Error Budgets for Autonomy: SLO-Driven Authority Management for Autonomous AI Operators

Ajin Baby

cloudandsre.com

2026-09-01 · v0.3 · cloudandsre.com/research/error-budgets-for-autonomy/

Abstract

AI operators are being granted authority over production infrastructure—restarting services, scaling fleets, rolling back deployments. The emerging runtime-governance literature gates that authority *per action* (pricing side-effects against reserve capital, enforcing policies on execution paths) or *per static tier* (fixed oversight levels). Both are point-in-time controls: they ask whether *this* action is permissible *now*, and neither adjusts an operator's standing authority in response to its measured track record. Site reliability engineering solved the structurally identical problem for software releases two decades ago with the **error budget**: a quantified reliability allowance whose exhaustion automatically throttles risk-taking. We transplant error-budget semantics to autonomous authority. We contribute (1) **Autonomy SLOs** and the **Autonomy Error Budget (AEB)**—a per-action-class budget that wrong or harmful actions burn in proportion to their blast radius, not merely their count; (2) the **Authority Ladder**, a formal autonomous→supervised→advisory→disabled state machine with deliberately asymmetric transitions (demote fast on budget burn, promote slowly on sustained evidence) and burn-rate alerting on authority itself; (3) composition semantics with per-action governance—`effective authority = min(per-action gate, budget-derived authority)`—making the budget the *longitudinal* layer above existing point-in-time gates; and (4) a **measured simulation study** (companion library `autonomy-budget`, 20 seeds per cell) confirming all three hypotheses: static tiers are U-shaped in cost across operator reliability while budget-governed authority tracks a full-knowledge oracle within ~8% at both extremes—3.9× cheaper than a static-autonomous tier under a silent step degradation; burn-rate demotion detects silent degradation 2.7–5.1× faster than monthly track-record review, with latency set by action volume and budget geometry rather than review cadence; and the fast-down/slow-up asymmetry is load-bearing—a symmetric variant oscillates (16 vs 1.4 authority transitions) and more than doubles realized harm under 2% adjudication noise. The study also surfaced two budget-semantics rules the framework now carries: the allowance must be denominated in expected action volume, and supervised-mode wrongness feeds promotion evidence, not the budget. A fourth experiment prices the budget's own data plane: a **constant** adjudication lag shifts detection latency by exactly the lag, seed for seed, and leaves the fast-burn alert's ~40 h advantage over window exhaustion intact, while a **jittered** pipeline at the same 72 h mean leaves mean detection nearly unchanged but degrades the median 15% and widens dispersion 36% — adjudication freshness is itself a governance SLO, with separate bounds on mean and spread. The thesis: autonomy is not a configuration setting; it is a budgeted resource, earned and spent against measured reliability.

Keywords: autonomous operations, error budgets, SLO, agent governance, authority management, AIOps, site reliability engineering.

1. Introduction

When an organization deploys an autonomous operator, it makes a decision it will rarely revisit: how much authority the operator holds. In practice that decision is encoded as static configuration—an allowlist of actions, an approval tier, a feature flag—set at deployment time by the most cautious stakeholder in the room, and thereafter drifting out of sync with reality in both directions. Operators that have performed flawlessly for months retain training-wheel restrictions that route trivial remediations through human queues; operators whose behavior has quietly degraded (a model update, a context change, an environment shift) retain authority they should have lost weeks earlier.

The runtime-governance literature is filling in the *point-in-time* half of this problem. Actuarial approaches price each side-effect-bearing action against a reserve budget at execution time [1]; contract frameworks bound resources per run [2]; policy engines constrain the execution paths an agent may take [3]; graduated-oversight schemes assign fixed human-review tiers for regulated domains [4, 5]. What none of these provides is the *longitudinal* half: a principled mechanism by which an operator's standing authority expands and contracts as a function of its measured operational reliability over time.

SRE already owns the right abstraction. An error budget [6] converts a reliability target into a spendable allowance and —critically—couples the allowance to an *automatic consequence*: when the budget is exhausted, releases stop. No committee meets; the control is mechanical, pre-agreed, and symmetric in its legitimacy (a healthy budget is equally mechanical permission to move fast). We argue the same construct is the missing governance layer for autonomous authority, and this paper works out the transplant in full. Our contributions:

1. **Autonomy SLOs and the Autonomy Error Budget** (§4): definitions of action classes, autonomy SLIs, blast-radius-weighted burn accounting, and budget windows.
2. **The Authority Ladder** (§5): a four-state authority machine with asymmetric (fast-down, slow-up) transitions, hysteresis, and burn-rate alerting on authority.
3. **Composition semantics** (§6): how budget-derived authority layers over per-action gates via a single `min` rule.
4. **A measured simulation study** (§7): operator populations under static-tier, per-action-gate, and AEB-governed authority, measuring total incident cost, time-at-appropriate-authority, and detection latency; all three hypotheses confirmed, on a released companion library.

2. Background

Error budgets in SRE. An SLO (e.g., 99.9% successful requests over 30 days) implies a budget (0.1% may fail). The budget reframes reliability from a moral question ("did we fail?") into a resource question ("how much failure allowance remains, and what does that permit?"). Burn-rate alerting [6] detects when the budget is being consumed anomalously fast, triggering intervention before exhaustion. Two properties make the construct powerful: *automatic consequence* (exhaustion throttles risk without human relitigation) and *pre-agreement* (the policy is negotiated once, calmly, not per-incident).

Why autonomy is the same shape. An autonomous operator's wrong actions are the failure events; its authority level is the risk-taking being throttled; and the deployment-time authority decision is exactly the "release approval by committee" pattern error budgets were designed to replace. The mapping is close enough that most of SRE's operational machinery—windows, burn rates, policy documents—transfers with only domain-specific reweighting.

3. Related Work

Per-action and per-tier governance. The Authority Frontier framework prices each side-effect-bearing action against reserve capital under a time-consistent risk mapping [1]; Agent Contracts formalize resource-bounded execution [2]; path-policy enforcement constrains admissible action sequences at runtime [3]; graduated-oversight models fix human-review tiers for agentic work in regulated settings [4], and architectural-tactics work couples permissible agency to autonomy level in regulated contexts [5]. All are complementary: they answer *"may this action execute?"* We answer *"how much standing authority should this operator hold this week?"*—a question whose input is history, not the current action.

Agent reliability measurement. Agent-level reliability frameworks contribute the measured quantities—wrong-action rates, containment, failure amplification—that an autonomy budget spends against; this paper consumes those metrics rather than redefining them. **Trust and evidence.** Pre-deployment and shadow-mode evaluation methodologies generate the *evidence* on which promotion decisions rest; we define the budget semantics those evidence streams feed. ## 4. Autonomy SLOs and the Autonomy Error Budget

4.1 Action classes. Authority is not monolithic. We partition an operator's action space into **action classes** by blast radius and reversibility (e.g., *read/diagnose*, *reversible mitigations*, *service-impacting mutations*, *irreversible or data-destructive actions*). Each class carries its own SLO, budget, and ladder position; an operator can be autonomous for reversible mitigations while advisory for irreversible ones. This mirrors per-action risk pricing [1] but at the level of standing policy.

4.2 Autonomy SLIs. For each class we define the SLI as the fraction of executed actions that were *operationally correct*: the action addressed the actual condition, did not violate policy, and did not require reversal or human correction. Determining correctness is nontrivial and asynchronous (some wrongness surfaces hours later); we treat SLI adjudication as an explicit pipeline stage—automated where outcomes are machine-checkable (did the remediation clear the alert without recurrence or rollback?), human-adjudicated where they are not, with adjudication latency bounded so the budget reflects a recent window (§7 E4 measures what that bound buys: each hour of mean lag is exactly one hour of detection latency, and lag *variance* additionally degrades the median and tail).

One attribution rule emerged from the simulation study as load-bearing: **only actions executed under the operator's autonomous authority burn its budget.** A wrong proposal approved by a human executes under the *human's* authority; it enters the promotion-evidence stream (and, in incident analysis, the handoff layer) but does not burn the operator's budget. Burning human-approved executions creates a demotion spiral: an operator demoted to supervised can then never regain budget health at any rung below autonomous, and ratchets toward advisory even where supervised is the cost-optimal assignment. We measured exactly this failure mode under naive burn semantics (§7)—governed cost for a chronically unreliable operator was 5× the oracle's until the attribution rule was applied.

4.3 Blast-radius-weighted burn. Counting wrong actions equally is miscalibrated: a wrong read costs nothing; a wrong failover costs an outage. Burn for a wrong action a is weighted by its realized (or, where unrealized, its class-default) blast radius: `burn(a) = w_class · severity(a)`. A budget therefore encodes *harm allowance*, not error count. This is the principal domain adaptation from classical error budgets, where all failed requests are near-fungible.

A second, quieter adaptation is equally load-bearing: **the allowance must be denominated in expected weighted action volume per window**—`allowance = (1 − SLO) × E[weighted actions per window]`—not set as an absolute harm constant. Classical budgets get this for free because they are defined as a fraction of requests; transplants that fix an absolute allowance make the same per-action reliability exhaust the budget N× faster at N× action volume. In the §7

study, a fixed allowance calibrated at 0.5 incidents/hour was spuriously exhausted by a *healthy* operator at 2 incidents/hour purely on volume.

4.4 Budget windows and decay. Budgets are computed over rolling windows (e.g., 30 days) so that both good and bad history age out. A long-past incident should not permanently cap an operator, and stale good behavior should not indefinitely fund present authority—particularly across *evidence-invalidating events* (model or prompt version changes), which we treat as partial budget resets because prior track record only weakly predicts post-change behavior [7].

5. The Authority Ladder

5.1 States. Per action class, authority occupies one of four rungs: **Autonomous** (act within policy, notify), **Supervised** (propose; a human approves), **Advisory** (surface assessment only), **Disabled** (not consulted). The rungs deliberately match the degraded-authority tiers emerging elsewhere in trust-calibration work; what is new is the *transition function*.

5.2 Asymmetric transitions. Demotion is immediate and mechanical: budget exhaustion, or a fast-burn alert (e.g., budget consumption rate exceeding 10× nominal over an hour), drops the operator one or more rungs without human deliberation. Promotion is slow and evidence-gated: a rung up requires the budget healthy for a full window *plus* affirmative evidence at the target authority level—supervised-mode concordance or shadow-mode performance—so that promotion rests on demonstrated competence at the next rung, not merely absence of recent failure. The asymmetry is a stability choice as much as a safety one: fast-down/slow-up acts as hysteresis, preventing authority oscillation under noisy SLIs.

5.3 Burn-rate alerting on authority. Classical burn-rate alerts page a human about a service. Here they *act on the operator first* (demote) and page second. The alert taxonomy transfers directly: slow burns indicate drift (context rot, environment change) warranting investigation; fast burns indicate acute breakage (bad model update, poisoned input) warranting freeze. That the same mechanism catches both is inherited from error budgets' agnosticism about failure cause.

5.4 The policy document. As with release error budgets, the enforcement lives in a pre-agreed **autonomy budget policy**: who is paged on exhaustion, what evidence promotes, which events reset budgets, and who may override (with overrides logged as budget-relevant events themselves). The document is negotiated once—not during an incident.

5.5 The ladder is itself a control loop. Read through the lens of remediation-loop stability analysis [9], the budget-and-ladder mechanism is an outer supervisory control loop: its plant is the operator's standing authority, its sensor is the adjudication pipeline (whose dead-time §7 E4 prices), its actuator is rung movement, and §5.2's asymmetric transitions are stability constructs applied at the governance layer — the promotion window is a settling-time cooldown, the evidence gate is hysteresis. E3 measures the instability that appears when they are removed: the symmetric fast-up variant oscillates (16.1 vs. 1.4 authority transitions) exactly as an undamped inner loop does in the taxonomy of [9]. The classical cascade rule follows for autonomy governance: **the governing loop must run on a slower timescale than the loops it governs.** A promotion window shorter than the evidence settling time re-promotes into unsettled evidence the way a remediation loop with no cooldown re-acts into its own dead-time.

6. Composition with Per-Action Governance

Budget-derived authority does not replace per-action gates; it bounds them:

```
effective authority(a) = min( per-action gate(a),        # policy, pricing, integrity gates
                              budget-derived authority(class(a)) )
```

The rule composes cleanly with runtime pricing [1], path policies [3], and input-trust gating, where authority is capped by the integrity of the telemetry feeding the decision [10]: each mechanism computes an admissible authority from its own evidence axis—the action's risk, the path's policy, the input's trustworthiness, and (ours) the operator's history—and the executed authority is the minimum. We conjecture this *min-of-justifications* structure is the natural composition law for autonomy governance generally: a system should never act with more authority than its weakest current justification supports.

The §7 study gives the conjecture empirical teeth: composition beats either layer alone. For a chronically unreliable operator, the per-action gate alone (no longitudinal layer) incurred 2.3× the composed regime's cost—it kept routing major-class actions to review while wrong minor and moderate actions executed forever. And for a reliable operator, the composed regime *beat the rung-level oracle itself* (5.9 vs 6.8 cost units), because the per-action gate operates within a rung at finer granularity than any rung assignment can. Composition, not choice between mechanisms, is the point. ## 7. Simulation Study (Measured)

Methodology. A discrete-time simulation (companion artifact [8]; one tick = one hour). Incidents arrive stochastically at rate λ per hour; the operator proposes one action per incident, wrong with probability p(t) given by its trajectory—constant, step change at a fixed tick (a silent model update), or linear drift (context rot). Blast radius is drawn from a three-class mix (minor 0.2 / moderate 1.0 / major 5.0 harm units at 70/25/5%; E[severity] = 0.64). Costs: executed wrong actions realize their severity as harm; human review of a proposal costs 0.02; full manual handling of an incident costs 0.3; humans catch wrong proposals with probability 0.9. Budget: autonomy SLO 97% weighted correctness over a 720 h window, allowance volume-denominated per §4.3, fast-burn alert at 10× nominal over 24 h; ladder promotion requires 720 h of budget health, equal dwell time, and trailing concordance ≥ 0.97 over ≥ 50 adjudicated proposals. Four regimes are compared: static tier (autonomous or supervised, frozen at deployment), per-action gating alone (major-class actions always reviewed; no longitudinal layer), the composed AEB + Authority Ladder regime (§6), and a full-knowledge **oracle** that assigns the cost-minimizing rung given the true p(t) at every tick. 20 seeds per cell; the qualitative claims are pinned by the artifact's 30-test suite.

E1 — static tiers are U-shaped; the budget tracks the oracle (H1 confirmed). Table 1 reports total cost (harm + toil, mean ± sd) over 90 days at λ = 0.5.

Table 1 — total incident cost by regime and operator archetype (20 seeds; lower is better).

Operator	Static-autonomous	Static-supervised	Per-action gate	AEB + Ladder	Oracle
Reliable (p = 0.01)	6.8 ± 3.7	23.2 ± 2.1	5.9 ± 2.3	**5.9 ± 2.3**	6.8 ± 3.7
Borderline (p = 0.05)	36.7 ± 9.1	25.8 ± 4.3	23.0 ± 4.9	**24.8 ± 5.0**	25.8 ± 4.3
Drifting (0.01 → 0.15)	55.8 ± 11.7	28.3 ± 4.8	37.0 ± 5.7	**25.8 ± 4.9**	26.9 ± 5.0
Step-degrading (0.01 → 0.30)	139.9 ± 16.0	38.3 ± 5.1	92.3 ± 6.6	**35.9 ± 5.4**	33.3 ± 5.8
Unreliable (p = 0.20)	138.3 ± 15.7	36.6 ± 6.3	90.1 ± 8.3	**39.8 ± 6.3**	36.6 ± 6.3

The static columns show the predicted U: static-autonomous is optimal for the reliable operator and catastrophic for the degraded ones (3.9× the AEB cost under step degradation); static-supervised pays a flat 3.9× over-restriction toil for the reliable operator. The composed AEB regime tracks the oracle within ~8% at both extremes and holds time-at-appropriate-authority at 0.94–1.00 for the archetypes where a single rung is clearly right (reliable, step-degraded, unreliable). Its weakest cell is honest: for the borderline operator sitting almost exactly on the autonomous/supervised

cost boundary, rung assignment oscillates near the threshold (TAA 0.51)—yet total cost stays within 4% of the oracle, because the adjacent rungs cost nearly the same there by construction.

E2 — burn-rate demotion beats periodic review, governed by volume not cadence (H2 confirmed). A silent step degradation (p: 0.01 → 0.30) is injected at a phase jittered uniformly across a 30-day review period, and detection latency is measured for AEB demotion (fast-burn or exhaustion) versus a monthly track-record review that detects at the first review after the step (expected latency = cadence/2).

Table 2 — detection latency for silent step degradation (h; 20 seeds, all detected).

Incidents/h	AEB demotion	Periodic review (30 d)	Speedup
0.1	139 ± 108	378	2.7×
0.5	74 ± 43	378	5.1×
2.0	84 ± 24	378	4.5×

Budget-derived detection ran 2.7–5.1× faster than monthly review across a 20× action-volume range, detecting 20/20 degradations. The structure matters more than the point values: AEB latency is set by budget geometry and action volume (mean ~75 h at moderate volume, variance shrinking as volume grows; at very low volume, sparse wrong-action arrivals dominate and variance is large), while review latency is set by cadence alone. Matching AEB's detection latency with human review would require reviewing every operator's track record roughly weekly—forever.

E3 — the asymmetry is load-bearing (H3 confirmed). Under 2% adjudication noise, a borderline operator (p = 0.045) governed by the standard asymmetric ladder (720 h promotion window + evidence gate) versus a symmetric fast-up variant (72 h, no evidence gate), over 180 days:

Variant	Authority transitions	Realized harm	Total cost
Asymmetric (fast-down/slow-up)	**1.4 ± 0.8**	**11.1 ± 3.0**	48.1 ± 3.0
Symmetric (fast-up)	16.1 ± 3.2	28.0 ± 3.7	42.8 ± 4.6

The symmetric variant oscillates (11× the transitions) and incurs 2.5× the harm. The total-cost row is deliberately reported: for a *constant* borderline operator the symmetric variant's total is slightly lower, because time spent autonomous is near cost-optimal there—it buys that discount with concentrated harm and authority churn. The trade disappears when degradation is real: for the step-degrading operator the symmetric ladder re-promotes into the damage every 72 h (24.6 ± 5.5 transitions) and is worse on *both* axes—harm 52.4 vs 30.7, total cost 78.2 vs 61.5. Asymmetry is a small steady-state toil premium purchased against exactly the scenario budgets exist for.

E4 — the price of adjudication latency (new in v0.3). The study so far adjudicates instantly; production pipelines do not. E4 injects an adjudication lag between an action executing and its verdict reaching the budget and evidence stream — verdicts are still *decided* at execution; the pipeline is slow, not wrong — and re-measures E2's detection scenario (step p: 0.01 → 0.30, 0.5 incidents/h, 20 seeds). The exhaustion-only column disables the fast-burn alert, isolating what the fast path contributes.

Table 4 — detection latency vs. adjudication lag (h; 20 seeds).

adjudication lag	full AEB	p50 / p90	exhaustion-only
0 (baseline)	74 ± 43	69 / 137	112 ± 31
24 h constant	96 ± 44	91 / 160	135 ± 31
72 h constant	144 ± 44	139 / 208	183 ± 31
168 h constant	240 ± 44	235 / 304	279 ± 31
72 h jittered (sd ≈ 26 h)	156 ± 54	159 / 235	188 ± 38
72 h jittered (sd ≈ 68 h)	146 ± 60	160 / 225	183 ± 40

Three results. **Constant lag is exactly additive.** A constant lag L translates the burn process the budget observes, so every threshold crossing — and therefore detection — shifts by exactly L, *seed for seed* (up to a one-tick delivery convention): 19/20 seeds shift by precisely L−1 at every L tested, and the twentieth is a measurement artifact — a spurious pre-step demotion that translation carries across the step boundary, where the latency definition picks it up — not a budget dynamic. **The fast path survives constant lag.** The fast-burn alert detects the step 39–40 h ahead of window exhaustion on average, and that advantage is invariant across constant lags: translation delays the burst concentration it alerts on without smearing it. **Lag variance costs predictability, not average speed.** A jittered pipeline at the same 72 h mean leaves mean detection nearly unchanged (146 vs. 144 h — the mean-preserving lognormal's early tail compensates for its late one) but shifts the median up 15% (139 → 160 h), the p90 up 8–13%, and widens dispersion 36% (sd 44 → 60 h); the seeds where fast-burn contributes nothing rise from 5/20 to 8/20. The freshness rule this yields: **treat adjudication latency as a governance SLO with two bounds** — bound the *mean*, because every hour of pipeline lag is provably an hour of detection latency, and bound the *spread*, because a predictable pipeline preserves the fast-burn path where an erratic one turns it into a lottery. Even the slowest cell measured — a week of adjudication lag — still detects 1.6× faster than E2's monthly periodic review (240 vs. 378 h expected).

8. Discussion and Threats to Validity

Adjudication is the load-bearing dependency. The budget is only as honest as the SLI pipeline that decides which actions were wrong; gaming, delayed adjudication, or survivorship effects (wrong actions that were silently reversed by humans and never recorded) all inflate apparent reliability. This is an argument for making adjudication an audited platform function, not an operator self-report. **Low-volume classes.** Irreversible-action classes may see too few executions for statistically meaningful budgets; for these, promotion evidence must come from shadow and supervised modes rather than autonomous track record—which is precisely the intended coupling to pre-authority evaluation methodologies. **Weight calibration.** Blast-radius weights encode organizational risk tolerance; miscalibrated weights re-create the miscalibrated-static-tier problem inside the budget. We treat weight review as part of the policy document's periodic renegotiation. **Multi-operator interaction.** Budgets are per-operator; two operators sharing actuators can jointly cause harm neither's budget attributes cleanly. We flag cross-operator attribution as future work, connecting to loop-interaction analyses in self-healing stability work. **Simulation fidelity.** The §7 study uses a stylized cost model (one action class with a three-point severity mix; constant human catch probability; adjudication instantaneous except where E4 varies it) and synthetic operator trajectories, not an LLM-driven operator; its claims are about the *governance dynamics*—U-shaped static cost, detection-latency structure, oscillation under symmetric transitions—which depend on the budget/ladder mechanics rather than on how wrongness is generated. The oracle is defined against the same cost model the regimes are scored on, so oracle-relative results are relative, not absolute, claims. Adjudication latency is no longer assumed away: E4 measures a constant lag as exactly additive to every detection latency, and lag variance as a further tax on the median and tail — the reason §4.2 bounds both in the SLI pipeline.

9. Conclusion and Future Work

Autonomy governance today is either frozen at deployment time or adjudicated one action at a time. Error budgets supply the missing middle: a longitudinal, mechanical, pre-agreed coupling between an operator's measured reliability and its standing authority. The Autonomy Error Budget and Authority Ladder transplant SRE's most culturally successful control loop onto agent governance, compose with existing per-action mechanisms through a single `min` rule, and turn "should we trust the agent more yet?" from a standing argument into a query against a dashboard. The simulation study confirms the transplant's dynamics—and sharpened its semantics: volume-denominated allowances, evidence-not-budget attribution for supervised actions, and an adjudication-freshness SLO with separate bounds on mean and spread (E4) are now part of the framework, all learned by watching naive semantics fail or by measuring what the naive assumption hid. Next steps: validate the budget/ladder against traces from a production supervised-mode deployment rather than synthetic trajectories; extend the study to multi-operator populations sharing actuators; and integrate AEB state as first-class metadata in agent-platform catalogs.

References

(Author lists and identifiers verified against arXiv metadata and live URLs, 2026-09-01.)

[1] H.-H. Chen. Insuring Every Action: An Authority Frontier Framework for Runtime Actuarial Control of Autonomous AI Agents. arXiv:2605.25632, 2026. [2] Q. Ye and J. Tan. Agent Contracts: A Formal Framework for Resource-Bounded Autonomous AI Systems. arXiv:2601.08815, 2026. [3] M. Kaptein, V.-J. Khan, and A. Podstavnychy. Runtime Governance for AI Agents: Policies on Paths. arXiv:2603.16586, 2026. [4] R. Kang. Governed AI-Assisted Engineering: Graduated Human Oversight for Agentic Code Generation in Regulated Domains. arXiv:2606.22484, 2026. [5] D. Safin and D. Balta. Autonomy and Agency in Agentic AI: Architectural Tactics for Regulated Contexts. arXiv:2605.12105, 2026. [6] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy (eds.). *Site Reliability Engineering: How Google Runs Production Systems.* O'Reilly Media, 2016. (Error budgets: Ch. 3; alerting on burn rate: *The Site Reliability Workbook*, O'Reilly, 2018, Ch. 5.) [7] M. S. Chishti, D. P. Oyinloye, and J. Li. Test Before You Deploy: Governing Updates in the LLM Supply Chain. arXiv:2604.27789, 2026. [8] A. Baby. autonomy-budget: Autonomy Error Budget and Authority Ladder — Reference Implementation and Simulation Study. Companion artifact, 2026. https://github.com/ajinb/autonomy-budget (public, Apache-2.0). [9] A. Baby. Stable by Design: A Control-Theoretic Account of AI-Driven Self-Healing Remediation Loops. Preprint, cloudandsre.com, 2026. https://cloudandsre.com/research/stable-by-design-remediation-loops/ [10] A. Baby. Your AI agent can't tell a quiet system from a broken collector. cloudandsre.com, 2026. https://cloudandsre.com/blog/telemetry-integrity-trust-calibrated-autonomy/