Zenon Greenpaper Series — Cover Letter

Document: A Verification-First Architecture for Dual-Ledger Systems

Status: Community-authored greenpaper (non-normative, non-official)

Authorship: Zenon Developer Commons (community)

Date: January 4, 2026

Purpose

This greenpaper proposes a verification-first architecture for dual-ledger systems. The central claim is simple: verification is foundational, while execution is constrained to remain verifiable under explicit, declared resource bounds.

The design targets real deployment constraints—intermittent connectivity, browser-native computation, and bounded storage/bandwidth—so that lightweight clients can independently verify correctness without replaying full execution or reconstructing global state.

What this paper contains

The architecture is organized around three tightly integrated pillars:

1. Bounded Verification
   Verification under explicit resource constraints, anchored to genesis trust roots with adaptive retention.
2. Proof-Native Applications (zApps)
   Application correctness established via cryptographic proofs rather than execution replay.
3. Composable External Verification (CEV)
   Trustless validation of external facts (notably Bitcoin) using proof-based predicates and explicit refusal semantics when evidence is insufficient.

What “non-normative” means here

This document is a design and formalization artifact, not an official Zenon Network spec and not a protocol upgrade proposal by itself. Its goal is to make assumptions explicit, define verification/refusal behavior precisely, and provide builders a coherent model to implement against—or to challenge.

How to read it

- Researchers / protocol engineers: prioritize definitions, predicates, refusal semantics, and any security-relevant invariants.
- Builders: focus on the verification interfaces, proof formats, retention behavior, and end-to-end flows (including offline recovery).
- Community reviewers: evaluate whether the stated constraints match plausible deployment realities and whether the proposal's assumptions are acceptable.

What feedback is most valuable

- Formal model vs. deployment reality: data availability, proof distribution/serving, and retention assumptions—i.e., what must exist in practice so verifiers don't frequently return REFUSED.
- Economic/incentive gaps: who pays for proof generation and serving; what incentivizes Momentum block production and external-commitment publication; how historical proofs remain available over time.
- Consensus binding: what Zenon actually uses, concrete finality/reorg guarantees, and how the abstract finality function should be instantiated operationally.
- Resource accounting with real numbers: realistic targets for storage/bandwidth/compute on browser/mobile, plus measured verification times (including WASM/browser constraints).
- Comparative analysis (empirical): how this stacks up against zkSync/StarkNet-style rollups, Celestia-style DA layers, and recursive-proof systems—especially on DA assumptions, trust roots, and proving vs. verifying costs.

Closing

The paper is intentionally explicit about refusal: if required evidence is missing or insufficient within declared bounds, a verifier should refuse rather than guess. If you believe this framing fails in practice (availability, incentives, consensus realities), that is exactly the critique we want—ideally with concrete counterexamples, measurements, or alternative designs.

— Zenon Developer Commons (community)