Zenon as a Verification and Accountability Layer for Decentralized Storage Networks

Bounded Verification of External Storage Commitments

Status: Feasibility Research / Architectural Analysis

Date: December 2025

Abstract

This paper examines a verifiability asymmetry in decentralized storage systems: while protocols like Filecoin can cryptographically prove storage correctness to validators and peers, resource-bounded end users (particularly browser-based verifiers) cannot efficiently validate these proofs without trusted intermediaries. We analyze how Zenon Network's architecture can serve as a verification and accountability layer that records externally-verified storage outcomes under bounded inclusion semantics. Zenon does not replace storage systems, execute storage proofs, or adjudicate correctness disputes. Instead, it provides a bounded-state commitment ledger where storage providers publish accountability commitments and where outcomes from external verification systems (such as Filecoin's consensus) are recorded for browser-verifiable audit. We formalize the trust boundaries, adversarial assumptions, commitment semantics, and information-theoretic limits of this approach.

Scope and Non-Goals

Non-Goal: Zenon does not store data, execute storage proofs (PoRep/PoSt), verify cryptographic correctness of external systems, or guarantee data availability or retrieval.

Goal: Formalize how Zenon can provide bounded-state verification of storage accountability commitments and externally-verified outcomes for resource-constrained clients, particularly browser-based verifiers operating without trusted infrastructure.

1. The Verifiability Asymmetry Problem

1.1 Storage vs. Accountability Verification

Decentralized storage networks such as IPFS and Filecoin address data distribution and economic storage incentives. Filecoin, in particular, employs cryptographic proofs (Proof-of-Replication, Proof-of-Spacetime) that validators can verify to enforce storage commitments. However, these proofs impose computational and bandwidth costs incompatible with resource-bounded clients.

End users—especially browser-based clients—face a verifiability asymmetry:

- Storage systems prove correctness to each other (validators, miners, nodes)
- Resource-bounded users cannot verify these proofs directly
- Users must trust gateways, RPC providers, or indexing services
- This reintroduces centralized trust into ostensibly decentralized systems

1.2 Core Invariant

> **Motivating Invariant:** A system provides meaningful decentralization only if resource-bounded clients can verify relevant system properties without trusting infrastructure operators.

Zenon addresses this asymmetry not by replacing storage or re-executing storage proofs, but by separating accountability verification from storage verification. Storage systems handle proof generation and validation among peers; Zenon records publicly committed accountability outcomes in a form verifiable by finite-state clients.

2. Architectural Position and Layer Separation

Zenon operates as a verification layer above storage execution:

```
Application / End User ↓ Zenon (accountability commitments + outcome recording) ↓ Filecoin (storage incentives + PoRep/PoSt execution) ↓ IPFS (content addressing + data retrieval)
```

> **Layer Separation Principle:** Zenon does not participate in storage, proof generation, or proof validation. It provides a commitment and outcome ledger that resource-bounded clients can verify independently.

3. Trust Boundaries and Adversarial Model

3.1 What Zenon Does Not Claim

Zenon makes no claims about:

- Correctness of storage proofs (PoRep, PoSt)
- Physical storage of data
- Data availability or retrievability
- Validity of external system consensus
- Adjudication of disputes between storage providers and clients

3.2 Formal Distinctions

Accountability: Public record of who committed to what terms

Truth: Actual state of physical storage (unknowable to any blockchain)

Adjudication: Determining correctness of disputed claims (not performed by Zenon)

Zenon records accountability commitments and enforces consequences of externally-verified outcomes. It does not judge whether Filecoin's consensus is correct, only that Filecoin's consensus reported outcome X, which Zenon then treats as a fact for accountability enforcement.

3.3 Adversarial Model

Zenon's accountability layer defends against:

- Providers claiming commitment without public record
- Providers retrospectively denying commitments
- Infrastructure operators falsifying accountability records
- Clients unable to verify commitment history without trusted parties

Zenon does **not** defend against:

- Compromise of external storage systems (IPFS, Filecoin)
- Censorship at the retrieval layer
- Majority attacks on Filecoin consensus
- Providers who honor commitments to Zenon but fail storage obligations
- Long-range historical revision beyond the minimal state frontier

> **Assumption:** Zenon assumes that external verification systems (e.g., Filecoin consensus) operate as designed and that their reported outcomes reflect honest majority validation. If Filecoin is compromised, Zenon faithfully records the compromised outcomes—it does not detect or prevent such compromise.

4. Data Model and Commitment Semantics

4.1 Content Addressing

Files stored on IPFS are identified by Content Identifiers (CIDs):

```
CID = H(F)
```

Where H is a cryptographic hash function and F is the file content. Filecoin storage deals reference CIDs. Zenon never processes file data—only CID references and associated commitment metadata.

4.2 Storage Commitment Structure

A storage provider publishes an accountability commitment to Zenon's ledger:

```
C = (CID, p, T, ρ, σ, τ)
```

Where:

- **CID**: IPFS content identifier being committed to
- **p**: Filecoin storage provider address (public key)
- **T**: Storage duration in epochs or block height range
- ρ: Economic parameters (collateral, reward schedule, slashing conditions)
- σ: Provider's cryptographic signature over (CID, p, T, ρ)
- τ: Zenon block height at commitment publication

This commitment is written to the provider's account-chain within Zenon's dual-ledger structure. The Momentum consensus layer globally orders and finalizes the commitment.

4.3 Commitment Semantics and Ordering

Definition (Commitment Semantics): Provider commitments on Zenon are:
• **Append-only**: Once a commitment is finalized in Momentum, it cannot be removed or altered
• **Non-superseding**: New commitments do not invalidate prior commitments to the same CID
• **Monotonic in accountability**: Additional commitments increase the accountability surface; they do not reset it
• **Globally ordered**: Momentum provides a total order of all commitments across all provider account-chains

Contradictory Commitments: A provider may commit to mutually exclusive storage terms (e.g., overlapping but conflicting duration or collateral for the same CID). Zenon does not prevent this—it records both commitments. External adjudication or social consensus determines which commitment is honored. Zenon enforces consequences only for commitments that are explicitly marked for settlement via outcome recording.

5. Bounded Inclusion of External Verification Outcomes

5.1 What Zenon Verifies

Zenon verifies structural and cryptographic properties of commitments:
• Signature validity (σ authenticates provider p)
• Well-formedness of commitment data
• Ordering and finality within Momentum consensus

Zenon does **not** verify storage proofs (PoRep, PoSt) or data availability.

5.2 Outcome Recording from External Systems

Filecoin produces cryptographic storage proofs that Filecoin validators verify. These proofs are computationally expensive and unsuitable for browser verification.

Instead of re-verifying Filecoin proofs, Zenon records outcome attestations:

```
E_t = (p, CID, result, t, attestation)
```

Where **result** ∈ {pass, fail, missing} indicates the outcome of Filecoin's verification at epoch **t**, and **attestation** is a verifiable reference to Filecoin consensus state (e.g., block hash, state root, or light client proof).

5.3 Semantic vs. Transaction-Identity Inclusion

Definition (Bounded Inclusion): Zenon employs semantic, outcome-based inclusion rather than transaction-identity inclusion. This means:

• Zenon accepts attestations that "provider p satisfied condition X according to Filecoin at epoch t"
• Zenon does not replay Filecoin transactions, re-execute zkSNARKs, or validate Merkle proofs of PoSt
• Zenon verifies only that the attestation is structurally valid, signed by a recognized oracle or relayer, and consistent with prior recorded state

Precision Trade-off: This approach sacrifices cryptographic re-verification of storage proofs in exchange for O(1) verification complexity with respect to storage size. Zenon trusts that Filecoin consensus correctly validated the proof—it does not independently verify proof correctness.

This design choice is necessary for bounded verification: a browser cannot store or process gigabytes of Filecoin state or execute expensive cryptographic proofs. By accepting outcome attestations, Zenon enables lightweight accountability verification while remaining honest about what is and is not being verified.

6. Minimal State Frontier and Information-Theoretic Limits

6.1 Finite Verifier Constraints

Resource-bounded clients (browsers, mobile devices) cannot store or process complete chain history. They operate under finite memory and bandwidth constraints.

> **Definition (Minimal State Frontier):** A client maintains a sliding window of the most recent k Momentum headers, provider account-chain proofs for relevant commitments, and no external chain data. The parameter k is chosen to balance security (longer window) against resource costs (shorter window).

6.2 Guarantees Within the Frontier

Within the k-block window, a client can verify:

- Existence and ordering of all storage commitments published by tracked providers
- Consistency of outcome attestations with recorded commitments
- Economic enforcement events (payments, slashing) tied to specific outcomes
- Absence of contradictory finalized state within the window

6.3 Limitations Beyond the Frontier

Outside the k-block window, a finite-state client **cannot** verify:

- Historical commitment existence prior to the window
- Long-range consistency of state transitions
- Whether current state is the result of a long-range attack that rewrites ancient history

> **Information-Theoretic Constraint:** This limitation is inherent to any finite-memory verifier. A client with bounded storage cannot distinguish a valid chain from a forged chain that diverges before the window, unless it either (a) stores complete history, (b) trusts a checkpoint authority, or (c) relies on social consensus about canonical chain state. Zenon's minimal state frontier is an explicit acknowledgment of this fundamental limit.

Practical mitigation: Clients can periodically checkpoint state roots signed by multiple independent parties or rely on weak subjectivity assumptions common in proof-of-stake systems.

7. Accountability Enforcement and Settlement

7.1 Enforcement Semantics

> **Clarification:** Zenon does not arbitrate disputes or determine the correctness of storage claims. Zenon enforces consequences of externally verified outcomes. If Filecoin consensus reports that provider p failed a PoSt check, Zenon can execute a slashing condition specified in the original commitment—but Zenon does not judge whether Filecoin's consensus was correct.

7.2 Concrete Flow Example: CDN-Style Storage

Consider a decentralized content delivery use case:

1. Application pins content to IPFS (CID generated)
2. Storage deals negotiated with Filecoin providers
3. Providers publish accountability commitments C to Zenon, including collateral ρ and slashing terms
4. Filecoin validators verify PoSt proofs according to Filecoin protocol
5. Outcome attestations E_t are submitted to Zenon by designated oracles or bridge relayers
6. Zenon executes settlement: providers passing verification receive rewards; providers failing trigger slashing per ρ
7. Clients query Zenon to verify accountability history without querying Filecoin directly

> **Key Property:** End users can verify who was committed to storage and what outcomes were recorded, without trusting HTTP gateways, Filecoin RPC nodes, or centralized indexers. They cannot verify that the storage proofs were correct (they must trust Filecoin consensus for that), but they can verify accountability and enforcement consistency.

8. Case Study: Censorship-Resistant Journalism Archive

A journalist publishes sensitive documents:

- Documents uploaded to IPFS, yielding CID
- Multiple geographically distributed Filecoin providers commit to long-term storage
- Commitments with economic stakes published to Zenon
- Periodic renewal commitments recorded as storage deals expire and renew

Years later, an auditor or researcher can verify:

- Which providers committed to storing the documents
- Duration and economic terms of commitments
- Whether commitments were honored (per recorded Filecoin outcomes)
- Whether any provider was slashed for failure

> **Important Limitation:** If the content is censored at the retrieval layer (IPFS gateways blocked), Zenon cannot restore access. Zenon provides an accountability audit trail, not availability guarantees. The audit trail itself is valuable: it creates verifiable evidence of storage commitments, enabling transparency and potentially legal or social accountability even when technical censorship succeeds.

9. Explicit Limitations and Non-Solutions

To avoid over-claiming, we explicitly enumerate what Zenon does **not** solve:

Data Availability: Zenon cannot guarantee files are retrievable; it only records commitments to store them

Censorship Resistance: Zenon cannot prevent network-layer or gateway-layer censorship of IPFS content

Proof Correctness: Zenon does not verify cryptographic correctness of PoRep or PoSt; it trusts Filecoin consensus

Dispute Resolution: Zenon does not adjudicate conflicting claims; it enforces pre-agreed consequences of external outcomes

Long-Range Security: Finite-state verifiers cannot detect long-range attacks beyond their state window without additional trust assumptions

Global Indexing: Zenon is not a search engine or content discovery layer for IPFS

> **Core Value Proposition:** Zenon provides verifiable accountability for storage commitments and externally-verified outcomes, enabling resource-bounded clients to audit these properties without trusted intermediaries. This is orthogonal to—not a replacement for—storage,

retrieval, and proof generation.

10. Why Separation of Concerns Matters

Most decentralized systems conflate execution, storage, verification, and user-facing audit. Zenon's architectural contribution is to separate these concerns:

Concern	Responsible System	Verification Scope
Storage Execution	IPFS	Content addressing, routing
Economic Incentives	Filecoin	PoRep/PoSt validity
Accountability Recording	Zenon	Commitment existence, outcome consistency
End-User Audit	Browser Client + Zenon	Bounded-state verification

Filecoin answers: *"Was the storage proof cryptographically valid?"*

Zenon answers: *"Who was publicly accountable, what did they commit to, and were agreed consequences enforced based on externally verified outcomes?"*

This separation enables browser-based audit of accountability without requiring browser clients to verify expensive cryptographic proofs or store gigabytes of chain state.

11. Comparative Analysis of Verification Properties

The following comparison is scoped strictly to **accountability verification**, not storage capacity, retrieval performance, or proof generation:

Property	IPFS	Filecoin	Zenon
Stores data	Yes	Yes (incentivized)	No
Economic enforcement	No	Yes (via PoSt)	Yes (via commitments)
Browser-verifiable accountability	No	No (proof verification too expensive)	Yes (bounded-state)
Bounded verification complexity	N/A (no proofs)	No (O(proof size))	Yes (O(1) wrt storage)
Accountability ledger	No	Implicit in consensus	Explicit commitment ledger
Adjudicates correctness	N/A	Yes (via consensus)	No (records external outcomes)

Note: "Browser-verifiable accountability" specifically means a client can verify commitment existence, ordering, and enforcement without trusting infrastructure. It does not imply verification of proof correctness or data availability.

12. Formal Interpretation Rule

Interpretation Rule: When Zenon records a storage outcome E_t = (p, CID, result, t, attestation), the following interpretation holds:

What is claimed: "Provider p was publicly committed to storing CID, and an attestation from a recognized external system (Filecoin) reports that p achieved result (pass/fail/missing) at epoch t."

What is NOT claimed:
• The data is currently retrievable
• The Filecoin proof was cryptographically re-verified by Zenon
• No censorship occurred at any layer
• Provider p actually stored the data physically (only that Filecoin consensus reported the outcome)

Audit Guarantee: A resource-bounded client can verify that the commitment and outcome records are internally consistent, properly ordered, and finalized by Zenon consensus. The client cannot independently verify that Filecoin's consensus was correct—that trust assumption is explicit and unavoidable under bounded verification.

13. Conclusion

Decentralized storage networks successfully implement cryptographic proofs of storage, but these proofs impose verification costs incompatible with resource-bounded end users. This creates a verifiability asymmetry: storage correctness can be proven to validators but not audited by everyday clients without trusted intermediaries.

Zenon addresses this asymmetry not by replacing storage systems or re-executing proofs, but by providing a bounded-state ledger for accountability commitments and externally-verified outcomes. This separation of concerns enables:

- Browser-based verification of who committed to storage and under what terms
- Audit of accountability enforcement without trusting infrastructure operators
- O(1) verification complexity with respect to storage size
- Explicit acknowledgment of trust boundaries and information-theoretic limits

IPFS provides content-addressed storage and retrieval.

Filecoin provides economic incentives and cryptographic proofs of storage.

Zenon provides verifiable accountability for commitments and externally-verified outcomes—enabling resource-bounded audit.

> **Architectural Role:** Zenon does not compete with storage networks. It makes their accountability properties auditable by clients who cannot verify expensive cryptographic proofs. This is a layer separation strategy, not a replacement architecture.

This paper presents a feasibility analysis and architectural mapping. Implementation would require further specification of oracle mechanisms, attestation formats, economic parameter choices, and security analysis under specific adversarial models.