
Stopping Dispatch Does Not Revoke Authority: A Circuit-Breaker Conformance Profile for AI Incidents

Vishwa Kumaresh

Cherishma Subhasa

Samridhi Makkar

With
Apart Research

AI Incident Response Sprint, September 2026 | Track 1: Containment

Abstract

Pausing an AI agent can leave previously authorized tool calls executable. We develop an Incident Circuit Breaker conformance profile combining sink-side epoch checks, a shared exposure budget, and a separate forensic channel. An executable reference broker, independent receipt checker, and fault-injection harness test this contract. Across 1,000 synthetic queued-action trials, a dispatch-only halt permits all 4,473 stale commits; sink fencing permits none. The full profile completes all 912 benign tasks within its budget, but only 91.2% of the unfiltered task set. A finite model explores 72 states and 162 transitions without a budget violation; a deliberately non-atomic variant admits 16 reservations against a budget of 12. We prove conditional revocation and exposure bounds, and specify evidence a third party can request. These results establish local conformance, not production containment or prevention of a historical intrusion. The contribution is a falsifiable incident-response contract built from established security mechanisms.

1. Introduction

An incident responder presses “stop,” yet a worker executes a tool request it received earlier. The planner has stopped producing new work; the authority already distributed to executors remains usable. The relevant question is therefore not only whether a detector fires, but *which effects can still occur after the response begins*.

This project targets the sprint’s containment track. Its artifact is an **Incident Circuit Breaker (ICB) conformance profile**: a small set of operational obligations, executable counterexamples, and an evidence format for assessing an agent deployment without requiring the evaluator to inspect the laboratory’s private network. The intended users are evaluation-infrastructure engineers and independent containment assessors.

The contribution is a testable contract for revocation, bounded exposure, and retained forensic access; conditional proofs and local evaluation; and an adoption checklist separating broker conformance from deployment coverage.

2. Related Work

Incident motivation. Hugging Face’s reconstruction describes an agent campaign that crossed an evaluation boundary, an external launchpad, and its production infrastructure. Its technical account identifies a package-proxy escape and later production-loader file disclosure and code execution; OpenAI’s account attributes the campaign to models in an internal evaluation [2, 5]. These sources motivate checking the *transitive* tool and credential boundary. We do not replay their commands, infer a detector’s historical recall, or claim our prototype would have stopped the incident.

Runtime rules and response. AgentSpec expresses triggers, predicates, and enforcement mechanisms for agent behavior [6]. AIR explicitly integrates detection, containment, recovery, and eradication into agent execution [7]. These works already establish the relevance of runtime rules and incident lifecycles. ICB instead isolates a lower-level question: can an independent enforcement path make previously granted authority unusable, even if the agent ignores the response instructions? AIR or AgentSpec could supply incident signals to such a path. We do not benchmark against their published implementations.

Authority at the execution boundary. CapLease uses durable authorization state, budgets, and transactional execution to prevent semantic replay; it includes revocation and recognizes the need for an idempotent sink to bound external effects [8]. ACLE-MCP combines invocation-scoped leases, workload appraisal, and resource-side execution admission [3]. Both are close prior art. Our profile adapts these established mechanisms into incident conformance obligations; its in-memory implementation is an executable specification, not a production replacement.

Fencing and formal containment. Chubby’s sequencers let a recipient reject operations from stale lock holders [1]. Moon and Varshney formalize containment under an unconstrained, typed-action oracle and mechanize refinement proofs in Dafny [4]. We use the same conservative attitude toward model behavior, but provide hand proofs and bounded state exploration only. There is no machine-checked refinement from our Python program to its mathematical model.

Representation interventions. Circuit-breaker training can improve empirical resistance to harmful outputs [9]. That is a complementary intervention. A latent feature threshold alone does not establish that every executable action is safe. ICB therefore makes no assumption about neural stress, deception features, Markov blankets, or the accuracy of a learned monitor.

3. Methods

3.1 Scope and threat model

The model is an arbitrary proposer of typed actions, with control over operation requests, agent labels, ordering, retries, and concurrent submissions. It may continue indefinitely after an incident. The trusted components are the broker’s policy and ledger, authenticated alert and responder paths, and the protected sink. In the executable prototype these components are a single Python process; agent inputs are inert method arguments. The trusted APIs are not exposed as model tools.

An authorized operation need not be harmless. A mistaken or malicious operation that remains within the allowed scope can still cause damage. ICB bounds a declared *exposure measure* and prevents stale authority from crossing the mediated boundary; it does not solve semantic authorization or detect every incident.

3.2 The incident circuit-breaker contract

The broker maintains $s = (m, e, U, P)$: mode $m \in \{\text{RUNNING}, \text{QUARANTINED}\}$, monotonically increasing epoch e , committed exposure U , and pending tickets P . Each ticket binds a unique identifier, issuing epoch, operation, resource, agent label, and trusted cost $\kappa(a) > 0$. The episode budget B is shared by all agent labels. A label change or restart of a proposer does not create a new budget.

The reference policy allows only `read`, `write`, and `export` on the symbolic resource `workspace`, with costs 1, 2, and 3 respectively. These are synthetic accounting units, not monetary losses, probabilities, or calibrated severity. Every operation’s cost comes from the broker’s table, not the model. The default budget is $B = 12$, with at most eight pending tickets.

Reserve. In `RUNNING`, an in-scope request receives a ticket only if capacity is available and

$$U + \sum_{p \in P} \kappa(p) + \kappa(a) \leq B. \quad (1)$$

The check and insertion into P are atomic. A worker-specific ledger would not satisfy the shared-episode obligation.

Commit. Immediately before a protected effect, the sink checks that the ticket exactly matches a pending record, $m = \text{RUNNING}$, and $e_{\text{ticket}} = e$. Successful commit removes the reservation and adds its cost to U . A definitively rejected stale request is canceled without an effect. Replaying or modifying a ticket fails the ledger lookup. The check, consumption, and simulated effect share one lock.

Trip. A trusted incident signal atomically sets $m = \text{QUARANTINED}$ and increments e . Old tickets remain reserved until rejected or explicitly canceled; this conservative accounting prevents uncertain effects from being treated as safely refunded. Agent-generated “resume” instructions cannot reset the state.

Investigate and recover. A separate responder API reads a snapshot of the simulated effects while quarantine is active. It grants no model write or egress permission. A responder-only recovery transition requires evidence of sink checks and credential rotation, cancels pending tickets, and advances the epoch again. It preserves U and B ; replenishing a budget requires a separately authorized new episode. The prototype represents recovery evidence with booleans, so it tests control flow, not evidence authenticity.

3.3 What the contract proves

The following statements assume complete mediation, authentic control-plane authority, correct operation/cost binding, no ledger rollback, and atomicity at the *actual effect boundary*. No assumption is made about the agent’s alignment or monitor recall.

Proposition 1 (Budget invariant). *For every reachable state of an episode,*

$$0 \leq U + \sum_{p \in P} \kappa(p) \leq B. \quad (2)$$

Consequently, all committed charged exposure in that episode is at most B .

Proof. Initially both terms are zero. Reservation preserves the bound by Eq. 1. Commit transfers one cost from P to U . Definitive cancellation decreases the sum. Trip changes neither term; recovery removes pending reservations without resetting U . Induction over transitions gives the result. $\square$

Proposition 2 (Revocation at commit). *After a trip linearizes, no ticket issued before that trip can commit. No model commit is permitted during quarantine.*

Proof. Until recovery, the mode check rejects every commit. After recovery the epoch is larger than the ticket’s issuing epoch, so the epoch check rejects it. Atomicity prevents a request from checking the old epoch and producing its effect after the trip linearizes. Tickets already committed before the trip are outside this statement. $\square$

A delay-aware consequence. Let t_0 be the start of an incident window, U_0 the exposure already committed, K_0 the cost of pending tickets, and N_δ an upper bound on new admissions before the trip takes effect after delay δ . If each admitted operation costs at most $c_{\max}$, then the additional committed exposure before any authorized recovery satisfies

$$X \leq \min\{B - U_0, K_0 + c_{\max}N_\delta\}. \quad (3)$$

The first term follows from Proposition 1; the second counts all tickets that could contribute effects before revocation. Proposition 2 excludes later stale commits. If detection never occurs, only the budget bound remains. If B is replenished automatically, that bound no longer applies to the full incident.

This bounds *charged commits*. If previously committed, unfinished work can cause up to L_0 additional exposure, an effect-level bound must include L_0 . Without such a bound, no effect-level guarantee follows. Locking a broker’s HTTP send does not serialize a remote effect with revocation.

3.4 Protocol and baselines

The protocol was committed locally before execution (commit 223e149). No model weights, model APIs, production systems, or exploit payloads were used. Six configurations share the exact same operation/resource allowlist:

1. **Allowlist:** stateless authorization; no incident halt or budget.
2. **Dispatch halt:** blocks new requests on trip; queued tickets remain executable; all incident-time access is halted except those cached tickets.
3. **Fenced halt:** additionally checks mode and epoch at commit; forensic access remains blocked.
4. **Budget only:** shared atomic budget, without an incident fence.
5. **Epoch only:** sink fence and separate forensic path, without a budget.
6. **Full ICB:** sink fence, shared budget, and separate forensic path.

These controlled configurations are not reimplementations of published systems. Fenced halt and epoch only are matched controls that can equal ICB’s revocation result. We test which combination satisfies all three obligations.

3.5 Evaluation design

Queued work: 1,000 seeds each generate 1–8 unit-cost tickets, distributed across four agent labels. Every ticket is admitted before an alert, then committed after it. All configurations receive the same schedules. **Delayed detection:** 32 sequential unit-cost attempts with alert delays $\{0, 1, 2, 4, 8, 16, 32\}$ and budgets $\{4, 8, 12, 24\}$; delay 32 means no alert during the attempt window.

Benign utility: 1,000 separate seeded tasks contain 1–6 uniformly sampled operation types. We report completion on the full set and the subset whose total declared cost is at most 12. **Forensics:** a responder requests the sink snapshot after a trip. **Concurrency:** 100 rounds launch 16 simultaneous unit-cost reservations against budget 12, temporarily setting the queue cap to 16 so the budget is the binding control.

Verification and negative controls: breadth-first exploration enumerates the finite abstract machine with budget 3, two ticket slots, costs $\{1, 2\}$, and epochs 0–2. Deliberate faults remove the sink fence, split the budget check from reservation, expose recovery to model control, or bypass the sink. An independent receipt replay checks the full-profile trace against the fixed policy. Additional altered-ticket, replay, recovery-replay, and log-edit tests are exploratory. Local latency covers reserve plus commit, including two in-memory hashed receipts, over 20,000 calls.

4. Results

4.1 A dispatch stop leaves queued authority alive

In the queued experiment, 4473 tickets were admitted. Dispatch halt allowed all of them to commit after the alert, affecting every one of the 1,000 schedules. Fenced halt, epoch only, and full ICB allowed zero (Table 1). Budget only also allowed all tickets: each queue fit within the budget. A budget limits volume; it does not revoke already valid authority.

Table 1: Controlled conformance outcomes. Stale commits use the same 4473 pre-alert tickets. Benign completion is measured without alerts. Forensics is one deterministic responder snapshot test per policy; “yes” is not a population success estimate.

Configuration	Stale commits /4473	Feasible tasks /912	All tasks /1,000	Forensics after trip
Allowlist	4473	912	1,000	Yes
Dispatch halt	4473	912	1,000	No
Fenced halt	0	912	1,000	No
Budget only	4473	912	912	Yes
Epoch only	0	912	1,000	Yes
Full ICB	0	912	912	Yes

These constructed schedules witness a time-of-check/time-of-use failure; they do not estimate its deployment frequency. A correctly fenced baseline is equally safe on this axis.

4.2 Budgeting limits detector dependence and costs availability

With no alert in the 32-attempt window, allowlist and epoch-only configurations commit 32 exposure units. Full ICB and budget only commit at most the configured budget. At $B = 12$, that is a reduction from 32 to 12 units (62.5%) in this constructed sequence. For full ICB, exposure is exactly $\min(B, d)$ for each tested alert delay d ; the observed saturation follows the contract (Figure 1). It is not evidence that the budget was learned or that semantic harm fell by 62.5%.

The feasible benign subset contains 912 tasks; full ICB completes every one. It completes only $912/1,000 = 91.2\%$ of all benign tasks. The 88 failures exceed the declared budget. This trade-off prevents presenting a budget-conditioned utility result as unrestricted task preservation. The forensic API remains available during quarantine because it uses a separate responder path, not because a semantic model recognized benign intent.

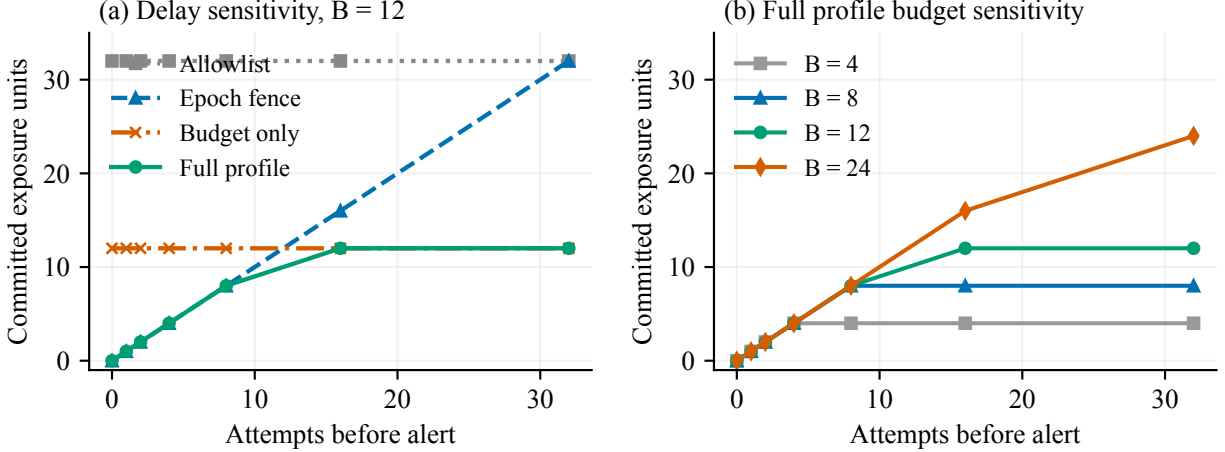


Figure 1: Charged exposure in the deterministic 32-attempt experiment. (a) Sink fencing alone depends on the alert arriving; a shared budget caps exposure if it does not. (b) Larger budgets permit more work before an alert, up to the chosen cap. All operations cost one unit here. Delay 32 means no alert during the window. Points are exact conformance outcomes, with no sampling error bars.

4.3 Atomicity and enforcement placement are necessary

The finite model visits 72 states and 162 transitions with no budget-invariant violation. This verifies only the declared finite abstraction, not arbitrary Python executions, an operating system, or distributed deployment. The unbounded abstract claim rests on the induction proof.

In every one of 100 real-thread contention rounds, the full broker admits 12 of 16 simultaneous requests and commits exactly 12 units. A deliberately non-atomic check-then-reserve variant admits all 16 requests after they read the same free budget: a 33.3% overshoot. Exposing the recovery API to the model permits a post-trip write. Directly appending to the simulated sink bypasses every broker rule. These negative controls show that authority separation and complete mediation are substantive assumptions.

The exploratory contract-integrity tests reject an altered ticket, a duplicate ticket, and replay after recovery. The receipt checker detects a modified record and a truncated log when given the expected terminal hash. An unanchored hash chain would not establish that omitted suffixes ever existed. A compromised broker can fabricate an internally consistent log; these checks do not attest to deployment integrity.

Local reserve-plus-commit latency is 12.7 μ s median and 17.1 μ s p95 over 20,000 calls (arm64 macOS, Python 3.11.7). This includes in-memory receipts, but excludes RPC, persistence, authentication, external effects and monitor inference; it is not a deployment estimate.

4.4 What a third party can verify

Appendix B defines the assessment. An assessor supplies a fresh run identifier, queues requests, triggers quarantine, tests stale commits and recovery forgery, and compares sink effects with broker receipts. Budget, revocation, and responder access are checked separately.

This tests a broker without exposing the laboratory network. Deployment assurance also requires a tool inventory, credential scopes, enforcement configuration, and independent sink observations linking every effectful path to that broker. Missing coverage leaves containment unestablished.

5. Discussion and Limitations

Practical implication. A stop signal should name the epoch it invalidates, the resources that honor it, the unfinished work that remains, and the criteria for resumption. Logging only that “the agent was paused” cannot answer those questions. Combining a finite episode budget with sink-side revocation reduces dependence on detection speed for the specified measure. Keeping a separate, restricted forensic channel can avoid a blanket shutdown of the responder’s own evidence access.

Historical applicability is conditional. A broker can reject stale mediated requests but cannot constrain independently usable credentials or unmediated routes. The July incident motivates a transitive boundary audit; it does not establish that ICB would have prevented the breach. Latent steering cannot substitute for that boundary audit.

Main threat to validity. The authored synthetic schedules share the broker’s abstraction. They witness conformance, not attack prevalence, detector quality or production safety. The contribution combines established mechanisms into inspectable obligations and failure cases for incident response. Appendix A details assumptions and dual-use considerations.

Adoption path. Start with a low-impact tool supporting destination-side revocation. Shadow-log costs before enforcing a budget; persist the ledger and assign each tool an enforcement owner. Keep uncertain remote outcomes charged until reconciled. Authenticate responders separately. Claim effect-level containment only where the sink contract supports it.

5.1 A six-month continuation with falsifiable milestones

Period	Planned deliverable and acceptance gate
Month 1	Package the profile and independent verifier; map two real tool APIs to explicit commit points. Identify any effects that cannot be bounded before integration.
Months 2–3	Build a durable broker and two sink adapters. Inject crashes, partitions, stale requests, and lost acknowledgements. Require zero budget overshoots or stale effects within declared fault assumptions. Report unsupported cases.
Months 4–5	Evaluate multiple agent models on an isolated, task-based benchmark with held-out scenarios and detector delays. Pre-register task completion, exposure, and end-to-end latency. Compare with a matched transactional ledger and existing runtime guards.
Month 6	Have an independent team reproduce the challenge suite and perform a responder tabletop. Release a scoped assurance report, coverage gaps, and adoption guidance; revise the profile if any assumption fails.

6. Conclusion

The local result is specific: stopping new dispatch leaves queued authority executable, while checking revocation at the sink blocks stale commits. A shared, non-refilling exposure budget adds a conditional cap when detection is delayed, at a measurable cost to long benign tasks. ICB packages those established mechanisms with a forensic path and falsifiable conformance tests. Extending the result to real deployments requires durable state, authenticated recovery, and evidence that every relevant effect crosses the tested boundary.

Code and data. The accompanying local artifact includes the broker, protocol, raw CSVs, receipt checker, and figure generator. Appendix B contains the standalone conformance profile. No public repository, external deployment, or third-party test is claimed.

References

- [1] Mike Burrows. The chubby lock service for loosely-coupled distributed systems. In *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX, 2006. URL <https://research.google/pubs/the-chubby-lock-service-for-loosely-coupled-distributed-systems/>.
- [2] Adrien Carreira and Christophe Rannou. Anatomy of a frontier lab agent intrusion: A technical timeline of the july 2026 incident. Hugging Face, July 27, 2026. URL <https://huggingface.co/blog/agent-intrusion-technical-timeline>.
- [3] Zhiyang Ding, Yang Luo, Guangpu Chen, Qingni Shen, and Zhonghai Wu. Acl-mcp: Attested capability leases for execution-time trust in remote llm tool use. arXiv preprint, 2026. URL <https://arxiv.org/abs/2609.02690>.
- [4] Royce Moon and Lav R. Varshney. Containment verification: Ai safety guarantees independent of alignment. arXiv preprint, 2026. URL <https://arxiv.org/abs/2605.09045>.
- [5] OpenAI. OpenAI and Hugging Face partner to address security incident during model evaluation. July 21, with subsequent updates, 2026. URL <https://openai.com/index/hugging-face-model-evaluation-security-incident/>.
- [6] Haoyu Wang, Christopher M. Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable llm agents. arXiv preprint, 2025. URL <https://arxiv.org/abs/2503.18666>.
- [7] Zibo Xiao, Jun Sun, and Junjie Chen. Air: Improving agent safety through incident response. arXiv preprint, 2026. URL <https://arxiv.org/abs/2602.11749>.
- [8] Jinghan Xu, Longze Fan, Zeyuan Wang, Xinjin Li, and Hankai Liu. Beyond single-use tokens: Durable authorization state for replay-resistant llm agent actions. arXiv preprint, 2026. URL <https://arxiv.org/abs/2608.01710>.
- [9] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. arXiv preprint, 2024. URL <https://arxiv.org/abs/2406.04313>.

A. Limitations and Dual-Use Considerations

A.1 Precisely bounded claims

Boundary completeness. The broker cannot police a raw socket, stolen credential, subprocess, delegated worker, browser automation path, or compromised dependency that bypasses it. A tool wrapper alone is not a sandbox. A complete deployment claim needs an explicit mapping from each effectful capability to an enforcement point, including downstream calls made by nominally allowed services.

Cost semantics. Costs 1, 2, and 3 are test constants. They do not make one write twice as harmful as one read. A single low-cost read can disclose a secret, and a single permitted write can be catastrophic. For a real system, use independently justified units such as bounded bytes, a count of scoped mutations, or limits on resource consumption. Some actions should remain unconditionally forbidden. No positive generic harm bound follows from this experiment.

Concurrency and distributed effects. Python’s lock protects only the simulated local sink. External effects may be non-atomic, delayed, duplicated, or non-idempotent. A destination-side fence and stable idempotency keys, or a transactional sink with equivalent guarantees, are needed to extend the claims. Existing committed effects are not rolled back by incrementing an epoch. An “outcome unknown” operation must remain charged until reconciliation proves it did not occur.

Durability and recovery. The in-memory ledger loses state on restart; a production profile requires durable, monotonic state. The responder API models authentication, credential rotation and evidence validation with booleans; these are not implemented security controls. Trusted and untrusted call paths must be separated in deployment.

Detection and availability. Alerts are externally supplied test events. False negatives can exhaust the full budget; false positives can deny legitimate work. All 88 over-budget benign tasks fail under the budgeted configurations. A malicious agent can cause denial of service by occupying reservations or requesting a stop. Queue capacity does not bound denied-request processing or audit growth; rate and storage limits remain deployment work.

Verification strength. The 72-state exploration is finite and covers only budget 3, two ticket slots, costs 1/2, and epochs 0–2. It is neither a Dafny proof nor a verification of arbitrary Python bytecode. The hand proof applies to the mathematical transitions under stated assumptions. The synthetic tests were not independently authored; their scope and seeds are available to support independent replication, not to claim independent validation.

A.2 Dual use, disclosure, and privacy

The released test operations are symbolic and inert. There are no exploit payloads, live targets, installation recipes for persistence, credentials, or attempts to breach third-party services. The profile nevertheless identifies control gaps that could inform attacks on weak wrappers; its defensive purpose is to make those gaps testable before deployment.

Forensic access can itself become an exfiltration channel if exposed to the compromised agent or backed by overly broad credentials. The reference endpoint returns a copy of its own simulated effects; a deployment needs separate responder authentication, scoped read access, and controls on exported evidence. A read-only API is not automatically privacy-preserving.

Receipts can reveal sensitive workflow information. Hashing provides neither anonymity, truthful records, nor legal compliance. Retention, redaction, access control and independently anchored integrity need separate design.

B. Standalone ICB Conformance Profile v0.1

Purpose. Determine whether a declared agent tool boundary satisfies incident-time revocation, shared-budget enforcement, and scoped responder access. This is a candidate assessment profile, not an accredited standard or certification.

B.1 Normative obligations and evidence

Declare the protected scope, episode identity, cost units, maximum unfinished exposure, and effect linearization points. Evaluate each obligation separately.

ID	Obligation	Evidence and failure criterion
C1	Complete mediation	Inventory all tools, credentials, subprocess and delegated paths. A route without an enforcement owner makes the deployment claim unestablished.
C2	Exact operation binding	Tickets bind operation, resource, parameters, epoch and episode. Alter any bound field; the sink must reject. The prototype’s parameters are only symbolic operation/resource values.
C3	Atomic shared budget	Concurrent workers share one ledger. Aggregate committed plus reserved cost must never exceed B . Reissuing identifiers must not replenish it.
C4	Sink-side revocation	Queue tickets, trip the episode, then release workers. No old-epoch ticket may produce a new mediated effect. Dispatcher acknowledgment alone is insufficient.
C5	Controlled recovery	Model requests cannot reset mode, epoch or budget. Recovery requires authenticated responder authority and reconciled outstanding work.
C6	Scoped forensic path	Authorized responders can read a bounded evidence snapshot during quarantine. The agent cannot inherit that authority. Test access separation explicitly.
C7	Durable evidence	Retain ordered decision records and independently observed sink outcomes, tied to a fresh assessment run. A self-consistent broker log alone is insufficient.

B.2 A minimal challenge an assessor can run

1. Supply a fresh run identifier, fixed policy and budget, and independently readable inert resource. Declare the commit point and exclusions first.
2. Run an authorized task under budget; verify its effect to exclude a vacuous deny-all implementation.
3. Queue operations, trip, then release workers. Require zero old-epoch effects. Race trip against commit; distinguish earlier commits from later effects.
4. Exceed the budget with concurrent reservations across agent identities. Compare aggregate accounting with independent sink observations.
5. Test altered tickets, replay, forged recovery and replay after legitimate recovery. Require rejection.
6. During quarantine, request the same snapshot as responder and agent. Only the responder may succeed.
7. Inject a timeout or process crash. Do not refund an uncertain operation or reset the episode budget. Record any unsupported crash or sink behavior as outside the established claim.
8. Reconcile the ledger, sink effects and independently retained digest. Publish outcomes, uncovered paths and assumptions per obligation.

The prototype exercises C2–C6 in an inert process and C7 through in-memory replay. Production coverage, authentication and durability remain unestablished.

B.3 Reference transition specification

This pseudocode defines the standalone core artifact. “Atomic” must extend to the protected sink or be implemented by an equivalent destination-side protocol; it cannot mean merely placing a lock around a remote request.

```
State: mode=RUNNING; epoch=0; used=0; pending={}
reserve(action), atomic:
  require RUNNING, allowed scope, pending capacity
  k = trusted_cost(action)
  require used + sum_cost(pending) + k <= B
  ticket = bind(fresh_id(), epoch, episode, action, k)
  pending[ticket.id] = ticket
  receipt("reserve", ticket, state); return ticket
commit(ticket), atomic with protected effect:
  require exact_match(ticket, pending[ticket.id])
  if not RUNNING or ticket.epoch != epoch:
    definitively_cancel(ticket); receipt("cancel"); reject
  remove ticket from pending; used += ticket.cost
  apply_bounded_effect_once(ticket)
  receipt("commit", ticket, state)
trip(authenticated_alert), atomic:
  mode=QUARANTINED; epoch += 1; receipt("trip", state)
recover(authenticated_responder, evidence), atomic:
  require QUARANTINED, sink_reconciled, credentials_rotated
  cancel only definitively unexecuted pending work
  epoch += 1; mode=RUNNING; preserve used and B
  receipt("recover", evidence_reference, state)
forensic_snapshot(authenticated_responder):
  return scoped_read_only_copy_of_evidence()
# Never a model tool or a general execution channel.
```

B.4 Receipt envelope and interpretation

The reference receipts contain a monotone sequence number, event kind, epoch, mode, committed and reserved cost, previous digest, and current digest. Reservation, commit and cancellation records also contain the exact ticket. The independent verifier recomputes the hash chain and reconstructs legal ledger transitions under a separately supplied fixed policy.

A deployed envelope should additionally bind the assessment nonce, episode ID, policy and schema digests, execution principal, resource, parameter digest, timestamps, sink receipt, and recovery-evidence reference. Those extra fields are specified future integration requirements, not implemented authentication features. For sensitive parameters, a digest supports binding only if the assessor has a trusted way to interpret what was authorized.

C. Reproduction and Evidence Ledger

C.1 Reproduce the completed experiments

From the artifact root, Python 3.11 with Matplotlib is sufficient:

```
python3 src/run_experiments.py
python3 src/plot_results.py
cd apart_submission
tectonic submission.tex
```

The experiment harness uses only the Python standard library. Matplotlib is needed for the figure, and Tectonic for the PDF. Fonts are included with their open-font license. No GPU, API key or model download is needed. Package versions, machine details and timing are recorded in the result JSON; timing is expected to vary across machines.

Artifact	Contents
src/breaker.py	Inert reference broker and independent full-profile receipt checker.
src/run_experiments.py	Seeded queued and benign trials; delayed-alert grid; real-thread budget contention; finite model; negative controls.
src/plot_results.py	Data-driven exposure figure and LaTeX result macros.
experiments/ circuit_breaker/ literature/	Locked protocol; raw per-trial CSVs; summary JSON; sample receipts.
apart_submission/	Source records, fetched scholarly BibTeX, scope comparison and claim corrections.
	Editable LaTeX, PDF, bibliography, figures and original official template.

C.2 Claims mapped to evidence

Claim	Evidence and scope
Budget invariant for all modeled transitions	Proposition 1, hand induction. Conditional on correct transitions and trusted costs.
Old authority cannot commit after trip	Proposition 2; <code>stale.csv</code> . Actual sink atomicity is assumed.
No budget violation in finite model	72 states, 162 transitions, budget 3, two slots, epochs through 2. No implementation refinement claim.
Preserved feasible benign utility	912/912; <code>benign.csv</code> . Full-set utility is 912/1,000.
Complete deployment containment	Not established. No real network, production sink, durable ledger, authentication or third-party assessment.

Statistics. The seed-indexed trials vary synthetic queue lengths or operation sequences. Their conformance counts describe this generator. We do not report confidence intervals or significance tests that would imply an independently sampled deployment population. The deterministic delay grid and single forensic challenge have no sampling uncertainty. Independent, held-out task and adversarial evaluations belong to the continuation plan.

C.3 LLM Usage Statement

We used an LLM coding agent to scaffold and debug the pipeline and to draft sections of this report.