NuWyre System Description

Effective date: *to be filled at signing* **Version:** v1.0 (subject to attorney revision) **Audience:** non-technical readers — judges, juries, opposing counsel, regulators **Custodian of record:** Jeremy Fish, Founder & CEO, NuWyre, Inc.

---

1. Plain-English overview

NuWyre operates an evidence-layer service for customer-facing artificial intelligence agents. When a customer organization deploys an AI agent that interacts with end users (e.g., over voice or chat), NuWyre captures a structured record of each interaction, attaches cryptographic integrity guarantees to the record, and preserves the record for the customer's regulatory and litigation needs. The result is a tamper-evident, externally verifiable log of AI agent activity that the customer can produce in response to subpoena, regulatory inquiry, or audit — without depending on NuWyre's continued operation.

This document describes the system in plain English so that a non-technical reader can understand what NuWyre records, how the records achieve tamper-evident integrity, and what the customer can do with them.

---

2. What gets recorded

When a customer's AI agent interacts with an end user, the customer's adapter software (provided by NuWyre or built to NuWyre's adapter specification) captures a structured record of the interaction. Each record — called an **event** — contains:

- **A timestamp** indicating when the interaction occurred (recorded to millisecond precision).
- **A unique session identifier** grouping multiple events from the same end-user interaction.
- **A sequence number** indicating the order of events within the session.
- **An agent identifier** indicating which AI agent produced the event.

- **A role label** indicating whether the event was an agent response, a user message, a tool call, or another role type.
- **Content:** the actual text of the agent response, user message, or system message. Content may be redacted under specific conditions; see § 5.
- **Classification labels** describing the type of content (e.g., consent-eliciting language, disclosure of personal health information, financial advice).
- **A cryptographic hash** (SHA-256, a widely-used cryptographic function) of the content.
- **A chain hash** linking each event to the previous event in the same session, so that altering any event's content breaks the chain in a way that can be detected automatically.

When the AI agent interacts with the user over a voice channel, the customer's adapter may additionally capture the **audio recording** of the call. Audio is stored separately, content-addressed by its own SHA-256 hash. The hash of the audio is recorded in the corresponding event so that the audio file's integrity can be verified independently of the event chain.

---

3. How the chain works

Each event's chain hash is computed by combining the previous event's hash with the current event's content hash. Mathematically, this is similar to how a paper ledger's pages reference each other: if someone tries to remove or modify a page in the middle of the ledger, the references to that page from later pages no longer match, and the modification is detectable.

In NuWyre's system, the chain runs at multiple levels:

1. **Per-session chain:** events within a single end-user interaction are chained in order. Removing any event from the middle of a session breaks the chain.
2. **Per-day root:** at the end of each calendar day (UTC), NuWyre computes a single cryptographic hash — a **Merkle root** — that summarizes all events recorded for a given customer that day. Removing any event from any session that day changes the daily root.
3. **External anchors** (see § 4): the daily root is recorded in three independent external systems, so that the customer's daily root for any past day can be verified against external records that NuWyre does not control.

This means that to falsify a NuWyre record, an attacker would have to:

- Alter the original event content,
- Alter every subsequent event in the same session (because each subsequent event's chain hash depends on the altered one),
- Recompute the daily root that includes the altered session,
- Alter the external anchors that recorded the daily root.

The last step is what makes the system tamper-evident. NuWyre records the daily root in three external places, two of which are operated by entities other than NuWyre, and one of which is recorded permanently on a public blockchain.

---

4. Dual anchoring

Every daily root is anchored in three independent external systems:

4.1 OpenTimestamps (Bitcoin blockchain)

NuWyre submits the daily root to the OpenTimestamps service, which aggregates the daily root with other submitters' hashes into a single Merkle root and inscribes that root onto the Bitcoin blockchain. Once the Bitcoin transaction is confirmed by the Bitcoin network (typically within an hour), the daily root's existence at the time of inscription is recorded permanently on a globally-replicated, decentralized public ledger.

To falsify the NuWyre daily root would require also falsifying the Bitcoin blockchain — which would require controlling a majority of the Bitcoin mining hash rate, an attack widely considered economically infeasible at Bitcoin's current scale.

4.2 RFC 3161 timestamp authority tokens

NuWyre additionally submits the daily root to multiple independent Time-Stamp Authorities ("TSAs") that operate under the RFC 3161 protocol. Each TSA returns a **timestamp token** — a cryptographically-signed assertion by the TSA that the daily root existed at the time the TSA received it. NuWyre captures these tokens at submission time and stores them with the daily root.

Currently NuWyre uses three independent TSAs (FreeTSA, Sectigo, and DigiCert). The certificate chains for each TSA are captured at submission time so that the tokens re-

main verifiable even if the TSA's certificates rotate.

4.3 Signed Git commits to a public repository

NuWyre additionally commits the daily root to a public Git repository at `https://github.com/NuWyre/anchors`. Each commit is digitally signed using the Ed25519 cryptographic key of NuWyre's founder, and the public Git commit history establishes the daily root's existence at the time the commit was made.

NuWyre's anchor publication pipeline is designed to be redundantly mirrored. The primary commit history is hosted at `github.com/NuWyre/anchors`; redundant mirroring to `codeberg.org/NuWyre/anchors` is part of the published architecture and is configured to run nightly via a GitHub Action in the anchor repo, with the mirror activating once anchor commits begin landing. NuWyre additionally supports integration of third-party archival services (Internet Archive, Software Heritage) as those services index the public anchor repo over time.

Mirror operational status (v1.0 draft): the GitHub primary anchor repo is established. The Codeberg mirror's GitHub Action and third-party archival indexing are operator-side configuration that activates as part of Phase 5 production deployment. Customers should refer to NuWyre's current operational status disclosures for the live mirror state.

4.4 Why three independent anchors

The three anchoring systems are chosen so that no single failure or compromise of any one anchor would invalidate the integrity claim:

- If the Bitcoin blockchain were somehow compromised, the RFC 3161 tokens and the Git commit history would still record the daily root's existence.
- If the RFC 3161 TSAs were all simultaneously compromised, the Bitcoin anchor and the Git commit history would still record the daily root.
- If GitHub were unavailable (or compromised), the Codeberg mirror (when operational) and any third-party archival copies (Internet Archive, Software Heritage) would record the daily root.
- If NuWyre's own infrastructure were unavailable or compromised, all three external anchors would still record the daily roots and would be independently verifiable using only the published anchor data and publicly-available verification software (see § 6).

5. How audio is bound

When the customer's adapter captures an audio recording of a voice-channel interaction, the audio file is stored separately from the event records. The audio file is stored in NuWyre's cloud object storage infrastructure (current subprocessor identified in NuWyre's published subprocessor list), under the subprocessor's standard server-side encryption at rest. The storage path is `customer-organization-id/sha256-of-audio.extension`.

The audio file's SHA-256 hash is recorded in the database in the `audio_records` table, alongside metadata about the recording (duration, MIME type, audio sample rate, retention class). The `audio_records` row references the event that the audio binds to.

When the customer or a third party retrieves an evidence bundle that includes audio, the bundle includes:

- The audio file binaries.
- The `audio_records` metadata.
- The event records that reference the audio.
- The chain hashes that bind the events.
- The daily root that includes the relevant session.

The recipient can verify that the audio file's content matches the recorded SHA-256, that the event references the audio, that the chain is intact, and that the daily root is recorded in the external anchor systems. If any of these verifications fail, the recipient knows the bundle has been tampered with after creation.

---

6. How verification works

NuWyre publishes an open-source command-line verification tool (the "verifier CLI") at `https://github.com/NuWyre/cli` along with its build provenance. The verifier CLI:

- Accepts a NuWyre evidence bundle as input.
- Parses the bundle's structure.
- Computes each event's content hash and verifies it matches the recorded hash.
- Reconstructs the chain hashes and verifies the chain is intact.

- Computes the per-session Merkle root and verifies it matches the recorded session root.
- Computes the per-day Merkle root and verifies it matches the recorded daily root.
- Verifies that the daily root appears in the OpenTimestamps anchor record (which references the Bitcoin blockchain via the OpenTimestamps calendar).
- Verifies that the daily root appears in the RFC 3161 timestamp tokens (which reference the independent TSAs' signatures).
- Verifies that the daily root appears in the signed Git commit history (which references the public anchor repository).

The verifier CLI requires no NuWyre infrastructure to perform any of these verifications. It does require:

- A copy of the evidence bundle (provided to the recipient by the customer).
- Internet access to public Bitcoin chain data, which the verifier obtains via HTTPS APIs operated by independent Bitcoin block-explorer services (currently `blockstream.info` and `mempool.space`). The verifier can be extended to use any compatible HTTPS API or a self-hosted Bitcoin node.
- Internet access to the public anchor repository (or any mirror thereof; when Codeberg mirror or Internet Archive archival copies are operational, the recipient does not need to trust any one host).
- The published verifier CLI itself, which is open source and can be independently audited or rebuilt from source.

The verifier CLI does **not** require:

- Access to NuWyre's servers.
- Trust in NuWyre's claims.
- Cooperation from any TSA at verification time.

This means that a recipient — for example, a regulator investigating a customer's TCPA compliance, or opposing counsel cross-examining a customer's records in litigation — can independently verify the integrity of a NuWyre evidence bundle without depending on NuWyre's continued cooperation, NuWyre's continued operation, or NuWyre's good faith. The verification is **adversarially robust**: even if NuWyre were to act in bad faith later, prior records would still be verifiable against the immutable external anchors.

7. What custody NuWyre maintains

NuWyre maintains custody of:

- The event records, evaluation records, audit log records, and other database records held in NuWyre's cloud database infrastructure (current cloud subprocessor identified in NuWyre's published subprocessor list).
- The audio binaries held in NuWyre's cloud object storage infrastructure (current cloud subprocessor identified in NuWyre's published subprocessor list), under the subprocessor's standard server-side encryption at rest.
- The signing keys used to produce evidence bundles' Ed25519 signatures. These keys are managed by NuWyre's key-management infrastructure: the development environment loads the key from a file on disk under restricted permissions; the production environment integrates with a cloud KMS service (target-state for Phase 5+ production deployment). Both environments restrict private-key bytes to NuWyre's authenticated server-side processes — the key is never exposed in browser bundles or client-side code.
- The Git commit signing keys used to anchor daily roots to the public anchor repository. These keys are managed by NuWyre's founder under physical security appropriate to their role.

NuWyre does **not** maintain custody of:

- The Bitcoin blockchain (operated by the global Bitcoin network).
- The RFC 3161 TSA infrastructure (operated by independent TSAs).
- The public anchor Git repository's hosting infrastructure (operated by GitHub and Codeberg).
- The third-party archival copies of the public anchor repository (operated by Internet Archive, Software Heritage).

The customer can also obtain a copy of their evidence bundles via the customer portal at any time, so the customer maintains an independent copy of their own records that does not depend on NuWyre's continued retention. This is an important property: the customer is not dependent on NuWyre's continued operation to access their own records, and is not trapped if NuWyre changes ownership, shuts down, or otherwise becomes unavailable.

8. Custody chain in summary

The chain of custody for a NuWyre evidence record runs as follows:

1. The end user interacts with the customer's AI agent.
2. The customer's adapter captures a structured event record and (when applicable) audio binary.
3. The adapter transmits the event to NuWyre's ingestion API, authenticated via a customer-specific API key.
4. NuWyre's ingestion API validates the event against the customer's schema, computes content and chain hashes, and writes the event to NuWyre's database.
5. The audio binary, if present, is written to NuWyre's object storage and the corresponding `audio_records` row is written to the database.
6. At end of day, NuWyre's daily-root job aggregates all events from all customers and writes per-customer Merkle roots.
7. Each daily root is submitted to OpenTimestamps, the three RFC 3161 TSAs, and the public Git anchor repository.
8. When the customer requests an evidence export, NuWyre's bundle-generation pipeline assembles the events + evaluations + audio + legal documents into a signed Ed25519 archive.
9. The customer downloads the bundle from NuWyre's customer portal.
10. The customer can verify the bundle locally using the verifier CLI without NuWyre's involvement.

At any point in this chain, the integrity claims are externally verifiable. The records are tamper-evident: any modification after creation produces a detectable inconsistency.

---

9. Document control

- **Version:** v1.0 (draft; subject to attorney revision)
- **Last updated:** 2026-05-13
- **Next scheduled review:** within twelve (12) months from effective date
- **Drafted by:** NuWyre, Inc.

- **Attorney revision status:** pending; budget line ~$2,000-$5,000 per substantive review per docs/build-plan.md v3.1.7
- **Distribution:** published at `https://nuwyre.com/legal/system-description.pdf`; included in every NuWyre evidence-bundle export's `legal/` subdirectory starting with the first export generated after the effective date.

---

10. Signature

Custodian signature: _____

Date: _____