e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

DECENTRALIZED CLOUD STORAGE

Prof. Ashwini Gharde*1, Rohit Karmokar*2, Jayanth Kuna*3, Om Mahindroo*4

*1Mumbai University, Department Of Computer Engineering, K.C College Of Engineering, Thane, Maharashtra, India.

*2,3,4Mumbai University, Department Of Computer Engineering, K.C College Of Engineering, Thane, Maharashtra, India.

ABSTRACT

This project proposes a decentralized cloud storage solution leveraging IPFS (InterPlanetary File System), Firebase for authentication and database, and smart contracts on Ethereum blockchain to securely manage user data access. Traditional cloud storage solutions are centralized, leading to concerns about data privacy, control, and availability. Our approach distributes storage, enhancing redundancy and security. Firebase ensures proper user identity and linkage of files to individual accounts. To avoid exposing the Pinata API key, a secure backend is used to handle uploads. The smart contract records metadata and ensures user-level segregation. This paper discusses the system design, architecture, and key implementation details. Results indicate improved user-level file separation, security, and transparency.

Keywords: Decentralized Storage, IPFS, Firebase, Smart Contracts, Blockchain, Pinata.

I. INTRODUCTION

Cloud storage systems have gained significant traction in recent years due to their ease of use and accessibility. However, most conventional cloud platforms are centralized, raising concerns regarding data breaches, vendor lock-in, and privacy issues. Decentralized storage models like IPFS offer a potential solution by distributing data across a peer-to-peer network. This paper presents a system integrating IPFS, Firebase, and Ethereum smart contracts to create a secure and user-oriented decentralized cloud storage platform. The core motivation is to ensure that data uploaded by one user cannot be accessed or overwritten by another, while also hiding sensitive credentials (like Pinata API keys) from the frontend using a backend approach.

II. METHODOLOGY

System Overview

The proposed system architecture includes the following components:

- **Frontend**: Built using React (Vite-based).
- **Authentication**: Managed via Firebase Authentication to link files with specific user IDs.
- **Decentralized Storage**: Files are uploaded to IPFS using Pinata.
- **Blockchain Integration**: A Solidity smart contract on the Ethereum testnet stores IPFS hashes and associates them with Firebase user IDs.

Secure Backend Upload Flow

- User logs in using Firebase.
- Frontend fetches a Firebase token.
- Token is sent, which verifies it using Firebase Admin SDK.
- If verified, backend uploads file to Pinata.
- Pinata returns an IPFS hash.
- The hash is stored on the blockchain using a Truffle-deployed contract.

III. MODELING AND ANALYSIS

Technologies Used:

- **Frontend**: React (Vite)
- **Backend**: Node.js + Express
- **Authentication & DB**: Firebase
- **Smart Contracts**: Solidity 0.5 on Truffle

- **Storage Layer**: IPFS (via Pinata)

Security Considerations:

- Pinata API key is never exposed on the frontend.
- Firebase verifies identity tokens server-side.
- Blockchain ensures immutable record-keeping of file metadata.

IV. RESULTS AND DISCUSSION

Key Achievements:

- Successful segregation of user-uploaded files.
- Backend upload system prevents exposure of Pinata credentials.
- Immutable storage of file metadata using smart contracts.

V. CONCLUSION

The proposed decentralized cloud storage system ensures secure and user-specific file storage using IPFS and blockchain integration. By leveraging Firebase authentication and a secure backend, it avoids common pitfalls such as API key exposure and file collision. The combination of web2 and web3 technologies ensures both usability and immutability. This system is a step forward in promoting user-controlled and tamper-resistant storage for real-world applications.

VI. REFERENCES

[1] Juan Benet, “IPFS - Content Addressed, Versioned, P2P File System,” 2014.

[2] Firebase Documentation, Google Developers,
firebase.google.com
Ethereum Smart Contract Best Practices, consensys.github.io/smart-contract-best-practices

[3] Pinata API Documentation,
pinata.cloud

[4] Truffle Suite Documentation, trufflesuite.com