NexArt Verification Model (Unified)

How AI execution can be independently verified

1. The problem

As AI systems move into production, organisations increasingly need to answer a simple but critical question:

Can you prove what actually happened?

Most existing approaches cannot.

• Logs are not proof. They are produced by the same system whose behaviour is being questioned.

• Reconstruction is unreliable. Re-running a system produces a new execution, not the original one.

• Enterprises need evidence, not assurances.

NexArt produces a sealed execution record — a Certified Execution Record (CER) — that is tamper-evident and independently verifiable.

Verification does not rely on trusting NexArt. It relies on cryptography.

2. What a CER is

A Certified Execution Record (CER) captures inputs, outputs, and metadata.

It is canonicalised into a deterministic representation and hashed using SHA-256.

If any part of the record changes, the hash breaks.

A CER is tamper-evident and independently verifiable.

3. The verification model

Verification consists of independent cryptographic checks plus a classification layer.

3.1 Integrity — has the record changed?

1. Canonicalise the CER

2. Recompute SHA-256

3. Compare with certificateHash

If they match, the record is unchanged.

Fully deterministic, independent, and requires only SHA-256.

3.2 Authenticity — who sealed it?

Verified using Ed25519 signature over a canonical payload.

Requires public key and signature verification.

Integrity and authenticity are separate checks.

3.3 Classification — interpreting the result

VERIFIED — strict cryptographic validity

LEGACY_VERIFIED — valid under legacy rules

UNVERIFIABLE — cannot be reproduced

FAILED — cryptographic failure

Classification is additive and does not replace verification.

4. Independent verification

Requires only the CER, a public key, and standard cryptographic libraries.

No NexArt dependency is required.

Protocol 1.2.0 uses nexart-v1 canonicalisation.

Protocol 1.3.0 uses RFC 8785 (JCS).

5. Timestamp model

Node-issued timestamp bound to certificateHash.

Provides ordering guarantee but not independent proof of existence.

External TSA (e.g. DigiCert) can be added without changing records.

6. Trust boundaries

NexArt proves integrity and authenticity (when signed).

It does not prove independent time of existence, completeness, or public log inclusion.

7. Why this matters

Enables dispute resolution, auditability, enterprise trust, and verifiable AI workflows.

The relying party verifies the record themselves.

8. Closing

Verification does not depend on NexArt.

A CER is deterministic, tamper-evident, and independently verifiable.

A proof that requires trusting the prover is not a proof.

What you now have

Consistent across Node + SDK

Audit-defensible

Enterprise-ready

Clear on limitations