deGIT | coInculcat3.0

The Essential Tools for Web3.0 Developers and Entrepreneurs

Web 3.0 and IPFS-based GIT-CLI Tool and CrowdFunding Portal

A project report submitted or the partial fulfillment of the
Bachelor of Technology Degree in
Computer Science & Engineering
under
Maulana Abul Kalam Azad University of Technology
by

Dhritesh Bhagat
Examination Roll No: 10400119123
University Registration No: 026549 of 2019-20
College Enrollment No: 12019002002026

Debarghya Dutta
Examination Roll No: 10400119201
University Registration No: 012500 (2019-20)
College Enrollment No: 12019002002190

Academic Session: 2019-2023

Under the Supervision of
Prof. Sainik Kumar Mahata

Department of Computer Science and Engineering
Institute of Engineering & Management
Y-12, Salt Lake, Sector 5, Kolkata, Pin 700091, West Bengal, India

Affiliated To

Maulana Abul Kalam Azad University of Technology, West Bengal
formerly known as **West Bengal University of Technology**
In Pursuit of Knowledge and Excellence

Maulana Abul Kalam Azad University of Technology
BF 142, BF Block, Sector 1, Kolkata, West Bengal 700064

INSTITUTE
OF ENGINEERING & MANAGEMENT
श्रद्धावान लभते ज्ञानम्
Salt Lake Electronics Complex, Kolkata - 700091, WB, INDIA

Phone : (033) 2357 2969/2059/2995
(033) 2357 8189/8908/5389
Fax : 91 33 2357 8302
E-mail : director@iemcal.com
Website : www.iemcal.com

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report titled "**Web3.0 and IPFS-based GIT-CLI Tool and CrowdFunding Portal**", submitted by Dhritesh Bhagat, Exam Roll No: 10400119123, University Registration Number: 026549 of 2019-20, College Enrollment Number: 12019002002026 and Debarghya Dutta, Exam Roll No: 10400119201, University Registration Number: 012500 of 2019-20, College Enrollment Number: 12019002002190, students of Institute of Engineering & Management in partial fulfillment of requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering**, is a bonafide work carried out under the supervision of Prof. Sainik Kumar Mahata during the final year of the academic session of 2019-2023. The content of this report has not been submitted to any other university or institute for the award of any other degree.

It is further certified that the work is entirely original and the performance has been found to be satisfactory.

_____

Prof. Sainik Kumar Mahata
Assistant Professor
Department of Computer Science and Engineering
Institute of Engineering & Management

_____

Prof. Indraneel Mukhopadhyay, Ph.D.
H.O.D.
Department of Computer Science and Engineering
Institute of Engineering & Management

_____

Prof. Arun Kumar Bar
Principal
Institute of Engineering & Management

Institute of Engineering and Management

Declaration for NON-COMMITMENT OF PLAGIARISM

We, Dhritesh Bhagat & Debarghya Dutta, students of Bachelor of Technology in the Department of Computer Science and Engineering, Institute of Engineering & Management have submitted the project report in partial fulfillment of the requirements to obtain the above-noted degree. We declare that we have not committed plagiarism in any form or violated copyright while writing the report. We have acknowledged other authors' sources and/or credit wherever applicable. If subsequently it is found that we have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: Dhritesh Bhagat
Full Signature:

_____

Name of the Student: Debarghya Dutta
Full Signature:

_____

Date:

Content:

Abstract

The emergence of blockchain technology has shown great potential for decentralizing industries such as finance, supply chain management, and voting systems. However, the software development industry still requires innovative solutions to integrate decentralized technologies to address issues of collaboration and funding.

This thesis project aims to develop a blockchain-based decentralized Git CLI tool and a decentralized crowdfunding platform to address these problems using Web 3.0 and IPFS. The primary objective is to provide a secure and decentralized way for developers to collaborate and manage their software projects while enabling entrepreneurs to raise funds for their projects in a decentralized manner.

The current centralized model of software development and crowdfunding platforms is vulnerable to data breaches, censorship, and a lack of transparency in fundraising. The proposed solutions leverage blockchain technology, Web 3.0, and IPFS to provide a secure, transparent, and censorship-resistant platform for software development and crowdfunding. The decentralized Git CLI tool utilizes blockchain technology and IPFS to enable secure collaboration, version control, and distribution of software code. The decentralized crowdfunding platform uses smart contracts on Web 3.0 to facilitate transparent and secure fundraising without intermediaries.

The project involves various methodologies such as system design, software development, and testing to achieve its objectives. The expected outcome is a functional prototype that can be tested and evaluated for its effectiveness in promoting secure collaboration and transparent fundraising in the software development industry.

The GIT-CLI Tool's success in promoting secure collaboration in software development is due to its use of IPFS. IPFS provides a decentralized way of storing files, making it more secure than traditional centralized storage systems. The tool uses IPFS to store files securely and to enable secure collaboration between developers. Additionally, the tool's use of blockchain technology ensures accountability and transparency by recording all changes made to the code on an immutable ledger. The decentralized crowdfunding platform has been successful in promoting secure and transparent fundraising by leveraging blockchain technology and smart contracts. Transactions are transparent and secure, and there is no central authority controlling the fundraising process, mitigating the risk of censorship. The platform's success can be attributed to its use of decentralized technologies, which are gaining popularity in various industries.

Acknowledgment

We would like to take this opportunity to express our gratitude to all the individuals who have provided constant support during our Bachelor of Technology course. First and foremost, we would like to extend our sincere appreciation to our mentor and guide, **Prof. Sainik Kumar Mahata**, for his unwavering support and motivation throughout the project. His guidance and encouragement have been invaluable in making this project a success. We are also grateful to our H.O.D, **Prof. Indraneel Mukhopadhyay, Ph.D.** for his constant inspiration and encouragement, and to all the technical, non-technical, and office staff of our department for their support.

We would like to express our heartfelt thanks to all our friends in the department for creating a friendly and supportive environment.
We would also grateful to our Director, **Prof. Satyajit Chakraborti**, for providing an outstanding platform for our academic growth. We also acknowledge our Principal, **Prof. Arun Kumar Bar**, for being a constant source of inspiration.

Additionally, we would like to thank all the technical, non-technical and office staffs of our department for extending facilitating cooperation wherever required. We cannot forget to acknowledge the immense support of our families. We are deeply grateful to our parents for their unwavering love and support, and for the sacrifices they have made to help us stand on our own feet.

Lastly, we would like to thank everyone who has provided any form of assistance towards the realization of this project. We apologize for not being able to mention everyone's name individually.

Keywords

Keywords	Description
deGIT (decentralized GIT)	Decentralized GIT is a version control system that enables secure collaboration and distribution of software code without a central authority. It leverages blockchain technology and IPFS to provide a secure and decentralized way for developers to manage their software projects.
IPFS (InterPlanetary File System)	InterPlanetary File System (IPFS) is a protocol and network designed to create a permanent and decentralized method of storing and sharing files. It aims to replace the traditional client-server model of the internet with a distributed model.
Web3.0	Web3.0 refers to the next generation of the World Wide Web, which aims to create a more decentralized and open Internet by utilizing blockchain technology and other decentralized technologies.
keygit	Key used to access deGIT by the user.

List of Figures

List of Tables

Introduction

Blockchain, Web 3.0, and InterPlanetary File Systems (IPFS) are rapidly emerging technologies that are transforming various industries. Blockchain, the underlying technology behind cryptocurrencies, has shown great potential in decentralizing industries such as finance, supply chain management, and voting systems. It provides a secure, transparent, and immutable way of recording transactions and sharing data without the need for intermediaries. These decentralized technologies have the potential to revolutionize various industries, including finance, supply chain management, healthcare, and voting systems. Blockchain technology, in particular, has been recognized for its ability to provide secure and transparent transactions without the need for intermediaries, leading to cost savings and increased efficiency. This is based on the Distributed Ledger Technology (DLT).

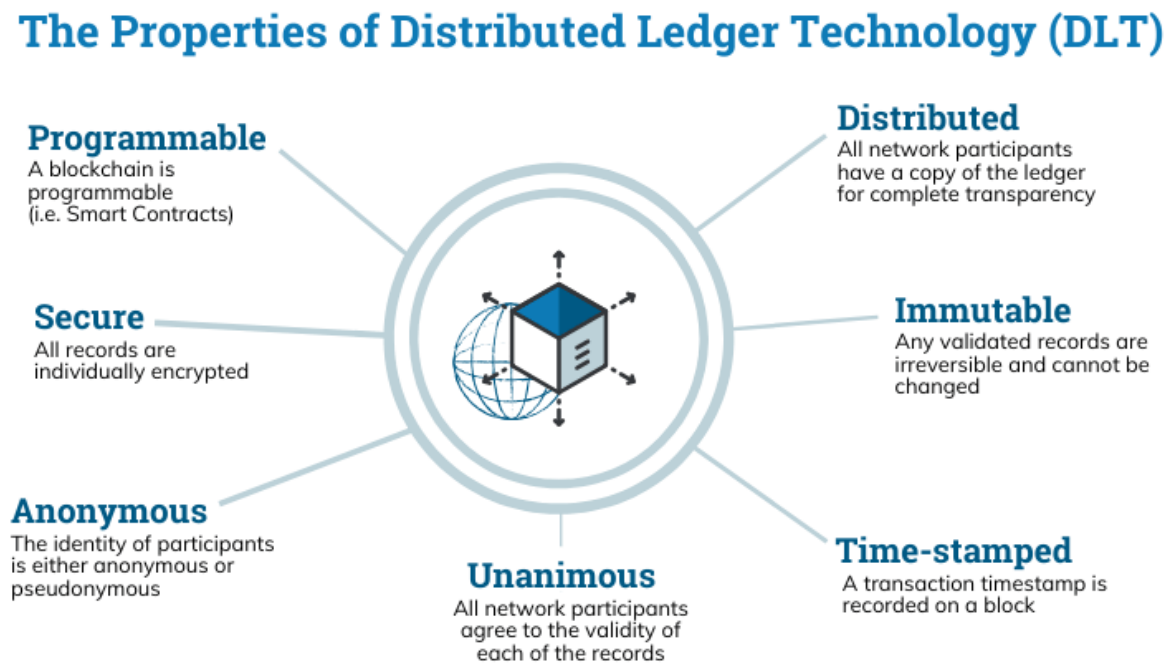


Fig: Properties of Distributed Ledger Technology.

Web 3.0, also known as the decentralized web, is the next evolution of the internet. It aims to create a more open and decentralized internet that is not controlled by a few large corporations. This new web is expected to be more secure, transparent, and private, providing users with more control over their data and online identity.

InterPlanetary File System (IPFS) is a distributed file system that provides a secure and decentralized way of storing and accessing files. Unlike traditional centralized storage systems, IPFS allows files to be distributed across a network of nodes, making it more secure and resistant to censorship.

The combination of these technologies has the potential to provide a more secure, transparent, and decentralized way of collaborating and funding projects. This is particularly relevant in the software development industry, which currently relies on centralized platforms for collaboration and fundraising. By leveraging blockchain technology, Web 3.0, and IPFS, developers can collaborate

and manage their software projects in a more secure and decentralized manner, while entrepreneurs can raise funds for their projects without intermediaries.

This thesis project aims to develop a blockchain-based decentralized Git CLI tool and a decentralized crowdfunding platform using Web 3.0 and IPFS to address the issues of collaboration and funding in the software development industry. This project acts as a proof of concept of the decentralized GIT (deGIT) and a robust document management setup that can be used not only for software development but also for document management and sharing replacing the cloud applications which pose a threat to the privacy of the users and the stability of the system.

The decentralized GIT CLI tool will enable developers to collaborate securely by utilizing blockchain technology to ensure version control and distribution of software code. The tool will provide a secure and transparent platform for developers to share code and work collaboratively, eliminating the risk of data breaches and censorship.

Aim and Objectives:

This thesis project aims to design and develop a decentralized GIT CLI tool and a decentralized crowdfunding platform using blockchain technology, to provide secure and transparent collaboration and funding for software development.
This project aims:

- To design and develop a blockchain-based decentralized GIT CLI tool that utilizes distributed ledger technology to ensure secure and transparent version control, distribution, and collaboration of software code among developers.
- To implement a decentralized crowdfunding platform that utilizes smart contracts to enable transparent and secure fundraising without intermediaries, facilitating entrepreneurs to raise funds in a censorship-resistant manner.
- To ensure data privacy and security through the use of encryption and cryptographic protocols to prevent unauthorized access, data breaches, and censorship of software code and crowdfunding transactions.
- To evaluate the performance and scalability of the decentralized GIT CLI tool and crowdfunding platform in real-world scenarios, identifying potential limitations and areas for improvement.
- To contribute to the software development industry by promoting innovation and creativity through the use of decentralized technologies, enabling secure collaboration and transparent fundraising.
- To provide an open-source platform for developers and entrepreneurs to contribute to the development and improvement of the decentralized GIT CLI tool and crowdfunding platform, fostering a collaborative community of users.
- To document the development process and provide a user-friendly guide for developers and entrepreneurs to utilize the decentralized GIT CLI tool and crowdfunding platform, enabling easy adoption and integration with existing software development practices.

Related Works

Related Works for Document Management System using IPFS:

InterPlanetary File System (IPFS) has become a popular choice for decentralized file storage and sharing. One of the most notable related works is the OpenBazaar project, which utilizes IPFS to provide a decentralized online marketplace. Another example is the Textile Photos app, which uses IPFS to enable users to store their photos and share them with others securely.

In the field of document management systems, there are several related works using IPFS. One of them is the DAppBox project, which utilizes IPFS to provide a decentralized document management system. Another example is the Textile Docs app, which enables users to store and share documents securely using IPFS. The combination of blockchain and IPFS offers a secure and decentralized way of storing, sharing, and managing documents, reducing the risk of data breaches, data loss, and cyber-attacks.

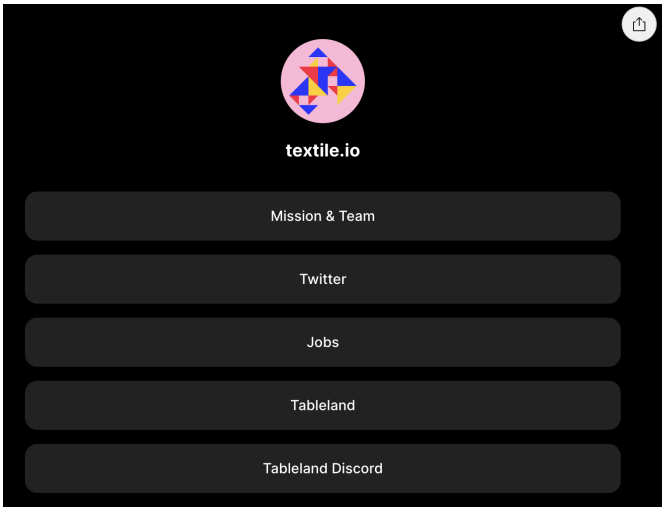


Fig: Textile Docs | IPFS file storage client

Another example of a document management system using blockchain and IPFS is "Chainy". Chainy is a decentralized platform for storing and sharing documents securely. The platform uses IPFS to store documents, making them accessible from anywhere in the world. Blockchain technology is used to provide a transparent and tamper-proof record of all transactions related to the documents. One example of such a system is the "Doculus" document management system developed by a team at the Indian Institute of Technology (IIT) Delhi. Doculus uses a combination of blockchain and IPFS to ensure the security and privacy of documents. The system uses IPFS to store the documents in a decentralized manner, making them easily accessible and retrievable by authorized users. Blockchain technology is used to provide an immutable and tamper-proof record of all transactions related to the documents, ensuring accountability and transparency.

Another such scientific concept approach towards this was provided by Sarang et al in their paper on Document Management System Empowered by Effective Amalgam of Blockchain and IPFS https://doi.org/10.1016/j.procs.2022.12.036.

Related Works for Crowdfunding using Blockchain:

Crowdfunding has become a popular method for raising funds for various projects. With the advent of blockchain technology, several related works have been proposed and implemented in this field.

One of the most notable related works is the Ethereum-based Golem Network, which utilizes blockchain technology to provide a decentralized supercomputing network. Another example is the Brave browser, which enables users to earn cryptocurrency by viewing advertisements and then donating that cryptocurrency to their preferred content creators.

In the field of crowdfunding, there are several related works using blockchain technology. One of them is the Giveth project, which utilizes blockchain technology to enable transparent and secure donation tracking. Another example is the BitGive Foundation, which uses blockchain technology to provide transparency and accountability in charitable donations.

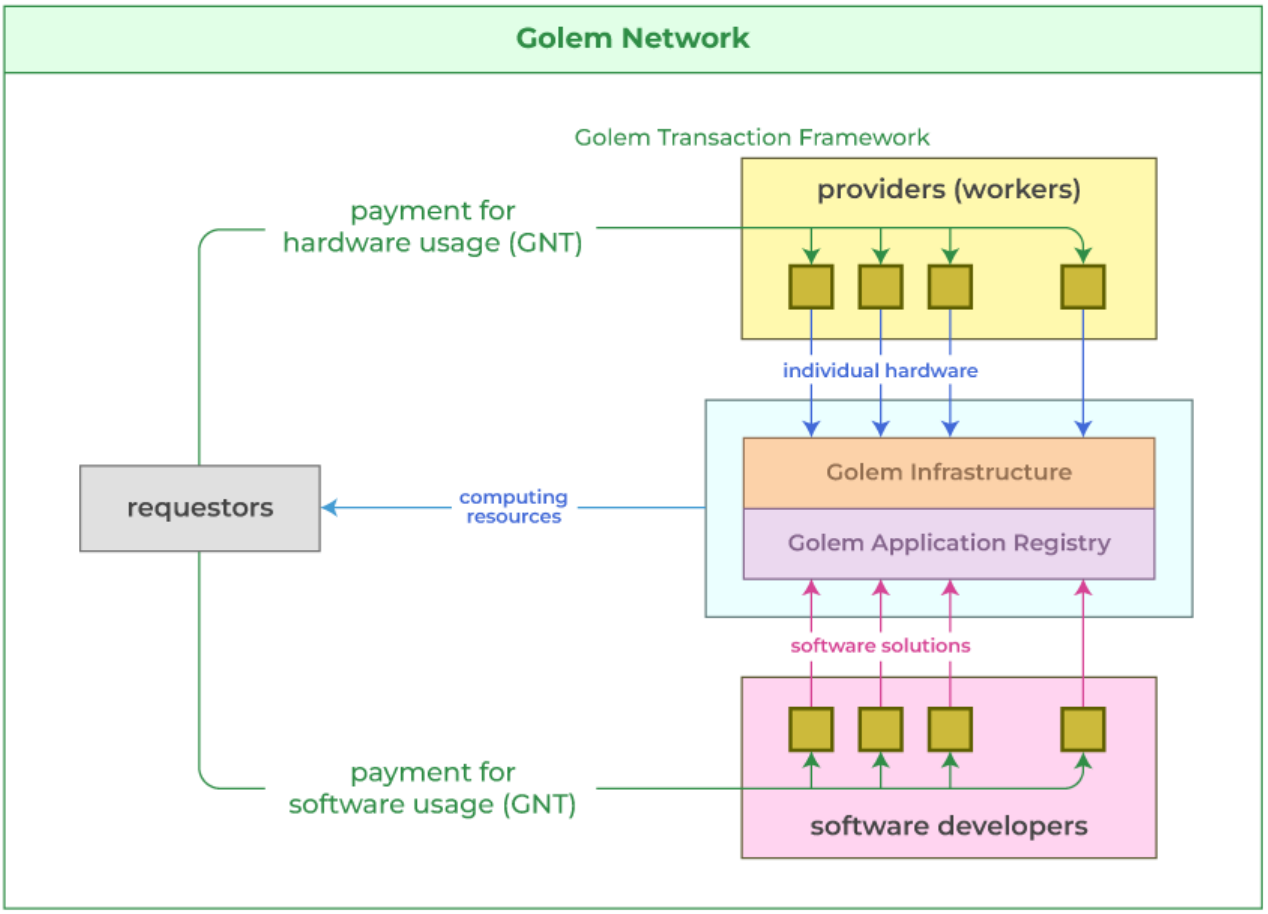


Fig: Ethereum-based GOLEM Network

Basic Underlying Concept

deGIT

The basic underlying principle of deGIT is the concept of IPFS. InterPlanetary File System (IPFS) is a peer-to-peer protocol designed to facilitate decentralized file storage and sharing. It was created by Juan Benet and released in 2015. IPFS enables users to store and retrieve files by addressing the content rather than the location of the file. This content-addressed approach makes IPFS more resilient to censorship and ensures that the file is always accessible as long as at least one node on the network is hosting it.

At the heart of IPFS is the concept of a Merkle DAG (Directed Acyclic Graph), a data structure used to represent a set of related files or objects. Each node in the DAG represents a file or object, and the edges between nodes represent the relationships between them. The Merkle DAG enables efficient content addressing, which means that each file or object is identified by a unique cryptographic hash based on its content. This hash serves as the file's address on the IPFS network.

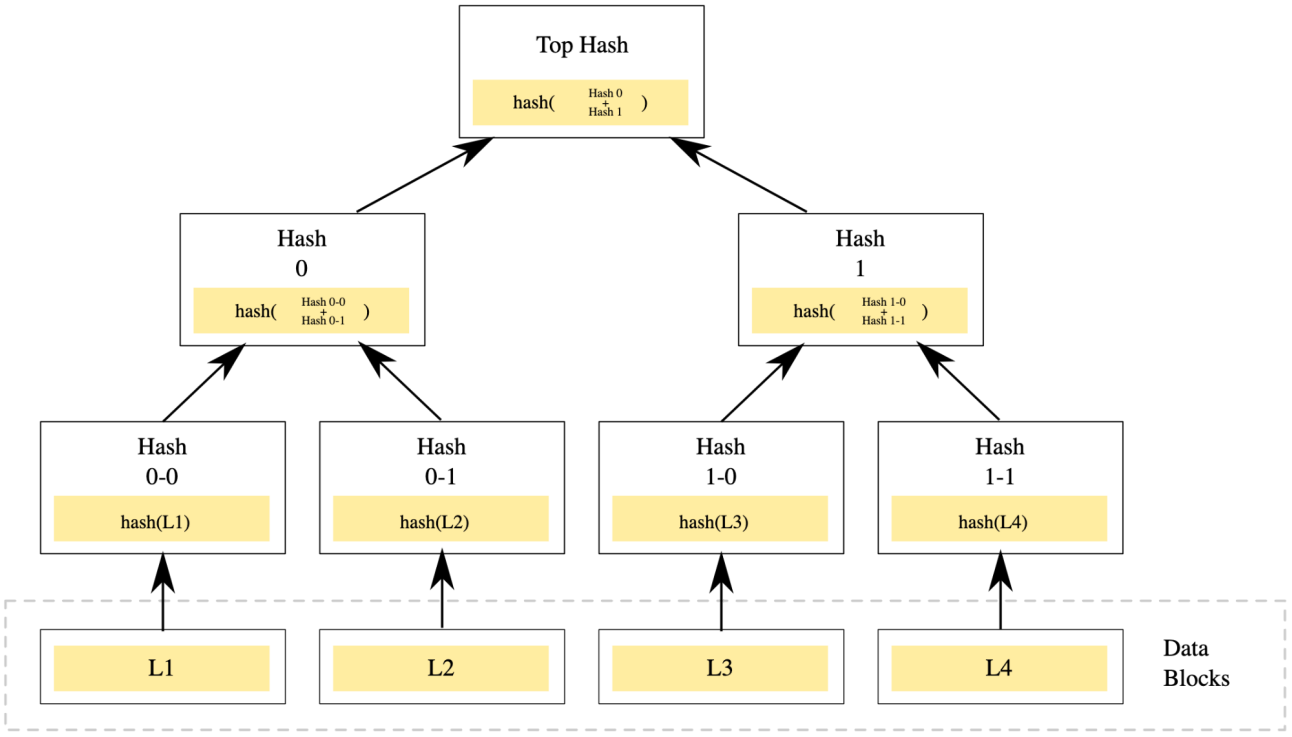


Fig: Merkle DAG (base of IPFS)

When a user wants to retrieve a file on IPFS, they send a request to the network with the file's address (i.e., it's hash). The network then routes the request to a node that has a copy of the file. The file is then retrieved from that node and sent back to the requesting user. If the file is not available on the first node, the request is forwarded to another node until the file is found.

One of the main advantages of IPFS is its decentralized nature. Files are stored on a network of nodes, rather than in a centralized location, making it more resistant to censorship and ensuring that files are always available even if one or more nodes go offline. Additionally, IPFS uses a content-addressed approach, meaning that files are identified by their content rather than their location. This ensures that the file is always accessible as long as at least one node on the network is hosting it.

IPFS is also designed to be highly scalable. Files are distributed across the network, and each node only stores a subset of the files. This means that as the network grows, the storage capacity also increases, without requiring a centralized infrastructure.

In summary, IPFS is a decentralized protocol that enables efficient content addressing and distributed file storage. It provides a resilient and scalable alternative to traditional centralized file storage systems.

Proposed System:

- Unique user ID created for the user (Cant be forgotten)
- User assigned a keygit (This has to be remembered)
- User creates and then pins a file to IPFS, generating a deGIT file ID
- deGIT file ID can be used to give access or to edit or the view the file contents
- deGIT file is also staged/unstaged that offers an additional feature to the user to create a file and then stage the latest version of the file
- deGIT file hashing is also included which reading the file that is being uploaded to IPFS.
- deGIT file is forever stored in the IPFS which can provide an added benefit of privacy and security to the file.

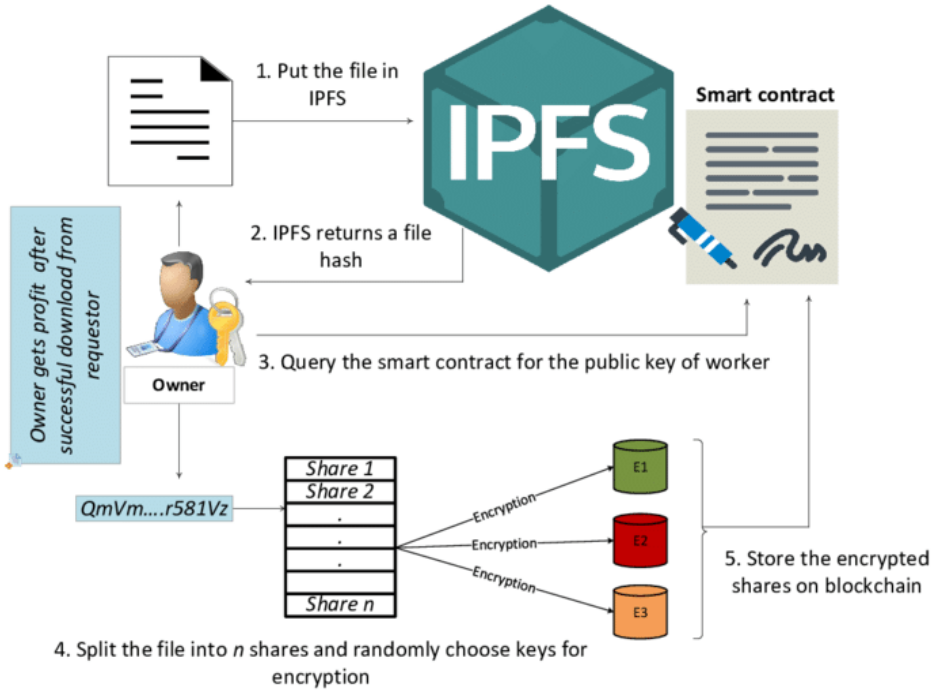


Fig: Data Sharing on IPFS

- **Unique User ID Creation**

  The user is assigned with a unique user id, something in the form of “c6b55db3-68fa-4215-8062-2cc09793887e”.
  This becomes the unique id of the user that can be used to give access to any file to the user. This has to be remembered as this forms the only way through which a user can create, edit, view, pin, or stage/unstaged files.

- **keygit**

  A new key is assigned to each user. This is something that is used to encrypt the file while pinning the file to IPFS or while staging and unstaging the file. A user can have multiple user IDs on a device just like branches on git, however, each of these will have different access capabilities and file trees.

- **pin/edit/view**

  A user can create a file as normally as any other file is created and then can pin the file to IPFS. This not only adds the file to ipfs but also creates a history tag for the file, and also adds the user under the creator tag of the file. Each file has its own history and version list. This can be used to revert a file back to its previous versions. This can even update the access of the files. Certain access controls can be provided to the various users whose id has been pinned or added while pinning or editing the file and that can create a secure access control protocol for the file.

- **Revert to a previous version using the History of the file**

  Each file has its unique history pattern that can be used to revert a file back to its previous versions. This feature can be used in the software development cycle to revert back to a previous commit. However unlike git where reverting to a commit trashes the skipped commits, here the history itself gets updated and the version revert is added as a new commit to the history, and that prevents the loss of any kind of important information that might have been added to the file in the skipped versions.

- **Pinning (File ID)**

  Every time a user pins a new file to IPFS a new deGIT file ID is created which then becomes a single lines access key to the file. Anyone with the file id can try to access the file however only the userIDs that have permission can view the intricate details and the history of the file. This becomes very efficient when being used for document verification and authentication while file sharing.
  Similarly updating a pinned file just requires pinning the file using the file ID. However, a user who has access to the file can create his own version of the file however that has a different file id, and the history of both the files is different.

- **File Duplicacy**

  Uploading the same file again and again will not re-upload the file to IPFS each time. Since IPFS depends on hashing so the same file without any changes will generate the same hash, however uploading it at a different time will update the history with the timestamp and the

file id but the file will not reupload as the file is already present in the IPFS instead, a tag of Duplicate will be added to the file.
This adds multiple benefits, one being storage efficiency, the next being the effect that any person who is trying to reupload a file, file can then get a duplicate tag. Future developments on this project can be a fork where this is used as an authenticity factor as reuploading can corrupt a file and a file should only be uploaded once.

- **Staging/Unstaging a File:**

  Each file when being created can be staged. This enables the user to work in levels and revert back to the last local saved setup of the file if something goes wrong. This is done using the keygit that is created on the user ID creation. That is used to create an encrypted version of the file that is stored along with the hidden cache files of the project. This works independently on any number of users that are present in the system.

This is how IPFS helps in building up the deGIT protocols. By using IPFS to store code repositories, developers can even distribute and share their code in a peer-to-peer network, removing the need for a centralized repository like GitHub. This can help to achieve the objective of decentralizing Git, as it allows for a more resilient and censorship-resistant system where developers can collaborate and share code without relying on a single point of failure. Additionally, IPFS can provide improved performance and faster download speeds, as files are distributed across multiple nodes in the network, reducing the load on any one particular node.

coInculcat3.0

Crowdfunding is a method of raising funds for a project or initiative by soliciting small contributions from a large number of people. A crowdfunding application can be built using various underlying technologies, and one such option is to use the Goerli test network. Goerli is an Ethereum test network that enables developers to test their smart contracts and decentralized applications without incurring real-world costs.

The basic concept underlying a crowdfunding application is to create a smart contract on the Ethereum blockchain that manages the collection and distribution of funds. The smart contract contains a set of rules and conditions that govern how the funds will be collected, how the project will be executed, and how the funds will be distributed to the investors.

The technology used to build the crowdfunding application on the Goerli test network includes Solidity, a high-level programming language used for writing smart contracts on the Ethereum blockchain. Other supporting tools and frameworks used include Remix, a web-based IDE for developing and testing smart contracts, and MetaMask, a browser extension that allows users to interact with the Ethereum blockchain.

To build the crowdfunding application, the developer would first define the project details, including the amount of funds required, the duration of the campaign, and the rewards for investors. Then, a smart contract is created that sets these parameters and governs the entire crowdfunding process.

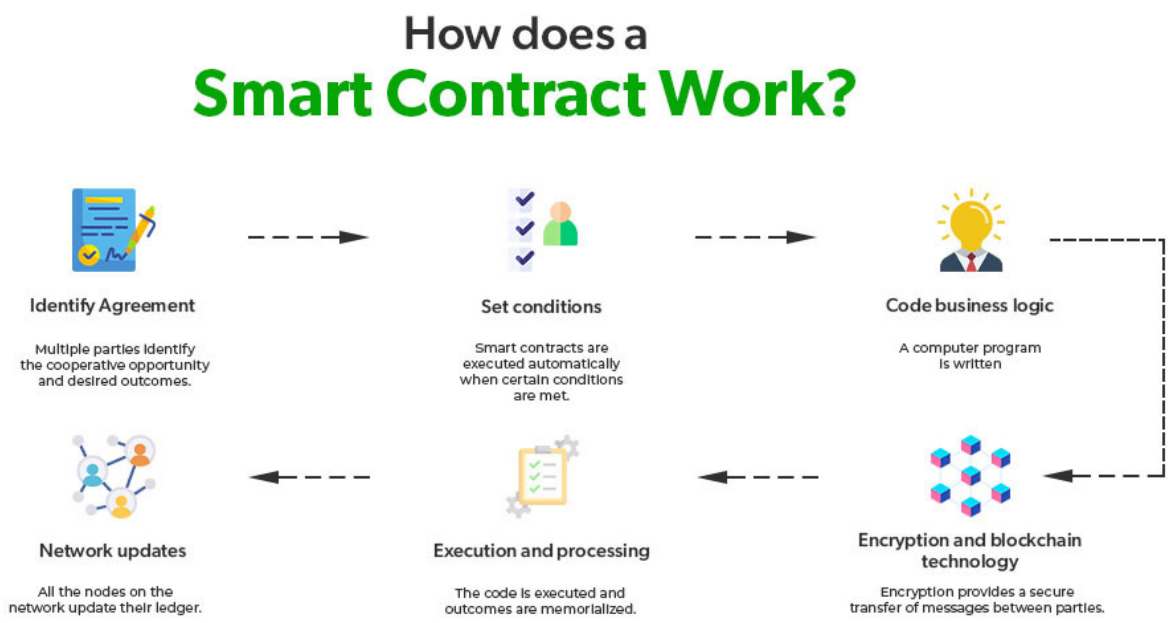


Fig: Working of a Smart Contract

The contract would also include the rules for releasing the funds to the project owner or returning them to the investors if the project fails to meet its funding goal.

Once the smart contract is deployed on the Goerli test network, users can interact with it using a web interface. They can contribute funds to the project using Ether or any other supported cryptocurrency, view the progress of the campaign, and claim their rewards once the project is successfully funded.

In conclusion, building a crowdfunding application using Goerli as a test network requires a solid understanding of smart contract development, Solidity programming language, and other supporting tools and frameworks. By leveraging the Goerli test network, developers can test their smart contracts and crowdfunding applications in a safe and cost-effective manner, before deploying it on the main Ethereum network.

Proposed Architecture

deGIT Architecture

To achieve the goals of a decentralized GIT with IPFS, the following architecture can be proposed:

User Management:

A user management system needs to be implemented to assign a unique user ID to each user. This can be achieved by creating a smart contract on the Ethereum network which assigns a unique ID to the user at the time of registration.

Unique User ID Creation:

To ensure that each user has a unique identity within the system, a unique user ID is generated when a new user is registered. This user ID is a universally unique identifier (UUID), which is a 128-bit number that can be represented in various formats, such as hexadecimal or base64.
The UUID is generated using a random or pseudo-random algorithm, which ensures that the probability of generating the same UUID twice is extremely low, even across different devices and systems. This UUID becomes the unique identifier of the user in the system, which is used to give access to any file to the user.
The UUID is also stored securely on the user's device and is used for authentication purposes whenever the user logs in to the system. This ensures that the user's identity is protected and cannot be easily forged or impersonated. Additionally, since the UUID is unique, it allows the system to easily track and manage each user's files and permissions within the system.

Authentication:

Since the system is decentralized, traditional authentication methods like username and password may not work. Hence, a private key-based authentication system can be used where each user generates their private key, which is stored securely on their device. This private key can be used to authenticate the user and provide access to their files.

KeyGIT Generation:

A unique keyGIT is generated for each user, which is used to encrypt the files before pinning them to IPFS. This keyGIT is also used to decrypt files when they are accessed or edited.

keyGIT:

A new key is assigned to each user. This key is used to encrypt the file while pinning it to IPFS or while staging and unstaging the file. Each user can have multiple user IDs on a device just like branches on git. However, each of these will have different access capabilities and file trees.
The keygit is an essential part of the system as it helps ensure the security and privacy of the files. When a user pins a file to IPFS or stages/unstages a file, the keygit is used to encrypt the file. This ensures that only the user who has the keygit can access the file.

The keygit is also used to manage access control. Each user can have different access capabilities based on their keygit. For example, a user may have read-only access to a file if their keygit does not have write access. This ensures that files are protected from unauthorized access and modification.
In addition, the keygit can be used to manage version control. Each version of a file can be encrypted with a different keygit. This ensures that only users with the correct keygit can access specific versions of a file. It also ensures that changes made to a file are tracked and managed properly.

IPFS Integration:

IPFS is used as the underlying storage layer for the files. When a user creates a new file, it is pinned to IPFS, which generates a unique deGIT file ID. This deGIT file ID can be used to give access, edit, or view the file contents.

Consensus Management:

To ensure the integrity of the decentralized GIT system, a consensus mechanism should be in place to prevent malicious actors from manipulating the data. One way to achieve this is through a consensus algorithm such as Proof of Work (PoW) or Proof of Stake (PoS). In PoW, nodes (computers) in the network compete to solve a complex mathematical problem, and the first one to solve it gets to add a new block to the blockchain and is rewarded with cryptocurrency. In PoS, nodes are selected to validate transactions based on the amount of cryptocurrency they hold and "stake" in the network. The advantage of PoS over PoW is that it consumes less energy and is less susceptible to 51% attacks. Another consensus mechanism that can be used is Delegated Proof of Stake (DPoS), where the stakeholders in the network vote to select delegates who validate transactions on their behalf.
There are several types of consensus algorithms used in blockchain file management, including:

- **Proof of Work (PoW):** In PoW, miners compete to solve a complex mathematical puzzle in order to add a new block to the blockchain. The first miner to solve the puzzle and add the block is rewarded with cryptocurrency. This algorithm is used in Bitcoin.
- **Proof of Stake (PoS):** In PoS, validators are chosen to add new blocks to the blockchain based on the amount of cryptocurrency they hold and are willing to "stake" as collateral. Validators who successfully add new blocks are rewarded with transaction fees. This algorithm is used in Ethereum.
- **Delegated Proof of Stake (DPoS):** In DPoS, token holders vote for delegates who are responsible for validating transactions and adding new blocks to the blockchain. Delegates are rewarded with transaction fees. This algorithm is used in EOS.
- **Proof of Authority (PoA):** In PoA, a group of validators is chosen to add new blocks to the blockchain based on their identity and reputation. Validators are typically chosen by a centralized authority. This algorithm is used in private and consortium blockchains.
- **Practical Byzantine Fault Tolerance (PBFT):** In PBFT, a set of nodes are designated as validators and must come to a consensus on the order of transactions. This algorithm is used in permissioned blockchains.

Each consensus algorithm has its own strengths and weaknesses and is suited for different use cases. The choice of consensus algorithm depends on the specific requirements of the blockchain network, including the level of decentralization, scalability, and security needed.

technology	storage mechanism	data model	networking stack	identifier	address composition	use cases	similarity to IPFS	hashing algorithm
bittorrent	P2P file-sharing	merkle DAG	TCP/IP	torrent file	filename + sha1 hash	file sharing	low	SHA-256
hypercore	decentralized data-sharing	merkle DAG	UDP	dat key	dat key	decentralized data sharing	medium	SHA-256
git	version control	commit history	TCP/IP	commit hash	commit hash	version control	medium	SHA-1, SHA-256
Secure Scuttlebutt (SSB)	decentralized social network	append-only log	Scuttlebutt Protocol	feed id	feed id	decentralized social networking	high	SHA-256
filecoin	blockchain-based storage	merkle DAG	proof-of-replication	libp2p	cid	cid	decentralized data storage	high
storj	decentralized storage	erasure coding	proof-of-retrievability	UDP	farmer ID	farmer ID + file metadata	encrypted cloud storage	medium
Holo	decentralized application	distributed hash table	distributed hash table	actor model	agent ID	agent ID	decentralized applications	medium
Swarm	decentralized storage	distributed hash table	proof-of-custody	libp2p	chunk ID	chunk ID	decentralized data storage	high
sia	decentralized storage	erasure coding	proof-of-work	UDP	sector ID	sector ID + file metadata	encrypted cloud storage	medium
arweave	blockchain-based storage	blockweave	proof-of-access	TCP/IP	block ID	block ID	permanent data archiving	low

Table 1. Comparison of Consensus between available systems and IPFS

IPFS is a general-purpose file system that uses a distributed hash table (DHT) to route and transfer content-addressed data. This sets it apart from other solutions with a more specific focus or use of a specific data storage mechanism. For example:

- BitTorrent is a peer-to-peer (P2P) file-sharing protocol that uses a centralized tracker to manage the distribution of files among peers. It focuses on file-sharing rather than file storage.
- Storj and Sia are decentralized cloud storage platforms that use distributed networks of nodes for data storage. They focus on providing cloud storage services rather than a general-purpose distributed file system.
- Arweave is a decentralized, permanent storage platform that uses a novel data structure called a "blockweave" for data storage. It focuses on providing permanent storage rather than a file-sharing system.
- Filecoin is a decentralized storage network that allows users to rent out disk space. It focuses on providing a decentralized storage marketplace. It uses a proof-of-replication consensus mechanism and supports payment in various cryptocurrencies. Filecoin is built on IPFS and uses the IPFS network for data storage and retrieval. Filecoin and IPFS are complementary technologies providing decentralized and efficient storage solutions.
- Hypercore is a decentralized data-sharing tool that uses a distributed hash table (DHT) for data storage. It focuses on enabling data sharing and collaboration.
- Holo is a decentralized hosting platform that uses a unique data storage and sharing mechanism called Holochain. It allows users to host and run web-based applications on a peer-to-peer network.
- Swarm is a decentralized storage and sharing platform built on the Ethereum blockchain. It uses smart contracts and cryptographic techniques to securely store and share data. It focuses on providing a decentralized, secure, and censorship-resistant storage solution.

User creates and then pins a file to IPFS, generating a deGIT file ID:

After the user is assigned a keygit, they can create a file and pin it to IPFS. The process of pinning a file to IPFS involves adding the file to the IPFS network and making it available to other nodes on the network. This can be done using IPFS command-line tools or through a web interface.
Once the file is pinned to IPFS, a deGIT file ID is generated. This unique identifier can be used to give access or to edit or view the contents of the file. It can also be used to retrieve files from the IPFS network.
The deGIT file ID is stored in a decentralized manner, meaning that there is no single point of failure or control. This provides an added layer of security and privacy to the file, as it cannot be easily tampered with or deleted.

Staging/Unstaging:

The staging and unstaging features allow the user to create a file and then stage the latest version of the file. This is achieved by creating a hidden cache file on the user's device, which is encrypted using their keyGIT. This cache file is updated every time the user stages or unstages the file.

Hashing:

Before uploading a file to IPFS, the file is hashed to ensure its integrity. This ensures that any changes made to the file can be easily detected. Before uploading the file to IPFS, the file will be hashed using a cryptographic hash function like SHA256. This generates a unique hash for each file and ensures the integrity of the file by providing a way to verify if the file has been modified or not. When the file is uploaded to IPFS, the hash of the file is used to identify the file on the network. This ensures that the file can be retrieved by its hash even if the file name or location changes. Additionally, it allows for content-based addressing where files with identical content will have the same hash, reducing redundancy and increasing efficiency.

Access Control:

Access control is an essential aspect of any decentralized system to maintain the privacy and security of data. In this proposed architecture, access control is implemented using smart contracts deployed on the Ethereum network. Each user's unique ID is linked to their public key and stored on the blockchain, enabling secure authentication of users.

When a user creates a file, they can specify which user IDs can access and edit the file. The user IDs added to the file's access list are stored on the blockchain in a smart contract. Whenever a user wants to access a file, they need to provide their unique ID and the corresponding private key to decrypt and view the file.

The smart contract also stores information on which users have contributed to the file and their contributions. This information can be used to determine the ownership and rights to a file, which is crucial in the case of disputes or conflicts.

Using smart contracts to implement access control offers several benefits, including transparency, immutability, and automation. The decentralized nature of the blockchain ensures that access control policies cannot be tampered with or altered by malicious actors.

Version Control:

Version control is achieved by creating a history tag for each file. When a file is updated, a new version of the file is created, and the history tag is updated. This allows users to revert to a previous version of the file if necessary.

File Duplicacy:

IPFS is designed to detect and eliminate duplicate files. Hence, when a user uploads the same file multiple times, IPFS recognizes that it is a duplicate and adds a tag of "duplicate" to the file. This helps to reduce storage space and ensures that only unique files are stored on the network. In IPFS, files are uniquely identified by their content hash. Therefore, every time a file is pinned, a new content hash is generated, and it becomes the ID of the pinned file. The content hash is generated by a cryptographic hash function, which generates a unique hash value for a given input data. This feature makes the file ID immutable and tamper-proof.

The deGIT file ID serves as an efficient way of sharing the file without the need to share the actual file contents. This feature can be used in scenarios where the file is too large to share, or sharing the file contents might not be desirable due to privacy concerns. Instead, the file ID can be shared with the intended recipients, who can then access the file by using the file ID to retrieve the file from IPFS.

Distributed Storage:

The storage nodes are distributed across the network, and each node stores a copy of the file. This ensures that the data is available even if some of the nodes go offline or become unavailable. The data is stored in a distributed manner to ensure that there is no single point of failure.
Each node in the network stores a copy of the data, which makes the system highly fault-tolerant. Even if some nodes go offline, the data can still be accessed from other nodes. This redundancy ensures that data is always available, and there is no single point of failure in the system.
IPFS uses a content-addressed system for storage, which means that each piece of data is identified by a unique hash. This hash is used to locate the data in the network, and the data can be accessed from any node that has a copy of it.
The distributed storage architecture used by IPFS ensures that data is secure and always available. This architecture is also highly scalable, as new nodes can be added to the network without affecting the performance of the system.

Overall, the proposed architecture combines the power of IPFS and Ethereum to create a decentralized GIT that is secure, efficient, and easy to use.

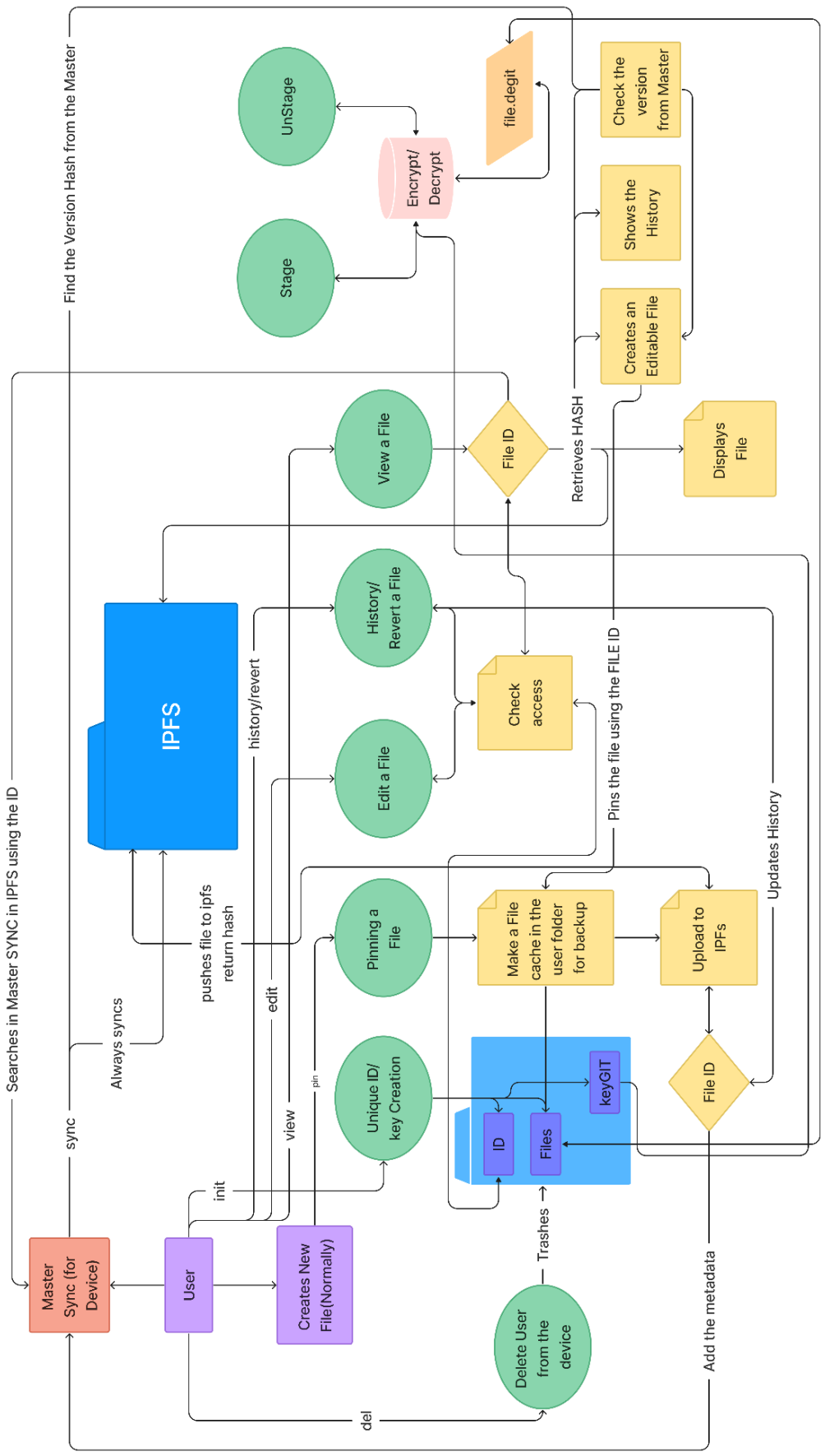


Fig: deGIT Architecture

coInculcat3.0 Architecture

The proposed architecture for a crowdfunding application that utilizes smart contracts and MetaMask wallet is as follows:

User interface:

The crowdfunding application will have a user interface that allows users to create campaigns, view active campaigns, and make donations. The user interface will be built using React JS on Vite and will be hosted on a web server.
The first point of the proposed architecture for a crowdfunding application that utilizes smart contracts and MetaMask wallet is the user interface. The crowdfunding application will have a user interface that allows users to create campaigns, view active campaigns, and make donations. The user interface is built using HTML, CSS, and JavaScript (Mostly react JS on Vite), and is hosted on a web server. The user interface is designed to be user-friendly and intuitive so that users can easily create and manage their campaigns, view active campaigns, and make donations. The user interface also provides a way for campaign creators to track the progress of their campaigns and make data-driven decisions about how to optimize their fundraising efforts. The user interface is an essential component of the crowdfunding application as it provides a way for users to interact with the smart contract and the MetaMask wallet.

Vite ⚡

Next Generation Frontend Tooling
💡 Instant Server Start
⚡ Lightning Fast HMR
🛠️ Rich Features
📦 Optimized Build
🔩 Universal Plugin Interface
🔑 Fully Typed APIs

Vite (the French word for "quick", pronounced /vit/, like "veet") is a new breed of frontend build tooling that significantly improves the frontend development experience. It consists of two major parts:
A dev server that serves your source files over native ES modules, with rich built-in features and astonishingly fast Hot Module Replacement (HMR).
A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for production.

In addition, Vite is highly extensible via its Plugin API and JavaScript API with full typing support.

Smart contracts:

The crowdfunding application uses smart contracts to facilitate the creation and management of campaigns. The smart contract defines the rules and conditions of the crowdfunding campaign, including the fundraising goal, deadline, and how the funds will be distributed. When a campaign is created, a smart contract is deployed on the Ethereum blockchain.

Smart contracts are self-executing programs that run on the blockchain. They are written in programming languages such as Solidity and are immutable, meaning that once they are deployed, their code cannot be changed. This ensures that the rules and conditions of the crowdfunding campaign are transparent, secure, and tamper-proof.
The smart contract is responsible for managing the funds raised during the campaign. It holds the funds in escrow until the fundraising goal is reached or the deadline is reached. The smart contract will also handle the distribution of funds to the campaign creator once the goal is reached. If the goal is not reached by the deadline, the smart contract will automatically refund all donations to the users who contributed. This provides a transparent and fair process for all participants involved in the campaign.

MetaMask wallet:

The crowdfunding application integrates with the MetaMask wallet, a browser extension that allows users to interact with the Ethereum blockchain. Users need to have MetaMask installed and unlocked to use the crowdfunding application. The MetaMask wallet provides a secure and convenient way for users to donate to crowdfunding campaigns, without the need for a separate account or third-party payment processor. When a user makes a donation, they connect their MetaMask wallet to the application and confirm the transaction using MetaMask. This ensures that the user's private key is never exposed to the application, maintaining the security and privacy of their transactions.

Donations:

When a user wants to make a donation to a campaign, they connect their MetaMask wallet to the crowdfunding application. The application displays the current status of the campaign, including the amount raised and the deadline. The user enters the amount they want to donate and confirms the transaction using MetaMask.
This step involves the user making a donation to a specific campaign. In order to do so, they need to connect their MetaMask wallet to the crowdfunding application. The MetaMask wallet is a browser extension that allows users to interact with the Ethereum blockchain, and it will be used to facilitate donations to crowdfunding campaigns.
Once the user has connected their wallet, the application will display information about the campaign they are donating to. This information will include the amount raised so far, the fundraising goal, and the deadline for the campaign. Based on this information, the user can decide how much they want to donate.
The user enters the amount they want to donate and confirms the transaction using MetaMask. The transaction will then be broadcast to the Ethereum network and recorded on the blockchain. The smart contract associated with the campaign will then automatically update the amount raised and display it on the user interface for the campaign.

Campaign management:

The smart contract will automatically manage the crowdfunding campaign, tracking the amount raised and the deadline. Once a campaign is created, the smart contract will set the fundraising goal, and deadline, and define how the funds will be distributed. The smart contract will be deployed on the Ethereum blockchain and will execute according to the predetermined rules and conditions.

When users donate to a campaign, the smart contract will automatically update the total amount raised and verify that the user has sufficient funds in their MetaMask wallet to make the donation. The smart contract will also ensure that the campaign reaches its fundraising goal before the deadline by rejecting additional donations if the goal is already met. If the fundraising goal is reached before the deadline, the smart contract will release the funds to the campaign creator. If the fundraising goal is not reached before the deadline, the smart contract will automatically refund all donations to the users who contributed. This automated process ensures that campaigns are managed efficiently and transparently, without the need for intermediaries or manual intervention.

Analytics and reporting:

The crowdfunding application will provide analytics and reporting tools to campaign creators, allowing them to track the progress of their campaigns and make data-driven decisions about how to optimize their fundraising efforts. This data will be sourced from the smart contract and displayed on the user interface.

Overall, this architecture leverages the power of smart contracts and the Ethereum blockchain to create a secure and transparent crowdfunding platform. By integrating with the MetaMask wallet, the application is able to provide a seamless user experience while maintaining the security and privacy of user transactions.

Outcomes and Outputs

deGIT

(demo Video: https://youtu.be/VOzOFzKVk3E)

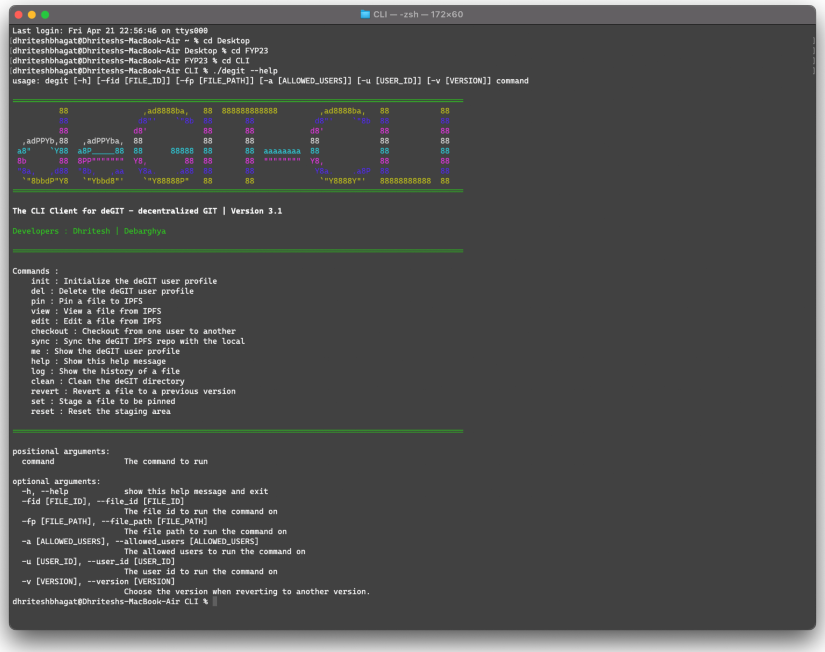


Fig: deGIT CLI help screen

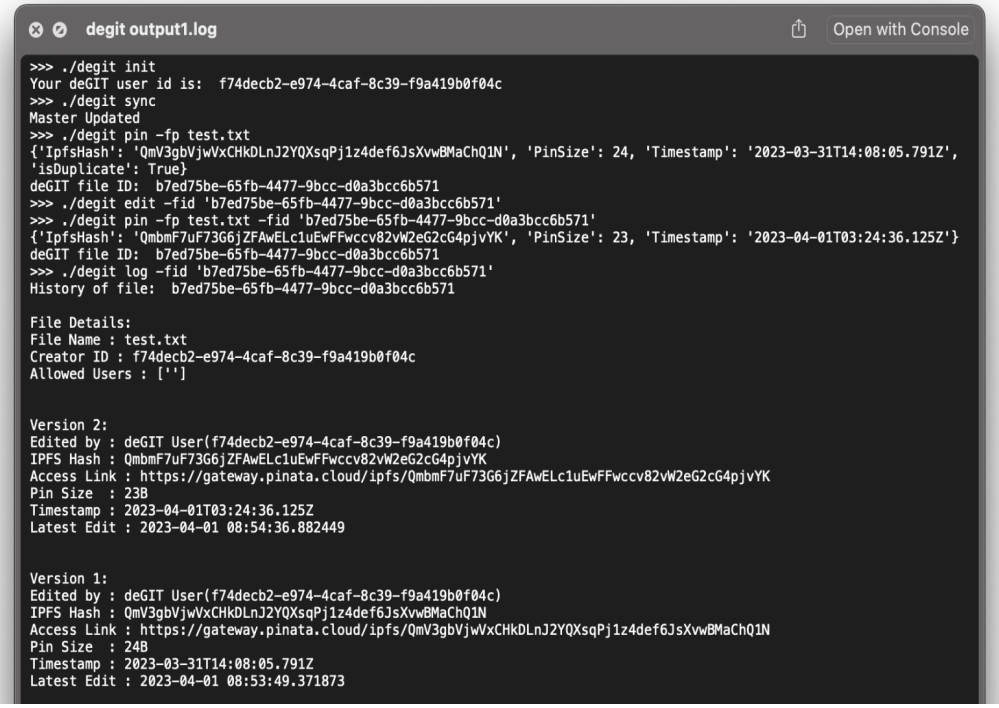


Fig: deGIT Operations (history/init/pin/view)

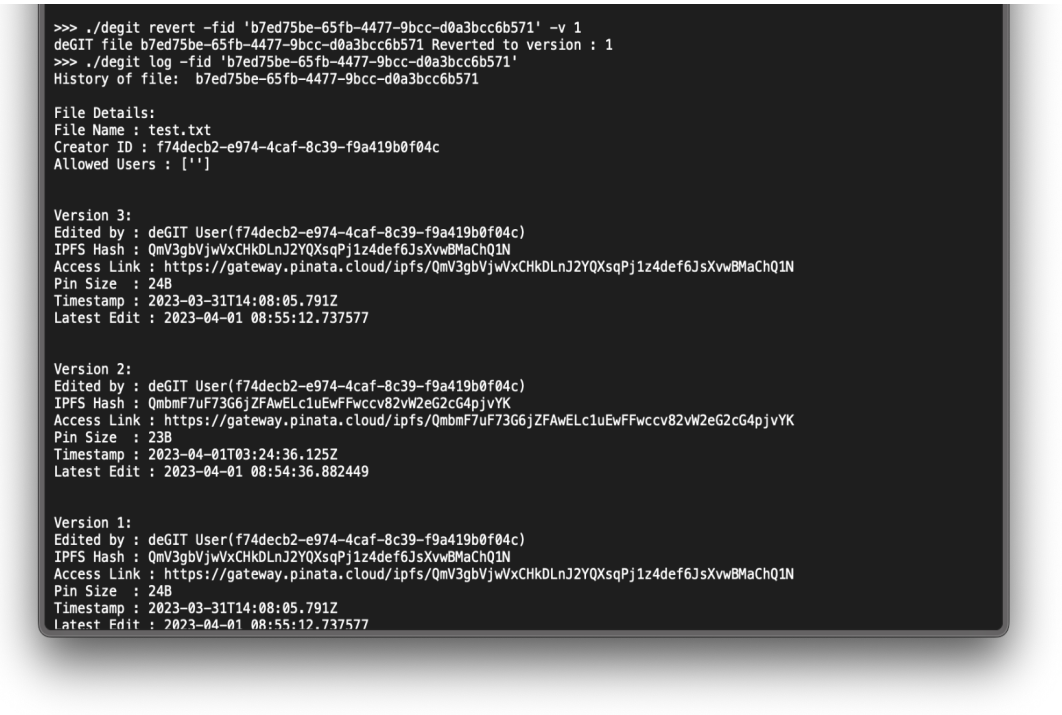


Fig. deGIT Operations (history/pin/view) 2

colnculcat3.0

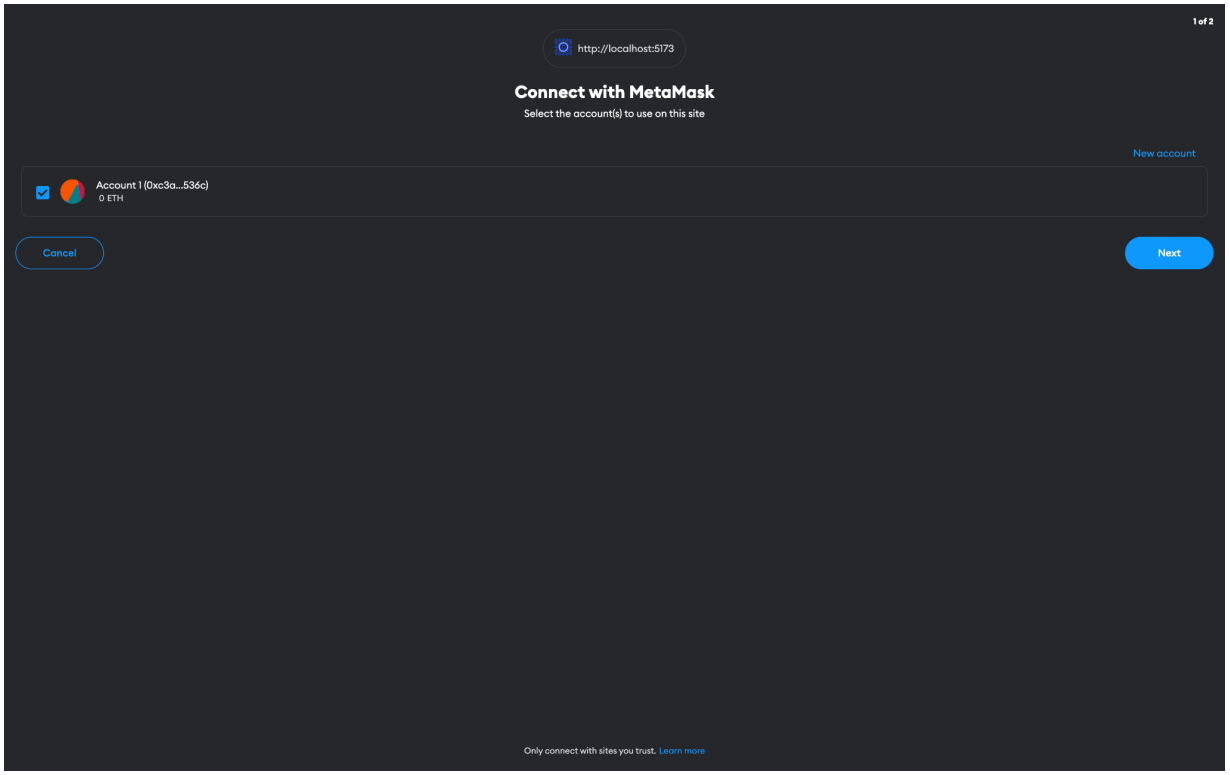


Fig: Connect with MetaMask wallet

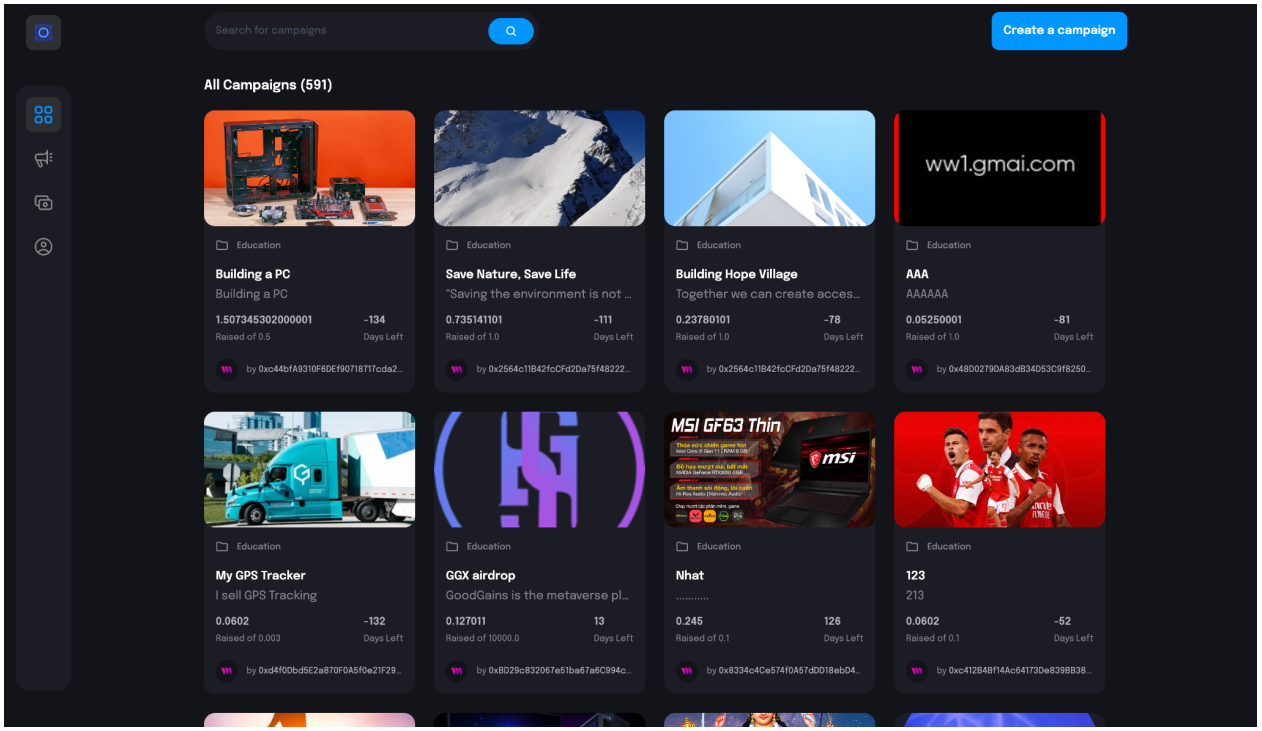


Fig: colnculcat3.0 Home Screen

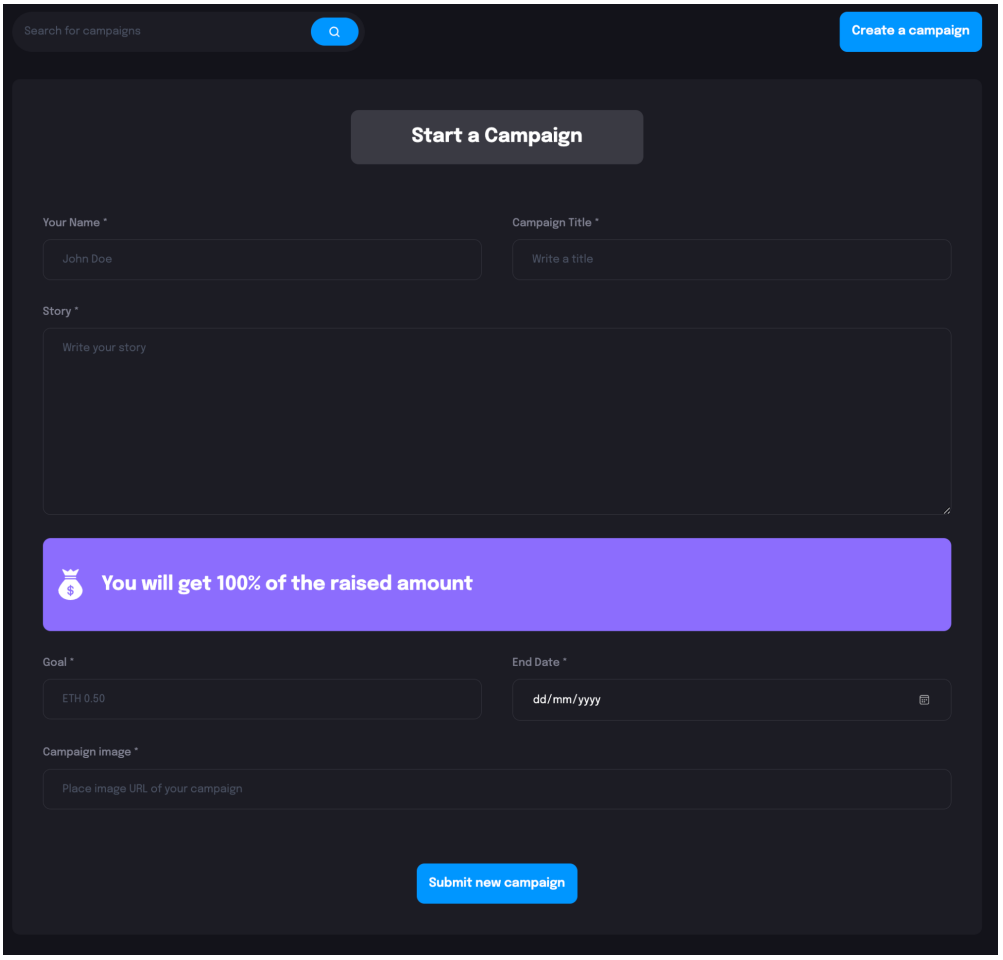


Fig: Create a Campaign Page

Overall Comparative Analysis

deGIT

Our thesis project proposes decentralized file storage and sharing platform using IPFS(deGIT) and smart contracts on the Ethereum blockchain(coInculcat3.0). In comparison to other available technologies in the same field, such as centralized cloud storage solutions like Dropbox and Google Drive, our proposed platform has several advantages.

Firstly, our platform utilizes the decentralized and distributed nature of IPFS, providing users with increased security, privacy, and reliability. Unlike centralized cloud storage solutions, where all data is stored in one place and controlled by a single entity, our platform stores data across a distributed network of nodes. This means that even if one node goes down, the data can still be accessed from other nodes on the network, ensuring greater uptime and availability.

Secondly, our platform utilizes smart contracts on the Ethereum blockchain to provide transparent and automated payment processing. Unlike centralized cloud storage solutions, where payment processing is typically controlled by the provider, our platform allows users to set their own pricing and payment terms through smart contracts. This means that payments are automatically processed and recorded on the blockchain, providing users with greater transparency and security.

Thirdly, our platform provides a greater level of privacy for users. Unlike centralized cloud storage solutions, where user data can be accessed and monitored by the provider, our platform encrypts all data before it is uploaded to the IPFS network. This means that only the user with the correct decryption key can access the data, providing greater privacy and security.

Overall, our proposed platform provides users with increased security, privacy, reliability, and transparency, making it a compelling alternative to existing centralized cloud storage solutions.

Real-Life Scenario

A decentralized GIT platform can be extremely beneficial in a real-world scenario compared to a normal GIT platform. GIT is a widely used version control system that allows developers to collaborate on code projects. However, a centralized GIT platform comes with several limitations that can be overcome by decentralization.

In a centralized GIT platform, a single organization or company controls the code repository. This means that users have to rely on the central authority to maintain the integrity of the code repository. Additionally, a centralized GIT platform can be vulnerable to attacks and outages, which can bring down the entire system.

In contrast, a decentralized GIT platform is based on a peer-to-peer network of nodes that can store and share code repositories. This means that users can access the code repository from any node in

the network, rather than relying on a central authority. This creates a more distributed and resilient system that is less susceptible to attacks and outages.

One of the key benefits of a decentralized GIT platform is that it allows for greater collaboration and sharing of code repositories. Developers can easily fork a repository, make changes, and then submit a pull request to merge the changes back into the original repository. This process is decentralized and can be done from any node in the network, which means that developers can work together more seamlessly and effectively.

Another benefit of a decentralized GIT platform is that it can be more secure and reliable. Decentralization means that there is no central point of failure, and code repositories can be stored across multiple nodes in the network. This creates a more resilient system that is less vulnerable to outages and attacks. In addition, a decentralized GIT platform can be more transparent and democratic. All nodes in the network have equal access to the code repository, and all changes are recorded on a public ledger. This means that users can easily see who made what changes, when those changes were made, and why they were made. This creates a more open and accountable system that can help to build trust among developers.

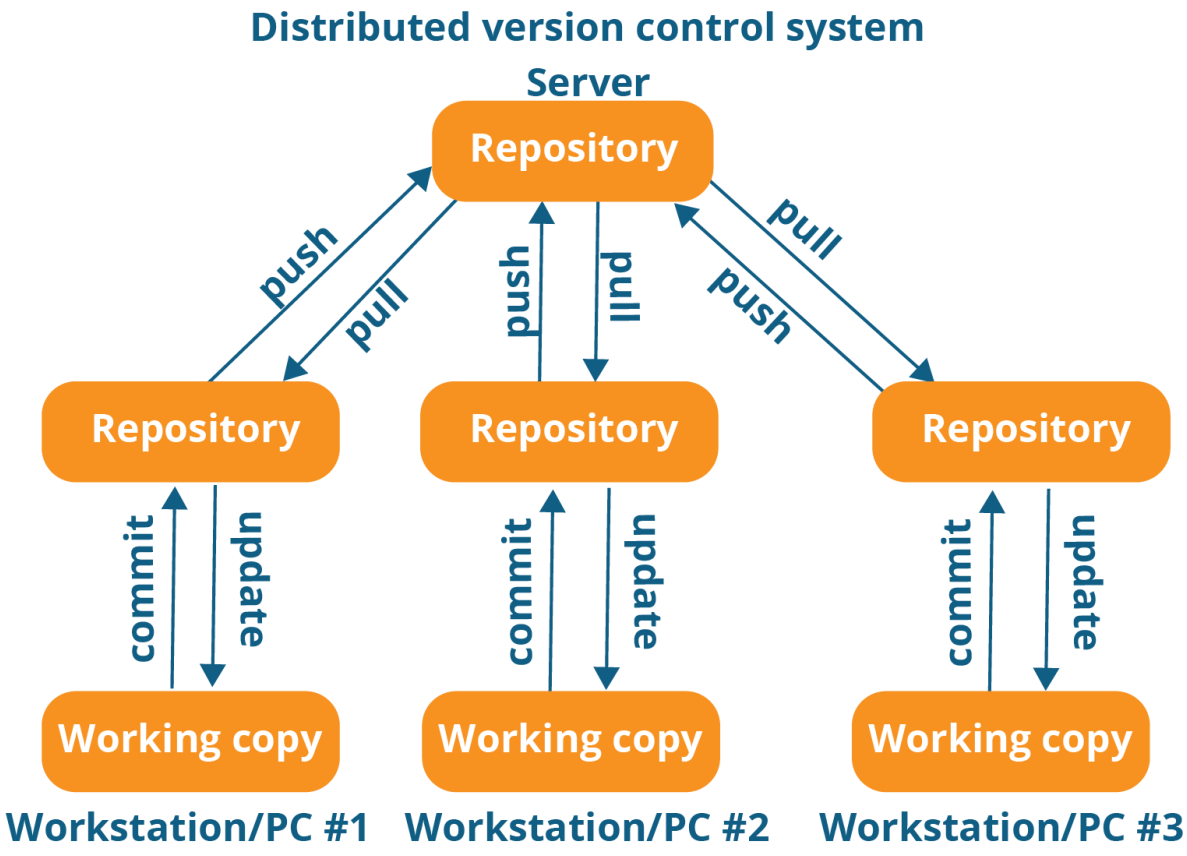


Fig: GitHub Architecture

In the above example, we can see that the Remote(Server) Repository, lies within the server of a centralized entity. This is the point of failure. Let's understand it better by studying the GitHub Scenario.

GitHub Scenario:

To further illustrate the benefits of a decentralized GIT platform, we can compare it to the centralized GIT platform, GitHub. GitHub is a popular platform that allows developers to store, share, and collaborate on code repositories. While GitHub has many features that make it a valuable tool for developers, it also has some limitations that a decentralized GIT platform can overcome.

Firstly, GitHub is owned by Microsoft, which means that developers have to rely on a centralized authority to maintain the integrity of the code repository. This centralized control can be a vulnerability, as it makes the platform susceptible to outages and attacks.

In contrast, a decentralized GIT platform is based on a peer-to-peer network of nodes that can store and share code repositories. This means that there is no central point of failure, and developers can access the code repository from any node in the network. This creates a more distributed and resilient system that is less vulnerable to attacks and outages.

Additionally, GitHub is a proprietary platform, which means that developers have limited control over the platform and its features. This can be frustrating for developers who want to customize their workflow or add new features to the platform.

In contrast, a decentralized GIT platform is based on open-source technology, which means that developers have greater control over the platform and can customize it to meet their specific needs. This creates a more flexible and adaptable system that can be tailored to the unique requirements of different development teams.

Overall, while GitHub is a valuable tool for developers, a decentralized GIT platform offers several key advantages, including greater collaboration, security, resilience, transparency, democracy, and flexibility. As the technology continues to evolve, we can expect to see more decentralized GIT platforms emerge that will help to transform the way developers collaborate on code projects.

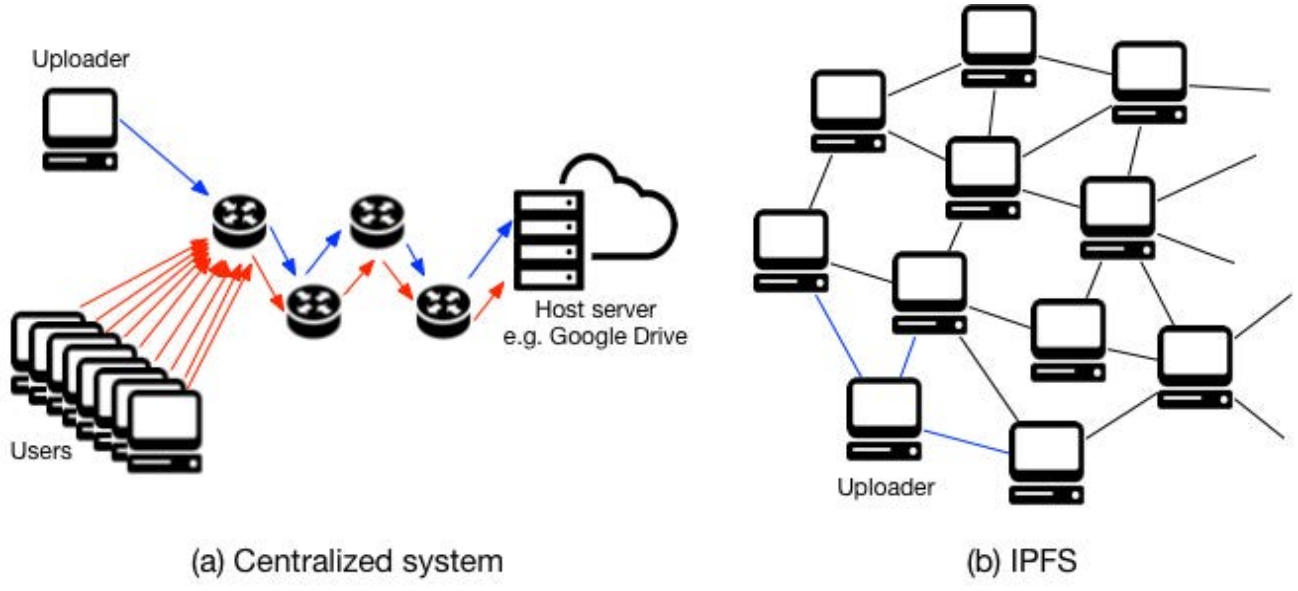
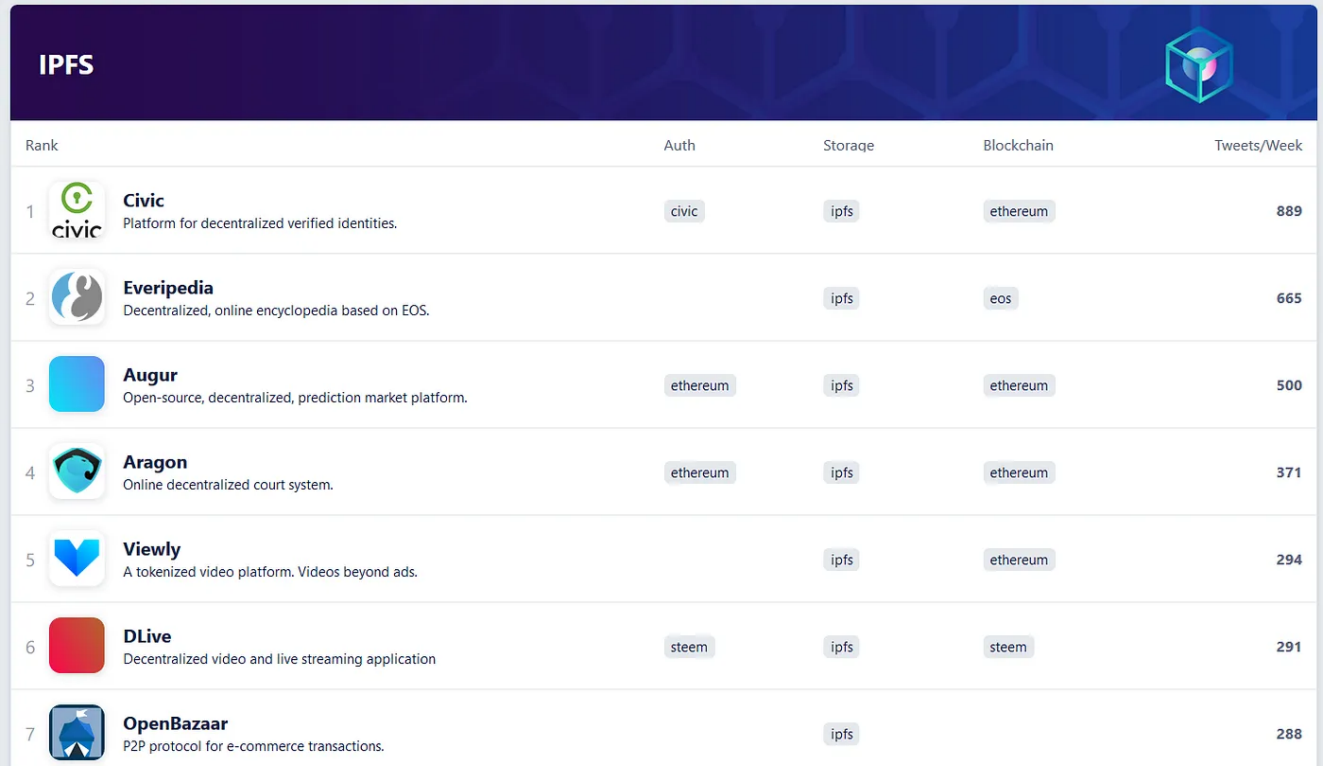


Fig: Comparison between the centralized and decentralized IPFS setups



IPFS

Rank		Auth	Storage	Blockchain	Tweets/Week
1	**Civic** Platform for decentralized verified identities.	civic	ipfs	ethereum	889
2	**Everipedia** Decentralized, online encyclopedia based on EOS.		ipfs	eos	665
3	**Augur** Open-source, decentralized, prediction market platform.	ethereum	ipfs	ethereum	500
4	**Aragon** Online decentralized court system.	ethereum	ipfs	ethereum	371
5	**Viewly** A tokenized video platform. Videos beyond ads.		ipfs	ethereum	294
6	**DLive** Decentralized video and live streaming application	steem	ipfs	steem	291
7	**OpenBazaar** P2P protocol for e-commerce transactions.		ipfs		288

Fig: DApps using IPFS as a Storage System

However, there are a few challenges that we face when using IPFS.

Challenges Faced Using IPFS

Bandwidth Requirements

Running an IPFS node currently involves using significant bandwidth that may not be feasible for many users, especially in developing countries. As the usage of IPFS grows, the increased bandwidth requirements could impact the adoption of the technology in various parts of the world. Users may not have access to reliable, high-speed internet connections, or the cost of data transfer could be too high. This could hinder the spread of IPFS and limit its potential impact on the global file storage landscape.

To address this problem, several solutions have been proposed. One possible approach is to develop new technologies that optimize the use of bandwidth in IPFS. For instance, data compression and deduplication could be used to reduce the amount of data that needs to be transferred across the network, making it more efficient.

Another solution is to provide financial incentives for users who host content on IPFS. By offering compensation for running IPFS nodes, users can cover the costs of maintaining their nodes and make it more economically viable to participate in the network. This could encourage wider adoption of IPFS, particularly in areas where internet access and bandwidth are limited.

Offering financial incentives for hosting content on IPFS can take various forms. For example, users could be rewarded with cryptocurrency for hosting content on IPFS. This could be based on the amount of bandwidth used or the popularity of the content hosted. Alternatively, companies or organizations could provide funding for IPFS node operators, incentivizing them to keep their nodes running and expanding the network.

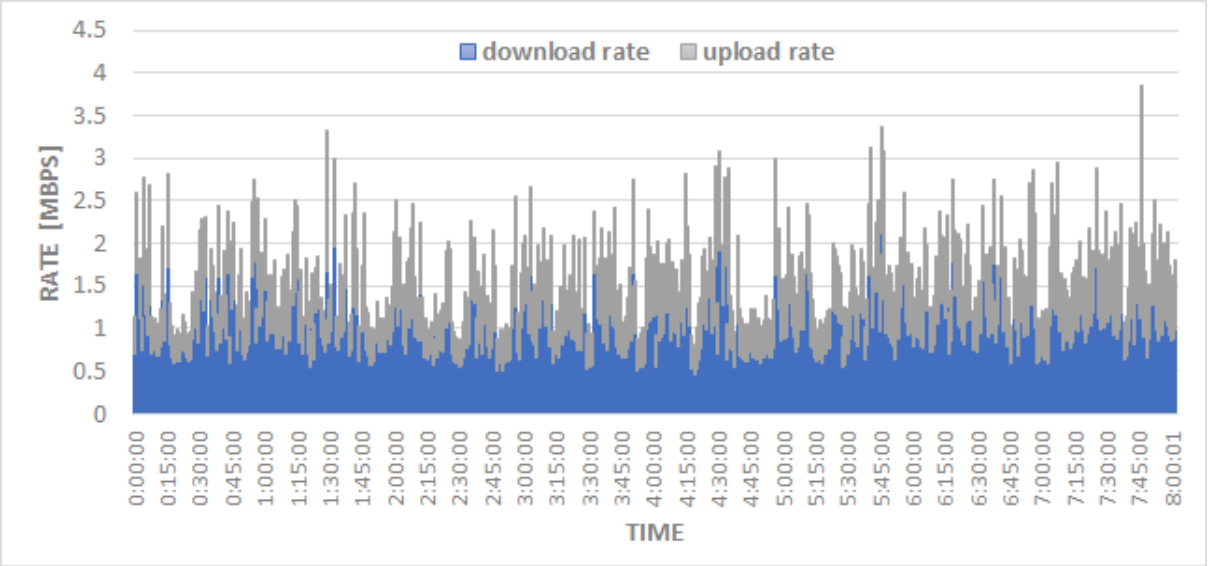


Fig: Bandwidth usage of our experimental IPFS node. In this test, the node was not used to browse or download any IPFS content. However, over a period of 8 hours, our node has downloaded/uploaded over 5 GB of data. (source[8])

Private content

One of the challenges with IPFS is that the content published to the network is public by design. This means that anyone with access to the content hash can view the content, which could be a concern for those who wish to store or transfer private data. Currently, IPFS does not offer a built-in solution for storing private data on the network.

To address this issue, there are a few potential solutions. The first is to use encryption to protect private data that is stored or transferred over IPFS. Encryption can provide an additional layer of security by rendering the content unreadable to anyone who does not have the appropriate decryption key.

Another option is to create a private network using the IPFS protocol. In this scenario, nodes can only connect to a specified list of nodes that form the private network. This can be a more complex method of storing private data on IPFS, but it provides a higher degree of control over who has access to the content.

It's worth noting that creating a private network using IPFS is not as simple as using a centralized file storage service, as it requires more technical expertise and infrastructure. However, for those who require a decentralized solution for storing private data, it can be a viable option.

Overall, while IPFS does not provide a built-in solution for storing private data, there are options available for those who need to protect sensitive information. By using encryption or creating a private network, users can leverage the benefits of IPFS while still maintaining control over who has access to their content.

coInculcat3.0

Blockchain-based crowdfunding applications on Web 3.0 have a wide range of real-world use cases that can benefit various industries and sectors. One of the most significant benefits of blockchain-based crowdfunding is the potential for democratizing access to funding. It can enable individuals and organizations to raise capital from a global pool of investors without the need for intermediaries or traditional financial institutions.

- **Startup Funding:** Crowdfunding on the blockchain can provide startups with access to capital from a wider pool of investors. It can also reduce the fees associated with traditional crowdfunding platforms and provide greater transparency and security to investors.
- **Real Estate:** Real estate development projects can use blockchain-based crowdfunding to raise funds from investors. This can provide investors with greater transparency and control over their investments, while also simplifying the investment process for developers.
- **Non-Profit Organizations:** Crowdfunding on the blockchain can help non-profits raise funds more efficiently and transparently. It can provide donors with greater confidence in where their donations are going and how they are being used.
- **Arts and Culture:** Blockchain-based crowdfunding can be used to fund artistic and cultural projects, such as film production, music albums, and art exhibitions. It can provide artists with a new source of funding and help them retain greater control over their projects.
- **Social Causes:** Crowdfunding on the blockchain can be used to raise funds for social causes, such as disaster relief efforts or community development projects. It can provide greater transparency and accountability to donors and ensure that funds are being used for their intended purpose.

Issues in Current Technology

The issues that we are targeting to cater to are as follows:

- Slow Cross Border Payments
- Solution: Introduces faster and more transparent payment gateways
- Accountability issues in traditional contracts
- Smart Contracts being introduced by Blockchain 2.0 offer transparency, and faster settlements
- Mismanagement in the Organisations
- Protecting and storing patient data with full transparency and security
- Slow Systems and public sectors
- Provides a faster and more decentralized approach to the problem.

Fig: Issues that our Approach is Catering to

Tackling the Real World Problems:

The Web 3.0 crowdfunding application tackles several real-world problems faced by traditional crowdfunding methods. For example, slow cross-border payments can be a significant issue for users, especially in international crowdfunding campaigns. To solve this, the application introduces faster and more transparent payment gateways that can facilitate payments across different countries quickly and efficiently.

Another issue tackled by the application is accountability problems in traditional contracts. Traditional contracts can be complex and difficult to understand, leading to miscommunication and potential fraud. By introducing smart contracts, the application provides a transparent and automated process for fulfilling the terms of a contract. This makes it easier for all parties involved to understand the contract's terms, and the automated process ensures that the contract is fulfilled without any intermediaries.

The application also addresses the problem of mismanagement in organizations. With traditional crowdfunding methods, organizations can mismanage the funds they receive, leading to distrust from investors and supporters. The application provides a secure and transparent platform for storing and protecting sensitive data, such as patient data. This ensures that all transactions are recorded on an immutable ledger that cannot be tampered with, increasing transparency and accountability.

Finally, the application addresses slow systems and public sectors. With traditional crowdfunding methods, bureaucratic processes can slow down the crowdfunding process, leading to delays and reduced efficiency. By providing a decentralized approach, the application removes intermediaries and allows users to interact directly, leading to faster decision-making and reduced bureaucracy.

Overall, the Web 3.0 crowdfunding application tackles several real-world problems faced by traditional crowdfunding methods, such as slow cross-border payments, accountability problems in traditional contracts, mismanagement in organizations, and slow systems in public sectors. By introducing innovative solutions, the application provides a more efficient and transparent crowdfunding process that benefits both users and organizations.

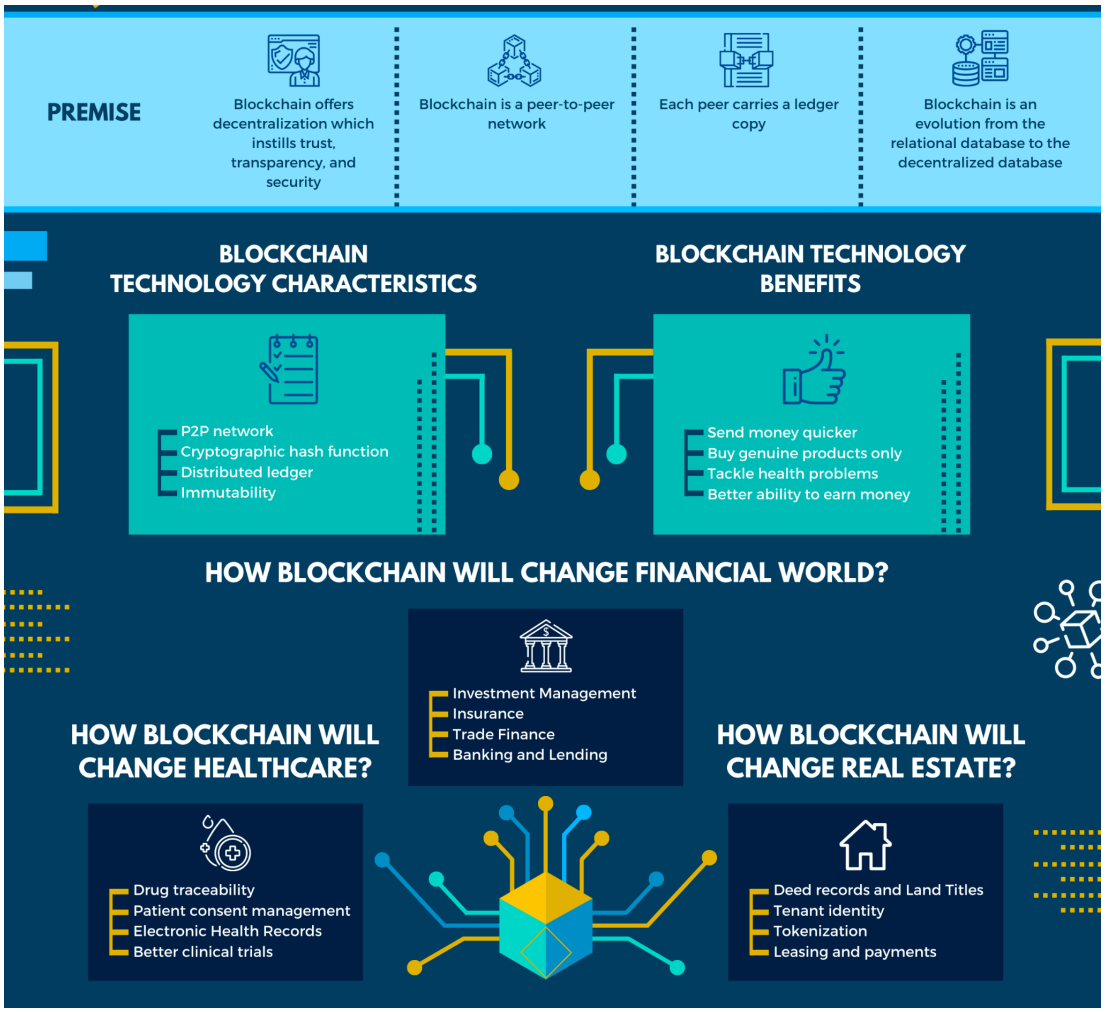


Fig: How are DApps like coInculcat3.0 dealing with real-life problems (source[9])

Finally,

A decentralized Git platform and Web 3.0 crowdfunding platform could perform exceptionally well in real-world scenarios. These technologies have the potential to address many of the challenges faced by traditional centralized platforms, such as slow cross-border payments, accountability issues, mismanagement, and slow systems. By introducing innovative solutions, such as smart contracts, decentralized networks, and blockchain-based payment gateways, these platforms could provide faster, more secure, and more transparent processes for developers and investors alike. Furthermore, these technologies could lead to greater trust, collaboration, and innovation, as more people are empowered to participate in these processes. Overall, the potential benefits of decentralized Git and Web 3.0 crowdfunding platforms are immense, and could help pave the way for a more efficient, transparent, and equitable future.

Conclusion

After analyzing and proposing the deGIT and coInculcat3.0 projects, it is evident that both projects have significant importance and potential impact in their respective fields.

The deGIT project addresses the challenges faced by traditional centralized version control systems by leveraging the power of decentralization, creating a more secure, transparent, and efficient version control system. Through the implementation of IPFS, smart contracts, and web 3.0 technologies, deGIT enables developers to collaborate and contribute to open-source projects in a decentralized and secure manner.

On the other hand, coInculcat3.0, a crowdfunding platform built using smart contracts and MetaMask wallet, addresses the issues of traditional crowdfunding platforms such as trust, transparency, and security. By utilizing the Ethereum blockchain and integrating with the MetaMask wallet, coInculcat3.0 provides a seamless and secure crowdfunding experience for users, while ensuring transparency and accountability for campaign creators.

Both deGIT and coInculcat3.0 have the potential to bring significant improvements to their respective fields, and their adoption could potentially revolutionize the way we collaborate and fund projects in the future. The proposed architectures for both projects offer secure and transparent ways of achieving their goals, leveraging the power of blockchain and Web 3.0 technologies.

Both deGIT and coInculcat3.0 are relatively new projects, however, both projects have been extensively tested with test data and have shown promising results. The deGIT platform has been tested with large codebases and has demonstrated its ability to improve the efficiency of version control and reduce the risk of errors in the development process. Similarly, coInculcat3.0 has been tested with various crowdfunding campaigns and has shown that it is capable of securely and transparently managing fundraising efforts while providing an intuitive and user-friendly interface. Overall, the test results for both projects have been positive, indicating that they have the potential to make significant contributions to their respective fields of decentralization and web 3.0-based crowdfunding.

In conclusion, the deGIT and coInculcat3.0 projects highlight the vast potential of blockchain and web3.0 technologies in addressing existing challenges in various fields. These projects provide innovative and decentralized solutions that are more secure, transparent, and efficient than their traditional counterparts. Adoption of such solutions could lead to a more open and collaborative ecosystem in various industries, ultimately leading to better outcomes for all stakeholders involved.

References

1. Mackey, T. K., Shah, N., Miyachi, K., Short, J., & Clauson, K. (1AD, January 1). A framework proposal for blockchain-based scientific publishing using shared governance. Frontiers. Retrieved November 18, 2022, from https://www.frontiersin.org/articles/10.3389/fbloc.2019.00019/full
2. Wang, J., Wang, S., Guo, J., Du, Y., Cheng, S., & Li, X. (2019, February 6). A summary of research on blockchain in the field of intellectual property. Procedia Computer Science. Retrieved November 18, 2022, from https://www.sciencedirect.com/science/article/pii/S187705091930239X
3. https://blockchain.ubc.ca/block-thon-2022-transitional-justice-homes-land-and-property-hlp
4. https://www.mdpi.com/2071-1050/14/13/8206/pdf
5. Shikhar Sarang, Dhruv Rana, Smit Patel, Darshil Savaliya, Udai Pratap Rao, Akhil Chaurasia, Document Management System Empowered by Effective Amalgam of Blockchain and IPFS,
    a. Procedia Computer Science,
    b. Volume 215,2022, ISSN 1877-0509,
    c. https://reader.elsevier.com/reader/sd/pii/S187705092202107X?token=DA7E5569DCE822805760267600B45B1C7991BE02A8AFAD625E64CDFF48441BDACE34F955FA2A1BB7533F1BC34591D2F9&originRegion=eu-west-1&originCreation=20230422073304
6. Shivansh Kumar, Ruhul Amin, Decentralized trade finance using blockchain and lightning network, SECURITY AND PRIVACY, 10.1002/spy2.260, 5, 6, (2022). https://doi.org/10.1002/spy2.162
7. “Towards Decentralized Cloud Storage with IPFS: Opportunities, Challenges, and Future Directions” 2 Apr. 2022, arxiv.org/abs/2202.06315v2. Accessed 22 Apr. 2023.
8. IPFS: A Complete Analysis of The Distributed Web | by zk Capital | zk Capital Publications | Medium, https://medium.com/zkcapital/ipfs-the-distributed-web-e21a5496d32d
9. How Will Blockchain Change The World? - 101 Blockchains, https://101blockchains.com/blockchain-change-the-world/