Bitcoin SPV Feasibility in the Zenon Ledger Model

Research Note — Cryptographic Feasibility and Architectural Implications

Abstract

This document presents a formal feasibility analysis demonstrating that Bitcoin transaction inclusion can be verified within Zenon using standard Simplified Payment Verification (SPV) primitives. We show that Zenon's dual-ledger architecture [Zenon 2018, 2019]—comprising asynchronous account chains and a consensus momentum chain—enables Bitcoin SPV verification with novel efficiency properties unavailable to monolithic blockchain architectures. The work establishes cryptographic feasibility and examines ledger-level implications but does not propose protocol activation or implementation commitments. The core contribution is demonstrating that Zenon's ledger semantics naturally support treating externally verified Bitcoin facts as ledger-native state transitions, achieving O(interested_accounts) rather than O(all_validators) verification costs.

1. Scope and Non-Claims

This document establishes cryptographic and architectural feasibility only. It does not claim current functionality, active development, deployment timelines, incentive mechanisms, networking design, or governance decisions. All cryptographic primitives referenced are standard Bitcoin SPV constructs [Nakamoto 2008]. The novelty lies in Zenon's architectural properties [Zenon 2018, 2019], not in the underlying cryptography.

What this document demonstrates: That Zenon's dual-ledger model [Zenon 2018, 2019] provides efficiency advantages for cross-chain verification unavailable to Ethereum, Cosmos, or Polkadot.

What this document does not address: Data availability, header relay incentives, eclipse resistance, economic alignment, or protocol governance—these are engineering questions that do not affect cryptographic validity.

2. Problem Statement and Verification Objective

2.1 The Statement to Verify

We define the external statement S as:

A Bitcoin transaction tx is included in the canonical Bitcoin blockchain and is buried under at least z blocks of proof-of-work.

Zenon does not execute Bitcoin logic, validate Bitcoin scripts, or replicate Bitcoin state. It only verifies cryptographic evidence sufficient to establish S deterministically.

2.2 Why This Matters

Most blockchains treat cross-chain verification as a global execution problem: every validator must process every verification. This creates three scaling bottlenecks:

1. **Computation:** All validators pay verification cost even if only one application uses it

2. **Storage:** All validators must maintain foreign chain state (e.g., Bitcoin headers)

3. **Consensus:** Cross-chain proofs must pass through global state machine

Zenon's architecture enables treating verification as a local account operation that momentum consensus checkpoints rather than executes.

3. Cryptographic Primitives

Bitcoin SPV relies on three standard primitives. These are well-established and require no novel cryptography.

3.1 Proof-of-Work Header Validity

Each Bitcoin block header H must satisfy:

$$\mathrm{SHA256}^2(H) \le T$$

where T is the difficulty target encoded in nBits. This is deterministically verifiable with two SHA-256 operations and one integer comparison.

3.2 Cumulative Chainwork Comparison

To identify the canonical chain, compute expected work per block:

$$W_i = 2^{256} / (T_i + 1)$$

Total chainwork $W_{chain} = \Sigma W_i$. The chain with maximum cumulative work is canonical. This is an objective function requiring no subjective agreement.

3.3 Merkle Inclusion Proofs

Each block header commits to a Merkle root R. Given transaction hash h and Merkle branch $\pi = (s\blacksquare,\ldots,s\blacksquare)$:

$$x\blacksquare = h$$
$$x_j = \mathrm{SHA256}^2(x_{j-1} \mathbin{||} s_j) \text{ or } \mathrm{SHA256}^2(s_j \mathbin{||} x_{j-1})$$
$$\text{Inclusion verified iff: } x_k = R$$

Proof size: O(log n) where n is transactions per block. For typical blocks (~2000 tx), proof is ~480 bytes.

3.4 Confirmation Security

With attacker hashpower fraction $q < p = 1-q$, the probability of reversing z confirmations is:

$$P_{attack} \approx (q/p)^z$$

Standard 6 confirmations: $P_{attack} < 0.1\%$ for q=0.1. Security increases exponentially with depth.

4. Verification Model

4.1 Deterministic Verification

Zenon validators can deterministically verify SPV proof packages by executing:

1. Header linkage verification (prevHash chains correctly)

2. Proof-of-work threshold check ($\text{SHA256}^2(H) \leq T$)

3. Cumulative chainwork dominance (this chain has most work)

4. Merkle inclusion proof (tx hash → Merkle root)

5. Confirmation depth (z blocks of subsequent work)

This verification is objective—it does not depend on voting, oracle committees, or subjective agreement. Given the same Bitcoin headers, all honest verifiers reach the same conclusion.

4.2 Header Availability and Minimal Threat Model

Practical consideration: While verification is deterministic given headers, header availability is the primary operational challenge. Possible strategies include: (a) bonded relayer networks with slashing for invalid data, (b) Pillar infrastructure services competing on reliability, (c) direct P2P sync with Bitcoin nodes, or (d) multi-source aggregation with quorum requirements.

Minimal threat model: The primary attack vector is *eclipse*—isolating a verifier from the canonical Bitcoin chain. Standard mitigation is source diversity: if a verifier queries N independent header providers and requires k-of-N agreement on chainwork (e.g., 2-of-3), an attacker must compromise k sources. Under typical network diversity assumptions, this provides security comparable to running a Bitcoin SPV light client. Applications requiring stronger guarantees can run full Bitcoin nodes directly. Implementation strategies and concrete adversarial scenarios are explored in Appendix B.

4.3 Proof Structure

A complete SPV proof consists of:

```
type SPVProof struct {
    Header       [80]byte     // Bitcoin block header
    TxHash       [32]byte     // Transaction to verify
    MerkleBranch [][32]byte   // ~480 bytes (log n proof)
    BlockHeight  uint64       // Height in Bitcoin chain
    Depth        uint32       // Confirmations
}
```

Total size: ~640 bytes

5. Ledger-Level Integration: The Architectural Advantage

5.1 The Key Insight

Zenon lacks a general-purpose virtual machine and does not permit user-defined validity predicates. This apparent limitation becomes an architectural advantage for cross-chain verification.

In Ethereum: SPV verification must occur in the EVM. Every validator executes every verification. A single Bitcoin proof costs all validators collectively.

In Zenon: SPV verification can occur in account-chain context. Only interested accounts perform verification. Momentum consensus checkpoints results without re-executing verification.

5.2 Dual-Ledger Verification Model

Phase 1: Account-Chain Verification (Asynchronous)

An account-chain block contains an SPV proof. The account executes local verification:

```
Account receives: SPVProof
Account verifies:
  - PoW validity: SHA256²(Header) ≤ Target
  - Chain work: IsCanonical(Header, LocalBitcoinView)
  - Merkle inclusion: VerifyProof(TxHash, Branch, MerkleRoot)
  - Depth: CurrentHeight - BlockHeight ≥ RequiredDepth
Account state: TxConfirmed = true|false
```

Phase 2: Momentum Checkpointing (Consensus)

Momentum consensus does NOT re-execute verification. Instead:

```
Pillar nodes sample account-chain blocks
If valid: Momentum records checkpoint (AccountID, BTCTxHash, Confirmed, Height)
If invalid: Account-chain block excluded from momentum history
Cost: O(1 signature verification) not O(SPV verification)
```

Note on sampling vs. safety: Pillar sampling is optional and affects liveness/hygiene, not safety. Safety derives from deterministic verification: any honest participant (Pillar or not) can verify an SPV proof is invalid. An account-chain block with an invalid proof cannot be checkpointed by honest Pillars regardless of whether they pre-sampled it—honest Pillars will reject it upon checkpoint verification. Sampling frequency is purely an optimization for early spam detection.

Adversarial model: Under Byzantine assumptions (>67% honest Pillar stake):
• **Invalid proof submission:** Detected deterministically by any verifier; excluded from momentum by honest majority. Sampling frequency irrelevant.
• **Eclipse attack on account:** Account receives false headers; account-chain block may be well-formed but based on wrong Bitcoin view. Honest Pillars checkpoint it if it passes Zenon validity (structure, signatures). **This is where header source diversity matters** (see 4.2).
• **>33% Pillar collusion:** Attackers can checkpoint arbitrary state. Sampling provides no additional security in this regime (general consensus failure).

5.3 Cost Distribution

This architecture achieves heterogeneous cost distribution:

- **Account performing verification:** Pays full SPV verification cost (~100 SHA-256 ops)

- **Pillar nodes:** May sample for validity (optional, not consensus-critical)

- **Other accounts:** Pay zero cost—completely unaffected by Bitcoin verification activity

This is impossible in monolithic architectures where all computation passes through a global state machine.

6. Efficiency Analysis: Zenon vs. Alternatives

6.1 Computational Cost Model

Assume:

• n = total validators in network

• k = accounts interested in Bitcoin verification (k << n)

• v = cost of single SPV verification (~100 SHA-256 operations)

Architecture	Per-Proof Cost	Network Scaling
Ethereum	O(n × v)	All validators execute in EVM
Cosmos IBC	O(n × header_sync)	All validators maintain light client
Polkadot	O(parachain_core)	Dedicated parachain required
Zenon NoM	O(k × v + ε)	Only interested accounts verify

Where ε_consensus is the marginal momentum checkpointing cost (one signature verification, ~0.1ms).

Key result: When k << n (few accounts use Bitcoin verification), Zenon achieves order-of-magnitude efficiency improvement.

6.2 Storage Scaling

Bitcoin header chain: 80 bytes/block × 870,000 blocks ≈ 70 MB (and growing at ~4 MB/year).

Ethereum/Cosmos: Every validator must store the full header chain or trust external checkpoints (introducing trust assumptions).

Zenon:

• Accounts store only headers relevant to their verifications (sparse storage)

• Momentum chain stores only finality checkpoints (compact: ~100 bytes per checkpoint)

• Pillar nodes MAY maintain full Bitcoin header view for sampling (optional infrastructure)

• Storage scales with usage, not network size

6.3 Concrete Example

Consider a Bitcoin bridge used by 10 accounts in a network of 30 pillar nodes. Each SPV verification requires approximately 100 SHA-256 operations (~1ms on modern hardware).

Metric	Ethereum (Global)	Zenon (Local)
Verifiers per proof	30 validators	10 accounts
SHA-256 ops per proof	30 × 100 = 3,000	10 × 100 = 1,000
Consensus overhead	30 × full EVM exec	30 × 0.1ms signature check
Unused accounts affected	100% pay gas	0% affected
Total network cost	~3,000 compute units	~1,000 compute + 3ms consensus

Result: Zenon achieves 3× reduction in verification cost, with consensus overhead relegated to lightweight checkpointing. As the ratio k/n decreases (more pillars, same bridge users), the efficiency gap widens further.

7. Security Reduction

7.1 Combined Security Model

System security reduces to the conjunction of two independent security domains:

1. **Bitcoin proof-of-work security:** Reversing a z-confirmation transaction requires (q/p)^z attack probability

2. **Zenon consensus safety:** Checkpointing an invalid proof requires >33% pillar weight collusion

Critical property: An adversary must compromise BOTH security domains to successfully attack:

- Compromising only Bitcoin: Cannot convince Zenon validators to accept invalid proof (verification is deterministic)

- Compromising only Zenon: Cannot create valid Bitcoin proof without actual Bitcoin confirmation

- Must compromise both: Security domains multiply, not weaken

7.2 Attack Vectors and Mitigations

Scenario A: Bitcoin Reorg (depth < z)

- **Attack:** Reorg Bitcoin chain after proof is submitted but before z confirmations

- **Mitigation:** Account-chain verification fails when re-checking against new canonical chain. Momentum does not checkpoint invalid state. No Zenon corruption.

Scenario B: Invalid Proof Submission

- **Attack:** Submit proof with fake PoW, invalid Merkle branch, or incorrect chainwork

- **Mitigation:** Local verification rejects. Momentum excludes invalid account-chain block. Attacker wastes only their own account resources (plasma/PoW).

Scenario C: Pillar Collusion (>33% attack)

- **Attack:** Colluding pillars checkpoint invalid Bitcoin proof despite verification failure

- **Mitigation:** This is a general Zenon consensus failure (not specific to Bitcoin SPV). Economic security: pillars risk slashing/delegation loss. Attack cost exceeds any plausible Bitcoin bridge value.

7.3 Formal Security Statement

Theorem (Informal): If Bitcoin provides (q/p)^z security for transaction finality and Zenon provides Byzantine fault tolerance with >67% honest stake, then a Bitcoin proof checkpointed in Zenon achieves security min(Bitcoin_security, Zenon_security).

Neither attack vector is amplified by SPV verification itself. The combined system is at least as secure as its weakest component.

8. Novel Architectural Properties

8.1 Heterogeneous Trust Models

Zenon enables users to opt-in to Bitcoin verification rather than forcing global participation:

```
// Account A: Bitcoin bridge (maintains Bitcoin header view)
VerifyBitcoinDeposit(proof) → MintWrappedBTC(user, amount)

// Account B: DeFi protocol (uses wrapped BTC)
SwapTokens(wrappedBTC, ZNN) → ExecuteTrade()

// Account C: Unrelated application
// Never touches Bitcoin, pays ZERO overhead for A's verification
```

This is architecturally impossible in Ethereum where every contract execution consumes global validator resources.

13. Conclusion

Core Finding: Bitcoin SPV provides sufficient cryptographic foundation for Zenon to verify Bitcoin transaction inclusion as a ledger-native fact. The novelty lies not in cryptography (which is standard), but in ledger semantics—Zenon can treat external consensus outcomes as first-class inputs without executing foreign logic or relying on secondary consensus layers.

13.1 What Makes This Special

Zenon's dual-ledger architecture enables Bitcoin SPV with properties unavailable to monolithic blockchains:

1. **Efficiency:** O(k) cost where k = interested accounts, not O(n) where n = all validators

2. **Scalability:** Storage and computation scale with usage, not network size

3. **Composability:** Multiple accounts can consume Bitcoin data without amplifying validator burden

4. **Heterogeneity:** Users who don't care about Bitcoin pay zero overhead

5. **Security:** Defense-in-depth combining Bitcoin PoW with Zenon finality

13.3 For Developers

If implemented, Zenon would offer Bitcoin integration that is:

- **Cheaper:** No continuous gas payments to maintain header chain

- **Simpler:** Verification is native ledger operation, not smart contract

- **More scalable:** Cost doesn't scale with network size

- **Equally secure:** Same Bitcoin SPV security guarantees as any other implementation

The question for developers isn't whether SPV is possible (it clearly is), but whether Zenon's architecture makes it practical at scale.

The answer is yes.

References

[Nakamoto 2008] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System." 2008.

[Zenon 2018] Zenon Network. "Zenon Whitepaper: A Scalable Dual-Ledger Architecture." 2018.

[Zenon 2019] Zenon Network. "Zenon Lightpaper: Network of Momentum." 2019.

[BTC Relay 2016] "BTC Relay: Ethereum Smart Contract for Bitcoin SPV." Consensys, 2016.

[Cosmos IBC 2021] "Inter-Blockchain Communication Protocol." Cosmos Network, 2021.

[Polkadot 2020] Gavin Wood. "Polkadot: Vision for a Heterogeneous Multi-Chain Framework." 2020.

Disclaimer

This document establishes cryptographic and architectural feasibility. It does not constitute active development, implementation commitment, or protocol activation plan. All decisions regarding implementation remain subject to Zenon governance processes.

Appendix B specifically represents engineering recommendations and does not modify the core feasibility claims of the research paper.