

DECENTRALIZED GOOGLE DRIVE

Submitted in partial fulfilment of the requirements for the award of

Bachelor of Engineering degree in
Computer Science and Engineering

By

KASARM HARSHAVARDHAN (Reg. No. 40110565)

KANTHALA SACHIN (Reg. No. 40110541)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY-1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC |12B Status by UGC |Approved by AICTE

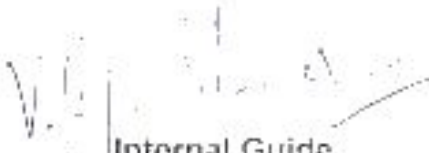
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KASARAM HARSHAVARDHAN** (Reg.No.40110565) and **KANTHALA SACHIN** (Reg.No.40110541) who carried out the project entitled "DECENTRALIZED GOOGLE DRIVE" under our supervision from November 2023 to April 2024.


Internal Guide

Ms. V. GOWRI MAHESWAR


HEAD OF DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Dr. L. SATHYANARAYANA
Head of Department
Dr. L. SATHYANARAYANA
Head of Department

Submitted for Viva voce Examination held on 13 / 4 / 2024


Internal Examiner


External Examiner

DECLARATION

I, KANTHALA SACHIN (Reg. No. 40110541) here by declare that the Project Report entitled "DECENTRALIZED GOOGLE DRIVE" done by us under the guidance of MS. V. GOWRI MANOHARI, M.E, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/4/2024
PLACE: CHENNAI


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **The Board of Management of Sathyabama Institute of Science and Technology** for their encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D.**, Dean, School of Computing and **Dr. L. Lakshmanan, M.E, Ph.D**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. V. Gowri Manohari, M.E**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Decentralized cloud storage is a Peer-to-peer network where each node provides the storage service to the customer's data. The storage system is based on the blockchain domain where it is completely decentralized. Blockchain ensures the user security and reliability of the user data. Peer-to-peer networking enables the user to store the data in different nodes across the network with security where the data are encrypted. This paper proposes a system which enables smart contract in the cloud storage system where it acts as an agreement between the client and the storage provider. Smart contract lets the user know about what data will be stored and the cost of storage. The decentralized cloud storage is reliable, secure and not power failure prone compared to the earlier systems. This system is integrated with a smart contract which enables to keep an agreement and the transaction will happen accordingly to the implemented smart contract.

Decentralized storage solutions using blockchain technology are gaining traction due to their potential to address data security, privacy, and accessibility concerns. This paper presents the concept and implementation of a decentralized Google Drive-like system utilizing blockchain technology. The proposed system leverages the decentralized and immutable nature of blockchain to create a secure and reliable storage platform for users.

In this system, files are encrypted and fragmented before being distributed across a network of nodes connected through a blockchain protocol. Each user has control over their encryption keys, ensuring data privacy and security. Smart contracts govern access control and incentivize participants in the network to contribute resources such as storage space and bandwidth.

Through this decentralized approach, the system aims to eliminate single points of failure, censorship, and unauthorized access commonly associated with centralized cloud storage providers. Users can access their files securely from any device without relying on a trusted third party. Additionally, the use of blockchain technology provides transparency and auditability, enhancing trust among users.

TABEL OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
3	REQUIREMENT ANALYSIS	8
	3.1 Functional Requirements	8
	3.2 Non-Functional Requirements	9
	3.3 Legal and Compliance Considerations	9
	3.4 Implementation Constraints	10
	3.5 Risks and Mitigation	10
4	DESCRIPTION OF PROPOSED SYSTEM	11
	4.1 Module Description	11
	4.1.1 Web3j	11
	4.1.2 Solidity	11
	4.1.3 React JS	11
	4.1.4 Truffle Suite	12
	4.1.5 Ganache	12
	4.1.6 Node.JS	12
	4.1.7 MetaMask	13
	4.1.8 IPFS Infura	13
	4.1.9 Ethereum	13
	4.1.10 Java script	13
	4.1.11 CSS	15
	4.1.12 HTML	16
	4.2 Proposed System	17
5	IMPLEMENTATION DETAILS	19
	5.1 Software Description	19
	5.1.1 Building Peer to Peer Network	19
	5.2 Node Setup	20

	5.2.1 Node Operations	20
	5.2.2 Motivation for Node	19
	5.3 Client Setup	20
	5.3.1 Communication with Node	20
	5.3.2 AES encryption and Decryption	21
	5.3.3 File Splitting	21
	5.3.4 Hash Encryption	22
	5.4 Blockchain Integration	23
	5.4.2 Smart Contract Development	26
	5.4.3 File Details record	27
	5.4.4 Token Transfer	28
	5.5 System Development Methodology	29
	5.5.1 Software Development Approach	29
	5.5.2 Requirement Analysis	30
6	RESULT AND DISCUSSION	31
	6.1 Client Application	31
	6.1.1 Result of Node App	31
7	SUMMARY AND CONCLUSION	36
	7.1 Conclusion	36
	7.2 Future Work	37
	REFERENCES	38
	APPENDIX	39
	(A) SOURCE CODE	39
	(B) SCREEN SHOTS	45
	(C) RESEARCH PAPER	47
	(D) CERTIFICATE	56

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	Architecture Diagram	17
5.1	P2P Network	19
5.2	Node Workflow	20
5.3	File Processing	21
5.4	Secured File Storage Protocol	22
5.5	Merkle Tree	25
5.6	Smart Contract Flow Diagram	27
5.7	Token Transfer	29
5.8	Scrum Technology	30
6.1	MetaMask	32
6.2	MetaMask Payment	33
6.3	File Uploading	34
6.4	File Uploading Retrieving Screenshot	35

LIST OF ABBREVIATIONS

ABBREVIATIONS	DEFINITIONS
API	Application Programming Interface
CCPA	California Consumer Privacy Act
CSS	Cascading Style Sheet
Dapps	Decentralized Applications
Defi	Decentralized finance
DOM	Document Object Model
ECDSA	Elliptic Curve Digital Signature Algorithm
GDPR	General Data Protection Regulation
HTTP	Hyper Text Transfer Protocol
HTML	Hyper Text Markup Language
IPFS	Inter Planetary File System
NFT	Non fungible Token
P2P	Peer to Peer

CHAPTER 1

INTRODUCTION

In the past decade, technological advancements have been made by research consortiums to adapt data sharing approaches. In such a way, research-based activities can improve through collaboration with the growing fields of Information Technology, Internet of Things and Digitization of every business, organizational work and projects, Information has become the biggest valuable asset for anyone. With the abundance of data and its ever-growing nature, it's equally important to store it in an organized way such that it's easily accessible and secure. For this purpose. Databases are used as a warehouse to store data. Database play a crucial role for any individual as well as any organization and business to store its data. Realizing the importance of data and insufficiency of storage, databases are replicated, distributed and backed up in different ways. Individuals store data in the cloud provided by different privately companies. Organizations set up their data centers at different part of the globe to store its data. For the security and bandwidth, data are scattered and replicated to different servers at different places. This seems to provide good solution for the management of rapidly increasing data. And also ensures data safety. In future, the rate of increment of data is sure to reach high. To cope with it, the current database system needs to be more reliable, safe and available all the time.

Cloud servers store the excessive amount of data, which is a centralized authority. There is various type of risks associated with a central authority, such as single point failure. To avoid such failure, third parties are involved to provide data backups. To eliminate third party for developing a trust-based model, a blockchain is introduced to provide trust and transparency. Decentralized storage is a solution, which allows storage of data independently on multiple nodes of the network in the form of a distributed ledger. Peer to Peer network is the distributed network where each node in the network communicates with each other directly or through a series of channels via other nodes. There is s no client server to access the resource. Each node will act both as a host or a client as needed. There is no any Central server for controlling the system flow and other nodes. Node is one of the member of P2P system which is willing to provide store for the clients in return to tokens. Node is fully responsible for

handling the client's data. It can view the encrypted data send by client which is stored in node's storage. Node can list all the data of the client and organize it. In our system, node is getting paid for storing the client's data by token of our system. According to the agreement between node and client, the node gets payment as file is downloaded by the client or duration of agreement finishes. Introduction of smart contracts is used here as an agreement between the client and the service provider where the user knows what data is stored and what is the price for the stored data. Smart contracts are the set of protocols in which the decentralized follows and the added advantage is it discloses any confusion between the users and the provider. Smart contracts are implemented in the truffle suite in this project.

IPFS is a BitTorrent like network where number of nodes are connected on the network so that the data can be uploaded and downloaded easily. IPFS is used here to distribute the data blocks across the network where several nodes can store the data which is encrypted. The address of this network is available in the IPFS so that the transactions of this network can be watched using wallets like ganache which is used in this project. Ganache is a private wallet where the user can watch the transactions and information about the block is also available. The frontend of this project is created using React which uses JavaScript programming language. This project proposes a decentralized cloud storage using IPFS system where the data stored on this network is more secure and reliable than the conventional storage system. It is a simple and secure file storage system where the file is distributed across the network and when the file needs to be retrieved, decryption is performed on the data to restore in its original form.

First, the data is selected and it is fed in the system where the data is first sharded into several number of blocks of bits. The number of shards is determined by the total number of nodes available in the network. The encryption is the most crucial point of this process where the data needs to be secure before distributing it across the network. After the data is separated into several shards which is considered as a data block is hash encrypted so that the receiver of this block doesn't understand the information. This process happens until the entire data is hashed encrypted. Introduction of smart contracts is used here as an agreement between the client and

the service provider where the user knows what data is stored and what is the price for the stored data. Smart contracts are the set of protocols in which the decentralized follows and the added advantage is it discloses any confusion between the users and the provider. Smart contracts are implemented in the truffle suite in this project. IPFS is a BitTorrent like network where number of nodes are connected on the network so that the data can be uploaded and downloaded easily. IPFS is used here to distribute the data blocks across the network where several nodes can store the data which is encrypted.

The address of this network is available in the IPFS so that the transactions of this network can be watched using wallets like ganache which is used in this project. Ganache is a private wallet where the user can watch the transactions and information about the block is also available. The frontend of this project is created using React which uses JavaScript programming language. The problem is the storage and processing limitation of network nodes. For this purpose, interplanetary file system (IPFS) is adapted, which is a p2p architecture. There is no risk of single point failure. It's similar to web3, but with different features. It performs content addressing and works in a similar way as bit torrent. Availability of data is ensured by storing it on a decentralized platform; IPFS.

CHAPTER 2

LITERATURE SURVEY

Existing System: Earlier, due to the non-usage of smart contracts in the previous proposed systems have issues with transparency of transactions. The cloud storage system doesn't ensure the user about the cost of the storage of data and what type of data will be stored. Some of the earlier systems didn't feature peer-to-peer networking which seems as a possible disadvantage. Currently, user uses his/her offline storage devices and other secondary storages for data backup and protection. Most of us often use the cloud service of Amazon, Google, Dropbox, Microsoft and others. A huge amount of users data are stored in cloud which is in fact someone's computer or storage devices. Such organization has complete authority over users data. In recent years, trend has increased rapidly in using those data without users acknowledge and permission by those company for their uses and pursue higher benefits from it. Most of the papers doesn't purpose a privacy policy where some information the transaction destination which can be improved with further developing the system. Some papers propose encryption algorithms aren't secure as encryption holds an important process in the cloud storage system where the data can't be interrupted by hackers. Advanced encryption algorithms are being developed which might provide better security to the users.

Proposed System: This paper proposes a decentralized cloud storage using IPFS system where the data stored on this network is more secure and reliable than the conventional storage system. It is a simple and secure file storage system where the file is distributed across the network and when the file needs to be retrieved, decryption is performed on the data to restore in its original form. First, the data is selected and it is fed in the system where the data is first sharded into several number of blocks of bits. The number of shards is determined by the total number of nodes available in the network. The encryption is the most crucial point of this process where the data needs to be secure before distributing it across the network. After the data is separated into several shards which is considered as a data block is hash encrypted so that the receiver of this block doesn't understand the information. This process happens until the entire data is hashed encrypted. Introduction of smart contracts is used here as an agreement between the client and the service provider where the

user knows what data is stored and what is the price for the stored data. Smart contracts are the set of protocols in which the decentralized follows and the added advantage is it discloses any confusion between the users and the provider. Smart contracts are implemented in the truffle suite in this project. IPFS is a BitTorrent like network where number of nodes are connected on the network so that the data can be uploaded and downloaded easily. IPFS is used here to distribute the data blocks across the network where several nodes can store the data which is encrypted.

The address of this network is available in the IPFS so that the transactions of this network can be watched using wallets like ganache which is used in this project. Ganache is a private wallet where the user can watch the transactions and information about the block is also available. The frontend of this project is created using React which uses JavaScript programming language.

Title : Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS

Author : Jin Sun

Year : 2014

Jin Sun proposed an access scheme for electronic medical records in IPFS with blockchain-Based Secure Storage. The author constructed attribute-based encryption scheme for secure storage and efficient sharing of electronic medical records in IPFS environment. The ideology that the author suggests is that we encrypt the data based on attributes and determine the attributes of the users. Each user's private key is related to their respected attributes whereas the ciphertext is related to the policy.

Title : Blockchain-based Decentralized Storage Scheme

Author : Yan Zhu

Year : 2019

Yan Zhu suggests a decentralized storage scheme instead of a conventional centralized system. The provider performs a data integrity certificate to the user and only after it is verified, the user pays the fee for storage. The payment information is stored in the blockchain which is secure.

Title : A Peer-to-Peer Cloud Storage Network

Author : Shawn Wilkinson

Year : 2016

Shawn Wilkinson used a peer-to-peer cloud storage which implements end- to-end encryption which would allow to share and transfer data without a need for third party data providers. The author implements a challenge algorithm which would cryptographically check the integrity and availability of the user data.

Title : Simple Decentralized Storage

Author : David Vorick

Year : 2014

David Varick introduced a simple decentralized storage system which enables the formation of contracts between peers. Contracts are like agreements between users and the provider which defines what data will be stored and at what price. Title : A peer-to-peer electronic cash system

Title: Peer to Peer Networking

Author : Satoshi Nakamoto

Year : 2008

Satoshi Nakamoto introduced peer-to-peer networking which changed the entire domain. The author suggests a user to send payments directly from one party to another without a need for financial institution. This paper provides a solution to double-spending problem with the introduction of peer-to-peer networks.

Title : IPFS - Content Addressed, Versioned, P2P File System

Author : Juan Bernet

Year : 2014

Juan Bernet introduced Inter Planetary File System which is a distributed peer-to-peer file system that connects the computing devices. The author designed the system in ways that it provides a high through-put content-addressed block storage model.

Title : Zero cash: Decentralized Anonymous Payments from Bitcoin.

Author : Eli Ben-Sasson

Year : 2014

Eli Ben-Sasson explains in detail about decentralized anonymous payments where the author points out some of the flaws in the bitcoin privacy guarantees. This paper fulfills that flaw by unlinking transaction from the payment's origin and it also reveals the payment destinations and the amount.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The decentralized Google Drive project seeks to redefine file storage and sharing by leveraging blockchain technology and peer-to-peer networking. Traditional centralized storage systems have limitations in terms of privacy, security, and reliability. By decentralizing storage and utilizing distributed ledger technology, this project aims to empower users with greater control over their data.

User Registration and Authentication: Users will be able to securely register accounts using authentication mechanisms such as password-based or biometric authentication. This ensures that only authorized users can access the platform and their files.

File Upload and Management: The platform will support the uploading and management of various file types and sizes. Users can organize files using folders or tags, allowing for efficient file navigation and organization.

Decentralized Storage: Files will be stored using decentralized storage protocols like IPFS (Inter Planetary File System) or Swarm. This approach ensures data redundancy and availability by distributing file chunks across a network of nodes.

Peer-to-Peer Sharing: Users can securely share files with others through peer-to-peer connections. Sharing permissions (e.g., read-only, edit) can be configured to control access to shared files.

Data Encryption and Security: All files will be encrypted before upload to protect data privacy. Access control mechanisms will be implemented to prevent unauthorized access to files.

Search and Retrieval: The platform will provide robust search and retrieval functionalities, allowing users to find files based on metadata or content efficiently.

Versioning and Backup: Versioning capabilities will track changes made to files over time, enabling users to revert to previous versions if needed. Backup mechanisms will ensure data resilience and prevent data loss.

3.2 NON-FUNCTIONAL REQUIREMENTS

Performance: The system must be responsive and scalable to handle a large volume of users and files. File retrieval and sharing processes should remain efficient under high load conditions.

Security: Strong encryption standards will be employed to protect user data from unauthorized access. Regular security audits and updates will be conducted to ensure ongoing protection against threats.

Reliability: The platform will ensure high availability and data durability by implementing redundancy measures. File replication and distribution across nodes will minimize the risk of data loss.

Usability: The user interface will be designed to be intuitive and user-friendly, catering to diverse user needs and preferences. Accessibility features will be considered to enhance usability for all users.

Compatibility: The platform will support multiple devices and operating systems, enabling users to access their files seamlessly from various devices. APIs or SDKs may be provided for third-party integration.

3.3 LEGAL AND COMPLIANCE CONSIDERATIONS

Data Privacy: Compliance with data protection regulations such as GDPR (General Data Protection Regulation) or CCPA (California Consumer Privacy Act) will be ensured to protect user privacy and rights.

Intellectual Property: The platform will respect intellectual property rights of users and third parties, ensuring that unauthorized distribution or use of copyrighted material is prevented.

3.4 IMPLEMENTATION CONSTRAINTS

Technology Stack: Careful selection of blockchain platforms, decentralized storage protocols, and frontend/backend technologies will be made to ensure compatibility, scalability, and performance.

Resource Limitations: Consideration of hardware and network constraints is essential for implementing peer-to-peer networking and decentralized storage effectively.

3.4 ASSUMPTIONS AND DEPENDENCIES

Internet Connectivity: Users are assumed to have stable internet connections to access the platform and upload/download files.

Blockchain Scalability: Challenges related to blockchain scalability and transaction costs will be addressed through optimization strategies and protocol improvements.

3.5 RISKS AND MITIGATION

Blockchain Reliability: Regular monitoring of blockchain network performance and reliability will be conducted to mitigate risks associated with blockchain technology.

Regulatory Changes: Continuous monitoring of regulatory landscapes will enable the platform to adapt to evolving legal requirements and ensure compliance.

CHAPTER 4

DESCRIPTION OF PROPOSED SYATEM

4.1 MODULE DESCRIPTION

4.1.1 *Web3j*

Web3j is a lightweight, highly modular, reactive, type safe Java and Android library for working with Smart Contracts and integrating with clients (nodes) on the Ethereum network. This allows you to work with the Ethereum blockchain, without the additional overhead of having to write your own integration code for the platform. It has complete implementation of JSON-RPC client (Application Programming Interface) API over Hyper Text Transfer Protocol (HTTP) and Inter Process Communication (IPC). It has android compatible version and supports Infuriated has complete support to Ethereum wallet.

4.1.2 *Solidity*

Solidity is a contract-oriented programming language for writing smart contracts. Solidity is a statically-typed programming language designed for developing smart contracts that run on the EVM. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM) with Solidity, developers can write applications that implement self- enforcing business logic embodied in smart contracts, leaving a no repudiable and authoritative record of transactions. Solidity support inheritance, including multiple inheritance with C3 linearization.

4.1.3 *ReactJS*

It is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. Complex React applications usually require the use of additional libraries for state management,

routing, and interaction with an API. It makes developer painless to create interactive UI. Properties commonly called props are passed from parent component to child. It also has feature of stateful components which can be passed to child components. React creates an in-memory data structure for virtual Document Object Model (DOM) and updates the browser DOM efficiently. Life cycle in ReactJS are hooks which allows execution of code at set of points during component's lifetime.

4.1.4 Truffle Suite

Truffle is the most popular development framework for Ethereum with a mission to make your life a whole lot easier. Truffle takes care of managing your contract artifacts so you don't have to. Includes support for custom deployments, library linking and complex Ethereum applications. A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier. Tighten the feedback loop between deployment, operation, and debugging; all within powerful development sandboxes. Share insights with the whole team.

4.1.5 Ganache

Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

The main features of ganache are

Blockchain log output - See the log output of Ganache's internal blockchain, including responses and other vital debugging information.

Advanced mining control - Configure advanced mining with a single click, setting block times to best suit your development needs

Ethereum blockchain - Byzantium comes standard, giving you the latest Ethereum features needed for modern dapp development. Examine blocks and transactions to gain insight about what's happening under the hood.

4.1.6 Node.js

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs

on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

4.1.7 MetaMask

MetaMask is an extension for accessing Ethereum enabled distributed applications, or "Dapps" in your browser! The extension injects the Ethereum web3API into every website's JavaScript context, so that dapps can read from the blockchain. MetaMask is a cryptocurrency wallet used to interact with the Ethereum blockchain. MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser.

4.1.8 IPFS Infura

Infura provides secure, reliable, scalable, and easy to use APIs to access the Ethereum network and the IPFS. In many cases, using this API this is preferable to embedding IPFS directly in your program it allows you to maintain peer connections that are longer lived than your app and you can keep a single IPFS node running instead of several if your app can be launched multiple times.

4.1.9 Ethereum

Ethereum, and its provision of smart contracts provided real functionality even if the results of its open system are dubious. Ethereum is a decentralized, open- source blockchain featuring smart contract functionality. It is the native cryptocurrency of the platform. It is the second-largest cryptocurrency by market capitalization, after Bitcoin. Ethereum is the most actively used blockchain.

4.1.10 JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

Most websites use it for client-side page behavior and all major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of other runtime systems, such as Node.js and Deno. These systems are used to build servers and are also integrated into frameworks, such as Electron and Cordova, for creating a variety of applications. Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

The main features of JavaScript are:

- Imperative and structured
- Weakly typed
- Dynamic
- Object-orientation (prototype-based)
- Functional
- Delegative

- Miscellaneous
- Vendor-specific extension

4.1.11 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file which reduces complexity and repetition in the structural content as well as enabling the css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. Internet media type (MIME type) is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL. There is no single, integrated CSS4 specification, because the specification has been split into many separate modules which level independently.

Modules that build on things from CSS Level 2 started at Level 3. Some of them have already reached Level 4 or are already approaching Level 5. Other modules that

define entirely new functionality, such as Flexbox, have been designated as Level 1 and some of them are approaching Level 2. The CSS Working Group sometimes publishes "Snapshots", a collection of whole modules and parts of other drafts that are considered stable enough to be implemented by browser developers. So far, five such "best current practices" documents have been published as Notes, in 2007, 2010, 2015, 2017, and 2018. Since these specification snapshots are primarily intended for developers, there has been growing demand for a similar versioned reference document targeted at authors, which would present the state of interoperable implementations as meanwhile documented by sites like Can I Use...and the Mozilla Developer Network. A W3C Community Group has been established in early 2020 in order to discuss and define such a resource. The actual kind of versioning is also up to debate, which means that the document once produced might not be called "CSS4".

4.1.12 HTML

The Hyper Text Markup Language, or HTML(Hyper Text Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages

Information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML.

An HTML Application (HTA; file extension ".html") is a Microsoft Windows application that uses HTML and Dynamic HTML in a browser to provide the application's graphical interface. A regular HTML file is confined to the security model of the web browser's security, communicating only to web servers and manipulating

only web page objects and site cookies. An HTA runs as a fully trusted application and therefore has more privileges, like creation/editing/removal of files and Windows Registry entries. Because they operate outside the browser's security model, HTAs cannot be executed via HTTP, but must be downloaded (just like an EXE file) and executed

4.2 PROPOSED SYSTEM

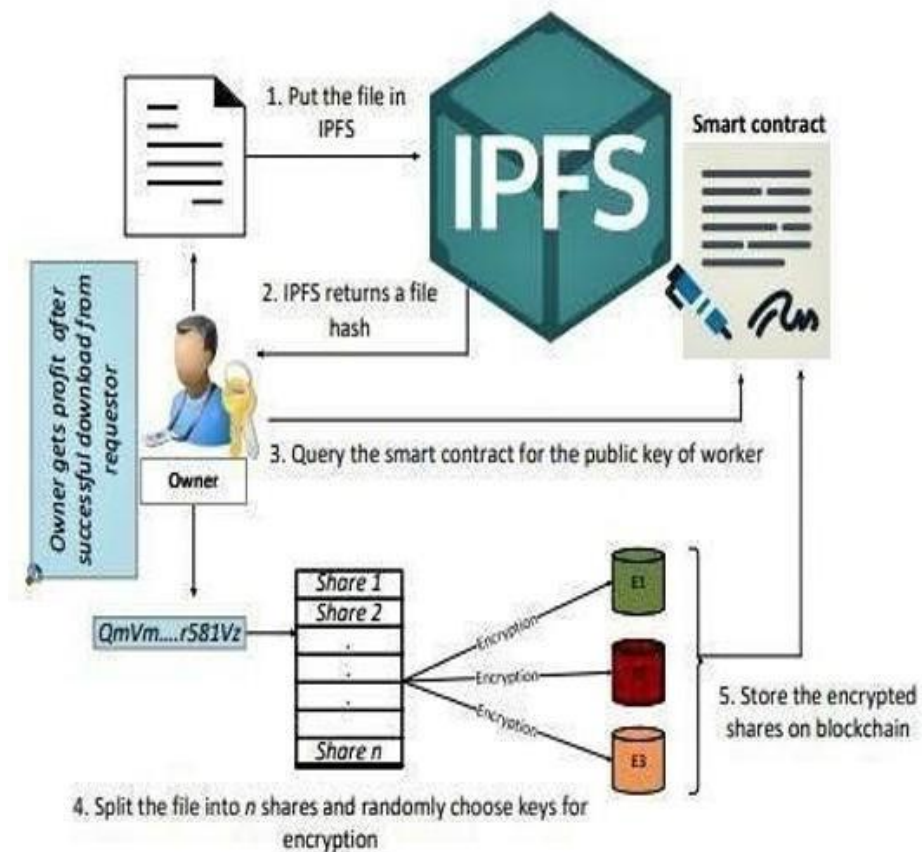


Fig: 4.1: Architecture Diagram

This project proposes a decentralized cloud storage using IPFS system where the data stored on this network is more secure and reliable than the conventional storage system. It is a simple and secure file storage system where the file is distributed across the network and when the file needs to be retrieved, decryption is performed on the data to restore in its original form. First, the data is selected and it is fed in the system where the data is first sharded into several number of blocks of bits.

The number of shards is determined by the total number of nodes available in the

network. The encryption is the most crucial point of this process where the data needs to be secure before distributing it across the network. After the data is separated into several shards which is considered as a data block is hash encrypted so that the receiver of this block doesn't understand the information. This process happens until the entire data is hashed encrypted. Introduction of smart contracts is used here as an agreement between the client and the service provider where the user knows what data is stored and what is the price for the stored data. Smart contracts are the set of protocols in which the decentralized follows and the added advantage is it discloses any confusion between the users and the provider. Smart contracts are implemented in the truffle suite in this project.

IPFS is a BitTorrent like network where number of nodes are connected on the network so that the data can be uploaded and downloaded easily. IPFS is used here to distribute the data blocks across the network where several nodes can store the data which is encrypted. The address of this network is available in the IPFS so that the transactions of this network can be watched using wallets like ganache which is used in this project. Ganache is a private wallet where the user can watch the transactions and information about the block is also available. The frontend of this project is created using React which uses JavaScript programming language.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 SOFTWARE DESCRIPTION

5.1.1 BUILDING PEER TO PEER NETWORK

Peer to Peer network is the distributed network where each node in the network communicates with each other directly or through a series of channels via other nodes. There is no client server to access the resource. Each node will act both as a host or a client as needed. There is no any Central server for controlling the system flow and other nodes.

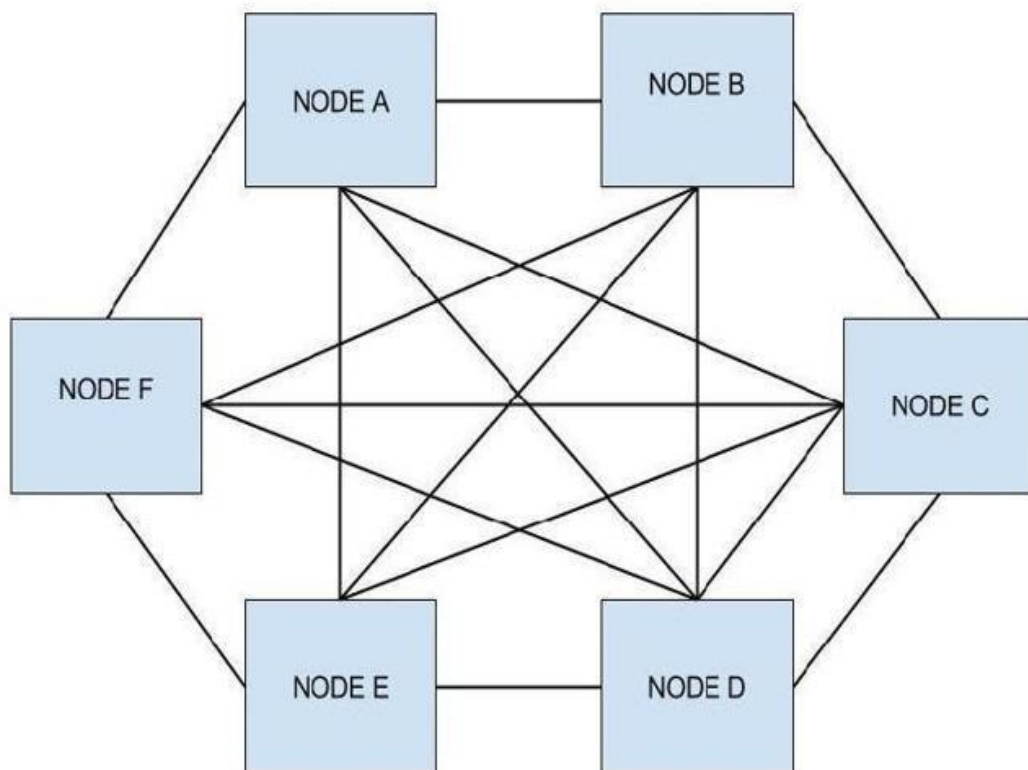


Fig: 5.1: P2P Network

5.2 NODE SETUP

Node is one of the member of P2P system which is willing to provide store for the

clients in return to tokens.

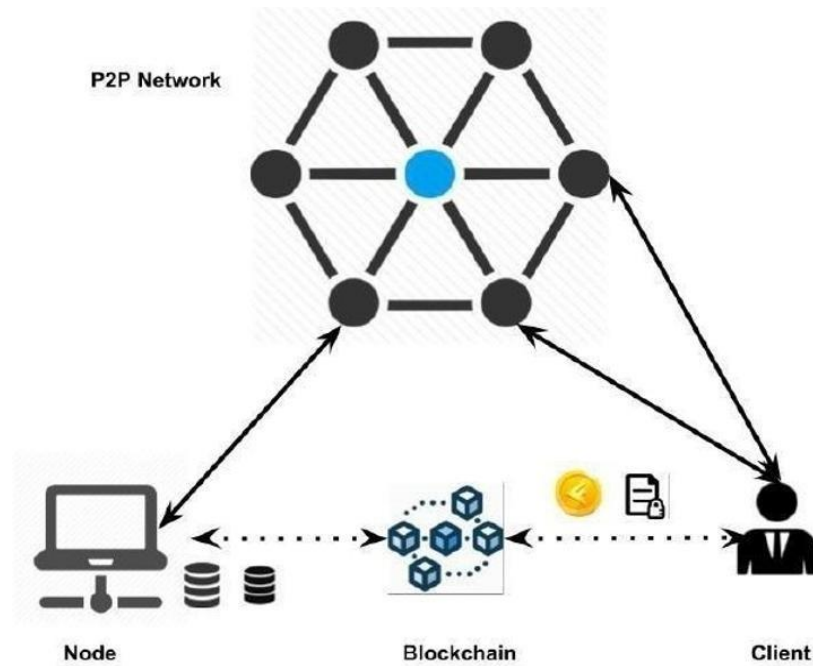


Fig. 5.2: Node Workflow

5.2.1 Node Operations

Node is fully responsible for handling the client's data. It can view the encrypted data send by client which is stored in node's storage. Node can list all the data of the client and organize it.

5.2.2 Motivation for Node

In our system, node is getting paid for storing the client's data by token of our system. According to the agreement between node and client, the node gets payment as file is downloaded by the client or duration of agreement finishes.

5.3 CLIENT SETUP

5.3.1 Communication with Node

File Processing is the main function of our app is to encrypt/decrypt the file, split it into user defined parts and upload it to nodes. Like that, download it from nodes, merge the chunk to single file and then decrypt to original file. Figure given below will give more clear idea about file processing in our system.

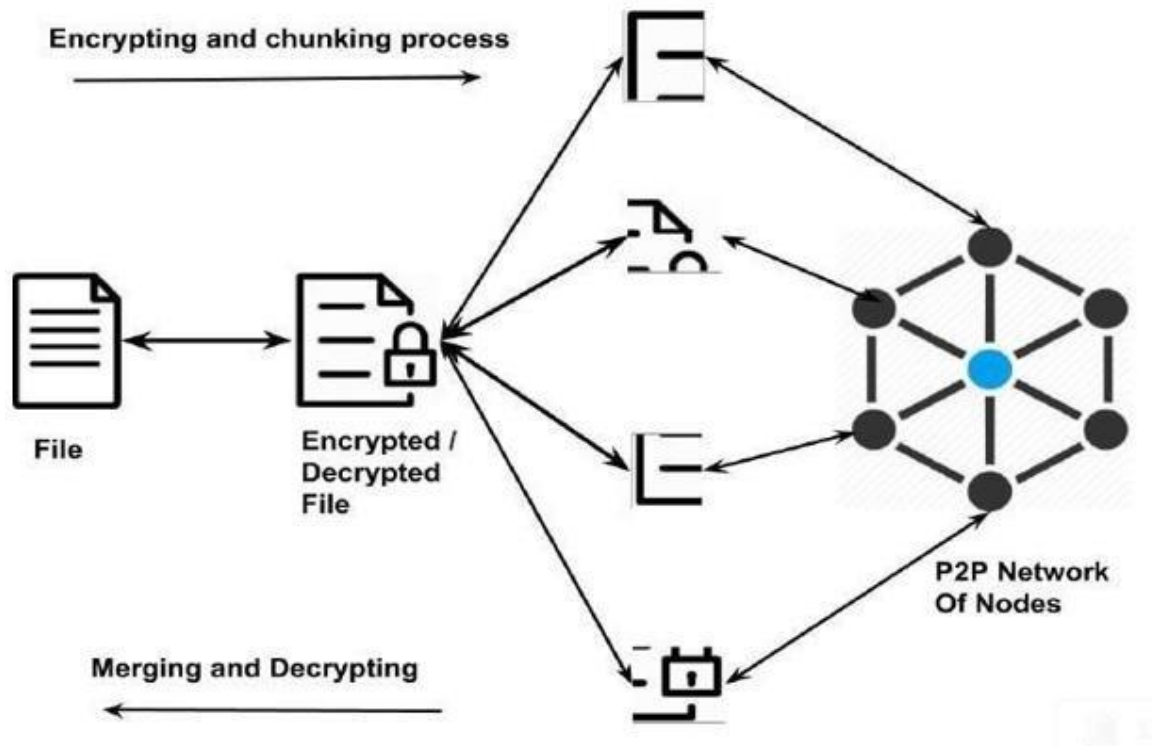


Fig. 5.3: File Processing

5.3.2 AES Encryption and Decryption

Using Random AES Key Generator, random key for AES gets generated. Like that, Random IV spec generator generates Random IV which is required during AES encryption. 24 byte AES key along with 16 byte Random IV encrypts the file. AES key is further encrypted with master key. Therefore, our encrypted file consist first 40 byte of key as header and remaining byte as encrypted data of the file. Finally, the encrypted file gets chunked into user defined number and gets transferred to network. Like that after downloading the chunks from network, all the parts get merged into encrypted file. First 40 bytes is separated as header and remaining as file cipher. Among 40 bytes, 24 bytes is separated and AES key is derived. After that remaining 16 bytes Random IV and AES key is used to decrypt the file which gives original file as an output.

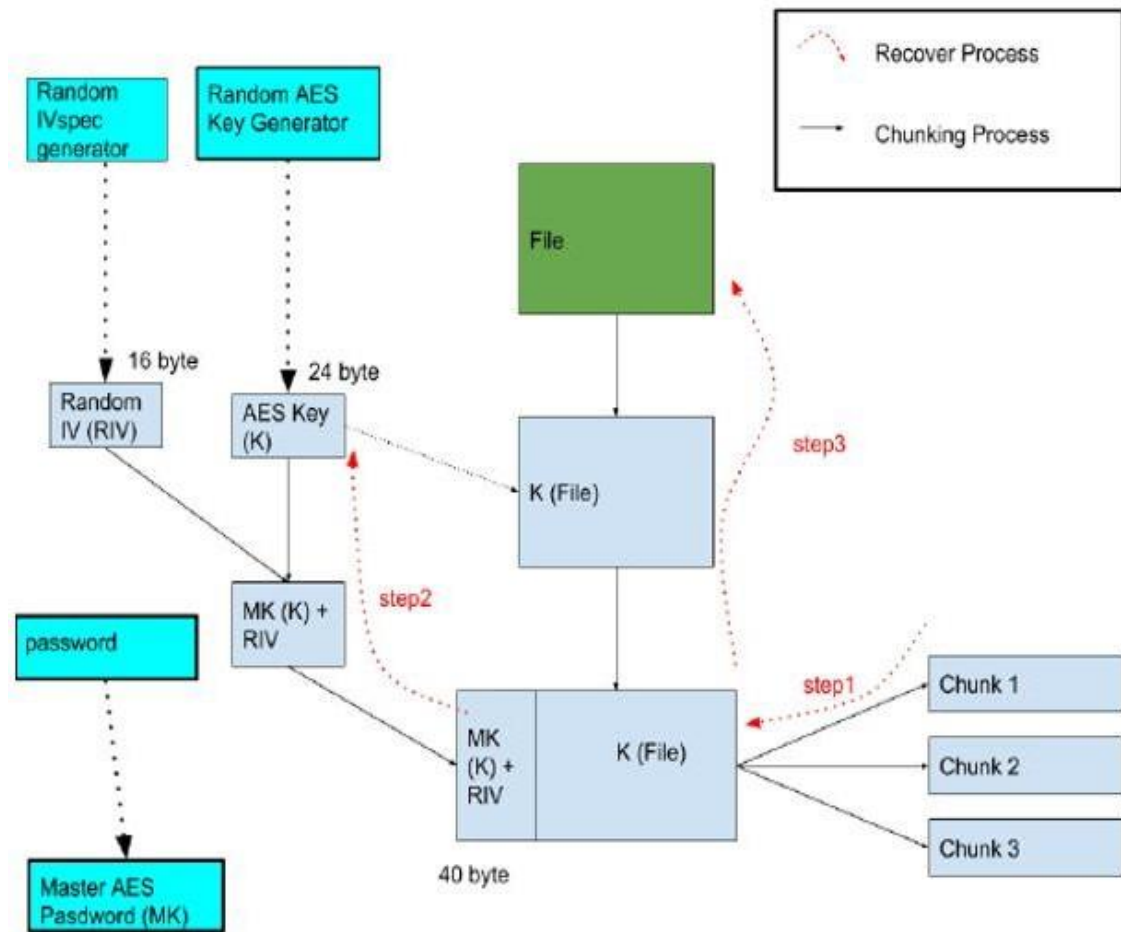


Fig. 5.4: Secured File Storage Protocol

5.3.3 File Splitting/Merging

The encrypted file is spitted into number of nodes selected and transferred to each node via network. Like that, during download those separate chunks from different nodes get merged into single cipher file.

5.3.4 Hash Encryption

Encryption is a two-way function; what is encrypted can be decrypted with the proper key. Hashing, however, is a one-way function that scrambles plain text to produce a unique message digest. With a properly designed algorithm, there is no way to reverse the hashing process to reveal the original password. An attacker who steals a file of

hashed passwords must then guess the password. Here's how it works: A user enters a password and an ID in a browser and sends it (preferably over a secure link) to the authentication server. The server uses the ID to look up the associated message digest.

The password submitted by the user is then hashed with the same algorithm, and if the resulting message digest matches the one stored on the server, it is authenticated. In this process the server does not store or need to see plain-text passwords. Stealing hashed files does the attacker little good because the attacker cannot reverse the hashing process.

But people rarely use completely random passwords there is a trick that can be used to help guess the passwords in the file. An attacker can run a collection of a million or so commonly used passwords through a hashing algorithm and get a list called rainbow table of associated message digests for these passwords. It is child's play for a computer to compare a file of stolen password hashes against a rainbow table. For every match, the table will show the password for that hash.

The protection against this is to salt the hash: Add a random number to each password before it is hashed. The resulting message digest is the product of both the password and the salt value and will not match anything on the rainbow table. Of course, the attacker can always try adding random values to common passwords to find a matching hash, but now the difficulty of guessing the password makes it impractical. The return on investment of such a process is so low that a stolen file of properly hashed and salted passwords is essentially worthless.

5.4 BLOCKCHAIN INTEGRATION

5.4.1 Blockchain

Blockchain is the growing list of records called blocks, containing structured information linked using the art of cryptography distributed globally . Each block in blockchain contains the cryptographic hash of previous block along with timestamp and data which typically varies with use-cases. Merkle trees are the fundamental part of blockchain. Every block contains the block header which is outcome of recursive cryptographic hashes of all the data nodes or transaction from bottom to up approach.

During hash generation, the order of data matters for the final hash of the block. If single detail of transactions or order of transaction changes then changes the Merkle hash.

Merkle Root summarizes all of the data in the related transactions, and is stored in the block header. This feature makes blockchain resistance to modification which is considered secure by design. A database is a collection of information that is stored electronically on a computer system. Information, or data, in databases is typically structured in table format to allow for easier searching and filtering for specific information. Large databases achieve this by housing data on servers that are made of powerful computers.

These servers can sometimes be built using hundreds or thousands of computers in order to have the computational power and storage capacity necessary for many users to access the database simultaneously. While a spreadsheet or database may be accessible to any number of people, it is often owned by a business and managed by an appointed individual that has complete control over how it works and the data within it. Blockchain is P2P network which relies on protocol for inter- node communication and validating the blocks. Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency bitcoin. Blocks in blockchain are the holder of valid transactions that are hashed and encoded in Merkle tree.

Every block contains the cryptographic hash of previous blocks along with its own data which therefore forms the chain. This iterative mechanism in each block conforms the integrity of previous block all the way back to genesis block. Block time is the average time it takes in the network to generate 1 extra block in blockchain. By the time block is generated, the data of that block is verified. This means lesser the block time, fare the transactions. A hard fork is a rule change such that the software validating according to the old rules will see the blocks produced according to the new rules as invalid.

Storing data in P2P network allows Blockchain to eliminate the pitfalls of centralization. There will be no central point vulnerability, no center point of failure in blockchain. It is open to public which makes it more user- friendly than traditionally owned records. Being permissionless and open, there is no need to guard against bad

actors. Ethereum is an open-source, public, blockchain-based distributed computing platform and operating system featuring smart contract(scripting) functionality.

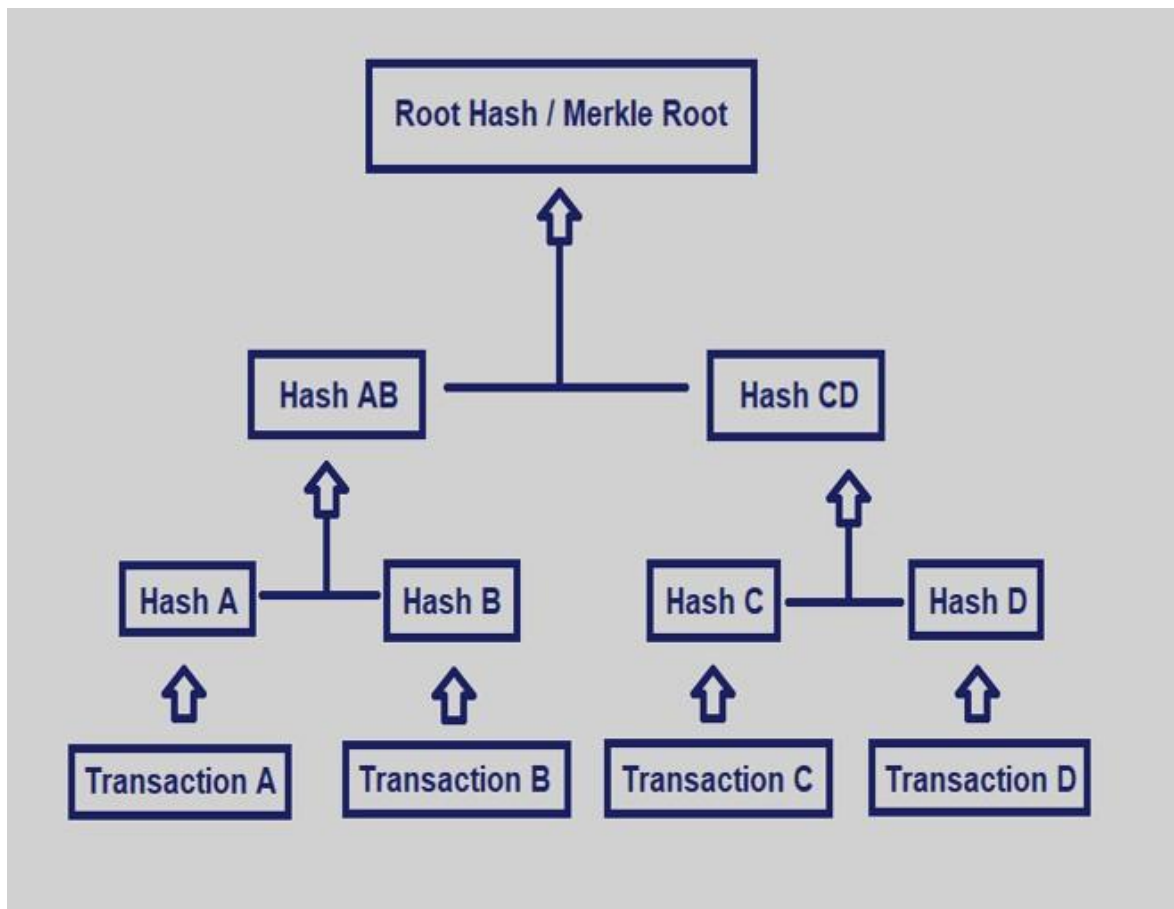


Fig. 5.5: Merkle Tree

Ether is a cryptocurrency whose blockchain is generated by the Ethereum platform. Ethereum provides a decentralized Turing-complete virtual machine, the Ethereum Virtual Machine (EVM), which can execute scripts using an international network of public nodes. "Gas", an internal transaction pricing mechanism, is used to mitigate spam and allocate resources on the network. Ethereum address are composed of the prefix "0X" a common identifier for hexadecimal, concatenated with the rightmost 20 bytes of the Keccak-256 (SHA-3) hash (big endian) of the ECDSA (Elliptic Curve Digital Signature Algorithm) public key.

Smart Contract Ethereum's smart contracts are based on different computer languages, which developers use to program their own functionalities. Smart contracts are high level programming abstractions that are compiled down to EVM bytecode

and deployed to the Ethereum blockchain for execution. They can be written in Solidity (a language library with similarities to C and JavaScript), Serpent (similar to Python, but deprecated), LLL (a low-level Lisp-like language), and Mutant (Go-based, but deprecated). There is also a research-oriented language under development called Viper (a strongly-typed Python-derived decidable language). ERC20 Token ERC-20 is a technical standard used for smart contracts on the Ethereum blockchain for implementing tokens. ERC stands for Ethereum Request for Comment, and 20 is the number that was assigned to this request. The clear majority of tokens issued on the Ethereum blockchain are ERC-20 compliant. The ERC20 token standard describes the functions and events that an Ethereum token contract has to implement.

5.4.2 Smart Contract Development

Smart contract acts as an agreement between client and postman during the transaction of data. Therefore, we have tried to ensure (look) from both parties during its development. We have tried to keep as less data as possible in blockchain network without compromising services. The key to these contracts is the decentralized network known as blockchain. Smart contracts use blockchain technology to verify, validate, capture and enforce agreed-upon terms between multiple parties.

Smart contracts on the blockchain allow for transactions and agreements to be carried out among anonymous parties without the need for a central entity, external enforcement, or legal system. The transactions are transparent, irreversible, and traceable. Blockchain is the perfect environment for smart contracts, as all the data stored is immutable and secure. The data of a smart contract is encrypted and exist on a ledger, meaning that the information recorded in the blocks can never be lost, modified, or deleted. Smart Contract act as a global database which gets deployed in the blockchain. There are 2 roles in our contract Node : stores client data, earns tokens Client stores data on node, pays tokens.

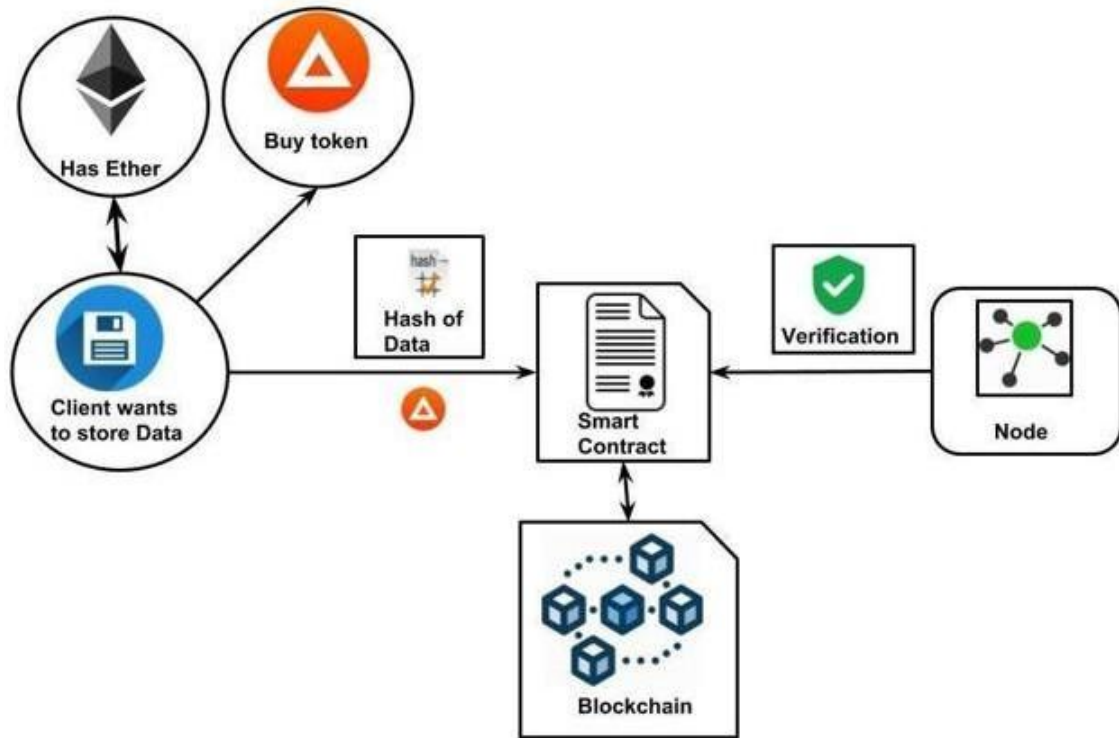


Fig. 5.6: Smart Contract Flow Diagram

User having ether in their Ethereum account can interact with the contract. Our system has its own ERC20 token which can be bought by ether. We have defined our own exchange rate. In order to store data in our system, client must pay agreed amount of tokens to the node as specified during node-client agreement in smart contract. Client gets facility to store data on node's storage. During this process, client and postman come on agreement based on storage size, bandwidth, time of storage and token amount. Once, agreement is done, all details of agreement between client, node and associated data chunk is written smart contract which gets deployed in Blockchain.

5.4.3 File Details Record

As client's chunked file gets stored in node's storage, there should be proper tracking mechanism of file along with its chunk, associated node and client in Smart Contract. Our system should ensure client for file retrieval. Smart contract tracks all the record associated with client, file and node. Usually data are stored as key-value data structure. Address with File Detail Array It tracks the owner of file along with array of owned file details by each individual. File Details contains the hash of file and it's

filename.

File Hash with Chunk Hash Array It tracks the chunk of file array associated with the main file so that when file hash is known, its chunk can be traced. **Chunk Hash with File hash** Every node doesn't have file hash where as smart contract needs file hash for getting chunk info. This data gap is fulfilled by this data structure where once chunk hash is known, file hash can be known immediately. **Chunk hash to index/File hash to index** Looping in smart contract is very expensive. Therefore, for every index of chunk hash in array is mapped with its chunk hash so that once chunk hash is known it's index can be immediately known. Like that, client can have number. **Chunk Index to Download index** All the chunk uploaded to node will not be downloaded. Therefore, this data structure tracks the chunk data which have download request so that payment agreement can be done.

5.4.4 Token Transfer

During our contract deployment, we created fixed amount of token which serves as payment mechanism in our system. Tokens can be exchanged with ether token.

For buying tokens, contract gives use option to buy it with ethers. Once tokens are purchased, users can use our system properly. We have made a system of locking the balance of 2 agreed parties so that both parties will perform their duty. Once the agreement is made between client and node, the specified amount of tokens from their balance is transferred to their lock balance. Here, lock balance means that those deposited tokens are restricted to use for other purposes. Those tokens get locked until agreement is not completed. When correct file is provided by node to client which gets verified by blockchain, then payment is provided to node from lock balance. Here, tokens of node is also transferred to lock balance so that if node doesn't behaves what is agreed in agreement then, client is compensated with those tokens. Once the agreement is done, fund is not returned back.

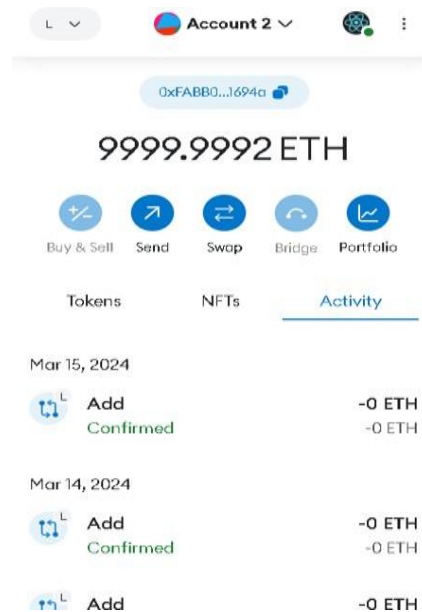


Fig. 5.7: Token transfer

5.5 SYSTEM DEVELOPMENT METHODOLOGY

5.5.1 Software Development Approach

Making huge and dynamic system using traditional approach such as waterfall model of software development cost more time and manpower. Therefore to meet the requirements of the system with more flexibility and timely delivery, we have chosen Scrum methodology under the Agile Development method. Instead of providing complete, detailed descriptions of how everything is to be done on a project, much of it was left up to the project development team.

The scrum team is self-organizing in that there is no overall team leader who decides which person will do which task or how a problem will be solved. So, each time every team member tried to solve problems and find solutions and in Scrum, a team is cross functional, meaning everyone is needed to take a feature from idea to implementation. Project progress were shown via a series of sprints. In keeping with an agile methodology, sprints are timeboxed to no more than a month long, most commonly two weeks.

We met at every start of sprint and figure out each individual commits of task and created backlog to keep track of work. We commonly had daily meetings during college team for discussing project's problems and solutions. This helped all team member to be fully synchronized which is one of the main benefit of Scrum methodology. We made sprint burndown chart and release burndown chart which helped us to know remaining work and track the overall project schedule. Hence scrum is adaptive, has small repeating cycles and there is short-term planning with constant feedback, inspection and adaption and is therefore chosen as the software development methodology.



Fig. 5.8: Scrum methodology

5.5.2 Requirement Analysis

The functional and non-functional requirements of this project are as listed below.

Function Requirement:

- The system shall encrypt, decrypt the data of user.
- The system shall split and merge the file.
- The system shall built P2P network and allow clients to connect.
- The system shall read and write data from Rinke by Test Net.

Non-Function Requirement:

- The system must ensure data retrieval of clients.
- The system must allow client to store all types of files.
- The system should be dynamic enough to easily adapt with increasing number of data.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 CLIENT APPLICATION

Based on the observations of the system, we came to the following observations and evaluations. The file storage service works well through all phases of the feature: Selection, Encryption, Splitting, Upload, Listing, Download, Merging, Decryption. Every record metadata is store in Smart Contract. The task is significantly smoother when a copy of all transaction is stored in a local database and reading from contract is done for verification check only during conflicting situation. This makes the user experience smooth.

The node can now receive a real time notification of every details. The node app is more user friendly and easily understandable about the existing real time system.

The file chunking-merging and encryption-decryption is highly dependent upon the two factors: File Size and Processor speed. The time of any of the file operation is linearly dependent upon the file size. So, uploading huge size file produces significant load on the Android Thread and in some case is forcefully killed by the system causing system crash. A webapp of the client side can be more impressive and easier to use for loading contract, wallet, handling heavy processing task is easier and faster.

6.1.1 RESULT OF NODE APP

The node interface communicates with the:

- The interface shows how much storage space the files have taken, number of files the node has received and how many files in total clients have downloaded.
- List of all the chunks received as file can be watched in this panel. The list shows hash, creation and expiry date along with the number of download count for individual files. Events like receiving files are shown in real time using web sockets.
- This interface shows information about a deployed node with its public key, node

id, private key. Other nodes connected or discovered by this node is shown with their node Identity, IP address and port.

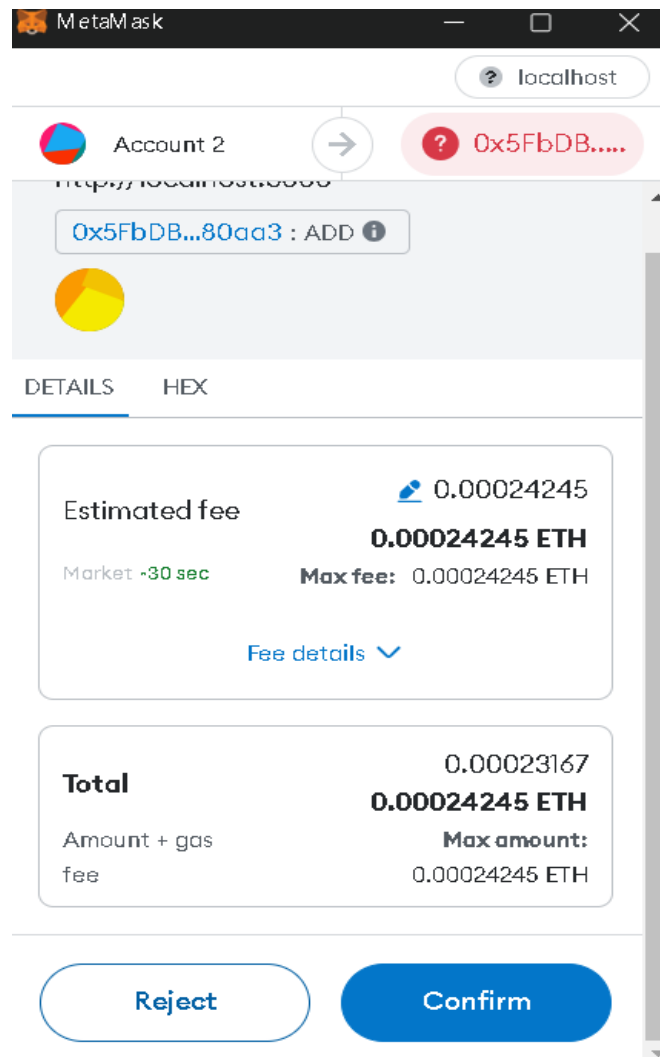


Fig. 6.1: MetaMask

MetaMask is a cryptocurrency wallet used to interact with the Ethereum blockchain. MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser.

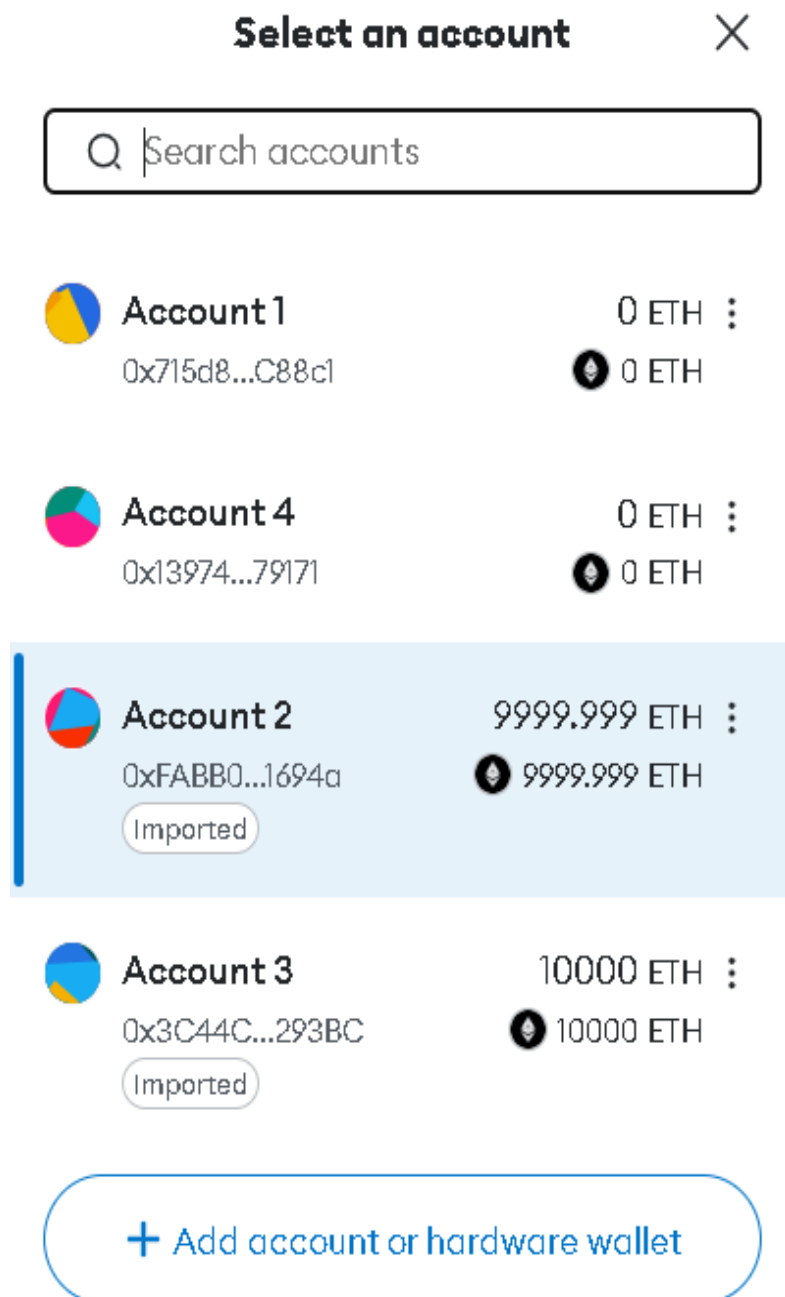


Fig. 6.2: MetaMask Payment

See the log output of Ganache's internal blockchain, including responses and other vital debugging information. Configure advanced mining with a single click, setting block times to best suit your development needs Byzantium comes standard, giving you the latest Ethereum features needed for modern dapp development. Examine all blocks and transactions to gain insight about what's happening under the hood.



Fig. 6.3: File uploading

With decentralized cloud storage, end-to-end encryption is standard on every file- each file is encrypted on a user's computer before it's uploaded, broken into pieces, and then spread out to uncorrelated Nodes across our network. Plus, centralized cloud storage costs a lot more than our decentralized network.

IPFS is used here to distribute the data blocks across the network where several nodes can store the data which is encrypted. is available in the IPFS so that the transactions of this network can be watched using wallets like ganache which is used in this project. Ganache is a private wallet where the user can watch the transactions and information about the block is also available. The frontend of this project is created using React which uses JavaScript programming language.



Fig. 6.4: File upload and retrieving screenshot

The contents of the retrieved block are decrypted and merged into the original file where the content is checked whether any loss of data is recorded. This type of cloud storage is secure and is cost effective compared to the centralized server. Each node in this network acts as a server and data is shared among them without knowing what the data contains and whose is it.

CHAPTER 7

SUMMARY AND CONCLUSION

7.1 CONCLUSION

The development and implementation of a decentralized Google Drive-like system using blockchain technology represent a significant step towards addressing the limitations of centralized cloud storage platforms. By leveraging blockchain's inherent features of decentralization, immutability, and transparency, the proposed system offers several key advantages for users and organizations seeking secure and privacy-focused data storage solutions.

Firstly, the decentralized nature of the system eliminates single points of failure and reduces the risk of data breaches or unauthorized access. Each user retains control over their encryption keys, ensuring that only authorized parties can access and decrypt their files. This enhances data privacy and security compared to traditional cloud storage models where data is stored and controlled by a central entity.

Secondly, the use of blockchain technology enables transparent and auditable access control mechanisms through smart contracts. Users can define fine-grained permissions for accessing their files, and these permissions are enforced automatically by the decentralized network. This not only enhances data security but also provides users with greater control over their data.

Furthermore, the decentralized storage network incentivizes participants to contribute resources such as storage space and bandwidth through token-based incentives governed by smart contracts. This creates a self-sustaining ecosystem where users are rewarded for contributing to the network's growth and security.

In conclusion, the decentralized Google Drive-like system presented in this project demonstrates the potential of blockchain technology to revolutionize data storage by putting users in control of their data and fostering trust through transparency and security. While there are challenges such as scalability and user adoption to overcome, the promising results of this project pave the way for further research and development in decentralized storage solutions that prioritize data sovereignty and integrity. Future

work will focus on refining the system, enhancing scalability, and exploring user-friendly applications to drive broader adoption of decentralized storage technologies.

7.2 FUTURE WORK

The future model can be improved with better encryption algorithms and this domain is still under development so better data distribution algorithms can be used. Blockchain-based cloud computing doesn't only decentralize data storage. More than ever, this technology is being used as part of data logistics platforms. Many authorized users are allowed access to the data at once and can interact, interpret, and change it as they see fit. This type of cloud computing solution allows employees to work together virtually and securely to analyze and maintain information. The virtualization of contractual agreements and other exchanges is also easier with the use of blockchain, leading to a wide increase in usage over many industries.

REFERENCES

1. Crosby.M, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," Applied Innovation, vol. 2, pp. 6–10, 2016.
2. David Vorick et al. Sia: Simple Decentralized Storage. 2014.
3. Eli Ben-Sasson et al. Zerocash: Decentralized Anonymous Payments from Bitcoin.2014.
4. Jin Sun et al. Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS.IEEE.2014.
5. Juan Bernet.IPFS - Content Addressed, Versioned, P2P File System.2014.
6. King.S - Ppcoin: Peer-to-peer crypto-currency with proof-of-stake.2014.
7. McConaghy.T, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen,R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," white paper, BigChainDB, 2016
8. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008.
9. Shawn Wilkinson et al. Storj: A Peer-to-Peer Cloud Storage Network. 2016.
10. Vitalik Buterin.*Ethereum*: A next-generation smart contract and decentralized application platform.2013
11. Wilkinson.S, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.
12. Yan Zhu et al. Blockchain-based Decentralized Storage Scheme.IOP publishing.1237.4.2019.

APPENDIX

(A) SOURCE CODE

Solidity:

1. Storage contract:

```
pragma solidity 0.4.24;
contract
SimpleStorage { string
ipfsHash;
function set(string x)
    public { ipfsHash =
        x;
    }

function get() public view returns
    (string) { return ipfsHash;
    }

}
```

2. Migration:

```
pragma solidity
^0.4.2; contract
Migrations { address
public owner;
uint public
last_completed_migration; modifier
restricted() {
    if (msg.sender == owner) _;

}

function Migrations()
    public { owner =
```

```

    msg.sender;
  }

  function setCompleted(uint completed) public restricted
  { last_completed_migration = completed;}

  function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}

```

Reactjs:

App.js:

```

import React, { Component } from 'react'

import SimpleStorageContract from
'../build/contracts/SimpleStorage.json' import getWeb3 from
'./utils/getWeb3'
import ipfs from
'./ipfs' import
'./css/oswald.css'
import './css/open-
sans.css' import
'./css/pure-min.css'
import './App.css'
class App extends
Component {
  constructor(props) {
    super(props)
    this.state = { ipfsHash: "", web3: null, buffer: null, account: null}

    this.captureFile = this.captureFile.bind(this);
    this.onSubmit = this.onSubmit.bind(this);
  }
}

```

```

componentWillMount() {

  // Get network provider and web3 instance.

  // See utils/getWeb3 for more info.
  getWeb3
  .then(results => { this.setState({ web3: results.web3
    })

    // Instantiate contract once web3 provided.
    this.instantiateContract()
  })

  .catch(() => {

    console.log('Error finding web3.')

  })
}

instantiateContract() {

  /*

  * SMART CONTRACT EXAMPLE

  *

  * Normally these functions would be called in the context of a
  * state management library, but for convenience I've placed them here.

  const contract = require('truffle-contract')

  const simpleStorage =
  contract(SimpleStorageContract)
  simpleStorage.setProvider(this.state.web3.currentProvide
  r)
  // Get accounts.

```

```

this.state.web3.eth.getAccounts((error, accounts)
=> {
simpleStorage.deployed().then((instance) => {
this.simpleStorageInstance = instance
this.setState({ account: accounts[0] })
// Get the value from the contract to prove it worked.
return
this.simpleStorageInstance.get.call(accounts[0])
  }).then((ipfsHash) => {

// Update state with the result.
return this.setState({ ipfsHash })
  })
})

}

captureFile(event) {
  event.preventDefault()
  t()
  const file = event.target.files[0]

  const reader = new
  window.FileReader()
  reader.readAsArrayBuffer(file)
  reader.onloadend = () => {
    this.setState({ buffer: Buffer(reader.result) })
    console.log('buffer', this.state.buffer)
  }

  onSubmit(event) {
    event.preventDefault()
    t()
    ipfs.files.add(this.state.buffer, (error, result) => {
      if(error) {

```

```

        console.error(error)
      }
    }
    return result
  }
  this.simpleStorageInstance.set(result[0].hash, { from: this.state.account
  }).then((r) => {

    return this.setState({ ipfsHash: result[0].hash })
    console.log('ipfsHash', this.state.ipfsHash)
  })

})

}

render() { return
(
  <div className="App">

    <nav className="navbar pure-menu pure-menu-horizontal">

      <a href="#" className="pure-menu-heading pure-menu-link">IPFS
File Upload DApp</a>

    </nav>

    <main className="container">

      <div className="pure-g">

        <div className="pure-u-1-1">

          <h1>Your Image</h1>

          <p>This image is stored on IPFS & The Ethereum Blockchain!</p><img
src={`https://ipfs.io/ipfs/${this.state.ipfsHash}`} alt=""/>

          <h2>Upload Image</h2>

          <form onSubmit={this.onSubmit} >

```

```
        <input type='file' onChange={this.captureFile} />

    <input type='submit' />

    </form>

  </div>

</div>

</main>

</div>);
}

}

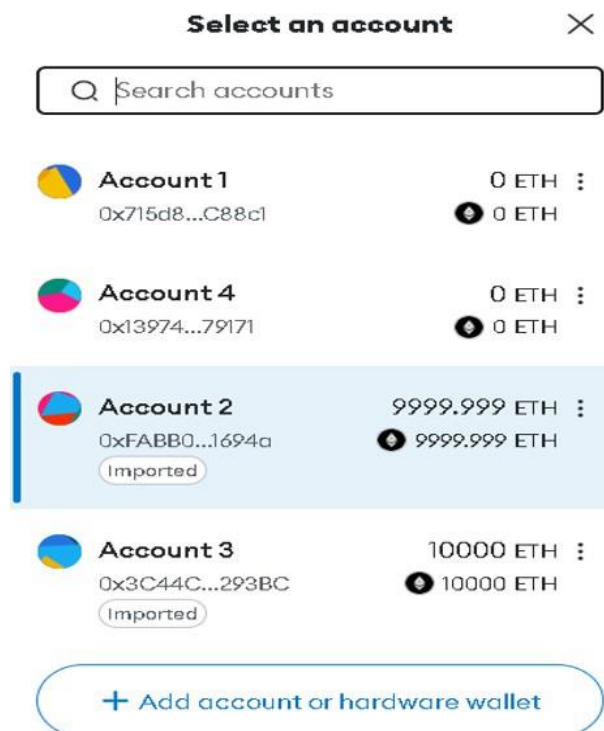
export default App
```

(B) SCREENSHOTS

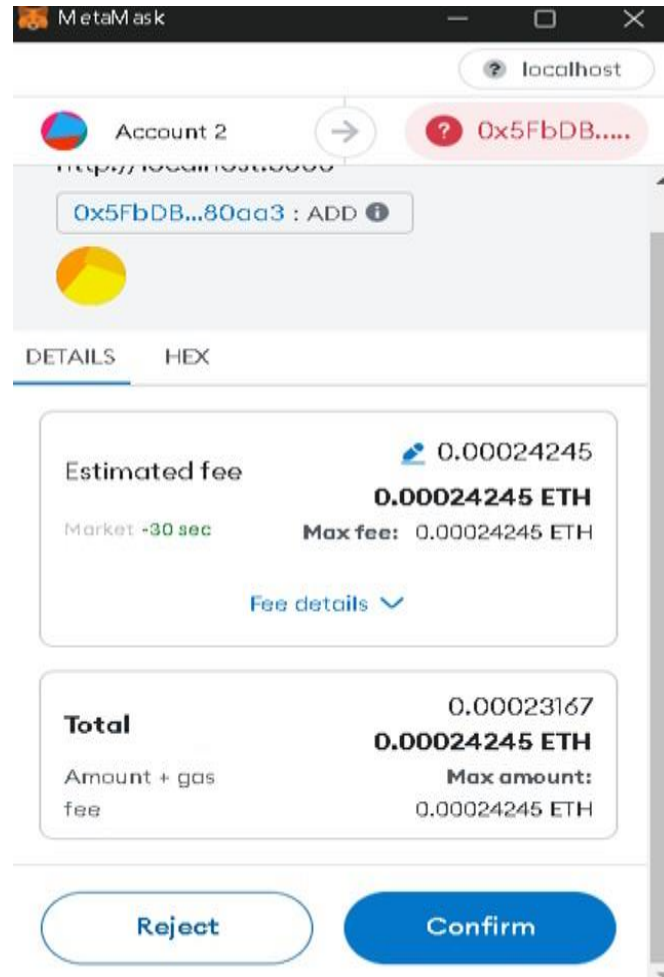
1. File Upload



2. Metamask



3. File Retrieving



(C) RESEARCH PAPER

Decentralized Google Drive

K. HARSHAVARDHAN

Department of CSE,

Sathyabama institute of science

and technology,

Chennai, India,

guravareddy8500@gmail.com

K. SACHIN

Department of CSE,

Sathyabama institute of science

and technology,

Chennai, India,

sachinreddykanthala@gmail.com

GOWRI MANOHARI V

Department of CSE,

Sathyabama institute of science,

and technology,

Chennai, India,

gowrimanohari.cse@sathyabama.ac.in

Dr. Nancy Noella R S

Department of CSE,

Sathyabama institute of science

and technology,

Chennai, India,

nancynoella.cse@sathyabama.ac.in

K. Karunya

Department of CSE,

Sathyabama institute of science

and technology,

Chennai, India,

karunsai.ifs@gmail.com

Abstract: Decentralized cloud garage is a peer-to-peer network where each node provides storage for customer records. Based on pc garage where the block chain area is completely decentralized. Block chain supplies User Security Data User Loyalty. It lets in peer-to-peer networking User to gather records on specific community nodes with safety Data is encrypted. This article proposes a gadget that permits use in a pointy settlement a cloud garage gadget, which acts as an agreement among the patron and the client Storage Company. A smart contract permits the user to recognise what

facts will be sent Storage fees are also saved. Cloud garage is reliable and secure. Not challenge to capacity disasters as compared to previous structures. This is the cause by combining pain with agreement you can hold the settlement and the transaction will take place in step with the contract done by means of the customer.

Keywords: Decentralized, Google Drive, cloud, Block Chain, Customers

INTRODUCTON

In the closing decade, this has been done due to technological advancement Research companies to adapt their approach to facts communication. Thus, the have a look at Performance may be enhanced through the use of developing sectors such as labor organization, Internet of Things and Digitization of Everything, and Information Technology. The details turned out to be really valuable. Given the multitude of stuff and the always expanding nature of factors, keeping everything organized so that it's readily available is equally important. and defense. Because of this rely. Data bases used to store data. He donated his company to the Treasury, and a database is a provider to any person or organization. Recognize the value and scarcity of information. Databases and storage are sponsored up, distributed, and duplicated in diverse ways. Large amounts of human information had been made to be had inside the cloud via various non-public agencies. Companies have their facts centers in one of a kind elements of the organization to keep their information. For protection and performance functions, records is dispersed and replicated on different servers. At unique places. It appears to provide a great control answer Fast developing facts. Even on at ease data. Future fees Data boom will truely be high. To triumph over this, the database gadget works be confident, safe and prepared whenever. Cloud servers shop an unlimited quantity of statistics, it's miles a government. There are differing types\ Risk associated with important authority which includes failure at some point. Don't try this Failing that, 0.33 parties are rehired. The third element have to be removed By growing a trust-primarily based version, a module is introduced to build agree with Transparency Decentralized Storage is a solution that permits you to acquire facts Independent f several network nodes within a distributed ledger. Peer-to-peer networks are those in which each community node can

communicate with every other node directly or via a few routes via other nodes. Customer service is not required to gain access to Knots assistance. Every node functions as a client or client. No manipulable media server Queue structure and other nodes. A node is one of the P2P contributors Organization ready to provide clients with garage for standards. Complete knot

The patron is chargeable for processing the records. Can ship encrypted statistics patron, which is saved at the garage node. You have get entry to to all customer facts and It ought to be organized. In our device, stored customers' node statistics is output from the signal. Our gadget. In settlement between the node and the patron, the node accepts Payment while the purchaser downloads the report or while the agreement expires. Implementation of clever contracts is described right here as a settlement between client and client A carrier provider lets the consumer know what's stored there and at what fee Save the facts. A smart agreement is a set of protocols Then comes decentralization, and it has the added gain of getting rid of all confusion. Among the users and the issuer. A smart agreement implemented in Trumpet Luxury in this document. IPFS is a torrent-like community in which the wide variety of nodes connected to the network in order that information may be effortlessly uploaded and downloaded. IPFS is used here to distribute facts blocks across more than one networks Lymph nodes shop information in an encoded shape. The cope with of this community Available in IPFS to take a look at transactions using this community Parsas changed into used as ganache in this assignment. Ganache is a private pockets\ It can take a look at the person's sports and quick data is likewise to be had. The front give up of this undertaking is built using React which makes use of JavaScript. Programming language. This plan gives decentralized cloud storage this community facts garage is extra secure

and reliable than the IPFS device Normal storage machine. This document device is straightforward and cozy the document is shipped over the community, and while the report needs to be repaired; Decryption is carried out to restore the records to its authentic shape. First, records Number of bit blocks. The variety of pieces is determined by means of the variety Available in lymphatic networks. Encryption could be very critical on this the technique through which facts is included before it's far disbursed over a community. After the facts is divided into numerous parts, it's far considered as a information block. No encryption due to the fact the receiver does not recognize this block this procedure keeps until all data is encrypted. Introduction to Consumer Contracts is used here for conference.

Between the client and the carrier issuer, the user knows what the facts is Cost of saved and stored facts. You are a pain Protocols that follow decentralization and this is an delivered gain It solves all of the confusion between the users and the provider. Pain subsides In this undertaking implemented in set of tubers. IPFS is a BitTorrent-like network. Data may be the quantity of nodes related to the community Easy add and download. IPFS is used here to distribute records. In a community where a couple of nodes can store encrypted information. The cope with of this network is to be had in IPFS and as a result the transactions of this network You can use plastic baggage like the ganache used on this assignment. Ganache is a non-public pockets in which person can track transactions there is likewise site visitor's statistics. The mission turned into evolved before this It uses React, which uses the JavaScript programming language. It is considered; Imposing storage and processing of the lymphatic network. But to this end It embraces Interplanetary File System (IPFS), a p2p structure. Devour There isn't any threat of failure in a single location. It is just like

the Internet, but with distinctive features. This is addressing content and works like BitTorrent. Available provided that the statistics is saved on a decentralized platform; I.P.F.S.

OBJECTIVE

To create a gadget primarily based on the Ethereum block chain that could distribute user facts in a decentralized allotted data community among friends. The unique goals are as follows:

- Cryptography, P2P networks, Internet blockchain and technology.
- Establish a distributed cloud storage infrastructure; • Support active decentralized research applications and cryptography.

AREA OF PROJECT SCOPE

In this venture we advocate a decentralized database gadget using decentralized networks and block chains. The objective of the program is to illustrate reliability. A decentralized database device the usage of block chain and smart contracts.

RELATED WORK

Title: A project to keep and get right of entry to electronic gadgets primarily based on block chain Medical information in IPFS

Author: Jin Sun

Year: 2014

Jin Sol proposed a device for accessing electronic scientific statistics in IPFS with blockchain-based totally cozy storage. The creator advanced an encryption attribute to set up a scheme for cozy garage and green exchange of electronic scientific information in an IPFS surroundings. The concept proposed via the writer is to decide the attributes and

characteristics of the user to encode the statistics. Everything A consumer's non-public key's associated with respectable attributes, however a cipher is related to a policy.

Title: Blockchain Storage Based Decentralized System

Author: Yang Zhu

Year: 2019

Yang Zhu proposes a decentralized garage desk to update the traditional centralized gadget. The provider leaves the user with a easy statistics certificates, and only if the person is demonstrated, the garage price is charged. Payment facts is stored on a relaxed block chain.

Title: Peer Cloud Storage Network

Author: Sean Wilkinson

Year: 2016

Sean Wilkinson used peer-to-peer cloud storage, which enables stop-to-stop encryption, permitting records to be shared and exchanged without the need for third-birthday party statistics companies. The author implements a verification set of rules that cryptographically verifies the integrity and availability of user information.

Title: Simple Decentralized At

Author: David

Year: 2014

David Worick proposed a simple decentralized garage system that lets in for peer-to-peer contracts. Contracts are much like agreements between customers and providers that determine what statistics is stored and at what fee.

Title: Peer-to-Peer Electronic Money System.

Author: Satoshi Nakamoto.

Year: 2008

The peer-to-peer network was introduced by Satoshi Nakamoto, who also changed the entire domain. The operator allows the user to transfer money instantly from one birthday party to another without having to get in touch with a banking institution. This article provides a response. Overview of peer networks Double-dipping problems.

Title: IPFS - Content Addressing, Versioning, P2P File System

Author: John Burnett

Year: 2014

Juan Burnett proposed the Interplanetary File System, a peer-to-peer disbursed file system for computing gadgets. The author designed the machine to offer a high-overall performance block garage version.

Title: Zerocash: Anonymous Money Transfer from Bitcoins.

Author: Eli Ben-Sasson

Year: 2014

Eli Ben-Sasson explains decentralized anonymous bills in an editorial wherein the writer points out a number of the flaws in Bitcoin's privacy guarantees. This article addresses this shortcoming by means of separating the transaction from the transaction supply opens price guidelines and amount.

EXISTING SYSTEM

Previously, due to non-use of smart contracts, ex the proposed systems have problems with transaction transparency. Cloud storage the machine does no longer assure facts garage and typing costs to the

consumer Data might be saved. Some in advance systems did no longer help peer interactions. Network, this looks like a capability disadvantage. A user is presently the usage of those Offline storage devices and different secondary storage for information backup and safety. Mostly they use Amazon, Google, Dropbox, and Microsoft and cloud offerings other. A large quantity of consumer records is saved in the cloud, which is essentially foreign laptop or garage devices. Such a gadget has complete manipulate over customers. Information the fashion of the use of this data without consumer enter has multiplied unexpectedly in latest years. These evaluations and companies are allowed to apply them and compete with others Benefits thru this. Most documents don't have any privateness coverage Information approximately the destination of the transaction can be updated with further development we create a device. Some articles advise that encryption algorithms are insecure. Because encryption is a essential system in a cloud storage device Hackers cannot smash into your information. Improved encryption algorithms at the moment are being developed. It is designed to provide higher protection to the users.

PROPOSED SYSTEM

This paper proposes a decentralized cloud garage the usage of IPFS. This network information garage gadget is extra comfortable and dependable than traditional storage device. It is a simple and secure report storage device wherein the document is distributed over the community, and whilst the report cannot be recovered, the encrypted information must be restored to its unique shape. First, records is selected and fed into the pc, wherein the statistics is first damaged into a couple of blocks. The wide variety of slices is decided by using the wide variety of nodes in the network. Encryption is a completely important part of this technique

wherein facts should be included before it is distributed across the community. After the statistics is divided into numerous portions, which can be treated as a block of facts, they're encrypted so that the recipient of the block cannot understand it.

This manner keeps until all facts is encrypted. Implementation of smart contracts is defined right here as an settlement between the consumer and the service provider in which the person is aware of what facts is saved and what is the cost of the stored records. A clever contract is a hard and fast of protocols that follows a decentralized mechanism and has the added advantage of getting rid of any confusion.

Between the customers and the provider. In this challenge clever contracts are achieved on a set of meals items. An IPFS network is just like BitTorrent, in which more than one nodes are related to the community so that facts may be effortlessly sent and retrieved. IPFS is used here to distribute information blocks over a network wherein a couple of nodes can store encrypted facts. The deal with of this community Available in IPFS, so the enterprise of this network can be monitored the usage of ganache used in this task. Ganache is a personal wallet wherein the person can song transactions and block statistics. Accessible the front-quit of this undertaking is built the use of React, which uses the JavaScript programming language.

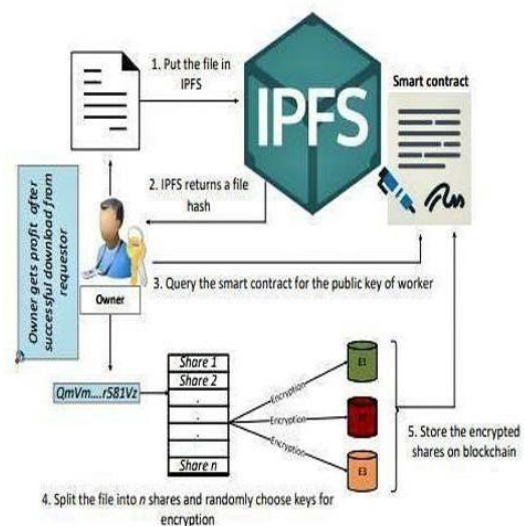
ARCHITECTURE DIAGRAM

This business gives decentralized cloud storage using IPFS system, data saved on this network is greater cozy and dependable than traditional storage machine. It is a easy and at ease report storage device wherein the record is shipped over the community and when the report wishes to be repaired, the encrypted records should be restored to its original

form. First, the records is selected and fed into the pc, where the facts is first divided into several blocks.

The variety of bits is decided by way of the quantity of nodes to be had within the network. Encryption is a completely important a part of this method where records ought to be included before it's miles allotted over the network. After the records is divided into several components, they are treated as facts blocks, which might be encrypted in order that the receiver of the block can't recognize the statistics.

This procedure maintains until all data is encrypted. The advent of clever contracts is used here as an settlement between the consumer and the provider issuer, in which the person knows what information is stored and what's the price for facts storage. A clever agreement is a hard and fast of protocols that follows a decentralized protocol and has the added advantage of casting off any confusion among customers and providers. In this venture, a clever agreement is finished on a set of tubers. An IPFS network is just like BitTorrent, in which more than one nodes are related to the network in order that data can be easily despatched and retrieved. IPFS is used here to distribute data blocks over a network where multiple nodes can store encrypted records. The cope with of this community is to be had in IPFS, so the transactions of this community can be visible in wallets like Ganache used in this venture. Ganache is a personal wallet where person can tune transactions and get facts. The front cease of this assignment is constructed using React, which uses the JavaScript programming language.



MODULES

Web3j

Web3j is a Java and Android toolkit that works with smart contracts and incorporates clients, or nodes, into the Ethereum network. It is lightweight, somewhat modular, reactive, and kind-secure. This allows you to work with the Ethereum blockchain without having to worry about having to write platform integration code yourself. It features a fully functional purchaser implementation for JSON-RPC (Application Programming).

Interface) API over Inter Process Communication (IPC) and Hypertext Transfer Protocol (HTTP). It supports Infura and has a corresponding Android version. Ethereum has a comprehensive pocket handbook.

Solidity

It's a powerful programming language for creating intelligent The preferred programming language for creating EVM-based smart contracts is called Solidity. It was created for the Ethereum Virtual Machine and is powered by Python, C++, and JavaScript (EVM). With truth, programmers can create apps that integrate enterprise logic that helps

itself into smart contracts, leaving behind a trustworthy and impenetrable record of actions. Several inheritances with C3 linearization and strong inheritance support are included.

React JS

It's a JavaScript library designed to create user interfaces. It is entirely dependent on Facebook and a network of independent contractors and agencies. React can be utilized as a framework for developing mobile applications or single webpage websites. Generally speaking, complex React projects need to use extra libraries for state control, routing, and API interactions. This allows builders to create an interactive interface painlessly. Properties, usually called props, are exceeded from figure to toddler elements. He also has a stamp of public affairs that is sent to children's parties. React presents an in-reminiscence information structure for a digital record object.

Model (dom) and browser dom beneficial replace. Lifecycles are hooks in React JS that allow code to run at set factors in a factor's existence.

Truffle Chut

Potato is a very famous Ethereum improvement surroundings that objectives to make your life plenty less difficult. Truffles takes care of the contract handling so you ought not to. Includes custom records, inking libraries, and help for complex Ethereum applications. A global-elegance development surroundings, check environment and business pipeline for deploying block chains

Ethereum Virtual Machine (EVM) to make life less difficult for developers. Strengthening the remarks loop among deployment, operations and debugging; these are all very useful development sandboxes. Share ideas with the whole institution.

Ganache

Quickly install a non-public Ethereum blockchain that you may use to run assessments, execute instructions, and take a look at popularity whilst monitoring chain pastime. Key Features of Ganache:

Blockchain document ebook. See Ganache's inner module release log, including responses and other essential debugging facts.

Financial mining energy. Set up superior mining with one click and set the time according to your mining wishes.

□ Ethereum Blockchain – Byzantium is solid and affords you with the modern Ethereum capabilities you need to build the state-of-the-art decentralized packages.

Barrier-free creation. Read all of the blocks and transactions to get an concept of what is going on interior.

Js Node

Node.js is a server-side, move-platform, open-source JavaScript engine that runs JavaScript scripts outside of web browsers. Node.js is a server-side, move-platform, open-source, JavaScript runtime that utilizes the V8 engine to run JavaScript code outside of a web browser. Node.js enables you to jot down command-line equipment the use of JavaScript and create server-aspect scripts, server-side scripts that run to create dynamic web page content earlier than sending it to the person's browser. Thus, Node.Js proposes a "JavaScript everywhere" approach that unifies net application development around one programming language instead of separate languages for the server and patron aspects. written portion

Metamask

MetaMask is an extension for getting access to Ethereum-enabled dispensed packages or "Dapps" to your browser! An extension to the Ethereum web3

API wraps the html context of every web page so that decentralized programs can study information from the blockchain. MetaMask is a digital currency wallet that communicates with Ethereum. MetaMask grants clients to shop and oversee account keys, broadcast exchanges, send and get hold of Ethereum-essentially based digital currencies and tokens, and safely connect with decentralized programs through an appropriate program or inherent cell program. IPFS Infura

Infura provides secure, reliable, scalable and clean-to-use APIs for getting access to the Ethereum network and IPFS. In many instances, it is most excellent to use this API to without delay generate an ipfs program. This allows you to aid peer connections longer than your utility, and if your application runs multiple instances, you may save an ipfs node for multiple runs. Once

Ethereum

Ethereum and its clever contracts have furnished real functionality, even if the outcomes of its open gadget are questionable Ethereum is an open source, decentralized blockchain with astute arrangement capacity. It is the best stage for digital currency. It is the second most noteworthy cryptographic cash through market capitalization after Bitcoin. Ethereum is the liveliest blockchain.

JavaScript

JavaScript, oftentimes condensed as JS, is a programming language that follows the ECMAScript detail. JavaScript is multi-paradigm, high-level, and mostly compile-time. It has convolutional language structure, dynamic composing, thing orientated model based and fine usefulness. Alongside HTML and CSS, JavaScript is one

Fundamental advances of the Internet. JavaScript empowers intelligence on site pages and is a

necessary piece of net bundles. Most internet browsers use it to control page lead at the client side, and all dominating net programs have JavaScript motors to deal with it. JavaScript supports event-driven, functional, and vital programming styles because it is a multi-paradigm language. This has its advantages

Programming points of interaction (APIs) for working with literary substance, dates, ordinary articulations, favoured markup frameworks, and the Record Article Model (DOM). A widespread ECMA script does now not have any enter/output (I/O) competencies consisting of networking, garage, or pics. In reality, it offers a web browser or different implementation device

JavaScript API for I/O. JavaScript engines have been at In the beginning, they were only used in web browsers, but they are now standard components of various runtime systems like Node. Deno and Js. These frameworks are utilized to assemble servers and are incorporated into stages like Electron and Cordoba to develop different projects. Despite the fact that there are likenesses among JavaScript and Java, including the language's name, punctuation, and related popular library, the 2 dialects are particular and feature exclusive strategies.

Key Features of JavaScript:

Compulsory and based

Weakly typed.

Will exchange.

Object orientation (in step with prototypes)

He wishes.

Representative

Others

Vendor useful extensions

CSS

Flowing Templates (CSS) is a style sheet used to make sense of the introduction of a report written in a language alluded to as HTML. CSS is the foundation of the entire Internet, alongside HTML and JavaScript. CSS is intended to part show and content material.

Counting design, hues and textual styles. This division can work on happy openness, presents greater adaptability and control over the detail of show capacities, allows in sharing of more than one net pages via making suitable CSS in a different css report, and lessens the intricacy and reiteration of content systems. Css file should be saved between pages that proportion the report and its layout to improve page load velocity. Content formatting and differentiation lets in the same net page to be rendered in special rendering modes which includes screen, print, and voice. (Through browser or display reader's voice) and contact discoveries in Braille. CSS also consists of guidelines for formatting if content material is accessed on a cell tool. The layer name takes place to determine a selected priority which speech sample is appropriate if more than one rule fits a given detail? It is a layered machine of precedence forecasting. Textual content/css Internet Multimedia Type (MIME kind) used with CSS within the RFC

HTML

Hypertext Markup Language or HTML (Hypertext Markup Language) is a markup language for reports showed in a web program. This might be worked with through innovation alongside Flowing Templates (CSS) and prearranging dialects which incorporate JavaScript. Internet browsers get HTML reports from the Web or a local store and convert them into media web pages. HTML portrays the semantic state of a page and

most importantly comprises of record type credits. HTML is the primary elements in HTML pages. With HTML

Designs, pix also, different items might be installed as intelligent administration at the showed website page. Documents can be marked up with semantic text systems like headings, paragraphs, symbols, links, commas, and other elements using HTML. HTML elements are designated by using the script using attitude bracket tags. Add content to the web page with tags. Other tags, along with <p> and <input />, provide facts approximately the record's textual content round and straightforwardly, and different labels can be added as sub-components. Programs don't show HTML labels, but use them to decipher content.

The page can embed packages written in a scripting language inclusive of JavaScript into HTML, which impacts the behaviour and content of net pages. The content of CSS determines the advent and layout of the content. The World Wide Web Consortium (W3C), the former developer of HTML and the +cutting-edge developer of CSS standards, helps the usage of CSS in the obvious presentation of HTML. An HTML software (HTA; document extension ".hta") is a Microsoft Windows application

A browser makes use of HTML and dynamic HTML to render a GUI software. A normal HTML document is constrained by way of the safety model of the web browser, communicates handiest with servers and uses simplest the items of page and site cookies. HDA runs as a totally relied on application and consequently has extra privileges inclusive of creating files, modifying and deleting documents and entries within the Windows registry. Because they function outside the browser's protection.

An HTA duplicate can't be performed through HTTP, however ought to be downloaded (as an EXE

record) and achieved from the nearby document framework.

RESULT AND DISCUSSION

Client Application

Based at the perceptions of the device, we show up at the accompanying perceptions and evaluations.

- A document storage device functions nicely in any respect ranges of provider implementation: read, encode, cut up, load, listing, load, merge; Decoding each entry's metadata is stored in a sharp settlement

□ the task is a lot easier if a duplicate of all exchanges is put away in a local data set and a review contract is made for confirmation most effective at the collision web site. It makes it easy to use.

□ Node can get notifications approximately all information in actual time. Node software may be very consumer pleasant and may easily understand the present actual time machine.

□ Fragmentation and encryption-decryption of files on the whole rely upon elements: file size and processing speed. Any file processing time relies upon on the scale of the record. Thus, downloading a large document puts a great load at the Android thread and in a few cases destroys the energy of the machine and causes the gadget to crash. The patron facet of the web application could be very appealing and smooth to apply to make agreement loading, pockets and commercial enterprise tactics clean and fast.

Using a complicated node

□ Node interfaces with:

□ the interface indicates how a lot area the files have taken up and the quantity How many customer documents and general files acquired by means of node?

Obtained

□ a listing of all fragments can be discovered on this desk as a file. The listing suggests the hash, creation and expiration dates, and the variety of documents for each download. Results like report downloads are displayed in real time the use of the internet.

□ This connection point shows information roughly a free hub with public key, hub ID and individual key. Different hosts associated with or situated through this host are shown with host ID, IP cope with and port.

CONFIRM TRANSACTION Private Network

Ganache 1
627306...Ef57
99.792 ETH
57503.33 USD

345cA3...3e10

Amount 0.00 ETH 0.00 USD

Gas Limit 128913 UNITS

Gas Price 100 GWEI

Max Transaction Fee 0.012891 ETH 7.43 USD

Max Total 0.012891 ETH 7.43 USD

Data included: 132 bytes

RESET SUBMIT REJECT

MetaMask is a cryptocurrency wallet that interacts with Ethereum. MetaMask allows users to ship and receive account keys, random transactions, Ethereum-based cryptocurrencies and tokens, and securely connect with decentralized programs thru a web browser or like-minded. Built-in cellular feed app.

ACCOUNTS				BLOCKS		TRANSACTIONS		LOGS	
CURRENT BLOCK				GAS PRICE		GAS LIMIT		NETWORK ID	
4				2000000000		6721675		5777	
								HTTP://127.0.0.1:3545	
								MINING STATUS	
								AUTOMINING	
MINEMONIC				NO PEER					
candy maple cake sugar pudding cream honey rich smooth crumble sweet treat				m/44'/00'/0'/0'/account_index					
ADDRESS				BALANCE		TX COUNT		INDEX	
0x627306090aba83A6e1400e9345bC60c78a8BEf57				99.94 ETH		4		0	
ADDRESS				BALANCE		TX COUNT		INDEX	
0xf17f52151E8EF6C7334FAD080c5704D77216b732				100.00 ETH		0		1	
ADDRESS				BALANCE		TX COUNT		INDEX	
0xC5fd4076b0F3A5357c5E395ab970B5854098Fef				100.00 ETH		0		2	
ADDRESS				BALANCE		TX COUNT		INDEX	
0x821aEa9a577a9b4429989c15c88cf3087F3b5544				100.00 ETH		0		3	
ADDRESS				BALANCE		TX COUNT		INDEX	
0x0d104e623D18F9F8A5D0b95830F7d3839406C6AF2				100.00 ETH		0		4	

See the log output from the inner Ganache module, consisting of responses and different important debugging data. Set up advanced mining with one click on and lock in the time that high-quality suits your mining wishes. Byzantium is solid, presenting you with the brand new Ethereum functions wanted for decentralized software development. Read all the blocks and transactions to get an concept of what is going on interior.

Upload File to IPFS

FileName

Upload File No file chosen

In decentralized cloud storage, end-to-end encryption on every file comes as preferred: each file is encrypted at the user's laptop before being downloaded, broken into portions, after which dispensed to unrelated nodes on our community. Additionally, centralized cloud garage fees extra than our decentralized community. IPFS is used here to distribute facts blocks over a multi-node network Data can be stored in an encrypted format. Available in IPFS in order that the network activity of this

network may be viewed using a pockets like Ganache used on this venture.

File Uploaded

Name : Sample

LINK : [QmcPXZAmAXyDjUqcveb6K9pfjTZ3XBGGAhK3MBFENtsF1n](#)

The contents of the recovered quantity are decrypted and merged with the original file where the contents are hashed in order that no facts is misplaced. This type of cloud garage is comfortable and cost-powerful in comparison to centralized servers. Each node on this community acts as a server, and facts is shipped between them without understanding what they incorporate or who they belong to.

CONCLUSION

The assignment consists of a P2P network in which nodes can connect with the community and offer garage services to the patron. The device uses more than one cryptography and community algorithms and provides two essential offerings to the purchaser: comfy document storage and mystery sharing. The agreement among the events is ruled through a bigger contract and uses an ERC20 token because the charge of the provider. Both device stages may be implemented separately as additives for numerous use instances. Completed an exciting study undertaking and

Research in the subject of cryptography and cryptography. There is usually room for improvement in any assignment. The application may be in addition advanced by including the subsequent capabilities:

We are most effective have mobile app for clients. So can a web application should be done for this purpose.

So huge documents can be effortlessly encrypted, break up and merged.

- In our machine, we will upload a restore device to the defective component.
- The method to this is to introduce principal authority. This also can be completed by using the utilization of a third audience hub haphazardly settled on from the local area.
- Calculations and conventions utilized in cryptography strategies can be further customized.

FUTURE SCOPE

In the future this version may be progressed in order that better encryption algorithms may be developed, and this part can be used inside the development of even better statistics distribution algorithms. Block-based cloud computing does not just decentralize statistics garage. More than ever, those technological tools are getting used for facts logistics. Multiple consumer powers are allowed to get right of entry to the information concurrently, and they could have interaction, interpret, and alternate them as they see in shape. This sort of cloud computing answer allows.

Employees can paintings correctly and safely and analyse facts. Virtualization of contracts and different exchanges is less difficult to apply than blockchain, main to a extensive growth in its use in lots of industries.

REFERENCES

- 1] Crosby.M, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, pp. 6–10, 2016.
- [2] David Vorick et al. Sia: Simple Decentralized Storage. 2014.
- [3] Eli Ben-Sasson et al. Zerocash: Decentralized Anonymous Payments from Bitcoin.2014.
- [4] Jin Sun et al. Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS.IEEE.2014.
- [5] Juan Bernet.IPFS - Content Addressed, Versioned, P2P File System.2014.
- [6] King.S - Ppcoin: Peer-to-peer crypto-currency with proof-of-stake.2014.
- [7] McConaghy.T, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen,R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," white paper, BigChainDB, 2016
- [8] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008.
- [9] Shawn Wilkinson et al. Storj: A Peer-to-Peer Cloud Storage Network. 2016.
- [10] Vitalik Buterin.Ethereum: A next-generation smart contract and decentralized application platform.2013
- [11] Wilkinson.S, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.
- [12] Yan Zhu et al. Blockchain-based Decentralized Storage Scheme.IOP publishing.1237.4

(D) CERTIFICATE

