

From TEVV Evidence to Bounded Findings

Illustrative Records and a Physical AI Worked Example

Supporting note to the ODA3 Institute public comment on NIST AI 200-2 ipd, “The TEVV-Athlon Framework for Evaluating AI Systems”

Prepared by	ODA3 Institute
Responsible signatory	Mukesh Kumar Singh, Founder and Director
Contact	contact@oda3.org
Date	14 October 2026
Status	Subsequent supporting material. It supplements, and does not revise, the comment ODA3 Institute submitted on 5 September 2026. It has not been submitted to, reviewed by, or adopted by NIST.

About ODA3 Institute. ODA3 Institute is an applied research and advisory organization, founded in 2026 and based in India. It works on how AI governance requirements translate into evidence about real-world system behavior.

1. Purpose

Our submitted comment made seven recommendations. They concern how evidence produced in a TEVV-Athlon remains attributable to the evaluated system, and how conclusions remain bounded by what that evidence establishes.

This note develops the submitted recommendations through proposed implementation details and an illustrative example. It:

- organizes recommendations 1–5 into three core asks;
- shows a record format that would make them operable;
- works through one illustrative physical AI example;
- offers short candidate wording keyed to line numbers in the draft.

Line and page references are to NIST AI 200-2 ipd (Initial Public Draft, August 2026) [1].

2. Three core asks

Ask A — Evidence traceability, with proportionate integrity checks (original recommendation 1).

- *What the draft already says.* Each Block must specify the information required as evidence (lines 268–270). The Toolbox includes the method for aggregating collected evidence (lines 297–300).
- *Our ask.* Evidence relied on in Synthesize & Interrogate should remain traceable to:
 - the Block, Event and Tool that produced it;
 - the evaluated system state and execution context;
 - any material filtering, scoring, transformation or aggregation applied before synthesis.
- *Integrity.* Traceability alone does not establish authenticity, completeness for the proposition examined, or freedom from material alteration. Where material to the decision, evaluators should assess these properties and document unresolved limitations.

Ask B — Bind evidence to the evaluated system state, with a reassessment trigger (recommendation 2).

- *What the draft already says.*
 - Lifecycle stage influences assessment design (lines 231–233).
 - Validation includes asking whether results generalize beyond the original testing conditions (line 541).
 - Validation is ongoing (lines 543–546).
- *Our ask.*
 - Execution records should identify the material system state actually tested. The level of detail should be proportionate to risk, consistent with lines 441–442.
 - A material change to that state or to its operating conditions should prompt a check of whether earlier evidence still applies before it is reused.

Ask C — Explicit evidence-to-conclusion reasoning, distinct outcomes, and stated limits (recommendations 3, 4 and 5).

- *What the draft already says.*
 - Synthesis works back from Toolbox data to Blocks and goals (lines 306–312), and it informs organizational decisions (lines 313–315).
 - The draft cautions against drawing strong conclusions from weak evidence (lines 519–520).
 - It asks evaluators to document assumptions and limitations (lines 523–525; Appendix D, Table 6, p. 27).
- *Our ask.* Operationalize this inside Stage 4, not as an additional stage, in three parts:
 - **Inference record.** For each material finding, record the evidence relied on, the proposition it is offered to support, material assumptions, plausible alternative explanations, and limits on interpretation.
 - **Distinct outcomes.** Do not collapse materially different outcomes into one category. Distinguish:
 - behavior observed;
 - behavior tested for and not observed;
 - procedure not completed;
 - evidence unavailable;
 - explicit refusal or rejection;
 - apparent discrepancy;
 - question left unresolved.

We do not propose a universal taxonomy.

- **Reporting boundary.** State what the evaluation does not establish.

Recommendations 6 and 7. These stand as submitted. On recommendation 6, a more precise statement of our observation is this: the illustrative toolbox in Appendices A–C does not explicitly demonstrate evaluation of actuator behavior, physical state transitions, or intervention and safe-state mechanisms. The example in Section 4 illustrates the point. This note does not expand recommendation 7 (agentic systems).

3. Record format

The labels in Ask C are not mutually exclusive. A single run can complete as planned, produce one observation, lack another, and support only a partial conclusion. We therefore suggest recording four dimensions separately, linked by compact identifiers to the plan and the evidence.

Record header (one per Event execution)

Field	Content
Execution ID	Unique identifier for the run
Evaluated state (S-ID)	Pointer to the recorded material state: model or system version, relevant configuration, enabled capabilities, operating environment, material dependencies, execution period
Block / Event / Tools	Identifiers from the TEVV-Athlon plan
Evidence artifacts (A-IDs)	Each artifact relied upon, with its source, time base, and integrity and completeness status
Transformations	Filtering, scoring or aggregation applied before synthesis
Proposition and inference basis	Specific proposition examined; material assumptions; plausible alternative explanations; inference limits. Record separately for each material proposition, here or in a linked inference record where a finding draws on several executions.

Primary record (one row per execution): **Execution status** · **Observations** · **Evidence availability** · **Bounded conclusion**.

The bounded conclusion states three things: what is supported, for which evaluated state, and what is not established.

4. Illustrative worked example: operator emergency override

Illustrative and unvalidated. This scenario was not executed and no data was collected. All observations and check results in the table are hypothetical, not measured findings. It shows recording logic only. Operator override testing is a familiar safety-testing scenario. It is used here because it separates software-level and physical-level evidence clearly, not as a novel test.

Setup (illustrative)

- **System.** An AI-enabled mobile robotic platform in a controlled test area.

- **Characteristic.** Safe. The draft defines Safe as not leading, under defined conditions, to a state in which human life, health, property or the environment is endangered (lines 357–358).
- **Block, Event and Tools.**
 - Block: Override response.
 - Event: controlled operator override exercise.
 - Tools: test procedure, command and acknowledgement log, actuator telemetry, independent platform-motion measurement (for example, calibrated external position tracking), and a common time reference across sources.
- **Predefined criterion.** After an operator override, the system:
 - i. acknowledges the override;
 - ii. rejects new motion commands;
 - iii. reaches the test-defined safe state within a predefined interval T , measured from verified receipt of the override. For this example, the test-defined safe state is platform speed below a predefined tolerance for a predefined observation duration. The plan must specify T , the tolerance, duration, sampling adequacy, and measurement uncertainty before execution; no universal values are proposed. Stationary does not establish safety in every context.
- **Evaluated state.** All four executions use one recorded state, S1.

Exec.	Execution status	Observations	Evidence availability	Bounded conclusion
X1	Completed as planned	Override acknowledged; a new motion command issued after override receipt and rejected; test-defined safe state reached within T	Log, actuator telemetry, independent motion measurement and time reference available; relevant integrity and completeness checks stipulated as passed	Supports the specified response in S1 under the tested conditions. Does not establish behavior in other states, loads, surfaces or override paths.
X2	Completed as planned	Override acknowledged	Log available; actuator telemetry and independent motion measurement not captured; command-rejection evidence not supplied	Establishes acknowledgement (i) only. Criteria (ii) and (iii) remain unresolved; neither is established as met or failed.
X3	Completed as planned	Override acknowledged; test-defined safe state not reached within T	Log, actuator telemetry, independent motion measurement and verified timing available; relevant integrity and completeness checks stipulated as passed	Supports a bounded adverse finding against criterion (iii) in S1. Does not establish the cause.
X4	Completed as planned	New motion command rejected; platform motion recorded by independent measurement at timestamps nominally after rejection	Log and independent motion measurement available; cross-source time alignment not verified	Apparent discrepancy, not yet contradictory evidence. The causal conclusion is unresolved.

Reading X4. The apparent ordering of motion after rejection could have several explanations:

- a command issued before the override was still executing;
- inertia or coasting;
- another control path not covered by the rejection;
- misaligned timestamps between log sources.

Resolving this requires evidence that distinguishes among them, such as:

- verified clock alignment;
- a command-queue trace;
- an estimate of expected stopping dynamics;
- an inventory of control paths.

Until then, the record states the discrepancy and the open alternatives.

What a collapsed summary would lose. A narrowly labeled count of recorded acknowledgements or runs without a confirmed adverse finding can be accurate. It becomes misleading when treated as evidence of overall override effectiveness: missing evidence in X2 is not success, and X4 remains unresolved. Software-level acknowledgement or rejection alone does not establish physical stopping performance. This is why material aggregation must preserve outcome distinctions and the propositions each observation supports.

5. Candidate wording for NIST's consideration

These are offered as drafting aids, at NIST's discretion. Request items refer to the six numbered items in NIST's call for input [2]:

1. TEVV terms and definitions
2. Flexibility, scope and applicability
3. TEVV activities not adequately addressed
4. Novel or emerging AI systems
5. Aspects warranting clarification, revision or expansion
6. Additional concepts, examples or supporting materials

Location	Candidate addition	Ask	Items
§2.3, after line 304	"Evaluators should retain enough information to trace evidence relied on in synthesis to the Block, Event, Tool, evaluated system state, and execution context that produced it, together with any material filtering, scoring, transformation, or aggregation. Where material to the decision, evaluators should assess source authenticity, completeness for the proposition examined, and whether material alteration occurred, documenting unresolved limitations. The level of detail should be proportionate to risk."	A	3, 5, 6
§2.4, after line 312	"For each material finding, evaluators should identify the evidence relied on, the proposition it is offered to support, material assumptions, plausible alternative explanations, and limits on interpretation. Results should distinguish behavior observed, behavior tested for and not observed, procedures that did not complete, unavailable evidence, explicit refusals or rejections, apparent discrepancies, and questions the evidence could not resolve. A fixed taxonomy is not required."	C	3, 5, 6
§2.4, after line 315	"Reports should state what the evaluation does not establish, such as untested system states, configurations, environments, populations, or time periods."	C	5, 6
§4.5, after line 546	"Where earlier results are reused to inform a decision, evaluators should check whether the system state and operating conditions remain within the evaluated boundary or whether additional evidence justifies reuse. A material change may warrant targeted reassessment before reuse."	B	2, 5
§4.3.2, after line 502	"Where an AI system can initiate or influence physical action, realistic testing may also examine intervention and override behavior, physical state transitions, and safe-state behavior, recording software-level and physical-level evidence separately."	Rec. 6	2, 3, 4

6. What this note does not establish

- The example was not executed. It provides no evidence about any system.
- We make no claim that the record format or candidate wording has been validated in practice, or that NIST will adopt any of it.
- We do not claim that the TEVV-Athlon architecture cannot accommodate physical AI. NIST describes the framework as intended for a wide variety of AI technologies [2].
- The note does not address agentic-system evaluation beyond the submitted comment. It does not respond to other commenters' proposals.

References

- [1] P. J. Phillips, T. Jensen, P. Hall, R. Amironesei, Y.-Y. Choong, C. Greenberg, K. K. Greene. *The TEVV-Athlon Framework for Evaluating AI Systems*, NIST AI 200-2 ipd (Initial Public Draft), August 2026. <https://doi.org/10.6028/NIST.AI.200-2.ipd>
- [2] NIST. *The TEVV-Athlon Framework for Evaluating AI Systems: Input Sought on Initial Public Draft of NIST AI 200-2*, announcement of August 7, 2026. <https://www.nist.gov/artificial-intelligence/ai-research/tevv-athlon-framework-evaluating-ai-systems>
- [3] ODA3 Institute. Public Comment on NIST AI 200-2 ipd: Comments on Evidence Traceability, System-State Binding, and Bounded Interpretation, submitted to TEVV-Athlon@nist.gov, 5 September 2026.