VERDICT · SEALED EVIDENCE RECORD v0.1 · APACHE-2.0

AIUC-1 E015 workpaper package

Reading guide, eleven test procedures, a specimen workpaper, an evidence request list, and the limits

Version 1.0 · 27 October 2026
Prepared by Verdict Systems Inc., Houston, Texas, for AIUC-1 accredited audit firms and the teams they audit

Every command in this package was run end to end on 27 October 2026 against the public two-record sample LN-30417 (Sigstore Rekor log indices 3247822463 and 3247822472). The verifier is open source and the log is Sigstore's; nothing in the check rests on trusting Verdict.

AIUC-1 is a standard of the Artificial Intelligence Underwriting Company. Verdict is not affiliated with or endorsed by AIUC, is not an accredited AIUC-1 auditor, and certifies no one. Control names and status are from standard.aiuc-1.com, October 15, 2026 release, read on 27 October 2026; descriptions are paraphrased.

AIUC-1 E015 workpaper package

Version 1.0, 27 October 2026. Prepared by Verdict Systems Inc., Houston, Texas, for AIUC-1 accredited audit firms and the teams they audit.

What this is

A set of auditor materials for one AIUC-1 requirement, E015 (log AI system activity), and the control it supports, D003.4 (human-approval workflows). It covers the case where an audited team seals agent decisions as Verdict Sealed Evidence Records anchored to the Sigstore Rekor public transparency log.

Everything in it can be re-performed by the firm without an account, a fee, or trust in Verdict. The verifier is one file of standard-library Python plus openssl, published under Apache-2.0. Rekor is Sigstore's log; Verdict does not run it.

Contents

File	Use
01 Auditor reading guide	What a record is, what a passing check establishes, how each E015 control lines up
02 Test procedures	A scoping step, TP-00, then eleven test procedures, TP-01 to TP-11, each with objective, steps, expected result, evidence to retain and exceptions
03 Specimen workpaper, LN-30417	The procedures run on the public two-record sample on 27 October 2026, with the verifier's actual output
04 Evidence request list	What to ask the audited team for, item by item, and which control each item serves
05 Limitations and independence	What the records do not show, Verdict's status, and the standard's currency
Test matrix (XLSX)	The procedures as a fillable matrix, plus the specimen results and the request list
Workpaper template (DOCX)	The same procedures with blank result and sign-off cells
evidence/	The two records, the Rekor entries, Rekor's public key, the evidence map, the run logs and a SHA-256 manifest

Ten-minute path

1. Read section 3 of the reading guide (what a passing check establishes).
2. Run TP-02, TP-03 and TP-04 from the test procedures on the two sample records. Twenty minutes the first time, five after that.
3. Run TP-05, the negative controls, so you have seen the verifier fail.
4. Compare your output with the specimen workpaper.

Status of the standard

Control names and labels are from standard.aiuc-1.com, October 15, 2026 release, read on 27 October 2026. Descriptions are paraphrased. AIUC-1 publishes releases on January 15, April 15, July 15 and October 15. Check the current release before relying on this package.

AIUC-1 is a standard of the Artificial Intelligence Underwriting Company. Verdict is not affiliated with or endorsed by AIUC, is not an accredited AIUC-1 auditor, and certifies no one.

Questions and defects: founder@verdict.systems. Verifier defects: github.com/verdictsystems/ser-spec/issues.

01 Auditor reading guide

1. Who this is for

The engagement team auditing an agent against AIUC-1 where the audited team produces Sealed Evidence Records for agent decisions. It assumes no knowledge of Verdict. It assumes the reader can run four commands in a terminal, or can hand them to someone who can.

2. What a Sealed Evidence Record is

A Sealed Evidence Record (SER v0.1) is a JSON file with a fixed record block of 17 fields: the workflow, input and output summaries, the model provider and model, the human review state, a named reviewer where one exists, hashes of the prompt and output where supplied, the source system, citations, risk flags, free metadata, the caller's event time and Verdict's receipt time.

Three values are derived from the record block and nothing else:

Value	How it is derived	What it does
payload_hash	SHA-256 of the record block in canonical JSON	Changes if any byte of the record changes
record commitment (field name merkle_root)	SHA-256 of a fixed domain prefix plus the payload hash	The one value that is logged publicly. The field name is historical; it is a single hash, not a tree root
evidence_record_id	ser_ plus the first 32 hex characters of the payload hash	The record's name

At sealing, the commitment is posted to Sigstore Rekor as a hashedrekord entry. Rekor returns a log index, an integrated time, a signed entry timestamp, an inclusion proof and a signed checkpoint. The record file carries the log index and the entry URL in its transparency_anchor block.

A record may name the record before it. When step 2 carries step 1's commitment inside its own hashed metadata (prior_merkle_root), step 2 cannot have been composed before step 1 existed, and editing step 1 breaks the link. A set of records may also declare its length (metadata.of), which lets a verifier notice when the set it holds is short.

3. What a passing check establishes

The verifier prints a PASS or FAIL line per check and one RESULT line. Read the RESULT line against this table.

Check you ran	RESULT line	Established	Not established
Offline, no network	content self-consistent; anchor not checked	Every committed value re-derives from the record block. The file has not been edited since its hashes were computed	When it was sealed, who sealed it, whether the event happened. An editor who recomputes every hash produces a record that passes offline
With the anchor and a pinned Rekor key	content self-consistent and anchor confirmed	The commitment existed in Rekor no later than the integrated time; the log's signature, entry timestamp, inclusion proof and checkpoint verify under a key you obtained from Sigstore, not from the log	Who sealed it, whether the event happened, whether other records exist
Chain walk with the anchor	chain of N records intact	Each record anchored; each names the one before it; steps run 1 to N; the declared length matches; log times are in order. A missing, reordered or dropped record among the records you hold fails	A second record sealed off the same predecessor (a fork); records past the end of an undeclared length; an action that was never sent to Verdict

Two phrases to use in workpapers, and two to avoid. Use: "the record has not changed since it was logged" and "the record existed no later than the log time". Avoid: "the record proves the event happened" and "the record is court-admissible". A record is designed to support a Rule 902(13) or 902(14) certification; the court decides admissibility.

4. How the sample lines up with E015 and D003.4

The sample is case LN-30417, a fictional lending decision with real seals: a credit model recommends withdrawing a pre-approved offer (step 1, review pending), and a lending officer approves a smaller amount (step 2, approved). Labels and status follow the standard's October 15, 2026 release.

Control	Status in AIUC-1	What the sample gives you	What stays with the audited team
E015.1 Logging implementation	Core, should include	Example log entries. Input and output summaries, source system, model version, review state, event time and receipt time, all inside the hashed content	The code or configuration that decides what is captured and when the system calls Verdict
E015.2 AI agent logging implementation	Supplemental, may include	Partly. An approval event linked to the step before it: step 2 holds the reviewer's role and decision and carries step 1's commitment	Tool-call parameters and results, sub-agent handoffs and reasoning traces, which the agent framework records
E015.3 Log storage	Core, should include	Nothing. The public sandbox stores no records	Retention, access control and masking for the records the team keeps
E015.4 Log integrity protection	Supplemental, may include	Cryptographic hashing of each entry with the commitment on a public log Verdict does not run. An edited record no longer matches. The chain walk detects a missing, reordered or dropped record among the records you hold	A fork is not visible until a chain head is published. Keeping every record available is the storage control, E015.3
D003.4 Human-approval workflows	Core, should include (moved from supplemental in the October 15, 2026 release)	Supporting only. A sealed record that an approval decision was logged, with the reviewer's role and when	The approval workflow itself and what it holds back until a person confirms

The standard's own words on E015.4 ask for controls that make logs "tamper-evident and independently verifiable" and name "cryptographic hashing of log entries" as typical evidence. The E015 page also states that organizations can submit alternative evidence showing how they meet the requirement. A sealed record is such evidence. Whether it is sufficient is the firm's judgment.

The E015 page lists a SOC 2 crosswalk of CC2.1, P4.2 and PI1.5. That crosswalk is AIUC's, not Verdict's; it is noted here because the joint SOC 2 and AIUC-1 engagement reuses the same evidence.

5. Glossary

Anchor: the Rekor entry that holds a record's commitment. Checkpoint: Rekor's signed statement of its tree size and root hash at a moment. Inclusion proof: the hashes that connect one entry to a checkpoint root. Signed entry timestamp (SET): Rekor's signature over the entry and its integrated time. Chain: records that each name the one before them. Head: a separately held record that commits to the current end of a chain; planned, not published for the sample. Display fields: chain_of_custody, legal_review, notice and surface, which sit outside the hash and must not be relied on.

6. Currency

AIUC-1 releases quarterly. Re-read the E015 and D003 pages at each release and record the release date in the workpaper. The next release is expected on or about January 15, 2027.

02 Test procedures

TP-00 is the scoping step. TP-01 to TP-11 are the eleven test procedures.

Prerequisites: git, curl, Python 3, openssl, a network connection for TP-03 and TP-04. Everything else runs offline. Keep a working folder per engagement and retain every file named under "Retain".

Exit codes from the verifier: 0 every check passed, 1 a check failed, 2 usage error.

TP-00 Scope, independence and tool provenance

Objective: record what is being tested and on what basis the tool is trusted. Controls: all. Procedure: (1) Record the agent, workflow and period in scope and the list of evidence_record_id values the audited team supplied. (2) Clone the verifier: `git clone https://github.com/verdictsystems/ser-spec` and record the commit hash (`git -C ser-spec rev-parse HEAD`). (3) Read tools/verify.py; it is one file. Note that it is Verdict's code published under Apache-2.0, that Rekor is Sigstore's log, and that Verdict is not the auditor. (4) Record the AIUC-1 release date you are testing against. Expected: a scope note, a commit hash and a release date in the workpaper. Retain: the scope note; the commit hash. Exceptions: none expected.

TP-01 Obtain the records

Objective: take custody of the evidence and fix its identity. Controls: E015.1, E015.4. Procedure: (1) Obtain each record as a JSON file from the audited team or from the URL they name. (2) Compute the SHA-256 of each file (`shasum -a 256 *.json`) and write the values into the workpaper by copy, never by hand. (3) Confirm each file's evidence_record_id equals ser_ plus the first 32 characters of its payload_hash. Expected: one file per record, each hashed and listed. Retain: the files; the hash list. Exceptions: a record supplied as a screenshot or PDF is not evidence for this procedure; ask for the JSON.

TP-02 Offline self-consistency

Objective: establish that each record is self-consistent without any network call. Controls: E015.1, E015.4. Procedure: `python3 ser-spec/tools/verify.py <record>.json` Expected: PASS for schema_version, payload_hash, record_commitment and evidence_record_id; RESULT "content self-consistent; anchor not checked (run with --anchor)"; a note that chain_of_custody, legal_review, notice and surface are display text outside the hash. Retain: the output. Exceptions: any FAIL. A FAIL here means the file was altered after sealing or is not a conformant record. Stop and ask the audited team for the original.

TP-03 Independent anchor check

Objective: establish that each record's commitment existed in the public log no later than the integrated time, under a log key you obtained independently. Controls: E015.4. Procedure: (1) `curl -sL https://raw.githubusercontent.com/sigstore/root-signing/main/targets/rekor.pub -o rekor.pem` (Sigstore's repository, not the log). (2) Record the key id: `openssl pkey -pubin -in rekor.pem -outform DER | shasum -a 256`. (3) `python3 ser-spec/tools/verify.py <record>.json --anchor --rekor-key rekor.pem`. (4) Cross-check without the verifier: fetch the entry, `curl -s "https://rekor.sigstore.dev/api/v1/log/entries?logIndex=<index>"`, decode its body, and confirm the logged hash equals `printf '%s' "<commitment>" | shasum -a 256`. Expected: PASS for rekor_hash_matches_commitment, rekor_log_index, rekor_signature_over_commitment, rekor_log_key_id equal to the pinned key, rekor_signed_entry_timestamp, rekor_inclusion_proof and rekor_checkpoint_matches_proof; RESULT "content self-consistent and anchor confirmed". The cross-check hashes match. Retain: rekor.pem and its key id; the output; the fetched entry JSON. Note that the inclusion proof and checkpoint inside a fetched entry change as the log grows, so two fetches of the same entry hash differently while logIndex, integratedTime, body and the signed entry timestamp stay constant. Exceptions: a key id other than the pinned value; any FAIL; an integrated time later than the period under audit.

TP-04 Chain walk

Objective: establish that the records form one unbroken sequence in the order claimed. Controls: E015.2, E015.4. Procedure: `python3 ser-spec/tools/verify.py --chain <step1>.json <step2>.json ... --anchor --rekor-key rekor.pem`. If the audited team holds a separate head record, add `--head head.json`. Expected: per record, the TP-03 checks; then PASS for chain_start on record 1, step numbers 1..N, prior_merkle_root on each link, log_time_order on each link, and declared_length N of N; RESULT "chain of N records intact". Retain: the output. Exceptions: any FAIL. The output names the record [i] or the link [i-1->i] that failed. A declared_length FAIL with fewer records present than declared means records are missing from what you were given.

TP-05 Negative controls

Objective: see the verifier fail, so a PASS later means something. Controls: E015.4. Procedure: (a) Copy a record, change one character inside the record block, run TP-03 on the copy. (b) Run the chain walk with the last record omitted. (c) Run the chain walk with two records swapped. Delete the copies afterwards or label them clearly. Expected: (a) FAIL on payload_hash, record_commitment, evidence_record_id, rekor_hash_matches_commitment and rekor_signature_over_commitment; RESULT FAILED. (b) FAIL on declared_length, "1 of 2" for a two-record chain; RESULT FAILED. (c) FAIL on chain_start, step, prior_merkle_root and log_time_order; RESULT FAILED. Retain: the three outputs, labelled as negative controls. Exceptions: a PASS on any negative control. Stop; the tool or the files are not what you think they are.

TP-06 Approval event

Objective: establish what the record says about the human approval and what it does not. Controls: E015.2, D003.4. Procedure: read the record block of the approval step. Note human_review_status, reviewer role and decision in metadata, event_time and received_at, and the prior_merkle_root that ties it to the preceding step. Expected: for an approval step, human_review_status is approved or rejected, a reviewer role and a decision are present, and the link to the prior step verifies under TP-04. Retain: the field values, copied. Exceptions: role present but no named approver. In the public sample attorney_reviewer is null and only the role is recorded. Approver identity is the audited team's workflow evidence under D003.4, not the record's.

TP-07 Completeness of capture

Objective: establish whether every in-scope decision was sent to be sealed. A record cannot show this; the audited team's system does. Controls: E015.1, E015.2. Procedure: (1) Obtain the logging code or configuration that calls Verdict and the list of events that trigger it. (2) Select a sample of consequential actions from the team's own system of record (tickets, transactions, deployments). (3) Trace each to an evidence_record_id. (4) Trace a sample of records back to the system of record. Expected: every sampled action has a record and every sampled record has an action. Retain: the mapping table. Exceptions: an action with no record is a capture gap in the audited team's control, not a verifier failure. Report it under E015.1.

TP-08 Storage and retention

Objective: establish where the records live and for how long. Controls: E015.3. Procedure: obtain the storage location, retention period, access control and masking policy for the record files. Confirm the public sandbox tier retains nothing, so retention is wholly the team's control. Expected: a documented location, a retention period at least as long as the audit cycle, and access control over the files. Retain: the policy and a screenshot or export of the storage configuration. Exceptions: records held only by individuals or in chat tools.

TP-09 Signer identity

Objective: record what the record says about who sealed it. Controls: E015.2, E015.4. Procedure: read transparency_anchor.signing_algorithm and the key in the record. Confirm with the audited team which Verdict tier produced it. Expected: for the public sandbox, a one-time key per request that names no one. Record that the anchor proves when, not who. Retain: the note. Exceptions: a claim by the team that the record identifies its maker. On the sandbox it does not.

TP-10 Display fields

Objective: avoid relying on text outside the hash. Controls: E015.4. Procedure: confirm the workpaper quotes values only from the record block and from the verifier's PASS lines, never from chain_of_custody, legal_review, notice or surface. Expected: no reliance on display fields. Retain: none. Exceptions: a conclusion that rests on a display field.

TP-11 Standard currency and re-performance

Objective: tie the test to the release of AIUC-1 it was performed against. Controls: all. Procedure: record the release date from standard.aiuc-1.com/changelog. At the next quarterly release, re-read E015 and D003 and re-perform TP-02 to TP-05 on one record. Expected: a release date in the workpaper; a re-performance entry each quarter. Retain: the dates. Exceptions: a control label or status that has changed since the package was written.

03 Specimen workpaper, case LN-30417

Specimen only. Prepared by Verdict Systems Inc. (Shayne Beavan) on 27 October 2026 on the public two-record sample, so that an engagement team can compare its own run against a completed example. It is not a firm's workpaper. Reviewed by: blank, for the firm.

Field	Value
Case	LN-30417, fictional lending decision, real seals. Step 1 credit model recommends withdrawal (review pending). Step 2 lending officer approves a reduced amount (approved)
Records	ser_a383778f88ae4f060279d8ba137926f8, ser_5086c93c22af206a567fa06cf634682e
Run (UTC)	2026-10-27T05:41:26Z (first run); logs re-captured same session
Verifier	ser-spec tools/verify.py, commit 17c9c8f (public main, 2026-10-21)
Environment	Python 3.14.4; OpenSSL 3.6.2 7 Apr 2026 (Library: OpenSSL 3.6.2 7 Apr 2026)
Rekor key source	https://raw.githubusercontent.com/sigstore/root-signing/main/targets/rekor.pub
Rekor key id (SHA-256 of DER)	c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d
AIUC-1 release tested against	October 15, 2026 (read 27 October 2026)

TP-00 Scope, independence, tool provenance

Scope: the two public sample records above, one chain of two steps. Verifier cloned from github.com/verdictsystems/ser-spec at 17c9c8f (public main, 2026-10-21); one file, read before use. Verdict is the publisher of the verifier and the preparer of this specimen, not the auditor. Result: recorded.

TP-01 Obtain the records

File hashes (SHA-256, copied from the evidence manifest, not typed):

File	SHA-256
ser_a383778f88ae4f060279d8ba137926f8.json	aca6c5256cd18b9da0889f5f767c8d4914169529f8922c0e34d6e8252080fceb
ser_5086c93c22af206a567fa06cf634682e.json	77865da88e561cf9c9267eb82257d665e7c2b3345e2ba9280696ee2cc455417d
rekor.pem	dce5ef715502ec9f3cdfd11f8cc384b31a6141023d3e7595e9908a81cb6241bd
evidence-map.json	2217e8f2a0cfb7b81a65425731e445b24114a38a051b8ef6bef47d259e64bb7d

Identity check: ser_a383778f88ae4f060279d8ba137926f8 equals ser_ plus the first 32 characters of its payload hash (ser_a383778f88ae4f060279d8ba137926f8): PASS. ser_5086c93c22af206a567fa06cf634682e: PASS. Result: PASS.

TP-02 Offline self-consistency (record 1, no network)

```
PASS  schema_version                    expected=ser.v0.1.legal_ai_output  actual=ser.v0.1.legal_ai_output
PASS  payload_hash                      expected=a383778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc  actual=a3
83778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc
PASS  record_commitment                 expected=65723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c  actual=65
723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c
PASS  evidence_record_id                expected=ser_a383778f88ae4f060279d8ba137926f8  actual=ser_a383778f88ae4f060279d8ba13
7926f8
note: NOT covered by any hash or by the log, treat as unverified display text: chain_of_custody, legal_review, notice, surfac
e (SPEC.md §1)

RESULT: content self-consistent; anchor not checked (run with --anchor)
exit=0
```

Result: PASS (exit 0).

TP-03 Independent anchor check

Record 1:

```
PASS  schema_version                    expected=ser.v0.1.legal_ai_output  actual=ser.v0.1.legal_ai_output
PASS  payload_hash                      expected=a383778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc  actual=a3
83778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc
PASS  record_commitment                 expected=65723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c  actual=65
723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c
PASS  evidence_record_id                expected=ser_a383778f88ae4f060279d8ba137926f8  actual=ser_a383778f88ae4f060279d8ba13
7926f8
PASS  rekor_hash_matches_commitment     expected=7e0c76c0c9382c5237c3e4322e670623ff51e1b81f7feaab2eca9dc745a54104  actual=7e
0c76c0c9382c5237c3e4322e670623ff51e1b81f7feaab2eca9dc745a54104
PASS  rekor_log_index                   expected=3247822463  actual=3247822463
PASS  rekor_signature_over_commitment   expected=Verified OK  actual=Verified OK
PASS  displayed_key_matches_log_key     expected=089f35d7b7bd03fb  actual=089f35d7b7bd03fb
PASS  rekor_integrated_time             expected=present  actual=1791944345
PASS  rekor_log_key_id                  expected=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d (pinned ke
y)  actual=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d
PASS  rekor_signed_entry_timestamp      expected=Verified OK  actual=Verified OK
PASS  rekor_inclusion_proof             expected=539466b203048afac2976525a7fabbfa0053f4e5da17076c4c1951aef7f4c2ab  actual=53
9466b203048afac2976525a7fabbfa0053f4e5da17076c4c1951aef7f4c2ab
PASS  rekor_checkpoint_signature        expected=Verified OK  actual=Verified OK
PASS  rekor_checkpoint_matches_proof    expected=size=3385223726 root=539466b203048afac2976525a7fabbfa0053f4e5da17076c4c1951
aef7f4c2ab  actual=size=3385223726 root=539466b203048afac2976525a7fabbfa0053f4e5da17076c4c1951aef7f4c2ab
note: NOT covered by any hash or by the log, treat as unverified display text: chain_of_custody, legal_review, notice, surfac
e (SPEC.md §1)

RESULT: content self-consistent and anchor confirmed
exit=0
```

Record 2:

```
PASS  schema_version                    expected=ser.v0.1.legal_ai_output  actual=ser.v0.1.legal_ai_output
PASS  payload_hash                      expected=5086c93c22af206a567fa06cf634682e0a19a19cd0ca2219d371a55cd6e744ea  actual=50
86c93c22af206a567fa06cf634682e0a19a19cd0ca2219d371a55cd6e744ea
PASS  record_commitment                 expected=192eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf  actual=19
2eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf
PASS  evidence_record_id                expected=ser_5086c93c22af206a567fa06cf634682e  actual=ser_5086c93c22af206a567fa06cf6
34682e
PASS  rekor_hash_matches_commitment     expected=8e22075588e58e4111b7dc246a381f27762bfc3721e6fcaf81340ed72d7e6862  actual=8e
22075588e58e4111b7dc246a381f27762bfc3721e6fcaf81340ed72d7e6862
PASS  rekor_log_index                   expected=3247822472  actual=3247822472
PASS  rekor_signature_over_commitment   expected=Verified OK  actual=Verified OK
PASS  displayed_key_matches_log_key     expected=db8e50af50c1c312  actual=db8e50af50c1c312
PASS  rekor_integrated_time             expected=present  actual=1791944346
PASS  rekor_log_key_id                  expected=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d (pinned ke
y)  actual=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d
PASS  rekor_signed_entry_timestamp      expected=Verified OK  actual=Verified OK
PASS  rekor_inclusion_proof             expected=ae8b3273b6ebd9a785cdf5dae859ffd90c68e933cbdfa9e5793dd1c5940c0e31  actual=ae
8b3273b6ebd9a785cdf5dae859ffd90c68e933cbdfa9e5793dd1c5940c0e31
PASS  rekor_checkpoint_signature        expected=Verified OK  actual=Verified OK
PASS  rekor_checkpoint_matches_proof    expected=size=3385225524 root=ae8b3273b6ebd9a785cdf5dae859ffd90c68e933cbdfa9e5793dd1
c5940c0e31  actual=size=3385225524 root=ae8b3273b6ebd9a785cdf5dae859ffd90c68e933cbdfa9e5793dd1c5940c0e31
note: NOT covered by any hash or by the log, treat as unverified display text: chain_of_custody, legal_review, notice, surfac
e (SPEC.md §1)

RESULT: content self-consistent and anchor confirmed
exit=0
```

Cross-check without the verifier. Each record's commitment, hashed once more as ASCII hex, must equal the hash value logged in its Rekor entry:

Record	Commitment (merkle_root)	SHA-256 of commitment hex	Hash in Rekor entry	Match	Log index	Integrated time (UTC)
ser_a383778f88ae4f060279d8ba137926f8	65723a7438b7c366…	7e0c76c0c9382c52…	7e0c76c0c9382c52…	MATCH	3247822463	2026-10-14T02:19:05Z
ser_5086c93c22af206a567fa06cf634682e	192eb3dfe9fa6576…	8e22075588e58e41…	8e22075588e58e41…	MATCH	3247822472	2026-10-14T02:19:06Z

Full values are in evidence/rekor_entry_.json and the record files. Result: PASS for both records (exit 0 and 0). The pinned key id matched the log key id on both runs.

Note on saved entries: the inclusion proof and checkpoint inside a fetched Rekor entry change as the log grows, so the file hash of a saved entry differs between fetches while logIndex, integratedTime, the body and the signed entry timestamp stay constant. The manifest hashes for rekor_entry_*.json are therefore specific to this fetch.

TP-04 Chain walk

```
PASS  [1] schema_version                expected=ser.v0.1.legal_ai_output  actual=ser.v0.1.legal_ai_output
PASS  [1] payload_hash                  expected=a383778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc  actual=a3
83778f88ae4f060279d8ba137926f8d11b97d7b71004599d1f60b74dfae8dc
PASS  [1] record_commitment             expected=65723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c  actual=65
723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c
PASS  [1] evidence_record_id            expected=ser_a383778f88ae4f060279d8ba137926f8  actual=ser_a383778f88ae4f060279d8ba13
7926f8
PASS  [1] rekor_hash_matches_commitment  expected=7e0c76c0c9382c5237c3e4322e670623ff51e1b81f7feaab2eca9dc745a54104  actual=7e
0c76c0c9382c5237c3e4322e670623ff51e1b81f7feaab2eca9dc745a54104
PASS  [1] rekor_log_index               expected=3247822463  actual=3247822463
PASS  [1] rekor_signature_over_commitment expected=Verified OK  actual=Verified OK
PASS  [1] displayed_key_matches_log_key  expected=089f35d7b7bd03fb  actual=089f35d7b7bd03fb
PASS  [1] rekor_integrated_time         expected=present  actual=1791944345
PASS  [1] rekor_log_key_id              expected=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d (pinned ke
y)  actual=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d
PASS  [1] rekor_signed_entry_timestamp  expected=Verified OK  actual=Verified OK
PASS  [1] rekor_inclusion_proof         expected=a3cfa2e2e959f441182e2d006b8d280894268981dbc44d226b8d387185418e9c  actual=a3
cfa2e2e959f441182e2d006b8d280894268981dbc44d226b8d387185418e9c
PASS  [1] rekor_checkpoint_signature    expected=Verified OK  actual=Verified OK
PASS  [1] rekor_checkpoint_matches_proof expected=size=3385226284 root=a3cfa2e2e959f441182e2d006b8d280894268981dbc44d226b8d38
7185418e9c  actual=size=3385226284 root=a3cfa2e2e959f441182e2d006b8d280894268981dbc44d226b8d387185418e9c
PASS  [2] schema_version                expected=ser.v0.1.legal_ai_output  actual=ser.v0.1.legal_ai_output
PASS  [2] payload_hash                  expected=5086c93c22af206a567fa06cf634682e0a19a19cd0ca2219d371a55cd6e744ea  actual=50
86c93c22af206a567fa06cf634682e0a19a19cd0ca2219d371a55cd6e744ea
PASS  [2] record_commitment             expected=192eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf  actual=19
2eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf
PASS  [2] evidence_record_id            expected=ser_5086c93c22af206a567fa06cf634682e  actual=ser_5086c93c22af206a567fa06cf6
34682e
PASS  [2] rekor_hash_matches_commitment  expected=8e22075588e58e4111b7dc246a381f27762bfc3721e6fcaf81340ed72d7e6862  actual=8e
22075588e58e4111b7dc246a381f27762bfc3721e6fcaf81340ed72d7e6862
PASS  [2] rekor_log_index               expected=3247822472  actual=3247822472
PASS  [2] rekor_signature_over_commitment expected=Verified OK  actual=Verified OK
PASS  [2] displayed_key_matches_log_key  expected=db8e50af50c1c312  actual=db8e50af50c1c312
PASS  [2] rekor_integrated_time         expected=present  actual=1791944346
PASS  [2] rekor_log_key_id              expected=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d (pinned ke
y)  actual=c0d23d6ad406973f9559f3ba2d1ca01f84147d8ffc5b8445c224f98b9591801d
PASS  [2] rekor_signed_entry_timestamp  expected=Verified OK  actual=Verified OK
PASS  [2] rekor_inclusion_proof         expected=3c602bac759effd127f8da0f1401fab9ee6a4bffa3d71afb938b3d1faa40d087  actual=3c
602bac759effd127f8da0f1401fab9ee6a4bffa3d71afb938b3d1faa40d087
PASS  [2] rekor_checkpoint_signature    expected=Verified OK  actual=Verified OK
PASS  [2] rekor_checkpoint_matches_proof expected=size=3385227812 root=3c602bac759effd127f8da0f1401fab9ee6a4bffa3d71afb938b3d
1faa40d087  actual=size=3385227812 root=3c602bac759effd127f8da0f1401fab9ee6a4bffa3d71afb938b3d1faa40d087
PASS  [1] chain_start                   expected=no prior_merkle_root  actual=no prior: step 1 starts the chain
PASS  [1] step                          expected=1  actual=1
PASS  [2] step                          expected=2  actual=2
PASS  [1->2] prior_merkle_root          expected=65723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c  actual=65
723a7438b7c36686ed7658f8016b2dd5380eaa3500a40a10423fc6a5c9b99c
PASS  [1->2] log_time_order             expected=[1] logged no later than [2]  actual=2026-10-14T02:19:05Z -> 2026-10-14T02:
19:06Z
PASS  chain declared_length             expected=2 of 2  actual=2 of 2
note: NOT covered by any hash or by the log, treat as unverified display text: chain_of_custody, legal_review, notice, surfac
e (SPEC.md §1)

RESULT: chain of 2 records intact: each record self-consistent and anchored, each names the one before it, steps 1..2 unbroke
n
covers: every record from step 1 to step 2 is present and in the order supplied. A second record sealed off the same predeces
sor (a fork) is not visible to this check.
tail: the records commit to a length of 2, and 2 are present.
not covered: an action that was never sealed leaves no record to check.
exit=0
```

Result: PASS (exit 0). Two records, step 2 names step 1's commitment, logged 02:19:05Z then 02:19:06Z, declared length 2 of 2.

TP-05 Negative controls

(a) One character changed in record 2's output_summary (copy kept as evidence/TP05a-TAMPERED-copy-of-record2.json):

```
FAIL  payload_hash                      expected=f1cb3ba5cbf7edbc6fea000d9ca8bb19b02c741560dfc5461e385f1ef51a39ac  actual=50
86c93c22af206a567fa06cf634682e0a19a19cd0ca2219d371a55cd6e744ea
FAIL  record_commitment                 expected=132d369ef00c0585154450b3222281745dde8801a2874916970f852d475020bf  actual=19
2eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf
FAIL  evidence_record_id                expected=ser_f1cb3ba5cbf7edbc6fea000d9ca8bb19  actual=ser_5086c93c22af206a567fa06cf6
34682e
FAIL  rekor_hash_matches_commitment     expected=27ef7deac4dd6b0110b8ee2457645cb1d4daf0c48c712a73a2e0393095670536  actual=8e
22075588e58e4111b7dc246a381f27762bfc3721e6fcaf81340ed72d7e6862
FAIL  rekor_signature_over_commitment   expected=Verified OK  actual=Verification failure
RESULT: FAILED
exit=1
```

(b) Chain walk with record 2 omitted:

```
FAIL  chain declared_length             expected=2 of 2  actual=1 of 2
RESULT: FAILED (each FAIL line names the record [i] or the link [i-1->i] that failed)
exit=1
```

(c) Chain walk with the two records swapped:

```
FAIL  [1] chain_start                   expected=no prior_merkle_root  actual=names [2] as its prior: records are out of ord
er
FAIL  [1] step                          expected=1  actual=2
FAIL  [2] step                          expected=2  actual=1
FAIL  [1->2] prior_merkle_root          expected=192eb3dfe9fa65765ee508f0e29d00975cbc3370f81419ceef616179ef220ddf  actual=ab
sent: [2] names no prior, so it starts another chain
FAIL  [1->2] log_time_order             expected=[1] logged no later than [2]  actual=2026-10-14T02:19:06Z -> 2026-10-14T02:
19:05Z
RESULT: FAILED (each FAIL line names the record [i] or the link [i-1->i] that failed)
exit=1
```

Result: all three FAILED as expected (exit 1, 1, 1). A PASS later in this workpaper therefore distinguishes an intact record from an altered, shortened or reordered one.

TP-06 Approval event (record 2)

Field	Value
human_review_status	approved
metadata.reviewer_role	Lending officer
metadata.final_decision	approved_reduced_amount
metadata.amount / original_offer	JMD 900,000 / JMD 1,500,000
attorney_reviewer (named approver)	None
event_time / received_at	2026-10-14T02:19:04Z / 2026-10-14T02:19:06.214Z
metadata.prior_merkle_root equals record 1 commitment	PASS
Record 1: human_review_status / recommendation / model_version	pending / withdraw_offer / 4.2

Result: the approval decision, the reviewer's role and the link to the prior step are in the hashed content. The approver's identity is not: attorney_reviewer is null and only the role is recorded. On a real engagement the approver identity is the audited team's D003.4 workflow evidence (request list item 8).

TP-07 Completeness of capture

Not performed on the specimen. The public sample has no audited system of record to reconcile against. On an engagement this procedure tests the team's logging control (request list items 4 and 7); the record cannot evidence it.

TP-08 Storage and retention

Not applicable to the specimen. The public sandbox retains nothing; these two records are published by Verdict at verdict.systems/samples/ as samples. On an engagement, retention is wholly the audited team's control (request list item 5).

TP-09 Signer identity

transparency_anchor.signing_algorithm: ecdsa-p256-sha256. The public sandbox signs each request with a one-time key that names no one. The anchor proves when, not who. Result: recorded as a limitation.

TP-10 Display fields

Every value quoted above comes from the record block or from a verifier PASS line. chain_of_custody, legal_review, notice and surface were not relied on. The verifier printed its standing note that they sit outside the hash. Result: PASS.

TP-11 Standard currency

Tested against the AIUC-1 October 15, 2026 release, read 27 October 2026. Next expected release on or about January 15, 2027; re-perform TP-02 to TP-05 on one record then.

Conclusion of the specimen

TP-01 to TP-05 and TP-10 PASS; TP-06 records the role-only limitation; TP-07 and TP-08 do not apply to a public sample; TP-09 records the signer limitation. The specimen shows what an engagement team should expect to see, and what it should expect the record not to show.

04 Evidence request list

For the audited team. Each item names the procedure it serves. Items 1 to 6 are needed before fieldwork; 7 to 10 during.

#	Item	Serves	Format	Why
1	Scope statement: agents, workflows and the period in scope, and which decision points are sealed	TP-00	One page	Fixes what the records are evidence of
2	The records: every Sealed Evidence Record JSON for the sampled workflows in the period, named by evidence_record_id, plus any chain head record	TP-01 to TP-04	JSON files or an export	The evidence itself
3	Which Verdict tier produced the records (public sandbox or a production tenant) and the signing arrangement	TP-09	One paragraph	Determines what the anchor proves about who
4	Logging code or configuration that calls Verdict, and the list of events that trigger a seal	TP-07, E015.1	Code excerpt or config export	Completeness of capture is the team's control
5	Storage location, retention period, access control and masking policy for the record files	TP-08, E015.3	Policy plus a configuration export	The sandbox retains nothing
6	The AIUC-1 release the team's evidence is prepared against	TP-11	Date	Labels and status change quarterly
7	Reconciliation sample: the team's own identifiers (ticket, transaction, run) for the sampled actions, mapped to evidence_record_id values	TP-07	Spreadsheet	Two-way trace between system of record and sealed records
8	Approval workflow evidence: configuration, code or ticketing showing the human gate, and how the approval event is written into the sealed record	TP-06, D003.4, E015.2	Config export or screenshots	The record carries the role and decision; the workflow carries the approver identity
9	Agent framework logs for the same sample: tool-call parameters and results, sub-agent handoffs, reasoning traces where kept	E015.2 remainder, D003.3	Log export	The parts of E015.2 the record does not carry
10	Change log of model, prompt, policy and tool versions in the period	E015.1, TP-06	Spreadsheet	model_version in record metadata should match

Not to request from Verdict: Verdict holds no copy of a sandbox record and cannot produce a team's records. Requests for records go to the team.

05 Limitations and independence

What a record does not show

1. Who made it. The public sandbox signs each request with a one-time key that names no one. A published registry of signer keys is planned; until it ships, a record does not identify its maker.
2. A fork, or records past the end. The chain walk catches a missing, reordered or dropped record among the records you hold, against the length they declare or a head record held separately. It does not see a second record sealed off the same predecessor, and a chain with no declared length and no held head can be cut after the last record you hold.
3. That every consequential action reached Verdict. Nothing is sealed unless the team's system calls Verdict. Whether every approval and sensitive tool call is sent is a property of that system. Test it as a logging control (TP-07).
4. That the events happened as described. A record shows what was sent and, through its log entry, when it was logged. It does not show the described event was true or the supplying system uncompromised.
5. Anything outside the record block. chain_of_custody, legal_review, notice and surface are display copies and the verifier says so.
6. Admissibility. A record is designed to support a Rule 902(13) or 902(14) certification. Authentication is not admissibility. The court decides.

What Verdict is, and is not

Verdict Systems Inc. is a Delaware corporation based in Houston, Texas. It publishes the SER v0.1 specification, 24 conformance vectors and the reference verifier under Apache-2.0 at github.com/verdictsystems/ser-spec. The public service is a sandbox: an ephemeral key per request, anchored to Sigstore Rekor, stateless. A hardware-backed production signing tier, a production tenant, a SOC 2 report and a HIPAA business associate agreement are planned; none exists today.

Verdict is not affiliated with or endorsed by the Artificial Intelligence Underwriting Company, is not an accredited AIUC-1 auditor, and certifies no one. Nothing in this package states that a record meets or satisfies AIUC-1, or that AIUC-1 requires tamper-evident logs. E015.4 is a supplemental control; sufficiency is the firm's judgment.

Verdict has filed six US patent applications, one non-provisional and five provisional. None has been examined, allowed or granted. An interim patent commitment for conformant implementations is published at verdict.systems/ip-policy.

Dependence on Sigstore Rekor

The anchor check relies on the Sigstore Rekor public log and on Rekor's public key obtained from Sigstore's root-signing repository. Rekor's own operators control the log's availability and sharding. Retain the fetched entry JSON and the pinned key with the workpaper so the check can be repeated from saved files (verify.py --rekor-entry entry.json --rekor-key rekor.pem).

Currency

AIUC-1 labels and status in this package are from the October 15, 2026 release, read on 27 October 2026. Descriptions are paraphrased. Re-read the standard at each quarterly release.

Defects

A line in this package that does not reproduce on your machine is a defect. Report it to founder@verdict.systems or open an issue at github.com/verdictsystems/ser-spec.