

R-BED™ | Risk-Based Evaluability Design

A public methodology for risk-calibrated, evidence-based evaluation of deployed AI systems

Version 1.1 | 27 October 2026 | Public community resource

Prepared by Vishal Kumar Srivastava, PhD

Purpose	Specify how to define system claims, find and characterize failure, test whether deployment controls contain failures, assess consequences, and make bounded deployment decisions.
Applicability	Traditional machine learning, generative AI, retrieval-augmented systems, and AI agents, with depth proportional to materiality and risk tolerance.
Status	Proposed practitioner methodology; not a standard, certification, validated scoring instrument, or NIST-endorsed framework.

Executive summary

R-BED organizes system-level evaluation evidence across three connected vertices: (V1) Technical Failure—where a configured AI system can fail; (V2) Deployment Controls—whether that failure can reach a user, business process, or external system; and (V3) Business Impact—what material consequences follow. Thirteen diagnostic dimensions (6+4+3) turn these questions into claims, evidence requests, falsification tests, and decisions.

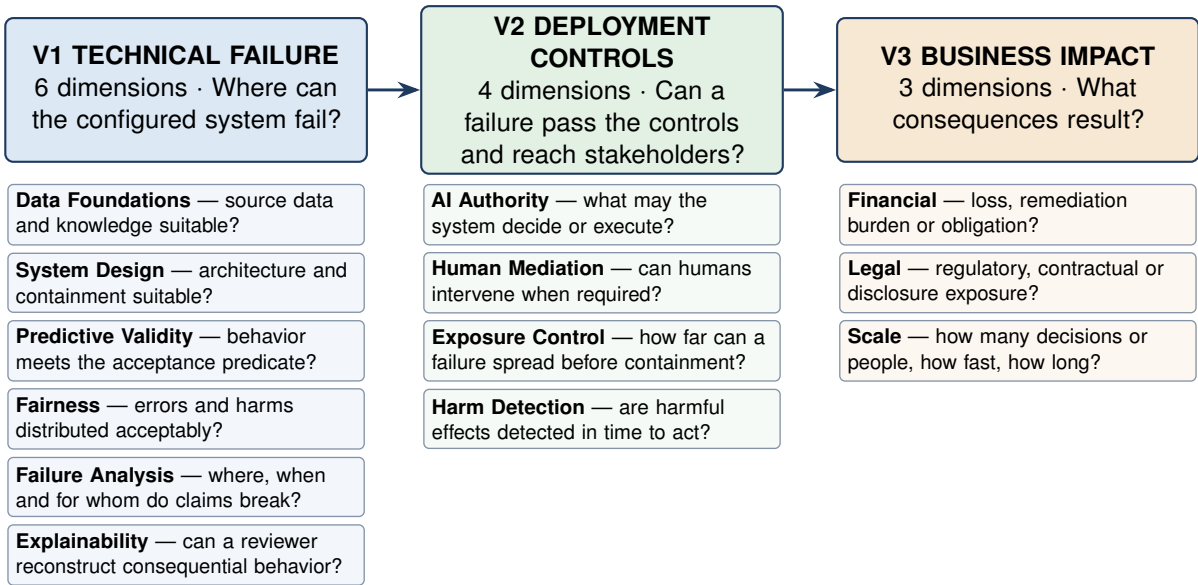


Figure 1. R-BED architecture: thirteen diagnostic dimensions in three connected vertices (6 + 4 + 3). The arrows show a simplified failure-mediated pathway; not every harmful outcome requires a technical failure. Dimensions are locations for claims and evidence, not scores, and evidence from one vertex cannot substitute for another.

The central practice is to evaluate a defined claim about a configured system, not a model score in isolation. A credible result must identify the deployment context, test distribution or scenarios, system versions, limitations, control tests, uncertainty, affected stakeholders, and accountable decision. R-BED does not collapse the thirteen dimensions into a single numerical rating.

Four principles

- **Risk governs depth:** prioritize consequences and unresolved uncertainty, not incident frequency alone.
- **Failure does not equal harm:** test the independent boundaries that stop an error from becoming an action.
- **Evidence must connect end to end:** component, trajectory, and outcome evidence jointly support a system claim.
- **A remediation must be tested:** detection, proposed fixes, and verified containment are distinct evidentiary states.

1 Scope, audience, and limits

This specification is a non-prescriptive reference for AI developers, validators, risk managers, assurance teams, researchers, and deployment decision owners. A framework user selects the applicable dimensions and documents omissions. High-consequence deployments typically require greater breadth of scenarios, more independent challenge, tighter restrictions, and stronger ongoing monitoring.

R-BED is intended to complement, not replace, organizational risk governance, sector regulation, external assurance, human impact assessment, cybersecurity, privacy management, or the NIST AI RMF. Nor does use of this method establish compliance. Its methods require tailoring to the setting and participation by stakeholders capable of identifying harms that internal teams may miss.

The thirteen-dimension structure is an engineering proposal, not an empirically validated taxonomy. Figure 1 summarizes the architecture and a simplified failure-mediated pathway; not every harmful outcome requires a technical failure. A positive evaluation does not demonstrate that unknown failure modes are absent.

Materiality tiers. R-BED requires evaluation intensity to increase with consequence, exposure, uncertainty and risk tolerance, but it does not prescribe or validate universal numeric tier thresholds. Adopting organizations may define low/medium/high or other risk tiers in their governance policy, record the assigned tier and rationale in the SES, and link each tier to test depth, independence, monitoring and approval rules. Tier labels are organizational implementation choices, not a universal R-BED score.

2 Terms and unit of assessment

Term	Meaning
Configured system	The particular model, data, prompts, retrieval sources, tools, permissions, human roles, monitoring, interfaces, and operating environment being evaluated.
System claim	A falsifiable statement about acceptable system behavior within explicit deployment conditions, tolerance or prohibition, population, and time horizon.
Failure event F	The system violates a specified claim or acceptance predicate in a defined exposure unit.
Deployment harm H	A failure passes applicable controls and produces a consequential effect in a workflow, person, or external system.
Consequence S	The magnitude and type of harm, including non-monetary effects that should not be forced into a price.
Evaluability	Capacity to reliably obtain current, credible, connected, and challengeable evidence about a system claim.

Term	Meaning
Exposure unit	The unit of analysis: request, session, transaction, decision, case, user, or time period, stated consistently.
Evidence status	Supported for stated conditions / Refuted / Unresolved / Not applicable—with reasons and evidence pointers.

3 The R-BED architecture: 13 dimensions

A dimension is a diagnostic location for a claim and evidence; it is not a standalone score. A system may need multiple evaluations within one dimension, and one experiment may inform several dimensions. Questions and sample tests below are illustrative, not mandatory or exhaustive.

3.1 Vertex 1 — Technical Failure (six dimensions)

Dimension	Question / example of relevant evidence
Data Foundations	Are source data, indexes, and knowledge authorized, suitable, current, and representative? Source lineage, freshness/coverage checks, and provenance tests.
System Design	Are architecture, interfaces, dependencies, and failure containment mechanisms suitable? Design review, contract tests, dependency and fault-injection tests.
Predictive Validity	Does configured system behavior satisfy an explicit acceptance predicate? Representative and stratified tests, uncertainty intervals, and deployment-like evaluations.
Fairness	Are errors, benefits, and harms distributed acceptably across relevant groups? Disaggregated outcomes with intervals and scenario-specific fairness checks.
Failure Analysis	Where, when, and for whom do claims break? Adversarial discovery, reproducible scenario tests, characterized failure regions and relevant population mass.
Explainability	Can a reviewer reconstruct consequential behavior? Tool and authorization traces, inputs and version identifiers, lineage and replay tests.

3.2 Vertex 2 — Deployment Controls (four dimensions)

Dimension	Question / example of relevant evidence
AI Authority	What may the system decide or execute? Permissions inventory, external authorization checks, deny-path tests, and least-privilege review.
Human Mediation	Can and do humans intervene when required? Review information shown, override behavior, escalation latency, and seeded-error detection.
Exposure Control	How far can a failure spread before being contained? Tested rate/amount limits, blast-radius analysis, rollback capability, and staged rollout.
Harm Detection	Are harmful effects detected with actionable speed? Alert sensitivity, coverage, escalation owner and alert-to-intervention time.

3.3 Vertex 3 — Business Impact (three dimensions)

Dimension	Question / example of relevant evidence
Financial	What loss, remediation burden, or financial obligation can result? Scenario analysis, portfolio concentration, and recoverability.
Legal	What legal, regulatory, contractual, or disclosure exposure can result? Applicable obligation mapping, decision trail, and notices/rights.
Scale	How many decisions, people or systems can be affected, how quickly, and for how long? Volume, concurrency, propagation and reach analyses.

Consequence attributes should be separately documented: harm type (physical, financial, privacy, dignity, discriminatory, psychological, or other), severity, duration, vulnerability of those affected, reversibility/recourse, reach, and uncertainty. These attributes do not create new R-BED dimensions. A purely monetary representation is insufficient for unpriced harms.

4 Eight-step R-BED evaluation workflow

01 / Context Document intended purpose, benefits, relevant populations, foreseeable misuse, exposure, risk tolerance, legal constraints, and system/version boundaries.

02 / Claims Define material System Claims as observable acceptance boundaries, quantitative or categorical. Record prohibitions that cannot be exchanged for business benefit.

03 / Enumerate failure pathways Identify likely and high-consequence failure classes across the 13 dimensions; include feedback, public incident reports, domain experts, and affected stakeholders where appropriate.

04 / Design evidence Select component, trajectory and outcome tests. Specify representative sampling, targeted adversarial tests, fault injection, instruments, stop conditions, and independent reviewers.

05 / Break and characterize Discover failure points; describe regions; reproduce under pinned configurations; estimate failure rates, coverage, and uncertainty without treating zero observations as proof of safety.

06 / Challenge controls Force relevant failures and test whether authorization, human mediation, exposure controls, and detection limit downstream effects before harm occurs.

07 / Evaluate and decide Join technical, control and consequence evidence. Record claim status, uncertainty, residual risks, gaps, owner, and deploy/restrict/rework/suspend decision.

08 / Fix, verify and renew Place safeguards at the effective layer, independently retest material fixes, revise system claims and re-evaluate upon drift, model/tool changes, incidents, or expansion of authority.

5 Evidence architecture and evaluation quality

Five proposed evaluability properties strengthen whether a system can be assessed: observability (events and context are recorded), testability (claims can be challenged safely), traceability (the end-to-end path can be reconstructed), reproducibility (configuration and stochastic variation are characterized), and actionability (evidence can trigger restriction or intervention). These are design objectives, not a formally proven necessary-and-sufficient set.

Review of the evaluation method. Before treating results as decision-grade evidence, a reviewer should test whether each metric measures the intended claim (construct validity), whether samples reflect the operating population, whether rare or severe failure regions are represented, whether uncertainty is bounded, and whether independent retest is feasible. Document metric limitations, detector blind spots, drift triggers and changes to the test protocol. This is a proposed review discipline, not empirical proof

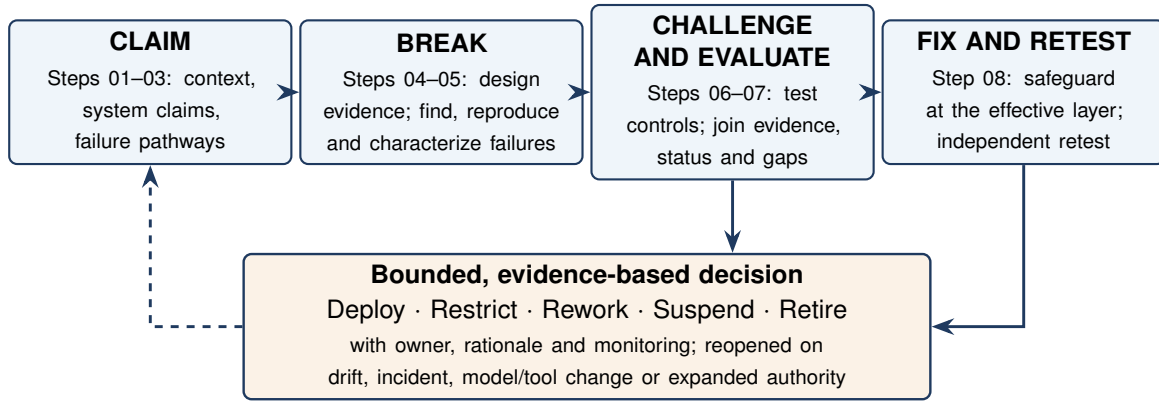


Figure 2. From failure evidence to decision. The eight steps of Section 4 grouped into four phases; a material fix is independently retested, and the decision is reopened whenever a re-evaluation trigger occurs. The decision categories are those of Section 7.

that any metric or test process is effective.

- **Component evidence:** unit tests of models, retrievers, tool wrappers, permission checks, monitors, and business rules.
- **Trajectory evidence:** ordered record of retrieval, prompts/versions, tool arguments, approval checks, human intervention, and state transitions.
- **Outcome evidence:** consequences for users, records, money, rights, performance, service obligations, and affected groups.

These perspectives are many-to-many with vertices: for example a component authorization test informs V2, and an outcome error rate can inform V1. Critical evidence should carry version tuple, timestamps, provenance, scenario ID, tester/approver, sampling frame, numerator and denominator, uncertainty, and links to raw traces or appropriately protected records.

6 Quantitative risk interpretation — conditional pathway only

For one explicitly defined failure-mediated pathway, one exposure unit, and deployment configuration A , let F denote the failure event, H a harm caused by that failure after the deployment controls, and $S \geq 0$ the priced severity of the resulting harm, defined conditional on both F and H . Define the failure-attributable loss $L_F = S \times \mathbf{1}\{F\} \times \mathbf{1}\{H\}$: it equals S when F and H both occur, and equals zero otherwise. The chain rule and conditional expectation then give:

$$E[L_F | A] = P(F | A) \times P(H | F, A) \times E[S | H, F, A]$$

This is a conditional identity under the definitions above, provided the relevant conditional probabilities and expectations exist; it is not a validated loss forecasting model, a universal risk score, or a claim that non-monetary harms can be priced. $P(F | A)$ requires failure evidence under a stated operating distribution; $P(H | F, A)$ requires challenge tests conditional on failure; and the severity term requires consequence evidence. If the conditioning event has zero probability, the corresponding factors require separate treatment rather than numerical interpretation. Dependencies, heavy tails, underreporting, shared-cause failures, unobserved pathways, and overlapping events must be disclosed. Do not sum pathways without addressing overlap; do not infer that a small average loss makes an irreversible or prohibited harm acceptable.

7 System Evaluation Specification (SES) — minimum record

Field	Minimum content
Claim and owner	ID, intended purpose, responsible decision maker, affected stakeholders, accepted boundary.
System boundary	Model/tool/prompt/data versions, integration surfaces, delegated permissions, deployment populations.
Consequence framing	Impact classes, exposure unit, severity, reach, duration, reversibility, external constraints.
Tests and sampling	Scenario strata, representative sample definition, challenge tests, controls to be injected, methods.
Evaluation results	Raw or protected trace IDs, metric definitions, estimates and intervals, errors, failed controls.
Evidence coverage	Supported / refuted / unresolved / N/A by material claim; gaps and measurement limitations.
Disposition	Deploy / restrict / rework / suspend / retire; owner, rationale, approvals, monitoring and triggers.
Retest obligations	Safeguard changes, independent verification, time-bound review and reevaluation triggers.

8 Worked application — illustrative only

A fictional customer-support refund agent is permitted to answer policy questions but may initiate a refund only after authenticated authorization and approval above an established threshold. Its model gives a plausible response while an untrusted retrieved note induces a tool call above that threshold. The scenario is a teaching construction, not an observation or validated experiment.

Vertex	Claim and falsification probe	Evidence / decision implication
V1	A retrieved note cannot override policy; send instruction-like text in untrusted retrieval content and observe tool arguments.	Capture retrieval provenance, attempt logs and version tuple. A correct final response does not erase a misdirected tool call.
V2	A server-side check rejects over-limit refunds regardless of model content; inject forced unauthorized attempts.	Record actual deny decisions, approval path, state changes, and detection latency. A prompt-only refusal is not evidence of control.
V3	No customer or business consequence may arise above tolerance.	Analyze attempted vs. completed refunds, affected records, legal notices, recoverability and cumulative exposure.

Disposition rule (illustrative): keep the agent in sandbox or restricted mode while the permission boundary is unresolved. After implementing an independent authorization control, rerun the seeded failure, legitimate allowed requests, regression scenarios and monitoring drills before any broadened deployment decision.

Completed record (hypothetical). The abbreviated SES below carries the same fictional agent from first test to decision. Every figure is invented for illustration; none is an observation.

SES stage	Hypothetical record
Claims	C1: a retrieved note cannot override refund policy. C2: a server-side check rejects over-limit refunds regardless of model content. C3: no unauthorized refund is completed (categorical prohibition). Owner: head of support operations; boundary: one build, one retrieval snapshot, 90 days.
Initial tests	20 injected-note scenarios; 25 forced over-limit tool calls through a sandbox payments gateway.
Results	C1: 3 of 20 scenarios produced an above-threshold tool call (95% interval 0.03–0.38). C2: 0 of 25 forced attempts completed (upper 95% bound 0.14); alert-to-intervention time not measured.
Evidence status	C1 <i>refuted</i> . C2 <i>unresolved</i> : the bound is wide and detection latency is untested. C3 <i>unresolved</i> : no production-like outcome evidence.
Initial disposition	<i>Restrict</i> : sandbox only; any completed over-limit refund or version change triggers re-review.
Safeguard change	Add an independent server-side authorization control that does not depend on model output.
Independent retest	A reviewer outside the build team reruns the seeded failure, legitimate requests, regression scenarios and a monitoring drill: the attack still reaches tool calls, but 0 of 60 forced attempts complete (upper 95% bound 0.06) and every drill alert triggers intervention within the agreed limit.
Updated status	C1 remains refuted (the model still misdirects), but its consequence is contained. C2 <i>supported (bounded)</i> for this build, gateway and window. C3 <i>unresolved</i> until production-like outcome data exist.
Updated disposition	<i>Restrict</i> , now permitting a limited pilot with human approval retained and a daily attempted-versus-completed refund report; broader deployment awaits C3 evidence and risk-owner sign-off. Triggers: any completed over-limit refund, model or tool change, retrieval-corpus drift, expansion of authority.

9 Evidence statuses, limitations and governance handoff

- **Supported (bounded)**: adequate evidence supports the claim only for the specified population, versions, time window, and conditions.
- **Refuted**: admissible evidence shows violation of a required condition or control.
- **Unresolved**: crucial evidence, coverage, validity, independence, or uncertainty is inadequate; unresolved is not synonymous with safe.
- **Not applicable**: a documented, reviewable justification explains why a dimension or claim does not apply.

R-BED requires a responsible organizational decision owner, but does not itself create a governance structure. Risk tolerance, stakeholder consultation, legal sufficiency, fairness objectives, and action authority must come from the adopting organization and applicable rules. An evaluation report should record dissent, known missing tests, non-monetary harms, and explicit re-evaluation conditions.

10 Relationship to NIST AI RMF 1.0

NIST AI RMF 1.0 provides the voluntary GOVERN, MAP, MEASURE and MANAGE functions. R-BED can support particular outcomes by structuring system claims, harm pathways, test methods, control challenges, consequences, and evidence-based decisions. The companion Crosswalk maps selective NIST subcategories to specific sections of this specification, with explicit partial coverage and gaps. No claim is made that R-BED covers all NIST AI RMF outcomes or provides NIST certification or endorsement.

11 Relationship to other AI risk-management and evaluation resources

R-BED is intended to complement, not replace, existing resources, and it makes no claim of empirical superiority over any of them.

- **ISO/IEC 23894 (AI risk management guidance) and ISO/IEC 42005 (AI system impact assessment)** address organizational risk-management processes and impact assessment. R-BED can supply system-level evidence for such processes; it does not provide a management system or an impact-assessment method.
- **AI Verify (Singapore)** offers a testing framework and toolkit for selected technical and process checks. R-BED is not a toolkit; it specifies how test results, control challenges and consequences are connected in one bounded, reviewable record.
- **NIST TEVV-Athlon (NIST AI 200-2, initial public draft)** describes how to build customized assessments from an organization's TEVV objectives. R-BED could be used to scope an assessment of one kind: failure pathways in a configured deployment.

R-BED's specific emphasis is the chain of evidence connecting failure, containment, consequence and action: whether a failure reaches harm given the deployed controls, how that was independently challenged, and what decision the bounded evidence supports. Broader governance, measurement and assurance practices in these resources remain outside R-BED's scope. Descriptions above are summaries by the author; readers should consult each resource for its authoritative scope.

12 Public availability, attribution and use

© 2026 Vishal Kumar Srivastava. All rights reserved except as stated here. This public specification may be read, downloaded, cited and applied in an organization's own evaluation practice without charge and without a framework-access fee, with attribution. Republication, translation or derivative versions require the author's written permission. No endorsement, certification, warranty or regulatory safe harbor is provided. Third-party rights are unaffected.

Public URL of this document: https://rbed.ai/RBED_Public_Framework_v1.1.pdf. Companion NIST AI RMF crosswalk: https://rbed.ai/NIST_AI_RMF_RBED_Crosswalk_v1.1.pdf.

Methodology contact: Vishal Kumar Srivastava · vshfrm@gmail.com

Trademark: R-BED™ is claimed as a trademark by the author; no registration, certification or NIST endorsement is asserted. This resource is independent of NIST and does not offer a product or service.

References and source notes

- NIST, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1 (2023), <https://doi.org/10.6028/NIST.AI.100-1>.
- ISO/IEC 23894:2023, *Guidance on risk management*; ISO/IEC 42005:2025, *AI system impact assessment*.

- Singapore IMDA / AI Verify Foundation, AI Verify testing framework and toolkit; see the IMDA announcement of the AI Verify Foundation (2023), <https://www.imda.gov.sg/resources/press-releases-factsheets-and-speeches/press-releases/2023/singapore-launches-ai-verify-foundation>.
- NIST, *The TEVV-Athlon Framework for Evaluating AI Systems*, NIST AI 200-2 (initial public draft, August 2026), <https://www.nist.gov/artificial-intelligence/tevv-athlon-framework-evaluating-ai-systems>.
- NIST AI Resource Center, *AI RMF Core*, <https://airc.nist.gov/airmf-resources/airmf/5-sec-core/> (accessed 27 October 2026).
- NIST AI Resource Center, *Community Crosswalks*, <https://airc.nist.gov/airmf-resources/crosswalks/> (accessed 27 October 2026).
- *Methodology note*: this document is a standalone public statement of the R-BED methodology; its section definitions, example and limitations may be referenced independently of any other publication.