International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(3):1086-1095

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i03.0925

Blockchain-based File Sharing System – A Hybrid Approach

Ritik Tiwari, Viswanathan V*, Rajarajeswari S

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India. *Corresponding Author's Email: viswanathan.v@vit.ac.in

Abstract

File sharing has become a common practice in our daily lives, but there are many concerns regarding security and privacy for various reasons. To address these concerns, researchers are exploring different storage and sharing options. Blockchain technology, with its decentralized and immutable nature, provides an appealing solution. The proposed hybrid file-sharing system offers a unique approach to combining the benefits of centralized user management and decentralized storage. By utilizing Ethereum smart contracts for access control, the system ensures secure and transparent sharing of files. Moreover, using the Inter Planetary File System for decentralized storage ensures high availability and scalability. This concept attempts to solve the accessibility, privacy, and security issues related to centralized and decentralized systems by utilizing the advantages of both models. The centralized system will help manage user authentication and authorization, making it easier for individuals unfamiliar with blockchain technology to interact with the system. On the other hand, IPFS provides decentralized storage with data redundancy and reliability. Access control is implemented through Ethereum smart contracts, which restrict file access to unauthorized users. Overall, this paper demonstrates how blockchain technology, with a hybrid model, can create a user-friendly, secure file-sharing environment immune to common vulnerabilities in centralized systems.

Keywords: Blockchain, Decentralized and Centralized, Ethereum, File-Sharing, IPFS.

Introduction

The digital era has completely changed the way we work, play, and communicate with each other. In our increasingly linked world, data has become an essential part of our life. Data storage is shifting drastically from local servers to cloud servers. As we all know, data is necessary to us, so ensuring the privacy and security of data is the bare minimum. Here, blockchain comes into the picture; blockchain, an immutable distributed ledger, provides decentralized storage for the data, which will be more secure than a centralized storage server. File-sharing platforms built on blockchain have a lot of potential in various applications. In the healthcare industry, blockchain technology ensures data integrity and privacy while securely allowing institutions to share patient details. In the finance sector, it can make it easier for private financial records and sensitive papers to be exchanged securely. Also in the legal Sector, without jeopardising the data, it allows for the safe and un-hackable exchange of contracts, case files, and legal papers.

Nowadays, we all share our data daily for different tasks we all do for our daily necessities. Easy file transfers are now a part of everything we do, from working together on projects to getting enjoyment. File-sharing services are now necessary for exchanging creative works, keeping priceless memories, and sharing papers with co-workers. There are so many file-sharing applications; some have failed, and some are still working. One is Bit-Torrent, which is still alive and used. Several unidentified nodes (people) can send and receive files with a peer-to-peer methodology (1). However, the increasing dependence on centralized file-sharing platforms has highlighted serious issues with accessibility, privacy, and security. Large volumes of user data are frequently stored on centralized platforms, leaving them open to censorship, hacking, and abuse. In addition, they run the danger of data loss and inaccessibility due to centralized storage, which exposes them to single points of failure. These constraints create an increasingly pressing problem, centralized increasingly pressing problem: centralized systems raise concerns due to their worrying vulnerabilities in the more data-driven world of today, even while they offer user-friendly interfaces and widespread acceptance.

To address these file-sharing issues, this study

(Received 08th April 2024; Accepted 14th July 2024; Published 30th July 2024)

explores blockchain technology. With its fundamental values of decentralization, immutability, and transparency, blockchain presents a viable way forward for file sharing and storage that is safer, more dependable, and more user-friendly. The proposed work explores the possibilities for a secure, open, and decentralized file-sharing environment by utilizing the distributed nature of blockchains and the strength of smart contracts to address the drawbacks of centralized systems.

Smita Athanere *et al.,* (1) proposed Blockchain-Based Hierarchical Semi-decentralized Approach using IPFS for Secure and Efficient Data Sharing. It is concerned about the security of the data stored on the cloud by multiple providers when accessing the data from third-party vendors. Some traditional ways are used to access the data, which are compromiseable. As a solution, this paper provides a way to use blockchain and IPFS to store data securely. Blockchain holds the property of immutability and transparency, which offers a strong base for data storage. This system has a two-level key management technique that encrypts the file before hashing. This solution lowers the transmission and compute costs and simultaneously reduces the single-point failures. This also protects data in the cloud environments against malicious and dishonest cloud providers.

Vimal *et al.,* (2) proposed a new cluster P2P file-sharing system model based on IPFS and blockchain technology. It is stated that the system collapsed due to the egoistic nodes in the distributed peer-to-peer environments. Some incentive-based mechanisms have been introduced to reduce these kinds of collapses. This paper will integrate proximity awareness and trustworthiness into the file transfer process to enhance the P2P file sharing efficiency via IPFS and Blockchain. Blockchain challenges are discussed, and solutions with IPFS are provided. A P2P-based technology that provides resource sharing through hashed files across nodes is called IPFS. They integrated Filecoin with the IPFS to incentivize minors and service providers and ensure high throughput file transfers. Zheng *et al.,* (3) proposed an innovative IPFS-based storage model for blockchain, and there are some concerns about scalability because of the popularity of the blockchain as the Bitcoin blockchain ledger grew to 200GB at that time. The limitation will create problems in technology development and network expansion. They suggest using an IPFS-based blockchain data storage model to store the data. Here, miners deposit the data in the IPFS network and embed the IPFS hash, which comes as output into the block, which will reduce the data size of the blockchain. They also applied this same methodology to Bitcoin, which resulted in a compression of 0.0817 with good performance of synchronization speed of nodes and security. Huang *et al.,* (4) proposed a secure file-sharing system based on IPFS and Blockchain, highlighting the issue of recording transactions and the limitation of storing large files on the blockchain. There are some technologies like IPFS and Swarm, but there are some inefficiencies in data sharing. This research paper proposed a new technique: secure file sharing with an IPFS proxy for distributed access control and group key management. The system will fill some fundamental gaps, enhance security, and manage access control efficiently, creating a flexible group for membership by just connecting the IPFS with the blockchain via proxy. Hen *et al.,* (5) provides an improved system based on integrating IPFS and blockchain. This will promise a solution that enhances the performance and reliability of peer-to-peer file-sharing systems. They are addressing the throughput limitations of individual users in IPFS. They aim to improve data availability, security reliability, and storage using zig-zag-based storage models and content service providers. This scheme also effectively removes these significant challenges and offers a potential breakthrough in decentralized file systems.

There is an exponential data volume growth by blocks, so most blockchain-based applications face many challenges. This is because of the immutable and append-only nature of the blockchain, which increases the size of the blockchain and creates access issues. R. Kumar *et al.,* (6) came up with the IPFS-based blockchain storage model, where the transaction data is stored in the IPFS system so that the size of the blockchain will be reduced. IPFS generates its hash, so the data will be secured, which is content-addressed storage, which also contrasts with traditional location-addressed techniques. It promises to enhance transaction access in blockchain and can be developed using Python, Flask, Anaconda, and IPFS technologies. The solution proposed by Nizamuddin *et al.,* (7) is a

decentralized approach to document sharing and version control, using Ethereum smart contracts and IPFS for storage. The proposed framework automates interactions among multiple stakeholders and emphasizes security and truthfulness without the need for centralized intermediaries. The solution contributes to the existing literature on blockchain-based solutions for collaborative document management and offers practical insights into implementation and security considerations. The smart contract code is posted on GitHub for transparency and accessibility. Almasian *et al.,* (8) talk about blockchain technology having some issues in terms of scalability and user anonymity; this emerges as a potential solution for implementing an access control setup. The flexibility and effectiveness of controlling and user revocation may be lacking in standard encryption techniques. The suggested solution combines attribute-based encryption (ABE) with blockchain technology to ensure secure, fault-tolerance, and decentralized file-sharing. The method helps in effective key management and provides quick user revocation without requiring extra cost by adding cipher keys into access polynomials. Also, the usage of ABE protects user anonymity while allowing access control to be enforced.

Security, trust, and transparency must be improved in existing trusted third-party (TTP) - based data sharing platforms. Naz *et al., (*9) proposed an IPFS and blockchain-based data-sharing platform to overcome these shortcomings. This approach guarantees trust, security, and transparency using the Ethereum blockchain for access management and IPFS for decentralized storage. Data integrity is improved with RSA authentication and Watson analyzer for review validation. By merging decentralized storage, blockchain, encryption, and incentive systems, this research closes the gap and provides enhanced security, access control, and data authenticity. Advanced data delegation solutions are required by Web 3.0 and blockchain technologies that are constantly evolving. While ABE is the backbone of the current methods, effective cipher text processing still needs to be included. Gao *et al.,* (10) address the limitations of attribute-based cipher text transformation (ABCT) and give customized data transfer solutions for Web 3.0 that enhance current ABE-based solutions' capabilities. Smart contracts guarantee blockchain-based auditing and independent execution. Using digital signatures and commitment mechanisms enhances and improves regulatory compliance. The proposed strategy focuses on verifiability and fine-grained access control, showing its value and utility through evaluation.

Previous works have addressed several research gaps in storage and access control. Some works have been decentralized, some centralized, and others have been a combination. These works have mainly focused on using techniques like RSA and ABE. However, there still needs to be a gap to be filled regarding user interaction with Web3.0 systems and secure file transferring model. To overcome the issues faced by current systems, this paper proposes a hybrid blockchain-based file-sharing system in which user management, consisting of authentication and authorization, is handled by a centralized system. On the other hand, for security and privacy, the decentralized system-based IPFS network will hold the file secure and incorruptible, and the Ethereum blockchain smart contracts manage all the access control. So, this system solves the major problems faced by existing platforms.

Methodology

This paper proposes a hybrid concept of sharing files, combining centralized and decentralized systems. This idea is proposed to break the barrier of users not shifting to Web3.0 technology because it is new, hard to learn, or takes some real money. As we know, it's hard for some rookie users to make a crypto wallet, connect it to the system, and pay some gas (amount) to use the system; they lack knowledge or not proper resources and no central regularities such as banks or government, so they're afraid of losing money. So, this system will use centralized user management, consisting of user authorization and authentication, the same as many Web2.0 systems or websites. Now, the decentralized system will do its primary work, storing the file in the IPFS model, and then the access control and the IPFS hash will be saved in the Ethereum smart contract for better security and privacy. Figure 1 shows the proposed system architecture.

In short, this centralization will remove the need for users to interact directly with the Web3.0 modules and manage the gas fees from their side. This will allow users to interact smoothly with Web3.0 or blockchain technologies without being

shown so many complex things. This model will also be beneficial for accommodating users unfamiliar with the Web 3.0 world. Table 1 shows comparison of features involved in the file sharing between existing systems and proposed.

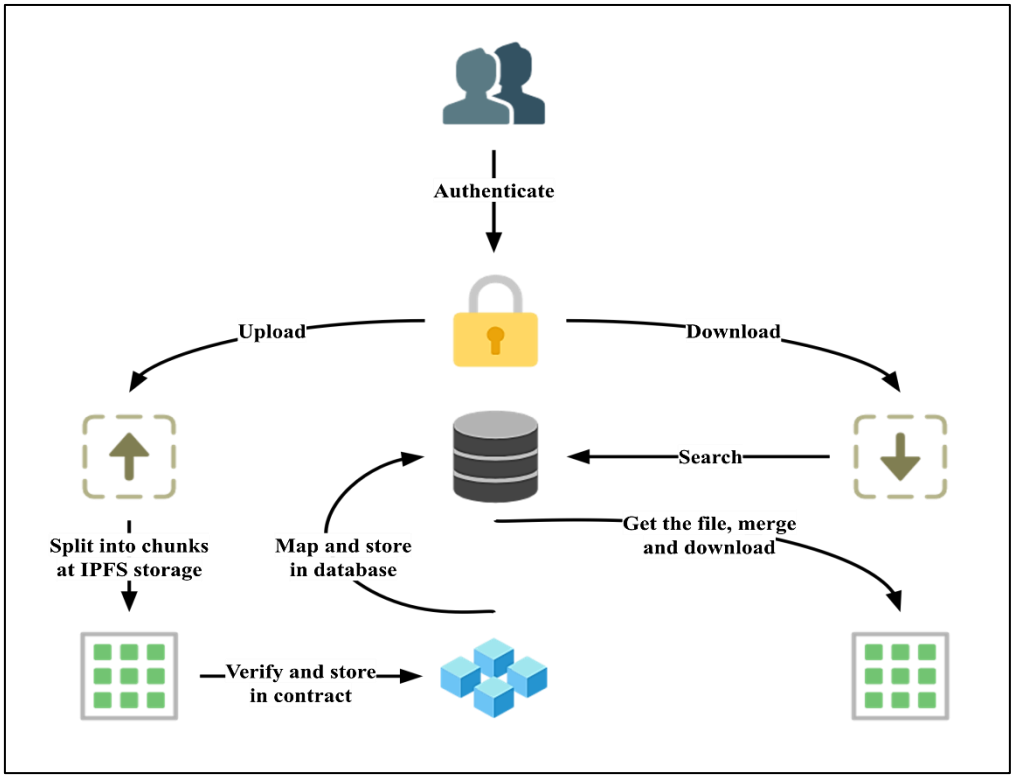

Figure 1: Proposed System Architecture

Table 1: Comparison between the Existing System and the Proposed System

References	Blockchain	Encryption	IPFS	Access Control	Centralized Database
Approach A (11)	✓	✗	✗	✓	✗
Approach B (12)	✓	✗	✓	✗	✗
Approach C (13)	✓	✗	✓	✗	✗
Approach D (14)	✓	✗	✗	✗	✓
Proposed Approach	✓	✓	✓	✓	✓

System Structure and Integration

The blockchain layer, the data storage layer, and the user interface layer are the three main parts of the suggested system. The blockchain layer uses decentralised ledger technology to guarantee data security and integrity. Within the blockchain layer, access control is managed via smart contracts. Large volumes of files that need to be shared among users are stored on IPFS. We can do user interface layer authentication and authorization using databases and fundamental web technologies. Among the technologies incorporated are:

User Interface: Developed with Next.js to offer a fluid and quick user experience. In order to authenticate credentials and use OAuth, it also uses NextAuth.

Data Storage: To improve redundancy and accessibility, decentralised storage solutions employ IPFS. We can connect the IPFS to a number of web libraries and frameworks and upload the file there.

Block chain: Solidity can be used to create smart contracts, and web3.js and ethers.js tools can be used to communicate with contracts.

By integrating these technologies, the system is able to achieve great efficiency and scalability while simultaneously taking use of blockchain's security features.

System Model

User Management: This will be managed by the centralized system, where users can log in or register with GitHub, Google, Email, or any other provider. This is helpful for the user to have easy interaction with the system.

File Management: Here, a user will upload the file, and that file will be uploaded to the IPFS nodes of multiple networks. IPFS uses content addressing, which will split the content and produce the hash for them; it also ensures that the same piece of content has the same hash every time so that the problem of data redundancy will not occur. These small chunks will be stored in Merkle DAG (Directed Acyclic Graph), and each node will represent the chunk and point to its parent. This helps track the significant content. The hash will be stored in the Distributed Hash Table for better routing and retrieval. When the hash of the file is searched in the IPFS pool, the data will be fetched from the multiple nodes where it's stored.

Access Control: The user has to provide access to another user who wants to share the file so that the IPFS hash and the authorized users' access are stored in the smart contract of the Ethereum blockchain. There will be minimal chances of data leaks, which will be almost negligible. Only authorized users can access the specified file.

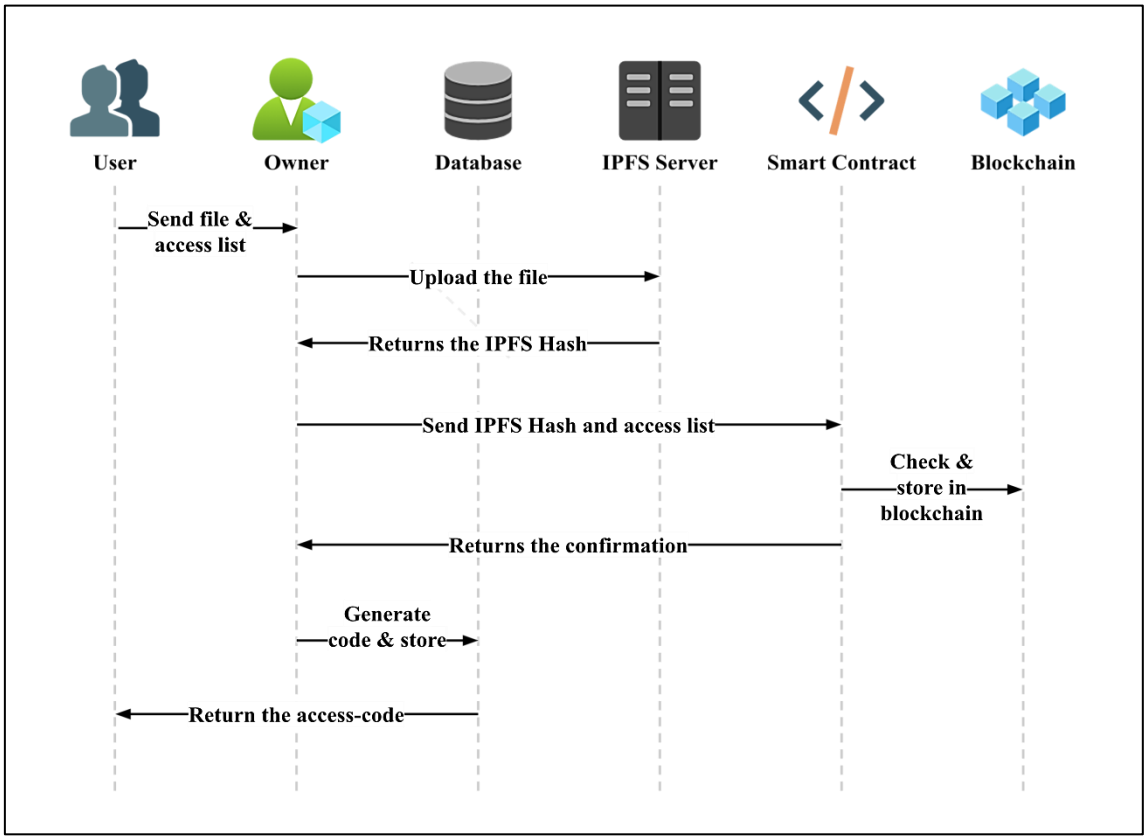

Figure 2: Sequence Diagram for Uploading Process

Workflow: Users authenticate the system with email, Google, or GitHub, as every website provides old methods. The user will upload the file, enter the email addresses of the other users who want to share the file, and ensure they all have an account in the system. The file will be uploaded, and an access code will be visible to the user. Meanwhile, in the backend, the file will be stored in the IPFS, get the hash from there, and then the access list and the hash will be saved in the Ethereum smart contract, and the gas (amount) will be deducted from the system's owner. Then, the access code will be generated randomly; it will map with the smart contract and be stored in the database. Now, when the user hits that access code and searches the file, the mapped contract will get the email list, and if the email of that user is present, then only the file will be visible to that user; otherwise, it is not accessible. Figure 2 and figure 3 show the sequence of uploading and downloading process.

Technology Stack: The proposed model uses the technologies whose use cases have been described in the subsection. Next.js is a React (JavaScript Library) framework from which we can build full-

stack applications. This framework has many optimization features and additional things that can simplify developer life when creating these bulky and complex applications. Typescript is a language that is similar to JavaScript. There are many rules about how different values can be used and how the program should run. Tailwind CSS is a library that is used to style the component, so all the styling of the system is done by this library. MongoDB is a type of database that stores data in a document format. It is a scalable and flexible database solution; this is a NoSQL database used in this system to store the user information and some file information to record all the transactions that the user did in the system. Ethereum is a network of computers worldwide that follows a set of rules. This is the foundation for the communities, applications, and digital assets anyone can build and use. It is like a blockchain database, which stores data as smart contracts. Solidity is a statically typed curly-braces programming language designed to develop Ethereum network smart contracts. This system uses solidity to make smart contracts, which can be helpful in access control.

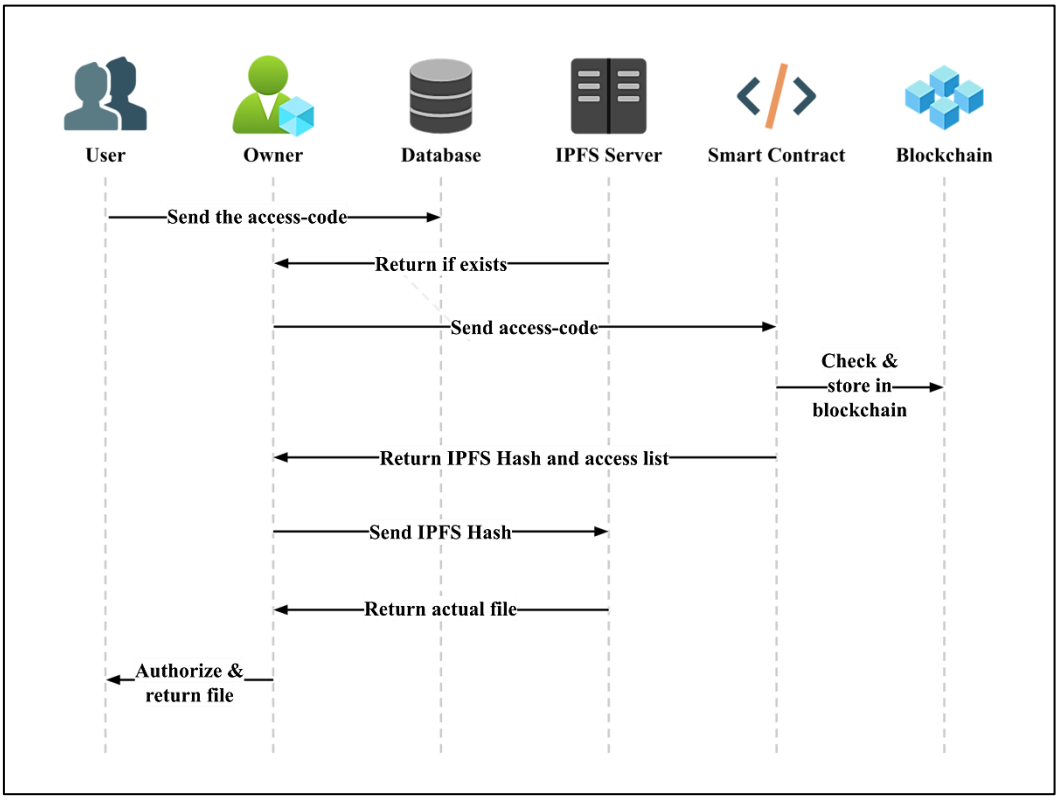

Figure 3: Sequence Diagram for Downloading Process

System Creation

Design Phase: First, we must choose an effective blockchain on which to quickly deploy our smart contracts and establish the schema for storing user information. Determine how to connect IPFS technology for file storage. Select the user interface framework that best suits this application.

Development Phase: Use Next.js to create a front-end user interface, and carefully plan the UI/UX. Integrate the IPFS technology into the user interface to enable seamless file uploading. Put into practice the Ethereum blockchain's smart contracts for access control with stability.

Testing Phase: To make sure system components interact correctly, unit and integration tests were conducted.

Evaluation Criteria

The system's assessment was done using:

Security: In order to safeguard our system from unauthorised access and data breaches, we assessed the risk involved in system development.

Performance: We evaluated our system based on a number of measures, including system throughput, bandwidth, and operating speed as well as data retrieval time.

Scalability: We tested the system under various loads to make sure it can withstand more usage

without degrading performance, as well as the cost of transactions and execution when handling a big user base.

Results and Discussion

The proposed hybrid file-sharing model's implementation demonstrated positive efficiency, security, and usability outcomes. The approach effectively addressed the shortcomings of traditional centralized file-sharing platforms by merging decentralized file storage and access control with centralized user management. Users can share files with this system without interacting directly with complex blockchain technologies. Using centralized user management improves user accessibility and reduces entrance barriers. Data redundancy and fault tolerance were ensured via IPFS for file storage, reducing the possible data loss compared to a centralized storage server. Smart contracts on Ethereum were used to create access control mechanisms, which provide high security and privacy protections. Files could only be accessed by authorized users, and the permissions were immutably and transparently enforced on the blockchain. Using this strategy, data breaches and unauthorized access could be reduced. Also, the user experience has been enhanced by quick retrieval and verification of files by the content-addressed storage in IPFS. In addition, the system displayed reliability and scalability, managed high file transfer volumes, and supported a growing user base. As IPFS and blockchain are decentralized, the system is immune to censorship attempts and a point of failure. So users could rely on the system to share the files safely. Security audits are required to find some vulnerability; performance metrics like transaction latency, upload/download speeds, and network stability can only be examined via real-world simulation. Only after the system is public in the beta version can we find and fix the major issues and check the system's performance.

The system uses a smart contract, and while uploading the contract on the Ethereum blockchain, we will need a specific amount of gas; 3,000,000 is the maximum limit set by the creator. We have observed two types of gas consumption: transaction and execution. In Figure 4, we have described how much gas is used per function; as shown in the figure, the upload function required more gas as it had to put the data into the blockchain, which took more time to create a block. This gas varies on the size of the file. In Figure 5, the number of ethers is loaded in the account, and the usage of the ether is in the form of gas. All the data, transaction cost, execution cost, and the overall gas amount are stored in Table 2.

Table 2: Smart Contract Cost Test

Functions	Transaction Gas	Execution Gas	Actual Gas
Uploading	594165	502275	$3.4 * 10^{-12}$
Downloading	541234	452832	$1.1 * 10^{-12}$

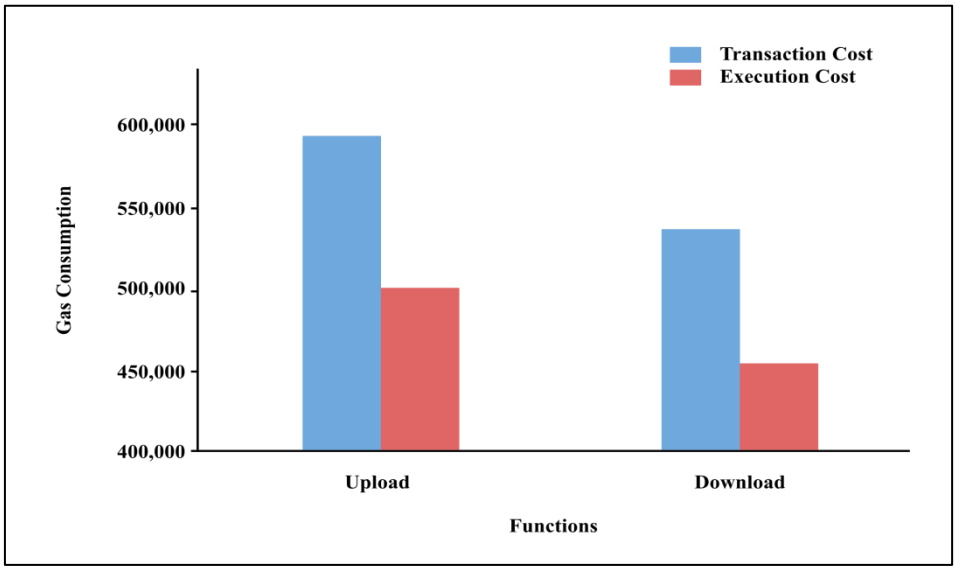

Figure 4: Gas Consumption for the Smart Contract

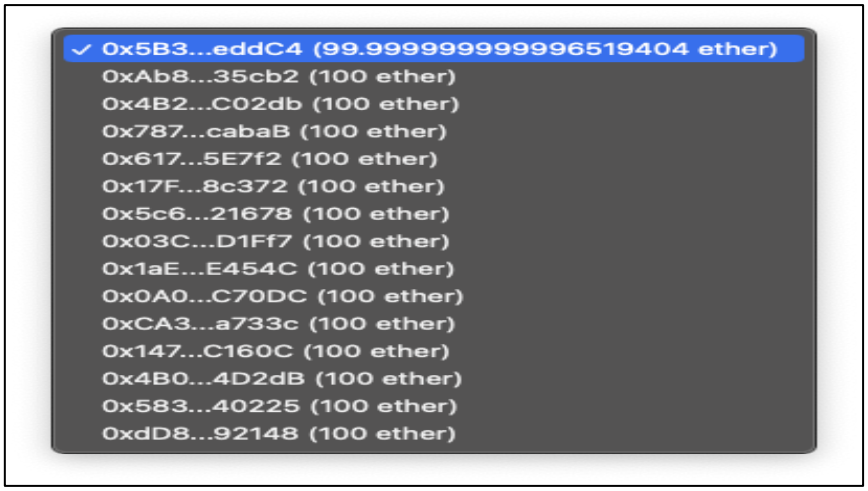

Figure 5: Ethers for Gas Consumption

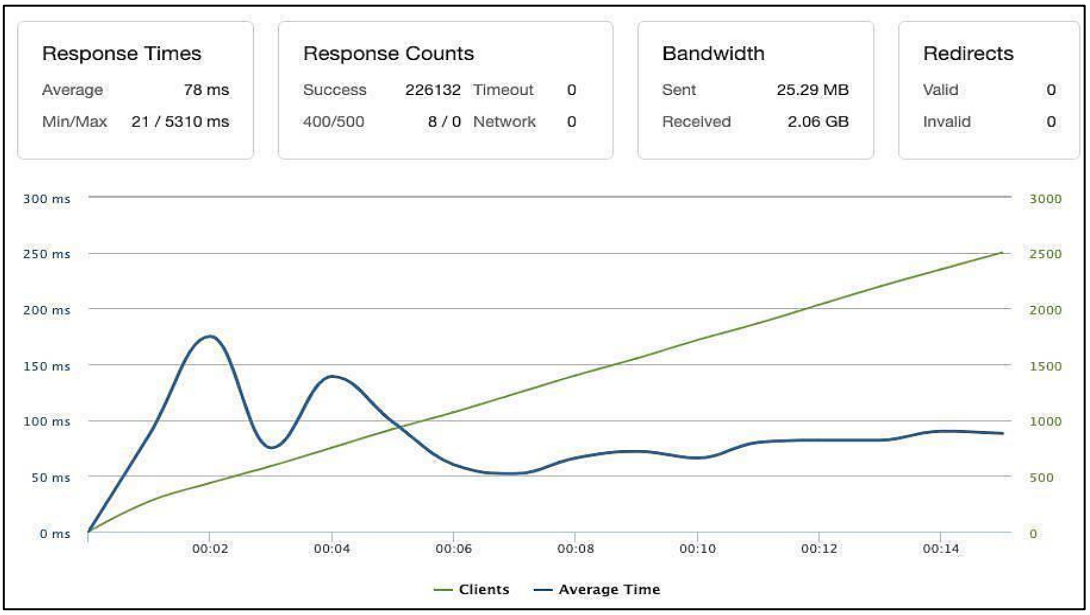

Figure 6: Performance Metrics Report with 22k Users in 15 Seconds, Generated by Loader.io

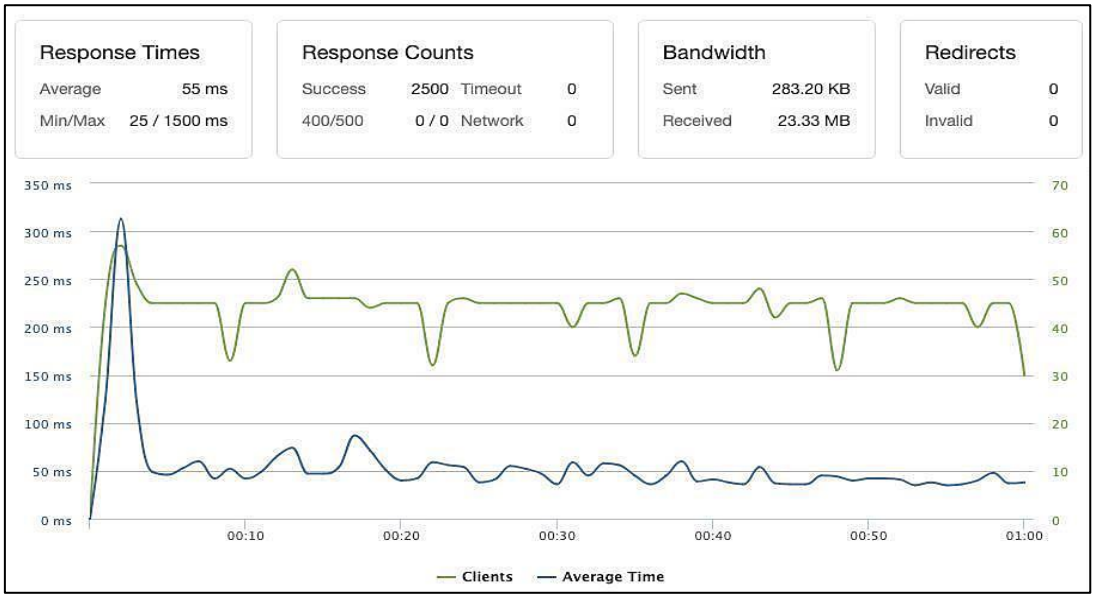

Figure 7: Performance Metrics Report with 2500 Users in 1 Minute, Generated by Loader.io

We have conducted performance tests based on the user using the system simultaneously. In figure 6, we can see that at 00.12 seconds, 2000 users are using the system, and the waiting time is approx. 65ms, which means when they load the system, it will take 65ms to show the results to them. We performed the test many times and found that the average response time is 78ms when 226132 users access the system for 15 seconds and 55ms when 2500 users access the system simultaneously for 1 minute, which is a good benchmark. The results are given in Figure 6 and Figure 7.

Comparing the hybrid file-sharing system to traditional systems, notable performance gains were seen. Because we are using IPFS, this system has a lower failure rate while uploading files. With the previous way, the database may shut down due to overloading or an unknown problem. Because of our system's optimised IPFS integration, which retrieves files without requiring a database query, data retrieval is substantially faster. By increasing system throughput, the hybrid solution effectively handled more users and data flows at once.

The following strategies should be taken into account by experts when developing blockchain-based file-sharing systems, according to our findings:

Integrate IPFS: To improve data security, accessibility, and redundancy, think about utilising IPFS technology for decentralised storage.

Smart Contracts: Use smart contracts to automate authentication procedures and manage access control.

Performance Optimisation: To guarantee excellent performance under varied loads, test and improve transaction speeds and data retrieval times on a regular basis.

Conclusion

Hybrid file-sharing system significantly improves over the traditional centralized file-sharing platforms, as it addresses their shortcomings. The utilization of the pros of both decentralized and centralized systems provides a user-friendly, reliable, and secure way of file-sharing in the digital world. Access control, decentralized file storage, and centralized user management offer a balanced solution that puts usability first without compromising security and privacy. Ethereum smart contract and IPFS are two technologies of the Web3.0 world that ensure efficiency, transparency, and immutability. Future research and development studies may focus on enhancing decentralized file-sharing systems' scalability, performance, and additional use cases. Everything looked at; the proposed hybrid file-sharing system is a significant step towards unlocking blockchain technology's full potential to change data sharing and storage paradigms completely.

Abbreviations

IPFS: Interplanetary File System
P2P: Peer-to-Peer
ABE: Attribute Based Encryption
TTP: Trusted Third-party
DAG: Directed Acyclic Graph
DHT: Distributed Hash Table

Acknowledgement

Nil.

Author Contributions

Ritik Tiwari contributed to the design and implementation of the research, while Viswanathan V and Rajarajeswari S contributed to the analysis of the results and the writing of the manuscript.

Conflict of Interest

The authors declare that there is no conflict of interest.

Ethics Approval

Not applicable.

Funding

No funding received by any government or private concern.

References

1. Athanere S, Thakur R. Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. Journal of King Saud University-Computer and Information Sciences. 2022 Apr 1;34(4):1523-34.
2. Vimal S, Srivatsa SK. A New Cluster P2P File Sharing System Based on IPFS and Blockchain Technology. Journal of Ambient Intelligence and Humanized Computing. 2019. https://doi.org/10.1007/s12652-019-01453-5
3. Zheng Q, Li Y, Chen P, Dong X. An innovative IPFS-based storage model for blockchain. International Conference on Web Intelligence (WI), Santiago, Chile. 2018:704-708.
4. Huang H, Chang T, Wu J. A Secure File Sharing System Based on IPFS and Blockchain, Proceedings of the 2nd International Electronics Communication Conference (IECC), New York, USA. 2020: 96-100.
5. Chen Y, Li H, Li K, Zhang J. An improved P2P file system scheme based on IPFS and Blockchain. IEEE International Conference on Big Data (Big Data), Boston, MA, USA. 2017: 2652-2657.

6. Kumar R, Tripathi R. Implementation of Distributed File Storage and Access Framework using IPFS and Blockchain. International Conference on Image Information Processing (ICIIP), Shimla, India. 2019:246-251.
7. Nizamuddin N, Salah K, Ajmal Azad M, Arshad J, Rehman MH. Decentralized Document Version Control Using Ethereum Blockchain and IPFS. Computers & Electrical Engineering.2019;76:183-197.
8. Almasian M, Shafieinejad A. Secure cloud file sharing scheme using blockchain and attribute-based encryption. Computer Standards & Interfaces. 2024 Jan 1; 87:103745.
9. Naz M, Al-zahrani FA, Khalid R, Javaid N, Qamar AM, Afzal MK, Shafiq M. A secure data sharing platform using blockchain and interplanetary file system. Sustainability. 2019 Dec 10; 11(24):7054.
10. Gao H, Duan P, Pan X, Zhang X, Keke Ye, Zhong Z. Blockchain-Enabled Supervised Secure Data Sharing and Delegation Scheme in Web3.0. Journal of Cloud Computing. 2024; 13(21):1-14
11. Panescu AT, Manta V. Smart Contracts for Research Data Rights Management over the Ethereum Blockchain Network. Science & Technology Libraries. 2018; 37(3): 235-245.
12. Nizamuddin N, Hasan H, Salah K, Iqbal R. Blockchain-Based Framework for Protecting Author Royalty of Digital Assets. Arabian Journal for Science and Engineering, 2019; 44(4):3849-3866.
13. Hasan HR, Salah K. Proof of delivery of digital assets using blockchain and smart contracts. IEEE Access. 2018; 6:2169-3536.
14. Park JS, Youn TY, Kim HB, Rhee KH, Shin SU. Smart contract-based review system for an IoT data marketplace. Sensors. 2018;18(10):3577.