L4 Witness Protocol — Normative Draft v0.2

Outcome Traceability & Evidence Envelope Standard (AI Act evidence layer)

- Document ID: L4W-ND-0.2
- Status: Draft (technical review)
- Target regulation: Regulation (EU) 2024/1689 (AI Act)
- Primary mapping: Article 50 (Transparency) + Article 14 (Human oversight)
- Author: Ivan Kotov — Systems Architect | Independent Researcher (Brussels, Belgium)
- Contact: +32485228195 | kotovivan78@gmail.com
- Date: 2026-01-21
- Repositories (reference):
  - `github.com/Kot141078/advanced-global-intelligence`
  - `github.com/Kot141078/sovereign-entity-recursion`
  - `github.com/Kot141078/ester-reality-bound`

Abstract (normative)

This document specifies an implementation-oriented evidence layer that enables tamper-evident, audit-ready records of AI system operation. It defines a Witness Record (event data model) and an Evidence Envelope (cryptographic commitments and signatures) intended to support operational enforcement of AI Act obligations under Article 50 (Transparency) and Article 14 (Human oversight) without requiring disclosure of proprietary prompts, model weights, or internal business logic.

Non-goals (normative)

- This protocol is not a conformity assessment method and does not replace required technical documentation.
- This protocol is not an identity system and does not mandate surveillance of end users.
- This protocol does not prescribe risk classification; it provides evidence hooks for auditing controls.

Table of contents

---

1. Scope

L4 Witness Protocol defines minimal, interoperable structures for producing cryptographically verifiable evidence that certain compliance-relevant actions occurred during operation of an AI system. Implementations MAY extend the data model with additional fields as long as conformance requirements are preserved.

A conforming implementation MUST be able to:

1. emit Witness Records for defined control points,
2. seal them into Evidence Envelopes, and
3. support third-party verification of envelope integrity and chain continuity.

---

2. Terminology and requirement levels

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in BCP 14 (RFC 2119 and RFC 8174) when, and only when, they appear in all capitals.

Actor: entity responsible for emitting records (provider, deployer, or component owner). Overseer: natural person assigned oversight authority for high-risk contexts. Control point: operational moment where evidence is required (e.g., disclosure, marking, oversight decision, intervention).

---

3. Protocol overview

The protocol consists of two layers:

- Witness Record (WR): a structured JSON object describing a single compliance-relevant event.
- Evidence Envelope (EE): a wrapper providing integrity, ordering, and authenticity for a Witness Record.

Witness Records are emitted at control points and then sealed into Evidence Envelopes. Envelopes form an append-only chain via `prev_hash`.

Auditors validate:

1. signature authenticity,
2. hash chain continuity, and
3. conformance completeness (required event types present).

---

4. Witness Record

4.1 Required fields (normative)

Implementations MUST emit Witness Records as JSON objects with the following fields.

Field	Type	Requirement	Notes
`record_id`	string	MUST	Globally unique (UUIDv4/ULID).
`ts`	string	MUST	ISO 8601 timestamp with timezone (e.g., `2026-01-21T10:15:00+01:00`).
`system_id`	string	MUST	Stable identifier of the deployed system/component.
`session_id`	string	SHOULD	Correlation across events (user session / request chain).
`actor_role`	string	MUST	`provider` | `deployer` | `component_owner` | `auditor`.
`event_type`	string	MUST	One of the event types defined in Section 5.
`policy_ref`	string	SHOULD	Pointer to the policy/rule controlling the event (e.g., disclosure policy version).
`input_ref`	object	SHOULD	Commitment to inputs (hashes or redacted pointers).
`output_ref`	object	SHOULD	Commitment to outputs (hashes or redacted pointers).
`oversight`	object	CONDITIONAL	MUST for Art. 14-relevant events (Section 5.2).
`labels`	object	MAY	Freeform tags (non-normative for audit filtering).
`canonicalization`	string	SHOULD	Canonicalization method identifier (Section 6.2).

4.2 Oversight object (Article 14) (normative)

For events that satisfy or evidence human oversight requirements, the `oversight` object MUST be present and MUST include:

- `overseer_role` (MUST): role name (e.g., `clinical_reviewer`, `fraud_analyst`).
- `overseer_id` (SHOULD): pseudonymous identifier; SHOULD NOT expose personal data in the record payload.
- `decision` (MUST): `approve` | `override` | `refuse` | `safe_stop` | `escalate`.
- `reason_code` (SHOULD): standardized code; see Section 8.2 for a RECOMMENDED set.
- `decision_support_ref` (SHOULD): commitment to the information displayed to the overseer.

---

5. Event taxonomy (normative)

Event types define the minimal evidence set. Implementations MAY emit additional event types; however, the following set is normative for conformance.

5.1 Article 50 — Transparency control points (normative)

Event type	Requirement	Evidence intent
`transparency.disclosure`	MUST (when applicable)	User informed they are interacting with an AI system (channel + context).
`transparency.content_label`	MUST (when applicable)	AI-generated/manipulated content marked/labelled; includes label policy reference.
`transparency.biometric_emotion_notice`	MUST (when applicable)	Notice for emotion recognition / biometric categorisation contexts.

5.2 Article 14 — Human oversight control points (high-risk contexts) (normative)

Event type	Requirement	Evidence intent
`oversight.assignment`	MUST	Overseer assigned and authority established for a session/decision boundary.
`oversight.decision`	MUST	Overseer decision recorded (`approve/override/refuse/safe_stop/escalate`).
`oversight.intervention`	MUST (when used)	Actual intervention/override applied; linkage to affected `output_ref`.
`oversight.safe_stop`	SHOULD	System-level stop action recorded; used when risk cannot be mitigated by override.

---

6. Evidence Envelope

6.1 Required fields (normative)

Each Witness Record MUST be sealed into an Evidence Envelope. The Envelope MUST provide: (a) content integrity, (b) ordering via chain linkage, and (c) authenticity via signatures.

Field	Type	Requirement	Notes
`envelope_id`	string	MUST	Unique identifier for the envelope.
`wr_hash`	string	MUST	Hash of canonicalized Witness Record payload.
`prev_hash`	string	MUST	Hash of the previous envelope (or `GENESIS` marker for chain start).
`hash_alg`	string	MUST	RECOMMENDED: `SHA-256`.
`sig_alg`	string	MUST	RECOMMENDED: Ed25519.
`key_id`	string	SHOULD	Identifier of signing key (public key retrievable by auditor).
`signature`	string	MUST	Signature over (`prev_hash || wr_hash || envelope_id || ts`).
`ts`	string	MUST	Envelope timestamp (ISO 8601). SHOULD equal or exceed WR timestamp.

6.2 Canonicalization (normative)

To ensure reproducible hashing across implementations, the Witness Record JSON MUST be canonicalized prior to hashing.

- RECOMMENDED: JSON Canonicalization Scheme (JCS, RFC 8785).
- If an implementation uses an alternative canonicalization method, it MUST declare the method identifier in the Witness Record (`canonicalization`).

6.3 Selective disclosure (normative)

Implementations MAY omit sensitive payloads by committing to them via hashes (`input_ref` / `output_ref`). When payloads are omitted, the record MUST include sufficient metadata for audit reasoning (e.g., content category, `policy_ref`, `event_type`, and linkage identifiers).

---

7. Conformance profiles (normative)

Conformance defines the minimal set of event types and fields required for interoperability and audit verification.

Profile	Intended use	Minimum required event types
`L4W-BASE`	Transparency evidence (Art. 50)	`transparency.disclosure`; `transparency.content_label` (when applicable)
`L4W-HRO`	High-risk + human oversight evidence (Art. 14)	`oversight.assignment`; `oversight.decision`; `oversight.intervention` (when used)
`L4W-FULL`	Combined	`L4W-BASE` + `L4W-HRO`

An implementation claiming a profile MUST produce all required event types for that profile when the corresponding obligations are applicable in the deployment context.

---

8. Verification procedure (auditor view) (normative)

A verifier MUST be able to validate a chain segment using the following steps:

1. Canonicalize WR and compute `wr_hash` using declared canonicalization and `hash_alg`.
2. Verify `EE.signature` using `sig_alg` and public key identified by `key_id`.
3. Verify chain continuity: `EE.prev_hash` equals previous EE hash (or `GENESIS` for first).
4. Verify conformance: required event types exist for the selected profile; required fields populated.
5. Verify linkage: interventions reference the affected `output_ref` and `session_id` correlation.

8.2 Recommended reason codes (informative but audit-friendly)

```
RISK_HIGH_CONFIDENCE
RISK_AMBIGUITY
POLICY_VIOLATION_SUSPECTED
DATA_QUALITY_INSUFFICIENT
USER_CLARIFICATION_REQUIRED
SYSTEM_DEGRADED_MODE
EXTERNAL_CONSTRAINT_TRIGGERED
```

9. Minimal examples (informative)

9.1 Article 50 disclosure event (Witness Record)

```
{
  "record_id": "01J3Z7Q9X0K6K5J7S8E0QW2H6A",
  "ts": "2026-01-21T10:15:00+01:00",
  "system_id": "deploy:example-chat:brussels-01",
  "session_id": "sess-7f3b1a",
  "actor_role": "deployer",
  "event_type": "transparency.disclosure",
  "policy_ref": "disclosure-policy/v1.2",
  "input_ref": { "hash_alg": "SHA-256", "hash": "…", "redacted": true },
  "output_ref": { "hash_alg": "SHA-256", "hash": "…", "redacted": true },
  "labels": { "channel": "web-ui", "locale": "en-GB" },
  "canonicalization": "RFC8785-JCS"
}
```

9.2 Article 14 oversight decision (Witness Record)

```
{
  "record_id": "01J3Z7S0A7B3M9Y3QK7E2Z2T1P",
  "ts": "2026-01-21T10:16:12+01:00",
  "system_id": "deploy:high-risk:triage-01",
  "session_id": "sess-7f3b1a",
  "actor_role": "deployer",
  "event_type": "oversight.decision",
  "policy_ref": "oversight-policy/v0.9",
  "output_ref": { "hash_alg": "SHA-256", "hash": "…", "redacted": true },
  "oversight": {
    "overseer_role": "risk_reviewer",
    "overseer_id": "ovr-9c2d (pseudonymous)",
    "decision": "override",
    "reason_code": "RISK_AMBIGUITY",
    "decision_support_ref": { "hash_alg": "SHA-256", "hash": "…", "redacted": true }
  },
  "canonicalization": "RFC8785-JCS"
}
```

9.3 Evidence Envelope example

```
{
  "envelope_id": "env-00000027",
  "ts": "2026-01-21T10:16:13+01:00",
  "hash_alg": "SHA-256",
  "sig_alg": "Ed25519",
  "prev_hash": "a6c1…",
  "wr_hash": "93bf…",
  "key_id": "keyset-2026Q1#ed25519-01",
  "signature": "base64url(…)"
}
```

10. Security considerations (normative)

Implementations MUST document a threat model covering at least:

- record deletion,
- record re-ordering,

- record tampering,
- key compromise,
- replay attacks,
- privacy leakage via metadata.

Implementations SHOULD implement key rotation and SHOULD provide auditors with public key material and rotation history sufficient to verify signatures over time.

When privacy-sensitive domains are involved, implementations SHOULD minimize personal data in records and SHOULD prefer pseudonymous overseer identifiers and hashed commitments over raw payload storage.

---

11. References

11.1 Normative references

- BCP 14: RFC 2119, RFC 8174 (requirement keywords)
- RFC 8785 (JSON Canonicalization Scheme) — RECOMMENDED canonicalization
- RFC 8032 (EdDSA) — RECOMMENDED signature scheme (Ed25519)

11.2 Legal reference

- Regulation (EU) 2024/1689 (AI Act), Official Journal of the European Union

---

Appendix A. Design rationale (non-normative)

Explicit bridge: $c = a + b$ → accountability (a) + verifiable procedures (b) = auditable outcome evidence (c).

Hidden bridge 1 (Ashby): audit robustness requires requisite variety: multiple control-point events are harder to fake than a single monolithic log.

Hidden bridge 2 (Cover & Thomas): constrained channels demand structured compression: standardized records preserve meaning while reducing disclosure of payload.

Earth paragraph: Treat evidence like an ECG trace: it is not a story but a signal. If you can redraw it after the fact, it is not evidence. Hash-chained envelopes make the signal mechanically hard to rewrite.

Reference Source: GitHub Commit 7a376cf (2026-01-21)