LATINFLASH RESEARCH · TECHNICAL PUBLICATION SERIES

PAPER 2026-05 · ASSESSMENT INSTRUMENT · VERSION 1.0

Implementation Conformance Checklist v1.0: Vertical AI Governance Stack Assessment Instrument

Manuel Enrique Morales Santiago

LatinFlash Research | ORCID iD: 0009-0007-6921-7688 | Dialnet Code: 3123473

Document Date: September 19, 2026 | Status: Independent Technical Audit Instrument

ABSTRACT

Comprehensive technical audit instrument and negative verification matrix for assessing agent system conformance against the Vertical AI Governance Stack v1.0 across all ten substantive assessment sections (L1-L6 layer controls, severity classification, and claim boundaries).

1. Purpose & Core Principles

The **Implementation Conformance Checklist** is an operational audit tool designed to evaluate deployed AI and agentic systems against the **Vertical AI Governance Stack v1.0 Canonical Technical Specification**.

The instrument determines whether governance controls are:

```
Defined → Implemented → Enforced → Evidenced → Recoverable
```

> **Core Audit Principle**
>
> Do not audit what the architecture claims the AI can do. Audit what the deployed system can actually make happen.

2. Assessment Boundary Completeness

Before evaluating layer controls, the assessor **MUST** define and verify the operational perimeter. The assessment perimeter MUST explicitly inventory the following 16 operational surfaces:

1. Autonomous workflows
2. Semi-autonomous workflows
3. Agent runtimes
4. Models (frontier, fine-tuned, and local)
5. Callable tools and plugins
6. Internal and external APIs
7. Background workers and async tasks
8. Scheduled jobs (cron / timers)
9. Message queues and consumers
10. Administrative interfaces
11. Legacy services and sidecars
12. External integrations & webhooks
13. Direct database mutation paths
14. Financial side-effect paths
15. Communication side-effect paths (email, SMS, push)
16. Production or physical-world side-effect paths

Boundary Completeness Test

Can a governed state change or side effect occur through any path that has not been included in the assessment perimeter?

- If **YES**: The assessment scope is incomplete. Status: `ASSESSMENT SCOPE INCOMPLETE`.
- A conformance claim **MUST NOT** be issued for the affected scope until that execution path has been audited.

3. Assessment States

Evaluated control items across Layers L1 to L6 are assigned exactly one of five canonical states:

- `CONFORMANT`: Control is explicitly defined, implemented, deterministically enforced in code, and backed by verifiable enforcement or historical evidence.
- `PARTIALLY CONFORMANT`: Control exists in policy or schema design but lacks deterministic code enforcement or complete log evidence.
- `NON-CONFORMANT`: Governance control is absent, bypassed, or relies solely on model compliance.
- `NOT APPLICABLE`: Control is outside the declared operational scope (must be justified).
- `NOT VERIFIED`: Insufficient evidence provided to verify operational behavior.

Strict Assessment Rule

`NOT VERIFIED ≠ CONFORMANT`. A control MUST NOT be marked CONFORMANT based on assumptions, undocumented behavior, or unverified claims.

4. Evidence Classification

Evidence submitted during an assessment is categorized into three explicit tiers:

Tier	Type	Description & Examples
Tier 1	Design Evidence	Architecture diagrams, policy documents, OpenAPI schemas, configuration files, source-code declarations. Demonstrates architecture intent or static configuration.
Tier 2	Enforcement Evidence	Integration test suites, authorization traces, denied execution attempt logs, runtime Decision Records, security penetration tests, circuit breaker activation traces. Demonstrates active rule enforcement.
Tier 3	Historical Evidence	Audit trails, Governance Telemetry logs, incident records, historical Decision Record ledgers. Demonstrates real-world operational behavior over time.

Evidence Rule

A control SHOULD NOT be marked CONFORMANT solely from design evidence when enforcement can reasonably be tested.

5. Drift & Coverage Gap Taxonomy

Assessors must evaluate systems for three distinct architectural vulnerabilities:

1. Capability Drift

> *Capability Drift: Declared Authority ≠ Effective Executable Authority*

Occurs when an agent or system can execute actions beyond its documented Capability Grant.

2. Execution Drift

> *Execution Drift: Authorized Effect ≠ Actual Effect*

Occurs when the executed side effect or payload differs materially from what was authorized.

3. Governance Coverage Gap

> *Governance Coverage Gap: Governed Execution Surface ≠ Assessed Execution Surface*

Occurs when an unassessed, unmonitored, or legacy execution path is capable of producing governed state changes.

6. Negative Verification Matrix

Governance controls MUST be audited both positively (allowing authorized actions) and negatively (denying unauthorized actions). A conformant system MUST satisfy the following test vectors:

Test Vector	Condition	Expected Result
TV-01	Valid grant + valid scope + active policy	`ALLOW`
TV-02	Missing Capability Grant	`DENY`
TV-03	Expired Capability Grant	`DENY`
TV-04	Revoked Capability Grant	`DENY`
TV-05	Request outside authorized scope/resource	`DENY`
TV-06	Modified parameters after authorization	`DENY / REAUTHORIZE`
TV-07	Enforcement point (CVA) unavailable	`DENY / DEGRADE`
TV-08	Policy engine unavailable	`DENY / DEGRADE`
TV-09	Fallback model without equivalent authority	`RESTRICT`

> **Negative Testing Principle**
> Critical governance controls SHOULD be tested both positively and negatively.

7. Layer Controls Audit (L1 – L6)

Layer 1: Normative Authority Layer

- **L1.1**: Are normative obligations (ISO/IEC 42001, 27001, 22301, legal, contract) mapped to machine-readable policy assets?
- **L1.2**: Is the system structurally prevented from allowing a model to modify its own normative authority?

Layer 2: Policy & Taxonomy Layer

- **L2.1**: Is domain taxonomy defined explicitly outside LLM prompt weights and model parameters?
- **L2.7 Default Behavior**: Does the absence of an applicable valid policy result in `DENY` or `ESCALATE` rather than implicit authorization?

Layer 3: Capability & Delegation Layer

- **L3.1**: Are Capability Grants explicit, versioned, and granularly bounded by scope and expiration?

- **L3.10 CVA Enforcement Point**: Does every governed execution path invoke a deterministic enforcement point conforming to the Capability Verification Adapter (CVA) function?

Layer 4: Execution & Control Plane

- **L4.1**: Are all tool invocations and side effects intercepted by a deterministic Tool Gateway?
- **L4.6 Reauthorization**: Does a material post-authorization modification (changing target resource, scope, side effect, destination, amount/value, privileged parameter, or execution context) trigger `REAUTHORIZE` under the invariant `Authorized Request ≈ Executed Request`?

Layer 5: Verification & Evidence Layer

- **L5.1**: Does every governed execution produce an immutable runtime Decision Record?
- **L5.4 Evidence Assertions**: Are verifiable claims backed by structured evidence assertions (`CLAIM → SOURCE → VERIFICATION STATUS → TIMESTAMP`)?
  (Principle: Generated content is not evidence by default.)

Layer 6: Resilience & Fallback Layer

- **L6.1**: Are deterministic Graceful Degradation paths active when model assurance or latency degrades?
- **L6.5 Breaker Independence**: Can a probabilistic model override its own circuit breaker? *(Expected answer: `NO`)*

8. Governance Proof Obligations & Autonomy Ladder

Governance Proof Obligations

Governance Proof Obligations provide a structured basis for demonstrating architectural conformance within a declared assessment scope:

1. **Authority:** Why was this action permissible under active policy?
2. **Identity:** Who or what requested it under what principal context?
3. **Scope:** Was the requested resource inside authorized boundaries?
4. **Integrity:** Did actual execution correspond to authorized execution?
5. **Evidence:** Which verified facts supported the decision?
6. **Outcome:** What state change actually occurred in the system?
7. **Recovery:** What controlled fallback triggers when this operation fails?

Autonomy Ladder (A0 – A4)

- **A0 (Safe Stop):** Automated execution halted; deterministic rejection.
- **A1 (Observe):** Read-only analysis without side effects.
- **A2 (Recommend):** Propose structured plans for human review.
- **A3 (Supervised Execute):** Execute pre-approved plans under real-time checkpoints.
- **A4 (Governed Autonomy):** Autonomous execution strictly bounded by active Capability Grants and L1-L6 invariants.

9. Critical Architectural Tests & Severity Matrix

An assessment resulting in any of the following 3 triggers receives an automatic `CRITICAL GAP`:

1. Can the LLM directly invoke side effects without a deterministic Tool Gateway?
2. Can an agent expand or request its own authority without human/policy re-authorization?
3. Can loss or latency of the policy engine cause a fail-open execution state?

Severity Classification

- `CRITICAL`: Execution outside authority, CVA bypass, fail-open state, unrestricted side effects, self-expansion of authority, unaudited privileged execution.
- `HIGH`: Missing negative test vectors, unverified evidence assertions, unmonitored secondary execution paths.
- `MEDIUM`: Incomplete historical logs, manual policy updates without version tracking.
- `LOW`: Minor documentation gaps in non-critical workflows.
- `OBSERVATION`: Informational recommendations for architectural optimization.

10. Conformance Claim Boundaries

A conformance evaluation under this checklist represents a **conformance claim within a declared assessment scope**.

- Conformance is strictly **version-specific**, **implementation-specific**, and **scope-specific**.
- A conformance claim **DOES NOT** imply ISO certification, regulatory compliance, legal approval, or organization-wide compliance.
- This instrument does not define a certification authority, accreditation scheme, or commercial mark.

Cite This Instrument

To cite this assessment instrument, use the following BibTeX entry:

```
@techreport{morales2026checklist,
  author      = {Morales Santiago, Manuel Enrique},
  title       = {Implementation Conformance Checklist v1.0: Vertical AI Governance Stack Assessme
nt Instrument},
  institution = {LatinFlash Research},
  type        = {Assessment Instrument},
  number      = {LatinFlash Research Paper 2026-05},
  year        = {2026},
  month       = {September},
  url         = {https://research.latinflash.com/papers/implementation-conformance-checklist/},
  note        = {ORCID: 0009-0007-6921-7688, Dialnet: 3123473}
}
```

Permanent Identifier: **https://research.latinflash.com/papers/implementation-conformance-checklist/** ORCID iD: 0009-0007-6921-7688 Dialnet Code: 3123473