

Operational Incompleteness and Evidence-Bounded Cyber Risk Intelligence

Versioned Assessment, Scenario Counterfactuals, and Conditional Assurance

Hassan Nasreddine
Fortisec Research
ORCID: 0009-0001-0339-0978

Research white paper, implementation-audited r3 • 18 October 2026

Abstract

A cyber-security assertion can be true in an examined architecture without being justified for the deployed system. This white paper presents Fortisec Operational Risk Intelligence (ORI) as an implementation-grounded, evidence-bounded assessment framework and delineates a stand-alone experimental verification layer. The existing ORI v6 assessment engine combines declared control posture, risk weights, evidence checklists, dependency consistency, and telemetry observations. These numerical outputs are engineering heuristics, not certified security probabilities, quantum-entropy bounds, or proofs of absence of attack paths. An experimental finite-graph kernel checks whether every source-to-target path in each explicitly supplied compatible graph traverses a designated control, and distinguishes established, refuted, unresolved, and inconsistent claims relative to that model family. The research branch exposes method-version metadata for new assessments, separates evidence declarations from independent authentication, evaluates expiry and invalidation metadata, and tracks the dependence of findings on evidence items. The research branch also recomputes deterministic one-answer scenarios and reports confidence-score differences in percentage points. A reproducible GitHub Actions suite has passed, including an exhaustive comparison over 4,096 four-node graph configurations against an independent reachability oracle. Neither global environment coverage, authenticity of submitted evidence, calibration of scoring weights, nor product-level integration is established. The documented engineering contribution is an implementation-specific synthesis and reproducible validation of established assurance ideas with explicit validity boundaries, not a new quantum-to-cyber theorem or new graph-search algorithm.

Keywords: cyber risk assurance; partial observability; evidence provenance; adversarial uncertainty; model checking; conditional claims; AI-assisted security operations.

1 Introduction: a security claim is not an alert count

Security tools routinely collect network flows, authentication events, API gateway records, cloud configurations, vulnerability inventories, and endpoint logs. Each is valuable; none automatically provides a complete description of an organization’s operational attack surface. The difference matters when a dashboard, analyst, or automated assistant moves from the statement “no relevant alert was recorded” to the stronger statement “the protected service has no accessible bypass.” The latter is a claim about paths and controls, not a count of events.

The challenge is neither unique to AI nor absent from existing cyber research. Formal methods analyze whether modeled systems satisfy security properties; opacity theory studies whether distinct states or trajectories can be discriminated by an observer; attack graphs model chains of attacker actions and dependencies [3, 7, 8, 4, 5, 6]. Modern AI risk guidance also recognizes that validity, reliability, transparency, and monitoring are context dependent [9]. The unresolved practical question addressed here is how to communicate the *scope of warranted conclusions* when evidence, architecture knowledge, and adversarial conditions are all imperfect.

We use *AI-native* descriptively for security operations in which AI systems meaningfully participate in interpretation, recommendation, or action. No premise is made that all such systems presume perfect observability. An AI model may summarize observations correctly while the underlying observation boundary is incomplete. Conversely, additional evidence can sometimes establish a targeted claim without requiring global knowledge of the environment.

Contribution and scope. This is a research perspective, not a new impossibility theorem, a quantitative detection benchmark, or a product validation study. Its contribution is an implementation-grounded account of an operational evidence-aware scoring engine and a precise boundary between heuristic risk inference and formal assurance. ORI production computes evidence- and coherence-conditioned assessments; a separate research branch contains a finite-graph checker, while automatic asset discovery, independent proof validation, and production integration remain unverified.

2 Prior work and the remaining operational gap

Opacity and diagnosability. In discrete-event systems, different executions can have indistinguishable observation traces. Opacity characterizes forms of observational concealment, with links to detectability, diagnosis, secrecy, and privacy [3]. The existence of observationally equivalent secure and insecure scenarios is therefore well established. We do not rename this phenomenon to claim priority.

Formal verification. Model checking evaluates formal properties over specified system models and can produce counterexamples when a property is violated [7]. Such results depend on the adequacy of the model and assumptions. A valid proof about an omitted-backend graph is not necessarily a valid proof about the operational network.

Attack paths and risk propagation. Attack-graph methods explicitly represent combinations of vulnerabilities, privileges, and reachable paths, including probabilistic approaches to assessing exposure [8]. The implemented ORI heuristics should be compared against these techniques; graph reachability is not itself a novel ORI algorithm.

Risk governance of AI. The NIST AI RMF provides a practical, lifecycle-oriented framework for managing AI risks and understanding contextual trustworthiness [9]. Our narrower topic is the logical status of a *specific cyber assurance claim*, including its dependence on observed configuration facts and trusted evidence sources.

What remains worth developing is not a new name for partial observability but an auditable interface between evidence provenance, explicitly scoped claims, falsifying configurations, and the next technically feasible check. Such an interface may be useful even where all its constituent reasoning methods are classical. Neither the aggregation of these established ideas nor the deployment of heuristic scores establishes a novel mathematical theorem. The engine described below should be evaluated against assurance cases, active diagnosis, abstraction refinement, and control-assessment systems.

3 Three distinct limits of operational knowledge

For a declared security claim S , distinguish the following sources of uncertainty.

Observation uncertainty. A real asset or event is represented in the model but is not uniquely determined by collected evidence (e.g., two backend routes yield the same WAF logs).

Structural model uncertainty. The assumed graph or transition system may omit a relevant identity, path, dependency, or integration. No finite check over the declared model can, by itself, establish that the model contains every real-world asset.

Evidence integrity uncertainty. A source can be stale, suppressed, spoofed, forged, or selectively reported. Agreement between correlated logs does not establish independence or authenticity.

These limits demand different remedies: targeted measurement, independent discovery/model reconciliation, and provenance or integrity checks. They must not be collapsed into a single arbitrary “confidence score.” Nor does reducing uncertainty necessarily reduce the actual exposure: discovering a hidden compromise may make the assessment more certain while worsening the assessed risk.

4 A minimal evidence-bounded assurance model

Let $\mathfrak{M}$ be a declared, nonempty class of operational models. A model may be a labeled directed graph, transition system, or richer policy model. Let A be a set of stated assumptions (coverage, freshness, integrity, identity mapping), and E an evidence set with provenance and collection times. A compatibility relation $M \models E, A$ means that model M respects the declared interpretation of the evidence and assumptions. The surviving possibilities are

$$\mathcal{K}(E, A) = \{M \in \mathfrak{M} : M \models E \text{ and } M \models A\}. \quad (1)$$

For a security property S , assign one of four conditional statuses:

$$\begin{aligned} \text{established} : & \quad \mathcal{K}(E, A) \neq \emptyset \text{ and } \forall M \in \mathcal{K}(E, A), M \models S, \\ \text{refuted} : & \quad \mathcal{K}(E, A) \neq \emptyset \text{ and } \forall M \in \mathcal{K}(E, A), M \models \neg S, \\ \text{unresolved} : & \quad \exists M_+, M_- \in \mathcal{K}(E, A) : M_+ \models S, M_- \models \neg S, \\ \text{inconsistent} : & \quad \mathcal{K}(E, A) = \emptyset. \end{aligned}$$

Assume classical bivalent satisfaction semantics: each admissible model satisfies exactly one of S and $\neg S$. Then these statuses are mutually exclusive and exhaustive. They are elementary semantic definitions, not original theorems. In particular, the inconsistent case must not be mistaken for a vacuously established safety claim. *Established* means established only within the declared model class and its assumptions; it does not entail global security of the real environment.

Illustrative output. A conditional assurance record should include (i) a uniquely identified claim and scope, (ii) model identifier/version, (iii) timestamped evidence and provenance, (iv) named completeness/integrity assumptions, (v) status and verification method, (vi) countermodel if unresolved, and (vii) invalidation conditions such as a topology change or an expired log window.

4.1 Evidence acquisition is a decision problem

Suppose candidate verification actions $q \in Q$ can obtain supplementary evidence at costs $c(q)$, with possible outcomes $z \in Z_q$. A useful question is whether a proposed check distinguishes currently compatible secure and insecure configurations. A check that merely adds more correlated copies of the same blind log may not help. A check that queries a separately maintained route table or tests a protected path may.

No universal optimal verification rule is claimed: useful prioritization depends on deployment costs, availability, possible interference, evidential reliability, and risk severity. If a probability distribution or adversarial uncertainty set is introduced, it must be justified independently; the existence of an uncertainty set does not provide calibrated probabilities.

5 Worked example: an API path outside WAF inspection

Claim. Every externally reachable request path to backend service B passes through an authorized inspection and authentication enforcement point W .

Observed evidence. WAF records show requests to the public route, a perimeter diagram places W in front of B , and alerts show no relevant bypass. Those observations alone do not establish the claim: they may be identical in a configuration containing an additional externally reachable route to B .

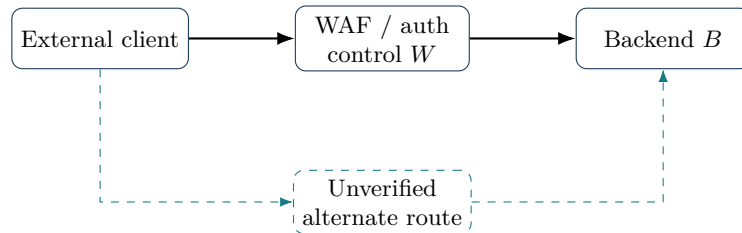


Figure 1: Illustrative secure-model route (solid) and a possible bypass route (dashed). A missing dashed edge in an inventory is not proof of its absence in the actual environment.

Compatible interpretations. Under a graph model that treats the inventory as incomplete, M_+ has only inspected external routes and satisfies S ; M_- also has an external-to-backend bypass path and violates S . With identical WAF-visible logs, the claim remains unresolved. A route-table export, gateway configuration reconciliation, separate cloud exposure discovery, or authorized path test may eliminate one interpretation. The specific result is conditional on the scope and reliability of those checks.

Adversarial evidence. If an attacker can suppress route records or manipulate an inventory service, a clean export cannot necessarily rule out M_- . Assurance must record which evidence sources are trusted and which compromises are excluded. A certification that silently presumes trustworthy logs would be misleading.

Operational decisions. The correct response to unresolved evidence is not always to declare the system insecure. It may be to avoid issuing a positive assurance, prioritize independent verification, or apply temporary restrictions until the ambiguity is resolved. The final decision depends on impact, available safeguards, and business constraints.

6 Research provenance: what the author’s mathematics does and does not establish

The present perspective is intellectually informed by the author’s mathematical work, but the domains must be separated precisely.

In *Essential Duality and Quantitative Channel Composability in Algebraic Quantum Field Theory* [10], complement-fixing channels, dual nets, and quantitative channel-order defects are characterized under precise operator-algebraic assumptions. The transferable *methodological question* is how to characterize admissible operations and composition constraints from explicit invariants. Enterprise API calls are not thereby identified with normal completely positive quantum channels, and Haag duality does not imply a cybersecurity completeness theorem.

The author’s results on block-coherence costs [11], rank-deficient noncommutative entropy estimates [12], and stability under block-separable constraints [13] illustrate careful control of assumptions, support structure, and stability bounds in their respective mathematical settings. They do not furnish an entropy formula for undetected lateral movement, a SOC confidence score, or a bound on unseen cyber assets. The energy-constrained Weyl-pair study [14] is part of the research record, but no direct technical bridge to this assurance proposal is asserted.

This distinction protects both the integrity of the mathematical papers and the credibility of the cybersecurity program. A genuine cross-domain theorem would require a specified mapping between objects and a proof preserving the relevant properties; that is not supplied here.

7 ORI v6: an implemented evidence-aware inference engine

Inspection of the ORI production-branch source at commit `cca1d088` identifies an executable assessment engine (`ori.engine.cjs`), HTTP routes (`ori.routes.cjs`), and a v6 interface (`public/ori/ori.js`). The engine normalizes answer categories, associates control-level risk weights, computes declared maturity and a separate operational-confidence score, analyzes dependency coherence, records evidence-checklist declarations, and processes certain uploaded telemetry-derived observations. These are implemented computations, not a formal verifier or proof that unknown assets have been found.

For a domain d with N_d applicable controls, define risk weights $r_i > 0$, declared control maturity $w_i \in [0, 1]$, declared-answer confidence $c_i \in [0, 1]$, evidence-strength factor $e_i \in [0, 1]$, dependency-coherence factor $k_i \in [0, 1]$, and domain applicability $a_d \in [0, 1]$. The engine calculates

$$M_d = \frac{\sum_i r_i w_i}{\sum_i r_i}, \quad OC_d = a_d \min \left\{ M_d, \frac{\sum_i r_i w_i c_i e_i k_i}{\sum_i r_i} \right\}. \quad (2)$$

This is a defined weighted heuristic. OC_d is *not* a posterior probability of security, and M_d is not a direct empirical measurement of compromise. The source implements an optional legacy mode: where evidence has not been engaged and strict assessment mode is disabled, $e_i = 1$; therefore, an assessment must state whether evidence enforcement was active. Where no evidence specification exists for a control, the engine also uses a neutral evidence factor, which does not mean the control has been independently verified.

Evidence and coherence pressure. For controls with positive claims, the implemented code calculates risk-weighted penalties of the following forms (where s_i denotes normalized claim strength):

$$BP_i = r_i s_i (1 - e_i) \log(1 / \max\{e_i, \varepsilon_i\}), \quad CP_i = r_i s_i (1 - k_i)^2 \log(1 / \max\{k_i, \varepsilon\}). \quad (3)$$

The engine sums the eligible contributions (with strict evidence treatment enabled by default in the experimental branch; legacy non-strict reporting requires explicit qualification). These are *design choices*, not corollaries of BKM coercivity or any operator-algebraic result. The logarithm increases the penalty near low evidence/coherence values; this qualitative behavior alone cannot confer quantum-theoretic validity. The parameters and thresholds require sensitivity and calibration studies.

Hidden-state hypotheses. For a named hypothesis h , ORI aggregates configured closure impacts q_j and associated evidence/coherence factors by

$$A_h = \prod_{j \in J(h)} (1 - q_j e_j k_j). \quad (4)$$

The output is a *heuristic residual-hypothesis score* for hypotheses enumerated in ORI’s question/evidence schema. It is neither a normalized probability nor a proof of logical admissibility. The research-branch engine no longer issues a `ruled_out` label from this score alone; any legacy score-based `ruled_out` label is not a proof of the nonexistence of an attack configuration. Hypotheses absent from the schema cannot be discovered by this computation. Treat low scores as prioritized investigation cues, not certificates of nonexistence.

Telemetry and provenance. ORI’s server-side routes provide endpoints for receiving specified telemetry files and applying processed observations to assessments; the engine records coverage information and can change hidden-hypothesis statuses when matching observations are present. Code presence establishes a processing path, not verified detection performance, complete source coverage, or integrity of externally supplied data. Inference outputs depend on the declared inventory, questions, evidence checklist, and input integrity; these limitations must accompany any assurance statement.

Finite-model assurance prototype (experimental branch). An isolated research branch now includes a finite, explicit-model path-assurance kernel (`ori.assurance.cjs`), separate from the released ORI scoring engine. For a declared finite collection $\mathcal{K}$ of compatible graph models, each model is checked for an external-to-backend route avoiding a designated control. The four status names encode universal or mixed satisfaction across the supplied model family, while a violating path is retained as a countermodel witness. GitHub Actions has passed the adversarial Node tests and an exhaustive 4,096-graph oracle comparison for four-node directed graphs with distinct source, target, and control. This test space is finite and limited; it does not show correctness for arbitrary input sizes or actual enterprise deployments. This is a testable prototype and is not yet connected to ORI production routes. It does not enumerate all possible real-world graphs, establish model completeness, or replace the existing heuristic engine. Future integration requires a validated fact-to-graph mapping, source provenance, temporal invalidation and adversarial test coverage.

Validation protocol. Compare ORI’s implemented heuristics with unweighted checklists and established evidence-based control-assessment baselines on the same synthetic and independently verified real-world scenarios. Separately test: monotonicity under stronger *verified* evidence, no spurious improvement under missing evidence, age decay, dependent-source conflicts, false reassurance under omitted assets, telemetry coverage, stability of rankings under weight perturbations, and consistency of source-reported claims with runtime output. No detection accuracy, calibrated confidence, or improvement over these baselines is established in this paper.

8 Research r3: scoring corrections and scenario counterfactuals

The research assessment contract carries `schemaVersion=ori-assessment/3` and `methodologyVersion=ori-v6-heuristic`. This identifies research scoring behavior separately from historical assessments and must not be treated as an in-place rescore of persisted records. The changes described in this section have passed the branch’s automated regression suite; they have not passed a release migration or calibration study.

Unknown-age evidence is not new evidence. Given required artefacts $E(q)$ and caller-declared selections

$D(q)$, the evidence-strength transform is the following implementation heuristic:

$$e_q = \min \left\{ 1, \max \left\{ 0, \sum_{j \in E(q) \cap D(q)} \max(0, s_j) \rho(\tau_j) \right\} \right\}, \quad \rho(\tau) = \begin{cases} 1, & 0 \leq \tau \leq 90, \\ 0.70, & 90 < \tau \leq 180, \\ 0.40, & 180 < \tau \leq 365, \\ 0.15, & \tau > 365 \text{ or age unknown.} \end{cases} \quad (5)$$

Here s_j is a configured artefact strength and τ_j is declared age in days. The research API no longer substitutes age zero for an omitted, null or invalid age: the last bucket is used conservatively. A missing artefact contributes zero. Where no artefact specification exists, the legacy neutral factor is one; this is *not* evidence verification. Where the strict-mode checklist is engaged and no artefact is selected, the factor is zero. An explicitly declared age of zero remains only a declaration, never independent authentication.

Bounded boundary pressure under cross-domain dependencies. The original coherence penalty divided the risk mass of unproven dependencies by the evaluated domain’s own risk mass. Since those masses are not subsets of each other, the resulting ratio can exceed one. An adversarial synthetic test produced a `boundaryActivation` value of 3108.3031, violating the intended $[0, 1]$ domain. The r3 implementation replaces the ratio with $u/(R_d + u)$, where R_d is nonnegative within-domain risk mass and u is nonnegative unproven dependency risk mass, and clamps accumulated boundary pressure to $[0, 1]$ after coherence and evidence penalties. In symbols, its coherence increment has the form

$$x = \frac{u}{R_d + u} \in [0, 1], \quad \Delta BA = x \ln(1 + x), \quad BA_d \leftarrow \min\{1, \max\{0, BA_d + \Delta BA\}\}. \quad (6)$$

The pathological input is retained as a regression. This is a correctness repair of a defined engineering index, not evidence that its numerical values are calibrated to attack likelihood.

A computed scenario difference, not a predicted gain. The historical `expectedConfidenceGain` was the ranking proxy $P_i = 100 (r_i / \sum_j r_j) (1 - w_i) (1 - c_i)$, not a re-evaluation of ORI. The research branch retains that field and `expectedGain` as deprecated aliases, introduces `prioritizationProxy`, and computes the following quantity for eligible priority entries:

$$\Delta_i^{\text{scenario}} = OC(A^{(i \rightarrow \text{complete})}, E, \sigma, O, Q) - OC(A, E, \sigma, O, Q). \quad (7)$$

Here A contains the original responses, $A^{(i \rightarrow \text{complete})}$ changes exactly the selected question’s response to `complete`, E is the *same* declared-evidence payload, σ is the same strict-mode setting, O is the same organization and applicability profile, and Q is the same question bank and dependency structure. Both terms execute the full scoring pipeline, including its evidence and coherence penalties. Results are displayed as `counterfactualDeltaOC` in *percentage points*, qualified “scenario.” The calculation is deterministic and testable but is *not* a causal impact estimate, remediation success probability, predicted future control state, or certified gain. In strict mode, changing only an answer without adding missing proof can produce zero change.

Important ranking caveat. The list of prioritized items is *still ordered by the legacy ranking proxy* P_i , not by $\Delta_i^{\text{scenario}}$. Moreover, scenarios are evaluated for the existing prioritized list, not for every conceivable intervention. These are deliberate scope limits; the new delta must not be described as the optimization criterion currently used for ranking.

9 A versioned and auditable evidence contract

The research branch introduces `ori.methodology.cjs`, which adds `schemaVersion`, `methodologyVersion`, `strictEvidenceMode`, and an evidence-assurance summary to newly generated assessments. The summary distinguishes required artefacts *declared*, *missing*, and *not evaluated*. Crucially, a checkbox selection is a statement made by the caller; it is not cryptographic verification or an authenticated inspection. The supplied evidence must not silently acquire the stronger status of an independently verified fact. These fields are research-branch features and should not be attributed to the deployed version.

An evidence record in the experimental `ori.evidence-lifecycle.cjs` carries an identifier, collection time, optional expiry time, a set of declared dependency identifiers, and a revocation flag. At an evaluation instant

t , the lifecycle mapping is schematically

$$L_t(e) = \begin{cases} \text{malformed,} & \text{incoherent time ordering,} \\ \text{invalidated,} & \text{revoked or declared dependency changed,} \\ \text{expired,} & \text{no freshness deadline or } t \geq t_{\text{expiry}}, \\ \text{unverified,} & \text{otherwise.} \end{cases}$$

The default *unverified* is intentional: temporally recent evidence does not establish provenance integrity or truth of the recorded claim. Dependency changes in this prototype are supplied by the caller, not detected from live infrastructure. Invalid timestamps and malformed declarations are rejected or classified according to the interface contract.

Dependent findings. A separate experimental module `ori.conclusion-lifecycle.cjs` evaluates a finding associated with required evidence identifiers $D(c)$. Given an evaluated evidence collection E_t , it classifies a finding as missing if a referenced item is absent, invalidated if a required item is revoked or a declared dependency changed, stale if a required item is expired or temporally malformed, and current declaration otherwise (subject to the stated precedence). Formally, “current declaration” means

$$D(c) \subseteq \text{dom}(E_t) \quad \text{and} \quad \forall e \in D(c), L_t(e) = \text{unverified,}$$

not that the finding is operationally certified. It is a freshness and completeness predicate on submitted metadata. This module is not yet linked transactionally to stored production reports or to automatic event-driven certificate revocation.

10 Experimental conditional graph assurance

For each finite graph $G = (V, E)$ and pairwise distinct $s, w, t \in V$, define the bypass predicate

$$B(G; s, t, w) := [\exists \text{ a directed path from } s \text{ to } t \text{ in } G[V \setminus \{w\}]].$$

Let $\mathcal{K}$ be an explicitly supplied finite collection of compatible graphs. For nonempty $\mathcal{K}$ the result is established when $\forall G \in \mathcal{K}, \neg B(G)$, refuted when $\forall G \in \mathcal{K}, B(G)$, and unresolved when both outcomes occur among the supplied graphs. An empty collection receives the label inconsistent *by the tool’s declared convention*, without asserting that the real operational environment is logically inconsistent.

Correctness scope. For a well-formed finite graph and pairwise distinct endpoints, breadth-first search on the graph with the control node removed is sound and complete for existence of a bypass path. A returned sequence is a witness whose adjacent vertices correspond to actual edges and which omits w ; absence of a returned path means no such path exists in *that graph*. With adjacency lists the reachability search takes $O(|V| + |E|)$ time per model, while the current prototype builds and validates graph representations and performs two searches per model; the end-to-end complexity therefore also includes validation, adjacency construction, and output costs. Applying the predicate to every supplied model realizes the four statuses by direct finite quantification. These are standard graph-theoretic facts, not novel results.

The experimental kernel also reports how many models contain an ordinary source-to-target route, and flags the *vacuous* case where no destination is reachable. Thus “every reachable path goes through the WAF” is not silently conflated with “the backend is reachable” or “the WAF ruleset is effective.” The program rejects control-equals-source or control-equals-target inputs, because those would otherwise undermine the interception interpretation.

WAF example. Let s be an internet ingress, w an inspection gateway, and t a backend API. Model G_1 contains edges $s \rightarrow w \rightarrow t$ only, whereas G_2 adds $s \rightarrow t$. Then $B(G_1) = 0$ and $B(G_2) = 1$. The tool’s status for $\{G_1, G_2\}$ is unresolved, and it exhibits $[s, t]$ as a countermodel path in G_2 . An inventory that omits G_2 would instead produce established: this is a model-coverage failure, not an error in graph traversal. No actual customer network is claimed to have been tested in this example.

11 Implementation traceability and reproducible testing

The automated research workflow checks Node syntax and runs tests for scoring regressions, prohibited claims, methodology, lifecycle rules, counterfactuals, API integration, DOM rendering and the graph checker.

Research-branch module	Verified or claimed function and limitation
<code>ori.engine.cjs</code>	Existing weighted ORI calculations; legacy terms partially corrected, numerical calibration not established.
<code>ori.methodology.cjs</code>	Version and explicit declaration-only evidence accounting, integrated in the research API route.
<code>ori.assurance.cjs</code>	Standalone exact finite-graph reachability checker; no production caller demonstrated.
<code>ori.evidence-lifecycle.cjs</code>	Expiry, metadata-based revocation and declared dependency changes; no authenticity verification.
<code>ori.conclusion-lifecycle.cjs</code>	Standalone evidence-to-finding freshness linkage; no event-driven production invalidation.

Table 1: Source-linked components reviewed on GitHub research branch. Counterfactual implementation and tests at commit `9d61193984e6c909160e4335d38dfbed82fb6c2f`.

GitHub Actions completed successfully at [workflow run 39132597831](#) for the r3 branch. It included the model-graph regressions, scoring and real HTTP POST–persist–GET tests, DOM report-rendering tests, adversarial numeric bounds, and separately recalculated one-answer scenario-delta tests. A prior run exercised all $2^{12} = 4,096$ combinations of directed edges without self-loops on four fixed nodes, comparing the checker with a separately implemented DFS oracle and validating the returned path edges. This exhaustive *bounded* experiment is much narrower than full code correctness, differential testing of browser/API behavior, or security validation under malicious input.

Publication and deployment gates. Independent reference data are not yet available to calibrate coefficients, sensitivities, or empirical security performance. No assumption of independent evidence sources is warranted. The research module signatures and report schema require historical-score differential testing, release migration tests and full dependency-install validation. Existing DOM checks are not full real-browser journey tests. The counterfactual ranking is still based on the old heuristic and the method is uncalibrated. A production assessment must remain tied to the engine/methodology version that generated it. The present paper therefore describes a tested research branch and an existing product foundation rather than announcing a production-ready formal assurance feature.

12 Hostile novelty test against assurance standards

The Object Management Group’s Structured Assurance Case Metamodel (SACM) already models auditable claims, supporting or opposing evidence, evidence provenance and artifact lifecycle. NIST’s OSCAL Assessment Results model similarly captures observations, supporting evidence, findings, time-bounded assessment metadata and continuous-monitoring workflows. Consequently, ORI cannot claim originality merely for linking evidence to claims, retaining provenance, or invalidating a finding when evidence changes. These are implementation patterns with substantial prior art [1, 2]. Standard finite-graph reachability, safety properties and counterexamples likewise precede this prototype [7, 4].

The defensible, narrower ORI-specific contribution is the concrete mapping between questionnaire-based posture inference, explicitly versioned declarations, a stand-alone model-conditional path checker, and a conservative evidence-lifecycle interface. Its value is a potentially auditable product architecture; whether it improves correctness, analyst workload, false reassurance or practical decision-making against OSCAL/SACM-compatible baselines is an untested empirical question. No independent novelty claim is made for the elementary graph algorithm or four-case classification.

Additional hostile limitations. The current research API stores lifecycle results in the report at creation time; it does not re-check them when an old report is later retrieved, and it does not bind each evidence identifier cryptographically to an observed artifact. The conclusion-lifecycle module consumes a separate list of raw evidence metadata and is not transactionally wired to the HTTP assessment flow or graph checker. Its precedence rule classifies missing input before invalidated or stale input; multiple simultaneous defects should be exposed individually rather than interpreted as an absolute severity ordering. A caller’s assertion that a dependency changed is not evidence that ORI independently detected the change. A claim marked “current declaration” remains unverified by design. These limits prohibit a claim of continuous operational assurance.

13 Research provenance and the quantum boundary

The author’s research on quantum entropy, block coherence, and operator-algebraic structures concerns mathematical settings distinct from the operational cybersecurity framework developed here. No quantum entropy inequality, BKM coercivity estimate, or operator-algebraic duality theorem is invoked to establish the validity of ORI scores or its conditional graph-assurance procedures. Any future connection between these domains would require a mathematically specified correspondence and independently established results. The conditional graph-verification procedure presented here applies established finite directed-graph reachability methods; it is not proposed as a new mathematical theorem.

14 Limitations and conclusion

The model is deliberately scoped. An unknown real-world dependency need not belong to $\mathfrak{M}$; no algorithm confined to that model can guarantee its discovery. Adversarial evidence manipulation can defeat assurance unless source-integrity assumptions are enforced. Conditional logical certainty is different from probabilistic risk and from actual safety; reducing uncertainty is not identical to reducing exposure. The proposed assurance semantics and countermodel approach arise from established formal methods and are not claimed as novel mathematical discoveries.

The actionable principle is nevertheless significant: *security confidence should be attached to a precisely defined claim, declared evidence dependencies, and explicit assumptions, rather than inferred from an absence of alarms*. For AI-assisted cybersecurity, such discipline is particularly relevant when machine-generated recommendations may conceal the uncertainty or limitations of their inputs. Operational incompleteness, as used here, names a practical failure of warranted inference rather than a newly proven universal property of cyber systems. ORI v6 already performs evidence-conditioned and dependency-sensitive risk inference, which provides a concrete engineering foundation. Its current scores and the one-answer scenario differences are engineering computations; neither should be sold as causal forecasts, formal certificates, or deductions from quantum operator-algebraic theorems. An isolated finite-graph research prototype now illustrates the separately checkable semantics but is not part of the deployed product. A future integrated ORI verification layer should be judged on independent correctness and decision value under realistic evidence gaps.

References

- [1] Object Management Group, *Structured Assurance Case Metamodel (SACM)*, version 2.3 (2023). [Specification](#).
- [2] NIST, *OSCAL Assessment Results Model*, technical documentation. [Documentation](#).
- [3] F. Lin, “Opacity of discrete event systems and its applications,” *Automatica*, 47(3), 496–503 (2011). [doi:10.1016/j.automatica.2011.01.002](#).
- [4] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, “Counterexample-Guided Abstraction Refinement,” CAV (2000). [doi:10.1007/10722167_15](#).
- [5] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, “Optimal sensor activation for diagnosing discrete event systems,” *Automatica* 46 (2010), 1165–1175. [doi:10.1016/j.automatica.2010.04.004](#).
- [6] NIST, *Assessing Security and Privacy Controls in Information Systems and Organizations*, SP 800-53A Rev. 5 (2022). [doi:10.6028/NIST.SP.800-53Ar5](#).
- [7] C. Baier and J.-P. Katoen, *Principles of Model Checking*, MIT Press (2008).
- [8] A. Singhal and X. Ou, *Security Risk Analysis of Enterprise Networks Using Probabilistic Attack Graphs*, NIST IR 7788 (2011). [doi:10.6028/NIST.IR.7788](#).
- [9] E. Tabassi, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1 (2023). [doi:10.6028/NIST.AI.100-1](#).
- [10] H. Nasreddine, “Essential Duality and Maximal Non-signalling Extensions in Algebraic Quantum Field Theory,” arXiv:2605.00075v3 (2026). [arXiv link](#).

- [11] H. Nasreddine, “Midpoint BKM Estimates and Boundary Coherence,” arXiv:2605.11024v2 (2026). [arXiv link](#).
- [12] H. Nasreddine, “Entropy Moduli and BKM Coercivity for Rank-Deficient Non-Commutative Markov Semigroups,” arXiv:2604.16616 (2026). [arXiv link](#).
- [13] H. Nasreddine, “Entropy Stability under Block-Separable Constraints: Fixed-Weight Rank-Lattice Envelopes and Convexification,” arXiv:2512.16192 (2025–2026). [arXiv link](#).
- [14] H. Nasreddine, “Energy-Constrained Commutator Variance for Weyl Pairs,” arXiv:2609.19163 (2026). [arXiv link](#).

Document status. Research-branch technical white paper, 18 October 2026. GitHub research reference: 9d61193984e6c909160e4335d38dfbed82fb6(r3); successful CI run 39132597831; editorial and implementation updates 18 October 2026. All experimental modules and provenance/freshness claims are conditional; no production formal-certification feature, empirical accuracy or mathematical priority is claimed. Release requires author and independent technical review.