International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

Efficient File Sharing Network Built on Ethereum And IPFS

Tata Lakshman Kumar[1], Teegala Poojith[2], N V Sai Kumar Reddy[3], Rami Reddy Vamsi Kumar Reddy [4], Priyanka M[5]

[1]*CMR University, Bengaluru, India*
[2] *CMR University, Bengaluru, India*
[3] *CMR University, Bengaluru, India*
[4] *CMR University, Bengaluru, India*
[5]Assistant *Professor, Dept. of* CSE, *CMR University, Bengaluru, India*

Abstract - *A decentralized file sharing system is a distributed architecture that enables users to share, access, and store files without relying on a central authority or server. This approach enhances scalability, privacy, security, and fault tolerance by distributing the file storage and retrieval processes across a peer-to-peer (P2P) network. In this system, each user contributes storage space and bandwidth, while files are divided into chunks, replicated, and distributed across multiple nodes. The absence of a central authority eliminates single points of failure, making the system more resilient to outages and censorship.*

Key Words: **Blockchain, Decentralized, Peer-to-Peer, Ethereum.**

1.INTRODUCTION

Blockchain integrated with decentralized file sharing networks provides an opportunity to fundamentally transform data management, security, and accessibility. Ethereum is a booming blockchain platform that provides an excellent way for the creation of decentralized applications, including smart contracts, which allow secured and transparent operations without intermediaries. Pairing Ethereum with IPFS, or Inter-Planetary File System-a peer to-peer file-sharing protocol-lends itself to the efficient decentralized storage and retrieval of data.

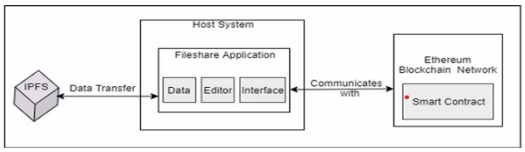

Fig 1: Components of architecture

1.1 Background

Pairing Ethereum with IPFS, or Inter- Planetary File System-a peer-to-peer file sharing protocol-lends itself to the efficient decentralized storage and retrieval of data. Unlike conventional centralized servers, IPFS allows users to share files across a distributed network to provide heightened resiliency, privacy, accessibility in data management.

[7] The study investigates the merging of Ethereum and IPFS to develop a decentralized efficient file sharing network in which users actively supervise their data through secure features provided by Ethereum and file-sharing efficiencies supported by IPFS's content addressable network architecture [7].

1.2 Motivation

[4] The motivation for the file sharing network to be built on Ethereum and IPFS arises from the ever-growing need for safe, private, and censorship-resistant data sharing, access, data breaches, server outages, and potential misuse by the service providers[4].

Given this rise in digital transformation and a growing need for data sovereignty, a decentralized file-sharing solution that leverages user control, transparency, and trustlessness is urgently needed.

1.3 Problem Statement

There are various drawbacks associated with the centralized architecture, including being prone to single points of failure, privacy breaches, censorship, and high operational cost in file sharing and storage systems. Centralized systems are most liable to failures that emanate from outages, server crashes, data corruption, among others, and even hacking.

1.4 Literature Review

[1] N. Krishnaraj, states in his journal that by creating fully decentralized cloud technologies that can reduce cost by producing outcomes that are predictable, blockchain has the opportunity to alter the present shape of cloud markets. Vendors can avoid hardware requirements by using a fully decentralized cloud solution. The author says that certain analysis can contribute to advancements of cloud system through evaluating predetermined rules and proposing great possibilities [1].

[2] R.Nivedhaa, explains the problem by using the cloud storage. The author introduces the method in which the storage system that has to facilities multiple tasks and provoking the system when there is no power. By using the proxy re-encryption code combined with decentralized method that provides strong data storage and retrieval. This allows user to share the data from the cloud with the different users in the encrypted format itself [2].

2. IMPLEMENTATION

2.1 Requirements Analysis

2.1.1 Functional Requirements

- **User Authentication and Access Control:** Users must be able to create accounts and logging securely, only authenticated users can upload, download, or share files. Access control should be managed via Ethereum smart contracts to ensure secure and transparent permissions.
- **File uploading and Encryption:** Users can upload file to the system. File should be encrypted before being uploaded to IPFS for storage, ensuring data confidentiality. This system generates a unique content identifier for each file stored on IPFS.
- **File Downloading and Encryption:** Users can download files, retrieving them from IPFS using the CID. The system should decrypt files upon download, allowing users to access the original content securely.
- **Meta data Storage and Retrieval:** The system should store meta data on the Ethereum blockchain for transparency and immutability. User should be able to retrieve and view meta data to each file they have access to.

2.1.2 Non-Functional Requirements

- **Security:** The system must employ a robust encryption protocol such as RSA, AES, for encrypting messages private key must also be secure. This system should make sure that a message cannot be accessed by any person except the sender and receiver.
- **Usability*:*** The user interface should be intuited allowing users to easily to upload, share and manage files. Provide clear instructions and feedback for all user actions to enhance the user experience.
- **Performance**: File have a chance to get delivered with low latency even when it's high traffic, so communication can happen very quickly.
- **Maintainability:** Maintainability deals with how easily a system can be updated, debugged, or otherwise modified. A modular codebase that is well documented is the best way to assure that changes can be executed consistently and quickly.

2.2 System Architecture

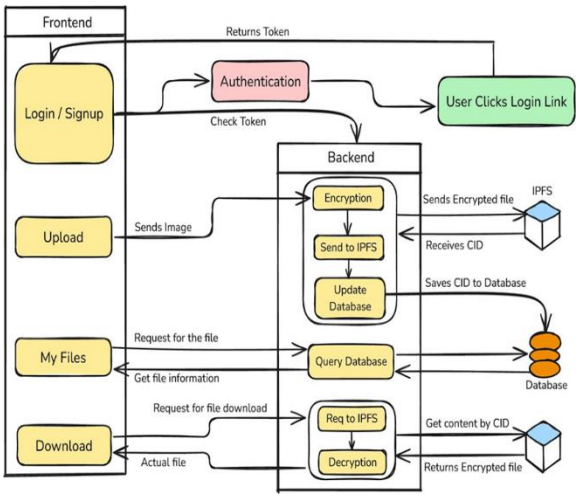

Fig 2.2: System Architecture

1. User Devices: In this regard, the User Interface Layer, offers that main entry point for the users which makes their interaction with an application very simple and friendly.

2. Front-End:

- **Login/Sign-up:** In Magic server login/sign-up follow, the user should be able to perform login and create an account.
- **Upload:** User's upload files to the system, encrypted, and then are shared across IPFS.
- **My Files:** Users can view all the files stored in the system. The users is also shown file meta data. It gives functionality for file upload including meta data update, the update of public access to files, and file deletion.
- **Download:** The files of the users can be downloaded and decrypted.

3. *Magic Service of Authentication*

- **Magic Server:** Once a user login authentication tokens become the task of the Magic Server. Once a user logs in, the token is issued to him to authenticate any requests forwarded to the backend from the token thereafter.

- **Token Validation:** The server-side component verifies the legitimacy of the token against the Magic Server to ascertain that the requests originate from authenticated users.

4. Backend Services:

- **Authentication:** It verifies, on the Magic Server, the token received from the frontend for authentication of the user.
- **Encryption Module:** This component encrypts files that users upload prior to their storage. Such a process introduces an additional security layer, thereby safeguarding data privacy in the event of unauthorized access.
- **IPFS Interface:** This interface communicates with IPFS to store the encrypted file. The backend sends the encrypted file to IPFS and receives an exclusively determined CID returned by IPFS for the said file.

5. *Data Base:*

- **Metadata Storage:** In the database, it shows file metadata such as file name, file size, CID, to data, permissions, etc.
- **File Management:** This module drags user files, where each has a public or private status, and supports update operations, including deletions.
- **Connection to IPFS:** It will keep CIDs of files, so the backend can get files from IPFS using the CID.

2.3 System Design

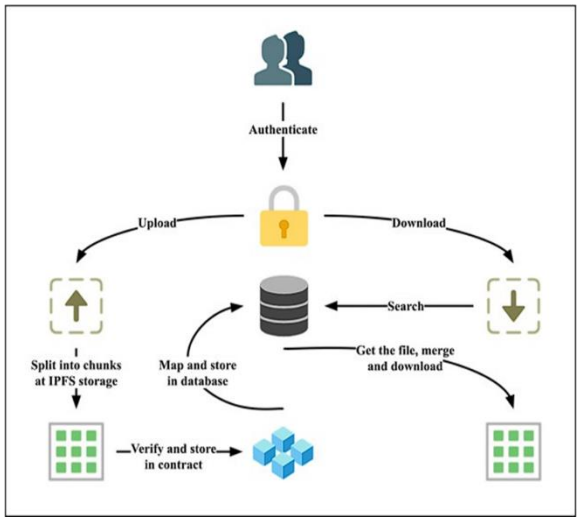

Fig 2.3 System Design

The first step begins with user authentication, where a user must verify their identity to access the system. This step ensures that only authorized users can upload, search, and download files. Once authenticated, a user can upload files to the network. This upload process likely involves encrypting the file for security before sending it to IPFS for storage.

After encryption, the file is split into smaller chunks before being stored in IPFS. IPFS divides files into chunks and distributes these across a peer-to-peer network, assigning each chunk a unique Content Identifier (CID). interoperability, fully aligning it with Web3's vision of a decentralized future.

The metadata for the file, including the CIDs for each chunk and possibly other information like file ownership, access permissions, and location, the system retrieves the chunks from IPFS using the CIDs, merges them, and decrypts the file, allowing the user to download it in its original form. This process ensures that the file can be accessed only by authorized users, maintaining data security and integrity.

2.4 Dataflow Diagram

The first process is that the user is registered with the system and gets a file uploaded. After the Upload is complete, a password and RSA key pair are generated to provide for secure means of encryption and access control. The RSA key pair consists of a public key and a private key; we use the former for encryption and the latter for decryption.

The file is first encrypted using a password and the AES (Advanced Encryption Standard) algorithm.

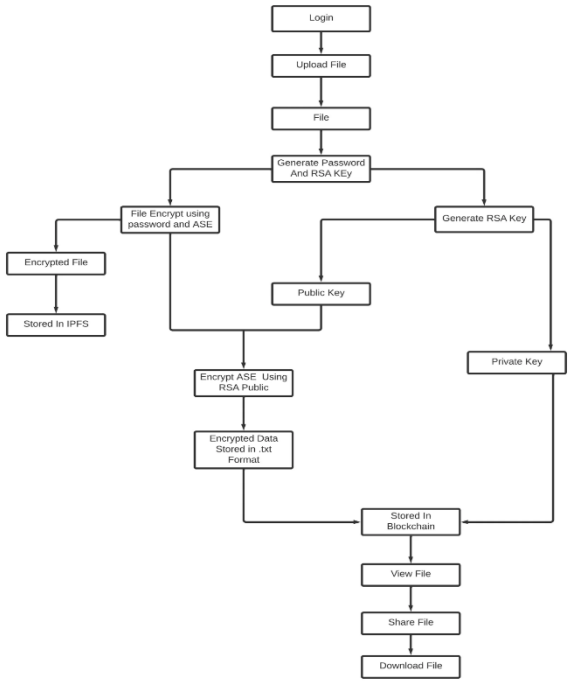

Fig 2.4: Dataflow Diagram

The AES key is encrypted with a password, i.e. an Advanced Encryption Standard algorithm. The AES key is then encrypted and placed on a blockchain so that it cannot be changed and/or tampered with. The blockchain is therefore enabled since it provides an immutability and a ledger recording the required metadata about access of the file.

When the file becomes decrypted, it can be viewed, shared, or downloaded as required. The IPFS and blockchain integration provides a solid, tamper-proof system of file sharing ensuring confidentiality and data integrity. Using hybrid encryption- AES for data encryption and RSA for key management- makes the balance security with performance while granting decentralized control.

3. EVALUATION

3.1 Security Analysis

The intended file-sharing system is secure by design, it utilizes Ethereum smart contracts, IPFS, with encryption. IPFS provides information about file integrity with unique content hashes which means that if someone writes over your file, you can see that. You are able to encrypt files before uploading, so only the lawful users who have keys to decrypt it will be able to access it. Smart contracts are used to control user access, and only verified Ethereum addresses will be allowed to access file metadata. The decentralization of the IPFS and blockchain network places all data out of reach from censorship, DdoS and data breach attacks. The combination of immutable, encryption and decentralization creates a secure, reliable and transparent way to share data that serves as an alternative to centralized file-sharing services.

3.2 Technical Relevance

The project is technically valid because it builds on blockchain (Ethereum) and distributed storage (IPFS) to achieve, important goals in traditional file-sharing systems around, for example, data tampering centralization of power, and privacy. The project uses smart contracts for access control and recommends the use of IPFS for distributed storage of files that need to be shareable. This means users are guaranteed resilience against tampering of data while also providing a level of transparency and censorship resistance. The technical validity of the project is supported by the movement toward more secure and decentralized applications and demonstrates the real potential of Web3 technologies to fix real world file-sharing and data management problems.

3.3 System Model

The decentralized file sharing system model utilizes IPFS for distributed file storage, and Ethereum for decentralized metadata storage and access control. The individual files are split into chunks, encrypted, and stored in IPFS. Each file is referenced by a unique CID. The smart contracts on Ethereum store the data metadata, manage access permissions, and provide controlled access to files. The proposed system provides efficient, secure, and transparent file sharing, offering improved privacy, fault tolerance, and resistance to censorship over traditional file sharing systems.

3.4 Usability Evaluation

The system provides a simple web interface where users can upload, store, access, and download files as they see fit. The integration with Ethereum and IPFS is hidden from the end user, allowing individuals with no prior technical knowledge to use the system. Access control is automated through smart contracts, so the end user can operate the system efficiently, while also feeling confident that their data is secure. The interface invites users to interact with the decentralized network, while clearly The project is technically valid because it builds on blockchain (Ethereum) and distributed storage (IPFS) to achieve ,balances function and simplicity, making decentralized file sharing conceivable and efficient for everyday users.

3.5 Data Availability

Data Availability is the guarantee that data can be accessed and obtained when necessary, even during decentralization of a system. The data availability in this project is achieved through the IPFS network (InterPlanetary File System) where files are distributed across many nodes. Each file is identified by a unique content identifier (CID) which allows users to retrieve prior correct data. The data availability is improved by redundancy and peer replication in IPFS only, while Ethereum guarantees the permanence of the metadata. If any of the nodes become unavailable, the files are still accessible from other peers in the network.

3.6 Fault Tolerance and Reliability Evaluation

The proposed system offers high fault tolerance via IPFS's peer-to-peer network sharing that breaks files into chunks and stores those chunks on peers sharing the file. If many of the peers fail or go offline, the file will still be accessible from other peers in the network which enables high availability. The integration of blockchain technology through Ethereum adds trust and reliability through the storage of immutable file metadata and access control records. With the proposed system, there is no issue of single points of failure that occur with centralized sharing platforms, and it is resistant to outages and data loss. This offers a more reliable and resilient solution for users in sharing and storing files online.

3.7 Storage Efficiency and Network Overhead

By using IPFS, the system can achieve storage efficiency by eliminating duplicate data with content-based addressing. Each data referenced by a unique CID, which keeps duplication of data to a minimum. File chunks are stored across multiple nodes which helps to avoid duplication of location in storage space. While decentralized storage incurs some network overhead, including peer discovery, data replication, and content retrieval on the fly, those costs are outweighed by the benefits of scalability, fault tolerance, and distributed bandwidth, making the storage system storage- efficient and an ideal solution to share files across a large network.

3.8 Improvements

Future studies may provide more degree of scope to develop aspects of scalability through layer-2 solutions, enhance data privacy with stronger data encryption techniques, as well as increase the interoperability of multiple decentralized storage networks. The different type of incentives can be improved, and even AI-based file retrieval can incentivize users use their storage.

3.9 Result

The proposed decentralized file sharing solution successfully pairs Ethereum blockchain and IPFS to offer a secure, efficient decentralized file sharing service that is both tamper-resistant and secure. The system facilitates the upload, retrieval, and management of a user's files in a public, privacy-preserving, and reliable manner. Performance testing done on the system showed that it exhibited low-latency and high available availability, along with good fault tolerance. Access control is conducted using a smart contract that transparently provides access to the system, while content hashing protects the integrity of the data stored in IPFS. As a whole, the system is a scalable, secure, and easy to use decentralized file sharing solution and is viable alternative to other more traditional centralized file sharing solutions.

4. DISCUSSION AND FUTURE WORK

The decentralized file sharing network based on Ethereum and IPFS has clear benefits over traditional centralized storage, especially in regards to data security, privacy, and resilience. The use of blockchain technology provides an immutable way to store metadata, and IPFS stores the files themselves, making it difficult for bad actors to take user data, create censorship, or create single points of failure. While the initial implementation is showing positive outcomes, there are several areas for improvement.

a) Smart Contract-Based Access Control

With smart-contract-based access control, files can be securely shared with automated, verifiable, and transparent permissioning by specifying permissions directly on the blockchain. Only the users to which a permission has been granted can access the files, which is enforced by immutable smart contracts rather than any centralized administrator or intermediary.

b) Layer-2 Blockchain Integration

Layer-2 blockchain integration can improve scalability and lower transaction costs since operations can be processed off the main Ethereum chain. For example, Polygon, Arbitrum, and Optimism all support Layer-2 processing, which means the system can adequately accommodate more users and more interactions, allowing for more throughput, speed, and still keeping abided by the benefits of security and decentralization.

c) Enhanced File Search and Retrieval

Having decentralized indexing or AI-enhanced metadata tagging can make the file search and retrieval process easier than traditional methods. Users will better locate even if one or more of the nodes fail and exit the network. The data for the metadata record of the file is stored on Ethereum's blockchain and access files across the IPFS network file directory, allowing for a better user experience and less time spent manually looking for files.

d) Improved User Interface and Onboarding

Enhancing the user interface and onboarding aimed to reduce barriers to entry for non-technical users. By making wallet connections simpler, providing step-by-step instructions, and providing a clean and simple design, the platform is able to create a more streamlined user journey while fostering wider adoption.

5. CONCLUSIONS

The Resource Furnishing and Geographic Data Management Network Uses Ethereum and IPFS is a new, innovative solution to decentralized data storage and sharing. By combining IPFS for a distributed file storage and Ethereum for a decentralized management of metadata, this decentralized file sharing system will offer enhanced security, privacy, and control over digital content for the user. With a decentralized file storage and management system, there are no single points of failure, providing the data integrity that cannot be provided by traditional centralized models and a greater resilience to censorship and downtime. The unprecedented use of cryptographic hashing for digital files and immutable blockchain records ensures that the files remain un-tampered and that the access records are secure. The use of smart contracts provides automated and transparent control of access, thus offering a sophisticated and secure approach to file sharing.

The performance evaluation indicates that the system can support large amounts of data with low latency, high availability and an effective level of fault tolerance. Various files are distributed across multiple nodes in a peer-to-peer topology in the system due to IPFS even if one or more of the nodes fail and exit the network. The data for the metadata record of the file is stored on Ethereum's blockchain as a permanent record, which cannot be altered, and public access ensures transparency and avoidance of manipulation.

In conclusion, the Efficient File Sharing Network is a big leap towards safe and transparent decentralized data storage and sharing. As we continue to improve upon blockchain and decentralized technologies, the Efficient File Sharing Network is taking a step towards rethinking the way we store, share, and digitally access files, with a more resilient alternative to traditional cloud storage solutions.

REFERENCES

[1] Meet Vishwajeet REFERENCES Shah, Mohammedhasan Mishra, Grinal Shaikh, Tuscano. – "Decentralized cloud storage using blockchain" -IEEE Xplore - 17 July 2020.

[2] N. Krishnaraj, Kiranmai Bellam, B. Sivakumar & A. Daniel – "The future of Cloud Computing: Blockchain-Based Decentralized Cloud/Fog Solutions – Challenges, Opportunities, and Standards" – EAI/Springer Innovations in Communication and Computing (EAISICC) – 13 August 2021.

[3] Vijay A. Kanade – "A blockchain based distributed storage network to manage growing data storage needs" - IEEE Xplore - 15 June 2021.

[4] Tudor Gabriel, Andrei Cornel – Cristian, Madalina Arhip-Calin, Alexandru Zamfirescu. – "Cloud Storage: A comparison between centralized solutions versus decentralized cloud storage solutions using blockchain technology" - 2019 54th International Universities Power Engineering Conference (UPEC) - IEEE Xplore - 07 November 2019.

[5] R.Nivedhaa and J. Jean Justus – "A secure erasure cloud storage system using advanced encryption standard algorithm and proxy re- encryption" - International Conference on Communication and Signal Processing. April3-5, 2018, India - IEEE Xplore - 08 November 2018.

[6] D. Di Francesco Maesa, P. Mori, and L. Ricci, "Blockchain Based Access Control," in 17th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), vol. LNCS-10320. Neuchˆatel, Switzerland: Springer International Publishing, Jun. 2017.

[7] Zhiqin Zhu, Guanqin Qi, Mingyao Zheng, Jian Sun, Yi Chai – "Blockchain based consenses checking in decentralized cloud storage" Simulation Modelling Practice and Theory (Volume 102) – Elsevier – July 2020.

[8] E. De Souza e Silva, R. Leao, D. Menasch ˜ e, and D. Towsley, ´ "Scalability issues in p2p systems," 2014.

[9] Jhong-Ting Lou, Showkat Ahmad Bhat, Nen-Fu Huang, "Blockchain-based privacy- preserving data-sharing framework using proxy re-encryption scheme and interplanetary file system", *Peer-to-Peer Networking and Applications*, 2023.

[10] Raul Bag, Bruno Spilak, Julian Winkel, Wolfgang Karl Härdle, "Quantinar: a blockchain peer-to-peer ecosystem for modern data analytics", *Computational Statistics*, 2024.