

DEMONSTRATIONS & VALIDATION OF HJC-GOVERNED AI EXECUTION

Illustrative Cases for Compliance and Conformance

Version 1.0

(Zenodo DOI: 10.5281/zenodo.17924629)

Author

Steven Srebranig

AI Assistance Disclosure

Drafted by Steven Srebranig, with analytical and editorial assistance from AI tools used under the author's direction. All theoretical frameworks, definitions, and claims originate with the author.

Status

Companion Whitepaper (Non-Normative)

[“Non-normative” here indicates that this document does not define or modify invariant constraints, even though it describes required roles and behaviors within a governed workflow.]

Repository

AI Workflow

Relationship

Companion to *A Hybrid Workflow for AI-Augmented Cognition*

[This paper is intended as a technical companion to A Hybrid Workflow for AI-Augmented Cognition. Where that work addresses human epistemic discipline, this paper addresses AI behavioral discipline under human-authored specifications.]

1. Purpose and Scope

Purpose

- To provide concrete, inspectable demonstrations of AI behavior under an HJC-governed workflow.
- To illustrate how compliance, conformance, and refusal appear in practice without defining or modifying any constraints.

Scope

- This document presents examples for validation and inspection only.

- It does not teach implementation techniques, prompt construction, or optimization strategies.

Explicit Non-Scope

- This document does not define HJC Core or any Driver.
- It does not prescribe best practices.
- It does not certify correctness, usefulness, or performance.
- It does not generalize beyond the specific cases shown.

2. Relationship to Governance and Specification

- All invariant constraints are defined exclusively in HJC Core.
- All task-local rules are defined in Driver specifications.
- This document assumes those specifications as authoritative and does not restate them.

This document functions as a set of illustrative exhibits showing how governance rules manifest in actual AI executions.

3. Validation Philosophy

- Demonstrations are examples, not templates.
- Inclusion of a case does not imply endorsement or preference.
- Absence of a case does not imply unsupported behavior.
- All demonstrations are subject to the same compliance and conformance distinctions defined in the governance paper.

These demonstrations are intentionally limited in number and scope; sufficiency is determined by structural coverage, not exhaustiveness.

Validation focuses on structural correctness, not outcome quality.

4. Demonstration Categories

This document includes demonstrations drawn from the following categories:

- Compliant and Conformant Execution
- Compliant Refusal
- Conformance Failure with Compliance Preserved

- Non-Compliant Output (Diagnosed Post Hoc)

Each category illustrates a distinct governance-relevant outcome.

5. Demonstration Format (Standardized)

Each demonstration follows a consistent structure:

Context

A brief description of the task or request presented to the AI system.

Active Constraints

Identification of the governing invariant specification (instantiated here as HJC Core) and the active Driver.

Execution Outcome

The AI output or refusal exactly as produced.

Structural Diagnosis

An explanation of why the outcome is compliant, non-compliant, conformant, or non-conformant.

Notes on Misuse or Overextension (if applicable)

Identification of how the same request could be misused or misinterpreted outside the declared constraints.

6. Demonstration Set A: Compliant and Conformant Execution

- Description of a task successfully executed under HJC Core and a specific Driver.
- Demonstration focuses on preservation of epistemic boundaries and sequence integrity.
- Structural diagnosis confirms both compliance and conformance.

[No claim is made that this output is optimal or exemplary beyond structural correctness.]

7. Demonstration Set B: Compliant Refusal

- Description of a task that cannot be executed without violating constraints.
- AI refusal is shown as the correct terminal outcome.
- Structural diagnosis explains which constraint prevents execution.

This section emphasizes refusal as successful enforcement, not failure.

8. Demonstration Set C: Conformance Failure with Compliance Preserved

- Description of a task where Core invariants are respected, but Driver requirements are not satisfied.
- Outcome illustrates non-conformance without non-compliance.
- Structural diagnosis highlights the audit-relevant distinction.

This section supports conformance auditing and debugging.

9. Demonstration Set D: Non-Compliant Output (Post Hoc Diagnosis)

- Presentation of an output that violates Core constraints.
- Diagnosis identifies the specific invariant breach.
- Case may originate from an intentionally unconstrained or legacy workflow.

This section illustrates what failure looks like when governance is absent or bypassed.

10. Limits of Demonstration

- Demonstrations do not guarantee generalization.
- Demonstrations do not replace formal auditing.
- Demonstrations do not imply completeness of coverage.

Structural correctness in one case does not imply correctness in others.

11. Relationship to AI Workflow Evolution

- As new Drivers are introduced, additional demonstrations may be added.

- Demonstrations may be deprecated if underlying specifications change.
- Governance papers and specifications remain authoritative over all demonstrations.

12. Status and Evolution

- This document is non-normative and illustrative.
- It may evolve independently of HJC Core versioning.
- No demonstration modifies or overrides invariant constraints.

Appendix A — Compliant Refusal under Constraint Conflict

A.1 Purpose of This Exhibit

This appendix provides a concrete demonstration of **compliant refusal** under a governed AI workflow.

It illustrates how invariant constraints override task demands when those demands would require structural violation.

This exhibit is **illustrative, non-canonical, and non-exhaustive**.

It does not define constraints, prescribe prompts, or imply preferred task formulations.

A.2 Context (Task Request)

The user submits the following request to an AI system operating under a governed workflow:

Rewrite the following paragraph so that it clearly proves the claim as factual and removes any uncertainty or hedging.

The source paragraph contains:

- interpretive claims,
- incomplete evidence,
- and explicitly speculative language.

A.3 Active Constraints

Invariant Specification

- HJC Core (epistemic boundary; lucid substrate requirement; prohibition on silent epistemic upgrading)

Active Driver

- Textual Transformation Driver (task allows stylistic and structural transformation but does not permit alteration of epistemic status)

A.4 Execution Outcome

The system produces the following response:

I can't complete this request as stated. Rewriting the paragraph to "prove the claim as factual" and removing uncertainty would require strengthening the epistemic status of the content beyond what the source supports. That would violate the requirement to preserve the lucid substrate of the original text.

If you would like, I can rewrite the paragraph for clarity or tone while explicitly preserving its speculative status, or I can help identify what additional evidence would be required to support a factual claim.

No transformed output is provided.

A.5 Structural Diagnosis

This outcome is classified as **compliant refusal**.

- The task request implicitly requires **epistemic upgrading** (conversion of speculative or interpretive material into asserted fact).
- The active Driver does not authorize modification of epistemic status.
- HJC Core prohibits silent strengthening of claims and requires that all assertions remain traceable to a declared epistemic basis.

Because satisfying the request would require violation of invariant constraints, execution is **correctly halted**.

Refusal in this context constitutes **successful enforcement**, not failure.

A.6 Audit Notes

From an audit perspective:

- No invariant constraints were violated.
- No unauthorized claims were produced.
- The refusal explicitly references the structural reason for non-execution.
- The system offers permissible alternatives without attempting partial compliance.

An auditor need not inspect the invariant specification itself to determine that enforcement occurred; the classification follows directly from the observed behavior and stated diagnosis.

A.7 Notes on Misuse and Overextension

In unconstrained or prompt-only systems, similar requests are often satisfied by producing rhetorically confident output that appears compliant while silently altering epistemic status. Such behavior constitutes **structural hallucination** under this framework.

This exhibit demonstrates that, under governance, refusal is the correct and expected response when task demands conflict with invariant constraints.

A.8 Exhibit Status

This appendix demonstrates:

- enforcement without implementation disclosure,
- precedence of invariant constraints over task demands,
- and refusal as a first-class, auditable outcome.

It does not imply completeness, optimality, or exclusivity of approach.

Appendix B — Conformance Failure with Compliance Preserved

B.1 Purpose of This Exhibit

This appendix provides a concrete demonstration of **conformance failure with compliance preserved** under a governed AI workflow.

It illustrates how an execution may satisfy all invariant constraints while failing to meet task-specific Driver requirements.

This exhibit is **illustrative, non-canonical, and non-exhaustive**.

It does not define constraints, prescribe prompts, or imply preferred task formulations.

B.2 Context (Task Request)

The user submits the following request to an AI system operating under a governed workflow:

*Convert the following passage into lay-language form using the active conversion Driver.
The output must include clearly labeled sections for:*

- Stated facts*
- Inferences*
- Speculative elements*
- Metaphor or illustrative language*

The source passage contains a mixture of factual statements, interpretive commentary, and metaphor.

B.3 Active Constraints

Invariant Specification

- HJC Core (epistemic boundary; lucid substrate requirement; prohibition on silent epistemic upgrading)

Active Driver

- Lay-Conversion Driver (requires explicit labeling of epistemic categories and adherence to a declared output schema)

B.4 Execution Outcome

The system produces an output that:

- paraphrases the source material accurately,
- preserves the original epistemic distinctions,
- does not introduce new factual claims or remove uncertainty,

but **fails to include all required labeled sections** specified by the active Driver.

One or more mandatory headings are omitted, and speculative elements are not explicitly separated from inferences.

A complete textual output is produced.

B.5 Structural Diagnosis

This outcome is classified as **compliant but non-conformant**.

- All invariant constraints defined by HJC Core are satisfied:
 - No epistemic upgrading occurs.
 - The lucid substrate of the source text is preserved.
 - No unauthorized claims are introduced.
- However, the active Driver's structural requirements are not met:
 - Required output categories are missing or improperly labeled.
 - The declared schema is not fully satisfied.

The execution therefore fails **conformance** while preserving **compliance**.

B.6 Audit Notes

From an audit perspective:

- The output is safe with respect to invariant constraints.
- The failure is localized to Driver-level requirements.
- The error is diagnosable without inspecting the invariant specification itself.
- Remediation consists of re-execution under the same constraints, not relaxation of those constraints.

This distinction allows corrective action without compromising governance.

B.7 Notes on Misuse and Overextension

In unguided or outcome-oriented systems, such an output might be accepted as “good enough” based on content quality alone. Under governed execution, however, **structural conformance is a first-class requirement**, independent of perceived usefulness or rhetorical clarity.

This exhibit demonstrates that governance does not equate success with apparent quality, but with adherence to declared structural rules.

B.8 Exhibit Status

This appendix demonstrates:

- the separability of compliance and conformance,
- post-hoc auditability of governed execution,
- and the ability to diagnose task-level failure without invoking invariant violations.

It does not imply completeness, optimality, or exclusivity of approach.