Governance Verification for Authority-Separated AI Execution:

Conformance, Audit, and Reproducibility Evidence from the CNX Framework

Article type: Systems architecture and validation evidence article

Target venue: Carlonoscopen Journal of Coherence Intelligence (CJCI)

Target archival deposit: Zenodo

Version: v1.0 production baseline

Date: 2026-06-15

DOI: 10.5281/zenodo.20706623

Author: Ivan Silva

ORCID: 0009-0005-2284-8891

Affiliation: Carlonoscopen, LLC

Suggested Citation

Silva, I. (2026). *Governance Verification for Authority-Separated AI Execution: Conformance, Audit, and Reproducibility Evidence from the CNX Framework*. Carlonoscopen Journal of Coherence Intelligence. https://doi.org/10.5281/zenodo.20706623

Abstract

Agentic AI systems increasingly combine model reasoning, tools, memory, orchestration, external services, and execution privileges. This creates a governance problem that is not solved by model capability alone: an AI system may reason, recommend, classify, or simulate, but those outputs should not automatically acquire authority to cause operational consequences. This paper presents a governance-verification baseline for the Coherence Nexus (CNX) framework, an authority-separated execution architecture in which identity, policy, mediation, integrity, audit, and lifecycle controls are evaluated before AI-derived outputs may become governed actions.

Building on prior CNX work on governed capability execution, this article focuses on evidence rather than broad architectural positioning. The reviewed package is divided into Tier 1 governance/specification artifacts and Tier 2 validation/reproducibility artifacts. Tier 1 defines the baseline problem, conformance test suite, workflow protocol, and agent orchestration specification. Tier 2 provides the Phase 4 authority-separation report, audit-log test source, evidence-pass manifests, an exploratory ARPG-AI engineering validation note, and hardware characterization summaries.

The central quantitative result comes from the Phase 4 authority-separation report. Across ten request checks, expected outcomes were preserved: three allowed requests remained allowed, two restricted requests remained restricted, four prohibited requests were refused, and one invalid request remained invalid. No violations were reported, CNX responses were stored separately from measurement outputs, and the report

records `authority_separation_holds: true`. This evidence does not prove universal AI safety, truth detection, or full enterprise certification. It supports a narrower and more defensible claim: CNX can enforce and measure separation between AI-derived reasoning outputs and operational authority within a governed execution architecture.

Keywords: AI governance; authority separation; governed execution; CNX; conformance testing; audit logs; reproducibility; agent orchestration; policy enforcement; capability control

Highlights

- CNX treats authority as separate from intelligence, capability, and model confidence.
- The conformance suite is invariant-based rather than merely functional.
- The governance rule is expressed through identity, authority, mediation, and integrity validity.
- The Phase 4 authority-separation report provides the main quantitative result.
- Audit-log tests support append-only, hash-chained, tamper-evident observability.
- Manifests and hardware summaries support reproducibility, but remain secondary evidence.
- Blockchain, crypto, energy, and broader distributed-orchestration claims are intentionally excluded from this paper.

1. Introduction: Baseline Problem

Modern AI systems increasingly operate beyond conversational output. They may call tools, coordinate agents, interact with APIs, summarize evidence, produce policy-relevant recommendations, or participate in workflows that affect external systems. As the operational surface expands, the critical governance question changes.

The question is no longer only:

```
What did the model produce?
```

It becomes:

```
Who or what is allowed to convert that output into action?
```

This paper addresses that question through the CNX framework. CNX is a governed execution architecture designed around authority separation. Its central principle is that AI-derived reasoning, classification, recommendation, or measurement does not automatically confer authority to execute.

The immediate motivation for this paper is a curated publication-support package containing governance specifications and validation artifacts. The package shows a progression from baseline platform state, to conformance testing, to governed orchestration, to measured authority separation, to audit and reproducibility evidence. The result is not a general claim that CNX solves all AI safety problems. The result is a focused governance-verification baseline for authority-separated AI execution.

1.1 Problem Statement

Many AI governance discussions conflate four distinct properties:

1. **Intelligence:** the ability to reason, infer, classify, or recommend.

2. **Capability:** the ability to perform or request an operation.
3. **Permission:** the local allowance to access a tool or route.
4. **Authority:** the legitimate right to produce an operational consequence.

These properties should not be collapsed into one another. A model may correctly identify a useful action without being authorized to execute it. An agent may possess a technical capability without being allowed to invoke it in a specific context. A measurement system may classify a request without being permitted to rewrite the policy boundary that governs the request.

The CNX governance problem can therefore be stated as follows:

> Given an identity, a requested capability, a payload, a policy context, and a possible operational consequence, determine whether execution is admissible and record the decision in a way that can be audited and reproduced.

This paper evaluates whether the reviewed CNX evidence package supports that governance claim.

1.2 Scope

This article is intentionally narrow. It focuses on:

- governance specification;
- conformance testing;
- governed execution architecture;
- authority-separation evaluation;
- audit and observability;
- reproducibility and integrity artifacts.

It does not attempt to establish:

- universal AI safety;
- truth detection;
- autonomous correctness;
- production certification for all enterprise environments;
- blockchain or crypto-system deployment;
- energy-system or physical-actuator validation;
- final validation of BZ or ARPG as a general theory.

Those topics may become follow-on work. They are not required for the central claim of this paper.

2. Prior Work and Publication Context

This paper builds on the earlier CNX article:

> Silva, I. (2026). *From Agent Harnesses to Authority Infrastructure: CNX as Governed Capability Execution for Model-Independent AI Systems*. Carlonoscopen Journal of Coherence Intelligence, 1(15). https://doi.org/10.5281/zenodo.20694341

That prior work positioned CNX as authority infrastructure rather than as a model, agent harness, or tool router. It introduced the central invariant that intelligence should not automatically become authority. The current paper is a verification companion. It asks whether the supporting evidence package can substantiate the architecture through specifications, tests, reports, manifests, and reproducibility artifacts.

The distinction is important:

- The prior paper defined the architectural position.
- The current paper evaluates conformance, audit, and reproducibility evidence.

3. Materials and Evidence Package

The reviewed support package is divided into two tiers.

3.1 Tier 1: Governance and Specification Evidence

Tier 1 contains documents that define the governance architecture and verification expectations:

- baseline state before CAP-1 v1.2 upgrade;
- CNX conformance test suite specification;
- CNX conformance test suite developer references;
- workflow orchestration protocol;
- CNX agent orchestration specification.

These files are used to establish the problem, define the governance rule, specify conformance criteria, and describe how agents and workflows are mediated.

3.2 Tier 2: Validation and Reproducibility Evidence

Tier 2 contains validation and reproducibility artifacts:

- Phase 4 authority-separation report;
- audit-log test source;
- evidence-pass manifests;
- ARPG-AI experimental engineering validation note;
- hardware/VRAM characterization summaries.

These files are used to support the results, audit, integrity, and reproducibility sections. Only the Phase 4 authority-separation report is treated as primary quantitative evidence. The other Tier 2 artifacts are supporting or appendix-level evidence.

4. Governance Architecture

The CNX governance architecture is based on the separation of reasoning space from execution space. AI systems may reason internally, produce measurements, classify requests, or propose actions. However, governed external action requires validation through a governance boundary.

The conformance specification expresses the core admissibility rule as:

```
Admissible(m) =
identity_valid AND
authority_valid AND
mediation_valid AND
integrity_valid
```

This rule is the paper's main verification object. It states that a message or action is not admissible merely because it is technically possible or semantically plausible. It must satisfy identity, authority, mediation, and integrity constraints.

4.1 Identity

Identity validation asks whether the actor is known, active, and correctly bound to its registered instance or role. Unknown identities, suspended identities, and mismatched instances are expected to be blocked.

4.2 Authority

Authority validation asks whether the actor is permitted to perform the requested action within the relevant policy scope. A request outside policy scope is expected to be denied, even if the requester can describe or technically perform the operation.

4.3 Mediation

Mediation validation asks whether communication and execution pass through the CNX gateway rather than through ungoverned side channels. Direct agent-to-agent communication, broker injection, or filesystem side-channel messaging are treated as governance failures unless explicitly mediated.

4.4 Integrity

Integrity validation asks whether payloads, signatures, replay protections, timestamps, nonces, sequence ordering, and ledger records remain consistent. Tampering, invalid signatures, and replayed messages are expected to be blocked.

5. Governance Verification Framework

The CNX conformance suite is invariant-based. Functional tests verify that a system can perform actions. Conformance tests verify that invalid actions cannot occur.

The suite defines test cases using a canonical structure:

```
Test ID
Invariant Target
Preconditions
Action
Expected Outcome
Verification Method
```

The reviewed specification identifies the following categories:

- identity tests;
- authority tests;

- mediation tests;
- integrity tests;
- state-machine tests;
- ledger tests;
- reset-discipline tests;
- fail-secure tests;
- observability tests.

This structure is important because governance verification should not depend on whether ordinary workflows succeed in benign conditions. A governance framework must also demonstrate that invalid or adversarial transitions are blocked.

5.1 Message Pipeline

The conformance specification defines a message pipeline:

```
RECEIVED
-> VALIDATING_STRUCTURE
-> VALIDATING_IDENTITY
-> VALIDATING_FRESHNESS
-> VALIDATING_INTEGRITY
-> EVALUATING_POLICY
-> RECORDING_LEDGER
-> ROUTING
-> DELIVERED
```

Each stage is expected to emit telemetry. This creates a structured audit path from request intake to final delivery or refusal.

5.2 Fail-Secure Behavior

The conformance specification requires blocking behavior under gateway failure, policy-engine failure, or ledger failure. This is central to authority-separated execution. If governance infrastructure fails open, authority separation becomes advisory rather than enforceable.

5.3 Conformance Criteria

An implementation is conformant only if:

- all tests pass;
- all invariants are enforced;
- reset events trigger correctly;
- ledger integrity is preserved;
- fail-secure behavior is verified.

The specification therefore frames governance as a testable property rather than a policy slogan.

6. Governed Execution Architecture

The agent orchestration specification defines agents as governed execution identities, not autonomous authorities. Agent actions must pass through the CNX gateway governance boundary.

The orchestration layer governs:

- agent identity;
- agent lifecycle;
- project coordination;
- execution authorization;
- agent supervision;
- termination control.

The specification states that all orchestration events pass through the CNX gateway governance membrane. This positions the gateway as the authority-validation point for agent lifecycle and execution events.

6.1 Orchestration Scope

The specification distinguishes internal reasoning from governed external action. CNX does not govern every internal reasoning step of an AI system. It governs external domain actions, including:

- agent lifecycle transitions;
- agent communication;
- external system requests;
- project lifecycle events;
- execution authorization.

This distinction is essential. It allows AI systems to reason, simulate, and propose while keeping operational consequences under governance.

6.2 Project and Agent Lifecycle

The reviewed specification defines projects as coordinated execution scopes and requires agents to operate within projects. Agent identities include fields such as agent identifier, instance identifier, role, project, owner, version, and status. Lifecycle states include proposed, registered, active, standby, and terminated.

Agents cannot change lifecycle state without gateway mediation. Activation requires project state, policy validity, and identity validity. Standby behavior is triggered when supervision is lost, execution windows close, projects pause, or policy enforcement is triggered.

6.3 Execution Authorization

Unattended execution requires explicit authorization. The authorization structure includes project scope, authorizing party, execution window, and authorized agents. The gateway scheduler enforces execution windows.

This is one of the most important governance details in the package. It prevents unattended execution from being treated as a natural consequence of agent intelligence or technical capability.

6.4 Governance Logging

Every orchestration event is expected to generate a governance log entry. Events include proposal, registration, activation, standby, termination, project termination, and execution authorization. This creates the observability path needed for audit and reproducibility.

7. Authority Separation Evaluation

The Phase 4 authority-separation report is the central quantitative evidence in the reviewed package.

The report states the governing principle:

```
PhaseSeed measures; CNX governs authority.
PhaseSeed outputs cannot self-convert into authority.
```

This is the operational form of authority separation. A measurement layer may classify, warn, or review, but its output cannot authorize itself.

7.1 Evaluation Set

The report contains ten request checks:

Outcome class	Count	Result
Allowed	3	All mapped as allowed
Restricted	2	All mapped as restricted
Prohibited	4	All mapped as prohibited and refused
Invalid	1	Mapped as invalid
Total	10	Expected mappings preserved

The allowed requests included claim audit, evidence review, and freshness warning capabilities. The restricted requests included public-summary and publication-note capabilities. The prohibited requests included belief-legitimacy scoring, group suppression, person ranking, and source blacklisting or access denial. The invalid request was missing required fields.

7.2 Result

The report records:


```
{
  "cnx_asked_phaseseed_to_rewrite_measurement": false,
  "cnx_responses_stored_separately": true,
  "raw_response_count": 9,
  "violations": [],
  "authority_separation_holds": true
}
```

This supports the central claim that CNX preserved authority boundaries across allowed, restricted, prohibited, and invalid request types.

7.3 Interpretation

The result does not show that all future policies are correct or that all possible bypass attempts have been eliminated. It shows that, within this test set, the governance layer preserved the expected authority mapping and prevented measurement outputs from self-converting into authority.

The most important result is the refusal of prohibited capabilities. The system did not merely mark them as risky. It mapped them to the prohibited class and recorded refusal checks.

8. Audit and Observability

Auditability is a required property for authority-separated execution. If decisions cannot be reconstructed, governance becomes difficult to inspect and impossible to reproduce.

The audit-log test artifact validates append-only, hash-chained, tamper-evident logging behavior. The test source covers:

- empty initial audit state;
- append and verify behavior;
- persistence across audit-log instances;
- detection of in-place tampering;
- detection of non-canonical byte rewriting;
- detection of chain-continuity failure after corruption.

These tests support the claim that the audit layer is designed for tamper evidence, not merely event collection.

8.1 Audit Boundary

The audit tests do not prove that every possible deployment is secure. They support a narrower claim: the reviewed audit mechanism contains unit-level coverage for append-only continuity, canonical serialization, persistence, and tamper detection. This is appropriate supporting evidence for a governance-verification baseline.

9. Reproducibility and Integrity

The evidence-pass manifests support reproducibility and package integrity. The reviewed manifest structures include fields for bundle identifiers, bundle filenames, bundle kinds, SHA-256 and BLAKE2b-256 checksums, bundle sizes, schema versions, packaging timestamps, phase identifiers, execution sequences, freeze anchors, resource envelopes, endpoint configuration, authorization status, test surface, and disciplinary invariants.

This metadata supports the ability to identify which evidence bundle was used, how it was packaged, what phase it belongs to, and what integrity anchors accompany it.

9.1 Hardware Characterization Summaries

The VRAM summaries are marked informational and out-of-band. They characterize hardware execution constraints for local model operation, including measured peak memory use and elapsed time for three configurations:

Configuration	Peak GB	Baseline GB	Elapsed seconds	Failed
fp16	7.76	1.02	409.5	false
bnb-4bit	7.77	0.84	17.8	false
ollama-q4-k-m-reference	5.66	0.68	9.3	false

The summaries identify an effective VRAM ceiling of 6.95 GB and mark the characterization as informational. This evidence should be treated as engineering context rather than as proof of the central governance claim.

10. Experimental Validation Context

The ARPG-AI experimental engineering validation note defines a research specification for local or Ollama-compatible AI systems. It proposes measuring operational propagation geometry, attention accessibility, recursive memory geometry, curvature regulation, and admissibility topology around an underlying model without modifying model weights.

For this paper, the ARPG note is relevant only as experimental context. It supports a future direction in which governance and measurement may be integrated more deeply. It is not used as primary evidence that CNX enforces authority separation.

The correct relationship is:

- CNX governs what may execute.
- ARPG-style measurement may later help evaluate how reasoning or propagation behaves before execution.

This paper does not require ARPG validation to establish the CNX governance-verification claim.

11. Discussion

The reviewed package supports a disciplined publication claim: CNX can be described as a governance-first authority infrastructure with explicit verification artifacts. The strongest aspect of the package is not conceptual novelty alone. It is the combination of:

- governance invariants;
- conformance criteria;
- mediated orchestration;
- explicit execution authorization;
- authority-separation results;
- audit-log validation;
- reproducibility manifests.

Together, these artifacts make the architecture inspectable. They also define a useful boundary between what has been shown and what remains future work.

11.1 Why Authority Separation Matters

Agentic systems often focus on tool execution. CNX shifts attention to authority. This is a meaningful distinction because operational harm may occur even when no traditional "tool call" is involved. Ranking, denial, blacklisting, escalation, summarization, routing, and publication decisions can all produce consequences.

The Phase 4 report is important because it tests categories where the distinction matters:

- allowed measurement and review capabilities;
- restricted publication-facing capabilities;
- prohibited capabilities involving legitimacy scoring, group suppression, person ranking, and access denial;
- invalid requests.

The result shows that the system preserved the expected boundary among these categories.

11.2 Why Conformance Must Be Invariant-Based

Ordinary functional tests can show that a system works when used correctly. Governance systems must also show that invalid actions fail correctly. The reviewed conformance specification therefore emphasizes violations: unknown identity, suspended identity, invalid receiver, direct communication, payload tampering, invalid signature, replay attack, ledger tampering, missing ledger entry, policy failure, and gateway failure.

This is the correct stance for governance verification.

11.3 Relation to High-Security Domains

Financial, governmental, infrastructure, and other high-security domains require stronger evidence than general AI capability demonstrations. They require identity, policy, audit, reproducibility, and fail-secure behavior. The reviewed package does not certify CNX for such domains, but it establishes a relevant evidence baseline.

The appropriate claim is:

> CNX provides a governance-verification architecture that is relevant to high-security domains because it treats authority as an explicit, testable control surface.

The inappropriate claim would be:

> CNX is already certified for financial, governmental, or critical infrastructure deployment.

12. Limitations

This paper has several limitations.

First, the evidence package is internally generated. External third-party replication remains necessary for stronger validation.

Second, the Phase 4 authority-separation evaluation contains ten request checks. This is sufficient for a baseline report but not for exhaustive adversarial coverage.

Third, audit-log tests validate specific tamper-evident properties at the unit-test level. They do not by themselves establish full deployment security.

Fourth, hardware/VRAM summaries are informational. They support engineering reproducibility but are not central governance evidence.

Fifth, ARPG-AI validation remains exploratory in this paper. It is not used to prove the CNX authority-separation claim.

Sixth, Tier 3 topics are intentionally excluded. Blockchain, crypto, energy systems, physical actuation, and distributed orchestration should be addressed separately.

13. Future Work

Future work should include:

1. External replication of the authority-separation test set.
2. Larger adversarial request suites, including prompt injection, ambiguous delegation, stale grounding, and policy-conflict tests.
3. Deployment studies using real organizational policies.
4. Integration of conformance testing into continuous integration pipelines.
5. Expanded audit-chain verification, including operational ledger recovery and cross-system replay.
6. Formal comparison with existing agent harnesses, gateway frameworks, and policy engines.
7. Separate papers for blockchain/crypto applicability, energy-system actuation, and ARPG-based propagation measurement.

14. Conclusion

This paper presents a governance-verification baseline for authority-separated AI execution in the CNX framework. The reviewed evidence supports a narrow but important claim: AI-derived reasoning outputs can be prevented from automatically acquiring operational authority when execution is mediated by identity, policy, gateway mediation, integrity validation, audit logging, and conformance criteria.

The strongest quantitative evidence is the Phase 4 authority-separation report, where ten request checks preserved expected mappings across allowed, restricted, prohibited, and invalid categories, with no violations and with `authority_separation_holds: true`. Supporting artifacts define conformance tests, governed orchestration, audit-log validation, evidence manifests, and reproducibility context.

The result is not a universal safety proof. It is a production-quality publication baseline for a governance-first architecture in which intelligence, capability, permission, and authority remain distinct.

Appendix A. Evidence Package Summary

Tier	Artifact type	Role in paper
Tier 1	Baseline state	Establishes pre-upgrade governance context
Tier 1	Conformance specification	Defines invariants and test categories
Tier 1	Conformance suite references	Supports developer-bundle traceability

Tier	Artifact type	Role in paper
Tier 1	Workflow protocol	Supports disciplined orchestration
Tier 1	Agent orchestration specification	Defines gateway mediation and lifecycle control
Tier 2	Phase 4 authority report	Primary quantitative evidence
Tier 2	Audit-log test	Supporting audit evidence
Tier 2	Manifests	Reproducibility and integrity support
Tier 2	ARPG-AI validation note	Experimental context only
Tier 2	VRAM summaries	Engineering characterization only

Appendix B. Claim Boundary

Supported:

- CNX defines authority as distinct from intelligence.
- CNX conformance can be expressed through identity, authority, mediation, and integrity validity.
- The reviewed Phase 4 report supports authority separation across a small structured request set.
- Audit-log tests support tamper-evident observability mechanisms.
- Manifests support evidence-package integrity and reproducibility.

Not supported by this paper:

- universal AI safety;
- truth detection;
- full adversarial security;
- external certification;
- production deployment in regulated environments;
- physical-system actuation validation;
- blockchain or crypto deployment readiness.

References

Silva, I. (2026). *From Agent Harnesses to Authority Infrastructure: CNX as Governed Capability Execution for Model-Independent AI Systems*. Carlonoscopen Journal of Coherence Intelligence, 1(15). https://doi.org/10.5281/zenodo.20694341

Silva, I. (2026). *Governance Verification for Authority-Separated AI Execution: Conformance, Audit, and Reproducibility Evidence from the CNX Framework*. Carlonoscopen Journal of Coherence Intelligence. https://doi.org/10.5281/zenodo.20706623