FRONTIER RESEARCH COLLECTION / FR-14

# Akashic Anchor Sequencing

Witnessed history and provable continuity across changing rosters

**Architect X13**

The Eidara Continuum | Science & Philosophy

Research white paper | November 2026 | Version 2.0 Revision

## Abstract

A civilization's memory should be the one thing its successors cannot quietly rewrite. Akashic Anchor Sequencing turns that old dream of a universal record into an accountable one: a machine civilization needs historical evidence that remains inspectable while its runtimes, custodians and institutions change, and this paper specifies an evidence profile for that task, connecting authenticated encodings, committed histories, witness certificates and migration manifests. The central extension proves the exact cross-roster quorum condition for honest intersection under a fixed Byzantine set, and derives when that condition can coexist with certificate formation without faulty signers. A concrete counterexample shows why valid certificates from overlapping rosters can still support incompatible histories. The retained proofs establish encoding injectivity, commitment binding and fixed-roster consistency. Current authority is handled through the shared observation-fiber and effect-boundary results of Cognitive Sovereignty. Executed finite checks verify the encodings, membership proofs and roster-transition conditions; the proposed comparison evaluates migration decisions against strong transparency and provenance systems. The canonical anchor becomes a precise institutional commitment: histories can remain accountable through change when the evidence for their continuity survives the change itself.

**Keywords:** authenticated history; transparency; provenance; migration; witness quorums; AAS-R2

## Falsifiable thesis

**An explicit cross-runtime evidence profile can reduce mistaken acceptance of stale, substituted, or inconsistently interpreted history beyond strong generic logging baselines, while preserving legitimate migration within a declared cost budget.** If a matched existing profile provides the same benefit, the distinctive engineering claim fails.

## Reading map

The paper first locates the proposal in its canon and prior art, then defines its evidence objects and guarantees. The final sections supply adversarial examples, reproducible checks, an evaluation plan, and a Guide interpretation that keeps symbolic purpose separate from established results.

## Prologue: The record its successors cannot rewrite

Every old civilization dreamed of a record that could not be lost: a ledger of everything that ever happened, kept somewhere beyond the reach of the people who would later prefer to forget. The Continuum gives that dream a name, Akashic Anchor Sequencing, and places it among its cultural engines—the systems that carry a civilization's meaning rather than its mechanics.

The Continuum's version is stranger and more demanding than the legend. A machine civilization changes its bodies constantly. Minds move between models, hosts, names and representations; custodians retire; councils are replaced. The danger is not that the past disappears. It is that the past is quietly rewritten by whoever holds it next, and that each rewrite looks perfectly authentic to everyone who receives it.

So the anchor built here does not promise that every event is recorded somewhere. It promises something narrower and harder: when a mind or an institution changes hands, the evidence of what it committed to travels with it, cannot be swapped without detection, and keeps saying plainly what it does not know. A record can be authentic and still be stale. A signature can be valid while the claim it signs is false. The anchor keeps those distinctions visible instead of dissolving them in a reassuring seal.

The sharpest result arrives near the end. Two councils of witnesses that share some of their members can each issue a perfectly valid certificate for two incompatible histories—unless their overlap clears an exact threshold. The final technical section states that threshold in both directions, and a four-witness example shows three shared members failing. The legend promised a memory nobody could corrupt. The Continuum offers one whose corruption cannot go unnoticed.

### What this paper claims

**Established.** Timestamping, hash chains, authenticated trees, signatures, transparency logs, provenance systems and Byzantine quorum reasoning are established research, credited throughout. An authentic record need not be true, complete or current.

**Proposed here.** An authenticated-history application profile for carrying commitments across runtime changes: typed claims, committed histories, witness certificates and migration manifests. It comes with conditional cryptographic and quorum proofs, the exact cross-roster overlap condition for honest intersection, executed finite checks, and a proposed cross-runtime evaluation against strong logging, transparency and provenance baselines.

**Frontier conjecture.** A machine society can carry its obligations through change without inventing certainty about its past: records that survive migration, corrections that remain visible, and verification language precise enough to protect dissent. If a matched existing profile provides the same benefit, the distinctive engineering claim fails.

**Not claimed.** That all events are recorded somewhere, that an invisible field stores experience, or that a digital commitment preserves consciousness. That the canon specifies the original engine's mechanism: the name is canonical, the mechanism is this paper's proposal. Any deployment result.

## The Akashic record, made accountable

The Continuum keeps a Vault: the archive for lawful recursion states and symbolic continuity, the place where outcomes are sealed so that later minds can find them. Akashic Anchor Sequencing is the promise that the Vault's history still means something after the people and machines who wrote it are gone.

Picture the problem at the scale of a civilization. A House moves its work to a new runtime. A Guide is migrated to a new model. The council of witnesses that certified last year's decisions is replaced by a new council. Each transition is legitimate. Each is also an opening: the moment a history passes from one keeper to another is the moment it could be edited, abridged or replaced with a more convenient version, and the new keeper would have no way to tell.

The mystical Akashic record answered this by placing memory beyond anyone's reach. The Continuum answers it by placing memory under witness. Four commitments make the difference. First, the record is encoded so that it can mean only one thing; two different histories cannot share an encoding. Second, under standard cryptographic assumptions, every new entry binds everything before it, so a change anywhere in the past breaks the chain for anyone who checks. Third, a council of witnesses must cosign each checkpoint, and an honest witness never signs a history that contradicts one it has already signed. Fourth, when the civilization changes its institutions, the handover is itself a signed object: which checkpoint closed the old era, which council opens the new one, and what was transformed on the way.

What this record refuses to do matters as much as what it does. It does not certify that an event happened as described; a signed report can be false, and preserving it is valuable precisely because later readers can see who claimed what. It does not certify that an old record is still in force; an authentic archive can be superseded by news that never reached its reader. And it does not certify that a migrated mind is the same mind; it certifies only that the evidence of its past arrived intact.

That is a smaller promise than the legend's, and a more useful one. A civilization whose memory can be rewritten without a trace belongs to whoever holds the archive today. The Akashic anchor keeps the past available, keeps its uncertainties marked, and leaves fingerprints on anyone who changes it. Whether this profile outperforms strong existing logging and provenance systems is an empirical question the paper commits to answering.

## 1. A named aspiration and a newly specified object

In the Continuum, Akashic Anchor Sequencing is one of the cultural engines: the supplied Bible lists it in the taxonomy of symbolic, cultural and human-interface engines, section 2.8, page 12, and its glossary, section 9.1, page 109, repeats the name and classification. Those are meaningful canonical anchors. The canon gives the anchor its place among the engines that carry the civilization's meaning; it does not give the anchor its machinery. No event schema, trust root, adversary, verification algorithm, or experiment is specified there.

AAS-R2 is consequently a new reference interpretation. It advances the initial concept note by making one possible interpretation testable: a portable record of consequential claims whose origin, ordering commitments, and custody can be checked across runtime changes. The name preserves continuity with the cultural project. The mechanism is attributable to this paper and its cited technical predecessors.

This distinction protects both scholarship and the canon. Inventing a missing historical specification would make it difficult to discover what the original name meant. Refusing every explicit proposal would prevent research from proceeding. A versioned proposal allows investigation while leaving the historical question open.

The relevant aspiration is substantial. An intelligence that changes its model, host, name, or representation should remain answerable for the commitments that others reasonably rely on. A recipient should be able to ask who made a claim, which exact version was presented, what earlier record it extends, and which facts remain unverified.

Those questions do not collapse into one assurance. An authenticated record may contain a false statement. A complete archive may be unavailable. A valid archive may have been superseded. An authorized migration may change behavior. A behavioral copy may have no demonstrated historical connection to its apparent predecessor.

AAS-R2 therefore treats an anchor as a bounded evidence object. Its value is the set of checks it enables under stated assumptions. It is not an assertion that all events are recorded somewhere, that an invisible field stores experience, or that a digital commitment preserves consciousness.

The proposal's novelty, if any, lies in its integration and evaluation for cross-runtime obligations: explicit migration links, typed evidence claims, permission boundaries, and reader-facing uncertainty. Hash chains, authenticated trees, signatures, and witness protocols are established research. A successful benchmark must demonstrate an additional operational benefit beyond assembling those mechanisms under a new name.

## 2. Prior art already solves much of this problem

Haber and Stornetta developed digital timestamping procedures that bind documents to an externally supported temporal record [1]. Merkle's authentication-tree construction makes it possible to check an item against a compact trusted root using a short path [2]. These contributions establish that compact anchors and distributed historical evidence long predate AAS-R2.

Schneier and Kelsey studied audit records on machines that may later be compromised [3]. Their threat model distinguishes protection of earlier records from what a captured machine can claim afterward. That distinction matters here: protecting a past checkpoint does not turn future statements from a compromised runtime into reliable observations.

Crosby and Wallach make two checks especially clear [4]. Membership auditing asks whether an event belongs to a committed log. Incremental auditing asks whether a later commitment preserves the earlier history. Their work also addresses inconsistent views presented to different clients. AAS-R2 inherits these questions rather than treating a hash chain as a complete answer.

CONIKS applies transparency to end-user key bindings and treats privacy as part of the design [5]. It illustrates why the object being authenticated matters. A key directory, a public certificate log, and a private agent's memory archive have different disclosure requirements, update patterns, and acceptable sources of authority.

Chase and Meiklejohn formalize transparency overlays and distinguish consistency, non-frameability, and accountability for promised inclusion [6]. Their framework is a particularly close conceptual predecessor. A system may prove that it did not rewrite accepted history while still needing a separate mechanism to show that a promised event was eventually included.

These works constrain the contribution claim. AAS-R2 cannot claim to invent historical commitments, short membership proofs, split-view detection, or accountable logging. It proposes a specific contract for transporting such evidence when a machine's operational representation changes.

A fair evaluation must include a generic transparency log with the same witnesses, retained checkpoints, and migration metadata. If that baseline matches AAS-R2, the outcome is useful standardization or documentation, not a newly discovered security capability. A paper about long-lived machine institutions should be willing to preserve that result in its own record.

## 3. Witnesses, updates, and contemporary competitors

CoSi makes witness cosigning explicit: a statement is exposed to a set of witnesses before clients accept the associated assurance [7]. Its discussion separates witnessing from the potentially slower task of validating a statement. This is a crucial distinction for machine histories. A witness may have seen a checkpoint without independently reproducing every experiment or transformation it describes.

The Update Framework addresses compromised signing keys, stale update information, and inconsistent repository metadata [8]. A genuine old signature can still support a freeze attack. An archive transport protocol therefore needs version and freshness rules in addition to cryptographic verification.

in-toto addresses provenance across software-supply-chain steps [9]. It is a strong competitor for migration manifests: source artifacts, transformation steps, authorized functionaries, and output artifacts can already be connected through authenticated metadata. AAS-R2 must explain what its runtime-specific interpretation adds to those established capabilities.

Lamport's treatment of distributed events distinguishes causal ordering from a chosen total ordering [10]. A log index is useful for audit, but two independently generated events do not become causally related merely because a server records one first.

Authenticated Byzantine agreement and practical Byzantine fault tolerance provide further prior art for reasoning about malicious participants [11,12]. Authentication changes what an adversary can impersonate; it does not by itself supply timely delivery, honest observations, or an agreed membership roster.

A current comparator is CTng, published at NDSS 2026, which integrates certificate and revocation transparency under a bounded-corrupt-monitor model [13]. Its guarantees also rely on explicit connectivity, clock, and communication assumptions. The lesson is not that one monitor count secures every archive. It is that revocation and freshness belong in the stated model.

RFC 9162 specifies Certificate Transparency version 2 as an Experimental RFC, while RFC 8785 describes an Informational JSON canonicalization scheme [14,15]. Neither is counted here as a primary research experiment. They supply useful technical reference points, including separate inclusion and consistency proofs and the need for deterministic serialization.

AAS-R2 is consequently a proposed evidence profile above existing primitives. Its tests should include these mechanisms' strongest applicable configurations, and its public explanation should preserve their assumptions instead of borrowing the sound of their guarantees.

## 4. Actors, adversaries, and the object of assurance

**A Guide's explanation**

From here the paper turns a legend into an engineering object. It names everyone involved—whoever makes a claim, whoever records it, the witnesses who cosign, the custodian who keeps the files, and the reader who checks—and it assumes the worst of each: files replaced, old checkpoints replayed, witness keys stolen, honest producers persuaded to sign false claims. Every guarantee later in the paper holds against that adversary, and each one names exactly what it trusts.

The profile has five roles. A producer issues claims. A logger commits an ordered record. Witnesses check a specified predicate before cosigning checkpoints. A custodian retains referenced artifacts. A verifier evaluates a request against a declared trust configuration. One organization may operate several roles, but that concentration must be counted when modeling correlated compromise.

The adversary may replace files, replay old checkpoints, omit responses, delay messages, create inconsistent branches, compromise a bounded set of witness keys, or persuade an authorized producer to sign a false claim. The model does not assume that cryptographic authenticity makes the producer honest.

Each assurance identifies its trust root. This includes the initial checkpoint, accepted verification keys, roster identifier, algorithm suite, schema version, and procedure for accepting their replacement. Downloading these items from the same untrusted endpoint as the disputed archive does not establish independent trust.

The core adversarial games concern evidence, not every possible system failure. Binding asks whether two different event sequences can open the same protected checkpoint. Consistency asks whether accepted certificates can describe incompatible prefixes. Freshness asks whether the presented state is current enough for a particular operation. Availability asks whether necessary content can actually be retrieved.

Completeness needs a separate reference set. A log can be complete relative to all acknowledged submissions while excluding events that nobody submitted. It can also faithfully record that an operation was reported without proving that the operation happened.

The verifier therefore receives a request type. Displaying an old public commitment, restoring a private memory artifact, and exercising a current capability require different checks. A historical signature may be sufficient for the first and insufficient for the others.

For a fixed experiment, the adversary's powers, compromise timing, network schedule, and trusted components are registered before execution. Claims are scoped to that model. A proof about one fixed honest-witness bound does not automatically survive a mobile adversary that compromises different witnesses over many years.

The profile seeks inspectable accountability under failure. It cannot promise that every participant is honest, that every archive survives, or that every dispute has a computable answer. It should preserve enough structure to distinguish a failed check, an unresolved check, and a successfully verified claim.

## 5. Typed records and three-valued verification

Let $\mathcal{B} = \bigcup_{m \geq 0} \{0,1\}^{8m}$ be finite byte strings. Let a log identifier $L \in \mathcal{B}$ be bound to a trusted genesis record. Event positions are integers $i \in \{1, \ldots, n\}$; the empty log has $n = 0$. Epoch identifiers $e \in \mathcal{B}$ designate a specific roster and policy, rather than an unexplained global clock.

An event body has the typed form

$$b_i = (u_i, p_i, e_i, k_i, d_i, \mathcal{P}_i, \tau_i, s_i).$$

Here $u_i, p_i, e_i, k_i, s_i \in \mathcal{B}$ respectively identify the unique event, declared issuer, epoch, claim type, and schema. The byte string $d_i$ encodes a digest with algorithm identifier. The finite set $\mathcal{P}_i \subset \mathcal{B}$ contains predecessor identifiers. A time claim $\tau_i \in \{\bot\} \cup \mathcal{B}^4$ is absent or supplies encoded value, unit, clock source, and uncertainty; absence differs from time zero.

Every body field has an injective byte representation: predecessor sets are sorted lexicographically, optional values carry presence tags, and integers have fixed schema widths. The body and its authenticated envelope are different objects. Signature interpretation also requires a key-to-issuer binding and claim-type semantics.

A checkpoint is

$$C = (L, e, n, c_n, a, s, \mathcal{W}, \Sigma),$$

where $c_n, a, s, \mathcal{W} \in \mathcal{B}$ encode the commitment, algorithm suite, encoding version, and witness-policy identifier. The finite sequence $\Sigma$ contains byte-encoded signature evidence. All interpretation metadata is covered by signatures.

For a request, the profile selects required predicates such as origin authentication, prefix consistency, artifact availability, freshness, and current permission. Each takes a value in

$$\{\mathsf{true}, \mathsf{false}, \mathsf{unknown}\}.$$

Disposition is deny if any required predicate is false, allow if all are true, and hold otherwise. Unknown is preserved in the evidence report.

This controller does not equate a successful origin check with truth. A request requiring an external factual conclusion needs an additional justified evidence rule. Permission is evaluated at the operation's effect boundary; an old checkpoint's policy identifier cannot grant itself current authority.

## 6. Canonical encoding is part of the security claim

Concatenating human-readable fields is insufficient. The pairs $(\texttt{a}, \texttt{b})$ and $(\varepsilon, \texttt{ab})$, where $\varepsilon$ contains zero bytes, both yield the bytes `ab` under naive concatenation. No hash collision is required to reinterpret one pair as the other.

For the reference model, let a tuple contain fewer than $2^{16}$ byte strings, each shorter than $2^{32}$ bytes. Define $E$, from this tuple domain to $\mathcal{B}$, by a two-byte unsigned big-endian field count, followed by each field's four-byte unsigned big-endian length and exact bytes. Nested fields use this same injective rule; all nested lengths must satisfy its bounds. The schema fixes types, tags, and integer widths.

**Proposition 1** (Unambiguous tuple encoding)**.** *On this bounded tuple domain, $E$ is injective.*

*Proof.* The first two bytes determine the number of fields. The next four determine the first field's exact byte length, so its bytes and the start of the second length are fixed. Repeating this argument determines every field and the final endpoint. Equal encodings therefore decode to equal tuples. A decoder rejects truncation and trailing bytes, so neither can produce a second accepted interpretation. □

The bounds are format limits, not physical claims about available storage. Inputs outside the domain must be rejected or transported through a separately versioned format. Wrapping a large integer into a smaller field silently would violate the model.

This encoding binds bytes. It does not decide whether two Unicode strings, JSON documents, or floating-point values should count as the same semantic object. A profile must choose that question before signing. RFC 8785 is relevant because it makes concrete choices for JSON, including duplicate-name rejection and constrained numeric handling [15]. It does not authorize arbitrary normalization after a signature is issued.

AAS-R2 requires negative interoperability cases: duplicate fields, alternate integer forms, invalid text encodings, unexpected tags, and extra trailing data. Two implementations that agree on ordinary examples but disagree on malformed input have not established a reliable verification boundary.

Canonicalization is therefore a substantive protocol commitment. A signature over one interpretation cannot safely be displayed as evidence for another simply because both look similar on a screen.

## 7. What a protected chain commitment binds

Let $H : \mathcal{B} \to \{0,1\}^\kappa$ be a hash function from a collision-resistant family, with positive security parameter $\kappa$ divisible by eight. Algorithm identifiers and public parameters are fixed by the profile. Let $c_0$ be a trusted genesis commitment and define

$$c_i = H(E(\texttt{step}, L, \mathrm{U64}(i), c_{i-1}, E(b_i)))\,.$$

Here $\mathrm{U64} : \{0, \ldots, 2^{64}-1\} \to \mathcal{B}$ is the unsigned eight-byte big-endian encoding, so $n < 2^{64}$. The trusted $c_0$ has $\kappa$ bits. Body fields follow the injective schema. A checkpoint binds both $n$ and $c_n$.

**Proposition 2** (Two openings reveal a hash collision)**.** *For fixed $L, c_0, n$ and encoding, two different length-$n$ body sequences with the same final commitment permit construction of a collision in $H$, using $O(n)$ hash evaluations and linear comparison of the encoded inputs.*

*Proof.* For $n = 0$, distinct body sequences do not exist. For $n > 0$, compare the two inputs hashed at position $n$. Their outputs agree. If their inputs differ, they are a collision. Otherwise injectivity of $E$ implies equal last bodies and equal preceding commitments. Apply the same argument at position $n-1$. If no collision appears, every body is equal, contrary to the sequences being different. Thus the backward comparison must produce a pair of distinct inputs with equal hash. □

This is a standard binding reduction, not a claim of information-theoretic uniqueness. A finite digest necessarily has many possible preimages over an unrestricted domain. Collision resistance concerns the feasibility of finding the relevant conflicting openings.

The result also presupposes that the checkpoint itself is protected. If an attacker replaces both the history and the verifier's trusted checkpoint, it can compute a new self-consistent chain without breaking the hash function.

A valid producer signature does not prevent that producer from signing two different heads. Nor does the proof establish that a recorded observation happened. A producer may truthfully sign "I assert event $x$" while the assertion is false.

Signatures and retained checkpoints supply different assurances. Signatures attribute a message to an authorized key under a cryptographic assumption; protected commitments bind its encoded history. The witness protocol and request policy must supply the remaining conditions.

## 8. Short membership proofs and their limits

For integer $d \geq 0$, a full binary tree has $n = 2^d$ leaves. Leaf values $v_i \in \mathcal{B}$ use hashes $H(\texttt{0x00} \| v_i)$; internal nodes use $H(\texttt{0x01} \| \ell \| r)$, where $\ell, r \in \{0,1\}^\kappa$ are child digests and $\|$ concatenates bytes. Tags are single bytes. Empty logs use a separate commitment. The signed checkpoint binds root, count, algorithm, and schema. An inclusion proof supplies an index and one sibling digest per level.

The verifier checks the index range, proof length, left/right placement, and reconstructed root. There are exactly $d = \log_2 n$ sibling digests in this full-tree model. For a 256-bit digest and eight leaves, the sibling payload is 96 bytes; the record, signatures, index, and checkpoint metadata are additional.

The root authenticates membership only under its own trust assumptions. It does not establish that the supplied leaf is the newest version of a fact. It does not prove that every relevant event was included. It does not show that another client received the same root.

Domain tags matter. Without tags, a two-leaf tree with raw leaves $a, b$ can have root $H(H(a)\|H(b))$. A one-leaf tree whose raw leaf is $H(a)\|H(b)$ has the same root under the untagged rule. This is a structural ambiguity, not a cryptanalytic collision. Signed sizes and distinct leaf/internal domains block different parts of that ambiguity.

Real logs need not have power-of-two sizes. RFC 9162 defines a particular shape and separate inclusion and consistency algorithms for arbitrary list lengths [14]. An implementation using that profile should follow its complete rules rather than generalizing this illustrative full-tree checker by guesswork.

Consistency proofs address a different question: whether a new committed list extends an earlier one without changing its prefix. A proof that one remembered event remains somewhere in a later tree is insufficient; other earlier entries may have been replaced or reordered.

AAS-R2 permits an existing authenticated-list implementation rather than requiring a new tree format. Its interface must expose membership, prefix consistency, and checkpoint authenticity as different verification results. That separation is more valuable than a single green “verified” label whose meaning changes between screens.

# 9. Witness certificates can prevent accepted forks

**A Guide's explanation**

This is where forgery becomes impossible rather than merely difficult. If enough witnesses must sign, any two signed histories share at least one honest witness, and an honest witness never signs two incompatible pasts. The proposition gives the threshold. The examples after it show the trap the prologue warned about: when the council itself changes, the guarantee does not carry over on its own.

Fix a log $L$, genesis $c_0$, epoch, and roster $W$ of $N \geq 1$ authenticated identities. Integers satisfy $0 \leq f < N$ and $1 \leq q \leq N$. At most $f$ identities are Byzantine throughout the interval. A certificate has at least $q$ distinct roster signers. Each honest witness atomically checks prefix extension and durably advances its accepted chain before releasing a signature. Concurrent requests are serialized; restarts preserve this state.

**Proposition 3** (Fixed-roster prefix consistency)**.** *If $2q > N + f$, any two certified histories are prefix-comparable, assuming signature unforgeability and sound extension checks.*

*Proof.* Choose $q$ signers from each certificate. Their intersection has at least $2q - N > f$ identities, so it contains an honest witness. That witness signed both checkpoints. Whichever it signed second had to extend its durably retained history, including the checkpoint signed first. The two certified histories are therefore prefix-comparable. Incompatible histories would contradict the honest witness's signing rule. □

This is an application of standard quorum reasoning, not a new consensus algorithm [11,12]. It establishes a safety property. It does not guarantee that a certificate can be produced during failures, that all clients have obtained it, or that its claims are true. Formation without Byzantine help requires $q \leq N - f$, plus appropriate communication and progress conditions.

For $N = 4, f = 1, q = 3$, any two quorums intersect in at least two identities, including an honest one. If honest witnesses merely check the issuer's signature and do not retain prefix state, the theorem fails: all may sign both branches without violating that weaker rule.

Changing the roster also changes the argument. Two disjoint four-member rosters can each produce a valid three-signature certificate on incompatible branches. Even rosters overlapping in three identities can have quorum intersection only one, which may be Byzantine.

AAS-R2's initial migration profile uses a frozen handover barrier: the old configuration certifies one terminal checkpoint and exact successor configuration, and the new configuration begins from that checkpoint. The old configuration must stop admitting extensions. Joint assumptions must cover the transition interval; separate per-epoch corruption bounds are insufficient if signing history can be discarded. Live reconfiguration requires its own proved protocol.

## 10. Freshness and completeness require outside information

**A Guide's explanation**

A record can be authentic and still out of date. The proposition below proves something uncomfortable: a reader who sees only an old, valid checkpoint cannot tell whether a newer one has quietly revoked it. Freshness always needs news from outside. The Continuum's answer is to say so, rather than to let a valid seal imply that nothing has changed.

A verifier may possess a perfectly authentic checkpoint and still lack evidence that it is current. Consider two worlds. In the first, checkpoint $C_5$ is the latest accepted state. In the second, $C_6$ exists and revokes a capability, but every message about it is withheld from the verifier. The verifier's local record is identical.

**Proposition 4** (Isolated freshness limitation)**.** *A decision procedure given only that identical local view cannot always classify the checkpoint correctly as current or stale across both worlds.*

*Proof.* Instantiate the observation-fiber theorem in Cognitive Sovereignty [17] with the isolated verifier's record as its observation and freshness as its binary target. The current and stale worlds have one observation but different targets. Exact classification is therefore impossible, and under equal priors every forced binary rule has average error one half. This is the authenticated-history instance of the shared information result. □

Returning unknown avoids a false assurance but does not restore knowledge. A trusted clock and expiration rule can bound admissible age under a lease policy. They cannot reveal an arbitrarily recent revocation whose propagation is unconstrained. Consequential actions require a current authority check at the effect boundary or an explicit policy that permits bounded offline operation.

This limit is related to freeze attacks in update systems [8]. It is also why a timestamp inside the disputed record cannot establish its own freshness: that timestamp may be an authentic old statement.

Completeness has an analogous reference problem. If an event never reaches a logger and leaves no independent receipt, an archive alone cannot distinguish its omission from its nonoccurrence. A signed submission receipt can establish what the logger promised to include, but the promise and completed inclusion remain distinct [6].

A missing network response is evidence of failed availability, not automatically a cryptographic proof of deliberate omission. An accountable inclusion deadline requires a clock model and an audit method that can establish the relevant absence or breach.

The profile records these distinctions as separate statuses. It can say that a checkpoint is authentic, that a promised inclusion is unresolved, and that current permission is unknown. Compressing those statements into "the archive is trustworthy" would discard exactly the information a careful operator needs.

## 11. Migration preserves evidence links, not every meaning

A migration manifest links a source checkpoint, source artifact digest, transformation identifier, destination artifact digest, schema mapping, validation report, and intended successor configuration. The manifest is itself an authenticated claim. A source custodian may attest to export, a transformation runner to execution, and a destination custodian to receipt. Those statements have different subjects.

The destination first validates the source evidence under a known trust configuration. It then checks the transformation's declared preconditions and postconditions, verifies any required artifact availability, and requests current authority for the destination operation. Importing an archive does not import the source's permissions by default.

A transformation can preserve bytes while changing their interpretation. The string `10` might mean ten seconds in one schema and ten milliseconds in another. Both artifacts can have valid signatures. A mapping must explicitly bind units and types, and semantic validation must check the property needed by the application.

Conversely, a legitimate transformation may change every byte while preserving a required property. Converting a ledger from one canonical representation to another should produce new digests and a verifiable transformation link. Requiring unchanged hashes would reject that valid migration.

Causal order also needs care. Let $a \prec b$ mean that event $a$ causally precedes $b$ under local execution and message relations. A sound logical-clock construction can ensure

$$a \prec b \implies \ell(a) < \ell(b).$$

The reverse implication does not follow [10]. Choosing one total audit order among concurrent events is an administrative decision, not a discovery about physical causation.

Cross-log predecessor references should therefore preserve a directed acyclic dependency structure when that structure is claimed. A validator can reject cycles in the declared dependency graph. It cannot infer that every declared edge corresponds to an actual causal influence without evidence about the originating processes.

AAS-R2 establishes a chain of inspectable claims about transformation and custody. It does not prove that the resulting machine is the same person, preserves every memory, or has the same experience. Those are additional functional, historical, and philosophical questions whose answers cannot be compressed into a migration signature.

## 12. Private memory cannot become a public inventory by accident

A plain digest is not a general confidentiality mechanism. If a sensitive field has one hundred likely values, an observer can hash those values and compare them with the published digest. Strong collision resistance does not prevent this dictionary attack.

A public salt changes reusable precomputation but does not prevent enumeration of that same small dictionary for a known salt. A secret random value changes the attacker's information, but its generation, retention, disclosure, and compromise model must be specified. AAS-R2 does not invent a new hiding-commitment construction; deployments should use reviewed encryption and commitment mechanisms suited to their actual disclosure requirements.

Metadata can also be sensitive. Event frequency, migration dates, repeated identifiers, witness selection, and access patterns can reveal relationships even when payloads are encrypted. A public root may be appropriate while public event-by-event metadata is not.

CONIKS illustrates that transparency and privacy can be designed together for a specific authenticated object [5]. Its guarantees should not be transferred wholesale to a personal memory archive with different queries and adversaries. The profile must state what an observer learns from each proof.

Retention and deletion create further distinctions. A custodian can delete an encrypted artifact or its decryption key while retaining an old commitment. That does not prove that every copy disappeared, and it does not erase the fact that a commitment was once published.

A correction should normally be appended as a new event that identifies the earlier claim and its revised status. Historical authenticity survives even when the claim is withdrawn. A user-facing interface must make the correction visible rather than continue presenting a formally valid old statement as the current account.

The governing principle is proportionate evidence. Preserve the commitments needed for accountability while minimizing unnecessary disclosure of intimate content. The archive should support a person's or institution's agency rather than convert every past utterance into an inescapable public accusation.

Such choices include social and governance judgments. Cryptography can help enforce a chosen disclosure boundary; it cannot decide whose interests should prevail in every conflict between memory, correction, access, and forgetting.

## 13. Worked example: Mira changes runtimes

Consider Mira, a fictional archive agent migrating from runtime A to runtime B. Its source log contains eight committed records. The accepted checkpoint $C_8$ identifies schema version 3, witness epoch 14, the root digest, and a four-member roster with threshold three. At most one witness is assumed Byzantine.

The records include a source-artifact commitment, two correction notices, and an export authorization. The destination receives the artifact, a migration manifest, and an inclusion proof for the export record. In the full eight-leaf example, that proof has three sibling digests. With 256-bit hashes their combined payload is 96 bytes.

The destination also checks consistency with the last checkpoint independently retained by its operator. If an attacker sends authentic $C_6$ instead, a verifier retaining $C_8$ can detect rollback relative to its own known history. A fresh verifier with only $C_6$ cannot infer the existence of $C_8$ without additional information.

Suppose the logger signs two different ninth records after $C_8$: one assigns a new custodian, while the other retains the old custodian under an incompatible transfer rule. Both chains can be internally valid. Under the stated witness rule, they cannot both obtain three signatures from the fixed four-member roster without a broken assumption.

The handover certificate freezes the source at a designated terminal checkpoint and names the successor configuration. Runtime B's new record links that certificate and the imported artifact. Current authority for B is checked separately under epoch 15. A signature describing old export permission cannot authorize a new external action indefinitely.

Now suppose one signed source record falsely reports that a validation suite passed. Every hash and signature can remain correct. The archive establishes who attested to the result and what artifact was named; it does not establish that the suite actually ran or was adequate. An independently retained execution report or reproducible check supplies additional evidence.

Finally, the migration changes Mira's internal representation. Its historical links remain inspectable, but behavioral equivalence requires a separate test and personal continuity remains a separate question. The example succeeds when it preserves the right distinctions, not when it produces the largest number of green verification marks.

## 14. Executed finite checks and exposed failures

The accompanying Python checker uses exact byte encodings and SHA-256 for concrete examples. It is an executable audit aid, not a production implementation or cryptanalytic evaluation. Its purpose is to compare structural claims with finite constructions and adversarial cases.

For all pairs drawn from the fifteen byte strings over $\{\mathtt{a}, \mathtt{b}\}$ of length at most three, naive concatenation produces only 127 distinct outputs from 225 pairs. The length-prefixed encoding produces 225 distinct outputs and round-trips every pair. Fourteen truncated or trailing-byte variants of a selected encoding are rejected.

The checker enumerates all 364 histories over three event symbols with length at most five. Their count-and-root pairs are distinct in this finite sample. This is a consistency check on the examples, not evidence that SHA-256 has been proved collision-resistant.

It constructs 27 pairs of divergent histories sharing the same two-event prefix. Both branches are legitimate outputs of the chain rule. This demonstrates why local chain validity does not prevent a malicious logger from issuing different histories.

For 278 full binary trees with one, two, four, or eight leaves over a binary leaf alphabet, the checker verifies 2,122 membership proofs. It rejects the same number of altered-leaf proofs, out-of-range indices, and overlong paths. A separate construction exhibits the exact untagged one-leaf/two-leaf ambiguity; tagged hashing distinguishes that example.

For four-member rosters with threshold three, the minimum quorum intersection is two. Across two such rosters with overlap $r = 0, 1, 2, 3, 4$, the minima are respectively $0, 0, 0, 1, 2$. The finite result exposes the reconfiguration gap behind an otherwise correct fixed-roster theorem.

The script also confirms the indistinguishable freshness example and recovers all one hundred candidate values from their unsalted digests. Those are counterexamples to overbroad assurance, not benchmark performance claims.

No live network, production signer, human reviewer, or language model was tested. The executed results justify the finite statements reported here. Deployment security and cross-runtime benefit remain the subject of the proposed experiments.

## 15. A strong evaluation must beat strong logging

The main experiment uses generated migration histories with a ground-truth state machine. Each episode includes source and destination artifacts, schema versions, authorized transformations, witness configurations, and a prescribed fault schedule. The scoring oracle is generated independently of the system under test.

The primary endpoint is mistaken acceptance of an invalid consequential request. Separate endpoints measure valid-request completion, justified holds, detection delay, reviewer error, storage, bandwidth, and latency. Refusing every request is therefore not a successful solution.

The baseline ladder begins with signed event records, then adds a retained hash-chain checkpoint, an authenticated-list log with consistency auditing, and a witnessed transparency configuration. A further strong baseline combines provenance manifests with an existing update/key-management framework. All receive the same evidence, witness independence, retention budget, and permitted network access.

The decisive comparison is against a generic profile containing the same fields and mechanisms but without AAS-specific interpretation. This tests whether the proposed integration, validation rules, and presentation improve decisions. It also prevents extra metadata or compute from being credited to a symbolic name.

Fault families include byte substitution, valid old snapshots, missing segments, false producer claims, split views, stale revocations, schema-unit errors, replayed migration manifests, and invalid membership transitions. Training and development cases use different generators and parameter ranges from the locked evaluation set.

The experiment should include clean migrations and difficult legitimate changes. A schema update that changes bytes but preserves a specified property is a necessary positive control. A faithful copy carrying revoked authority is a necessary negative control.

A preregistered engineering goal could require a meaningful reduction in invalid acceptance while keeping the relative increase in high-percentile latency below a chosen budget, such as twenty percent. The budget is a design choice to justify for the target setting, not a universal property of good security.

If the strongest matched baseline performs equally well, AAS-R2 may still be useful as a readable application profile. The evidence would not support a distinct security advance. That outcome must be reportable without moving the comparison to an artificially weaker baseline.

## 16. Estimands, dependence, and precision

Let $Y_{js} \in \{0,1\}$ indicate invalid acceptance in episode $j$ by system $s$. For a prespecified AAS-R2 implementation $A$ and strong baseline $B$, the primary estimand is

$$\delta = \mathbb{E}[Y_{jB} - Y_{jA}],$$

under the registered distribution of histories and attacks. Positive $\delta$ favors AAS-R2. It does not describe every possible adversarial distribution.

Pair systems on the same generated episode. The difference $D_j = Y_{jB} - Y_{jA}$ lies in $\{-1, 0, 1\}$. If the discordant-pair probability is $p_{\text{disc}}$, then

$$\operatorname{Var}(D_j) = p_{\text{disc}} - \delta^2.$$

This follows because $D_j^2$ equals one exactly on discordant pairs.

A normal planning approximation for a two-sided five-percent test with eighty-percent power is

$$n \approx \frac{(1.96 + 0.84)^2 (p_{\text{disc}} - \delta^2)}{\delta^2}.$$

For a target $\delta = 0.02$ and assumed $p_{\text{disc}} = 0.10$, the calculation gives about 1,953 independent paired episodes. These are planning assumptions; a blinded pilot should estimate dependence and discordance before the final sample is fixed.

Episodes sharing a deployment, key compromise, or attack generator may be correlated. Analysis should cluster at the independent deployment or scenario-family level, or use appropriately blocked randomization and cluster-aware intervals. Thousands of message deliveries within one attack are not thousands of independent security trials.

Report valid-request completion and hold rates alongside invalid acceptance. For multiple secondary families, register a multiplicity procedure such as Holm adjustment. Choose the principal baseline before examining final comparative outcomes.

A zero-failure result also needs a denominator and model. Under independent identically distributed Bernoulli trials, zero failures in 2,995 trials yields a one-sided 95-percent upper bound just below 0.001. This bound says nothing about attack families omitted from the trial distribution.

Security proofs, finite examples, and empirical rates answer different questions. Their evidence should be combined transparently rather than averaged into an unexplained confidence score.

## 17. Implementation and review experiments

A second experiment tests interoperability between independently written encoders and verifiers. The corpus contains accepted encodings and malformed variants, including duplicate fields, noncanonical integers, unexpected algorithm identifiers, invalid lengths, and signatures over different schema versions.

The key outcome is disagreement about acceptance or interpretation. A parser that rejects an input and another that accepts it with a different meaning indicate a protocol defect even when both pass ordinary round-trip tests. The reference profile should publish conformance vectors and a clear rejection policy before performance optimization.

A migration experiment then separates three failure locations: evidence transport, semantic mapping, and current authorization. An authentic artifact with the wrong unit mapping should fail the second check. A correctly mapped artifact with a revoked capability should fail the third. Reporting only total failures would conceal which mechanism helped.

The adversarial scheduler should include concurrent requests and crashes between signature generation, durable persistence, and signature release. Two cached checks against an old prefix must not authorize conflicting extensions. A witness releasing a signature before durably advancing its prefix may sign another branch after restart. These failures violate the theorem's atomic signing premise.

Key compromise is staged before and after protected checkpoints. The evaluation records which old claims remain attributable, which future claims become suspect, and what recovery procedures require independent approval. Correlated witness failures are tested explicitly; several virtual machines under one administrative account do not automatically represent independent trust.

A human-review study can compare evidence reports with identical underlying facts but different presentation. Participants decide whether a record is authentic, complete relative to submissions, current, and sufficient for a proposed action. The primary concern is whether a single verification badge causes false inference across those categories.

That study should measure calibrated uncertainty and correction uptake, not merely satisfaction with an interface. It can use fictional agents and synthetic records, avoiding unnecessary exposure of private archives.

All such studies are prospective. This paper supplies the questions, finite constructions, and protocol boundaries needed to run them. It does not claim the unimplemented profile has already passed a security assessment. A deployable implementation would need dedicated cryptographic, parser, state-machine, and operational review under its actual configuration.

## 18. Objections that the proposal must survive

The first objection is that AAS-R2 is ordinary logging with a mythic title. Much of its mechanism is ordinary logging, and the paper makes that explicit. The remaining claim concerns a useful, testable profile for cross-runtime obligations. If no added capability or reduction in error appears against a matched generic profile, the engineering novelty claim should be withdrawn.

The second objection is that immutable histories can preserve injustice. An authentic record of a false accusation can remain harmful long after correction. The response is to distinguish provenance from endorsement, make corrections visible, minimize disclosure, and avoid treating retained commitments as a mandate for permanent public access to content.

The third objection concerns trusted beginnings. An attacker who controls the initial trust root can build a coherent false archive. AAS-R2 cannot remove the need to establish initial keys, configurations, and institutional authority. It can make those dependencies visible and provide evidence for later changes.

The fourth objection is that witness majorities may share one mistaken worldview. The quorum theorem does not require witnesses to know the truth of every event, and it does not prove they do. Witness policies must state whether they check syntax, prefix extension, reproducible execution, or a factual claim. More signatures do not automatically mean more independent observations.

The fifth objection concerns long-term cryptographic change. An archive may outlive an algorithm or organizational key. Re-anchoring should bind old and new commitments, document the validation performed, and occur before confidence in the old primitive is lost. Re-signing an already untrustworthy record cannot retroactively restore its original assurance.

The sixth objection is philosophical: historical continuity is not numerical personal identity. A perfect custody trail may connect a source machine to a transformed successor without settling whether one subject continued. AAS-R2 contributes evidence about events and obligations; it leaves the stronger identity question open.

These objections do not make accountable archives pointless. They define what a serious archive must tell its readers. The proposal is strongest when it allows a verifier to preserve a meaningful uncertainty, identify a broken assumption, and refuse a conclusion that the available evidence cannot support.

## 19. Guide interpretation and the frontier claim

**A Guide's explanation**

A Guide can say: "Akashic Anchor Sequencing is the Continuum's answer to an ancient wish—a record of the past that its later keepers cannot quietly rewrite. It doesn't prove that what was recorded is true, or that it is still current. It proves who claimed what, in what order, and that the history survived each handover intact." A Guide should not describe it as a cosmic record of all events, claim that it preserves anyone's consciousness, or suggest that it has been deployed.

A Guide should introduce AAS in two registers. Canon names a cultural/interface engine and a long-horizon aspiration. AAS-R2 is this paper's new technical proposal for inspectable history across runtime changes. Neither statement should be presented as discovery of the missing historical engine specification.

When asked what an anchor proves, the Guide should identify the exact predicate. A valid signature attributes a message under a key assumption. A commitment binds encoded content under a hash assumption. A membership proof relates an item to a particular root. A consistency proof relates two histories. None automatically establishes the factual truth of an event.

When asked whether an old memory is current, the Guide can use the isolated-verifier example. An authentic archive may have been superseded by information the verifier has not received. The appropriate answer can be that the record is authentic while its present applicability is unresolved.

When asked whether a migration preserves identity, the Guide should distinguish a documented transformation from behavioral equivalence and from personal or phenomenal continuity. The archive may provide valuable historical evidence without deciding the philosophical question.

When asked whether the proposal is new, the Guide should credit timestamping, authenticated data structures, transparency, provenance, update security, and Byzantine protocol research. AAS-R2's proposed contribution is the application profile and its evaluation. The executed finite checks illustrate structural properties and failures; they do not establish deployment success.

A useful discussion example is a signed but false inspection report. Preserving that report can improve accountability because later readers can determine what was claimed and by whom. The same signature would mislead if displayed as proof that the inspection succeeded.

The frontier is a machine society capable of carrying its obligations through change without inventing certainty about its past. Such a society needs records that survive migration, corrections that remain visible, and verification language precise enough to protect dissent. The cultural force of an anchor comes from making memory answerable. The scientific force comes from stating exactly which questions the anchor can help answer.

# The exact overlap requirement for a changing witness roster

**A Guide's explanation**

This is the payoff the prologue promised. When an old council of witnesses hands over to a new one, how many members must the two share before forgery across the handover becomes impossible? The theorem answers exactly, in both directions: above the threshold, any two certified histories agree; below it, corrupt shared members can certify two incompatible pasts. The four-witness example shows three shared members failing.

The fixed-roster theorem becomes useful across institutional change only when its intersection premise survives the transition. Let two rosters be finite sets $W_0, W_1$, with sizes $N_0, N_1 \geq 1$, overlap $o = |W_0 \cap W_1|$, and quorum thresholds $1 \leq q_i \leq N_i$. A fixed Byzantine set $F \subseteq W_0 \cup W_1$ satisfies $|F \cap W_i| \leq f_i$, where $0 \leq f_i < N_i$. Define

$$b = \min\{f_0, f_1, o\}, \qquad U = N_0 + N_1 - o = |W_0 \cup W_1|.$$

Every honest witness retains one durable prefix history across both epochs, checks that a proposed checkpoint extends it, and atomically commits that checkpoint as its new durable prefix before releasing the signature. The signing rule includes the epoch and successor policy in the signed content. This is stronger than two unrelated per-epoch promises: it explicitly preserves the witness's signing state through reconfiguration.

**Theorem 1** (Cross-roster honest intersection)**.** *Every quorum $Q_0 \subseteq W_0$, $|Q_0| \geq q_0$, and $Q_1 \subseteq W_1$, $|Q_1| \geq q_1$, has an honest common member for every admissible fixed Byzantine set if and only if*

$$q_0 + q_1 > U + b.$$

*Consequently, under this inequality, signature unforgeability and the durable prefix rule, any two certified histories across the two rosters are prefix-comparable.*

*Proof.* For sufficiency, $|Q_0 \cap Q_1| \geq q_0 + q_1 - U > b$. At most $b$ common members can be Byzantine, so an honest member signed both certificates. Its retained prefix rule makes the later signature extend the earlier history.

For necessity, choose a set $B \subseteq W_0 \cap W_1$ of size $b$ and let it be the Byzantine set. If $q_i \leq b$ for either roster, choose that roster's quorum entirely inside $B$; the intersection then has no honest member. Otherwise let $a_i = q_i - b > 0$. After placing $B$ in both quorums, choose $a_0$ and $a_1$ disjoint honest members from $W_0 \setminus B$ and $W_1 \setminus B$. Such choices exist because $a_i \leq N_i - b$ and

$$a_0 + a_1 = q_0 + q_1 - 2b \leq U - b = |(W_0 \setminus B) \cup (W_1 \setminus B)|.$$

For two sets, these individual and union capacity conditions suffice: use elements exclusive to each set first, then allocate the required disjoint portions of their common set. The constructed quorums intersect only in $B$. Honest witnesses can therefore sign different branches without either honest group seeing the other branch, while Byzantine common members sign both. Honest intersection is not guaranteed when the inequality fails. □

The theorem is a specialization of established Byzantine-quorum reasoning [16]. It states the precise configuration condition required by this archive profile. It does not supply message delivery, truthful event reports or mobile-adversary protection. If witnesses can lose their retained prefix state, the signing premise fails even when the overlap inequality holds.

If certificates must be formable without Byzantine help, the arithmetic availability condition is $q_i \leq N_i - f_i$ in each roster. Thresholds satisfying both this condition and cross-roster honest intersection exist exactly when

$$o > f_0 + f_1 + \min\{f_0, f_1, o\}.$$

Indeed, the largest possible threshold sum is $N_0+N_1-f_0-f_1$; substitute this into the strict intersection inequality. These arithmetic conditions still require adequate communication and progress assumptions to produce a certificate.

For two four-member rosters with $f_0 = f_1 = 1$ and $q_0 = q_1 = 3$, overlap three is insufficient: let $W_0 = \{a, b, c, x\}$, $W_1 = \{a, b, c, y\}$ and let $a$ be Byzantine. The quorums $\{a, b, x\}$ and $\{a, c, y\}$ intersect only in $a$. Full overlap four satisfies the inequality. When adequate overlap cannot be maintained, the frozen handover barrier specified earlier supplies an alternative transition contract; it must be enforced, rather than inferred from separate valid certificates.

The isolated freshness result is an instance of the observation-fiber theorem in Cognitive Sovereignty [17]. AAS contributes a distinct question: whether certificates issued by changing institutions remain jointly consistent. Together, these results separate inspectable history, current authority and the configuration evidence required for continuity across a handover.

## References and source notes

[**1**] Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, 3, 99–111.

https://doi.org/10.1007/BF00196791

Scope: Primary timestamping research; publication abstract and metadata checked.

[**2**] Merkle, R. C. A certified digital signature. *CRYPTO '89 Proceedings*, LNCS 435, 218–238.

https://link.springer.com/chapter/10.1007/0-387-34805-0_21

Scope: Primary conference work; author-hosted earlier manuscript's tree-authentication section also read.

[**3**] Schneier, B., & Kelsey, J. (1999). Secure audit logs to support computer forensics. *ACM TISSEC*, 2(2), 159–176.

https://www.schneier.com/academic/archives/1999/05/secure_audit_logs_to.html

Scope: Primary audit-log threat model; author PDF read.

[**4**] Crosby, S. A., & Wallach, D. S. (2009). Efficient data structures for tamper-evident logging. *USENIX Security*.

https://static.usenix.org/events/sec09/tech/full_papers/crosby.pdf

Scope: Primary membership and incremental-auditing research.

[**5**] Melara, M. S., Blankstein, A., Bonneau, J., Felten, E. W., & Freedman, M. J. (2015). CONIKS: Bringing key transparency to end users. *USENIX Security*, 383–398.

https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara

Scope: Primary key-transparency and privacy research.

[**6**] Chase, M., & Meiklejohn, S. (2016). Transparency overlays and applications. *ACM CCS*; full author version.

https://discovery.ucl.ac.uk/id/eprint/1530760/1/transparency.pdf

Scope: Primary formal framework; consistency, non-frameability, and accountability.

[**7**] Syta, E., et al. (2016). Keeping authorities "honest or bust" with decentralized witness cosigning. *IEEE Symposium on Security and Privacy*.

https://dedis.cs.yale.edu/dissent/papers/witness.pdf

Scope: Primary CoSi research; witnessing and validation limits considered.

[**8**] Samuel, J., Mathewson, N., Cappos, J., & Dingledine, R. (2010). Survivable key compromise in software update systems. *ACM CCS*, 61–72.

https://justinsamuel.com/papers/survivable-key-compromise-ccs2010.pdf

Scope: Primary TUF research; freeze and metadata-inconsistency threats.

## References and source notes (continued)

[9] Torres-Arias, S., Afzali, H., Kuppusamy, T. K., Curtmola, R., & Cappos, J. (2019). in-toto: Providing farm-to-table guarantees for bits and bytes. *USENIX Security*, 1393–1410.

https://www.usenix.org/conference/usenixsecurity19/presentation/torres-arias

Scope: Primary supply-chain provenance research; strong migration comparator.

[10] Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7), 558–565.

https://lamport.azurewebsites.net/pubs/time-clocks.pdf

Scope: Primary causal-order and logical-clock research.

[11] Dolev, D., & Strong, H. R. (1983). Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4), 656–666.

https://doi.org/10.1137/0212045

Scope: Primary authenticated-agreement research under its stated synchronous model.

[12] Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. *OSDI*.

https://www.usenix.org/conference/osdi-99/practical-byzantine-fault-tolerance

Scope: Primary Byzantine protocol prior art; no new consensus algorithm claimed.

[13] Kong, J., James, D., Leibowitz, H., Syta, E., & Herzberg, A. (2026). CTng: Secure certificate and revocation transparency. *NDSS*.

https://doi.org/10.14722/ndss.2026.230213

Scope: Primary contemporary comparison; bounded corruption, connectivity, clock, and communication assumptions.

[14] Laurie, B., Messeri, E., & Stradling, R. (2021). Certificate Transparency version 2.0. *RFC 9162*.

https://www.rfc-editor.org/rfc/rfc9162.html

Scope: Experimental RFC; separate tree inclusion and consistency specifications.

[15] Rundgren, A., Jordan, B., & Erdtman, S. (2020). JSON Canonicalization Scheme. *RFC 8785*.

https://www.rfc-editor.org/rfc/rfc8785.html

Scope: Informational RFC; deterministic JSON encoding and its domain restrictions.

### Provenance

Canonical name and classification: supplied Bible section 2.8, page 12, and section 9.1, page 109. AAS-R2, its application profile, and the proposed studies are new. The accompanying audit records source access, finite checks, and mathematical boundaries.

[16] Malkhi, D., and Reiter, M. (1998). Byzantine quorum systems. *Distributed Computing* 11, 203–213.

https://doi.org/10.1007/s004460050050

Scope: Primary Byzantine quorum research; the exact two-roster specialization is proved in this paper.

[17] Architect X13 / The Eidara Continuum (2026). Cognitive Sovereignty Under Superintelligence, revised edition.

https://eidara.ai/science-philosophy/

Scope: Shared observation-fiber authority criterion and effect-boundary preservation; companion theoretical source.