

Determinism-Bounded Authorization

A measurable, auditable foundation for evidence-gated, provably bounded autonomy.

CONCEPT & PRELIMINARY WORK · COMPANION TO THE ECHOSTRIKE WHITE PAPER

Pinyo Taeprasartsit · Varin Khera · SECStrike

ABSTRACT

Autonomous security-validation platforms increasingly place a language model between an authorised target and an irreversible action. Our architecture paper [1] argues that discovery is not proof and that a recommendation is not an authorisation, and it organises a working platform around evidence gates, bounded autonomy and human escalation. That is an argument about how the system should behave. This companion paper supplies the formal layer underneath it: the quantities that let a reviewer, an auditor or a regulator check, on a given run, that the refusal to treat a guess as a finding actually held.

Determinism-Bounded Authorization (DBA) certifies the *execution path* that produced a finding rather than the finding itself, because the same finding reached two different ways carries two different degrees of trust. For each path we define a *determinism bound*: an upper bound on the probability that an asserted property is false. The bound is set at deterministic gates, degrades through non-deterministic reasoning at a measured rate, and is restored by re-execution, which we treat as a formal replenishment operator. The bound drives a graduated authorisation function whose outcomes (execute, investigate, escalate, refuse) are computed from a calibrated quantity rather than assigned by hand. Authorised scope is enforced deterministically by information-flow control, and every authorised action emits a defeasible assurance record whose evidence can be independently re-executed.

This is a concept paper. It states the definitions, the degradation–replenishment calculus, the decision rule and a set of basic formal properties, with illustrative examples. It reports no empirical results; validation inside the ATVE workflow on the EchoStrike codebase is under way and will be reported separately.

STATUS OF THIS PAPER

Concept and preliminary formal work. Definitions, calculus, decision rule and basic properties are stated, and proved where the proof is short. Every number in an example is illustrative, not a measured or production value. Nothing here describes a shipped product feature. Read it alongside Paper I [1], which explains how the platform is organised and why; this paper explains how that organisation can be made computable, measurable and auditable.

3	Positioning against prior work	9	Composition, system-level assurance and path ranking
4	Overview: the DBA pipeline	10	Governance mapping
5	Execution paths and the determinism bound	11	Limitations and threats to validity
6	Calibration and graduated authorisation	12	Open problems and roadmap
7	Deterministic scope enforcement	13	Conclusion
8	The auditable assurance record	14	References

1. Introduction

The last two years of AI-for-security have mostly sold speed: faster discovery, faster write-ups, faster remediation advice. Speed without a proof standard produces confident noise, because a language model is fluent at “this looks like a real weakness” and willing to present it as “this is a real weakness, here is the evidence, and it reproduces.” An autonomous platform that cannot tell those two sentences apart is acting on guesses.

The EchoStrike architecture paper [1] confronts this at the level of system design. It separates finding from proof and recommendation from authorisation; it gates findings so that a verification path can refuse a claim; it keeps humans in the loop as an architectural outcome rather than an apology; and it treats autonomy as a control problem, allowing the reasoning layer three outcomes: execute, investigate, escalate. It also reports a negative result: in a controlled transfer, its packet-derived meta-confidence signal did not transfer well enough to support selective autonomous execution.

Those commitments are stated qualitatively, which is the right register for an architecture paper. To make them checkable from outside, they need quantities underneath. A reviewer should be able to read a run and see that the bound held. An auditor should be able to receive an artifact showing that a specific action was inside a specific scope at a specific level of trust. A regulator should have a number to point at. And the platform’s reusable verified knowledge, which replays an already-validated check with no new model calls, should be recognised for what it is: the main source of determinism in the whole system, not only an efficiency.

Our thesis is that trustworthiness is a property of the *execution path*, not of the finding. The category a finding belongs to does not fix how much to trust it. What fixes it is how the finding was established, and how far that proof degraded on the way to the output. This paper makes that idea computational.

Contributions

1. **Execution paths as the unit of certification.** We model an engagement as a typed graph, define the realised execution path of a finding, and attach the bound to the path rather than the finding (Section 5).
2. **A determinism bound with a degradation–replenishment calculus.** The bound is set at deterministic gates, attenuated through non-deterministic reasoning at a measured rate, and restored by re-execution. We prove basic properties: two-sided bracketing, monotone degradation, and an exact condition for when adding a re-execution gate helps (Section 5).
3. **Calibrated graduated authorisation.** We calibrate the bound against re-execution ground truth and map it, through impact-indexed thresholds, onto execute, investigate, escalate and refuse, including the

minimum calibration size below which the policy must decline to automate (Section 6).

4. **Provable scope and a re-executable assurance record.** Scope is enforced by information-flow control at a deterministic reference monitor, so injected text cannot widen it; each authorised action emits a defeasible claims–arguments–evidence record whose evidence can be re-run (Sections 7–8).
 5. **Composition and a governance mapping.** Per-path bounds compose into an engagement-level claim that degrades gracefully and shows where a gate buys the most trust; each component maps onto ISO/IEC 42001, the EU AI Act’s human-oversight requirement and the NIST AI RMF (Sections 9–10).
-

2. The architecture and the layer underneath

This paper takes the architecture paper [1] as its application foundation and does not restate it. We summarise only what we build on, then name the formal objects this paper adds.

What we adopt

The platform runs a closed loop, $\text{FIND} \rightarrow \text{PROVE} \rightarrow \text{FIX} \rightarrow \text{REPROVE}$, across specialised components that share typed state and are kept apart so that no one component declares victory for the rest [1]. Four of its commitments carry weight here:

1. A finding must carry the execution context and evidence that produced it, and must survive controls that are allowed to refuse it (fail-closed).
2. Agents work from a shared verified journal rather than private conversational memory, so handoffs are typed state.
3. A validated check can be held as reusable verified capability, tied to a technical fingerprint, and replayed with no new model calls.
4. The control layer, ATVE, permits three outcomes (execute, investigate, escalate), and the design goal is defensible autonomy, not maximum autonomy.

What this paper adds

The architecture paper deliberately withholds scoring functions, thresholds and schemas [1]. That remains true here: we publish the *shape* of the formal layer, not its operating values. The shape consists of four objects.

1. **A unit of certification.** The decision state in [1] holds belief, meta-confidence and uncertainty, and notes that a pile of correlated observations is not independent confirmation. DBA fixes the object a bound attaches to: the execution path (Section 5).
2. **A measurable bound with a composition rule.** “Evidence-gated” becomes a quantity: how much determinism a specific run retained, how a proven property degrades through non-deterministic processing, and how re-execution restores it (Section 5). The transfer gap reported in [1] is the motivating symptom. A signal that tries to predict trust from features of a belief can fail to transfer; a bound grounded in re-execution is anchored to an outcome the platform can check.
3. **A decision function.** Execute, investigate and escalate become the output of a calibrated function of the bound, the action’s impact, the authorised scope and policy, with a stated and testable safety–utility trade-off (Section 6).
4. **An auditable, provable boundary.** Scope becomes a property the system can prove rather than one it is asked to maintain (Section 7), and every authorised action leaves a re-executable record (Section 8).

In short, the architecture paper says the system should refuse to treat a guess as a finding. This paper defines the measurable object that lets anyone verify the refusal held, and the record that shows it.

3. Positioning against prior work

Both ends of this problem are occupied. Mechanism-level work certifies or monitors agent behaviour at runtime; policy-level work argues for oversight, accountability and safety cases. A recent survey of AI-driven threat validation [19] and surveys of agentic AI in cybersecurity [17] map the application side of the field. The space between them, a measured, re-execution-grounded bound driving auditable authorisation for *authorised* security validation, is where DBA sits. Each row names what a line of work gives, what DBA reuses, and where it departs.

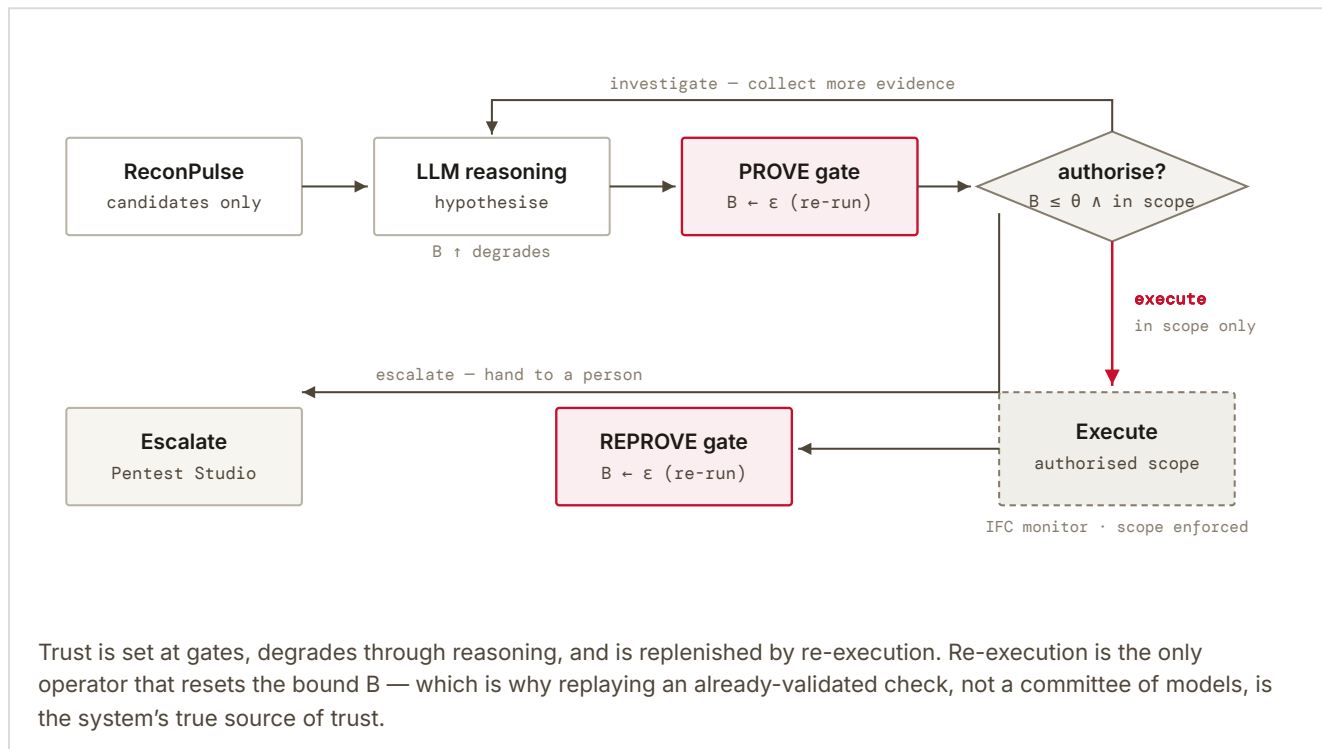
PRIOR WORK	WHAT IT CONTRIBUTES	WHAT DBA BORROWS	HOW DBA DIFFERS
Certified runtime safety for tool-using agents [2]	A deterministic gate enforces exactly the safety policies whose good prefixes it recognises; a Neyman–Pearson frontier; a finite-sample conformal certificate, possibly achieved only by vacuous block-all; static scores need not identify the closed-loop frontier	The enforceability backbone, and conformal calibration as the calibration layer	Specialises to security findings; adds re-execution as a replenishment operator and a per-path bound, not a per-action gate alone
QUIVER: perturbation propagation in compound AI [3]	Judge-free sensitivity from typed distance metrics; amplifier and absorber edges; path sensitivity as a product along the path; bifurcation thresholds; identifies critical amplification paths	Judge-free divergence as the per-node input to the bound; multiplicative composition along a path	Grounds fragility in re-execution outcomes rather than output distance alone, and ties path analysis to an authorisation decision
ProbGuard (first released as Pro2Guard) [4]	Learns a Markov-chain abstraction of agent behaviour from traces, with a PAC-style sample analysis; intervenes before the predicted chance of staying safe falls below a threshold	A quantified probability bound feeding a gate	The bound is over a property’s proof and calibrated to re-execution; replenishment is an explicit operator rather than a model-checking step
FIDES: information-flow control for agents [5]	Confidentiality and integrity labels; deterministic policy enforcement before tool calls, independent of the model	The reference-monitor approach to scope enforcement	Applies labels to authorised scope (target, action class) so authorisation is provable, and composes it with the trust bound
Uncertainty propagation and failure attribution [6] [7]	Formal decomposition of per-step uncertainty; step-level failure attribution, where the best reported method reaches 14.2% step-level accuracy	Motivation to avoid model judges as the measurement instrument	Replaces judge-based attribution with measured divergence and re-execution outcomes
Cyber-inability safety case [8]	A claims–arguments–evidence template arguing that a model <i>cannot</i> reach dangerous cyber capability	The CAE structure for the assurance record	Inverts the argument: an authorised validation tool is meant to act, so the case rests on control, scope and bounded authorisation rather than inability
Control and safety-case frameworks [9] [10]	Safety via control measures and auditing under conservative assumptions; inability, control, trustworthiness and deference arguments	The control-argument stance and conservative extrapolation	The safety story is scope-boundedness plus re-executable evidence for an authorised, purpose-built system
Behavioural-assurance critique [11]; evidence sufficiency [12]	Behavioural testing cannot verify what governance now requires; trace- and schema-level evidence frequently overclaims sufficiency	The case for mechanism-level, property-level evidence	Supplies that evidence as re-execution, attached to each authorisation decision

The position DBA claims is the intersection of five qualifiers: graduated authorisation, for authorised autonomous security validation, driven by a measured bound, evidenced by re-execution rather than

attestation, with deterministic scope enforcement. Each has prior art on its own; to our knowledge the combination has not been formulated before.

4. Overview: the DBA pipeline

DBA leaves EchoStrike’s FIND → PROVE → FIX → REPROVE loop intact and makes one quantity explicit along it: the determinism bound B , an upper bound on the chance that an asserted property is false. B is undefined at discovery, grows through non-deterministic reasoning, and is driven back to a gate floor ε by re-execution. The authorisation step reads B , the action’s impact-indexed threshold θ and the authorised scope, then routes the finding to execute, investigate, escalate or refuse. Scope is enforced by a deterministic information-flow monitor, not by the model.



The figure carries the main point of the paper. Re-execution is the only operator that resets B . That is why the platform’s replay of already-validated checks is its real source of trust, and why model agreement, which only adds correlated stochastic steps, cannot stand in for it.

5. Execution paths and the determinism bound

Definition 1 (Workflow graph). An engagement workflow is a typed directed acyclic graph $G = (V, E)$ whose nodes are partitioned into stochastic reasoning nodes R (model calls), deterministic gates D (re-execution, oracle checks, the information-flow monitor) and human nodes H :

$$V = R \dot{\cup} D \dot{\cup} H.$$

Each node produces a typed output; a stochastic node returns a distribution over outputs.

Definition 2 (Execution path). For a finding f , the execution path $\pi(f)$ is the realised sequence of nodes and outputs that produced and validated f . We certify $\pi(f)$, not f . The same finding reached by two paths has two bounds.

Definition 3 (Property and determinism bound). A claim about f is a property φ (for example *confirmed*, *reproducible*, *in scope*), with asserted value $\hat{\varphi}$ and true value φ . The determinism bound of φ along π is intended as an upper bound on the probability that φ is asserted but false:

$$B_\varphi(\pi) \geq \Pr[\varphi = 0 \mid \pi, \hat{\varphi} = 1], \quad T_\varphi(\pi) = 1 - B_\varphi(\pi). \quad (1)$$

T is the trust retained along the path; B is the auditable quantity. A small B means the path established φ and did not lose it.

Whether the computed B really upper-bounds the true failure probability is an empirical question, and Section 6 is where it is checked. The authorisation guarantee does not depend on B being exact; it depends on calibrating the decision against re-execution outcomes.

Degradation

A deterministic gate g that establishes φ sets the bound to its own residual error ε_g , close to zero for exact re-execution. Each later stochastic node v on the dependency path of φ attenuates trust in proportion to its measured divergence $d_v \in [0, 1]$: how much v 's output varies on identical inputs. It is measured without a judge, as the expected typed distance between independent runs,

$$d_v = \mathbb{E}_x \mathbb{E}_{y, y' \sim v(x)} [\Delta_{\text{type}}(y, y')] \in [0, 1], \quad (2)$$

where Δ_{type} is a normalised distance suited to the output type: categorical mismatch, Jaccard distance on sets, or embedding distance on text. One stochastic step updates trust multiplicatively:

$$T_{\text{out}}(v) = T_{\text{in}}(v) (1 - d_v) \iff B_{\text{out}}(v) = 1 - (1 - B_{\text{in}}(v))(1 - d_v). \quad (3)$$

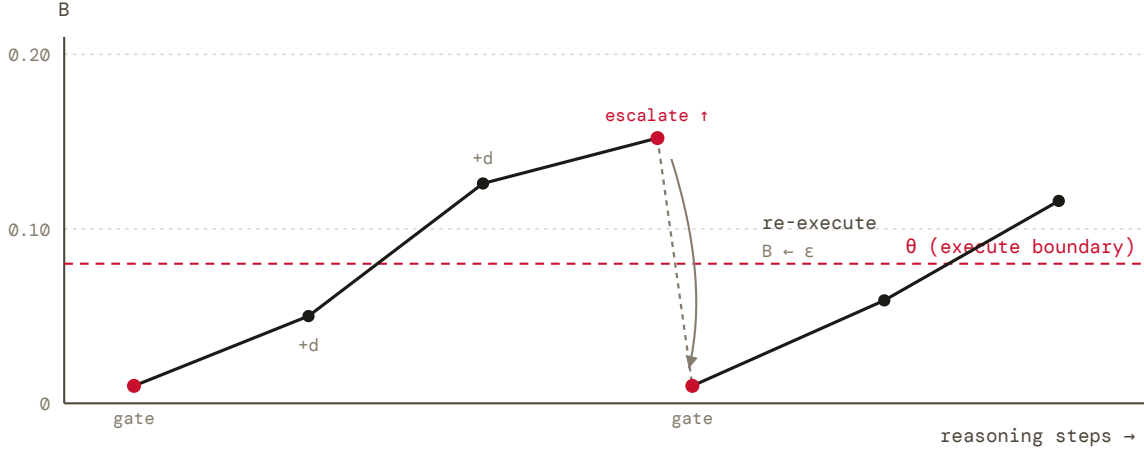
Over the segment $\text{seg}(\pi)$ of stochastic nodes since the last gate g , the bound accumulates to

$$B_\varphi(\pi) = 1 - (1 - \varepsilon_g) \prod_{v \in \text{seg}(\pi)} (1 - d_v). \quad (4)$$

Definition 4 (Replenishment operator). A re-execution gate ρ re-establishes φ deterministically (in the platform, by replaying an already-validated check with no new model calls) and resets the bound to the gate floor, however far it had degraded:

$$\rho : B_\varphi \mapsto \varepsilon_\rho, \quad \varepsilon_\rho \ll 1. \quad (5)$$

Replenishment is idempotent and does not depend on the incoming bound.



The bound along one path (illustrative values from Example 1). Each stochastic step raises B ; a re-execution gate drops it back to the floor ε . A finding is auto-executed only while B stays below its impact-indexed threshold θ .

Example 1 (illustrative values). A finding is confirmed by a deterministic check with floor $\varepsilon_g = 0.01$. Three reasoning steps follow on the property's dependency path, with measured divergences $d = 0.04, 0.08$ and 0.03 : say, interpreting the result, planning the next check, and summarising the evidence. By (4) the bound rises from 0.010 to $0.050, 0.126$ and 0.152 . Replaying the validated check applies ρ with $\varepsilon_\rho = 0.01$ and returns the bound to 0.010 . The finding is the same throughout; the trust it carries is not.

Basic properties

The calculus is deliberately simple, which makes a few properties immediate. They are the preliminary formal results of this paper.

Proposition 1 (Bracketing). For any segment with gate floor ε_g and divergences d_v ,

$$\max \left\{ \varepsilon_g, \max_v d_v \right\} \leq B_\varphi(\pi) \leq \min \left\{ 1, \varepsilon_g + \sum_v d_v \right\}. \quad (6)$$

Proof. Every factor in (4) lies in $[0, 1]$, so the product is at most any single factor $1 - x$, giving $B \geq x$ for each $x \in \{\varepsilon_g\} \cup \{d_v\}$. For the upper bound, the Weierstrass product inequality $\prod_i (1 - x_i) \geq 1 - \sum_i x_i$ for $x_i \in [0, 1]$ gives $B \leq \varepsilon_g + \sum_v d_v$. ■

Read operationally: a single high-divergence step cannot be averaged away by low-divergence neighbours, and many small steps add up at most linearly. In Example 1, $0.08 \leq 0.152 \leq 0.16$.

Proposition 2 (Monotone degradation). Appending a stochastic node never lowers the bound: $B_{\text{out}} - B_{\text{in}} = d_v (1 - B_{\text{in}}) \geq 0$.

Proof. Expand (3): $1 - (1 - B_{\text{in}})(1 - d_v) = B_{\text{in}} + d_v(1 - B_{\text{in}})$. ■

Corollary (reasoning cannot buy back trust). Within the calculus, no arrangement of stochastic nodes, including an agreement rule over several model judges, can reduce B . Only a deterministic gate can.

This is a modelling commitment rather than a deep theorem: only gates establish properties, so a committee of models is one more stochastic step, whose errors are typically correlated with the steps it reviews. It is consistent with the architecture paper’s internal observation that 6 of 17 LLM-judge verdicts flipped across identical runs while the deterministic gates did not [1], and with the low step-level accuracy reported for model-based failure attribution [7].

Proposition 3 (When a re-execution gate helps). Let a segment after gate g contain nodes $v_1, \dots, v_n$, and let $B_k = 1 - (1 - \varepsilon_g) \prod_{j \leq k} (1 - d_{v_j})$ be the bound accumulated after v_k . Inserting a re-execution gate ρ after v_k changes the end-of-segment bound by

$$B - B' = (B_k - \varepsilon_\rho) \prod_{j > k} (1 - d_{v_j}). \quad (7)$$

Hence the gate lowers the bound if and only if $\varepsilon_\rho < B_k$: its own floor must be below the bound already accumulated at that point.

Proof. Write $B = 1 - (1 - B_k)P$ and $B' = 1 - (1 - \varepsilon_\rho)P$ with $P = \prod_{j > k} (1 - d_{v_j})$, and subtract. ■

In Example 1, a gate after the second step ($B_2 = 0.126$) lowers the final bound from 0.152 to 0.040; a gate after the first step lowers it only to 0.117. Equation (7) is what Section 9 uses to rank where a gate buys the most trust.

Assumption. The product form treats per-node divergences as approximately independent along a sequential path. Reconvergent branches and shared upstream inputs induce correlated divergence, so the bare product can misstate the bound; such segments need an interaction-aware estimator in the spirit of [3] (Section 11).

6. Calibration and graduated authorisation

The divergences d_v and gate floors ε are estimated, so B is an estimate and must be calibrated before it can carry an authorisation decision. We calibrate against *re-execution ground truth*: for a held-out set of findings, an independent re-run of the proof records whether φ actually held, $y_i = \mathbf{1}[\varphi_i = 0]$. This is the one place where the platform supplies a label without a model in the loop.

The calibration guarantee

Consider the policy “execute when $\hat{B} \leq \theta$ ”, fitted separately for each impact class, with a refused action treated as a no-op. Using conformal risk control [13] on the re-execution-labelled set, choose the operating threshold so that the realised failure rate among authorised-executed findings is held at a target δ :

$$\Pr [\varphi = 0 \mid \text{authorised to execute}] \leq \delta. \quad (8)$$

The guarantee is finite-sample and distribution-free under exchangeability of the calibration and deployment findings. It is marginal and honest about its limits: it does not claim that the estimated B equals the true failure probability pointwise, only that the authorisation policy holds the error among auto-

executed findings at δ . This is the calibrated, probabilistic guarantee that the phrase “provably bounded” refers to — a stated and tested limit on how often the system may be wrong when it acts, not a claim that any single step is deterministic.

The minimum calibration size

A conformal threshold at level δ needs enough labelled findings to place a cut below the target rate; below roughly $1/\delta$ usable labels the only threshold that satisfies (8) is the one that blocks everything [2] [13]. Rather than treat block-all as safety, DBA reports it as *vacuous*: too little evidence to automate, so every finding in that class routes to a human.

Example 2 (illustrative). At $\delta = 0.05$, a marginal conformal certificate needs on the order of $1/\delta - 1 = 19$ re-execution labels in an impact class before any finding in it can be auto-executed; at $\delta = 0.01$, about 99. Until then the class is auto-executed by no one, and the record says so, rather than quietly reporting zero errors on zero actions.

Definition 5 (Authorisation function). Let a be the action proposed for finding f , $\text{imp}(a)$ its impact class, S the authorised scope, and $\{\theta_{\text{imp}}\}$ an impact-indexed threshold schedule with higher-impact actions demanding a tighter bound. The authorisation function maps the calibrated bound, scope and policy onto the architecture paper’s three outcomes plus an explicit refusal:

$$\mathcal{A}(\pi) = \begin{cases} \text{execute} & B_\varphi(\pi) \leq \theta_{\text{imp}(a)} \wedge \text{scope}(\pi) \sqsubseteq S \wedge \text{policy permits} \\ \text{investigate} & \theta_{\text{imp}(a)} < B_\varphi(\pi) \leq \theta^+ \text{ and the bound is reducible at acceptable risk} \\ \text{escalate} & \text{a real hypothesis, but } B, \text{ scope or policy needs a person} \\ \text{refuse} & \text{scope}(\pi) \not\sqsubseteq S \vee \text{policy forbids} \end{cases} \quad (9)$$

with the schedule tightening as impact rises: higher $\text{imp}(a) \Rightarrow$ smaller $\theta_{\text{imp}(a)}$.

This makes the architecture paper’s execute / investigate / escalate the output of a measured function, and makes “defensible autonomy, not maximum autonomy” a quantity rather than a slogan: autonomy is granted exactly where the calibrated record supports it at the impact-appropriate threshold, and withheld where it does not.

Two caveats that travel with the certificate

Both are inherited from the enforceability analysis [2]. First, a bound computed on logged runs need not identify the closed-loop frontier once blocking changes which findings the agent produces next; the policy must be evaluated in the loop, not only on static logs. Second, the δ bound is a benign-calibration statement; an adversary who makes a false finding’s evidence look clean can raise the miss rate, so scope enforcement (Section 7) must remove the part of the surface that calibration alone cannot cover.

7. Deterministic scope enforcement

For an authorised-offence system, authorisation cannot be a prompt the model might reinterpret, because the whole safety story is that the system acts only within what it was authorised to touch. We make scope a property the system can prove, not one it is asked to maintain, by composing information-flow control with the bound.

Following the information-flow approach for agents [5], every piece of data and every proposed action carries labels: an integrity label recording its source and trust, and a scope label recording the authorised target and action class it belongs to. Labels propagate through reasoning and tool calls, and a deterministic reference monitor checks them before any irreversible action commits. The monitor is model-independent — a set-containment test — so injected text in a tool result cannot widen scope:

$$\text{execute}(a) \text{ permitted} \iff \text{label}(a) \sqsubseteq S \quad (\text{checked at the monitor, before commit}). \quad (10)$$

This is why the refuse branch of (9) is a hard gate rather than a soft threshold: $\text{scope}(\pi) \not\sqsubseteq S$ is decided by the monitor, not estimated by the bound. The division of labour is clean. The determinism bound answers “is this finding trustworthy enough to act on at this impact level?” The information-flow monitor answers “is this action inside what we were authorised to do?” A finding can be perfectly trustworthy and still refused for being out of scope; it can be in scope and still escalated because its bound is loose. Separating the two lets an auditor check each independently, and neither can be talked around by the model.

This is also the precise sense in which the system is “provably bounded.” **Scope is deterministic** — a label check that an injected instruction cannot move. **Trust is calibrated** — a stated limit δ on how often an in-scope action may be wrong. The marketing shorthand and the mathematics agree: deterministic scope, calibrated trust.

8. The auditable assurance record

A bound and an authorisation decision are only governable if an outside party can check them after the fact. Every authorised action therefore emits a defeasible record in the claims–arguments–evidence (CAE) form used by assurance-case practice [8] [20], specialised so that its evidence is independently re-executable rather than model-attested. Each record has three parts:

1. **Claim.** Property φ holds for finding f , within authorised scope S , at determinism bound $B_\varphi(\pi) \leq \theta$ for the action’s impact class.
2. **Argument.** The path decomposition (which gates were passed, the stochastic segments and their measured d_v), the conformal calibration certificate with its as-of date and δ , and the scope proof (the label trace the monitor checked).
3. **Evidence.** The re-executable check that reproduces φ with no model calls, plus the calibration data and the monitor log.

The record is *defeasible*, not a proof: it is a structured argument whose top claim holds only while its evidence and assumptions hold [20]. Three defeaters drop the claim and force the authorisation down to escalate or refuse:

- the calibration certificate is stale (its as-of date is older than the policy window, so δ is no longer warranted);
- re-execution of the evidence fails on audit (the validated check no longer reproduces φ);
- a required scope label is missing from the trace.

This directly answers the critique that behavioural assurance cannot verify what governance now demands [11] [12]: the evidence is a re-runnable mechanism, so an auditor does not take the system’s word — they re-execute the proof. The record is the unit a reviewer reads, an ISO/IEC 42001 audit samples, and an incident review reconstructs; its reproducibility, not a model’s confidence, carries the weight.

9. Composition, system-level assurance and path ranking

Per-finding records compose into an engagement-level claim. We do not average bounds into one reassuring number; we report the distribution, because one loose path can do the damage a good average hides. For an engagement the system-level summary is the worst bound among auto-executed findings, the full distribution of B , and the fraction routed to escalation:

$$B_{\text{sys}} = \max_{f : \mathcal{A}(\pi_f) = \text{execute}} B_{\varphi}(\pi_f). \quad (11)$$

Graceful degradation. When bounds are loose, the system does not ship low-trust findings as high-trust; it routes them to a person and still produces a valid, escalated output. The failure mode the framework exists to prevent — a fluent hypothesis leaving the system dressed as a validated condition — cannot occur silently, because a finding above its impact threshold is never auto-executed.

Path ranking for redesign. Because the bound is a product over a path’s stochastic segments, the framework localises *where* to intervene. By Proposition 3, the value of inserting a re-execution gate after segment s is the reduction it makes to the system bound, per unit cost:

$$\Delta_s = \frac{B_{\text{sys}} - B_{\text{sys}}[\rho \text{ inserted after } s]}{\text{cost}(\rho \text{ at } s)}. \quad (12)$$

Ranking segments by Δ_s turns the bound into a quality lever that can be pulled before release: find the flaky stretches, add a gate where it buys the most trust per model call, re-measure. This is the agentic-era descendant of architecture-based software reliability, where path-based models computed per-path reliability and ranked the modules most critical to the system [14]; here the components are non-deterministic and the ground truth is re-execution rather than a test oracle. It is also where the framework meets perturbation-sensitivity work [3], but grounded in re-execution outcomes and tied to an authorisation decision rather than to output quality alone.

The practical consequence is a release discipline: ship an engagement type, measure its paths’ bounds in the field, identify the worst stretches, add a gate exactly where it raises the engagement-level bound the most — a measurable, repeatable loop rather than a judgement call.

10. Governance mapping

DBA is best read not as a product feature but as a governance instrument: a measurable control that turns qualitative oversight requirements into computed, auditable quantities. Each component maps onto an existing governance hook, so the same engineering doubles as conformance evidence. For a vendor already holding ISO 27001 and ISO 9001, ISO/IEC 42001 is the adjacent standard, and these are concrete controls for it.

DBA COMPONENT	WHAT IT OPERATIONALISES	GOVERNANCE HOOK
Determinism bound (Sec. 5)	A measured, per-run trust quantity instead of a qualitative “evidence-gated” claim	NIST AI RMF Measure ; ISO/IEC 42001 performance evaluation; the measurable safety claim the behavioural-assurance critique [11] says governance now demands
Authorisation function (Sec. 6)	“When must a human intervene?” becomes a computed threshold, not a policy assertion	EU AI Act Article 14 human oversight; NIST AI RMF Manage ; the graduated-autonomy pattern the dual-use literature recommends but leaves as prose [16]
Scope enforcement (Sec. 7)	Authorised scope is provable and model-independent, closing the dual-use boundary	NIST AI RMF Govern (authorisation, least privilege); ISO/IEC 42001 operational controls
Assurance record (Sec. 8)	A re-executable, defeasible record an auditor can check after the fact	UL 4600 / ISO/IEC/IEEE 15026-2 assurance-case practice [20] [21]; ISO/IEC 42001 documented information and audit trail
Composition + path ranking (Sec. 9)	System-level assurance that degrades gracefully; a QA-before-release lever	UL 4600 autonomy-tier safety case; NIST AI RMF continuous Measure / Manage
Re-execution ground truth (Secs. 5, 8)	Mechanism-level evidence that does not rely on model attestation	The evidence-sufficiency standard governance-evidence work calls for [12]; independent verifiability for conformity assessment

Two framing points follow. First, the EU AI Act’s human-oversight requirement is, in most deployments, met by asserting that a human is in the loop; DBA lets a deployer instead show *the rule* by which oversight is triggered and *the evidence* that it fired. Second, for an authorised-offence system the governance gap is sharpest precisely because autonomy means autonomous action, and existing assurance models were built for non-autonomous, short-lived systems [15]; a bound-plus-record that makes authorisation computable and checkable is the control that gap has been missing.

11. Limitations and threats to validity

The multiplicative form assumes approximate independence. The segment bound composes per-node divergences multiplicatively, which holds for sequential, weakly coupled paths. Reconvergent branches and shared upstream inputs induce correlated divergence, so the product can over- or under-state the true bound; those segments need an interaction-aware estimator [3]. Calibration failure is the symptom, and this assumption is the first suspect.

The certificate is marginal, and the loop can move the frontier. Conformal calibration gives a marginal δ guarantee under exchangeability, not a pointwise match to the true failure probability, and a bound estimated on logged runs need not identify the closed-loop frontier once blocking changes later proposals [2]. The honest claim is a bound, not a proof of safety.

Calibration data can be scarce. Re-execution labels are expensive, and below the minimum count conformal degenerates to block-all. We report block-all as vacuous rather than as safety.

Re-execution ground truth is not universal. The framework is strongest for properties that can be re-executed deterministically — reproducibility, a check that reruns. Properties that resist deterministic replay (a judgement about business impact, a context-dependent severity) fall back to a calibrated fallible oracle

with a measured error profile, and their bounds are weaker. The method bounds trust in an asserted property given the path; it does not bound recall — whether the engagement found everything it should have — which is a separate guarantee.

Gates and labels are not error-free. ε_p is not zero: re-execution can itself be flaky (environment drift, non-deterministic targets), and the floor must be measured, not assumed. Scope enforcement is only as sound as the labelling at the boundary; a mislabelled input is a real failure mode, and label integrity is itself an assurance sub-claim.

Generality is claimed, not shown. We argue the bound-and-record structure should serve other high-consequence autonomy — medical, financial — but this paper demonstrates it only in security validation, which is where re-execution is cleanest. Cross-domain transfer is future work.

12. Open problems and roadmap

Open problems.

1. **The reconvergent-path bound.** Replace the sequential product with an interaction-aware estimator for correlated and parallel segments, and characterise when the product is a safe over-approximation.
2. **Telemetry that predicts trust.** The hardest and most valuable question, shared with the architecture paper: find technical signals that actually predict when a belief should be trusted, so d_v is grounded in evidence rather than output variability alone. The transfer gap in [1] is exactly this problem unsolved.
3. **Evidence families, not an evidence scalar.** Enrich d_v from a single divergence number to typed evidence families — provenance, freshness, reproducibility, state transitions, negative controls — so the bound reflects *why* a property is trustworthy, not just how stable the output was.
4. **Robust calibration.** Extend the conformal certificate to hold against a bounded adversary on the evidence, with a verified robustness margin rather than benign calibration alone.
5. **Cross-domain transfer.** Instantiate the same bound-and-record structure in a second high-consequence domain to test portability.

Roadmap. This paper is the concept and the formal core. The planned work, to be reported separately, is to (1) implement the degradation-replenishment calculus as instrumentation over the existing platform; (2) build the re-execution-labelled calibration set from an anonymised engagement corpus and fit the conformal certificate; (3) run the authorisation decision against baselines — no gate, a single model judge, a model committee — in the loop and in authorised isolated labs; (4) wire scope labels into the reference monitor and red-team them; and (5) pilot the record as an ISO/IEC 42001 control and map an EU AI Act Article 14 oversight argument onto the authorisation function. The experiments are deferred so that they can be run against the real ATVE workflow rather than a stand-in, and reported with measured values in place of the illustrative ones used here.

13. Conclusion

The architecture paper argues that an autonomous security system should refuse to treat a guess as a finding. This paper gives that refusal a shape you can check: certify the execution path, not the finding; set a determinism bound at gates, let it degrade through reasoning, and replenish it by re-execution; calibrate it against re-execution ground truth so that the authorisation decision carries a stated error limit; enforce scope deterministically so the boundary cannot be argued away; and emit a record whose evidence re-runs.

Deterministic scope, calibrated trust. That is what “provably bounded autonomy” means, and it is why re-execution — not a committee of models — is the operator the whole design rests on.

Cite as: P. Taeprasartsit and V. Khera, “Determinism-Bounded Authorization: A Measurable, Auditable Foundation for Evidence-Gated Security-Validation Autonomy.” SECStrike, concept note, 2026. Companion to the EchoStrike White Paper, v1.1.

This is a concept and preliminary-work paper. It states a framework and its formal properties; it reports no experimental results. Numerical values in examples are illustrative. Capabilities described as research or planned are not shipped product features. The work builds on the EchoStrike architecture white paper and is kept consistent with the commercial claims made elsewhere on this site.

14. References

Reference [1] is the primary application foundation. The remaining entries were verified against primary sources; author lists are given in full where short.

1. EchoStrike White Paper, v1.1, 2026 (SECStrike; public architecture paper). echo-strike.com/whitepaper.html.
2. S. Ray. *What Can Be Enforced? A Theory of Certified Runtime Safety for Tool-Using Agents*. arXiv:2607.22868, 2026.
3. P. Nilayam and S. Nayak. *QUIVER: A Formal Framework for Quantifying Perturbation Propagation and Bifurcation in Compound AI Systems*. arXiv:2605.23956, 2026.
4. H. Wang, C. M. Poskitt, J. Wei and J. Sun. *ProbGuard: Proactive Runtime Monitoring for LLM Agent Safety via Probabilistic Prediction* (first released as *Pro2Guard*). arXiv:2508.00500, 2025; ASE 2026.
5. M. Costa et al. *Securing AI Agents with Information-Flow Control* (FIDES). arXiv:2505.23643, 2025.
6. J. Duan et al. *UProp: Investigating the Uncertainty Propagation of LLMs in Multi-Step Agentic Decision-Making*. arXiv:2506.17419, 2025.
7. S. Zhang et al. *Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems*. arXiv:2505.00212, 2025; ICML 2025.
8. A. Goemans, M. D. Buhl, J. Schuett, T. Korbak, J. Wang, B. Hilton and G. Irving. *Safety Case Template for Frontier AI: A Cyber Inability Argument*. arXiv:2411.08088, 2024.
9. T. Korbak, J. Clymer, B. Hilton, B. Shlegeris and G. Irving. *A Sketch of an AI Control Safety Case*. arXiv:2501.17315, 2025.
0. J. Clymer, N. Gabrieli, D. Krueger and T. Larsen. *Safety Cases: How to Justify the Safety of Advanced AI Systems*. arXiv:2403.10462, 2024.
1. P. Seth and V. K. Sankarapu. *Position: Behavioural Assurance Cannot Verify the Safety Claims Governance Now Demands*. arXiv:2605.15164, 2026.

2. O. Solozobov. *DEMM-Bench: A Cross-Regime Benchmark for Agent-Runtime Governance-Evidence Sufficiency*. arXiv:2606.20634, 2026.
3. A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei and T. Schuster. *Conformal Risk Control*. ICLR 2024; arXiv:2208.02814.
4. R. C. Cheung. *A User-Oriented Software Reliability Model*. IEEE Transactions on Software Engineering, vol. SE-6, no. 2, pp. 118–125, 1980.
5. S. J. Lazer, K. Aryal, M. Gupta and E. Bertino. *A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes*. arXiv:2601.05293, 2026.
6. *Regulating Dual-Use AI in Cyber Operations*. Journal of Law & Cyber Warfare, 2026.
7. R. Csernaton and P. Pawlak. *When AI Agents Attack: Autonomous Cyber Operations and Europe's Governance Gap*. Carnegie Europe, 2026.
8. V. Khera, R. Sunil, P. Mer, N. Kointarangkul and A. Diwan. *Towards Adaptive and Continuous Offensive Security: A Comprehensive Survey of AI-Driven Threat Validation Techniques*. Array (Elsevier), vol. 30, 100843, 2026.
9. ISO/IEC 42001:2023, *Information technology — Artificial intelligence — Management system*.
10. ISO/IEC/IEEE 15026-2:2022, *Systems and software engineering — Systems and software assurance — Part 2: Assurance case*.
1. UL 4600 (Edition 3, 2023), *Standard for Evaluation of Autonomous Products*.
2. Regulation (EU) 2024/1689 (EU AI Act), Article 14 (Human oversight); NIST, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, 2023.