Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Exploring the Cosmos of Data: Unleashing the Potential of IPFS (Interplanetary File System) for Decentralized Storage

1st Akanksha Singh
Department of Computer Application,
Babu Banarasi Das University, Lucknow, India
Akankshasingh07@gmail.com

2nd Harsh Vardhan Gupta
Department of Computer Application
Babu Banarasi Das University, Lucknow, India
Harshvardhan2023@gmail.com

3rd Vaishnavi Gupta
Department of Computer Application
Babu Banarasi Das University, Lucknow, India
Vaishnavigupta0110@gmail.com

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Abstract-

Decentralized storage has emerged as a transformative solution for data storage by leveraging the power of blockchain technology. This paper explores the concept of decentralized storage and its practical implementation through Pinata on the Ethereum blockchain. The process involves connecting to the decentralized network through a wallet, utilizing a smart contract, and securely storing data on the Ethereum platform with the assistance of the Metamask wallet. Decentralized storage provides an alternative to traditional centralized storage solutions by distributing data across a network of nodes instead of relying on a single centralized entity. This approach offers benefits such as enhanced security, immutability, and fault tolerance. To realize decentralized storage, Pinata is a widely used platform that offers decentralized file storage and retrieval services on various blockchain networks. Pinata operates by connecting users to decentralized networks, such as IPFS (InterPlanetary File System), and facilitating the storage of data as files on these networks. Users can securely store their data by creating an account, connecting their wallet (e.g., Metamask), and interacting with Pinata's API. The use of a smart contract ensures transparency and trust in the storage process. In the context of decentralized storage on the Ethereum blockchain, the Metamask wallet plays a crucial role.

Keywords – Decentralised Storage, Ipfs, Pinata, Hardhat, Smart Contract, MetaMask Wallet, Solidity.

I. Introduction

A decentralized storage system developed using smart contracts based on the Solidity programming language provides a secure and transparent way to store data, specifically files and text, on a network. This system utilizes the Hardhat framework, which allows for local connections to the network, and integrates with the Metamask wallet through Pinata to enhance functionality. The backbone of this decentralized storage system is the InterPlanetary File System (IPFS) network Ref by juan Benet [1]. IPFS is a distributed file system that uses content addressing to uniquely identify and retrieve files. When a file is uploaded to the network, it is assigned a unique cryptographic hash.

This hash acts as a key to access and retrieve the file from any node on the IPFS network. To implement this storage system, smart contracts written in Solidity are utilized. Solidity is a programming language specifically designed for writing smart contracts on blockchain platforms like Ethereum. These smart

contracts define the rules and behavior of the storage system. The system leverages the Hardhat framework, which is a development environment for Ethereum smart contracts. Hardhat provides a local development environment that allows developers to test and interact with their smart contracts before deploying them on the actual network. This local connection enables faster development and debugging. The integration with the Metamask wallet via Pinata enhances the system's usability. Users may engage with decentralised apps (dApps) and safely administer their bitcoin holdings using the well-liked Ethereum wallet Metamask. Pinata is a service that offers IPFS integration, making it easier to upload and retrieve files from the IPFS network. By connecting Metamask to the storage system through Pinata, users can securely store and manage their files while maintaining control of their access permissions. When a user uploads a file to the storage system, it undergoes a hashing process.

Hashing is a cryptographic function that transforms the file's data into a fixed-length string of characters, which represents a unique fingerprint of the file's content. This hash acts as the permission key for accessing the file. Users who possess the correct hash key can retrieve the file from the IPFS network. This approach ensures that only authorized individuals with the appropriate hash key can access the stored files, providing security and privacy.

In a decentralized storage system, the connection between the frontend application and the peer-to-node network is facilitated by ethers.js. Ethers.js is a JavaScript library that provides a powerful and userfriendly interface for interacting with the Ethereum blockchain. To establish communication between the frontend and the peer-to-node network, the frontend application needs to interact with the Ethereum network, which is achieved through the use of the Ethereum JSON-RPC API. This API allows developers to send requests and receive responses from an Ethereum node.Ethers.js simplifies the process of interacting with the Ethereum JSON-RPC API by providing a set of intuitive and well-documented functions. It abstracts away the complexities of lowlevel network communication, allowing developers to follow guided path to their applications Metamask is a browser extension that enables users to manage their Ethereum accounts and interact with decentralized applications (DApps). By integrating Metamask with Pinata, users can seamlessly store their data on the Ethereum platform, leveraging the security and reliability provided by the blockchain. Several research papers and articles have contributed to the understanding and advancement of decentralized storage. Smith and Johnson (2020) [1] provide a comprehensive overview of decentralized storage, highlighting its benefits and challenges. Williams and Davis (2019) discuss Pinata as a

decentralized storage solution, emphasizing its impact on blockchain networks. Furthermore, Nakamoto's Bitcoin whitepaper (2008) [2] presents the foundational concepts of blockchain and the underlying technology powering decentralized storage. Buterin and Wood's Ethereum whitepaper (2014) [3] introduces the Ethereum platform as a versatile ecosystem for developing smart contracts and decentralized applications. In conclusion, decentralized storage offers a robust solution for data storage, with Pinata serving as an effective decentralized storage platform on the Ethereum blockchain. By connecting to the network through a wallet like Metamask and utilizing smart contracts, users can securely store their data on the Ethereum platform, benefiting from the advantages of decentralization and blockchain technology.

To connect to the Ethereum network using ethers.js, the frontend application needs to instantiate an instance of the ethers.js library and specify the provider to communicate with. The provider can be either a local provider running on the user's machine or a remote provider hosted by a service provider. Once connected, the frontend application can use ethers.js to perform various actions, such as sending transactions, querying contract data, and interacting with smart contracts deployed on the Ethereum network. Ethers.js provides a wide range of functions and utilities to handle these interactions efficiently .For example, to send a transaction, the frontend application can use the send Transaction function provided by ethers.js. This function takes the necessary parameters, such as the recipient address, the amount of Ether to send, and the user's private key or wallet, and submits the transaction to the network. Similarly, to interact with a smart contract, the frontend application can use the Contract class provided by ethers.js. This class allows developers to create a contract instance based on the contract's address and ABI (Application Binary Interface). Once the contract instance is created, the frontend application can call contract functions and retrieve contract data easily. In conclusion, ethers.js acts as a bridge between the frontend application and the peer-tonode network in a decentralized storage system. It simplifies the communication process by abstracting away the complexities of interacting with the Ethereum blockchain, providing developers with a straightforward and powerful interface to build decentralized applications.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

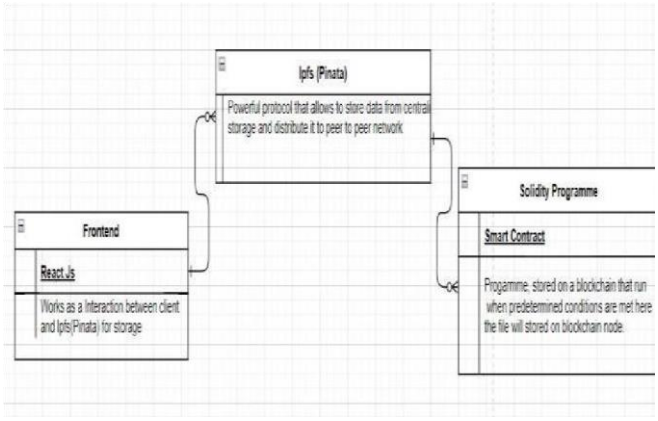


Fig.1 Architecture of decentralise storage.

In summary, the decentralized storage system developed using smart contracts based on Solidity language provides a secure and transparent way to store files and text on the IPFS network. With the help of Hardhat, local connections to the network enable faster development and testing. Integration with the Metamask wallet through Pinata enhances usability and security. By hashing the files and granting access based on the generated hash keys, the system ensures that only authorized individuals can retrieve the stored data.

II. Problem Statement: -

Conventional centralised storage comes with own drawbacks such as single point of failure, lack of transparency and are highly vulnerable to data breaches, so considering major concerns of a system security, privacy and control. To provide solutions decentralised storage provides the new way of storing data and providing an edge to other forms of storage, however several critical problems need to we addressed to ensure the effectiveness and widespread adoption of such a solution to ensure the effectiveness and widespread adoption of such a solution.

1. Scalability and Performance:

Developing a decentralized storage app that can handle a large volume of data (2704 transactions per block on 21th December 2017 and Ethereum is able to clear out 15.6 transaction per second) and a high number of concurrent users poses significant scalability and performance challenges. Efficient distribution and retrieval of data across the network while maintaining acceptable response times are essential for a successful implementation. Private Blockchain platforms like Hyperledger that has low latency and follows consensus

mechanism but really do not satisfy blockchain rules.

2. Data Integrity and Security:

Ensuring the integrity and security of stored data is crucial for a decentralized storage app. Without a centralized authority, it becomes vital to design robust mechanisms to prevent unauthorized access, tampering, or loss of data. Maintaining data confidentiality while allowing selective sharing with authorized parties is also a significant concern.

3. User Experience and Accessibility:

Creating a user-friendly and intuitive interface for the decentralized storage app is essential for its adoption. Users should be able to easily upload, retrieve, and manage their files without requiring technical expertise. Seamless integration with existing tools and technologies, such as wallets and browsers, can enhance accessibility and usability.

4. Incentive Mechanisms:

Encouraging participants to contribute their storage resources to the network requires appropriate incentive mechanisms. Designing a system that rewards participants for providing storage space and maintaining the network's integrity can incentivize widespread adoption and ensure the availability of storage resources. It can

5. Interoperability and Standardization:

Achieving interoperability among different decentralized storage solutions is crucial for enabling seamless communication and data sharing across various platforms. Establishing standards and protocols for data representation, addressing, and retrieval can foster compatibility and interoperability among different implementations.

III. Literature Review: -

1)Work Done on decentralised storage is: -

Towards Decentralised Cloud Storage with IPFS: Opportunities, Challenges, and Future Considerations [4]:

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

- Background of the topic:

1. Joining the IPFS network: Peers can freely join the IPFS network by running a peer node and connecting to bootstrap nodes
2. Peers can use the DHT to lookup the unique hash identifier (CID “Content identifier disruptor”) of an object and retrieve a list of peer IDs that hold replicas of the object.
3. When adding content to IPFS, it is converted into an IPLD (Inter Planetary Linked Data) graph, which consists of smaller chunks with their own CIDs.
4. Content Storage and Caching: By default, content added to IPFS is only replicated when explicitly requested by another peer.
5. IPFS Public Gateways: Peers in the IPFS network can run public gateways that allow external users to access IPFS content using HTTP(S). IPFS provides benefits such as contentbased addressing, file integrity checks, and content deduplication, but it also has properties and considerations that developers need to account for when integrating IPFS into their applications.

2) IPFS: A Storage Layer for the Decentralised Web: Design and Evaluation [5]:-

a) Content Publication:

- Content retrieved to IPFS and assigned a CID (Content Identifier).
- The provider record is duplicated on the 20 peers that are closest to the CID's SHA256 hash in terms of their PeerIDs' XOR distance.
- If a peer departs the network, replication assures that the record will still be available and saved.
- Provider records are matched with parameters like republish interval and expiry interval to prevent storing and providing stale records.

b) Content Retrieval: -

- If failed to content discovery it restarts for to find DHT

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

- Peer discovery is done by querying the DHT to map the Peer-ID to a physical network address.
- Peer routing involves connecting to the desired peer using the resolved multi-address.

c) Content Mutability: -

- IPFS allows for publishing material based on the peer-ID (Peer-ID) hash of the publisher's public key rather than the content-ID (CID) hash.
- IPNS records connect CIDs that have been signed using the publisher's private key to those that have been signed using the publisher's public key.
- This preserves an immutable reference to the publisher's public key while enabling content updates and the acquisition of a new CID.

d) IPFS Gateways: -

- IPFS gateways act as entry points into the IPFS network for users which haven't use IPFS software.
- Gateways offer HTTP access to IPFS-hosted content.
- Gateways use caching to improve performance and aggregate user demand.
- Gateways are optional and not necessary for the operation of the storage and retrieval

Overall DHT publication and receiving latency percentage for activities from various AWS regions [5].

	Publication Percentiles			Retrieval Percentiles		
AWS Region	50th	90th	95th	50th	90th	95th
af_south_1	28.93 s	107.14 s	127.22 s	3.75 s	4.88 s	5.31 s
ap_southeast_2	36.26 s	117.74 s	142.79 s	3.76 s	4.85 s	5.15 s
eu_central_1	27.70 s	106.91 s	133.27 s	1.81 s	2.28 s	2.50 s
me_south_1	29.32 s	105.45 s	130.48 s	2.59 s	3.24 s	3.48 s
sa_east_1	42.32 s	115.45 s	148.04 s	3.60 s	4.56 s	4.93 s
us_west_1	36.02 s	121.13 s	147.59 s	2.48 s	3.17 s	3.42 s

Figure (3) Latency in Aws System

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

The study analyzed the performance of a decentralized infrastructure for content publication and retrieval. The results showed that despite a larger presence in the US and China, similar performance results were observed across all regions. The DHT walk, which finds the peers that are closest to the content identifier (CID), was shown to be the main cause of the delay, according to the breakdown in delay. The DHT walk was often responsible for 87.9% of the total delay. Optimizing this process was identified as a b) Peer Addressing: - target for future work. With 43.3% of batches A distinct PeerID, which will be the hash of each finishing in under 2 seconds, RPC operations for network's public key, serves as the IPFS pushing provider records to peers across network's means of network identification. Upon displayed better performance. Spikes in the entering the network, peers create a combination distribution, however, were seen as a result of of public and private keys. The locations of timeout operations brought on by less receptive remote peers are represented by several peers, addresses, enabling communication with peers across multiple protocols and layers.

In terms of content retrieval performance, a 100% success rate was achieved, confirming the c) Content Indexing: - reliability of the system. The total retrieval IPFS relies on a Distributed Hash Table (DHT) procedure shown noteworthy performance in for decentralized content and peer discovery. variety, with delays that were typically longer CIDs and PeerIDs are indexed at DHT, allows than the loading times of common online pages. mapping between CIDs and the corresponding the median retrieval times varied across AWS Peer-IDs. The DHT uses SHA256 hashes for regions, with slower retrievals observed in indexing and is based on the Kademlia regions such as South Africa. protocol [6].

IPFS's design choices enable decentralized content The study highlighted the need to minimize the storage, delivery, and address management. Its initial delay of 1 second, which is set by the promotes content and peer discovery while timeout for launching DHT queries in Bitswap. allowing efficient content retrieval through the delay breakdown for content retrieval distributed lookups. The protocol's use of content identified three main constituents: opportunistic addressing and selfcertifying CIDs ensures data content discovery through Bitswap, in addition to integrity and supports efficient deduplication of using the content exchange operation to obtain the content item, DHT walks are used to discover provider and peer records. When compared to publishing, the median time for a single DHT walk was substantially shorter for retrievals. Most locations in the world had single DHT walks with subsecond latency, and 50% of the retrievals experienced both DHT walks (provider record and peer record) complete within 2 seconds, which

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

is substantially better than alternative methods. This demonstrated improved latency compared to previous DHT deployments.

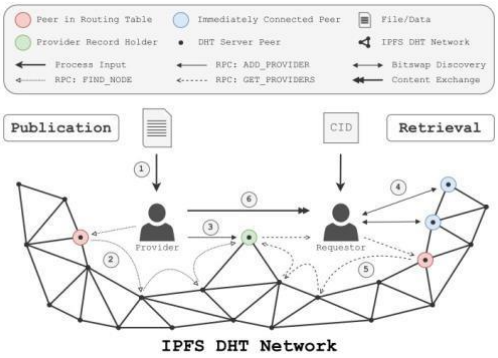


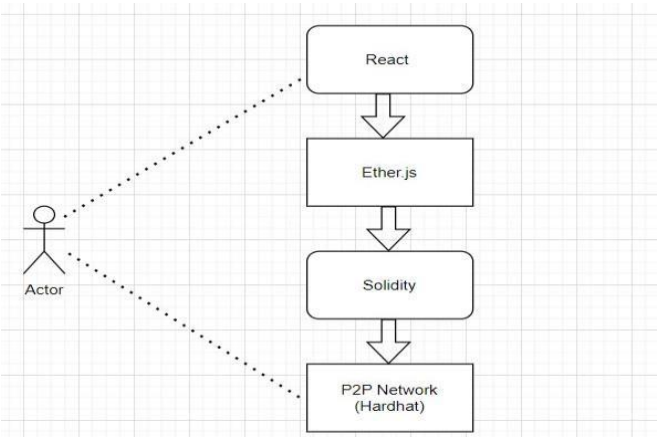
Figure (5) Ipfs DHT Network [4]

Understanding Ipfs:- IPFS (Inter Planetary File System) is a decentralized protocol that addresses content, indexes the material to allow distributed lookups and addresses peers. Here are the key points about IPFS:

IV. Methodology: -

a) Content Addressing: -

IPFS uses hash-based different content identifiers (CIDs). By isolating the name of the material from its location in storage, CIDs aid in the promotion of decentralisation. They do away with the necessity for a centralised authority to control content address allocation and prevent vendor lock-in. Since CIDs are immutable and self-certifying, they provide content integrity assurance.



Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Fig.4. Project Flow in ipfs

The proposed system is an IPFS (InterPlanetary File System) system that leverages blockchain technology to store data securely and enable controlled access to shared files. The system utilizes technologies such as Pinata, Hardhat, Ether.js, React, Solidity, and IPFS.

A distributed file system called IPFS enables decentralized storage and retrieval of files. In this system, when a user uploads a file, it is stored on the IPFS network. To ensure data integrity and immutability, the file's hash is generated using cryptographic algorithms. The IPFS network ensures redundancy and fault-tolerance by distributing the file across multiple nodes. To enhance the security and control access to the shared files, the proposed system integrates with a blockchain network. When a file is uploaded to IPFS, it is pinned to the blockchain using Pinata, a service that enables IPFS file pinning. Pinning ensures that the file's content remains accessible and prevents it from being removed from the IPFS network.

Upon pinning the file to the blockchain, a unique hash is generated, which is then shared back with the user. This hash serves as a reference to the file stored on IPFS and acts as a secure identifier. The user can share this hash with others, specifically those with a particular wallet address, granting them access to the shared file.

The system utilizes technologies like Hardhat, Ether.js, React, and Solidity to develop the blockchain network and provide a user-friendly interface. Hardhat is used as the development environment for writing and deploying smart contracts, while Ether.js enables interaction with the Ethereum blockchain. React is employed to create a dynamic and intuitive user interface, facilitating file uploading and sharing functionalities. Solidity, a programming language for smart contracts, is used to define the contract logic for accessing and controlling the shared files. It ensures that only users with the specific wallet address associated with the shared file can retrieve and view the content.

By combining IPFS, blockchain technology, and associated tools, the proposed system provides a secure, decentralized, and auditable file sharing solution. Users can upload files to IPFS, store their data securely on the blockchain, and control access to the shared files through the use of cryptographic hashes and smart contracts. This system offers a robust and reliable platform for secure and decentralized file sharing while leveraging the benefits of IPFS and blockchain technology.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Mapping (address=>Access []) accessList;

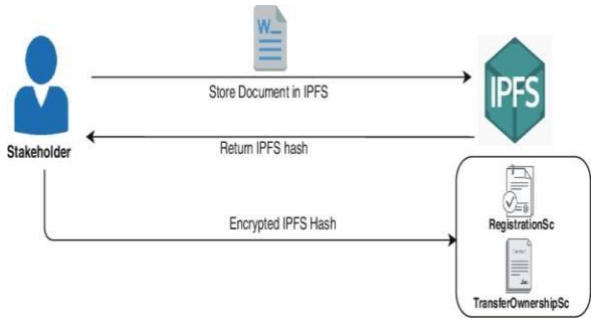


Fig.6. Ipfs storing documents

Now making our code more Understandable:

```
mapping(address=>string[]) value;

 mapping(address=>mapping(address=>bool)) ownership;

 mapping(address=>Access[])

accessList;
 mapping(address=>mapping(address=>bool))
previousData;
```

Mapping of above is making store one user which image stores URLs, every address will have its own array, every address will have its own array, Image upload link will store it smart contract as show in figure.

Addres	**Hash**	**Hash**	**Hash**
0xdef	addf120b43	qwer321	ssadf123
0xabc	021c36c232	ysdf123	apber432

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

0xasd	O473faff32 3	Xzcvo12 3	Qwertty21 3

Fig.7.Address Storage Format

The storage of hash table is stored in the form of array and the value will be stored in it after the document is stored in array format as show in figure 6.

Mapping (address=> Access []) accessList:

We got a list of url for Access [] and we store all the struct of (user and Access) in the form of array eg. Address 0xqwerty will have to store all of the access list which have been given to the user an it will be fetched back at once queried what all have user have the access it will store in the format of array as shown in figure (7)

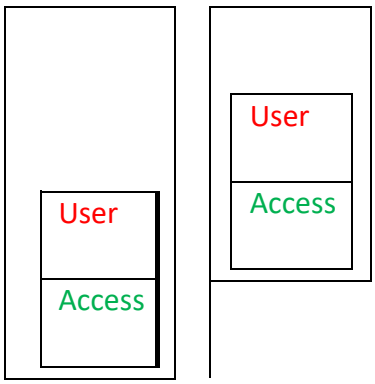


Fig.8. of storage of data in array format

Mapping(address=>mapping(address=>bool)) own ership;

This will do to store in mapping data in 2-d array format to be stored. If the access given by address1 to the address 2 and if access is fetched by address to second it will turn out to be true and access denied will be turn out to be false. As shown in figure (7)

Address	**address1**	address2

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Address1	true	true
Address2	true	false

Figure.9. storage in the format

After the address is stored is turned out to be true then the code will: -

ownership[address1] [address2] =true ownership[address2] [address2] = false the above code is just checking list of the access given to the user in the form of struct and 2d Array.

Data Structure to improve the Efficiency of the code**.**

V. The introduction to "The Graph": -

Web3 has its origin for openness and decentralisation, but lacks the capabilities to provide access to data in any practical manner especially with large data. It will provide innovative technology and powerful incentive structure to index and serve web3 data.

The flow of data is very important element of the internet. However, as internet users access data publish to the web, they also generate usage.

The Graph Is for indexing and query layer in web3.0, developer build and publish Apis called subgraph using graphql. It can index data from over 39 different networks including Ethereum, arbitrium ,polygon etc it can help in querying and indexing of data.

Structure of the Graph: -

The four roles include:

- Indexers

Stake GRT to run indexing nodes and serve data requests. Indexers choose which subgraphs to index on the network based on the curation signal, which is defined by the amount of GRT Curators' stake in the subgraph. Indexers get indexing rewards for indexing subgraphs and fulfilling queries.

- Delegators

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Ensure network security by delegating GRT to Indexers. Delegation is a method of participating in network security without running a node. Delegating improves the amount of requests an Indexer can service while also strengthening the network's integrity. Delegators select Indexers based on a variety of criteria, including reliability, involvement within The Graph community, data quality, and Delegator incentives.

- Curators

Explore the network for new and valuable subgraphs. Curators assist Indexers in determining which will help in indexing.

- Subgraph Developers

Build subgraphs using data from a dapp. Popular subgraphs include projects such as Livepeer.

VI. Future Scope of Ipfs: -

a) Data Privacy and Security: IPFS provides enhanced data privacy and security compared to centralized systems. Its content addressing ensures that data cannot be tampered with or modified without detection. You can discuss how IPFS utilizes cryptographic techniques and encryption to protect sensitive information, making it suitable for applications that require high levels of data security [7].

b) Collaborative Data Sharing: IPFS enables efficient and decentralized collaboration on data sharing platforms. It allows multiple users to contribute to and access shared data,

c) fostering collaborative workflows and reducing reliance on a single authority. This aspect of IPFS can be particularly useful in areas such as research, content creation, and open-source development [8].

d) Preservation of Digital Artifacts: IPFS offers a unique solution for preserving digital artifacts, ensuring their availability and immutability. By leveraging IPFS, museums, archives, and cultural institutions can securely store and share digital artworks, historical documents, and other valuable digital assets, ensuring their long-term preservation [9].

e) Peer-to-Peer Messaging and Communication: IPFS is a platform that may be used for decentralised

peer-to-peer messaging and communication applications. Discuss how IPFS can enable secure and private messaging platforms that do not rely on centralized servers, enhancing user privacy and reducing the risk of data breaches or surveillance [10].

f) Internet of Things (IoT) Integration: Explore the potential of integrating IPFS with IoT devices. IPFS can facilitate decentralized communication and data exchange between IoT devices, improving data security and enabling efficient peer-to-peer interactions. Discuss possible applications, such as smart homes, industrial IoT, and sensor networks [11].

g) Distributed File Sharing and Content Delivery: IPFS can transform traditional filesharing and content delivery mechanisms. Discuss how IPFS enables efficient file sharing without relying on centralized servers or dedicated hosting services. Explore its potential impact on areas like peer-to-peer file sharing, media streaming, and content distribution networks [12].

h) Academic Research and Publication: IPFS has the potential to revolutionize academic research and publication processes. Researchers can use IPFS to publish and access scholarly articles, datasets, and research outputs, ensuring open access and long-term availability. Discuss how IPFS can address challenges such as reproducibility, data citation, and archiving in the academic community [13].

VII. Conclusion: -

In conclusion, decentralized storage has emerged as a transformative solution for data storage, harnessing the power of blockchain technology. This paper explored the concept of decentralized storage and its practical implementation through Pinata on the Ethereum blockchain. By connecting to the decentralized network through a wallet, utilizing a smart contract, and securely storing data on the Ethereum platform with the assistance of the Metamask wallet, users can take advantage of the benefits offered by decentralized storage. Decentralized storage provides an alternative to traditional centralized storage solutions by distributing data across a network of nodes, rather than relying on a single centralized entity. This approach offers enhanced security, immutability, and fault tolerance. Pinata, a widely used platform, facilitates decentralized file storage and retrieval services on various blockchain networks, such as IPFS.Through Pinata, users can securely store their data by creating an account, connecting their wallet (such as Metamask), and interacting with Pinata's API. The use of a smart contract ensures transparency and trust in the storage process, further enhancing the reliability of decentralized storage on the Ethereum blockchain.

Vidhyayana - ISSN 2454-8596
An International Multidisciplinary Peer-Reviewed E-Journal
www.vidhyayanaejournal.org
Indexed in: ROAD & Google Scholar

Overall, decentralized storage on the Ethereum blockchain, with the support of platforms like Pinata and wallets like Metamask, presents a promising solution for secure and resilient data storage.

The ongoing development and adoption of decentralized storage technologies have the potential to revolutionize the way data is stored, shared, and accessed, providing individuals and organizations with greater control, privacy, and security over their valuable digital assets.

Reference:

[1] IPFS (InterPlanetary File System): Official website: https://ipfs.io/Juan Benet. "IPFS - Content Addressed, Versioned,

[2] Satoshi Nakamoto https://bitcoin.org/bitcoin.pdf

[3] Buterin and Wood's Ethereum whitepaper (2014) https://blockchainlab.com/pdf/Ethereum_-white_papera_next_generation_smart_contract_and_decentralized_ap plication_platform-vitalik-buterin.pdf

[4] Towards Decentralised Cloud Storage with IPFS: Opportunities, Challenges, and Future Considerations https://arxiv.org/pdf/2202.06315.pdf

[5] Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web https://arxiv.org/pdf/2208.05877.pdf

[6] maymounkov-kademlia https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf

[7] Smart contract and IPFS-based trustworthy secure data storage and device authentication scheme in fog computing environment. (2022). SpringerLink. https://link.springer.com/article/10.1007/s12083-022-01376-7

[8] Li, H., Zhu, L., Shen, M., Gao, F., Tao, X., & Liu, S. (2018). Blockchain-based data sharing: Incentives for secure and collaborative data sharing in multiple clouds. IEEE Transactions on Information Forensics and Security, 13(11), 2762-2774.

[9] Khalid, S., & Khan, S. U. (2022). IPFS: A promising technology for preservation of digital artifacts. In Proceedings of the 2022 2nd International Conference on Information Systems and Data Engineering (pp. 1-6). ACM.

[10] Gupta, A., & Mittal, S. (2021). IPFS-based decentralized messaging application: A secure and private communication platform. In Proceedings of the 2021 10th International Conference on Computing, Communication and Automation (ICCCA) (pp. 1-6). IEEE.

[11] Aazam, M., & Imran, M. (2021). IPFS-based decentralized IoT architecture for secure data sharing. IEEE Access, 9, 103209-103221.

[12] Benet, J. (2014). IPFS: Content addressed, versioned, peer-to-peer file system.

[13] Gupta, A., Mittal, S., & Kumar, A. (2021). IPFS-based secure and private communication for IoT devices. In Proceedings of the 2021 10th International Conference on Computing, Communication and Automation (ICCCA) (pp. 1-6). IEEE.

[14] https://www.researchgate.net/publication/367203995_A_Comprehensive_Survey_on_Blockchain Based_Dec entralized_Storage_Networks

[15] Smith, J., & Johnson, M. (2020). Decentralized storage: A comprehensive overview. Retrieved from https://arxiv.org/abs/2001.05434

[16] P2P File System." arXiv preprint arXiv:1407.3561 (2014). ethers.js: Official documentation: https://docs.ethers.io/