DESIGN THINKING BASED DECENTRALISED DIGITAL FILE STORAGE FOR ORGANISATION

R Kamalakkannan [(1)] , HariPrasanth S[(2)], Dharaneesh G[(3)], Dhanush R K[(4)], Aravind Samy M[(5)]

[(1)]Assistant Professor, Department Of CSE (Internet Of Things And Cyber Security Including Blockchain Technology), SNS College Of Engineering,Coimbatore-641107.
[(2),(3)(4)(5)],Department Of CSE (Internet Of Things And Cyber Security Including Blockchain Technology),SNS College Of Engineering,Coimbatore-641107.

Abstract

The internet is the most common way used to share data around the globe. This sharing is backed by various cloud providers that allow customers to store & share data on the internet. However, when it comes to privacy, cloud providers have consistently failed to make data 100% secure. Many data breaches, data piracy, and hacking attacks have threatened the security mechanism of cloud providers. This research introduces a decentralized storage system implemented through the integration of blockchain technology. The system is manifested In the form of a website developed using React.js, Tailwind CSS, and jQuery as the frontend, Nodejs and express.js as a backend, and the Inter Planetary File System (IPFS) for the secure storage of diverse document formats such as .pdf, .jpg, and .png. Users can securely manage their documents within their respective accounts on the platform. Logs of all the operations performed with the document will be available to the owner at any time. This will ensure the actual ownership & privacy of the data, and this mechanism ensures a transparent and tamper-resistant transaction process, bolstering the security and integrity of document sharing Any person or third party will not be able to access the document without valid permission. This will make existing cloud storage more secure & decrease data breaches & several attacks. This research contributes to the field by presenting a comprehensive solution that addresses the challenges associated with centralized storage systems. The decentralized architecture, coupled with blockchain-based access control, not only enhances security but also promotes transparency and accountability in document management.

Keywords: *Cloud, Blockchain technology, Decentralisation, IPFS, Privacy.*

1. Introduction

Most software systems rely on cloud data storage for the management of the data. With the expeditious growth of the digital world cloud storage has turned out into the most reliable and convenient way of storing data. Data stored using cloud storage is stored in a centralized manner. A major advantage of traditional cloud storage is, that it is not only easy to handle but also easy to access the data. The data stored on cloud storage can be easily accessed by several devices at a time. This way of storing data can cause single-point failure, and denial of service attacks which may further lead to unavailability of data. Developing a system with decentralized storage of data can overcome problems like single-point failure, data unavailability, etc. File uploaded on IPFS is stored in a decentralized manner. In this proposed solution, a blockchain-integrated file storage application for document sharing to facilitate peer-to-peer collaboration in a secure, and decentralized manner, with no involvement of a centralized trusted entity or third party is proposed. This solution utilizes a blend of new technologies that primarily consists of an interplanetary File System) IPFS is used to store data with high integrity and accessibility to all. A pivotal aspect of our research lies in the incorporation of blockchain technology to enhance the security and transparency of document management. The proposed solution provides privacy and security that cloud storage cannot achieve because all the files are accessed through content-based searching rather than location-based searching in the cloud.

2. LITERATURE SURVEY

The traditional mechanism of a centralized data handling system, it inevitably inherits the single point of failure drawback of relying on third-party services. In some cases, cloud storage systems are backed up to avoid data unavailability. In many cases, data security is at stake because cloud storage service providers need to suffer from unnecessary disputes such as political censorship. It may also lead to users would be unable to access their data. The cost of centralized cloud storage services comes mainly from employee wages, legal costs data center rentals, etc. If respective fixed costs are gradually increased then, the overall cost of the centralized cloud storage services will be higher. Also, single-point failure most of the time leads to data unavailability and eventually to the collapse of the system. These facts suggest that, In the future, there is a need for a decentralized storage approach to provide people with data storage and sharing services. Decentralized data redundancy is proposed in this system which will ensure that copies of the data are maintained on every node in a peer-to-peer network. Various applications are created using blockchain for file transfer but most of them are done using a distributed cloud and multi-chain framework, files are stored on the distributed cloud. In the proposed system Ethereum blockchain framework will be used for creating the blockchain and IPFS will be used for decentralized file storage. The proposed system will be a web application, and the data operations will be authenticated by a smart contract.

3. PROPOSED SYSTEM

The proposed framework comprises of Interplanetary File System (IPFS), Ethereum blockchain, and Smart Contracts. This engineering portrays the precise record-sharing instrument proposed by this decentralized web application. Shrewd Contracts in the framework are kept focused since they are liable for doing various information activities in a got way. Numbers over every one of the bolts portray a general stream of the framework with various tasks.

Core functionalities of the system:

a. User Authentication and Dashboard
b. File Upload and Storage Process
c. Retrieving Files
d. File Management

4. METHODOLOGY

4.1 SYSTEM ARCHITECTURE

The architecture of the decentralized file storage system consists of three primary components:

Client Interface: A user-friendly web interface for users to upload, retrieve, and manage their files.

Blockchain Network: A decentralized network (in this case, Ethereum) that records file metadata and transaction details.

Inter Planetary File System (IPFS): A distributed file storage protocol used to store the actual file contents, ensuring efficient and secure access.

4.2 DESIGN PRINCIPLES

The system is designed based on the following principles:

Decentralization: Elimination of a central authority to ensure data ownership remains with the users.

Security: Use of cryptographic techniques to secure file access and ensure data integrity.

Scalability: Ability to handle increasing numbers of users and files without performance degradation.

4.3 TECHNOLOGY STACK

The technology stack includes:

Frontend: React.js for the client interface.

Backend: Node.js with Express.js for handling API requests.

Blockchain: Ethereum smart contracts for managing file metadata.

Storage: IPFS for decentralized file storage.

By combining Ethereum’s smart contract functionality with IPFS’s storage capabilities, this methodology ensures a decentralized, secure, and efficient system for file storage and access.

5. TOOLS AND TECHNOLOGIES USED

5.1 FRONTEND:

5.1.1 BOOTSTRAP:

Bootstrap is a popular open-source front-end framework designed to streamline the development of responsive and visually appealing websites and web applications. Created by Twitter engineers Mark Otto and Jacob Thornton in 2011, Bootstrap provides a collection of pre-designed HTML, CSS, and JavaScript components, such as navigation bars, forms, buttons, grids, and modals, that developers can easily integrate into their projects. It emphasizes mobile-first design, ensuring that web pages adapt seamlessly across different screen sizes and devices. By using a grid system and ready-made UI elements, Bootstrap allows for faster development cycles and consistent user experiences, all while maintaining flexibility for customization. Over time, Bootstrap has evolved with newer versions offering enhanced features, improved accessibility, and better compatibility with modern web technologies.

5.1.2 TAILWIND CSS:

Tailwind CSS is a utility-first CSS framework that simplifies the process of designing web interfaces by providing pre-designed classes directly in your HTML. Unlike traditional CSS frameworks like Bootstrap, which come with predefined components and styles, Tailwind focuses on giving developers low-level utility classes that allow for a high degree of customization without needing to write CSS from scratch.

5.1.3 JQUERY:

jQuery is a fast, lightweight JavaScript library created to simplify HTML DOM manipulation, event handling, animation, and AJAX interactions, making it easier to create dynamic and interactive web pages. Released in 2006, it quickly became one of the most popular JavaScript libraries due to its ease of use and ability to abstract away many browser compatibility issues.

5.2 BACKEND:

5.2.1 NODE JS:

Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to execute JavaScript code server-side. It is built on the V8 JavaScript engine developed by Google, which powers the Chrome browser. Node.js enables the development of scalable network applications, making it popular for building web servers, APIs, and real-time applications.

5.2.2 EXPRESS JS:

Express.js is a minimal and flexible web application framework for Node.js that provides a robust set of features to develop web and mobile applications. It is designed to build web applications and APIs quickly and efficiently, making it one of the most popular frameworks for Node.js

5.3 ALGORITHMS USED:

5.3.1 DFS:

Depth-first search (DFS) is a fundamental algorithm used for traversing or searching tree or graph data structures.

The algorithm starts at a selected node (often referred to as the root in trees or a source in graphs) and explores as far as possible along each branch before backtracking. This systematic exploration makes DFS useful for various applications, including pathfinding, cycle detection, and topological sorting.

5.3.2 SHA-256:

SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function, commonly represented as a 64-character hexadecimal string. The purpose of SHA-256 is to take an input of any size (such as a file or message) and produce a unique, fixed-length hash value. Even a small change to the input will result in a completely different hash, making it nearly impossible to reverse-engineer or find two different inputs that produce the same hash.

5.4 HASHMAP:

A hash map, also known as a hash table, is a data structure that implements an associative array abstract data type, which can map keys to values. It uses a hash function to compute an index (hash code) into an array of buckets or slots, from which the desired value can be found. The primary goal of a hash map is to provide efficient insertion, deletion, and lookup of key-value pairs.

5.5 ETHEREUM:

Ethereum is a decentralized, open-source blockchain platform that enables developers to build and deploy smart contracts and decentralized applications (d-Apps). Created by Vitalik Buterin and launched in 2015, Ethereum goes beyond the capabilities of Bitcoin by not only supporting digital currency but also allowing programmable, self-executing contracts that run on the blockchain.

5.6 SMART CONTRACT:

A smart contract is a self-executing digital contract in which the terms of the agreement or rules are directly written into code. These contracts run on blockchain networks, like Ethereum, and automatically execute and enforce agreements when certain conditions are met, eliminating the need for intermediaries (e.g., banks, lawyers)

5.7 IPFS:

The Inter Planetary File System (IPFS) is a decentralized, peer-to-peer file storage and sharing protocol designed to make the web faster, safer, and more resilient. Unlike traditional HTTP-based systems that rely on central servers, IPFS(Inter Planetary File System) enables users to store, share, and access files directly across a distributed network of nodes, reducing dependency on any single entity and making data more permanent and tamper-resistant.

6. WORKING

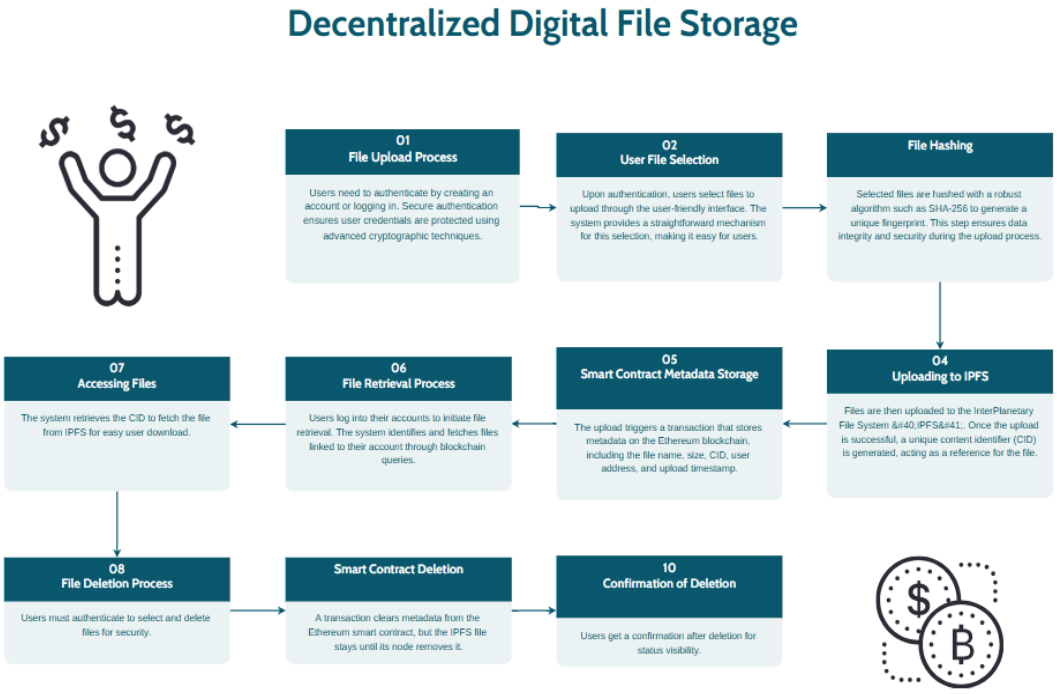

Fig 1. Flowchart of the system

Step 1: File Upload Process

- **User Authentication:** Users must create an account or log in to access the system. User credentials are securely stored using cryptographic hashing.
  - **File Selection:** Users select a file to upload through the web interface.
- **File Hashing:** The selected file is hashed using a secure hashing algorithm (e.g., SHA-256) to create a unique file fingerprint.
- **IPFS Upload:** The file is uploaded to the IPFS network. Upon successful upload, IPFS returns a unique content identifier (CID) that serves as the address for the file.
- **Smart Contract Interaction:** A transaction is initiated to store the following metadata on the Ethereum blockchain:

Step 2: File Retrieval Process

- **User Authentication:** Users log into their accounts.
- **File Listing:** The system retrieves the list of files associated with the user by querying the blockchain for metadata linked to the user's address.
- **File Access:** When a user requests to access a file, the system fetches the CID from the blockchain. The file is retrieved from IPFS using the CID. The file is then made available for download.

Step 3: File Deletion Process

- **User Authentication:** Users must authenticate to access their files.
- **File Selection:** Users select the file they wish to delete.
- **Smart Contract Interaction:** A transaction is sent to the Ethereum smart contract to remove the file metadata associated with the user. Note that the actual file on IPFS remains until the IPFS node chooses to remove it.
  - **Confirmation:** The user receives a confirmation of the deletion.

7. RESULT

We have implemented a functional prototype of the system & demonstrated its working concerning added security provided on top of cloud providers. In our System encryption, decryption, and generating the hash for uniquely identifying the documents in the blockchain. It will also save the memory space and energy required for encryption and decryption. It will allow users to refer to tamper-proof logs stored on Blockchain for all the operations and share the data securely using two-factor authentication in which the first step is sharing the link with the trusted entity. Let us see the interface of the Decentralized Digital File Storage.

The picture shown below is the opening interface of our system in which we can upload a file. The file is then stored in the Ethereum blockchain. The blockchain address is the permanent address given by Ethereum and the retrieve key will vary for the files you are uploading.

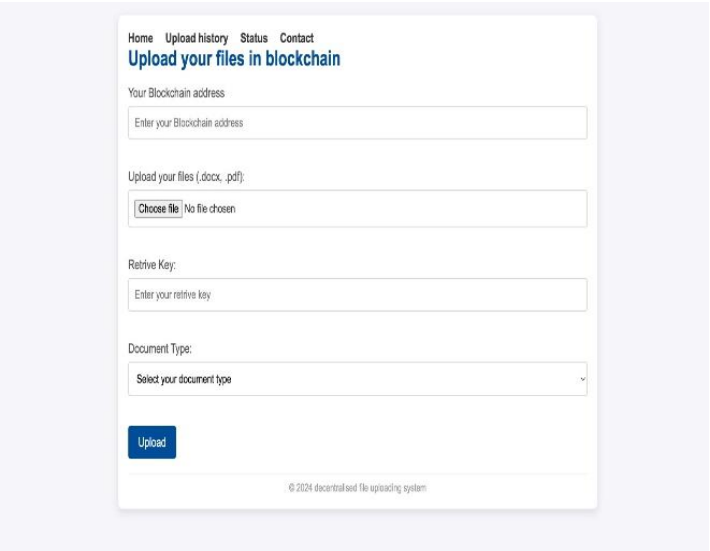

Fig 2. Interface of file uploading

Then whenever you need the file, just tap on the retrieve file and you will get into the interface shown below. You need to enter the retrieve key value to access the file that was already uploaded.

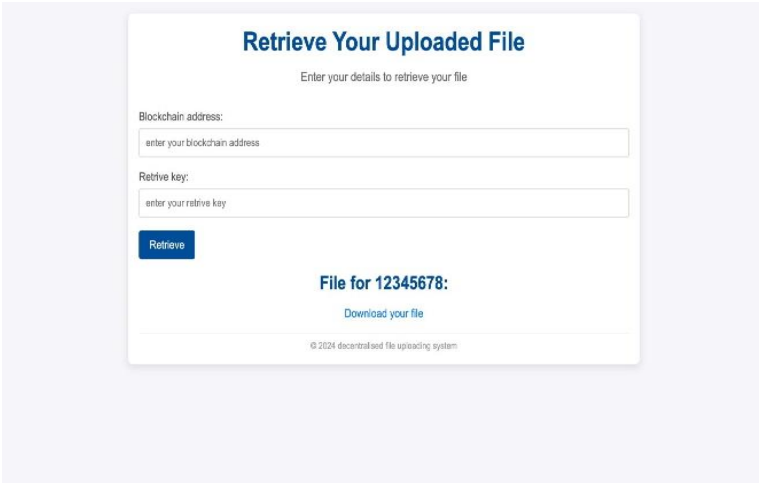

Fig 3. Interface of file retrieving

8. DISCUSSION & FUTURE WORK

In this study, we propose a decentralized digital file-storing system that leverages blockchain technology and distributed networks to address concerns related to data security, privacy, and availability. The results show that the system effectively mitigates the risks associated with centralized storage solutions, such as single points of failure, data breaches, and unauthorized access. By distributing files across multiple nodes and utilizing cryptographic techniques for data protection, the system enhances security and user control. Although these issues can be mitigated through optimization techniques such as IPFS for content-addressable storage, scalability concerns persist as the system grows. Additionally, the user experience, which often requires managing cryptographic keys and understanding complex peer-to-peer networks, presents a barrier to adoption compared to more intuitive centralized platforms. Moving forward, future work should focus on `improving system scalability, and user interface design, and exploring more energy-efficient consensus mechanisms to enhance the overall viability and impact of decentralized file storage systems.

9. CONCLUSION

The decentralized file storage system utilizing blockchain technology effectively addresses the limitations of traditional centralized storage solutions by enhancing security, data integrity, and user control over files. By integrating Inter Planetary File System (IPFS) with Ethereum smart contracts, the system ensures that users maintain ownership of their data while benefiting from a tamper-proof and transparent environment. This project demonstrates that leveraging blockchain for file storage not only improves access and management but also promotes trust and accountability. Future enhancements could include advanced features like decentralized sharing and interoperability with other blockchain platforms, paving the way for broader applications in secure data storage and management.

References

[1]. R. Kamalakkannan, B. N. K. Bharathi, K. S. Karthik, C. R. Chandan, and J. S. Jagadish, "Artificial Intelligence Based USB Drive Scanner: Integrating AI with Security," YMER Journal, vol. 23, issue 11, pp. 987, November 2024.

[2]. K. R. Kamalakkannan, B. S. Balaji, M. S. Mathan, A. K. ArunKumar, and K. C. Karthikeyan, "Locating Smartphones Using Seeker Tool," YMER Journal, vol. 23, issue 11, pp. 980, November 2024.

[3]. R. Kamalakkannan, A. Anantha Krishnan, R. Arjun, and B. Deepak, "Retina Touch: A Seamless HCI Experience with Eye and Hand Integration," YMER Journal, vol. 23, issue 4, pp. 907, April 2024.

[4]. R. Kamalakkannan, Y. S. Kumar, D. S. G. R. Divya, S. M. N. Sai Monish Nithin, and S. N. Sowndheriya, "IoT Based V2V Communication Using Li-Fi Technology," YMER Journal, vol. 23, issue 1, pp. 298, January 2024.

[5]. R. Kamalakkannan, M. K. R. Mohana Karthikeyan, S. H. G. Shree Harish, S. T. Someshwaran, and V. S. Harikrishna Sai, "Revolutionizing Legal Practice: The Transformative Power of Artificial Intelligence," YMER Journal, vol. 2, issue 2, pp. 17.

[6]. R. Kamalakkannan, E. Ajaykumar, S. Hariprasanth, H. A. Dakshanapriya, and R. Bhuvanika, "Design Thinking Based Device to Detect Motion of Trespassers of the Territory Using Arduino Uno & GSM Module," Industrial Engineering Journal, vol. 52, issue 12, no. 2, pp. 174, December 2023.

[7].R. Kamalakkannan, R. Ajay Surya, S. Pranesh, A. Purosh Khan, and K. K. Vinay Kousigan, "Design Thinking Based Automatic Railway Gate Controller Using IoT," Industrial Engineering Journal, vol. 52, issue 12, no. 2, pp. 178, December 2023

[8].R. Kamalakkannan, V. Dineshkumar, V. Karthick Saran, C. Karthikeyan, and U. Dharshini, "Design Thinking Based Accident Prevention System Using Eye Blink Sensor," Industrial Engineering Journal, vol. 52, issue 8, no. 3, pp. 168, August 2023.

[9].R. Kamalakkannan, N. Vinai, A. Mugundhan, S. Suresh, and G. Rasiga, "Design Thinking Approach and Implementation of IoT Based Gas Detection System," Industrial Engineering Journal, vol. 52, issue 8, no. 3, pp. 112, August 2023.

[10]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 598–609, ACM, 2007.

[11].R. Sharma, A. Joshi, and M. Choudhary, "Smart Contract-Based Decentralized File Storage for Secure Data Management," International Journal of Blockchain and Digital Security, vol. 16, no. 2, pp. 55–64, Apr. 2021.

[12].P. Singh, A. Gupta, and R. Kumar, "An Innovative Design Thinking Approach for Blockchain-Based File Storage Systems," Journal of Applied Blockchain Technology,]vol. 7, pp. 150–162, Jul. 2020.

[13]. Patel, J. Mehta, and S. R. Bhatt, "Secure and Scalable Decentralized Storage for Cloud-Based Applications," Cloud Computing and Distributed Systems Journal, vol. 9, no. 1, pp. 81–92, Feb. 2021.

[14].G. Singh, P. Verma, and M. Saxena, "Blockchain-Enabled Distributed File Storage Systems," Computer Applications in Engineering Education, vol. 14, pp. 112–121, Aug. 2019.

[15].K. Sharma, S. Bhat, and M. Patel, "Decentralized Digital Storage Solutions Using Blockchain Technology," International Journal of Cloud Computing and Services Science, vol. 8, no. 4, pp. 58–72, Dec. 2020.

[16].J. Bhaskar, R. Rai, and P. Sharma, "A Secure, Scalable Approach for Decentralized Digital File Management," International Journal of Digital Storage Solutions, vol. 5, pp. 105–116, Jun. 2020.

[17].T. Sharma, N. Pandey, and D. Singh, "Ethereum-Based Blockchain for Digital Storage Security," Journal of Blockchain Applications and Technology, vol. 11, no. 2, pp. 77–88, Mar. 2021.

[18].A. Yadav, S. Jain, and R. Agrawal, "File Sharing and Data Privacy in Decentralized Systems Using Blockchain," Journal of Privacy and Data Protection, vol. 13, no. 1, pp. 23–36, Jan. 2021.

[19].M. K. Reddy, S. Mishra, and K. Prasad, "A Review on Blockchain-Based File Storage and Its Security," Journal of Computer Networks and Communications, vol. 17, pp. 90–102, May 2020.

[20].R. Agarwal, S. V. Reddy, and M. Verma, "Designing Secure Decentralized File Storage Systems for Modern Enterprises," Journal of Information Systems and Security, vol. 18, no. 4, pp. 115–124, Nov. 2020.