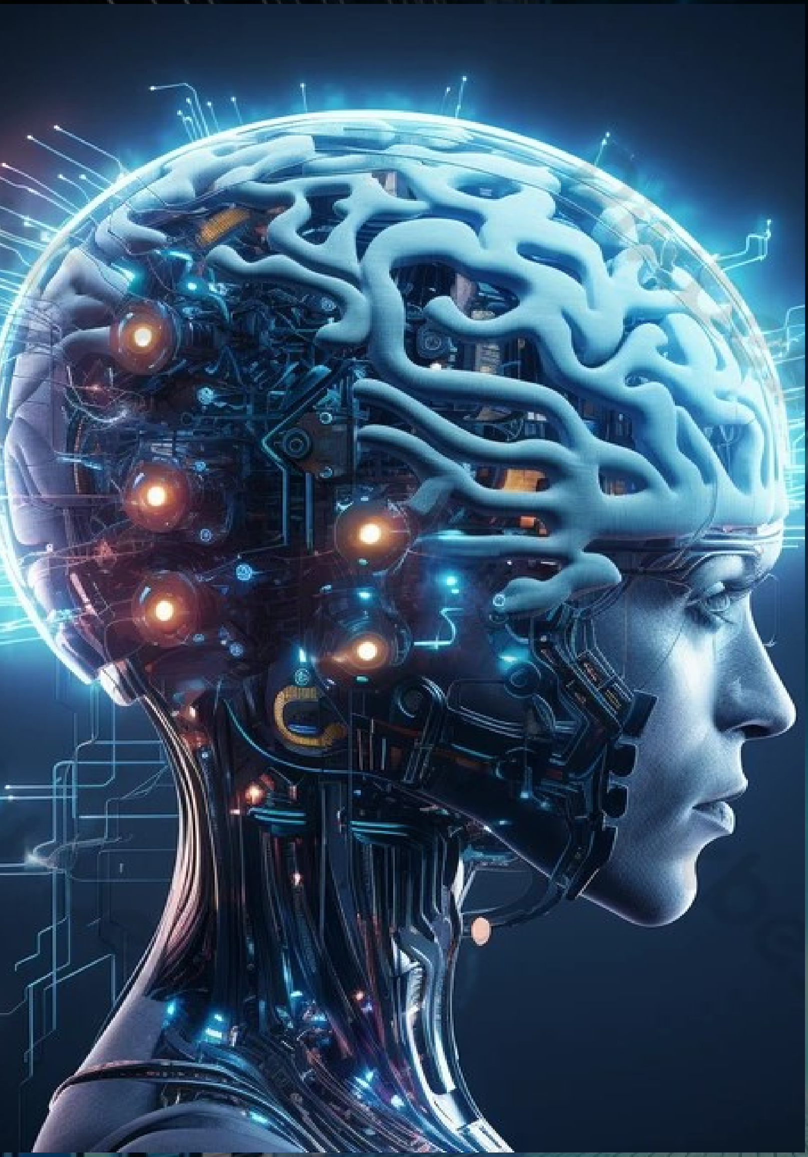
Cloudatabox

BLUEPAPER

Version : 0:1



CDB Crypto Economics

Cloudatabox (CDB) Crypto economics is the study of how cryptocurrency can incentivize usages of a blockchain network. This page covers how Cloudatabox manages incentivization within the network. The Cloudatabox Virtual Machine is here, allowing developers to build applications and smart-contracts that utilize the Cloudatabox storage network. And also help to earn more CDB tokens.

Learn The Basics

Want to understand the fundamental concepts of Cloudatabox, but aren't sure where to start? The Basics section is the perfect place to begin your journey!

Store Data

Want to understand the fundamental concepts of Cloudatabox, but aren't sure where to start? The Basics section is the perfect place to begin your journey!

Build With Cloudatabox

Eager to build on top of the Cloudatabox network but aren't sure how to begin? Head over to the Developers section to find examples and guides to help build your project.

Become a Provider

Eager to build on top of the Cloudatabox network but aren't sure how to begin? Head over to the Developers section to find examples and guides to help build your project.

What Is Cloudatabox

This section aims to provide a comprehensive overview of Cloudatabox platform to developers & serves as a reference that developers can check back on.

Cloudatabox is a peer-to-peer network that stores files, with built-in economic incentives and cryptography to ensure files are stored reliably over time. In Cloudatabox, users pay to store their files on storage providers. Storage providers are computers responsible for storing files and proving they have stored them correctly over time. Anyone who wants to store their files or get paid for storing other users' files can join Cloudatabox. Available storage, and the price of that storage, are not controlled by any single company. Instead, Cloudatabox facilitates open markets for storing and retrieving files that anyone can participate in.

Cloudatabox is built on top of the same software powering IPFS protocol, which is a peer-to-peer distributed storage network that leverages content addressing to allow permanent references to the data & avoids relying on specific devices or cloud servers for addressing the content.

THE MARKETING

Cloudatabox is different from IPFS because it has an incentive layer on top to incentivize contents to be reliably stored and accessed.

Cloudatabox enables several use cases, from Web3 native NFT and metaverse/game assets storage, incentivized permanent storage, to archiving Web2 datasets as a cheaper alternative to cloud storage. For example, NFT Storage utilizes Cloudatabox to provide a simple decentralized storage solution for NFT contents and metadata, while Internet Archive leverages Cloudatabox to back up their contents. Cloudatabox also supports a wide range of formats of data, including audio and video files, allowing Web3 platforms such as Audius and Huddle01 to leverage Cloudatabox as the decentralized storage backend for music streaming and video conferencing.

Cloudatabox Native Currency

The native currency of Cloudatabox is CDB token which is on Storechain Blockchain. CDB is a utility token used to incentivize persistent storage on the Storchain network. Storage providers mine CDB tokens by providing reliable storage service or committing storage capacity on the network. It has a maximum circulating supply of 12 Billion CDB, meaning that no more than 2% billion CDB tokens will ever be created.

As a utility token that aligns participants' incentives with the long-term growth of the network, Cloudatabox issuance is aligned with the overall provable utility of the network. The majority of Cloudatabox supply would only be minted if the network achieved growth and utility targets at scale.

Specifically, Cloudatabox uses a dual minting model for block reward minting:

Baseline Minting

Up to 20% CDB tokens are minted based on the performance of the network. These tokens would only fully release if the Storchain network reached a Yottabyte of storage capacity in under 10 years, estimated to be ~1000x larger than today's cloud storage capacity.

Simple Minting

CDB tokens are released on a 2-year half-life based on time, meaning that 60% of these tokens will be released in approximately 10 years time.
Additionally, CDB tokens are held back in the mining reserve to incentivize future types of mining.

CDB Vesting Schedule

Mining rewards undergo a vesting schedule to encourage long-term network alignment. For example, 75% of block rewards earned by miners vest linearly over 180 days, while 25% are made immediately available to improve miner cash flow and profitability. And the remaining CDB tokens are vested to Storchain teams & Cloudatabox Foundation over 4 years and SAFT investors over 2 years, as shown in the vesting table here.

Collateral and slashing

To encourage good behavior from network participants, during block reward mining, storage providers must lock Cloudatabox tokens as pledge collateral for consensus security, storage reliability, and contract guarantees. Pledge collateral is determined by projected block rewards that a miner would earn. Collateral and all earned rewards by storage providers are subject to slashing throughout the lifetime of a sector if the storage does not pass a reliability check.

Storchain Blockchain

A Storchain blockchain is a distributed database layer 1 blockchain, that is shared among the nodes of a computer network. This page covers how the Storchain blockchain is designed, and the various functions it has.

Tipsets

The Storchain blockchain is a chain of tipsets rather than a chain of blocks. A tipset is a set of blocks with the same height and parent tipset. Therefore, multiple storage providers can produce blocks for each epoch to increase network throughput. Each tipset is assigned a weight, so the consensus protocol directs nodes to build on the heaviest chain. This provides a certain level of security to the Cloudatabox network by preventing a node from intentionally intervening with other nodes to produce valid blocks.

Actors

An Actor in the Storchain Blockchain is the equivalent of the smart contract in the Ethereum Virtual Machine. It is essentially an 'object' in the Storchain network with a state and a set of methods that can be used to interact with it.

Built-in actors

There are several built-in system actors that power the Storchain network as the decentralized storage network.

- System Actor - General system actor.
- Init actor - Initializes new actors and records the network name.
- Cron Actor - A scheduler actor that runs critical functions at every epoch.
- Account Actor - Responsible for user accounts (non-singleton).
- Reward Actor - Managing block reward and token vesting (singleton).

- Storage Miner Actor - Storage mining operation and validate storage proofs.
- Storage Power Actor - Keeping track of the storage power allocated at each storage provider
- Storage Market Actor - Managing storage deals.
- Multisig Actor - responsible for operations involving the Storchain multi-signature wallet.
- Payment Channel Actor - Set up and settle payment channel funds.
- Datacap Actor - Responsible for data cap token management.
- Verified Registry Actor - responsible for managing verified clients.
- Ethereum address Manager (EAM) Actor- responsible for assigning all Ethereum compatible addresses on Storchain Network, including EVM smart contract addresses and Ethereum account addresses.
- EVM Account Actor - a non-singleton built-in actor representing an external Ethereum identity backed by a secp256k1 key.

User-programmable Actors

Along with the maturity of SVM (Storchain Virtual Machine), developers can write actors and deploy them to the Storchain network in the same way as other blockchains. Other blockchains refer to these programs as smart contracts. User-programmable actors can also interact with built-in actors using the exported API from built-in actors.

You can check out this talk on How Storchain Actors Work to learn more:

Distributed Randomness

Storchain uses distributed and publicly verifiable random beacon protocol - Drand as the randomness beacon for the leader election during the expected consensus to produce blocks. This randomness guarantees that the leader election is secret, fair, and verifiable.

Nodes

Nodes in the Storchain network are primarily identified in terms of the services they provide to serve the Storchain storage network, including chain verifier nodes, client nodes, storage provider nodes, and retrieval provider nodes. Any node participating in the Cloudatabox network should provide the chain verification service as a minimum. Storchain is targeting multiple protocol implementations to guarantee the security and resilience of the Storchain network.

Addresses

In the Storchain network, addresses are used to identify actors in the Storchain state. The address encodes information about the corresponding actor, providing a robust address format that is easy to use and resistant to errors. There are five types of addresses in Storchain. Mainnet addresses begin with the letter S, and Testnet addresses begin with the letter t.

S0/t0: an ID address for an actor in a more "human friendly" way. For instance, S0123261 is the ID for a storage provider.

S1/t1: a secp256k1 wallet address with encrypted key pair. Essentially, this is a wallet address generated from the secp256k1 public key.

S2/t2: an address represents an actor (smart contract) and is assigned in a way that makes it safe to use during network forks.
S3/t3: a BLS wallet address generated from a BLS public encryption key.

S4/t4: the addresses which were created and assigned to user-defined actors by user-definable "address management" actors. This address can receive funds before an actor has been deployed to the address.

S410/t410: the address space managed by Ethereum Address Manager (EAM) built-in actor. The original Ethereum addresses can be cast as S410/t410 addresses & vice versa to enable existing Storchain network.

Storchain Consensus

Let's quickly cover how consensus works in the Storchain network.

Expected consensus

Expected consensus (EC) is the underlying consensus algorithm used by Storchain. EC is a probabilistic Byzantine fault-tolerant consensus protocol that runs a leader election among a set of storage providers to submit a block every epoch. Like proof-of-stake, Storchain uses proof-of-storage for the leader election, meaning the likelihood of being elected depends on how much provable storage power a miner contributes to the network. The storage power of the network is stored in the storage power table and managed by the Storage Power Actor.

At a high level, the consensus process relies on Drand to provide distributed and verifiable randomness to keep leader election secret, fair and verifiable. All the election participants and their power are drawn from the Power Table, which is calculated and maintained over time by the Storage Power Consensus subsystem. Eventually, EC takes all valid blocks produced in this epoch and uses a weighting function to select the chain with the highest weight to add blocks.

Block Production Process

The process of producing a block for each epoch can be briefly described as follows:

- Elect leaders from eligible miners.
- Miners need to check if they are elected.
- An elected miner gets the randomness value to generate Winning Post.
- If all above is successful, miners build and propagate a block.
- Verify whether a miner won the block and verify the leader election.
- Eventually, select the heaviest chain to add blocks.

Finality

Storchain enforces a version of soft finality whereby all miners at round N will reject all blocks that fork off before round N - F. F is set to 900. This is important to enforce finality at no cost to chain availability and functionality.

Proofs

As a decentralized storage network, Storchain is built on the proof-of-storage in which miners contribute their vacant storage space to the network to store data and then provide proofs for the client to verify if their data has been stored throughout a period.

Proof of replication

Using proof-of-replication (PoRep), storage providers prove that they have created a unique copy of the client's data and stored it on behalf of the network.

Proof of spacetime

Storage providers also need to continuously prove that they store clients' data for the whole lifetime of the storage deal. There are two types of challenges as part of the proof-of-spacetime (PoSt) process:

- Winning PoSt guarantees that the storage provider maintains a copy of data at a specific time.
- Window PoSt is used as proof that a copy of the data has been continuously maintained over time.

Slashing

If storage providers fail to provide reliable uptime or act maliciously against the network, they will be penalized by slashing. Storchain implements two kinds of slashing:

- Storage fault slashing to penalize storage providers for not being able to maintain healthy and reliable storage sectors for the network.
- Consensus fault slashing to penalize storage providers to sabotage the liveness and security of the consensus process.

Cloudatabox Storage Model

A storage model defines how data is stored within a system. This page covers the basic aspects of Cloudatabox storage model.

The Cloudatabox storage model consists of three components:

- CDB Providers
- CDB Deals
- CDB Sectors

CDB Providers

Cloudatabox Providers, as the name suggests, provide a service to users of the network. There are two types of provider:

- Cloudatabox Storage providers
- Cloudatabox Retrieval providers

Deals

In the Cloudatabox network, CSPs and CRPs provide their storage or retrieval service to data clients through deals. These deals are negotiated and agreed upon between two parties, and include terms like data size, price, deal duration, and collateral.

The deal-making process happens off-chain. Once both parties agree to the terms of a deal, that deal is published on-chain for the rest of the network to see and validate.

Sectors

Sectors are the basic units of provable storage where storage providers store clients' data and generate PoSt on behalf of the Storchain network. Sectors have standard sizes and a lifetime that storage providers can extend before reaching the end of the lifetime. 32 GiB and 64 GiB sector sizes are supported.

Storage market

The storage market is the data entry point into the network where storage providers and clients negotiate and publish storage deals on-chain.

Cloudatabox Storage providers

Cloudatabox Storage providers, often called CSPs, are responsible for storing files and data for clients on the network and providing cryptographic proofs to verify storage. The vast majority of providers on the Cloudatabox network are CSPs.

Cloudatabox Plus

The mission of Cloudatabox Plus (CDB+) is to maximize the amount of useful storage on the Cloudatabox network. The aim is to bring more meaningful and valuable data into the Cloudatabox network by offering verified clients cheap, or even free, storage. This mechanism is designed and operates around Datacap, the storage quota allocated to verified clients to store data and boost incentives for storage providers.

Verified clients can onboard data into Cloudatabox using STR, which they apply from community-selected notaries. In return for storing verified storage deals, storage providers receive STR with a 10x boost to their storage power which eventually increases their block rewards as an incentive.

❖ Storchain: a STR coin will be allocated to a verified client to spend in the storage deals carrying a 10x deal quality multiplier.

❖ Notaries: community-selected notaries govern the program by verifying storage clients and allocating STR Coin to verified clients.

❖ Verified clients: clients are active network participants with STR allocation for their data storage.

Storage on-ramps

❖ To simplify the process of storing data on the network, there are many storage helpers to provide an easier way to integrate Cloudatabox storage as well as CIPFS into your applications or smart contracts.

❖ Storage helpers provide libraries that abstract Cloudatabox deal-making into simple, streamlined API calls and store the data on CIPFS to provide more efficient and fast retrieval for your content.

Retrieval Market

The retrieval market refers to negotiating retrieval deals for a provider to serve stored data to a client. In this agreement, the client agrees to pay the retrieval provider a certain amount of CDB

Basic Retrieval

Currently, Cloudatabox nodes support direct retrieval from the storage miners who originally stored the data. Clients can directly send retrieval requests to a storage provider to retrieve their data by paying some CDB for retrieval.

Clients need to provide enough information to the storage provider for the data retrieval request, including:

Storage provider ID: The ID of the storage provider where the data is stored.
Payload CID: also called Data CID.
Address: The address initially used to create the storage deal.

Saturn

Saturn is a Web3 CDN in Cloudatabox retrieval market which serves the data stored on Cloudatabox with low latency and at low cost. It consists of independent retrieval providers specifically dedicated to that business, making retrieval an efficient, fast, and reliable operation.

Cloudatabox Retrieval providers

Retrieval providers, often called CRPs, are responsible for providing users quick access to their data. They focus on rapid access to data, rather than long-term storage. Most of the time storage providers also provide retrieval access to their users as part of the same system. However, more and more stand-alone CRPs are joining the network.

Deal making

The lifecycle for a deal within the storage market contains four distinct phases:

Discovery: the client identifies potential CSPs and asks for their prices.
Negotiation: once the client has selected an CSP both parties agree to the term of the deal.
Publishing: the deal is published on-chain.
Handoff: the deal is added into a sector where data storage can be proven by the CSP.

Programming on Cloudatabox

Once data has been stored, it is possible to run computations and calculations on that data, without needing to retrieve the data from a storage provider. This page covers the basics of how programming.

Compute-over-data

When it comes to data, a common need beyond storage and retrieval is data transformation. The goal with the compute-over-data protocols is generally to perform computation over IPLD, which is the data layer used by content-addressed systems like Cloudatabox. There are working groups working on different types of computing on Cloudatabox data, such as large-scale parallel compute (e.g., Bacalhau) and cryptographically verifiable compute (e.g. Lurk), etc. For example, Bacalhau is a platform for public, transparent, and optionally verifiable distributed computation. It enables users to run arbitrary Docker containers and Web Assembly (wasm) images as tasks against data stored in the Inter Planetary File System (IPFS).

It is worth noting that Cloudatabox is uniquely positioned to support large-scale off-chain computation since the storage providers have to compute resources such as GPUs and CPUs collocated with their data. By supporting compute-over-data on the Cloudatabox network, we enable a new paradigm of computing on the data where the data exists rather than moving the data to external compute nodes.

Storchain Virtual Machine

The Storchain virtual machine (SVM) is a runtime environment for smart contracts on the Cloudatabox network. Smart contracts enable users to run any bounded computation, including those that create and enforce rules for storing and accessing data on the network. The SVM is responsible for executing these smart contracts and ensuring they are executed correctly and securely.

SVM is designed to support native Cloudatabox actors written in languages that compile to WASM, as well as smart contracts written for foreign runtimes, including Solidity contracts for Ethereum Virtual Machine (EVM), Secure ECMAScript (SES), and eBPF. The reference FVM and SDK are written in Rust.

According to the SVM roadmap, we initially support smart contracts written in Solidity and eventually support any language that compiles to WASM.

The SVM enables compute-over-states on the Cloudatabox network and allows developers to build endless new use cases on top of Cloudatabox. Some example use cases are:

Data organizations

SVM can create a new kind of organization – one built around datasets of various kinds.

Data DAO and tokenized datasets

The SVM enables the creation and management of data-based decentralized and autonomous organizations – data DAOs. The SVM allows a group of individuals, or organizations, to curate and preserve data collection. Data DAOs can govern and monetize data access and pool the returns into a shared treasury to fund the collections preservation and long-term growth. One could even exchange those data tokens between peers and request computation services on that data, such as validation, joins, analysis, feature detection, and extraction, moving into machine learning.

Perpetual Storage

SVM allows users to store once and have repair and replication bots manage the repetitive storage deal creation tasks so that data can be stored perpetually. Using a smart contract, users can provision a wallet with CDB, and storage providers can use that to fund data storage permanently. Repair bots can monitor the storage deals and replicate the data with other storage providers when necessary. This process gives users long-term storage permanence.

Financial services for miners

SVM can provide a variety of financial services for storage providers. The needs of these SPs are unique to the Cloudatabox ecosystem.

Lending and staking protocols

Users can lend out Cloudatabox to storage providers to use it as storage collateral and receive interest in return. These loans can be undercollateralized based on the on-chain storage history of past storage provider performance. Community members can use this history to generate reputation scores, enabling everyone to identify good borrowers. On top of that, loans can be automatically paid back to investors by using a Multisig as the storage provider's owner address, including lenders and a third party, to help negotiate payback. New SVM-enabled smart contracts give every CDB token holder access to new yield opportunities on their holdings while also benefiting the whole Storchain economy by allowing entry ramps for providing storage on the network.

Storchain Infrastructure

We expect that SVM will gain feature parity with other chains that persist. This is required for any EVM chain to operate but is not necessarily tied to storage primitives.

Decentralized exchanges

Users on SVM need to be able to exchange CDB token for other tokens issued on-chain. This may be a decentralized exchange such as a fork of Uniswap or Sushi or involve building a decentralized order book similar to Serum on Solana.

Token bridges

While not immediately on the roadmap, bridges are needed from EVM chains, Move chains, and Cosmos chains to bring wrapped tokens from other ecosystems into the fold. With the current launch, we are more focused internally since the value proposition of Cloudatabox is unique enough that it does not need to bootstrap TVL from other chains. However, in the long run, we expect SVM to be part of a broader family of blockchains.

Besides these, there are a lot more use cases that the SVM could enable, such as data access control (Medusa), retrieval and trustless reputation systems, replication workers, storage bounties, and L2 networks. To learn more about what you can build on top of SVM, check out our Request for Startup post.

If you are interested in building these use cases, there is a list of solution blueprints that might help as a reference point regarding how some of these could work on a high level:

- DataDAO Solution Blueprint
- Perpetual Storage Solution Blueprint
- Lending pool cookbook

Storchain EVM

The Storchain EVM runtime (SEVM) is the Ethereum Virtual Machine (EVM) virtualized as a runtime on top of the Storchain Virtual Machine (SVM). It will allow developers to port any existing EVM-based smart contracts straight onto the SVM and make them work out of the box. SEVM emulates EVM bytecode at the low level, supporting contracts written in Solidity, Vyper, and Yul. The EVM foreign runtime is based on preexisting OSS libraries, including Sputnik and Revm. You can find out more details in the EVM <> SVM mapping specification.

Because Storchain nodes offer the Ethereum JSON-RPC API support, SEVM is also completely compatible with any EVM development tools, such as Hardhat, Brownie, and MetaMask. Most smart contracts ported to Storchain shouldn't require changes or auditing. For example, new ERC-20 tokens can be launched on the Storchain network or bridged directly to token pools on other chains.
Developers can deploy actors on either the SEVM or native SVM; which one should you choose?

The decision can be summed up as such: if you want better performance, write actors that are compiled to WASM and deployed to native SVM. If you are familiar with Solidity and want access to the EVM ecosystem of tools, but don't mind slightly less performance, deploy to the SEVM.

To sum it up, the SEVM allows current Web3 developers to quickly start writing actors on the Storchain blockchain while using all of the tools, software packages, and languages they are used to while having access to Cloudatabox storage deals as a native.

The difference between SEVM and EVM contracts is that contracts deployed on SEVM can interact with built-in actors to interact with Cloudatabox-specific actors, such as miner actors, as mentioned in the built-in actor section. This allows developers to build Storchain-native decentralized applications for the new use cases mentioned above. Smart contracts deployed to the Ethereum blockchain have no direct access to the Storchain network or Cloudatabox-specific actors.

To allow Solidity smart contracts on SEVM to seamlessly call methods on Storchain built-in actors and access Cloudatabox-specific syscalls idiomatically, a Storchain-Solidity API library has been developed, you can use it for building your use cases, such as interacting with storage deals.

If you build on SEVM, you might find some of the example contracts here helpful.

Code | Blame | 12 lines (10 loc) · 155 Bytes

```
# Compiler files
cache/
out/

# Ignores development broadcast logs
!/broadcast
/broadcast/*/31337/
/broadcast/**/dry-run/

# Dotenv file
.env
node_modules
```

Code Blame 22 lines (22 loc) · 410 Bytes

```json
{
  "dependencies": {
    "@zondax/filecoin-solidity": "2.0.0-beta.1"
  },
  "devDependencies": {
    "husky": "^8.0.3",
    "lint-staged": "^13.1.2",
    "prettier": "^2.8.4",
    "prettier-plugin-solidity": "^1.1.3"
  },
  "scripts": {
    "prepare": "husky install"
  },
  "husky": {
    "hooks": {
      "pre-commit": "lint-staged"
    }
  },
  "lint-staged": {
    "*.sol": ["npx prettier --write"]
  }
}
```

Code Blame 16 lines (12 loc) · 194 Bytes

```makefile
check-deps:
	@cargo -V > /dev/null
	@forge -V > /dev/null

build: check-deps
	forge build
.PHONY: build

test: check-deps
	forge test
.PHONY: test

clean: check-deps
	forge clean
.PHONY: clean
```

Networks

The Storchain network has several different networks for testing, staging, and production purposes. This page contains information on available networks.

Mainnet

Mainnet is the live production network that all nodes on the Storchain network are connected to. It never resets.

Testnets

Test networks, or testnets, are version of the Storchain network that attempt to simulate various aspects of the Storchain mainnet. Since they are for testing, they should not be used for production applications or services.

Calibration

Calibration Testnet is the most realistic simulation of the mainnet, where prospective storage providers can experience more realistic sealing performance and hardware requirements due to the use of final proofs constructions and parameters, and prospective storage clients can store and retrieve real data on the network. Clients can participate in deal-making workflows and storage/retrieval functionality. It also has the same sector size as the mainnet.

Public endpoint
Blockchain explorer

The CDB token

CDB is the cryptocurrency that powers the Storchain network. This page explains what CDB is, how it can be used, and its denominations. Uses CDB plays a vital role in incentivizing users to participate in the Storchain network and ensuring its smooth operation. Here are some ways in which CDB is used on the Storchain & Cloudatabox network:

Network payments

When a user wants to store data on the Cloudatabox network, they pay in CDB to the storage providers who offer their storage space. The payment is made in advance, for a certain amount of time that the data will be stored on the network. In addition, storage providers choose their own terms and payment mechanisms for providing storage and retrieval services so other options (such as fiat payments) can be available soon.

Blockchain rewards

Storage providers are also rewarded with CDB tokens for providing their storage space and performing other useful tasks on the network. CDB is used to reward storage providers who validate and add new blocks to the Storchain blockchain. Providers receive a block reward in STR coin & CDB tokens for each new block they add to the blockchain and also earn transaction fees in STR Coins & CDB tokens for processing storage and retrieval transactions.

Governance

As members of the Cloudatabox community, CDB token holders are encouraged to participate in the Cloudatabox governance process. They can do so by proposing, deliberating, designing, and/or contributing to consensus for network changes, alongside other stakeholders in the Cloudatabox community- including implementers, Core Devs, storage providers, and other ecosystem partners.

Wallets

Wallets provide a way to securely store Cloudatabox tokens & STR coins, along with other digital assets. These wallets consist of a public and private key, which work similarly to a bank account number and password. When someone sends cryptocurrency to your wallet address, the transaction is recorded on the blockchain network, and the funds are added to your wallet balance. Similarly, when you send cryptocurrency from your wallet to someone else's wallet, the transaction is recorded on the blockchain network, and the funds are deducted from your wallet balance.

There are various types of cryptocurrency wallets, including desktop, mobile, hardware, and web-based wallets, each with its own unique features and levels of security. It's important to choose a reputable and secure wallet to ensure the safety of your digital assets.

Compatible Wallets

We do not provide technical support for any of these wallets. Please use caution when researching and using the wallets listed below. Wallets that have conducted third-party audits of their open-source code by a reputable security auditor are marked recommended below.

Security

Wallets that have gone through an audit have had their codebase checked by a recognized security firm for security vulnerabilities and potential leaks. However, just because a wallet has had an audit does not mean that it's 100% bug-proof. Be incredibly cautious when using unaudited wallets. Never share your seed phrase, password, or private keys. Bad actors will often use social engineering tactics such as phishing emails or posing as customer service or tech support to lure users into handing over their private key or seed phrase.

cloudatabox

Storchain Interplanetary Consensus

Storchain Interplanetary Consensus (SIPC) powers planetary-scale decentralized applications (dApps) through horizontal scalability of Storchain, Ethereum and more.

What is SIPC?

Storchain Interplanetary Consensus (SIPC) is a framework that enables on-demand horizontal scalability of networks, by deploying "subnets" running different consensus algorithms depending on the application's requirements.

What is Storchain horizontal scalability & why is it important for dApps?

Storchain Horizontal scalability generally refers to the addition of nodes to a system, to increase its performance. For example, adding more nodes to a compute network helps distribute the effort needed to run a single compute task. This reduces cost per task and decreases latency, while improving overall throughput.

In web3, horizontal scalability refers to scaling blockchains, for desired performance. More specifically, scaling the ability of a blockchain to process transactions and achieve consensus, across an increasing number of users, at desired latencies and throughput. SIPC is one such scaling solution, alongside other popular layer 2 solutions, like sidechains and rollups.

For decentralized applications (dApps), there are several key motivations to adopt scaling - performance, decentralization, security. The challenge is that these factors are known to be conflicting goals.

How does SIPC achieve horizontal scalability?

SIPC is a scaling solution intentionally designed to achieve considerable performance, decentralization and security for dApps. It achieves scaling through the permissionless spawning of new blockchain sub-systems, which are composed of subnets. Subnets are organized in a hierarchy, with one parent subnet being able to spawn infinite child subnets. Within a hierarchical subsystem, subnets can seamlessly communicate with each other, reducing the need for cross-chain bridges. Subnets also have their own specific consensus algorithms, whilst leveraging security features from parent subnets. This allows dApps to use subnets for hosting sets of applications or to shard a single application, according to its various cost or performance needs.

How is SIPC unique as a scaling solution?

Earlier, we talked about the challenge of scaling solutions to balance performance, security and decentralization. SIPC is a standout framework that strikes a considerable balance between these factors, to achieve breakthroughs in scaling.

- Highly customizable without compromising security. Most L2 scaling solutions today either inherit the L1's security features but don't have their own consensus algorithms (e.g. rollups) or do the reverse (e.g. sidechains). They are also deployed in isolation and require custom bridges or protocols to transfer assets and state between L2s that share a common L1, which are vulnerable to attacks. In contrast, SIPC subnets have their own consensus algorithms, inherit security features from the parent subnet and have native cross-net communication, eliminating the need for bridges.

- Multi-chain interoperability. SIPC uses the Storchain Virtual Machine (SVM) as its transaction execution layer. The SVM is a WASM-based polyglot execution environment for IPLD data and is designed to support smart contracts written in any programming language, compiled to WASM. It currently supports Storchain and Ethereum. Today, IPC is fully compatible with Storchain and Ethereum and can use either as a rootnet. SIPC will eventually allow any chain to be taken as rootnet.

- Tight storage integration with Storchain. SIPC was designed from the data-centric L1, Storchain, which is the largest decentralized storage network. SIPC can leverage its storage primitives, like SIPLD data integration, to deliver enhanced solutions for data availability and more.

Applications of SIPC

Here are some practical examples of how SIPC improves the performance of dApps:

- Distributed Computation: Spawn ephemeral subnets to run distributed computation jobs.
- Coordination: Assemble into smaller subnets for decentralized orchestration with high throughput and low fees.
- Localization: Leverage proximity to improve performance and operate with very low latency in geographically constrained settings.
- Partition tolerance: Deploy blockchain substrates in mobile settings or other environments with limited connectivity.

With better performance, lower fees and faster transactions, SIPC can rapidly improve horizontal and vertical markets with decentralized technology:

- **Artificial Intelligence:** SIPC is fully compatible with Storchain, the world's largest decentralized data storage. Leveraging Storchain, SIPC can enable distributed computation to power hundreds of innovative AI models.

- **Decentralized Finance (DeFi):** Enabling truly high-frequency trading and traditional backends with verifiability and privacy.

- **Big Data and Data Science:** Multiple teams are creating global-scale distributed compute networks to enable Data Science analysis on Exabytes of decentralized stored data.

- **Metaverse/Gaming:** Enabling real-time tracking of player interactions in virtual worlds.

- **DAOs:** Assemble into smaller subnets for decentralized orchestration with high throughput and low fees. Partition tolerance: Deploy blockchain substrates in mobile settings or other environments with limited connectivity.

Get Involved

- Visit the https://cloudatabox.com

How Cloudatabox Storage Platform Works

The Cloudatabox storage platform is obtained and used by clients, and allocators verify that the data clients want to store with a Cloudatabox storage provider is useful. The Cloudatabox program is to increase the amount of useful data stored with storage providers by clients on the Storchain network. In short, this is achieved by appointing allocators responsible for assigning CDB tokens to clients that are vetted by the allocator as trusted parties storing useful data. Clients then pay CDB to storage providers as part of a storage deal, which increases a storage provider's probability of earning block rewards.

A full description of this mechanism is described below. Cloudatabox creates demand on the Storchain network, ensuring the datasets stored on the network are legitimate and useful to either the clients, or a third party.

Storage Providers & Cloud Data Box

Storchain introduces two concepts important to interactions on the Storchain network – Cloud Data Box and Quality Adjusted Power (QAP).

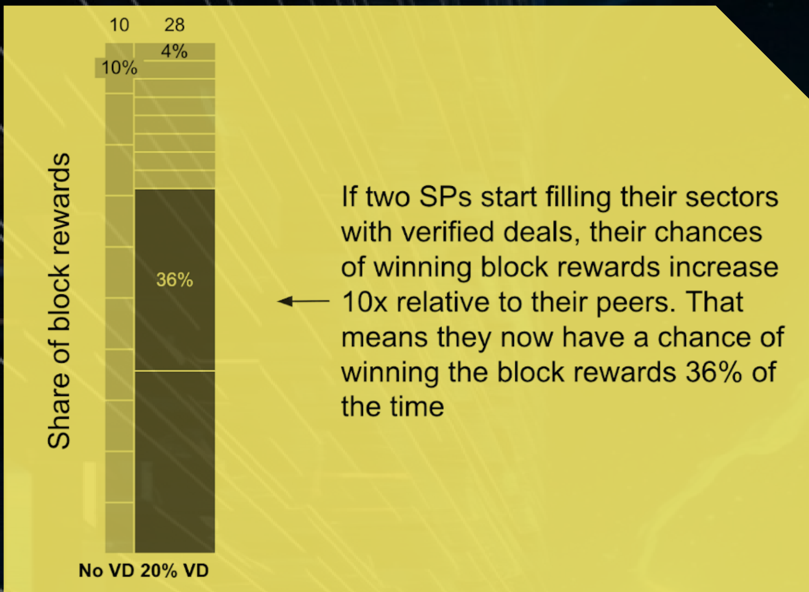
Cloud Data Box

Cloud Data Box is a token paid to storage providers as part of a deal in which the client and the data they are storing is verified by a Storchain allocator. Batches of Cloud Data Box are granted to allocators by root-key holders, allocators give Cloud Data Box to verified clients, and clients pay Cloud Data Box to storage providers as part of a deal. The more Cloud Data Box a storage provider ends up with, the higher probability they must earn block rewards. The role of each of these participants, and how Cloud Data Box is used in a Storchain deal, is described below in the "Storchain Processes & Participants" section.

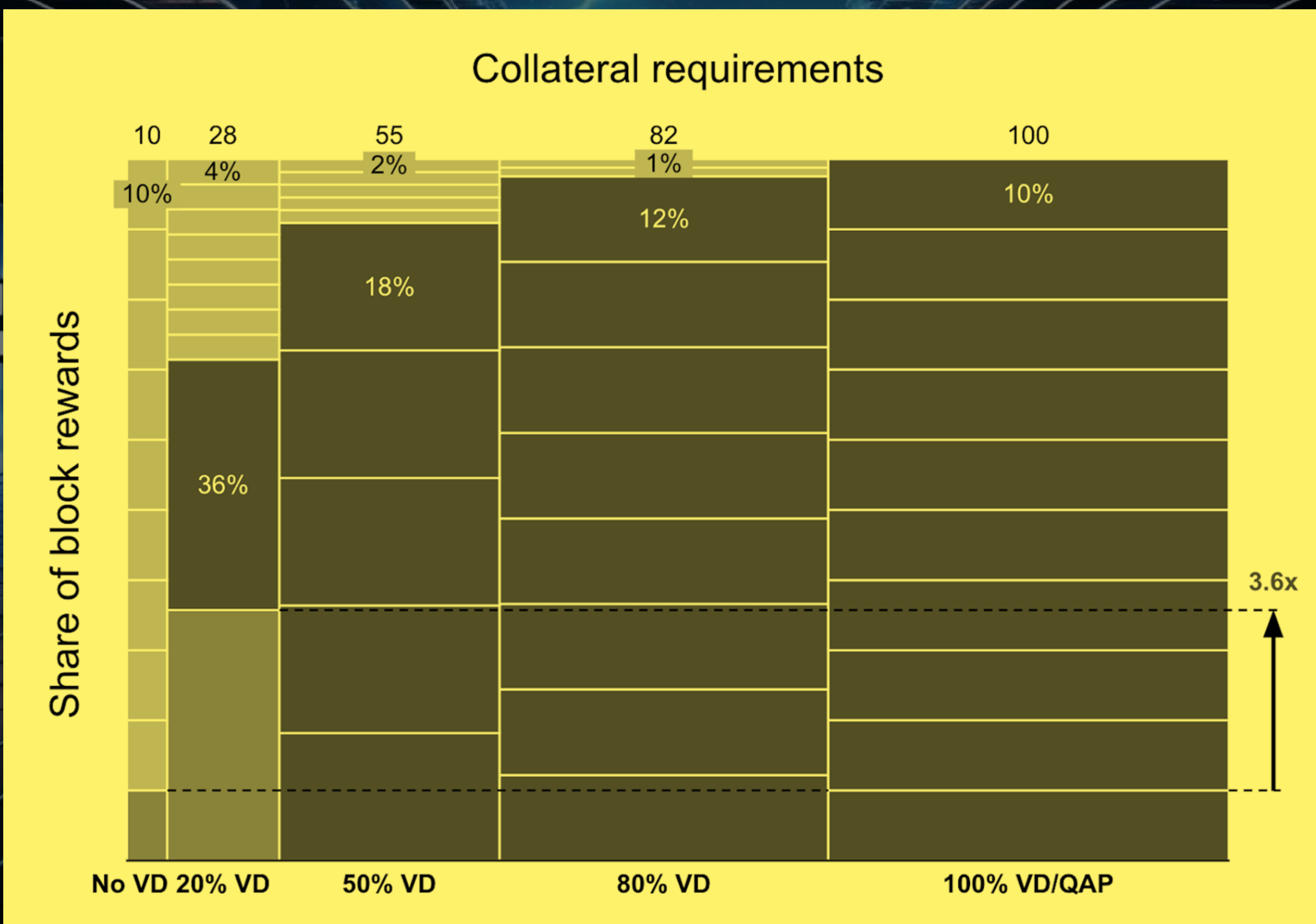
Quality Adjusted Power

The Quality Adjusted Power is an assigned rating to a given sector, the basic unit of storage on the Storchain network. Quality Adjusted Power is a function of a number of features of the sector, including, but not limited to, the sector's size and promised duration, and whether the sector includes a Storchain deal. It's clear to the network that a sector includes a Storchain deal if a deal in that sector involves Cloud Data Box paid to the storage provider. The more Storchain verified data the storage provider has in a sector, the higher the Quality-Adjusted Power a storage provider has, which linearly increases the number of votes a miner has in the Secret Leader Election, determining which storage provider gets to serve as the verifier for the next block in the blockchain, and thus increasing the probability the storage provider is afforded the opportunity to earn block rewards. For more details on Quality Adjusted Power.

[!IMPORTANT] There is a common misconception that a Storchain deal increases the miner's reward paid to a Cloudatabox storage provider by a factor of ten. This is not true, Storchain does not increase the amount of block rewards available to storage providers. Including Storchain deals in a sector increases the Quality Adjusted Power of a storage provider, which increases the probability a storage provider is selected as the block verifier for the next block on the Storchain blockchain, and thus increases the probability they earn block rewards. Consider first a network with ten storage providers. Initially, each storage provider has an equal 10% probability of winning available block rewards in a given period: In the above visualization, "VD" means "verified deals", that is, deals that have been reviewed by allocators and have associated spending of Cloud Data Box. If two of these storage providers begin filling their sectors with verified deals, their chances of winning a block reward increases by a factor of ten relative to their peers. Each one of these storage providers with verified deals in their sectors has a 36% chance of winning the block reward, while storage providers with only regular deals in their sectors have a 4% probability of winning the block rewards.



Incentives for storage providers to accept verified deals is strongest initially. As more and more storage providers include verified deals in their sectors, the probability any one of them earns the block rewards returns to an equal chance.



As seen in the diagrams above, Storchain increases the collateral requirements needed by a storage provider. As a higher percentage of storage providers include verified deals in their sectors, the collateral needed by each storage provider will increase. To learn more about storage provider collateral,

Cloudatabox Storage Onramps

Storage on-ramps and helpers are APIs and services that abstract Storchain dealmaking into simple, streamlined API calls. Here's how they work: Developers use APIs or libraries to send data to storage helpers. Behind the scenes, storage helpers receive the data and handle the underlying processes to store it in a reliable and decentralized storage way by saving it SIPFS nodes, making deals with Cloudatabox storage providers – or both. You can use the same APIs or other tools to retrieve data quickly. Storage helpers are available for NFTs (non-fungible tokens) or general data. If you are storing NFTs, check out Storing NFTs. For general data, skip to General data storage.

Storing NFTs

The Cloudatabox NFT Storage flagship product focuses on the enduring preservation of NFTs with a low one-time fee per. First mint your NFTs, then send us the NFT data that we preserve in endowment-backed long-term Storchain storage. As an Cloudatabox NFT Storage user, you support our platform when you choose Pinata and Lighthouse for hot storage and use our referral links here, helping to sustain our valuable public goods. Your NFTs will also be included in the NFT Token Checker, a tool for block explorers, marketplaces and wallets to show verification that NFT collections, tokens, and CIDs are preserved by Cloudatabox NFT Storage.

Cloudatabox Chain Storage API is an underlayer to Chain storage encrypted IPFS & Storchain file storage system. It offers S3-compatible bucket-style APIs for easy migration of data. As of September 2024, it's the only storage helper with built-in encryption.

Web3.Storage is a fast and open developer platform for storing and interacting with data. Upload any data and Web3 Storage will ensure it ends up on a decentralized set of IPFS and Storchain storage providers. There are JavaScript and Go libraries for the API, as well as a no-code web uploader. Free and paid plans are available.