is developed using React.js with Vite for fast builds, TailwindCSS for responsive design, and Ethers.js to enable interaction with the Ethereum smart contracts. User authentication and authorization are handled via MetaMask, allowing cryptographic wallet-based login without passwords. The smart contract logic includes role-based access control (RBAC), supporting secure file sharing between different user roles, and emits events on file uploads to support real- time updates. Furthermore, the system integrates monetization through smart contract functions that enable pay-per-download and token rewards, establishing an incentive-driven environment. Scalability is achieved by storing only hashes on-chain and actual data off-chain, while future enhancements like NFT-based ownership, access control lists (ACLs), and support for Layer 2 networks (Polygon, Arbitrum) are envisioned to optimize performance and cost. This multi-layered architecture ensures data integrity, decentralized governance, and economic sustainability for secure digital content sharing.

V. System Architecture

The architecture of the proposed blockchain-based decentralized file storage system is designed with modular, scalable, and secure components, distributed across three

main layers: Frontend Interface, Blockchain Layer, and Decentralized Storage Layer. Each layer interacts seamlessly to facilitate file upload, authentication, metadata recording, secure retrieval, and monetization.

A. Frontend Interface (Client Layer)

The user interacts with the application through a web-based frontend developed using React.js and styled with TailwindCSS. Vite is used for faster builds and optimized performance. This interface provides functionalities such as user login (via wallet), file upload, viewing stored files, and initiating download transactions. The frontend communicates with the Ethereum blockchain using Ethers.js, a lightweight JavaScript library that handles contract interaction, wallet authentication, and transaction management.

B. Authentication and Wallet Integration

User authentication is handled through MetaMask, a browser-based wallet that enables cryptographic verification. When a user logs in, their wallet address is used as a unique identifier across the system. MetaMask also facilitates secure signing of transactions like file uploads, downloads, or payment transfers without requiring traditional usernames or passwords. This ensures decentralized identity management and enhanced privacy.

C. Blockchain Layer (Smart Contract & Metadata Management)

All file metadata is stored on the Ethereum blockchain via custom Solidity smart contracts. These smart contracts record critical information such as:

a) IPFS content identifier (CID)

b) File name and size

c) Uploader's wallet address

d) Upload timestamp

e) Access permissions

f) Each file upload triggers a smart contract function, which logs the data immutably on-chain and emits an event that can be tracked by the frontend. The contract includes role-based access control (RBAC) to determine which users can access specific files, enhancing security and data governance.

D. Decentralized Storage Layer (IPFS + Pinata)

Files are uploaded and stored on the InterPlanetary File System (IPFS), a peer-to-peer content-addressed network. The system uses Pinata, an IPFS pinning service, to maintain high availability and persistent access to uploaded files. When a file is uploaded:The file is hashed and broken into chunks by IPFS.A unique CID is generated, representing the content.This CID is stored in the smart contract, while the file resides off-chain in IPFS.Using off-chain storage via IPFS allows the system to scale efficiently while keeping only the essential metadata on-chain.

E. Monetization and Smart Payment System

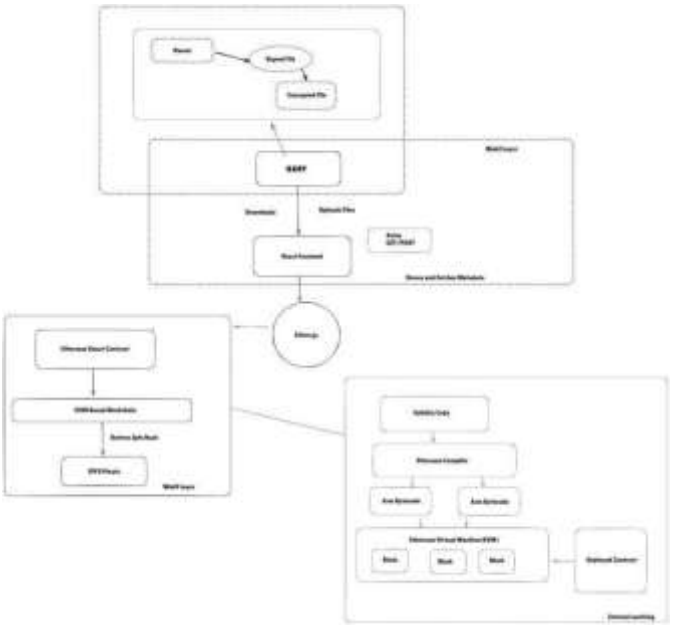
The smart contract supports micropayment mechanisms, allowing users to pay a small fee (in ETH or tokens) to access or download premium files. This functionality turns the system into a decentralized marketplace where content creators are rewarded directly. The payments are trustless, automated, and transparent.

F. Event-Driven Interaction and Feedback

a) The frontend listens for blockchain events emitted by the smart contracts (e.g., FileUploaded, AccessGranted) using Ethers.js. These real-time updates improve user experience by reflecting transaction status and system changes immediately after they are mined on the blockchain.

G. . Scalability and Future Readiness

To address gas cost and performance challenges, the system is designed to support Layer 2 solutions such as Polygon or Arbitrum in future versions. These networks offer faster and cheaper transactions while maintaining compatibility with Ethereum. Features such as file versioning, NFT-based ownership, and access control lists (ACLs) are also envisioned to enhance functionality.



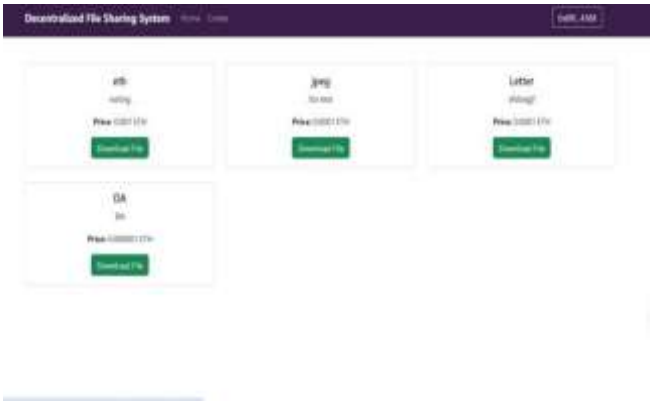
VI. Implementation Details

The implementation of the proposed system involves a three-tier architecture integrating frontend, blockchain, and decentralized storage components. The user interface is developed using React.js with Vite as the build tool for faster rendering, and TailwindCSS for clean and responsive UI design. Authentication is enabled through MetaMask, allowing users to securely log in and authorize transactions using their blockchain wallet. Ethers.js facilitates interaction between the frontend and Ethereum smart contracts, which are written in Solidity and deployed using the Hardhat development framework. These smart

contracts are responsible for recording file metadata—such as filename, size, upload timestamp, IPFS hash (CID), and uploader address—on the Ethereum blockchain. File content is stored off-chain using IPFS, with Pinata used as a reliable pinning service to ensure persistent file availability. Upon upload, the CID generated by IPFS is linked to the user's account through the smart contract. Role-based access control is enforced at the contract level to ensure secure file sharing. Additionally, monetization is achieved by integrating micropayment logic in the smart contract, allowing users to pay a small fee to access premium files. The system is tested on the Holesky Ethereum testnet to simulate real-world blockchain transactions without incurring gas costs, ensuring functional correctness and deployment readiness.

VII. Results / Features Demonstrated

The implemented system successfully demonstrates a decentralized file-sharing platform that combines blockchain immutability with scalable off-chain storage. Key features include seamless file uploads to IPFS with automatic CID generation and persistent pinning via Pinata. File metadata is securely recorded on the Ethereum blockchain through Solidity smart contracts, providing tamper-proof ownership and access control records. User authentication is effectively managed using MetaMask, enabling cryptographic wallet-based login without reliance on centralized servers. The smart contract enforces role- based access control, allowing only authorized users to view or download files. Monetization capabilities are integrated, supporting micropayments via smart contracts for pay-per- download functionality. Real-time blockchain event listening using Ethers.js enhances user experience by updating the frontend dynamically upon file uploads or access events. The system is deployed and rigorously tested on the Holesky Ethereum testnet, confirming end-to-end functionality, secure transaction processing, and decentralized governance. Overall, the platform provides a secure, transparent, and user-empowered environment for digital content sharing.



VIII. Future Scope

The proposed system can be improved by adding file versioning to allow users to manage different versions of their files. More detailed access controls can be introduced to provide better permission management for different users. Future work may include deploying the system on Layer 2 solutions to reduce blockchain transaction costs and improve speed. Enhancing privacy through better encryption techniques and ensuring compliance with data protection laws will also be important for practical use.

IX. Conclusion

This paper presents a secure and decentralized file-sharing system that leverages Ethereum blockchain and IPFS to provide transparent, tamper-proof metadata management and scalable file storage. By integrating wallet- based authentication and smart contract-driven access control, the platform empowers users with full ownership and control over their data. The inclusion of micropayment features enables direct monetization, fostering an incentivized sharing ecosystem. The proposed architecture addresses key limitations of centralized systems, offering enhanced security, privacy, and censorship resistance. Future improvements will focus on scalability, advanced access controls, and regulatory compliance to further strengthen the platform's usability and adoption.

X. REFERENCES

[1] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arXiv preprint* arXiv:1407.3561, 2014. [Online]. Available: https://arxiv.org/abs/1407.3561

[2] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," 2014. [Online]. Available: https://ethereum.org/en/whitepaper/

[3] E. Litke and D. Stewart, "Decentralized Storage Networks: A Survey," *IEEE Access*, vol. 10, pp. 34294–34322, 2022. DOI: 10.1109/ACCESS.2022.3148292. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3148292

[4] Pinata, "Pinata IPFS API Guide," 2023. [Online]. Available: https://docs.pinata.cloud/

[5] Ethers.js, "Ethers.js Documentation," 2023. [Online]. Available: https://docs.ethers.org/

[6] Hardhat, "Hardhat Tutorials," 2023. [Online]. Available: https://hardhat.org/

[7] European Commission, "General Data Protection Regulation (GDPR)," 2018. [Online]. Available: https://gdpr-info.eu/