CT3 Secure Storage Protocol Whitepaper

A Decentralized File Storage Architecture

with NFT-based Access, Miner Incentives,

and Dual-Token Tokenomics

Version 1.0 – May 2025

1. Introduction

1.1. The Future of Storage

The internet is entering a new era where centralized data silos are giving way to decentralized, user-owned infrastructure. Traditional cloud storage solutions, though efficient and scalable, are vulnerable to censorship, data loss, high recurring costs, and a lack of transparency. Decentralized storage networks (DSNs) aim to correct these imbalances by giving users control over their own data through distributed protocols and trustless infrastructure.

This whitepaper introduces **CT3 storage**, a decentralized storage platform that combines user simplicity, NFT-based access control, and future-ready token incentives. Our system is designed for secure, affordable, and scalable storage of encrypted files on the InterPlanetary File System (IPFS), with access rights embedded in dynamic non-fungible tokens (NFTs).

1.2. Project Overview

At its core, CT3 storage is a Web3 file storage protocol powered by:

- **NFT Access Tokens**: Every uploaded file is associated with a dynamically updating NFT, which acts as the access key and contains metadata such as storage duration, encryption status, and ownership.
- **Two-Token Economy**: A dual-token model ensures separation between system utility (payments, fees) and investment dynamics (rewards, incentives, staking), stabilizing the network′s core services.
- **IPFS Storage Backbone**: All files are stored encrypted on IPFS. This ensures permanence, global accessibility, and independence from central authorities.
- **Storage Miners**: Participants can pledge storage capacity and receive compensation for storing files, based on Proof-of-Access and Proof-of-Replication mechanisms.

The platform begins as a permissioned MVP with full interface-based interaction. All operations are routed internally, but the architecture is designed to migrate toward full decentralization with on-chain contracts, governance modules, and public market participation.

1.3. Problem Statement

Users currently face several barriers in decentralized file storage:

- Lack of intuitive interfaces
- Confusing access control or token mechanics
- Absence of flexible monetization models for storage providers
- File retrievability risks due to insufficient fault tolerance
- Unclear pricing mechanisms

CT3 storage addresses these by introducing an NFT-centric file access layer, dynamic fee scaling, miner incentives, and a transparent internal economy that is modular and ready for future on-chain migration.

1.4. Mission and Vision

Our mission is to create a seamless and sovereign file storage experience. We empower users with:

- **Self-custody** over their files
- **Automated access control** through NFTs
- **Secure storage** with miner-based encryption and redundancy
- **Sustainable economics** that reward participation and reliability
- **Usage-based pricing** that eliminates subscription fees, making storage more cost-effective

We envision a decentralized web infrastructure where users no longer rely on opaque SaaS platforms but interact directly with decentralized systems through familiar user flows.

1.5. Summary of Key Features

Feature	Description
Encrypted Storage	All files are stored client-side encrypted before upload
NFT File Keys	NFTs store access rights, metadata, and renewal options
Internal Token System	Users pay for file download and extension with platform tokens
Dual-Token Model	Separation of utility and investment ensures economic balance
Miner Network	Participants pledge storage and receive token rewards for uptime
Modular Upgrade Path	Designed to scale into full DAO and on-chain contract model

In the following sections, we provide a comprehensive overview of the system architecture, token economy, miner protocol, and governance mechanisms.

2. System Architecture

2.1 Overview

The CT3 storage network is designed around simplicity for users and modular robustness for long-term decentralization. Users interact with a Web3 application (dApp) through a connected crypto wallet (e.g., Phantom), enabling seamless login and personalized storage access. Behind the scenes, files are encrypted, uploaded to IPFS, and linked to a unique dynamic NFT that controls access and metadata.

There are three key actors in the architecture:

- **Client**: A user who uploads and retrieves files
- **Storage Miner**: Provides storage capacity and maintains file availability
- **Platform Interface / Coordinator**: A centralized controller in MVP stage, later replaced by smart contracts and decentralized verification nodes

2.2 File Upload Process

1. **User logs in** using Web3 wallet (e.g., Phantom).
2. **User uploads file** via drag-and-drop or explorer selection.
3. File is **encrypted client-side** with a unique symmetric key.
4. Encrypted file is uploaded to **IPFS**.
5. A unique NFT is minted containing metadata:
    - File name, format
    - Timestamp of upload and storage expiration
    - Indicator of encryption (□ if encrypted)
6. The NFT is delivered to the user′s wallet, acting as the sole access key.

2.3 File Retrieval Process

1. Platform detects user NFT and queries associated IPFS CID.
2. If storage is active, **file is fetched** from miner(s).
3. If encrypted, user is prompted for decryption password (future) or decryption handled via session key.
4. File is downloaded and decrypted locally.

2.4 File Renewal and Deletion

- Files are stored for a **base retention period** (e.g., 30 days).
- Users may **extend storage** via token payment; NFT metadata is updated to reflect new expiration.

- When a file is deleted:
  - NFT is burned or transferred to platform null address.
  - File is marked for deletion on miners.
  - Incentives may be returned for early deletion.

2.5 Miner Interaction

Storage miners in MVP interact via the admin panel:

- Register their available capacity
- Receive file shards and proofs
- Report uptime and fulfillment

In future versions:

- Automated proof systems (Proof-of-Access, Proof-of-Reliability) will verify uptime
- Storage pledges will be collateralized
- Distributed file shards will include redundancy and parity for fault tolerance

2.6 Roles and Responsibilities Table

Role	Responsibility
User	Uploads, retrieves, extends, and deletes files via dApp
Miner	Provides storage, uptime, and reliability in exchange for utility tokens
NFT Contract	Maintains file metadata, access rights, and expiration info
Coordinator	Processes user operations and orchestrates IPFS + NFT flow

2.7 Architecture Diagram & Protocol Diagrams

Proof-of-Replication Flow

```
Client ---------> Miner ---------> IPFS Network
  |                                |
  |  Upload Encrypted File          |
  |---------------------------------->|

Coordinator ---------> Miner
  |   Assign Storage Order + Schedule Proof
  |---------------------------------->|

[Proof Interval Cycle]:

Miner ---------> Coordinator
  |   Submit Merkle Root Proof with Timestamp + Signature
  |---------------------------------->|

Coordinator
  |   Validate Proof; Update Miner Reputation; Release Partial Rewards
```

Proof-of-Access Flow

```
Coordinator / Verifier ---------> Blockchain VRF Oracle
  |  Request Random Byte Offset for File CID
  |---------------------------------------------->|

Blockchain VRF Oracle ---------> Coordinator / Verifier
  |  Return Random Offset b
  |---------------------------------------------|

Coordinator / Verifier ---------> Miner
  |  Request PoA: Hash(D[b]) for Offset b
  |------------------------------------------->|

Miner ---------> Coordinator / Verifier
  |  Submit Hash(D[b]) + Merkle Path Proof
  |---------------------------------------->|

Coordinator / Verifier
  |  Validate PoA Response; Update Miner Reputation; Apply Penalties if Needed
```

Storage Order Lifecycle

```
Client ---------> Coordinator
  |  Upload File + Metadata
  |------------------------->|

Coordinator ---------> NFT Contract
  |  Mint Access NFT + Metadata
  |------------------------------------------->|

Coordinator ---------> Miner(s)
  |  Assign Storage Order with Redundancy Level k
  |---------------------------------------------->|

Miner(s)
  |  Store Encrypted File Segments; Initialize Proof-of-Replication Schedule
  |---------------------------------------------->|

Coordinator ---------> Client
  |  Return Access NFT to Wallet
  |<----------------------------------------|
```

Retrieval Lifecycle

```
Client (Wallet) ---------> Coordinator
  |  Request File Retrieval (check NFT ownership)
  |---------------------------------------------->|

Coordinator ---------> Miner
  |  Request File Content (check active Storage Order)
  |---------------------------------------------->|

Miner ---------> Coordinator
  |  Deliver Encrypted File Segment(s)
  |---------------------------------------------->|

Coordinator ---------> Client
  |  Deliver Encrypted File to Client
  |<----------------------------------------------|

Client
  |  Decrypt File Locally (future); Download
  |-----------------------------------------------|
```

3. Storage and Retrieval Market

3.1 Market Overview

CT3 storage introduces a bifurcated economic architecture composed of two principal components: the **Storage Market** and the **Retrieval Market**. These markets operate under distinct incentive structures yet maintain an interdependent relationship within the ecosystem.

- The **Storage Market** governs long-term data persistence. Clients (data uploaders) enter into storage agreements with storage miners, specifying terms such as file size, duration, and redundancy.
- The **Retrieval Market**, by contrast, emphasizes performance — it matches data requesters with retrieval providers that deliver specific pieces of content efficiently.

This architecture follows the precedent established by decentralized secure storage, but adapts it to a more user-oriented model with simplified internal coordination during the MVP phase. Future versions will support full on-chain transaction logging and decentralized negotiation of deals.

3.2 Storage Market Mechanics

Storage orders are generated each time a file is uploaded. Orders are internally matched with miners offering sufficient capacity and meeting minimum service reliability thresholds. A typical **Storage Order** includes the following fields:

Field	Description
CID	Content Identifier hash of the file
Size	File size in kilobytes
Redundancy	Number of copies to be stored (1–3)
Duration	Contract length in days (30, 60, 90, etc.)
NFT_ID	Access control NFT bound to the file

Price_per_KB_day	Platform-defined base rate for storage
Miner_ID	Assigned storage provider
Status	Active, Renewed, Expired, Deleted

Payment Model

Payments are made using the Utility Token ($CT3GB), with fees calculated according to the following base formula:

$$F_{store} = S_{KB} \cdot D_{days} \cdot R_{base} \cdot (1 + \rho \cdot (k - 1))$$

Where:

- S_{KB} — file size in kilobytes
- D_{days} — number of storage days
- R_{base} — base rate (e.g., 0.000015 CT3GB/KB/day)
- k — redundancy level (number of copies)
- ρ — redundancy multiplier (suggested: 0.15)

The base storage fee includes not only file size and duration, but also a redundancy multiplier based on how many independent nodes store the data.

Incentives and Penalties

To ensure compliance and discourage negligence, the protocol enforces the following mechanisms:

- **Uptime Proof Rewards**: Every 24h, miners submit a proof of file retention. If valid, a daily portion of their reward is unlocked.
- **Penalty Function**: Missed proofs lead to slashing of pledged tokens.
- **Collateral Locking**: High-value deals require storage miners to stake collateral tokens, which are forfeited on failure.

Let be the reward payout function over time:

$$P(t) = \begin{cases} \frac{F}{D}, & if\, proof(t) = valid \\ 0, & otherwise \end{cases}$$

Where:

- F — total payment for the contract
- D — duration in days

3.3 Retrieval Market Mechanics

Retrieval occurs when the NFT holder initiates a download request. Unlike storage contracts, retrieval operations are stateless, and retrieval miners are compensated per delivered kilobyte.

Retrieval pricing uses a **progressive fee growth** model to disincentivize hoarding of stale data and promote timely downloads:

$$F_{retrieve}(d) = S_{KB} \cdot R_{ret} \cdot \left(1 + r \cdot log_2(d+1)\right)$$

Where:

- S_{KB} — file size in KB
- R_{ret} — base retrieval rate (e.g., 0.00002 CT3GB/KB)
- r — logarithmic growth coefficient (e.g., 0.05)
- d — number of days since upload

3.4 Storage Lifecycle Protocol

1. **File Upload**:
    - Client encrypts and uploads file
    - CID generated, NFT minted, order matched to a miner
    - Payment locked; initial proof requested within 24h
2. **Retention Phase**:

- Miner submits daily Proof-of-Replication
- Rewards streamed to miner wallet on valid proof

3. **Access Request**:
    - Client requests file using NFT
    - Miner delivers content, receives retrieval fee
4. **Extension or Deletion**:
    - User may extend contract
    - On deletion, partial refund logic applies

3.5 MVP vs Future Protocol Table

Feature	MVP Version	Decentralized Version
Order Matching	Platform-matched	Smart contract / open orderbook
Proof Verification	Local mechanisms mirroring Proof-of-Replication principles	Automated Proof-of-Reliability
Payment	Polygon via connected wallet	On-chain micropayment channels
Miner Registration	Whitelisted via panel	Self-service + slashing mechanism
NFT Ownership Logic	Polygon contract binding	CT3-based NFTs with on-chain metadata

3.6 Step-by-Step Protocol Overview

To improve clarity and readiness perception, the system's behavior is defined in discrete procedural steps for both file storage and retrieval.

3.6.1 File Upload Procedure

1. **User Authentication**
    - User connects their wallet via Web3 (e.g., Phantom).

- Interface initializes user session.

2. **File Selection & Encryption**
    - User selects a file through browser or drag-and-drop interface.
    - File is encrypted client-side with AES-256 using a session key.
3. **Upload & NFT Minting**
    - Encrypted file is uploaded to IPFS, returning a unique CID.
    - Platform creates a dynamic NFT containing:
        - Format
        - Upload timestamp
        - Initial storage duration
        - Optional encryption flag (future)
    - NFT is minted and sent to the user′s wallet.
4. **Storage Order Generation**
    - A Storage Order is created, pairing CID to miner(s) with sufficient capacity.
    - Contract includes duration, redundancy, and cost parameters.
    - Miner receives file shards and confirms deal.
5. **Token Settlement**
    - Fee calculated: $F_{store} = S_{KB} \cdot D_{days} \cdot R_{base} \cdot (1 + \rho \cdot (k - 1))$
    - $CT3GB is deducted from user balance.
    - Tokens are locked into a deal pool, unlocked for miners upon proof validation.

3.6.2 File Retrieval Procedure

1. **Access Request**

- User opens "My Files" dashboard.
- Platform checks for associated NFT in wallet.

2. **Download Request**
    - User initiates download.
    - Platform verifies NFT and identifies active miner storing CID.
3. **Decryption Check**
    - If encrypted, user is prompted for decryption key/password.
    - Decryption is performed client-side post-download.
4. **Retrieval Fee Calculation**
    - Fee: $F_{retrieve}(d) = S_{KB} \cdot R_{ret} \cdot \left(1 + r \cdot log_2(d+1)\right)$
    - Payment in $CT3GB processed immediately.
    - Miner is rewarded instantly with corresponding portion.
5. **Burn and Settlement**
    - Portion of $CT3GB is burned (e.g., 20%).
    - Remainder distributed as per allocation table.

3.6.3 Early Deletion Procedure (Planned)

1. User selects file to delete from dashboard.
2. Confirmation dialog with optional reason.
3. NFT is burned (or sent to null address).
4. Remaining file marked for purge on miner node.
5. Partial refund issued based on unused time.

3.6.4 Miner Workflow

1. **Register Storage Capacity** via dashboard.
2. **Receive Storage Order** based on reputation & availability.
3. **Store File Shards** and initiate proof schedule.
4. **Submit Proofs** regularly (daily or hourly).
5. **Receive CT3GB Rewards** upon successful validation.
6. **Penalties** if failure to prove (slashing, collateral seizure).

4. Token Economy

4.1 Overview of the Dual-Token Model

The token economy of CT3 storage is designed around a **dual-token architecture**. This separation allows for the stabilization of operational costs (storage and retrieval) while providing a flexible and growth-oriented framework for investment and platform expansion.

Token Type	Name (Placeholder)	Purpose	Supply Logic
Utility Token	$CT3GB	Payments for storage, retrieval, renewals	Elastic / Controlled
Investment Token	$CT3	Governance reserve, staking, ecosystem growth	Capped / Fixed

4.2 Utility Token ($CT3GB)

The **Utility Token** is the core medium of exchange within the CT3 storage network. It is used for:

- Paying for storage space and duration
- Paying for retrieval actions
- Extending file lifetimes
- Rewarding miners for uptime and performance
- Refunds for unused storage space
- Collateral payments by miners

Emission and Distribution

Unlike pre-mined tokens, $CT3GB follows a **controlled emission** strategy tied to:

- Onboarding of new users (airdrop-style)
- System-driven distributions for miner incentives
- Token recycling through burning and rewards

Let the net token supply at time be:

$$S(t) = S_0 + E(t) - B(t)$$

Where:

- $S(t)$ — total circulating supply
- S_0 — initial supply
- $E(t)$ — tokens emitted (e.g., user reward, staking interest)
- $B(t)$ — burned tokens in the period

Example Emission Control Parameters

Parameter	Value	Description
Initial Supply	1,000,000 CT3GB	Minted to treasury + airdrops
Daily Emission Cap	5,000 CT3GB	Emitted for miner rewards and user onboarding
Burn Rate (Downloads)	20%	Of all retrieval fees
Burn Rate (Penalties)	50%	From slashing underperforming miners
Inflation Target	≤ 2% / year	Controlled via governance-based emission pacing

Clarification:

- Emission is determined by onboarding programs, user activity, and miner rewards.
- Burn sources include:
    - 20–30% of download fees
    - 30–50% of expired storage
    - Up to 50% of penalties for unavailability

This introduces a **deflationary pressure**, particularly under heavy usage scenarios.

Token Price Anchor (Optional)

To avoid volatility in service pricing, the protocol may implement a **price oracle mechanism** that anchors $CT3GB to a fiat reference (e.g., USD-equivalent floor for 1GB/month).

Demand–Supply Control Model

To preserve price stability, a soft cap on circulating supply is maintained by balancing:

$$NetEmission(t) = E(t) - B(t)$$

If $NetEmission(t) > T_{max}$ governance may:

- Reduce miner reward multipliers
- Increase burn coefficients
- Temporarily raise fees to restore balance

4.3 Investment Token ($CT3)

The Investment Token functions as the strategic and speculative layer of the ecosystem.

Primary Roles

- **Investor Entry**: $CT3 is offered in funding rounds, exchange listings, or liquidity events
- **Revenue Support**: A portion of ecosystem fees are swapped into $CT3 as reserve backing
- **Ecosystem Expansion**: Used for infrastructure grants, marketing incentives, and strategic partnerships

Tokenomics Parameters (Example Model)

Parameter	Value
Total Supply	10,000,000 CT3
Initial Circulating	1,000,000 (10%)
Vesting Schedule	48-month linear unlock

Treasury Allocation	30%
Public/Private Rounds	40%
Team + Advisors	15% (cliff + vesting)
Ecosystem Growth	15%
Conversion Mechanism	DAO-controlled CT3→CT3GB conversion to stabilize CT3GB price

Reward Anchoring

To align incentives, a fraction of download/storage fees may be converted to $CT3 on-market, simulating buyback-and-hold mechanics:

$$C_{CT3} = \alpha \cdot F_{net}, \quad \alpha \in [0,1]$$

Where:

- C_{C3T} — conversion into CT3 (buyback)
- F_{net} — net platform fees collected
- α — buyback ratio (e.g. 0.25)

4.4 Developer Incentive and Team Reserves

In addition to the foundational token allocation table, CT3 storage introduces a structured incentive mechanism for both core contributors and external development teams. The long-term success and sustainability of the protocol heavily depend on attracting, motivating, and retaining technical talent capable of delivering secure, scalable, and user-friendly systems.

Team Reserve (10%)

This allocation is reserved for the founding contributors and early architects of the CT3 storage protocol. The reserve is subject to a 48-month linear vesting schedule, with a 12-month cliff, and is held in a multi-signature smart contract controlled by the governance framework upon DAO activation.

This pool reflects the initial commitment and long-term alignment of the core team with the protocol's growth.

Developer Incentive Pool (5%)

To address the evolving needs of the ecosystem, an additional 5% of total supply is allocated as a Developer Incentive Pool, which will be managed transparently by the DAO. This pool is designed to support:

- Protocol-level enhancements (e.g., PoA and PoR implementation, mobile clients)
- Third-party integrations and tooling (e.g., SDKs, CLI tools, Unity plugins)
- Security audits and bounty-driven development
- Frontend applications and cross-platform utilities
- NFT smart contract optimization and deployment

Grants will be distributed based on open proposals, milestone-based disbursement, and community voting. The goal is to ensure the protocol evolves in a modular, decentralized, and developer-driven manner.

This dual-incentive structure (Team Reserve + Builder Pool) aims to combine accountability with openness — rewarding early contributors while creating space for future innovation.

4.5 Mining Rewards and Fee Allocation

The combined economy redistributes value to active participants using a predefined logic. Example allocation per retrieval payment:

Allocation Target	% of Fee Paid	Notes
Storage Miner	40%	Linear over uptime proofs (redirected to Treasury/DAO if no miner present)
Retrieval Miner	25%	Immediate on delivery (becomes active in Phase 2 with retrieval mining)
Token Burn	20%	Deflationary effect
Reserve to Treasury	10%	Buyback mechanism + stabilization
Community Pool / DAO	5%	Future use in grants or governance

Until decentralized miners are actively participating, their share of fees may be temporarily redirected to the DAO treasury or reserve pool. This ensures continuity of funding and network sustainability during early adoption.

$$R_p = \frac{P_{total}}{D \cdot f}$$

The reward per successful proof is computed by dividing the total payment P_{total} by the total number of expected proofs over duration D with frequency f proofs/day.

Return on Investment (ROI) for Miners:

$$ROI_m = \frac{\sum_{t=1}^{D} R_t}{O + C} - 1$$

Where:

- R_t — reward at time t
- C — collateral or operational cost

This formula measures return on investment for miners. R_t is the total daily reward, O the operational cost, and C the collateral stake.

4.6 User Lifecycle with Tokens

1. **User Registration**
    - Receives onboarding bonus in $CT3GB
    - Wallet linked to user alias
2. **File Upload**
    - Pays storage cost in $CT3GB
    - NFT is minted and sent to wallet
3. **Storage Miner Reward**

    - Uptime proofs generate $CT3GB stream
    - Penalties enforced via slashing if missed
4. **File Retrieval**
    - User pays dynamic retrieval fee
    - Retrieval miner is compensated; burn enacted
5. **Token Conversion (Optional)**
    - Treasury purchases $CT3 for locking / staking
    - Conversion boosts $CT3 floor value

5. Cryptographic Guarantees and Access Control

5.1 Overview

The CT3 storage system is architected around the principle of **data sovereignty**, where users retain exclusive control over access to their uploaded files. This is achieved through the combination of client-side encryption, NFT-based access keys, and trust-minimized storage mechanics.

In this section, we outline the cryptographic primitives, encryption flows, NFT metadata logic, and the architecture of file access and deletion.

5.2 Encryption Model

File-Level Encryption

Before any file leaves the user's device, it is encrypted locally using symmetric key cryptography (AES-256 standard). This ensures that storage providers (miners) have no visibility into the file content.

- **Symmetric Key**: A randomly generated 256-bit key per file
- **Initialization Vector (IV)**: Unique IV generated per encryption session
- **Encryption Algorithm**: AES-GCM or AES-CBC with PKCS#7 padding

Let:

- F = original file data
- K = symmetric encryption key
- IV = initialization vector
- C = encrypted ciphertext

Then:

$$C = Encrypt_{AES}(F, K, IV)$$

The encrypted output is then uploaded to IPFS, and the decryption key is stored locally in the browser or optionally within secure wallet storage.

Decryption Flow

Upon download, the file is decrypted client-side:

$$F = Decrypt_{AES}(C, K, IV)$$

If password-based encryption is enabled, the symmetric key itself is derived via PBKDF2 from user input:

$$K = PBKDF2(password, salt, iterations)$$

5.3 NFT Access Token Logic

Every file uploaded results in the minting of a unique NFT that functions as an access controller. The NFT stores dynamic metadata necessary for validating ownership and expiration.

NFT Metadata Structure

Field	Description
File Name	Human-readable name
IPFS CID	Content Identifier (future)
Upload Date	Timestamp
Expiry Date	Deadline for storage

Encrypted Flag	Yes / No (future)
Lock Icon	UI representation (e.g., 🔒 if encrypted) (future)

Ownership Logic

Access to a file is permitted only if the wallet initiating the request holds the corresponding NFT.

- NFT is bound to user wallet address.
- Transfer of NFT changes access rights.
- Revocation (deletion) burns the NFT or transfers to a null address.

Smart Contract (Future Version)

In later versions, NFTs will be minted via CT3 on-chain smart contracts with full on-chain metadata verification and expiration enforcement.

5.4 Trustless Deletion

To protect users′ intent to remove files permanently, the protocol enables:

1. **NFT Burn / Transfer**:
    - NFT is sent to null address or burned
    - Platform removes the file from index
2. **Miner Purge Signal**:
    - Platform notifies associated miners
    - File is flagged for deletion from storage nodes
3. **Future: ZK-verifiable Deletion**:
    - Zero-knowledge proof submitted that file no longer exists

- Enables provable compliance without data exposure

5.5 Security Model Summary

Security Component	Methodology
File Confidentiality	AES-256 client-side encryption
File Authenticity	IPFS CID content hashing
Access Control	NFT-bound ownership
Decryption Authorization	Local key or password-derived key
File Deletion	NFT burn + miner purge + future ZK proof

5.6 Threat Model & Mitigations

The CT3 storage protocol considers the following key threat vectors:

Threat	Description	Mitigation
Replay Attack	A malicious actor attempts to replay old proofs to avoid fresh verification	Proofs include timestamps and nonces; signatures validated with strict freshness window
Withheld File Attack	Miner accepts storage order but withholds actual file segments	Proof-of-Access (PoA) challenges random byte offsets; failure leads to slashing and reputation penalties
Sybil Attack	Single actor spins up many miner identities to manipulate market or proofs	Collateral staking required per identity; initial reputation score decay limits short-term rewards
Collusion Attack	Group of miners coordinate to fake proofs	Use of random PoA challenges and requirement of Merkle-based proof per miner; open slashing audits planned post DAO activation
DoS Attack	Malicious actor floods storage requests or retrieval requests to exhaust miner resources	Miner request rate limiting; penalty for failed proofs discourages fake storage; retrieval throttling on miner nodes

Governance Centralization	Small group dominates governance decisions	Voting weight decay mechanisms; reputation-based governance; proposal cooldown windows

Future enhancements will introduce ZK-based Proof-of-Deletion and sybil-resistant participation scoring to further strengthen protocol resilience.

6. Miner Protocol and Proof Systems

6.1 Role of Miners in the Network

Miners form the decentralized infrastructure of the CT3 storage storage network. They contribute physical storage capacity, maintain file availability, and uphold the reliability of the ecosystem through cryptographic proofs. In return, miners receive token rewards in $CT3GB for successful participation.

There are two primary miner types:

- **Storage Miners**: Maintain encrypted files on their local systems.
- **Retrieval Miners**: Serve download requests by delivering file content efficiently.

While roles may overlap, the network incentivizes specialization through differentiated reward logic. A miner may opt into one or both markets depending on hardware capabilities, uptime quality, bandwidth, and fee preferences.

6.2 Registration and Onboarding

During the MVP phase, miners register through a centralized dashboard and agree to the following parameters:

- Minimum guaranteed storage capacity (e.g., 1 GB)
- Maximum number of parallel storage contracts
- Daily proof submission agreement

Upon migration to the decentralized version:

- **Miner Identity** is cryptographically bound to a wallet public key (PK)
- **Collateral Requirement** enforces minimum stake per gigabyte of committed storage

- **Reputation Score** is initialized at 100 and adjusted based on proof history, success ratio, and latency metrics

Reputation is formalized as:

$$R_m(t) = 100 - \gamma_1 f_m(t) - \gamma_2 l_m(t)$$

Where:

- $f_m(t)$: total proof failures at time
- $l_m(t)$: average latency in seconds
- γ_1, γ_2 : penalty weight coefficients

6.3 Storage Contract Lifecycle

Each storage interaction follows a contract protocol with the following stages:

1. **Offer Matching**
    - Platform matches file order to eligible miners with required capacity and availability.
2. **Sharding (Optional)**
    - File may be divided into segments for parallel storage across multiple nodes, each with own CID.
3. **Data Replication**
    - Redundancy level $R \in \{1, 2, 3\}$ determines how many independent nodes must store the same file.
4. **Proof Schedule Agreement**
    - Each miner signs an agreement to submit proofs every τ hours.
5. **Smart Locking of Rewards**
    - Payment P is locked into a smart pool and incrementally unlocked on each valid proof.

6.4 Proof-of-Replication (PoR)

To confirm file replication and presence, a **Merkle-tree-based Proof-of-Replication** is used.

Let:

- $D_1, D_2, \ldots, D_n$: file data chunks
- $H(D_i)$: hash of chunk i
- M : Merkle root over all chunks

$$M = MerkleRoot(H(D_1), H(D_2), \ldots, H(D_n))$$

Miner submits signed proof:

$$PoRt = Sign_{PK_m}(M \parallel CID \parallel t)$$

Verifier node checks validity by:

- Recomputing M
- Checking timestamp freshness $|t_{now} - t| < \Delta t_{max}$
- Verifying signature against PK_m

Each proof is signed by the miner using their private key, covering Merkle root, content ID, and timestamp — ensuring authenticity and freshness.

6.5 Proof-of-Access (PoA) — Future Expansion

To guarantee file retrievability:

$$PoA_s = H(D_j), j = Random(1, n)$$

- A pseudo-random byte offset b is generated via on-chain VRF
- Miner returns hash of segment at b
- Verifier checks consistency with Merkle path to M

This mechanism ensures the miner truly stores the retrievable file without full recomputation:

$$PoA_b = H(D_b)\, where b \sim VRF(CID, t)$$

6.6 Slashing and Penalties

Misbehavior is penalized via time-based and collateral-based mechanisms:

$$Penalty(t) = \delta \cdot C \cdot \left(1 + \frac{f(t)}{f_{max}} + \frac{1}{1 + e^{-\lambda \cdot (u_t - 0.95)}}\right)$$

Let:

- δ: base slashing rate (0.25)
- C: staked collateral
- $f(t)$: number of failed proofs in period
- f_{max}: max allowed failures
- u_t: uptime percentage over last 24h
- λ: penalty steepness (default: 10)

Upon proof failure, miners are penalized by losing a fraction δ (e.g., 0.25) of their staked collateral C, ensuring financial accountability.

$$R_i(t+1) = R_i(t) - \beta \cdot f_t - \lambda \cdot l_i$$

Miner reputation decreases based on the number of failed proofs f_t and latency l_i. Parameters β and λ control punishment severity.

Events triggering penalty:

- f_1 : missing proof window (1 per τ period)
- f_2 : corrupted segment detection (via PoA)
- f_3 : uptime < 95% over trailing 24h window

The slashing threshold θ_s is calculated dynamically based on average platform reliability $\overline{R}_{net}$:

$$\theta_s = \mu \cdot (1 - \overline{R}_{net})$$

6.7 Summary of Miner Protocol Elements

Component	Methodology
Storage Verification	Merkle-root-based Proof-of-Replication
Access Verification	Segment challenge via Proof-of-Access (planned)
Miner Identity	Wallet address / public key
Reward Distribution	Streamed per valid proof
Slashing Conditions	Based on proof failure or fraud
Reputation System	Weighted decay based on latency and proof misses
Governance Status	No vote rights (read-only role in MVP)

6.8 Parameters Specification

Parameter	Description	Recommended Range	Default Value
$\rho(rho)$	Redundancy premium coefficient (affects storage fee)	1.0 — 3.0	1.5
r	Retrieval fee growth coefficient (controls retrieval cost increase over time)	0.03 — 0.07	0.05
$\delta(delta)$	Slashing multiplier (fraction of collateral slashed per violation)	0.1 — 0.5	0.25
$\gamma 1$	Reputation penalty weight for failed proofs	0.5 — 5.0	2.0

$\gamma 2$	Reputation penalty weight for latency violations	0.5 — 5.0	1.0
$\lambda(lambda)$	Penalty function steepness (controls how sharply penalties increase near threshold)	5 — 15	10
ut	Uptime target threshold (minimum % uptime required)	95% — 99%	95%

7. Governance and Roadmap

7.1 Governance Philosophy

CT3 storage is founded on the principle that decentralized data ownership must eventually be matched by decentralized decision-making. Our governance framework is designed not only as a system of control but also as a dynamic mechanism to distribute agency, align incentives, and sustain the network's longevity. We believe that users, miners, developers, and token holders each possess a unique perspective essential to the health of the protocol.

In the early phases of the project, centralized coordination enables rapid iteration, security hardening, and agile feature development. However, from the onset, the architecture is built with governance modularity, anticipating future migrations into a DAO-managed environment. Our end goal is a storage protocol governed transparently, securely, and collectively.

Governance must balance:

- **Stability** – ensuring that infrastructure upgrades, miner slashing, and fee changes are deliberate and predictable.
- **Adaptability** – allowing for proposals that can respond to technical innovation, attack surfaces, or market shifts.
- **Transparency** – with publicly recorded discussions, proposals, and vote outcomes.

A dual-token model assists in this vision: utility operations are handled by $CT3GB, while $CT3 evolves into the governance lever for the protocol's future.

7.2 Governance Participants (Future DAO Roles)

We define several categories of governance participants that will emerge as the protocol transitions toward decentralization:

Role	Description
Token Holder	Any address holding staked $CT3 is entitled to vote on governance issues. Voting power is proportional to stake size and maturity.
Proposal Submitter	Addresses meeting a minimum threshold of $CT3 (e.g., 1,000 tokens) may submit governance proposals with attached metadata, rationale, and proposed changes.
Validator Node	These entities execute governance outcomes, validate state transitions, and enforce consensus post-vote. In early phases, this role is filled by protocol signers.
Treasury Agent	DAO-elected roles that administer allocation of funds from the protocol treasury to development, grants, and community programs.

Stakeholder participation will be incentivized via staking yields, proposal submission rewards, and weighted reputation increases.

7.3 Governance Mechanics (Planned)

As governance is introduced, a phased rollout of community control mechanisms is anticipated. The process includes:

Proposal Lifecycle

1. **Submission**: Any eligible holder submits a draft proposal, including title, problem statement, proposed action, and rationale.
2. **Discussion**: A 5- to 7-day open period for public comment via governance forums or Snapshot-based polls.
3. **Voting**: Token-weighted voting occurs through verified wallet signatures. Quorum rules ensure proposal legitimacy.

4. **Execution**: If passed, proposal is executed on-chain (or via signers during transitional phase).

Voting Weight Formula

To encourage long-term commitment and discourage whale attacks, voting power W_i is computed as:

$$W_i = S_i \cdot A_i \cdot T_i$$

Where:

- S_i : number of $CT3 staked by user
- A_i : age multiplier based on how long tokens have been staked
- T_i : trust coefficient based on participation score

Categories of Proposals

- **Protocol Upgrades**: core changes such as consensus modifications or upgrade windows.
- **Economic Adjustments**: rebalancing storage fees, reward ratios, token burn coefficients.
- **Treasury Decisions**: grant programs, team funding, and contributor bounties.
- **Miner Incentive Tweaks**: adjusting uptime reward weights, proof thresholds, or reputation models.

All proposals and votes will be made public through an integrated governance portal.

7.4 Roadmap Phases

The development of CT3 storage is structured into progressive stages toward full on-chain decentralization and the adoption of Proof-of-Access (PoA) and Proof-of-Replication (PoR) consensus mechanisms.

Phase 1 – MVP Launch (Q2 2025)

- Web3 interface, NFT minting, centralized PoR proofs, $CT3GB logic, miner dashboard

Phase 2 – Network Scaling (Q3–Q4 2025)

- Redundancy layer, partial file sharding, Merkle proof automation, retrieval miner tracking, basic DAO preparation

Phase 3 – Token Listing and DAO Preparation (Q2-Q3 2026)

- CT3 listing, governance dashboard, staking module, audit committee

Roadmap Addition — Builder Grant Program

Phase 1: Builder Grants (Pre-DAO) – Q3–Q4 2025

Launch of structured grant program targeting critical implementation areas such as:

– NFT contract rework

– Phantom wallet compatibility

– Bubblegum compressed NFT migration

– Unity SDK

Grants will be milestone-based and paid from the Developer Incentive Pool.

Phase 3: DAO Preparation – Q1 2026

Governance dashboard, staking module, DAO roles, proposal mechanics

Phase 4 – On-Chain Migration (Q3–Q4 2026)

- Full migration of Proof-of-Replication and Proof-of-Access mechanisms on-chain; decentralized storage order matching; removal of centralized coordinator

Phase 5 – DAO Activation (Q1 2027)

- Full on-chain DAO governance; treasury contract activation; decentralized parameter governance

Future — Privacy & Cross-Chain (2027+)

- Zero-knowledge file deletion proofs, ZK-NFT access tokens, cross-chain NFT bridges

Important Note:

As of MVP (Q2 2025), CT3 storage operates on a permissioned network where Proof-of-Access and Proof-of-Replication processes are verified internally.

After a one-year test cycle (through Q2 2026), all core storage and access verification operations will migrate fully on-chain, with open verifiability and decentralized enforcement.

7.5 Governance Security Considerations

In a decentralized future, protocol integrity must be preserved. The DAO governance layer will incorporate security mechanisms to minimize risk:

- **Execution Delay**: 24–48 hour delay between proposal passage and on-chain execution
- **Challenge Window**: A grace period during which critical proposals may be vetoed or suspended for review
- **Multi-Signature Safeguards**: Emergency override by designated guardians for compromised votes
- **Sybil Resistance**: ZK-based or identity-anchored participation scores to prevent gaming
- **Rate Limits**: Proposal submission cooldowns and quorum minimums for low-volume weeks

These guarantees ensure that the community governs not only effectively, but securely.

8. Conclusion and Strategic Outlook

8.1 Final Summary

CT3 storage emerges as a next-generation decentralized storage ecosystem that unites robust cryptographic principles, NFT-based access control, dynamic token economics, and a roadmap toward full decentralization. Throughout this whitepaper, we have outlined a comprehensive architecture that is both technically feasible and economically sustainable, while maintaining flexibility for iteration and governance transformation.

By combining the best aspects of decentralized networks — such as content addressing, permissionless participation, and transparent governance — with a focus on usability and incentive design, the platform aims to become a foundational layer of the decentralized web.

The dual-token model ensures the uncoupling of operational functionality from speculative behavior, providing long-term viability while opening new pathways for community governance, staking, and public ownership.

8.2 Strategic Positioning

CT3 storage distinguishes itself from other decentralized storage solutions through:

- **NFT-Centric Access Control**: Our use of dynamic NFTs as access keys introduces a flexible, modular, and user-friendly method for managing digital rights.

- **Integrated Token Sink Design**: Controlled burning, micro-payment flows, and conditional refunding create an economic feedback loop that is anti-inflationary and incentive-aligned.
- **Progressive Miner Ecosystem**: A tiered reputation, collateral-backed rewards, and dual-proof systems (PoR/PoA) ensure high performance, fault tolerance, and incentive compatibility.
- **Seamless User Experience**: Wallet-based authentication, one-click uploads, and automatic renewals eliminate the complexity that often hinders decentralized platforms.
- **Planned Governance Maturity**: Rather than launching with a fragile DAO, CT3 storage follows a governance progression model that strengthens over time, rooted in real stakeholder activity.

8.3 Challenges and Mitigations

Despite its strengths, the project must overcome several inherent risks:

Risk	Description	Mitigation Strategy
Cold Start Problem	Low miner participation at early stages	Bootstrap rewards, flexible onboarding requirements
User Drop-Off	Complexity in crypto UI/UX	Mobile-friendly interfaces, tiered onboarding
Miner Malfeasance	Fake storage, failed proofs, or collusion	Merkle-based proofs, collateral staking, on-chain slashing
Token Volatility	$CT3GB price swings may affect storage affordability	Dual-token model: $CT3GB pegged via oracle, $CT3 used for treasury + CT3GB conversion via DAO
Governance Centralization	Early governance dominated by few stakeholders	Voting weight decay, proposal submission thresholds, Sybil resistance

To maintain predictable storage costs, $CT3GB is pegged to a fiat value via an on-chain oracle. The DAO may convert $CT3 into $CT3GB as needed to maintain peg stability and liquidity.

8.4 Future Outlook

The future of CT3 storage lies not only in data preservation, but in powering a truly user-owned internet. As IPFS, zero-knowledge proofs, and on-chain coordination mature, the project is poised to integrate deeply with:

- **Cross-chain ecosystems** – via NFT bridges and token interoperability
- **DePIN (Decentralized Physical Infrastructure Networks)** – enabling localized storage grids
- **Web3 identity frameworks** – allowing privacy-respecting authorization and access logs
- **Decentralized compute** – pairing storage with trustless execution environments

As decentralization becomes not just an ideal but a regulatory necessity, platforms that successfully abstract away complexity while preserving control will define the future.

CT3 storage is not just a storage solution. It is an infrastructure layer for sovereign data, a testbed for incentive-aligned crypto economies, and a blueprint for user-first, protocol-driven innovation.

8.5 Delegated Architecture & Open Contributor Model

CT3 storage is designed with a deliberately modular architecture. This choice enables rapid development by encouraging external contributor participation through open grant programs. While the core protocol logic and governance mechanics are developed by the founding team, several functional modules — such as NFT minting pipelines, wallet connectors, and storage node UX — are delegated to the broader developer ecosystem.

This model ensures flexibility, avoids central bottlenecks, and increases resilience. By distributing implementation responsibilities through transparent, DAO-governed incentives, CT3 storage seeks to empower a global network of builders, rather than relying on a fixed team with limited bandwidth.

As the project transitions into its decentralized phases, these modular components will progressively evolve into auditable, DAO-governed modules maintained by a growing contributor base.

9. Future Development Directions

9.1 Towards Full Decentralization

While the current implementation of CT3 storage operates within a semi-centralized framework to accelerate development and maintain operational integrity, the long-term trajectory is firmly oriented toward full protocol decentralization. This process will be staged and modular, reducing central points of failure while maximizing community participation.

Planned decentralization layers include:

- **Storage Proofs**: Transitioning from platform-validated proofs to verifiable on-chain PoR and PoA submissions.
- **Order Matching**: Replacing internal logic with open smart contract-based order books for storage and retrieval agreements.
- **Governance Execution**: Migrating treasury controls and upgrade proposals to fully autonomous, contract-enforced outcomes.
- **Token Emission Control**: Implementing decentralized oracles and governance-defined token parameters for emission pacing and burn ratios.

This migration will be guided by stress-tested performance benchmarks and ongoing audits to ensure each component remains secure, performant, and predictable under decentralized conditions.

Token Inflation / Deflation Governance

DAO will control the following parameters to maintain sustainable token economics:

- **Emission Caps:** Annual emission target for $CT3GB will be governed to maintain ≤2% net inflation per year under normal network load.
- **Dynamic Burn Ratios:** Burn ratios for retrieval fees and expired storage can be adjusted via DAO votes to counterbalance inflationary effects.
- **Reward Multipliers:** Miner reward multipliers and staking incentives can be tuned based on market conditions and token velocity.
- **Oracle Pegging:** DAO will govern reference prices used for dynamic rate adjustments to ensure fair fiat-denominated storage costs.

Initial parameters will be tuned conservatively during DAO activation and gradually optimized based on live network performance and community input.

Liquidity Provision (LP Pools) and Token Locking:

To further stabilize the token economy and support healthy market liquidity, CT3 storage DAO will deploy strategic LP pools and token locking mechanisms:

Pool	Target Exchange(s)	Initial LP %	Lock Period
$CT3GB/USDC	Uniswap v3 (Arbitrum Nova)	15% Treasury Allocation	6 months
$CT3/USDC	Uniswap v3 (Arbitrum One)	10% Treasury Allocation	12 months
$CT3GB/$CT3	Balancer Pool	5% Treasury + Community	12 months

DAO-governed LP programs will aim to provide balanced liquidity depth while minimizing impermanent loss. All LP incentives and lock periods will be transparently published via the governance portal.

9.2 Integration with DePIN Infrastructure

Decentralized Physical Infrastructure Networks (DePIN) represent a critical opportunity for CT3 storage to extend beyond the virtual layer. As decentralized compute and bandwidth provisioning grow, localized micro-mining clusters and community-hosted storage zones will enable edge-based file replication and low-latency retrieval.

- **Micro-storage zones** could be operated in co-working hubs, academic networks, or rural telecom installations.
- Miners may opt-in to regional staking pools backed by local resource agreements.
- Regional slashing parameters and uptime validators will improve fault isolation.

Such an approach reinforces physical decentralization and improves data sovereignty across borders.

9.3 Cross-chain and L2 Expansion

To reach broader adoption and increase token utility, CT3 storage will explore deployment on multiple L1 and L2 ecosystems:

- **EVM-compatible L2s** (e.g., Arbitrum, Base, Scroll) for cheap microtransactions and NFT minting.
- **Polygon and Move-based chains** for high-speed, cost-sensitive retrieval markets.
- **IBC/Interchain modules** to enable cross-chain NFT access, file migration, and storage token payments.

Bridging $CT3GB and $CT3 across ecosystems will be essential for maximizing composability, especially with dApps that embed media files, documents, or private content.

9.4 Enabling Privacy-Preserving File Sharing

The next frontier of Web3 storage lies in empowering not just ownership, but **selective disclosure**. Planned privacy upgrades include:

- **ZK-NFT Binding**: Creating NFTs that can be used as zero-knowledge access keys without revealing the identity of the holder.
- **View-only Access Tokens**: Timed or session-based download permissions issued as sub-NFTs.
- **On-chain audit trails**: Enabling transparency over access history without exposing file contents.

This would enable private file sharing in legal, medical, financial, and enterprise contexts, where cryptographic compliance is paramount.

9.5 Developer Ecosystem and Extensibility

CT3 storage will include native support for third-party module development:

- **Storage marketplace APIs** for dApps to create custom upload/download flows
- **Integration SDKs** for popular frameworks (React, Next.js, Vue, Unity)
- **Smart contract templates** for automating NFT issuance, renewals, and access control

A grant system will incentivize contributions from open-source developers, researchers, and integration partners. These efforts are core to making CT3 storage a foundational protocol — not just a platform.

Developer Documentation Roadmap:

Comprehensive API documentation is planned using OpenAPI standards. SDKs for key integrations will be provided for:

- **JavaScript (React / Next.js / Node.js)**
- **TypeScript**
- **Go (for backend integrations)**
- **Python (for automated file workflows)**

9.6 Comparative Storage Pricing: Web2 vs CT3 storage

Storage Provider	Cost (1GB/month)	Notes
AWS S3	~$0.023	No encryption, charges for egress
Google Cloud	~$0.020	No NFT access, opaque pricing
CT3 storage	~$**0.0039**	Encrypted, NFT-controlled access

Based on current CT3GB price assumption of 1 CT3GB = $0.05 → CT3 storage: $0.0225 / GB/mo.

9.7 Stress Test Simulation Results

To evaluate the scalability, economic stability, and incentive compatibility of the CT3 storage protocol, we conducted preliminary simulation modeling on key system dynamics.

The following are example models to guide future detailed benchmarking and serve as an illustration of expected system behavior under varying conditions.

1 Miner Rewards vs Redundancy Level and Time

Objective: Understand how redundancy level k and storage duration D affect miner rewards and economic incentives.

Simulation Parameters:

- File size: 100 MB
- Base rate $R_{base} = 0.000015$ CT3GB/KB/day
- Redundancy coefficient $\rho = 0.15$
- Redundancy levels: k = 1, 2, 3
- Duration: D = 30, 60, 90 days

Results:

- Total miner reward increases linearly with D and grows with k due to redundancy premium.
- Optimal balance: k=2 provides good fault tolerance vs. cost.
- Higher k values incentivize miner participation but increase user cost.

Example Curve:

- k=1 → 45 CT3GB (90 days)
- k=2 → ~62 CT3GB (90 days)
- k=3 → ~79 CT3GB (90 days)

2 Retrieval Cost vs Age of Data

Objective: Model how file age impacts retrieval cost, and incentivize timely downloads.

Simulation Parameters:

- File size: 50 MB
- $R_{ret} = 0.00002$ CT3GB/KB
- Growth coefficient $r = 0.05$
- File age (d): 0 → 60 days

Results:

- Retrieval cost grows logarithmically with age.
- Cost increase slows after ~30 days, keeping retrieval accessible for old data.

- This model helps prevent network spam of stale files.

Example Curve:

- Day 0 → 10 CT3GB
- Day 10 → ~13 CT3GB
- Day 30 → ~16 CT3GB
- Day 60 → ~18 CT3GB

3 System Cost vs CT3GB Price Fluctuation

Objective: Evaluate impact of token price volatility on system affordability and miner incentives.

Simulation Scenario:

- Baseline CT3GB price = $0.05
- Simulate token value ×10 ($0.50) and ÷10 ($0.005)

Observations:

- Built-in price oracle and dynamic rate adjustment planned to stabilize effective fiat costs.
- Without adjustment:
    - ×10 token price → storage becomes prohibitively expensive → user drop-off.
    - ÷10 token price → miner rewards decrease → miner churn risk.
- With oracle-adjusted base rate (planned Q2 2026):
    - Stable cost of ~$0.02–$0.03/GB/month maintained.

Conclusion:
Dynamic pricing controls are critical to maintaining system usability and miner incentive alignment under market volatility.

9.8 Summary of Recent Improvements

This updated version of the CT3 storage Whitepaper (v1.1) includes the following enhancements in response to community and stakeholder feedback:

- Clarified **system roadmap**, including explicit timeline for full **on-chain migration of Proof-of-Replication and Proof-of-Access** mechanisms.
- Introduced detailed **parameter specifications** for key economic and reputation formulas.
- Added **Threat Model** section outlining mitigation strategies for key protocol risks.
- Added **Protocol Flow Diagrams** for improved architecture transparency.
- Provided initial **Stress Test Simulation Results** to demonstrate economic robustness under varying conditions.
- Revised **Conclusion** to better reflect CT3 storage's strategic vision and governance trajectory.

These improvements aim to enhance transparency, engineering maturity, and long-term trust in the protocol.

9.9 NFT Architecture and Cost Optimization

As CT3 storage leverages NFTs for decentralized access control and file ownership tracking, it is critical that these NFTs are minted, stored, and rendered in a way that is both cost-efficient and wallet-compatible.

Challenges Identified

- Display inconsistency in Phantom Wallet: NFTs minted via standard EVM instructions sometimes fail to consistently show in the collectibles tab, especially if metadata is malformed, or if the `collection` field is missing or unverified.
- High minting cost: On Polygon mainnet, traditional minting via EVM's Candy Machine or token-metadata instructions often exceeds \$2–3 USD per mint, due to transaction size and lack of optimization.

Recommended Solution: Compressed NFTs via Bubblegum

To address these issues, the team recommends migrating to Compressed NFTs (cNFTs) using the Bubblegum protocol on Polygon. Key benefits include:

- \~98% lower minting cost (approx. \$0.01–\$0.02 per NFT)
- Merkle-tree based metadata storage → significantly reduces on-chain load
- Fully Phantom-compatible when metadata and verified collection are correctly configured
- Supports collection-based queries, display, and resale integration

Required Implementation Steps:

1. Migrate existing NFT mint logic to use `Bubblegum` and `MerkleTree` format

2. Register a Verified Collection with the token-metadata program (via `collection.verified = true`)

3. Ensure all metadata follows the EVM JSON schema (with valid URI, creators array, collection key, etc.)

4. Test cross-wallet display compatibility across Phantom, Solflare, Backpack

This transition will both significantly reduce on-chain costs and ensure full display reliability in Phantom and other Polygon wallets.

10. Project Team, Repository & Legal

10.1 Core Contributors

The development of **CT3 storage** is driven by a globally distributed and pseudonymous team with extensive experience in blockchain architecture, cryptographic storage systems, and decentralized governance.

- **Leandro Gomes**— Head of Partnerships & Governance
  Develops and implements a strategy for partnerships with blockchain infrastructures, universities and data storage facilities. Leads the formation of decentralized autonomous governance structures (DAO), ensuring the sustainability and transparency of the storage system. **LinkedIn:** https://www.linkedin.com/in/leandro-gomes-903211202/

- **Rodrigo Pereira** — Chief Marketing Officer (CMO)
  Develops and executes the global marketing strategy to position the company as a leader in decentralized storage solutions. Oversees brand development, product positioning, and communication campaigns across multiple markets. Builds strategic alliances with blockchain ecosystems, technology partners, and industry media to drive adoption and market growth. **LinkedIn:** https://www.linkedin.com/in/rodrigo-pereira-5b786099/

- **Monique Amaral** — Chief Financial Officer (CFO)
  Oversees the company's global financial strategy, ensuring sustainable growth and compliance across all jurisdictions. Manages budgeting, forecasting, and capital allocation to support the expansion of decentralized storage infrastructure. Implements transparent financial practices and risk management frameworks aligned with blockchain-based operations. **LinkedIn:** https://www.linkedin.com/in/monique-amaral-215173239/

A portion of the team remains pseudonymous to preserve decentralization, avoid jurisdictional targeting, and allow future community-led governance transitions.

10.2 Codebase and Repository (Preview Access)

- GitHub: https://github.com/nebula-vault/core-protocol
- Testnet Explorer: https://testnet.CT3 storage.io
- Code License: MIT / Business Source License 1.1 (BSL, 24-month lock)
- Smart Contracts: Polygon (Anchor framework), Bubblegum (Merkle tree compression), EVM Token Metadata, Custom PoR validator programs
- Frontend: React / TypeScript / Polygon Wallet Adapter / Anchor IDL / web3.js
- IPFS Gateway Layer: implemented via js-ipfs with fallback to public pinning

 MVP deployment available to select testers upon request.
Internal testnet is live with NFT minting, basic PoR simulation, and role-based miner dashboard.
Contracts to be audited by a third-party firm (pending agreement).

10.3 Legal Disclaimers

This document is provided for **informational purposes only** and does **not constitute a public offering or investment solicitation**. CT3 storage is not a registered entity, and participation in the testnet or token economy implies full acceptance of associated risks.

- Legal incorporation (in progress): Cuillin Technology Limited
- Company address: 167-169 Great Portland Street, 5th Floor, London, W1W 5PF United Kingdom
- Company number: 14453567

- KYC/AML policy: not applicable during MVP testing phase
- DAO governance and treasury controls will become active post-launch with voting via $CT3 token holders.

All users are advised to conduct their own legal due diligence and consult a licensed advisor before participating in token purchases, mining, or DAO voting proposals.

11. Simulated Protocol Use Case

This use case illustrates the full storage cycle within the CT3 storage ecosystem, demonstrating NFT-minting, miner proof submission, download logic, and token dynamics. The formulas and fee models below reflect cost-efficient structures based on simulated demand and gas benchmarking.

Scenario: Alice Uploads, Bob Stores, and Retrieves

1. File Upload (Client: Alice)

- Alice connects her Phantom wallet.
- Uploads a 20 GB file for a storage duration of 30 days.
- File is encrypted on-device.
- File is uploaded to IPFS; returned CID = `QmXYZ...`
- NFT is minted containing:
    - File CID (immutable content reference)
    - Expiration timestamp (now + 30d)
    - Flags: encryption status, password required for decryption

2. Cost of Upload (Base Storage Fee Calculation)

The base fee accounts for storage size, duration, redundancy factor *k*, and unit rate:

$$F_{store} = S_{GB} \cdot D_{days} \cdot R_{base} \cdot (1 + \rho \cdot (k - 1))$$

Where:

- S = 20 GB
- D = 30 days
- R_{base} = 0,0005114058954666667 CT3GB/GB/day
- ρ = 0.15, redundancy impact coefficient

For a 20 GB file stored for 30 days at k=2, total cost is:

$$F_{store} = 20 \cdot 30 \cdot 0{,}0005114058954666667 \cdot (1 + 0.15) = 0.3528700678743\ CT3GB$$

- **20% burned**: 0.07057401357486 CT3GB
- **80% streamed** to contract across 30 days: 0.28229605429944 CT3GB

3. Storage Matching (Miner: Bob)

- Miner Bob accepts storage offer.
- Locks collateral *C = 0.01374609375 CT3GB*
- Begins submitting Merkle-based PoR every $\tau = 12$ hours
- Reward per proof R_p is:

$$R_p = \frac{0.659713605152}{30 * 2} = 0.0109952267525333\ CT3GB$$

4. Retrieval Fee (Dynamic Cost Protection) To prevent exponential growth, the fee uses bounded logarithmic logic:

$$F_{retrieve}(d) = S_{GB} \cdot R_{ret} \cdot \left(1 + r \cdot log_2(d+1)\right)$$

Where:

- $R_{ret} = 0.0000707625$ CT3GB/GB
- $r = 0.05$, daily cost increase
- $d = 10$: retrieval on day 10

$$F = 20 \cdot 0.00002202412989786669 \cdot \left(1 + 0.05 \cdot log_2(11)\right) \approx 0{,}0004404825979573338 \cdot (1 + 0.05 \cdot 3.46) \approx 0{,}0004404825979573338 \cdot 1{,}1729715809318648628 \approx 0{,}000516733 CT3GB$$

Split:

- **Burned** (25%): 0,0001033466 CT3GB
- **Miner payout**: 0,0004133864 CT3GB

5. Proof Verification and Slashing If Bob misses 2 consecutive proofs:

$Penalty = \delta \cdot C, \delta = 0.25 \Rightarrow 0.25 \cdot 0{,}11782736386 \ = 0.0034365234375\ CT3GB$
CT3GB is used as collateral token due to its predictable storage-pegged value.
Collateral in CT3GB ensures penalty stability and protects against speculative token volatility.

- Reputation reduced based on proof failures and latency:

$$R_i(t+1) = R_i(t) - 0.0034365234375 \cdot f_t - \lambda \cdot l_i$$

Where $\lambda = 0.01$, l_i is average latency in seconds.

6. File Expiry (NFT Burn and Cleanup)

- After day 30, file auto-expires unless renewed.
- NFT is burned on-chain.
- Bob's node deletes the file.
- Final rewards are released.

Summary Table

Stage	Action	Cost (CT3GB)	Token Logic
Upload	Store 20GB for 30d	0.3528700678743	20% burned, 80% to miners
Retrieval	File downloaded on day 10	0,000516733	25% burned, 75% to miners
Proofs	60 total PoR ($\tau = 12h$)	-	0.00057275390625 CT3GB per valid proof

Missed PoR	Bob fails 2	-	Slash: 0.0034365234375 CT3GB from collateral
Expiry	File deleted after 30d	-	NFT burn, storage ends

The improved simulation reflects economic balancing: enough to reward miners and protocol sustainability, while keeping user costs grounded — a 20MB file for 30 days costs \~31.75 CT3GB total.

12. Glossary & References

12.1 Glossary of Terms

Term	Definition
NFT (Non-Fungible Token)	A unique digital asset used here to control access to uploaded files and track metadata.
IPFS (InterPlanetary File System)	A peer-to-peer file storage protocol where encrypted files are stored and identified via CIDs.
CID (Content Identifier)	A hash-based address of a file stored on IPFS.
PoR (Proof-of-Replication)	A cryptographic method to prove that a file has been physically replicated and stored by a unique miner, ensuring authenticity and availability over time.
PoA (Proof-of-Access)	A proof system to verify that a file is not only stored but retrievable from a storage node.
$CT3GB	The utility token used for paying storage, downloads, and rewarding miners.

$CT3	The investment/governance token with staking, buyback, and voting privileges.
Merkle Root	A hash representing the root of a binary hash tree, used to verify the integrity of data.
VRF (Verifiable Random Function)	A cryptographically secure way to generate randomness, often used for unbiased proof selection.
DAO (Decentralized Autonomous Organization)	A governance structure in which decisions are made by token holders through smart contracts.
Slashing	A penalty mechanism that confiscates part of a miner′s stake in case of proven misbehavior.
DePIN	Decentralized Physical Infrastructure Network, where physical services are provided via Web3
ZK (Zero-Knowledge)	Cryptographic methods that prove knowledge of data without revealing the data itself.
Snapshot	An off-chain voting platform commonly used by DAOs to record governance votes.

12.2 Notations and Mathematical Symbols

Symbol	Description
$S(t)$	Token supply at time t
$B(t)$	Tokens burned at time t
$E(t)$	Tokens emitted at time t
C	Collateral posted by a miner
R_t	Reward distributed at time t
δ	Slashing multiplier
θ_s	Slashing threshold
W_i	Voting weight for user i
S_i	Staked tokens by user i
A_i	Age multiplier for user i
T_i	Trust coefficient for user i
M	Merkle root of a file′s data
t	Timestamp or time parameter
$H(x)$	Hash of input x
D_b	Data chunk at byte offset b

12.3 References and Further Reading

1. IPFS — Documentation
2. Polygon Foundation — EVM NFT Standard
3. Vitalik Buterin — DAOs, DACs, DAs and More

4. Arweave — Permaweb and Storage Pricing
5. ZKProof.org — Zero-Knowledge Proofs Primer
6. Messari Crypto Theses — Web3 Infrastructure Reports
7. OpenZeppelin — Smart Contract Security Standards
8. Snapshot.org — Governance tooling for DAOs
9. Chainlink — VRF: Verifiable Randomness