Project Overview and Use Case

StorageChain is a decentralized, cloud storage, and file sharing platform that uses a network of nodes to store data in a distributed manner. Decentralized cloud storage disperses user data over geographically and statistically different nodes, IPFS encryption protects it from hackers, and there is no single point of failure, reducing risk of breach to effectively zero.

The **StorageChain Decentralized Storage Network** or (**DSN**) project has three main components: IPFS (peer-to-peer) file sharing technology, data "sharding" and encryption technology, and a blockchain based distributed ledger. StorageChain operates its own native blockchain (**STOR**), which enables its community to store, retrieve, and share files on the network. The StorageChain network is a decentralized peer to peer storage network cloud that is comprised of a group of storage nodes that store and retrieve files stored on the network. Storage Chain is built on **IPFS Technology**. IPFS (**InterPlanetary File System**) is a distributed file system that uses a peer-to-peer network to store and share files. The IPFS network is decentralized, meaning that it does not rely on a central server or authority to store and manage data. Instead, data is stored on multiple nodes (computers) in the network, and the nodes work together to manage and share the data.

In contrast, the blockchain component is a decentralized, distributed ledger that records transactions between multiple parties. StorageChain uses a network of computers to validate and record transactions in a secure and transparent way. While IPFS and blockchain technology both have decentralized elements, they are designed for different purposes but combine their strengths to enable fast, secure, and inexpensive benefits of the StorageChain technology.

Decentralized storage is one of the most natural and meaningful, community driven crypto/blockchain use cases in existence today.

IPFS Technology

STOR is built on InterPlanetary File System Technology (IPFS) which is a distributed file system that seeks to connect all computing devices with the same system of files. In other words, it is a peer-to-peer protocol for sharing files, similar to BitTorrent. It works by breaking files into smaller pieces and distributing them across a peer-to-peer network, so that any peer can access the complete file by retrieving its component pieces from multiple sources. The IPFS network is self-organizing and self-healing, which means that data is still accessible even if some parts of the network are down or unavailable. IPFS uses a content-addressable naming system, where each file is given a unique identifier based on its content, rather than its location, which makes it resistant to censorship and tampering.

IPFS breaks a file into smaller pieces, called "shards," and then distributes those shards with encryption across a network of computers, called "nodes." Each shard is given a unique identifier, called a cryptographic hash, which allows it to be retrieved from anywhere on the network. When a user wants to access a file on the IPFS network, their computer sends a request for the file to the network, and the nodes on the network work together to retrieve and reassemble the file for the user.

One of the main benefits of IPFS is that it allows for more efficient distribution of large files, since the file is retrieved from multiple sources rather than just one central server. This can also make the file more resistant to censorship and or hacking, as it is more difficult to block or gain access to a file that is distributed, sharded, and encrypted across a decentralized network. StorageChain can be used to store and share large files, such as videos or software, but it can be used for any type of file.

History of Decentralized Data Storage

Decentralized data storage refers to a system for storing and managing data in a distributed manner, rather than relying on a single, centralized server or location. This type of data storage has a long history, with the earliest decentralized systems dating back to the 1970s.

One of the earliest examples of decentralized data storage was the concept of peer-to-peer (P2P) networks, which were developed in the 1970s and 1980s. P2P networks allow users to share resources, such as computer processing power or storage space, directly with each other, rather than relying on a central server. This allows for greater redundancy and resilience, as the failure of any one node in the network does not necessarily lead to the failure of the entire system.

Another early example of decentralized data storage was the development of distributed file systems, which allowed users to access and share files over a network of computers. These systems were designed to be more reliable and scalable than traditional file systems, which rely on a single server to store and manage files.

In recent years, decentralized data storage has gained renewed interest with the rise of blockchain technology and decentralized applications (dApps). These systems use distributed ledger technology to store data in a decentralized manner, providing greater security and transparency.

Overall, decentralized data storage has a long history and has evolved over time to become a key component of modern computing systems. It offers several benefits, including increased reliability, scalability, and security, making it an important tool for storing and managing data in the digital age.

Current Landscape of Decentralized Data Storage

Some of the current players in the decentralized data storage space include:

1. Filecoin: A decentralized storage network that uses blockchain technology and a native token (FIL) to enable users to buy and sell storage space.

2. STORJ: A decentralized cloud storage platform that uses a network of peer-to-peer nodes to store and manage data.

3. SiaCoin: A decentralized storage platform that uses a network of underutilized hard drive capacity to create a distributed data storage marketplace.

4. Flux: A decentralized data storage and communication platform that uses a network of nodes to store and manage data in a secure and private manner.

5. Arweave: A decentralized, permanent storage platform that uses a unique data structure called a blockweave to store data in a durable and tamper-proof manner.

The migration of cloud data storage from centralized servers housed by big technology companies like Google, Amazon, and IBM (web2) to decentralized data cloud storage operated by a decentralized network of storage and bandwidth providers (web3) is currently in its infancy. The figure below shows the landscape of web2 vs web3 cloud data storage providers.

Current Market Opportunity

Global Data Cloud Storage*

Total Addressable Market:

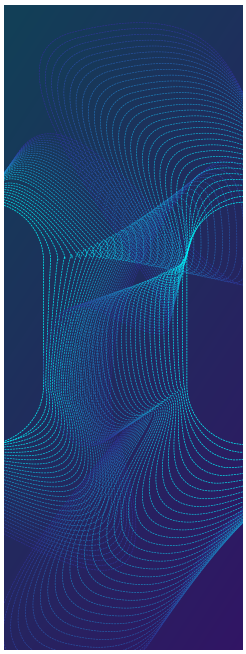
- 2022: $78B with a CAGR of 18.5% increasing to $183B by 2027

Serviceable Available Market:

- Individuals, SMB, and professional content creators.
- 2022 $40B with a CAGR of 20% increasing to $100B by 2027

Serviceable Obtainable Market:

- Early adopters, Influencers, entertainment industry
- Moving from Web2 to Web3
- 2022 $5B with a CAGR of 60% increasing to $53B by 2027



*MarketsandMarkets Research

StorageChain Value Proposition

Data is becoming more expensive to store and Big Technology companies own, manage and have access to their customers data and original content. Historically companies would run their own internal and proprietary data centers. The common belief was that handling their own data would never be trusted to a third party. That all changed at the advent of cloud data storage solutions. It was much less expensive and more secure to store data on the cloud, but the ownership and control of the data itself was the trade-off. Today data storage is evolving to the faster, more secure, and less expensive solution of decentralized data cloud solutions. The transition is currently in its early stages and the potential upside is enormous. Data breaches and hacks are commonplace with the big siloed centralized data storage providers.

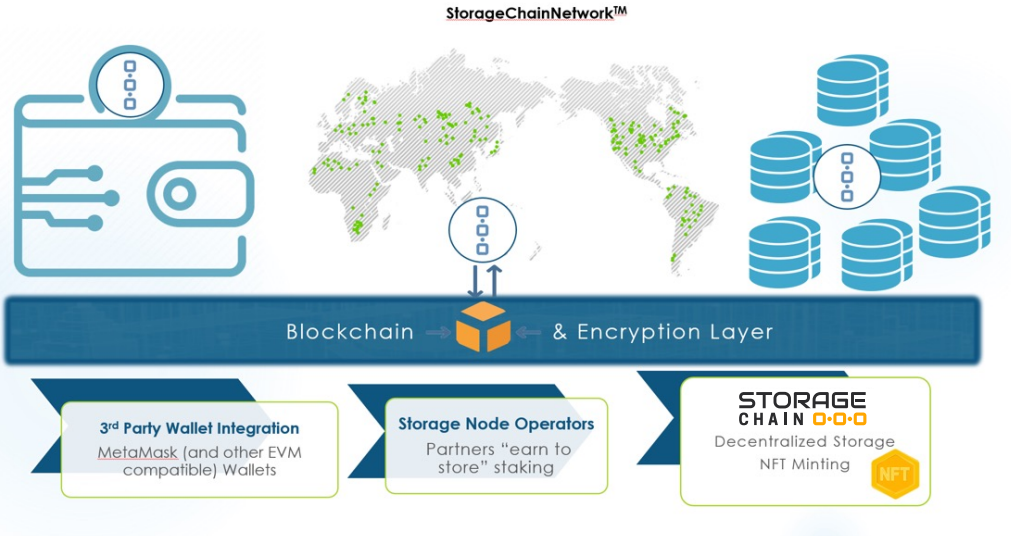
With StorageChain, customers get:

1. To own and manage their content and data
2. Ultra-secure, data “sharding” with encryption
3. 10x faster uploads
4. 75-80% less expensive than Web2
5. Ability to mint NFT’s to further monetize original content
6. Ability to transfer and share large media files easily and securely

StorageChain Storage Node Network customers get:

1. Ability to earn $STOR for providing storage space and bandwidth
2. Ability to earn $STOR for providing POS Validator nodes
3. Get involved early-on with a potential high growth company

StorageChain Ecosystem



Business Model

Our customer profile includes Individuals, SMB, and enterprise clients. Individuals and companies with large volumes of heavy media (video) files for private, fast, secure and inexpensive storage and retrieval are ideal for StorageChain.

Network node operators will earn coins for storage provided from storing and retrieving data from their hosted nodes. The payments will be made in STOR coins. Customers will have the ability to pay for the data storage services either by credit card (monthly recurring) or with STOR coins.

Monthly Unit Economics (75-80% less than Web2):

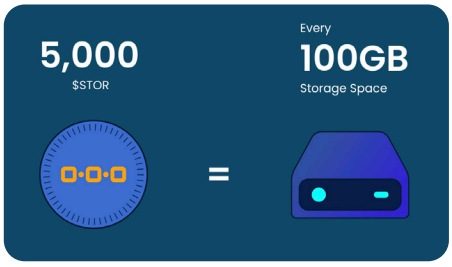
StorageChain WEB3		DropBox WEB2	
150GB Free Storage		2GB Free Storage	
Per user / month		Per user / month	
$2.50	Up to 2TB	$11.99	Up to 2TB
$5.00 No Minimum User	Up to 5TB	$18.00 3 Users Minimum	Up to 5TB
$.0045 per GB No Minimum User	Over 5TB	$30 3 Users Minimum	Up to 50TB

Comp Product Features

	STORAGE CHAIN	STORJ	Filecoin	Dropbox	iCloud
Up to 150 GB of Free Storage	●	○			
Mint NFTs for Original Content	●		○		
Sharding + Encryption Security Layer	●	○	○		
Decentralized Data Storage	●	○	○		
Earn Rewards for Participation	●	○	○		
Smart Contract Storage	●	○	○		
File Synch to Mobile or Desktop	●	○	○	○	○
Productivity Tools	●			○	○
File Sharing	●	○	○	○	○
Content Collaboration	●	○	○	○	○
Usage Based Payment Plans	●	○	○	○	○
Minimum Node Storage Requirements	100GB	550GB	600GB	N/A	N/A

Earning Rewards on the StorageChain Network

In the StorageChain network, node operators are responsible for storing and serving data to users in exchange for STOR coins. The amount of STOR that a node operator earns is determined by the amount of data they store and serve to users, as well as the amount of bandwidth and resources they contribute to the network. Storage node operators must set-up and host their nodes individually (or in a third party data center) and then add and manage them in the StorageChain Network through an easy, user-friendly web-based interface. Anyone can be a storage node operator, as long as the **minimum requirements of 100GB of available storage and 400GB of bandwidth** are met. Storage node operators can earn up to approximately 1,000 STOR coins per month per node (at 100GB+ of storage space and 95+% node uptime). There is a **5,000 STOR coin collateral requirement** for participating in the network.



Node operators can earn STOR coins in two primary ways: by participating in the storage and retrieval of data on the network, and by participating in the network's consensus process. The full percentage of earned rewards will require a minimum network node uptime, and proof of storage. Node operators can also earn STOR coins by participating in the network's POS consensus process, which is responsible for validating transactions and adding new blocks to the blockchain.

When a customer stores more than 150GB of data on the StorageChain network, customers are required to pay a monthly fee in USD or STOR coins.

The rewards for participating in the STOR network are designed to incentivize node operators to contribute their resources and help maintain the decentralized nature of the platform. By rewarding node operators for their contributions, the StorageChain network provides a secure, decentralized, and cost-effective alternative to traditional cloud storage solutions.

How to Set Up a Storage Node on the StorageChain Network

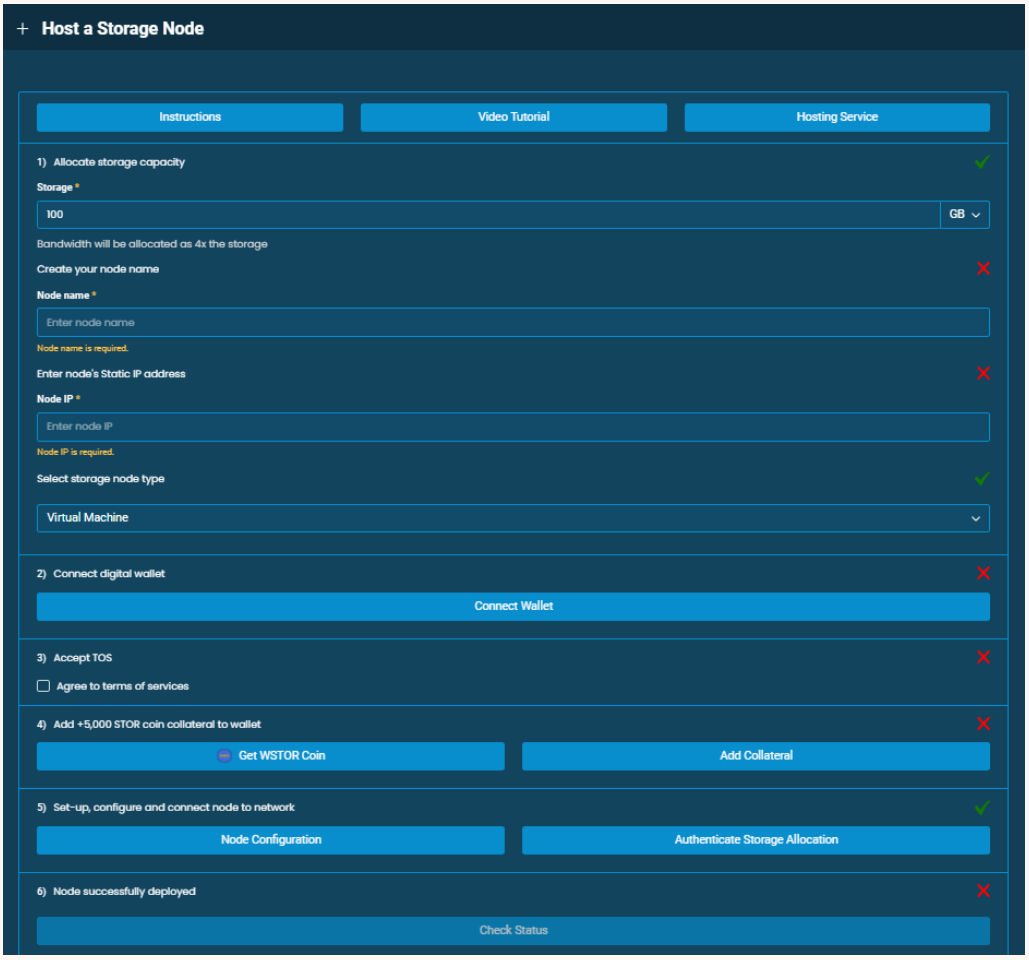
Create a StorageChain account at www.storagechain.io

Navigate to the “My Storage Nodes” tab and follow the set-up instructions. In order to earn rewards you must use the storage node to validate transactions or to store StorageChain user/customer data.

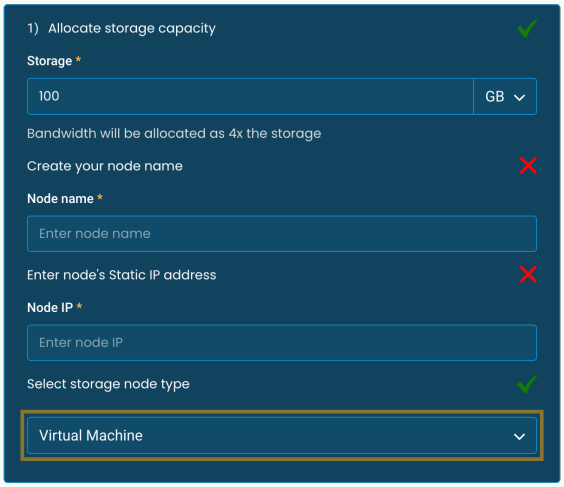
Anyone can be a storage node operator, as long as the minimum requirements of 100GB of available storage and 400GB of bandwidth are met. Node operators can earn STOR coins in two main ways: by participating in the storage and retrieval of data on the network, and by participating in the network’s consensus process.

Minimum System Requirements to Host a Node:

100GB of available storage is the minimum required.
400GB is the system default setting.
Bandwidth = **4X** available storage.
5,000 $STOR collateral per 100GB hosted



For Virtual Machine Users:



Step 1: Allocate Storage Capacity

- Sign in to StorageChain and Setup a Node
- Choose storage space allocation. Allocation will be verified in **Step 6**, so double check the storage capacity of your virtual machine first.
- Name your node (Operator can run multiple nodes).
- Enter the static IP address provided by Virtual hosting service.
- Choose Virtual (3rd party hosted) machine type.

Step 2: Connect Digital Wallet

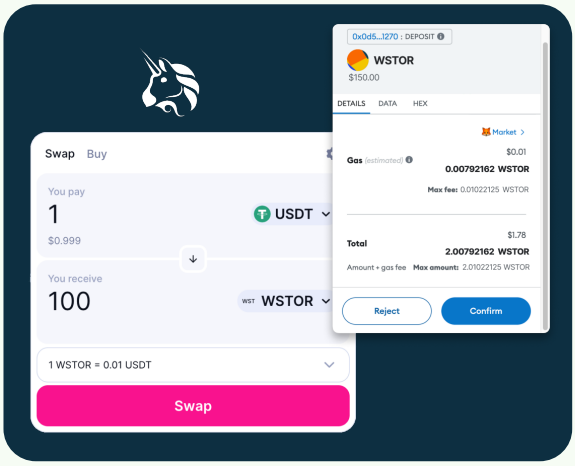
- Register for a StorageChain account if you haven't already!
- After login, simply click the "Wallet" icon on the top right nav or the "Connect Wallet" button.
- You can easily connect any EVM compatible wallet including MetaMask.
- Collateral and reward payments will now be enabled.

Step 3: Acquire WSTOR Collateral

- Acquire the WSTOR collateral from the Uniswap Exchange: https://app.uniswap.org
- Add WSTOR as a custom token on the Ethereum network in your MetaMask wallet: **Click token > import token > add custom token**

ENTER TOKEN CONTRACT ADDRESS:

```
0x50B275a15E4F5004AA96F972a30E6A9C718b203f
```



ENTER TOTAL DECIMALS: 18

- You will need a minimum of 5,000 WSTOR Tokens for every 100GB of storage provided. WSTOR (an ERC-20 Token) is wrapped STOR to enable trading on the Ethereum network. Please note, gas fees for the swapping will be much lower after 5PM Pacific standard time.
- Once the WSTOR is acquired you will need to unwrap the WSTOR Tokens (Ethereum Mainnet) to STOR Coins (StorageChain Mainnet): https://wrap.storagechain.io
- You now have the collateral in STOR Coin to set-up a node.
- All rewards are paid in STOR Coin. To swap your rewards from STOR to WSTOR go to https://wrap.storagechain.io and select wrap STOR to WSTOR. You will see your WSTOR under the Tokens section of the MetaMask wallet when connected to the Ethereum Mainnet network.

WSTOR can be swapped with any asset available on the Ethereum network.

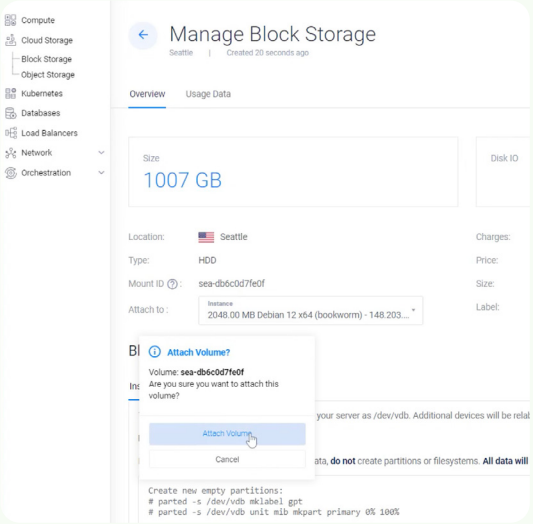
- When a node is deleted by the operator, the collateral will be returned to the connected wallet.

Step 4: Setup Virtual Server, and Mount Block Storage

We can work with any virtual hosting service of your choice, but the set-up may be slightly different for each provider. First, you will need to create a virtual machine instance and mount block storage volume to create a StorageChain storage node. The instance should have **4 GB of RAM and 2 CPU cores**, which are the minimum recommended requirements. The OS (operating system) for the virtual machine should be Linux; we recommend either **Debian, Fedora or Ubuntu** as excellent options.

Here is an example utilizing the hosting service on **Vultr.com** to guide you through the process of setting up your own virtual machine instance with block storage.

Register on VULTR, then create an instance (a virtual machine).



You will need to acquire block storage for the volume, and mount that volume to the virtual machine. Please note that the block storage volume should be in the same region as the virtual server. Make sure you attach the block storage to the virtual machine instance.

You will need **4x the bandwidth** of the storage space selected.

Step 5: Configure Your Virtual Node

Start by opening a **Windows PowerShell** Prompt or Open a **Terminal on Mac**. Hint: Search for Windows PowerShell from the Windows search submit box in the system tray. A virtual machine can be managed through the Terminal or PowerShell only. There is no other UI.

You can login to your machine without a password by setting up an SSH key. Make sure to add the SSH Key by generating it in your local machine. You can do so by following this link or you can login with the machine IP and password.

Now, login to your virtual machine by using the following commands in PowerShell or Terminal.

```
PowerShell (Windows) or Terminal (Mac)

1. PS C:\Users\user_name> ssh root@Paste your Virtual Machine IP address and press enter
```

Assuming you have successfully completed **Step 4** and mounted the block storage to the virtual instance, the next step is to attach the storage mount. No need to worry about the instance names; we can copy and paste the given commands below in green:

```
2. sudo apt update
3. sudo apt install nodejs npm
   Type “yes”  or “y” if terminal wants your consent
4. sudo npm install -g pm2
5. Create new empty partitions:
   # parted -s /dev/vdb mklabel gpt
   # parted -s /dev/vdb/ unit mib mkpart primary 0% 100%
   Create new empty filesystem:
   # mkfs.ext4 /dev/vdb1
```

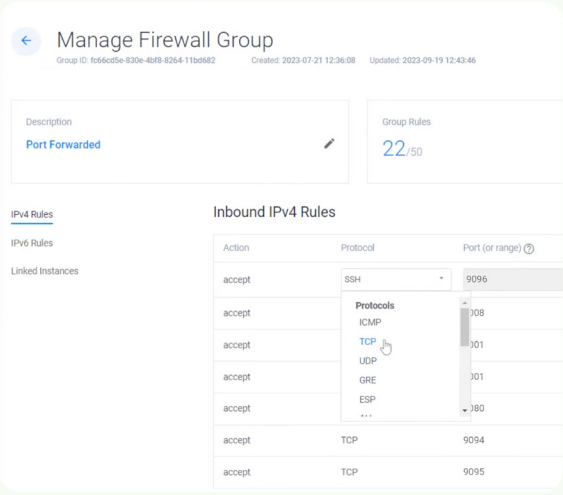
We’re using the “**ext4**” file system for this purpose. Linux primarily supports ext4 and a few other file systems, while Windows uses NTFS and macOS uses HFS or ext4.

As soon as the process has been executed, we need to wait for these blocks to load without interruption. Once completed, we need to mount the storage using the following commands in green below:

```
6. Mount block storage:
   # mkdir /mnt/blockstorage
   # echo >> /etc/fstab
   # echo >> /dev/vba1 /mnt/blockstorage ext4 defaults,noatime,nofail 0,0 /etc/fstab
   # mount /mnt/blockstorage
```

Step 6: Setup Port Forwarding

The port forwarding set-up process may be different for each virtual machine or a cloud-based data center, so you may have to consult your provider. In this example, we are using the process for Vultr.com as an example.



Navigate to Server Information page:
Settings > Firewall > Manage > Select Your Firewall ID.

Ports to forward:

3008 4001 5001 8080
9094 9095 9096

Make sure to choose the TCP option under Protocols and forward all 6 ports on both **IPv4 Rules** and **IPv6 Rules**.

Step 7: Verify Your Virtual Node

Check the storage using the given commands below in green to verify the mounted storage.

First, connect to all the ports and download the zip file onto your virtual server.

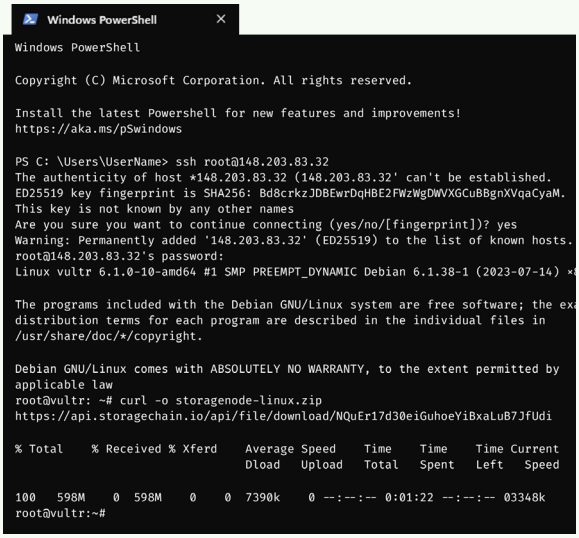
PowerShell (Windows) or Terminal (Mac)

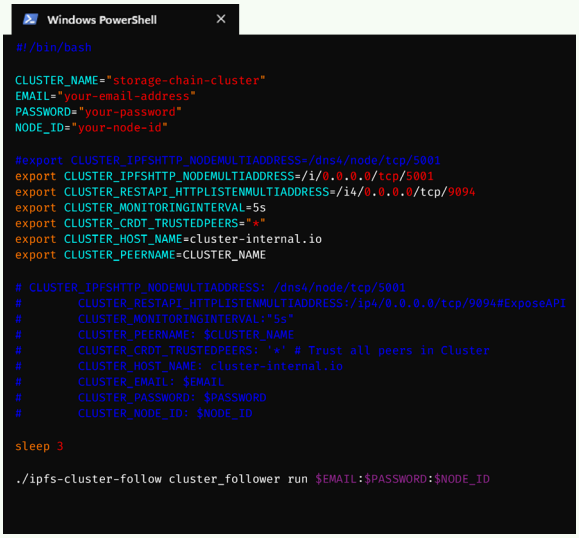
1. df -h
2. sudo ufw allow 3008,4001,5001,8080,9094,9095,9096/tcp
3. curl -o storagenode-linux.zip https://api.storagechain.io/api/file/download/alCb6R27qgbiTtU2771CNfu2m65pAQXe

When zip is downloaded, unzip the file by using the commands in green below:

1. root@vultr:~# unzip ./storagenode-linux.zip -d ./storagenode-linux
2. root@vultr:~# cd storagenode-linux/
3. root@vultr:~/storagenode-linux# vi startup.sh

Once unzipped, update the node credentials by typing the letter "I" at first and using the arrow key to move your cursor up and down to update your information.





When completed press ESC

Hods SHIFT and type a colon :

Type wq!

Type ./main.sh

NFT Minting

StorageChain gives users the ability to mint NFT's for any file types that are "mintable" which includes: JPG, PNG, GIF, FVG, MP4, WEBM, MP3 and WAV files.

A grayed-out NFT icon will appear next to a file that has been uploaded to the system, and is "mintable". A user may elect to click on that icon to mint an NFT. Once the single click to mint process is completed a user may move the newly minted NFT into any EVM compatible digital wallet including MetaMask. The NFT's are ERC-721 compliant and are fully interoperable with digital wallets and exchanges that support this format.

The $STOR Coin

StorageChain is a native cryptocurrency, also known as "mainstream" or "base" cryptocurrency and has its own independent blockchain. This means that STOR is not built on top of another blockchain like Ethereum, and we have our own unique set of rules and protocols. StorageChain currently pays the highest percentage rewards of any decentralized data storage service.

Many new projects in 2022-2023 are ERC-20 tokens which are digital assets that are built on top of the Ethereum blockchain using Ethereum's smart contract functionality. These projects are typically much easier to build and launch but can have downside considerations including being reliant on a third party's technology and decision making.

The top 4 reasons it is important that StorageChain is a "native" coin:

1. **Infrastructure:** Native cryptocurrencies have their own independent blockchain, while ERC-20 tokens are built on top of the Ethereum blockchain. This means that native cryptocurrencies have their own infrastructure and are not reliant on any other platform, while ERC-20 tokens are reliant on the Ethereum network and its infrastructure.
2. **Scalability:** Native cryptocurrencies may have their own scalability solutions built into their blockchain, while ERC-20 tokens may be limited by the scalability of the Ethereum network.
3. **Decentralization:** Native cryptocurrencies may have a more decentralized network of nodes compared to ERC-20 tokens, which are reliant on the Ethereum network and its nodes.
4. **Security:** Native cryptocurrencies may have more robust security measures in place, as they have their own independent blockchain and are not reliant on any other platform.

StorageChain, a revolutionary blockchain platform, is poised to transform the world of data storage and management. It harnesses the power of Substrate, an advanced technology also utilized by the renowned Polkadot blockchain network. This strategic implementation ensures that StorageChain benefits from the robustness, scalability, and security that Substrate offers.

With StorageChain's innovative architecture, users can expect a seamless and modular experience, thanks to the integration of the Frontier Pallet running on v0.9.37. This pallet, developed specifically for StorageChain, enhances its efficiency and provides a solid foundation for future updates and improvements.

By leveraging Substrate's inherent features, StorageChain ensures a secure and reliable ecosystem for storing and managing data. Substrate's proven track record in the blockchain space guarantees that StorageChain is built on a solid foundation of security measures and best practices.

Furthermore, StorageChain's modular design enables swift updates and enhancements, ensuring that the platform remains up to date with the latest technological advancements. This agility allows for the integration of future features and improvements, enhancing the overall user experience.

The collaboration between Substrate and the Frontier Pallet on StorageChain establishes a powerful synergy that propels the blockchain platform into a league of its own. The combination of Substrate's battle-tested technology and the Frontier Pallet's specialized functionalities ensures that StorageChain stands out as a cutting-edge solution in the field of decentralized data storage and management.

As the industry evolves, StorageChain is well-positioned to adapt and evolve alongside it. The team behind StorageChain is committed to continuous innovation, with a roadmap that promises ongoing updates and optimizations. By harnessing the power of Substrate and the capabilities of the Frontier Pallet, StorageChain aims to redefine the standards of efficiency, security, and modularity in the blockchain ecosystem.

Incentive Plan: Network Growth

OVERVIEW

The purpose of this section is to outline the Incentivization plan for the storage node operators in order to grow the network for storage space and subsequently the bandwidth before the system could potentially be overrun by users.

RATIONALE

Prioritizing the growth for the storage network now will result in a more resilient, robust and Utility-First version of the StorageChain network. By working with both internal and external validators on the network, more rewards are available to incentivize storage node providers to contribute to the system with attendant higher gross payouts. This will transition to fully distributed node validation as the network grows, to become an even more decentralized node validator operation.

Validators

As of now, multiple validators distributed across internal servers are active. We have multiple validators active, and will add more as we make progress on node count and general network growth.

Transaction Capacity

Spread across our validators, we have a transactional capacity of up to 2300 Transactions per second.

Community Tokenomics

From the community tokenomics, we have a wide net of assets we can use to incentivize network validators.

Reward Distribution for Validators

For the distribution of the rewards earned by the internal and external validators, they will be forwarded to the same payout and rewards pool as the storage node providers.

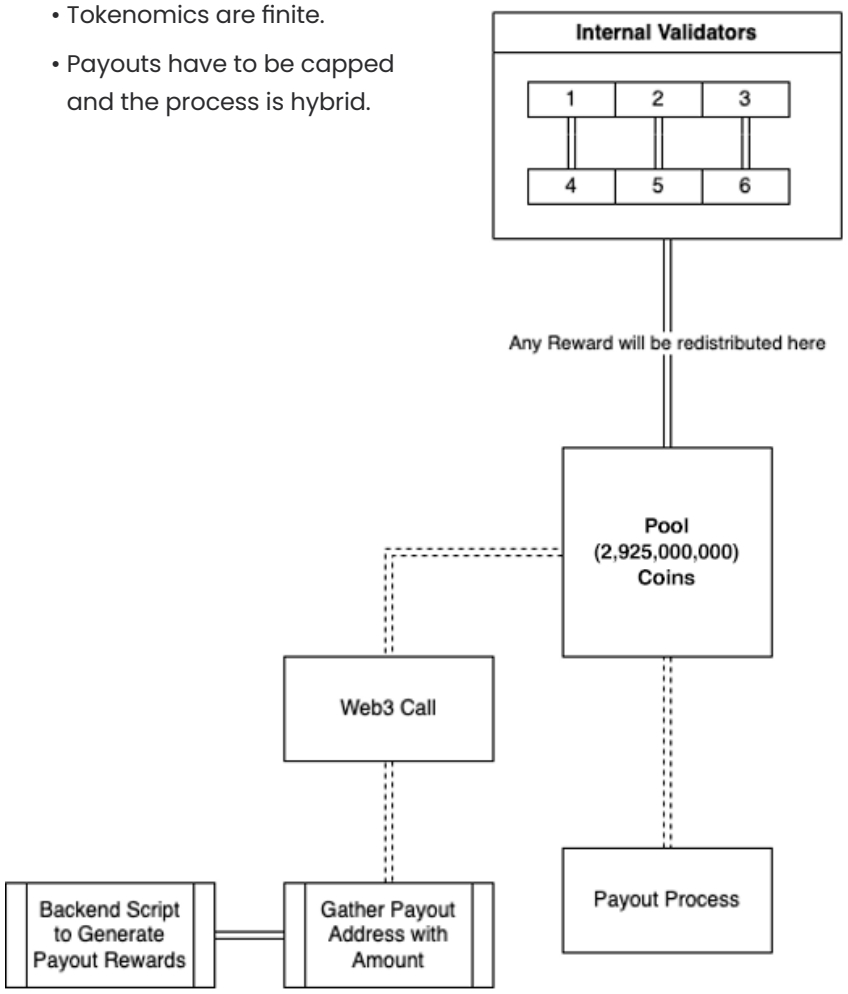
IMPLEMENTATION PLAN

Considering the factors above, we will run validator nodes in an incentivized format wherein, they will allow for payouts and rewards to be earned. We will offer validator rewards calculated differently than storage providers rewards and the payouts would also have a staking requirement. There will be two separate pools setup to payout rewards. One pool for validator nodes and one for storage nodes, both that would be distributed on a monthly basis. The following flowchart illustrates the process for validator nodes:

RISKS

The proposed plan is foolproof; however, some concerns have to be taken into account:

- Tokenomics are finite.
- Payouts have to be capped and the process is hybrid.



Validators

6 Internal Validators working in perfect sync to maintain the blockchain's transactional needs.

Pool (Smart Contract)

The pool will work as a contract that will only be able to be interacted with by the StorageChain consensus mechanism.

Tech Stack

A decentralized storage solution using Kafka, Zookeeper, NestJS, and IPFS Cluster Swarm.

OVERVIEW

The project is a web 3 application that will serve as a decentralized storage system backed by proof of stake and community contributions and rewards.

IPFS CLUSTER SWARM

IPFS Cluster Swarm is a tool that allows you to manage and orchestrate IPFS clusters. An IPFS cluster is a group of IPFS nodes that work together to provide a reliable and scalable storage solution.

By following these steps, you can use IPFS Cluster Swarm to manage and orchestrate your IPFS clusters. IPFS Cluster Swarm provides a powerful set of tools for managing IPFS nodes and content, and it can be used to create a reliable and scalable decentralized storage solution.

To use IPFS Cluster Swarm, you will need to follow these steps:

1. **Install IPFS Cluster:** You can download and install IPFS Cluster from the official website https://cluster.ipfs.io/.

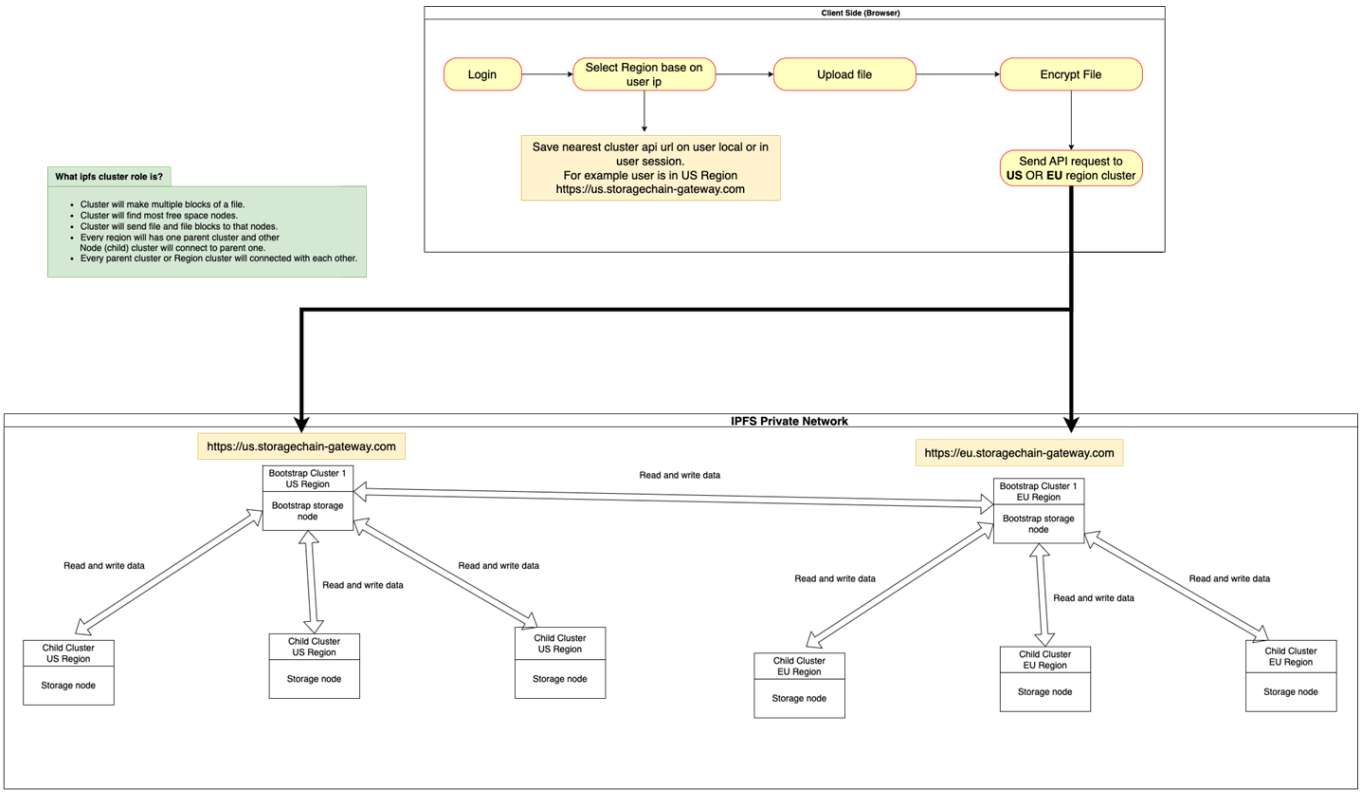
2. **Initialize IPFS Cluster:** Once you have installed IPFS Cluster, you need to initialize it by running the following command:
   `ipfs-cluster-service init`
   This command will create a new IPFS Cluster repository on your system.

3. **Start IPFS Cluster:** To start IPFS Cluster, run the following command:
   `ipfs-cluster-service daemon`
   This command will start the IPFS Cluster daemon and allow you to manage your IPFS nodes

4. **Add IPFS nodes:** You can add IPFS nodes to your cluster by running the following command:
   `ipfs-cluster-ctl peers add <peer-multiaddress>`
   This command will add a new IPFS node to your cluster.

5. **Pin content:** To pin content to your cluster, run the following command:
   `ipfs-cluster-ctl pin add <CID>`
   This command will pin the content to your IPFS nodes and ensure that it is available on the network.

6. **Monitor cluster:** WIP - This will be a customized application that will be automatically installed on your system along with the node to manage the usage and see the details of usage. You can also monitor your IPFS cluster using the IPFS Cluster API or the IPFS Cluster web interface if configured manually.



WEB SERVICES

Defining the Basic Structure of individual components with some basic code examples

1. **Data Ingestion with Kafka**
   Kafka can be used to ingest data from various sources, such as user uploads or application logs. Here's an example of how to produce data to a Kafka topic using the kafka-node library:

```
const kafka = require('kafka-node');
const Producer = kafka.Producer;
const client = new kafka.KafkaClient();
const producer = new Producer(client);

producer.on('ready', function () {
  const payloads = [
    { topic: 'my-topic', messages: 'hello world' }
  ];
  producer.send(payloads, function (err, data) {
    console.log(data);
  });
});
```

This code sets up a Kafka producer and sends a message to the my-topic topic. You can replace the message with data from your application or user uploads.

2. **Data Processing with Kafka**
   Kafka can be used to process the ingested data and distribute it to the appropriate IPFS Cluster Swarm nodes. Here's an example of how to consume data from a Kafka topic using the kafka-node library:

```
const kafka = require('kafka-node');
const Consumer = kafka.Consumer;
const client = new kafka.KafkaClient();
const consumer = new Consumer(
  client,
  [{ topic: 'my-topic', partition: 0 }],
  { autoCommit: false }
);

consumer.on('message', function (message) {
  console.log(message);
  // Pin message to IPFS Cluster Swarm nodes
});

consumer.on('error', function (err) {
  console.log('Error:', err);
});
```

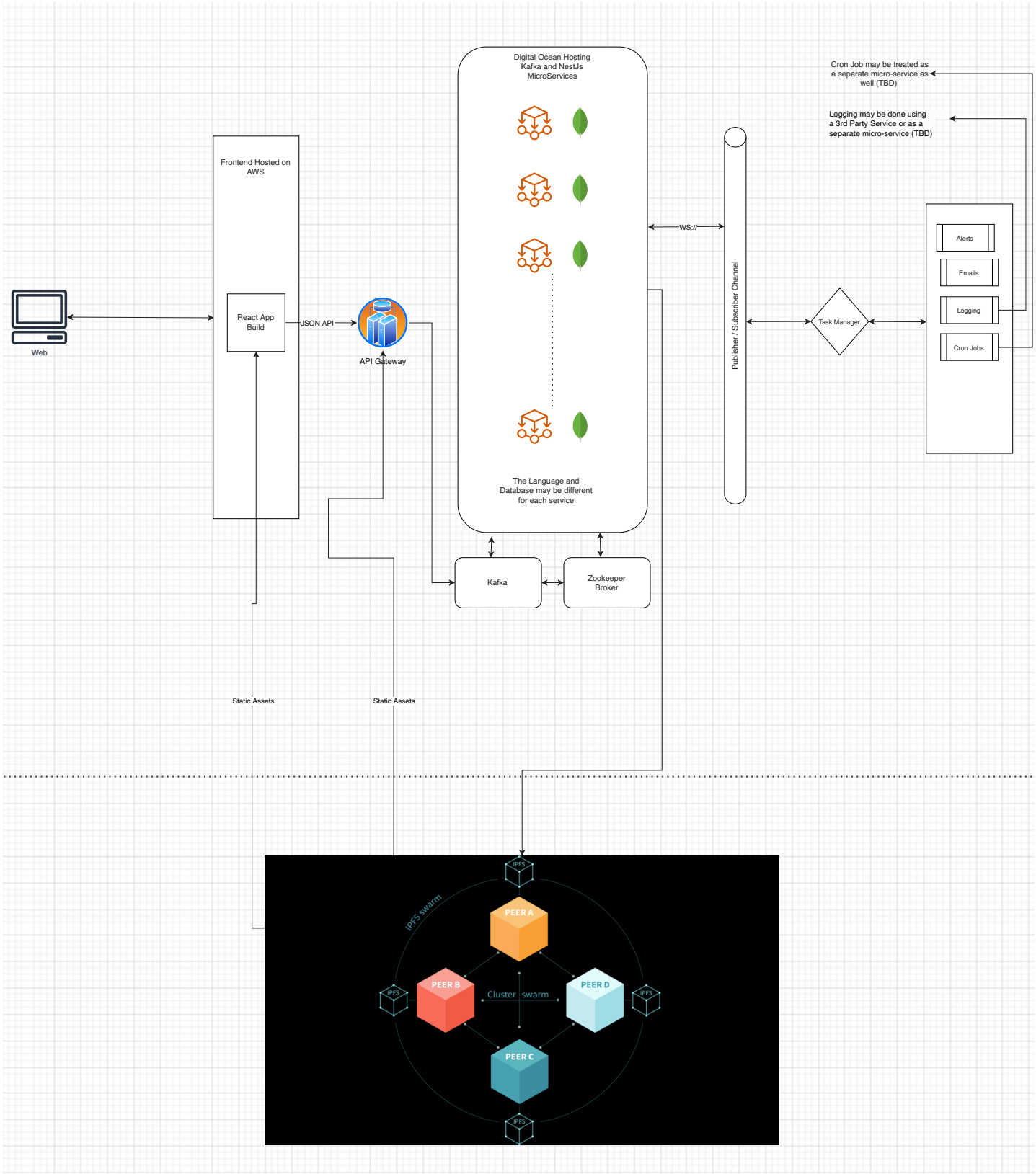
This code sets up a Kafka producer and sends a message to the my-topic topic. You can replace the message with data from your application or user uploads.

3. **IPFS Cluster Swarm Integration with NestJS**
   NestJS can be used to build the application logic that interacts with Kafka and IPFS Cluster Swarm. Here's an example of how to use the IPFS Cluster API to pin data to the IPFS Cluster Swarm nodes:

```
import { Injectable } from '@nestjs/common';
import { Cluster } from 'ipfs-cluster-api';

@Injectable()
export class IpfsClusterService {
  private readonly ipfsCluster: Cluster;

  constructor() {
    const ipfsClusterUrl = 'http://localhost:9094';
    this.ipfsCluster = new Cluster(ipfsClusterUrl);
  }

  async pinData(cid: string): Promise<void> {
    await this.ipfsCluster.pin.add(cid);
  }
}
```

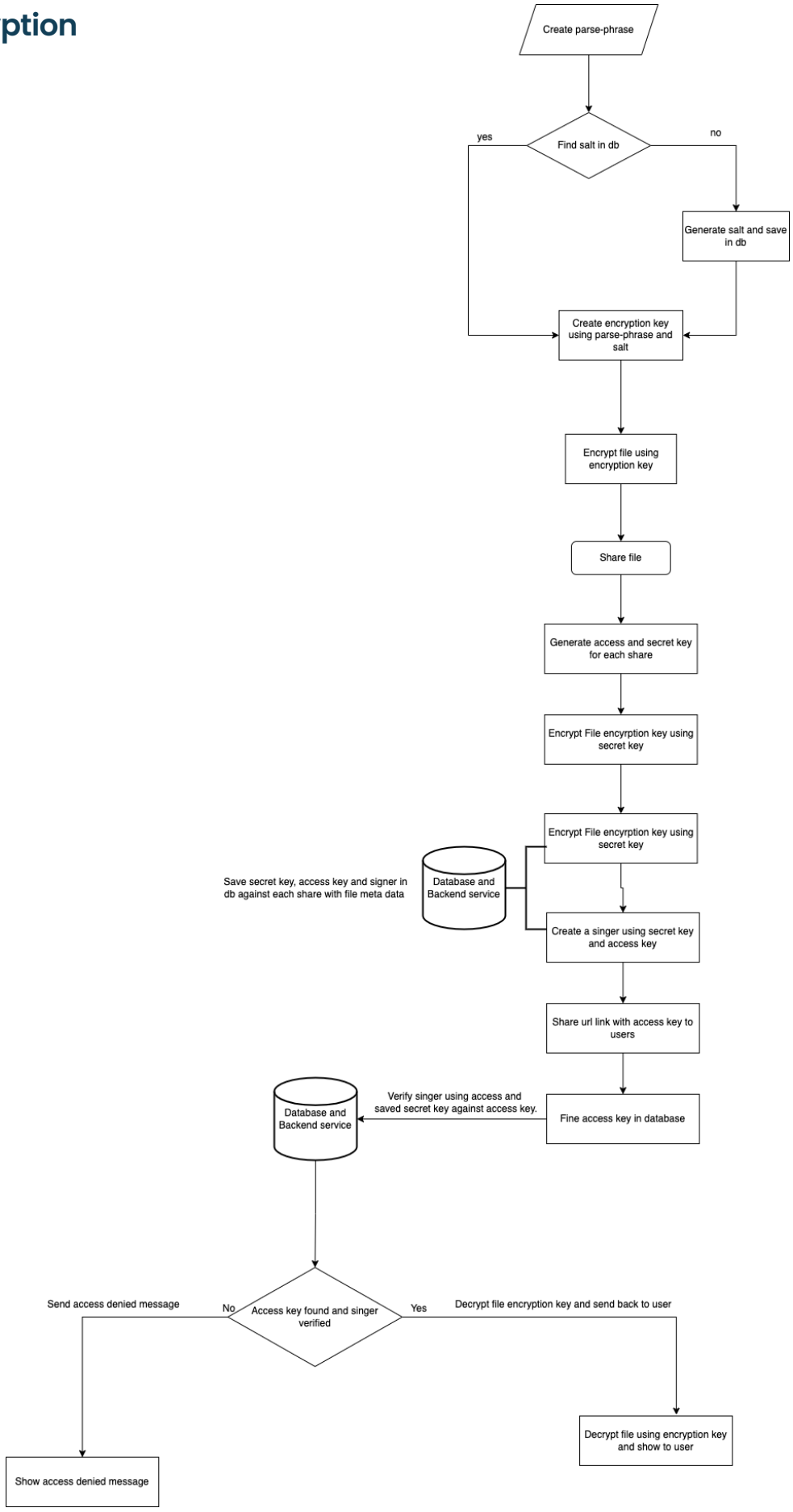
4. **Configuration Management with Zookeeper**
   Zookeeper can be used to manage the configuration and state of the IPFS Cluster Swarm nodes. Here's an example of how to use the node-zookeeper-client library to manage the configuration of the IPFS Cluster Swarm nodes:

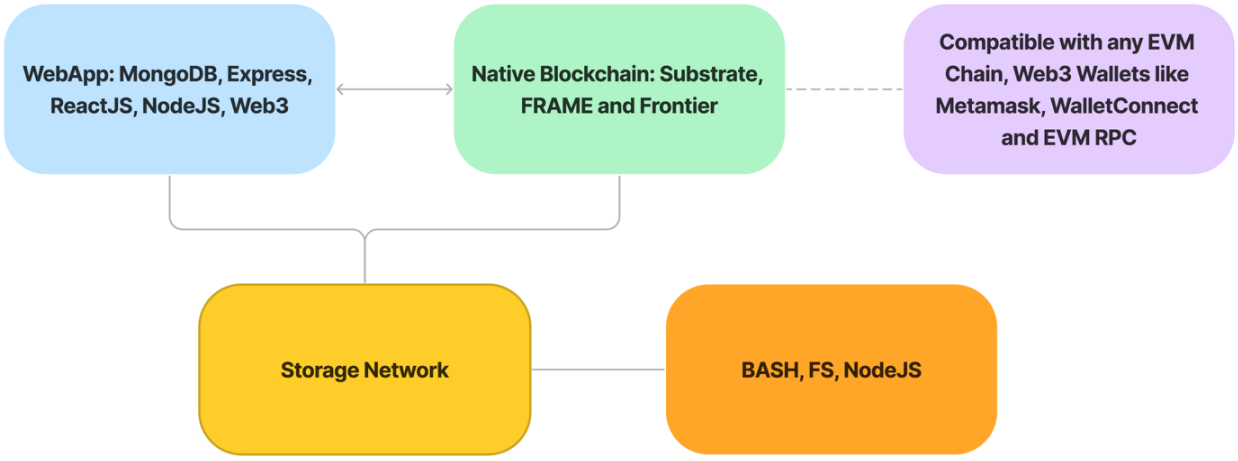
```
import { Injectable } from '@nestjs/common';
import zookeeper from 'node-zookeeper-client';

@Injectable()
export class IpfsClusterConfigService {
  private readonly zkClient: zookeeper.Client;

  constructor() {
    const zkConnectionString = 'localhost:2181';
    this.zkClient = zookeeper.
createClient(zkConnectionString);
    this.zkClient.connect();
  }

  async setConfig(configPath: string, configData: string):
Promise<void> {
    if (!(
```



Front End and Connectivity: React JS & Web3JS, Connectivity Ethers.JS, Blockchain: Substrate (EVM Compatible), Back End: Node JS

Encryption


Create parse-phrase
Find salt in db
yes
no
Generate salt and save in db
Create encryption key using parse-phrase and salt
Encrypt file using encryption key
Share file
Generate access and secret key for each share
Encrypt File encyrption key using secret key
Encrypt File encyrption key using secret key
Save secret key, access key and signer in db against each share with file meta data
Database and Backend service
Create a singer using secret key and access key
Share url link with access key to users
Database and Backend service
Verify singer using access and saved secret key against access key.
Fine access key in database
Send access denied message
No
Access key found and singer verified
Yes
Decrypt file encryption key and send back to user
Decrypt file using encryption key and show to user
Show access denied message



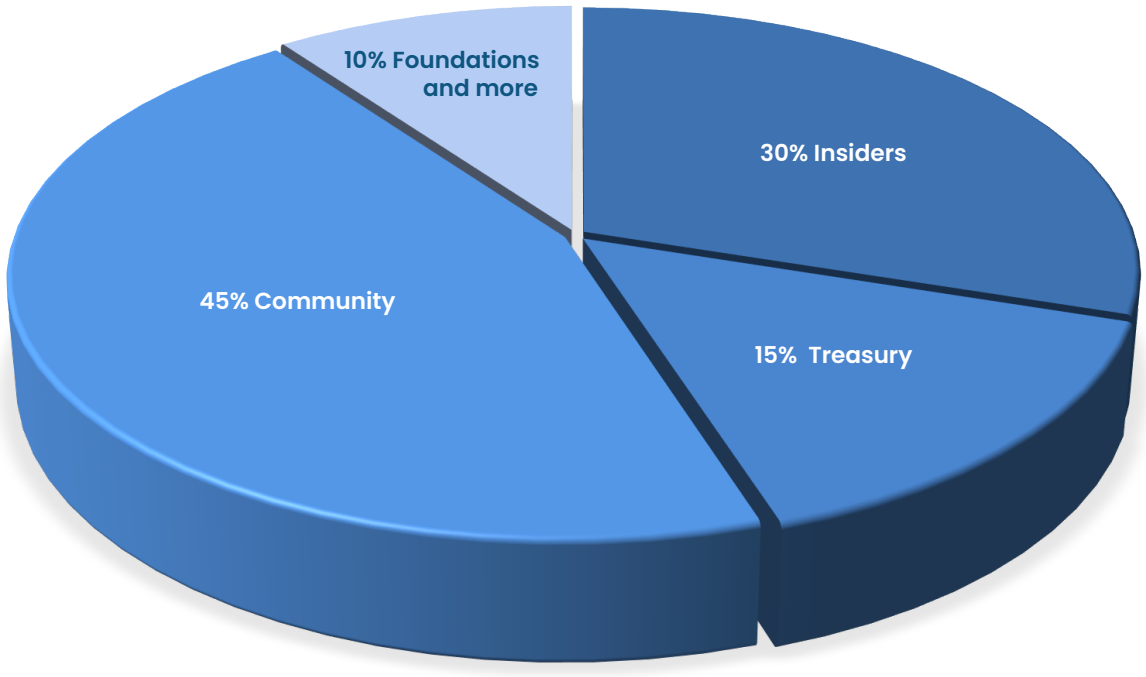
The Tokenomics for STOR currency

SYMBOL	STOR
Platform (native - wrapped):	ERC 20
Consensus Algorithm:	PoS
Total Coins Authorized:	6,500,000,000
Circulating Supply:	2,016,000,000
Network Size (in total # of nodes to start):	~200 nodes
Node Collateral Requirement (in # of STOR):	5,000
Est. Current Price per Node:	$65
Circulated Coins locked in MN's:	1,000,000
% of Circulating Coins Locked in MN:	<1%
Reward amount per month in coins/node:	1,000
Reward revenue per month/node:	$13.00
Reward revenue per year/node:	$156.00
Reward Coins Issued/Month - Total Network:	200,000
Reward Coins Issued/Year - Total Network:	2,400,000
Reward Revenue/Month - Total Network:	$2,600
Reward Revenue/Year - Total Network:	$31,200
Reward Coins Issued/Year as a % of Total Coins Auth.	<1%
**Est. Years Remaining of Reward Coins Issuance:	+40 years

** is Community Allocations (including future staking) coins divided by annual reward coins = est. reward years remaining.

***__actual rewards may be higher or lower__ than shown above based on market conditions and other factors.

Token Allocation



Consensus Mechanism

The StorageChain project uses a proof-of-stake (PoS) consensus mechanism, which allows users to earn rewards by "staking" their coins (essentially, holding them in their own wallet and actively participating in the network). Staking rewards are a way for users to earn ongoing income by supporting the project. StorageChain is a decentralized, cloud storage platform that uses a network of nodes to store data in a distributed manner. Nodes on the StorageChain network can earn rewards for providing storage and bandwidth to the network by participating in staking. To participate in staking on the StorageChain network, you will need to become a network partner and set up a node to store data. You will also need to hold a minimum amount of StorageChain tokens (STOR) as collateral and to qualify for rewards. The amount of rewards you can earn will depend on a variety of factors, including the amount of storage and bandwidth you provide to the network, the amount of STOR you hold, and the overall health and performance of your node.

Treasury Detail

1. Storage node operational expenses (40%)
2. Advertising & Marketing (10%)
3. Ongoing Technology Development and Support (30%)
4. Personnel costs (20%)

Community Acquisition

StorageChain (STOR) knows the importance of community. Social media posts, discussion threads, memes, video tutorials, and contests that promote authentic engagement are core to a successful project. Members of the validator and storage node partner community want to earn incentives and be fully informed. They are the ones discussing the project on social media and hosting/using our products. A huge focus of StorageChain (STOR) is our community. The community also helps lead to greater decentralization of the data storage project over time.

COMMUNITY ENGAGEMENT PLATFORM	USE
Discord	Core, Discussion
Twitter	Core, Marketing
Instagram	Strategic
TikTok	Strategic
Facebook	Tactical
YouTube	Tactical, Marketing
Telegram	Discussion

Community Management

The presence of a STOR community manager will help engage the community and instill confidence in the project. The social media threads need moderation and attention from a member of the STOR team. The team will provide a roadmap to achieving the company's goals from a network growth and partnership standpoint. New community members will be welcomed and our commitment is to have any questions or concerns answered or addressed in a timely manner.

STOR will invite the community members who are interested in our project, and will tap into the network effect by properly incentivizing the community that promotes authentic engagement. Some of the core drivers:

1. Incentives (create storage nodes)
2. AMA (Ask me anything) events
3. Meme contests
4. Merch giveaways
5. Hackathons
6. Collaborate with other projects
7. Integrating with third-party apps like Hubspot, Marketo, and Slack.

Launch Plan

STOR has plans to build a dynamic and vibrant user community of people that are interested in earning incentives and being part of the seismic shift from centralized data storage to de-centralized data storage. We feel that the StorageChain use case is among the strongest in the crypto/web3 landscape.

STOR plans to initially target the media industry and individuals with heavy usage of media or software files. StorageChain is optimized for the sharing of larger video and media files.

STOR is developing a unique brand identity and messaging that sets our project apart from others in the market. A few launch initiatives:

1. Creating quality marketing content: Share valuable information about our project through blog posts, videos, white and lite papers, and other types of content that educate and engage our target audience.
2. Leverage social media: Utilize popular social media platforms to reach and engage with our target audience and promote content.
3. Network with the community: Attend events, participate in online forums and chat groups, and engage with the crypto community to build relationships and create a network of support for the STOR project.
4. Attend to events, conferences, and industry meet-ups.
5. Soft-launch Beta of the product in September 2023.
6. Partner with blockchain influencers and key players: Collaborate with industry influencers and key players to bring awareness to our project and reach a wider audience.
7. Utilize paid advertising: Use paid advertising channels such as Google AdWords, Facebook Ads, and others to drive targeted traffic to the STOR project's website and increase visibility.