# ALCOA+ by Public Consensus

*Externally Verifiable Audit Trails via State-Governed Anchoring*

TJ Jung / HMIC LLC / Version 1.0 / August 2026

https://hmic.dev/proof/whitepapers/alcoa-plus-by-public-consensus/

Reference architecture and open-source demonstration. This paper does not claim that ALCOA RubberStamp, by itself, makes a system compliant with 21 CFR Part 11, EU GMP Annex 11, or another regulated requirement.

# Contents

## Executive Summary

In the CDMO industry, commercial personnel are often deliberately excluded from quality audits. That firewall protects sponsor confidentiality, preserves audit independence, and limits access to regulated information on a need-to-know basis.

I worked inside that separation for eleven years.

I do not see audit findings, batch records, deviation narratives, sponsor quality documents, or internal quality responses.

I do help schedule audits and connect sponsors with my Quality team. I can see that an audit process occurred through scheduling metadata, follow-up communication, and downstream operational effects, without seeing the protected audit content.

The payload remains private. The witness layer is visible.

That distinction is the starting point of this paper.

HMIC proposes applying the same discipline to GMP-relevant records: separate the protected payload from an externally verifiable witness layer. The payload remains inside the validated, access-controlled system. The witness layer records cryptographic evidence that a declared state existed no later than an external attestation. When the commitment also binds a fresh, independently verifiable prior state, the two bounds define a commitment interval.

This paper describes ALCOA+ by Public Consensus, a practical pattern for externally verifiable audit evidence using state-governed anchoring.

The method does not claim to prove that a manufacturing event was scientifically correct, that an operator made the right decision, or that a process step was properly executed. Existing GMP controls remain necessary.

The narrower claim is stronger:

A covered record that is later presented for verification will fail to match its preserved commitment if it was altered, replaced, or silently corrected after that commitment was created.

The system does not prevent bad data. It makes silent post-commitment change detectable when the record and proof are verified.

The basic pattern is simple:

1. Hash the record or record set.
2. Optionally bind that commitment to a fresh, independently verifiable prior state when a lower time bound is required.
3. Anchor the resulting commitment to a public-consensus network or other independently inspectable witness.
4. Preserve the proof path from individual record to external anchor.
5. Treat corrections as new witnessed events, not replacements of prior state.

The result is a supplemental evidence layer where verification can be performed independently, underlying GMP data can remain private, records can be checked without administrative access to the originating system, and corrections can retain a separately witnessed chronology. When the workflow binds each commitment to a fresh prior state and later external inclusion, the interval between those bounds becomes analyzable data.

The most important consequence is not merely tamper evidence. In the full two-bound protocol, it is the creation of a new compliance signal: externally bounded audit intervals.

How long did an error remain undetected? What decisions were made while the wrong state was active? Which shift, system, equipment train, or process area repeatedly produces delayed corrections? Who consistently discovers hidden record-state problems before the organization realizes they exist?

Current audit trails often preserve what changed. State-governed anchoring preserves when declared state became externally witnessable, when it changed, and how long the organization lived between those states.

That interval is not noise. That interval is data.

## 1. A Gap in Timing

The current state of GMP data integrity reflects technological timing, not a failure of character or regulatory will.

When 21 CFR Part 11 was issued in 1997, the industry faced a practical witness problem. To prove that an electronic record existed at a certain time, the recordkeeping system generally had to trust an internal authority: a server clock, a database, a validated application, an administrator, an institutional process, or a person signing within that process.

In that environment, internal audit trails were the practical ceiling of deployable electronic record integrity -- not a compromise born of laziness.

The industry did not choose internal witnesses because they were perfect. It chose them because they were available, governable, inspectable, and compatible with the systems of the time.

That choice was rational.

But the world changed.

Cryptographic primitives such as Merkle trees and linked timestamping had already established the mathematical basis for efficient bulk verification and tamper-evident ordering. Public-consensus networks later made it practical to place commitments into a globally replicated history that no single record owner administers.

The primitives this method depends on are not new. What has been missing is not the technology but the translation: a recordkeeping pattern expressed in the language of ALCOA+, audit trails, corrections, and inspection readiness. This paper closes a distance. It does not assign a fault.

The ingredients for stronger externally verifiable record integrity have been sitting in the public domain for decades.

What has been missing is not the mathematics. What has been missing is the GMP translation layer.

Pharma needs a recordkeeping pattern that speaks its own language: ALCOA+, audit trails, corrections, deviations, batch release, sponsor confidentiality, inspection readiness, and quality oversight -- not a generic blockchain story.

This paper proposes that translation -- as a natural maturation of recordkeeping, not an accusation against current practice.

## 2. Purpose and Scope

This paper proposes a supplemental evidence layer for GMP-relevant electronic records.

The goal is not to replace validated systems, electronic batch records, laboratory information management systems, quality management systems, manufacturing execution systems, document management systems, or existing audit trails.

The goal is to add an independently verifiable witness surface to records and record sets that already exist inside controlled systems.

The protocol described here is intended to support records such as:

- GMP batch records
- deviation closures
- CAPA milestones
- stability data packages
- analytical reports
- release documentation
- audit-response packages
- controlled procedural records
- other records where integrity, timing, and correction history matter

The protocol is not a substitute for:

- GMP execution
- scientific correctness
- operator training
- instrument calibration
- validated system controls
- electronic signature controls
- access control
- quality review
- deviation investigation
- product disposition
- regulatory inspection readiness

The external witness can prove that a commitment existed no later than its attestation. When a fresh prior-state binding is also present, it can define a bounded commitment interval. It does not prove that the declared state was true.

That distinction is critical.

A false value can be anchored. An incomplete investigation can be anchored. A premature conclusion can be anchored. A bad decision can be anchored.

The value of the protocol is that once a state is declared and witnessed, it cannot quietly become something else.

The system narrows ambiguity. A record either matches its prior commitment or it does not. A correction either exists in the subsequent witnessed history or it does not. A gap between original state and corrected state either occurred or it did

not.

The protocol does not create truth. It makes declared state accountable to time.

## 3. Regulatory Context: ALCOA+, Metadata, and Audit Trails

GMP data integrity is traditionally described through ALCOA and ALCOA+ principles. Records should be attributable, legible, contemporaneous, original, and accurate. Expanded formulations also emphasize completeness, consistency, endurance, and availability.

In practice, these properties are enforced through a combination of validated computerized systems, procedural controls, access restrictions, audit trails, electronic signatures, training, quality review, deviation and CAPA processes, backup and retention policies, and periodic inspection.

These controls remain necessary.

The problem is not that current systems lack controls. The problem is that many controls are verified within the same institutional boundary that creates, administers, and maintains the records.

A system records its own timestamp. A user signs within that system. An audit trail exists inside that system. An administrator may have privileged access to that system. A reviewer inspects records exported from that system. A regulator or sponsor may later review what the organization presents.

This is not meaningless. It is the foundation of modern GMP operations.

But it is still an internal witness model. The witness, the payload, and the administrative control surface are often inside the same trust boundary.

HMIC's position is that certain ALCOA+ properties can be strengthened by adding an external witness layer. The payload remains internal. The witness becomes independently inspectable.

This paper does not argue that internal audit trails should be replaced. It argues that high-value GMP records should be able to answer a simple external question:

Given this record, this manifest, and this proof, can an independent verifier confirm that this exact declared state matches an externally witnessed commitment--and, when both time bounds are present, identify the interval in which that commitment was constructed?

If yes, the audit trail has gained an evidentiary property it did not previously have.

## 4. Core Thesis: Public Witness, Private Payload

The core thesis is witness/payload separation.

The payload is the protected record content: batch data, analytical values, deviations, investigations, quality decisions, release determinations, sponsor-confidential information, proprietary process details.

The witness is a cryptographic commitment to that payload and its declared state: record hash, manifest hash, Merkle root, prior external state reference, inclusion proof, timestamp proof, correction relationship, and the proof path from record to anchor.

The payload stays private. The witness is externally verifiable.

This is not a new legal or operational idea. Sealed envelopes, notary records, registered mail, laboratory notebooks, patent filings, and third-party attestations all express the same general principle: a party can prove that something existed at a point in time without revealing the full content publicly.

The difference here is the witness.

Traditional mechanisms rely on an institution: a notary, a postal service, a patent office, a document custodian, a validated system, or a regulated company. Integrity of the timestamp correlates with integrity of the institution.

A public-consensus network provides a different type of witness. It is not trusted because it is benevolent. It is useful because it is independently inspectable, globally replicated, economically defended, and adversarially maintained. Verification does not require trust. It assumes distrust. It requires only that the network exists and its history can be inspected.

The witness does not need to know the payload. The witness only needs to preserve the commitment.

That is enough.

## 5. What ALCOA RubberStamp Is

ALCOA RubberStamp is HMIC's free, open-source reference implementation of the lowest-friction form of this method. Version 1.0 demonstrates local file hashing, deterministic manifest generation, versioned Merkle commitments, optional prior-state binding, OpenTimestamps submission, proof preservation, and local verification.

Its scope is deliberate. It demonstrates the method; it does not replace regulated source systems, serve as the system of record, or hold GMP payload in the public domain.

The tool exists to make the method easy to inspect, easy to test, and easy to understand.

A scientist, student, small lab, independent researcher, quality professional, technical reviewer, or software vendor should be able to run the primitive without buying an enterprise platform. That is deliberate.

Provenance should not be exotic infrastructure. A person should be able to prove that a record, result, draft, report, package, or dataset existed in a specific form without needing to reveal the record publicly and without needing permission from a centralized service provider.

ALCOA RubberStamp is that starting point.

For GMP use, however, the reference tool is only the primitive. A regulated deployment still requires appropriate governance: intended-use definition, validation where applicable, SOPs, access control, retention, review, exception handling, and quality oversight.

The distinction is simple. The reference tool proves the method is practical. The quality system governs how the method is used.

## 6. Protocol Overview: State-Governed Anchoring

For any record or record set D, the full two-bound protocol defines the following pattern:

1. Compute a deterministic cryptographic commitment to the record set, including normalized record identity and content digests.
2. Bind the commitment to a fresh, independently verifiable prior state, such as a recently observed block hash, when a lower bound is required.

3 Submit the bound commitment to an independently inspectable witness layer.

4 Preserve and verify a proof of inclusion in a later external state.

5 Preserve the proof path linking the record set, manifest, commitment, prior-state reference, and external proof.

When both bounds are present, this creates a bounded window. Assuming the prior-state value was fresh and not predictable before its publication, the bound commitment could not have been constructed before that state was known. The later inclusion proof establishes that the submitted commitment existed no later than the attesting state.

In plain language: the record commitment existed after B_n and before B_n+k.

This does not prove the manufacturing event occurred in that interval. It proves that a specific declared record state was committed in that interval.

For the manufacturing event itself to inherit that evidentiary value, the governed workflow must require that hashing and commitment occur at the appropriate point of performance, review, signoff, correction, or release.

That workflow requirement is not optional. Without governed timing, the external proof establishes only that the commitment existed no later than its attestation. Without a fresh prior-state binding, it does not establish a lower bound. With both bounds embedded in a governed workflow, the protocol becomes an external witness to the declared state of that workflow.

## 7. Proof Semantics: What the Protocol Proves

The protocol proves five things.

### 7.1 Existence of a Declared State

If a verifier has the record, manifest, proof path, and external anchor, the verifier can confirm that the record state corresponds to the anchored commitment. The proof does not reveal the record to the public. The verifier must possess the record or authorized copy to check the hash.

### 7.2 Integrity of a Record Copy

If the record has changed, its hash changes. A mismatch between the current record and prior commitment reveals alteration, replacement, corruption, or use of the wrong version. This is not interpretation. It is deterministic.

### 7.3 Timing Semantics of Commitment

A later external attestation establishes an upper bound: the submitted commitment existed no later than the attesting state. A lower bound exists only when the commitment itself includes a fresh, independently verifiable prior state. Together those two facts create a bounded commitment interval. The interval may be narrow or wide depending on capture timing, submission delay, batching policy, network conditions, and proof maturity.

### 7.4 Correction Chronology

A correction is not a rewrite of the past. A correction is a new declared state with its own commitment, proof path, and external attestation. When the commitment also includes a fresh prior-state binding, it has a two-sided commitment interval. The original state remains part of the record history. The correction adds to the history.

### 7.5 Independent Verifiability

A verifier does not need privileged access to the originating system to verify a provided record/proof pair. The verifier needs the record or true copy, the manifest, the Merkle path where applicable, the external timestamp or inclusion proof, and access to the relevant witness substrate. The verifier can then independently determine whether the record matches the prior commitment.

## 8. What the Protocol Does Not Prove

Scientific and compliance credibility require stating the limits clearly.

The protocol does not prove that the original data were correct. It does not prove that the operator performed the step correctly. It does not prove that an instrument was calibrated. It does not prove that the correct SOP was followed. It does not prove that the signer was the biological person associated with the account unless the deployment includes appropriate identity and access controls. It does not prove that every relevant record was included unless the governed process requires inclusion and detects omission. It does not prove that local metadata were complete unless the manifest schema requires those metadata.

It does not eliminate the need for audit-trail review, deviation management, CAPA, validation, backup, retention, or inspection. It does not make a noncompliant process compliant.

It makes the history of declared state harder to falsify quietly.

That is the proper claim. And it is enough.

## 9. Architecture: From Individual Records to External Anchor

Anchoring every individual GMP record directly to a public network is unnecessary and operationally noisy. The method scales through hierarchical commitment.

At the lowest level, individual records or events are hashed. Those hashes are grouped into local manifests. Manifest hashes are composed into Merkle trees. Merkle roots are rolled upward through organizational layers. Only the top-level root needs to be externally anchored.

A representative hierarchy might be:

1. Event layer. Individual readings, observations, decisions, signatures, or record versions.
2. Activity layer. A controlled step, test, deviation milestone, or batch-record section.
3. Dual-control gate. Performer and verifier signoff, where required by procedure. This is GMP's existing two-signature rule; the protocol makes the signature cryptographic and the gate a precondition for upward propagation.
4. Department layer. Department-level Merkle root across governed activities.
5. Site layer. Site-level Merkle root across departments.
6. External anchor layer. Public timestamp proof or other independently inspectable witness.

The hierarchy is end-user-defined. A manufacturing site may organize by batch, suite, campaign, process area, or product. A laboratory may organize by sample set, method, analyst, instrument, or stability protocol. A quality unit may organize by deviation, CAPA, audit response, or release package.

Internal rollups may stay internal. Department-to-site composition can happen on end-user infrastructure while only the top-level commitment is submitted externally. Retained local commitments can detect later changes to covered materials, but they do not independently establish timing until they are bound to an external witness. The evidentiary meaning therefore depends on what was committed, what proof was retained, and when the external attestation occurred.

The protocol does not impose the business schema. It imposes the evidentiary invariant: every record included in the proof tree must be traceable from its local digest to the externally witnessed root.

## 10. Operational Principle: Corrections Add History

The operational rule is simple: hash new state when new state exists.

A record is created. Hash it. A decision is made. Hash it. A verifier signs. Hash it. A correction occurs. Hash it. A deviation is closed. Hash it. A batch is released. Hash it.

The original state remains where it was. A mistake discovered later does not erase the original commitment. The correction occupies its own position in time.

This is aligned with good documentation practice. Errors are not supposed to disappear. Corrections are supposed to be attributable, legible, dated, explained, and traceable to the original entry.

State-governed anchoring adds an external witness to that chronology. The result is not merely a better lock on the record. It is a better map of the process.

## 11. Audit Interval Intelligence

Most timestamping systems treat time as a label. This protocol treats time as a measurement surface.

In the full protocol, every anchored event has two relevant temporal bounds: the prior external state referenced by the commitment, and the later external state in which inclusion or timestamp proof is established. Every correction has a relationship to an earlier declared state. The interval between original state and corrected state is operationally meaningful.

That interval can answer questions current audit trails often make difficult to analyze at scale. How long did the incorrect state persist? Which downstream decisions occurred during that interval? Which batches, samples, lots, or investigations were potentially affected? Was the correction triggered by routine review, deviation investigation, inspection, customer inquiry, instrument failure, or operator intuition? Are certain shifts, systems, rooms, products, or process stages associated with longer correction latency? Which people repeatedly identify and resolve hidden blockages?

This is Audit Interval Intelligence.

The correction is not merely a compliance event. It is a signal.

A correction found after one hour and a correction found after three months are not the same event class. They may involve the same field, the same system, and the same procedural requirement, but their risk meaning is different. The interval is the difference.

A mature system should not only ask whether a record was corrected. It should ask how long the organization operated under the prior state.

# 12. Example Views Enabled by the Data Structure

The protocol naturally supports several views without requiring artificial reconstruction after the fact.

## 12.1 Linear Correction Timeline

Plot each correction event on a time axis. Patterns become visible: time-of-day clustering, shift-handover spikes, post-maintenance corrections, post-deviation cascades, end-of-campaign cleanup, product-specific correction signatures, reviewer-specific unlock patterns.

A conventional audit trail may contain the same events. When the full two-bound protocol is used in a governed workflow, externally bounded correction timing helps the organization distinguish ordinary record maintenance from delayed discovery.

## 12.2 Blockage Gantt

For each corrected state, draw a horizontal bar from the original declared state to the corrected state. The bar represents the duration during which downstream activity may have relied on the prior state.

A one-hour error and a three-month error have different risk profiles. A blockage discovered before review and a blockage discovered after batch release have different risk profiles. A blockage discovered internally and a blockage discovered during inspection have different risk profiles. This view translates abstract data-integrity history into operational risk.

## 12.3 Cluster and Unlock View

Corrections rarely occur in isolation. They cluster around causes: one instrument problem, one misunderstood procedure, one training gap, one material lot, one reviewer, one room, one handoff, one software configuration, one recurring documentation habit.

By grouping corrections around unlock events, the system can show when the organization finally discovered a hidden pattern. The unlock event matters. Sometimes it is a deviation. Sometimes it is a sponsor question. Sometimes it is an inspection. Sometimes it is a new operator who notices what everyone else had normalized.

## 12.4 Unlock Attribution

The Merkle hierarchy can preserve local metadata for operator, reviewer, department, shift, instrument, suite, product, and process stage without exposing that metadata publicly.

Internally, the organization can query the unlock events. Who was on shift when each blockage cleared? Group by name. Sort by aggregate blockage time eliminated.

A name comes up forty-three times. That person's job description does not include "diagnostic intuition." Their compensation reflects their job description. They are the most economically valuable person in the building, and nobody knew until the audit trail told them.

This is the audit trail equivalent of finally being able to pay someone what they are worth. And to do it before they leave.

This is not just compliance. It is operational intelligence.

# 13. ALCOA+ Reinterpreted Through External Witnessing

The protocol does not replace ALCOA+. It externalizes the properties that benefit from external witness.

Attributable. Existing control surface: user accounts, roles, electronic signatures, access controls. External-witness contribution: actor-linked digests and signature references can be included in the local manifest and proof path.

Legible. Existing control surface: human-readable or system-rendered record. External-witness contribution: none; legibility remains a source-system and record-format requirement.

Contemporaneous. Existing control surface: system timestamps, procedural expectations, review. External-witness contribution: external lower and upper bounds reduce sole dependence on local system clocks.

Original. Existing control surface: controlled source record or true copy. External-witness contribution: a digest can verify whether a presented copy matches the committed state; it does not, by itself, establish that the committed state was the regulatory original.

Accurate. Existing control surface: review, verification, deviation, CAPA, scientific controls. External-witness contribution: does not prove accuracy; preserves the declared state and correction chronology.

Complete. Existing control surface: required fields, procedural checks, batch-record review. External-witness contribution: manifest completeness can be governed and reviewed as part of the record package.

Consistent. Existing control surface: validated workflows, controlled formats, audit trails. External-witness contribution: deterministic hashing and verification support consistent integrity checking.

Enduring. Existing control surface: retention, backup, archive controls. External-witness contribution: external witness remains independently inspectable if proof materials are retained. The hash-and-anchor pattern has been documented in peer-reviewed literature since Gipp et al. (2015) and deployed in production systems including OpenTimestamps.

Available. Existing control surface: record retrieval, inspection support, system access. External-witness contribution: a verifier can check a provided record/proof pair without privileged source-system access.

The key point is restraint. The protocol does not claim more than it proves. It strengthens integrity evidence, timing evidence, correction chronology, and independent verifiability. It does not replace the quality system. It gives the quality system a stronger witness.

## 14. Threat Model

A practical threat model clarifies where the protocol helps and where traditional controls remain necessary.

After-the-fact alteration of a covered record. Addressed: yes. Hash mismatch reveals that the record no longer matches the committed state.

Backdating after the prior-state reference. Addressed: partially. Lower-bound strength depends on binding the commitment to known external state.

Local system clock manipulation. Addressed: partially. External bounds reduce reliance on local timestamps but do not remove the need for system controls.

Silent correction or replacement. Addressed: yes, if governed. A correction must appear as a new state; absence of a correction becomes meaningful.

False original entry. Addressed: no. Incorrect data can be honestly anchored.

Omission of a required record. Addressed: partially. Requires manifest completeness checks and procedural controls.

Credential sharing. Addressed: no. Requires identity, access management, and electronic-signature controls.

Privileged administrator tampering. Addressed: partially. External proof detects changes to covered records but does not prevent local misuse.

Metadata leakage. Addressed: partially. Requires batching, cadence, and manifest-design controls.

Loss of proof files. Addressed: partially. Requires retention and backup of manifests, Merkle paths, and timestamp proofs.

Public-network disruption. Addressed: partially. Can be mitigated by delayed submission, multiple substrates, or internal anchoring.

Hash algorithm obsolescence. Addressed: partially. Requires an algorithm policy and re-anchoring strategy over long retention periods.

This is the correct posture: the protocol is a control layer, not a complete control system.

## 15. Privacy and Confidentiality Model

Only commitments need to be public. The record itself remains private.

In the simplest case, the public witness sees only a digest or Merkle root. That root is computationally useless without the underlying record set and proof path. An external observer should not be able to reconstruct batch data, deviation content, analytical results, sponsor identity, product identity, or proprietary process information from the public anchor.

However, hash-only anchoring does not automatically eliminate all confidentiality risk. Deployment design must consider metadata leakage. Potential leakage surfaces include anchoring cadence, timing of submissions, volume of roots, proof size, manifest naming, department-level grouping, correction frequency, public association between a company address and operational activity, and repeated anchoring during sensitive campaigns or investigations.

These risks are manageable, but they are real. Mitigations may include batching multiple records into a single root, using neutral manifest identifiers, separating public anchor data from internal operational labels, using OpenTimestamps or another aggregation layer, anchoring at a cadence that does not reveal sensitive operations, anchoring to multiple substrates where appropriate, and preserving detailed metadata internally while exposing only the top-level commitment externally.

The principle remains: public verification, private payload, controlled metadata.

## 16. Substrate: Public Consensus as Witness

The protocol is substrate-agnostic. A valid external witness substrate must provide independent inspectability, a durable historical record, public or otherwise neutral verification, resistance to unilateral alteration, practical long-term accessibility, and clear proof semantics.

The Bitcoin timechain is a useful default substrate because its history is widely replicated and independently inspectable. A timestamp proof anchored to Bitcoin can be checked without relying on HMIC, the record owner, a sponsor, or the originating database administrator. Long-term verifiability still depends on retaining the record, manifest, proof material, algorithm identifiers, and software or migration procedures needed to interpret them.

The protocol does not require the end user to hold cryptocurrency. It does not require a wallet. It does not require placing GMP data on-chain. It does not require a direct on-chain transaction for every record.

A root can be anchored through an aggregation layer such as OpenTimestamps, which batches many commitments and anchors them periodically. The end user can retain a timestamp proof without directly interacting with Bitcoin

infrastructure.

The Merkle root is just a commitment. The chain is just a witness. The GMP data remain inside the controlled environment.

# 17. Implementation Modes

There are several levels of implementation maturity.

## 17.1 Mode 1: Supplemental File Timestamping

The simplest implementation hashes selected files or directories, builds a manifest, and obtains an external timestamp proof. This mode is useful for controlled document packages, release packages, audit-response exports, deviation closure packages, stability data snapshots, technical reports, and pre-submission evidence bundles.

This is not a replacement for the source system. With a verified external timestamp, it proves that a specific exported package existed in that form no later than the attestation. With an authentic fresh prior-state binding as well, it supports a two-sided commitment interval. This is the mode demonstrated by ALCOA RubberStamp.

## 17.2 Mode 2: Batch or Event Manifest Anchoring

A more mature implementation creates structured manifests for record sets. The manifest may include the file digest, record identifier, system source, version, actor or role, signature state, event type, local timestamp, prior external state, prior internal state, correction relationship, and Merkle path. The manifest itself is then hashed and included in a Merkle tree. This supports richer audit reconstruction while preserving payload privacy.

## 17.3 Mode 3: GMP Workflow Integration

In a full GMP deployment, anchoring is integrated into the governed workflow: hash at step completion, at verifier signoff, at deviation approval, at CAPA milestone, at batch release, at record correction, at audit-response submission.

This mode requires validation, SOPs, role definitions, exception handling, training, retention policy, and quality oversight. It also produces the strongest evidentiary value, because the act of hashing becomes part of the controlled process.

## 17.4 Mode 4: Hardware-Linked or Instrument-Linked Witnessing

The strongest implementations connect declared record state to physical or system-level sources: instrument-state capture, hardware-bound signing, device identity, equipment-cycle evidence, environmental-monitoring records, manufacturing equipment events, operator proximity or access events, validated interface events between systems.

These implementations go beyond standalone file stamping. They link the witness layer to the physical or governed system that generated the record. This is where the method becomes structurally powerful: the record is no longer merely stamped after export. The evidence layer becomes part of the manufacturing, laboratory, or quality workflow itself.

## 17.5 Mode 5: Site-Level Audit Intelligence

At the highest level, the organization treats witnessed state transitions as an analytics substrate. The site can measure correction latency, blockage duration, review lag, deviation aging, batch-release evidence timing, department-level record volatility, repeated unlock sources, and systemic hidden-state patterns.

This is where the protocol becomes more than timestamping. It becomes a way to see the quality system's temporal structure.

## 18. Validation and Governance Considerations

If the tool is used only as an informal supplemental proof for non-GMP exploratory purposes, the implementation burden may be low. If the tool is used in a regulated workflow, it must be governed according to intended use.

A serious deployment should define intended use, records in scope, systems in scope, manifest schema, hash algorithm, timestamp substrate, anchoring cadence, required confirmation depth or proof maturity, user roles, access controls, electronic signature relationship, SOPs, training, exception handling, correction handling, audit-trail review procedure, backup and retention, disaster recovery, proof verification procedure, periodic review, algorithm migration strategy, and open-source dependency assessment.

A site should also define what happens when anchoring fails. Possible failure states include a local hash generated but external proof pending, external network congestion, timestamp calendar unavailable, manifest incomplete, proof file missing, source record hash mismatch, correction relationship unresolved, and an operator attempting to bypass the governed anchoring step.

These should not be improvised during inspection. They should be proceduralized.

The point of external witness is not to avoid quality governance. The point is to give governance a stronger evidentiary surface.

## 19. Inspector or Sponsor Verification Package

A practical verification package may include:

1. The record or authorized true copy.
2. The local manifest identifying the record digest.
3. The complete versioned manifest, or a compact inclusion path when the implementation produces one.
4. The external timestamp proof or inclusion proof.
5. The prior-state reference used at commitment.
6. The verification command or procedure.
7. The verification output.
8. Any correction linkage.
9. The SOP governing record inclusion and proof retention.
10. The validation package, where applicable.

An implementation that emits compact inclusion proofs can let a reviewer verify one authorized record without receiving every other manifest entry. ALCOA RubberStamp v1.0 instead uses the complete local manifest to recompute the commitment; it does not yet emit compact per-record Merkle paths.

This creates a useful inspection posture: show the payload only to authorized parties; show the witness to anyone permitted to verify integrity. That is the practical value of witness/payload separation.

## 20. Why This Is Not Already Standard

Section 1 named the cause: timing, not negligence. This section gives that claim its dates.

Every primitive this method depends on has been public, documented, and freely available for years.

1979/1987 -- Ralph Merkle develops the tree-based authentication construction; the commonly cited conference publication appears in the CRYPTO '87 record.

1991 -- Haber and Stornetta publish How to Time-Stamp a Digital Document, establishing cryptographic timestamping as a chained, verifiable construct.

1997 -- Adam Back circulates Hashcash, applying computational proof-of-work to email anti-abuse.

2008 -- Nakamoto's Bitcoin whitepaper integrates all three primitives into a single deployed system.

2013 -- Araoz and Ordano release Proof of Existence, the first open implementation of arbitrary-document anchoring to a public chain.

2015 -- Gipp, Meuschke, and Gernandt describe decentralized document timestamping using Bitcoin. OpenTimestamps subsequently provides an open aggregation and verification protocol for Bitcoin timestamp proofs.

The asymmetry is the argument. Part 11 was issued in the same broad technical period as early proof-of-work proposals and only six years after Haber and Stornetta's timestamping paper. It established durable requirements for controls, signatures, record protection, and audit trails without requiring an external cryptographic witness. That was a rational technology-neutral choice, and it remains the governing baseline. External commitments now offer a supplemental evidentiary surface that did not then have a mature deployment path.

The later FDA data-integrity guidance, EU GMP Annex 11, and PIC/S data-integrity guidance continue to emphasize governance, validated controls, audit trails, review, security, retention, and risk management. They do not require the public-consensus witness described here. This paper proposes that witness as a supplement, not as an interpretation of what those authorities already require.

This is the gap. Not a failure to assign, a distance to cross. The mathematics has been waiting since 1991. The deployable substrate has been waiting since 2008. What has been missing is not the technology but the translation: a recordkeeping pattern that speaks the language of ALCOA+, audit trails, corrections, deviations, batch release, sponsor confidentiality, inspection readiness, and quality oversight, and a reference implementation that lets an operator begin without asking anyone's permission.

The public-consensus witness has been available. The GMP mapping has not been obvious enough. That is the gap this paper addresses.

## 21. Why HMIC Is Releasing ALCOA RubberStamp as Free Software

HMIC's mission is to make verifiable provenance easy, available, and inexpensive enough that scientists, researchers, small labs, students, independent builders, and individuals can use it without permission or budget approval.

Basic provenance should not be exotic infrastructure. A scientist should be able to timestamp a result. A student should be able to prove when a draft existed. An independent researcher should be able to preserve priority. A small lab should be able to verify records without buying an enterprise platform. An individual should be able to create evidence that survives institutional memory.

ALCOA RubberStamp is released as free and open-source reference software for that reason. The free tool demonstrates the primitive in the open. It allows the method to be inspected, tested, criticized, improved, and used.

It also draws a clear boundary. The reference layer is for access. Regulated implementation still requires judgment, governance, and validation.

The open software gives people the flashlight. The quality system decides where to shine it.

## 22. Practical Use Cases

The initial implementation is intentionally simple. A user selects a directory or record package. The tool hashes the files, builds a manifest, computes a root. The root is externally timestamped. The proof is retained. A future verifier can confirm whether the package matches the original commitment.

This supports practical use cases such as the following.

### 22.1 Batch Release Package Snapshot

Before or at release, a site can stamp a controlled export of the release package. This does not replace the release decision. After independent verification, it creates an external evidence point showing that the package existed in that form no later than the attestation; a fresh prior-state binding can add a lower bound.

### 22.2 Deviation Closure Package

A deviation closure package can be stamped at approval. If later revised, the revised package is stamped as a new state. The gap between original closure and later revision becomes explicit.

### 22.3 Stability Data Snapshot

A stability dataset or report can be stamped when reviewed, approved, or transferred. A future reviewer can confirm that the presented dataset matches the historical commitment.

### 22.4 Audit Response Evidence

A sponsor or regulator may ask for records during or after an audit. A site can stamp the response package when assembled. The timestamp proof does not prove the response is sufficient, but after independent verification it can establish that the package existed in that form no later than the external attestation.

### 22.5 Scientific Priority and Research Provenance

Outside GMP, scientists and independent researchers can stamp results, drafts, datasets, or observations. This supports priority, integrity, and later reproducibility without requiring publication of the underlying material.

## 23. Limitations of the Reference Implementation

The free reference implementation is intentionally minimal. Version 1.0 commits to normalized relative paths, file sizes, and file-content digests; records the commitment scheme and tool version; can bind a user-supplied fresh prior-state value; submits the resulting commitment through OpenTimestamps; and can later recompute the local commitment. OpenTimestamps verification is a separate step performed by the OpenTimestamps client.

Without a supplied and independently checked fresh prior-state value, the reference tool produces an upper-bound timestamp proof, not a two-sided time window. The operator remains responsible for deciding whether a prior-state source is appropriate and for retaining the evidence needed to verify it.

It does not automatically provide Part 11 compliance, validated workflow enforcement, electronic signatures, user authentication, role-based access control, instrument integration, hardware-linked signing, automatic batch-record capture, deviation workflow integration, CAPA workflow integration, long-term archive governance, audit-trail review, or inspection narrative.

Those are implementation and governance layers. This limitation is not a defect. It is the reason the reference implementation can remain simple, inspectable, and free.

The reference tool proves the primitive. The regulated environment governs the use.

## 24. Future Directions

The ALCOA+ by Public Consensus pattern can evolve in several directions: structured GMP manifest schemas, validated verification packages, dual-signature manifest gates, hardware-bound signing modules, instrument-linked state capture, batch-record system integration, LIMS integration, MES integration, QMS integration, deviation and CAPA chronology, automated correction interval analytics, site-level audit interval dashboards, multi-substrate anchoring, long-term hash algorithm migration, sponsor-verifiable release packages, and inspector-facing verification workflows.

The first step is not to build all of that at once. The first step is to make the primitive real, inspectable, and easy to run. That is what ALCOA RubberStamp is for.

## 25. Conclusion: Accountability to Time

The purpose of an external witness is not to achieve perfection. It is to make declared state accountable to time.

A GMP record does not become trustworthy because error is impossible. Error is always possible.

An operator can misread a value. An instrument can drift. A reviewer can miss a pattern. A procedure can be ambiguous. A system can preserve a timestamp while failing to preserve context. A correction can be justified. A deviation can be real. A batch can be acceptable despite an error.

This view accepts that. Human error is part of the process. The question is whether the record can quietly become something else after the fact.

A record may be right. A record may be wrong. A record may be corrected. A record may be superseded. A record may turn out not to matter.

But once witnessed, a later presentation cannot quietly become something it was not and still pass verification against the preserved commitment.

ALCOA RubberStamp is a free reference implementation of that principle: public witness, private payload, independent verification.

The future of audit integrity is not trustless manufacturing. Manufacturing will always require trust, training, judgment, and quality oversight.

The future is narrower and more practical: when doubt appears, the record should not ask for belief. It should carry proof.

## References


1 U.S. Food and Drug Administration. Data Integrity and Compliance With Drug CGMP: Questions and Answers Guidance for Industry. December 2018. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/data-integrity-and-compliance-drug-cgmp-questions-and-answers

2 21 CFR Part 11. Electronic Records; Electronic Signatures. Electronic Code of Federal Regulations. https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11

3 European Commission. EudraLex Volume 4, Annex 11: Computerised Systems, revision January 2011. https://health.ec.europa.eu/medicinal-products/eudralex/eudralex-volume-4_en

4 PIC/S. Good Practices for Data Management and Integrity in Regulated GMP/GDP Environments, PI 041-1, 1 July 2021. https://picscheme.org/docview/4234

5 Merkle, R. C. A Digital Signature Based on a Conventional Encryption Function. CRYPTO, 1987.

6 Haber, S. and Stornetta, W. S. How to Time-Stamp a Digital Document. Journal of Cryptology, 1991.

7 Back, A. Hashcash: A Denial of Service Counter-Measure. 1997 / 2002.

8 Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.

9 Araoz, M. and Ordano, E. Proof of Existence. 2013.

10 Gipp, B., Meuschke, N., and Gernandt, A. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. iConference, 2015.

11 OpenTimestamps. OpenTimestamps documentation and protocol resources. https://opentimestamps.org/

## Contact

ALCOA RubberStamp is released as free and open-source reference software. For questions, review, technical discussion, or GMP-oriented evaluation, contact:

TJ Jung Founder, HMIC LLC hmic.dev tj@hmic.dev

HMIC builds infrastructure for verifiable, audit-grade recordkeeping in regulated industries.

Canonical URL: https://hmic.dev/proof/whitepapers/alcoa-plus-by-public-consensus/