Verifiability and Fate-Sharing in a Cryptographic Hourglass

Stark · January 2026

Abstract

The architectural parallel between TCP/IP's foundational design philosophy and verification-first architecture reveals a profound continuity in systems thinking: both prioritize correctness at endpoints over centralized intermediation. Where the Internet's architects chose "fate-sharing" over replicated state and datagrams over virtual circuits, verification-first systems apply these same principles to cryptographic verification, pushing trust to the mathematical primitives themselves rather than delegating it to network authorities. Primary source citations demonstrate how Internet design philosophy offers both precedent and vocabulary for understanding verification-first blockchain architecture.

Executive Summary

The TCP/IP split of 1977-1981 created the minimal, robust foundation that enabled Internet scale. Zenon's dual-ledger separation of ordering (Momentum chain) from verification (bounded local checking) follows the identical architectural logic. Current Internet security (TLS/PKI, DNSSEC, BGP) failed to preserve end-to-end principles, creating the centralized trust hierarchies that verification-first systems now seek to eliminate.

Core thesis: Zenon applies the end-to-end principle from TCP/IP architecture to verification, treating correctness as a property that belongs at the edges, not in the network core.

Part I: TCP/IP's Separation of Concerns Established the Template

The Internet's architectural foundation was not inevitable but chosen. **David D. Clark's 1988 SIGCOMM paper** "The Design Philosophy of the DARPA Internet Protocols" explicitly states the priority hierarchy: survivability first, then service diversity, then network accommodation. Clark wrote that these goals "are in order of importance, and an entirely different network architecture would result if the order were changed." [Clark 1988, ACM SIGCOMM]

The critical architectural decision was separating TCP from IP. **Jon Postel's IEN #2 (August 1977)** provided the pivotal critique:

> *"We are screwing up in our design of internet protocols by violating the principle of layering. Specifically we are trying to use TCP to do two things: serve as a host level end to end protocol, and to serve as an internet packaging and routing protocol. These two things should be provided in a layered and modular way." [Postel, IEN #2, 1977]*

This separation created IP as what became known as the "narrow waist" or "hourglass model," a minimal spanning layer between heterogeneous lower-layer technologies and diverse upper-layer applications. **RFC 791 (September 1981)** codified this minimalism explicitly:

> *"The internet protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an internet datagram) from a source to a destination over an interconnected system of networks. There are no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols." [RFC 791]*

The rationale for this minimalism appears in Clark (1988):

> *"This building block was the datagram, which had also been adopted to support survivability. Since the reliability associated with the delivery of a datagram was not guaranteed, but 'best effort,' it was possible to build out of the datagram a service that was reliable (by acknowledging and retransmitting at a higher level), or a service which traded reliability for the primitive delay characteristics of the underlying network substrate." [Clark 1988]*

Part II: Fate-Sharing Eliminates Verification State in Network Core

The term "fate-sharing" originates from Clark's 1988 paper, where he articulated why connection state belongs at endpoints rather than in the network:

> *"The alternative, which this architecture chose, is to take this information and gather it at the endpoint of the net, at the entity which is utilizing the service of the network. I call this approach to reliability 'fate-sharing.'" [Clark 1988]*

The core principle:

> *"The fate-sharing model suggests that it is acceptable to lose the state information associated with an entity if, at the same time, the entity itself is lost." [Clark 1988]*

Clark identified two advantages of fate-sharing over replication:

> *"First, fate-sharing protects against any number of intermediate failures, whereas replication can only protect against a certain number (less than the number of replicated copies). Second, fate-sharing is much easier to engineer than replication." [Clark 1988]*

The consequence was stateless intermediate nodes:

> *"There are two consequences to the fate-sharing approach to survivability. First, the intermediate packet switching nodes, or gateways, must not have any essential state information about on-going connections. Instead, they are stateless packet switches, a class of network design sometimes called a 'datagram' network." [Clark 1988]*

This architectural decision (**no connection state in the network core**) directly prefigures Zenon's approach where verifiers maintain their own truth through local computation rather than depending on centralized verification state.

Part III: End-to-End Arguments Define Where Correctness Belongs

Saltzer, Reed, and Clark's 1984 paper "End-to-End Arguments in System Design" (ACM Transactions on Computer Systems) established the theoretical foundation for pushing complexity to network edges:

> *"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible." [Saltzer, Reed, Clark 1984]*

The authors presented this as a design principle for function placement:

> *"This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level." [Saltzer, Reed, Clark 1984]*

They described this principle as "a kind of 'Occam's razor' when it comes to choosing the functions to be provided in a communication subsystem." [Saltzer, Reed, Clark 1984]

The paper's "careful file transfer" example demonstrates why correctness must be verified end-to-end: intermediate network checks cannot guarantee application-level correctness. Only the endpoints possess complete knowledge of what constitutes correct behavior. **This is the precise argument for verification-first architecture**: cryptographic verification at endpoints replaces trust in intermediate authorities.

Part IV: Current Internet Security Abandoned End-to-End Principles

The modern Internet security stack (TLS/SSL, PKI, DNSSEC) systematically violates end-to-end principles by requiring trust in centralized intermediaries.

Certificate Authority Hierarchy Creates Structural Vulnerability

The TLS/SSL Public Key Infrastructure operates on a hierarchical trust model where trust flows from self-signed root certificates to end-entity certificates. As documented by ENISA: "In this sense, each CA is a single point of failure." [ENISA, Operation Black Tulip] As of August 2020, browser trust stores contained **147 root certificates from 52 organizations** (Mozilla Firefox), **168 root certificates from 60 organizations** (macOS), and **255 root certificates from 101 organizations** (Microsoft Windows) [Mozilla Wiki CA:IncludedCAs; Apple Support HT208127; Microsoft CCADB]. Any single Certificate Authority (CA) compromise allows certificate forgery for any domain.

RFC 5280 (Internet X.509 PKI Certificate and CRL Profile) establishes the technical framework:

> *"The CA may base this assertion upon technical means (a.k.a., proof of possession through a challenge-response protocol), presentation of the private key, or on an assertion by the subject." [RFC 5280]*

The trust model requires believing that CAs properly validate identity, a social trust assumption rather than a cryptographic guarantee.

DigiNotar Demonstrated Systemic CA Vulnerability

The **DigiNotar breach (2011)** proved the catastrophic potential of centralized trust. The Fox-IT investigation revealed total compromise:

> *"The investigation by Fox-IT showed that all eight servers that managed Certificate Authorities had been compromised by the intruder... In total, a non-exhaustive list of 531 rogue certificates with 140 unique distinguished names (DNs) and 53 unique common names (CNs) could be identified." [Fox-IT, Black Tulip Report]*

Fox-IT identified **300,000 Iranian Gmail accounts** as the main victims of subsequent man-in-the-middle attacks. Dutch MP Van Dam stated the actions "had put lives at risk." DigiNotar declared bankruptcy within three weeks of public disclosure.

The **Comodo breach (March 2011)** demonstrated how the federated CA model extends attack surface. The attacker compromised a reseller, not Comodo directly, yet issued fraudulent certificates for multiple high-value domains including mail.google.com, login.yahoo.com, login.skype.com, addons.mozilla.org, and login.live.com (nine certificates total across seven domains).

BGP Routing Has No Native Verification

Border Gateway Protocol has no built-in authentication. As documented by network security researchers:

> *"By default, BGP does not embed any security protocols. It is up to every autonomous system to implement filtering of 'wrong routes'... BGP assumes all participants provide accurate routing information, adhering to established rules. This inherent trust is where vulnerabilities emerge." [Cloudflare BGP Research]*

RPKI (Resource Public Key Infrastructure, RFC 6480) attempts to add cryptographic verification, but adoption remains limited: "As of today, July 27, 2023, only about 45% of the IP prefixes routable on the Internet are covered by some ROA on RPKI... Based on our recent study, only 6.5% of the Internet users are protected by ROV from BGP origin hijacks." [Cloudflare]

DNSSEC Centralizes Trust in Single Anchor

DNSSEC provides cryptographic signatures for DNS data but depends on a single trust anchor, the root zone key signing key (KSK). Per ICANN:

> *"The DNS is protected at the highest level using a seal of authenticity, known as the 'root zone key signing key'. This is often referred to more simply as the 'key signing key,' 'root key,' the root 'KSK,' or the 'trust anchor' for the DNS." [ICANN]*

Physical storage is limited to two facilities (El Segundo, California and Culpeper, Virginia), with only 14 Crypto Officers worldwide authorized to participate in key ceremonies.

Certificate Transparency Provides Detection, Not Prevention

RFC 6962 describes Certificate Transparency's goal:

> *"Certificate transparency aims to mitigate the problem of misissued certificates by providing publicly auditable, append-only, untrusted logs of all issued certificates." [RFC 6962, Section 1]*

The append-only property is achieved using Merkle Trees. However, CT emerged as a response to CA compromises; it detects mis-issuance after certificates exist, not before. Organizations must actively monitor logs, and log operators themselves become centralization points.

Part V: Zenon's Dual-Ledger Architecture Separates Ordering from Verification

Zenon Network implements architectural separation parallel to TCP/IP's split. The **Network of Momentum whitepaper** (March 31, 2020) describes:

> *"The protocol comprises of a dual ledger architecture, a meta-DAG created by participating consensus nodes, a projection of the meta-DAG that represents the transactional ledger, a proof-of-work link between relayed transactions emitted by clients." [Zenon Whitepaper]*

The **block-lattice layer** (transaction ledger) stores settled transactions where each user has an independent account chain that can be updated asynchronously. The **meta-DAG layer** (consensus ledger) contains transactions required by the virtual voting algorithm.

The architectural rationale appears in community documentation: "The idea behind a dual ledger system is to decouple consensus from chain weight. This is very important because consensus and chain weight are fundamental for a L1."

Momentum Chain Provides Minimal Ordering Primitive

Momentums are packages of confirmed account blocks produced at regular intervals, similar to blocks in traditional blockchains but comprising information from both the DAG layer and Block-Lattice layer. Momentums are tamper-proof and irreversible once published. Participating nodes are selected based on a delegation-based virtual vote to produce Momentums.

This is the "minimal datagram building block" principle applied to consensus: Momentum provides ordering without mandating execution semantics.

The ordering stream itself need not originate from any particular transport or network topology. What matters is the cryptographic structure: tamper-evident, sequentially hashed commitments with Merkle inclusion proofs. Asynchronous or delay-tolerant delivery changes latency, not correctness.

Virtual Voting Eliminates Verification Intermediaries

The whitepaper describes virtual voting:

> *"Virtual voting - the concept that voting is not done with explicit messages. Instead, a node computes the state of the ledger based on the information received throughout many epochs from the network." [Zenon Whitepaper]*

Consensus derives from protocol rules, not explicit vote messages. After receiving confirmation from a supermajority ($\zeta = N \times 2/3 + 1$), nodes reach consensus locally. The whitepaper states:

> *"If a node will know the transactions between epoch ε_0 and epoch ε_1 and it will apply a deterministic ordering algorithm and in case of double spends, a deterministic tie-breaker algorithm, thus all the remaining honest nodes will arrive at the same decision." [Zenon Whitepaper]*

This is **fate-sharing applied to verification**: each node maintains its own truth through local computation. No central verification state exists that could fail independently of the nodes depending on it.

Part VI: The Narrow Waist Concept Maps to Verification Primitives

Academic research explicitly connects Internet architecture to blockchain design. **Hardjono, Lipton, and Pentland's "Towards a Design Philosophy for Interoperable Blockchain Systems" (arXiv: 1805.05934, 2018)** directly maps DARPA Internet design philosophy to blockchain:

Internet Concept	Blockchain Equivalent
Survivability	Transaction completion despite chain failures
Autonomous Systems (AS)	Blockchain systems as autonomous domains
Gateways (BGP)	Cross-chain bridges/gateways
End-to-end principle	Value semantics at endpoints
Datagram (minimum unit)	Minimum cross-chain transaction unit

The MIT research notes: "The two level view follows the end-to-end principle by placing the human semantics (value) at the ends of (outside) the mechanical systems." [Hardjono, Lipton, Pentland 2018]

Akhshabi and Dovrolis's "The Evolution of Layered Protocol Stacks" (ACM SIGCOMM 2011) demonstrated that protocol stacks naturally evolve into hourglass shapes. Section 6 examines "the evolutionary kernels of the architecture, i.e., those few nodes at the waist that survive much longer than other nodes." The narrow waist concept applied to verification suggests **minimal verification primitives** (hash functions, digital signatures, Merkle proofs) that many applications build upon, the same principle that made IP the universal interconnection layer.

This architecture is agnostic to the origin of its ordering stream. Any source providing a tamper-evident, sequentially hashed chain of events (block headers with Merkle inclusion proofs) can serve as input. Endpoints consume the stream, execute locally, and verify outcomes under bounded resources. The verification layer treats all such streams as equivalent: no architectural preference distinguishes internal chains from external ledgers or legacy systems with signed logs.

Part VII: Verification Without Trust Inverts the PKI Model

The fundamental distinction between current Internet security and verification-first architecture:

TLS/PKI Model	Verification-First Model
Trust delegated to Certificate Authorities	Verification is cryptographic
CA must properly validate identity	Trust only cryptographic primitives
~100+ centralized trust points	Mathematical certainty over social trust
Single CA compromise affects all domains	No centralized trust to compromise
Post-hoc detection (CT)	Pre-execution verification

The verification-first principle as articulated in Zenon design documents: "L1 should be minimal, robust and as efficient as possible... A simple and robust L1 with minimal features will always outperform over-engineered and complex designs that try to accomplish too many things at once. Bitcoin does exactly this: it is indeed minimal (non-Turing complete) and very robust."

Heavy execution happens off-chain; verification happens on-chain through cryptographic proofs that are **source-agnostic**, with validity determined by mathematics rather than trust in the proof provider.

Part VIII: Cryptographic Primitives Enable Trustless Verification

The theoretical foundation for verification without trust rests on self-verifying data structures:

Merkle trees (Ralph Merkle, patented 1979) allow verification that a data element belongs to a large dataset in O(log n) time. Each block header contains a Merkle root summarizing all transactions, enabling lightweight verification without trusting the data provider.

Hash chains create tamper-evident history where each element contains the hash of the previous element. Blockchain block linking (where each block's hash includes the previous block's hash) provides integrity guarantees through mathematics rather than authority.

Zero-knowledge proofs extend this paradigm by enabling verification without revealing underlying data. ZK-rollups bundle thousands of transactions into single proofs, maintaining correctness guarantees while scaling verification.

These primitives replace the CA hierarchy's social trust with mathematical certainty. Where RFC 5280 requires "an assertion by the subject" validated by CA judgment, cryptographic verification requires only that hash functions and signature schemes remain secure, a much narrower trust assumption.

Conclusion: Architectural Principles Persist Across Technological Paradigms

The TCP/IP designers faced a choice: virtual circuits with centralized state, or datagrams with endpoint responsibility. They chose the latter, creating the architectural foundation that enabled Internet scale. Forty years later, Internet security has drifted toward exactly the centralized trust model the original architecture avoided: Certificate Authorities as trusted intermediaries, DNSSEC root keys held by a single organization, BGP routing without verification.

Zenon's verification-first architecture represents a return to first principles. The dual-ledger separation of ordering from verification parallels the TCP/IP split. Fate-sharing manifests in nodes maintaining their own truth through local computation. The end-to-end argument applies directly: correctness belongs at endpoints, verified through cryptographic proofs rather than delegated to intermediate authorities.

The narrow waist concept suggests the future: minimal verification primitives (hash functions, signatures, zero-knowledge proofs) as the spanning layer between diverse execution environments and applications. Where IP provided the minimal building block for packet delivery, cryptographic verification provides the minimal building block for trustless computation.

Clark's 1988 observation remains relevant:

> *"There are two advantages to fate-sharing over replication. First, fate-sharing protects against any number of intermediate failures, whereas replication can only protect against a certain number." [Clark 1988]*

Verification-first architecture extends this principle: cryptographic proofs protect against any number of dishonest intermediaries, whereas trust-based systems can only protect against a certain number of compromised authorities.

Future Research Directions

This architectural framework suggests several avenues for further investigation.

Transport layer independence. The verification-first model requires eventual delivery of tamper-evident sequences, not the TCP/IP stack specifically. Extending the transport layer to mesh networks, delay-tolerant protocols, or pure P2P gossip could enable deployment in resource-constrained or partition-prone environments while preserving the core sequencing requirement. The key constraint is cryptographic structure, not synchronous connectivity.

Proof distribution economics. When verification capacity is scarce, proofs become economic objects. Refusals (verification failures due to missing data or exceeded resource bounds) function as demand signals, suggesting formal analysis of proof distribution markets: pricing mechanisms, caching incentives, and equilibrium conditions under adversarial withholding.

Post-quantum verification primitives. The current narrow waist assumes collision-resistant hash functions and ECDSA/Schnorr signatures remain secure. Transitioning to post-quantum primitives (hash-based signatures, lattice constructions) while maintaining bounded verification costs presents an open design challenge.

Primary Source Reference Index

TCP/IP Foundational Documents

- **Cerf, V.G. and Kahn, R.E.** "A Protocol for Packet Network Intercommunication." *IEEE Transactions on Communications*, Vol. COM-22, No. 5, May 1974, pp. 637-648.
- **Clark, D.D.** "The Design Philosophy of the DARPA Internet Protocols." *ACM SIGCOMM Computer Communication Review*, Vol. 18, No. 4, August 1988, pp. 106-114.
- **Saltzer, J.H., Reed, D.P., and Clark, D.D.** "End-to-End Arguments in System Design." *ACM Transactions on Computer Systems*, Vol. 2, No. 4, November 1984, pp. 277-288.
- **Postel, J.** "Comments on Internet Protocol and TCP." IEN #2, 15 August 1977.
- **Postel, J. (Editor)** "Internet Protocol - DARPA Internet Program Protocol Specification." RFC 791, September 1981.
- **Postel, J. (Editor)** "Transmission Control Protocol - DARPA Internet Program Protocol Specification." RFC 793, September 1981.

Internet Security Standards

- **RFC 5280**: Internet X.509 Public Key Infrastructure Certificate and CRL Profile

- **RFC 6962**: Certificate Transparency
- **RFC 8555**: Automatic Certificate Management Environment (ACME)
- **RFC 6480**: An Infrastructure to Support Secure Internet Routing (RPKI)

Incident Reports and Analyses

- **Fox-IT** "Black Tulip Report" on DigiNotar compromise (2011)
- **ENISA** "Operation Black Tulip: Certificate authorities lose authority"
- **van der Meulen, N.** "DigiNotar: Dissecting the First Dutch Digital Disaster." *Journal of Strategic Security* (2013)

Blockchain Architecture Research

- **Hardjono, T., Lipton, A., and Pentland, A.** "Towards a Design Philosophy for Interoperable Blockchain Systems." arXiv:1805.05934, 2018.
- **Akhshabi, S. and Dovrolis, C.** "The Evolution of Layered Protocol Stacks." *ACM SIGCOMM*, 2011.
- **Network of Momentum Whitepaper** "Leaderless BFT dual ledger architecture." DRAFT v0.1, March 31, 2020.

Theoretical Foundations

- **Merkle, R.** "A Certified Digital Signature." 1989.
- "On The Hourglass Model." *Communications of the ACM*, 2019.