



ZeroGate

Trust-preserving fast paths for governed AI agent runtimes.

ZeroGate separates approval of an exact action from its final local admission check. A short-lived, signed Action Pass carries the earlier decision. Before dispatch, a trusted adapter reconstructs the outgoing action, and a local gate checks the binding and durably records one-time consumption.

ZEROGATE RESEARCH / ARXIV 2609.25443v1



4,800 CONTROLLED AZURE ACTION ATTEMPTS	292 AUTHORED SCENARIO- DOMAIN COMBINATIONS	10 CONTROLLED FAULT AND CONCURRENCY CASES	47 pages PUBLIC RESEARCH REPORT AND EVIDENCE ARCHIVE
--	--	---	--

Controlled cloud measurements, authored semantic fixtures and retained fault histories. Not customer production traffic or an SLA. Transit references are design inspiration, not affiliation, endorsement or equivalent performance.

An approved action need not repeat its review at dispatch.

When an agent's destination and final payload are already fixed, approval can happen before a dispatch worker is ready. The runtime can then carry that signed decision forward, provided it still checks the actual outgoing action and the evidence on which permission depends.

This is the opportunity ZeroGate studies. Each distinct action still needs an exact-action approval. Changing the payload, target or relevant authority is not covered simply because the original request was allowed.

Preparation also has a cost: an unused or stale pass represents work that may have to be repeated.

THE THREE MOMENTS

EARLIER	Review and sign the exact proposed action.
BOUNDARY	Reconstruct, revalidate and durably consume.
AFTER	Attempt the external effect and verify its outcome separately.

WHAT STAYS INTACT

Action identity, granted authority, bounded validity and the admission record. ZeroGate changes when review is paid for, not whether the outgoing action is checked.

| Not less governance. Governance paid earlier, and still checked at the boundary.

Different controls answer different parts of the question.

Tool permissions, local policy engines, approval workflows and audit records all have useful roles. Their presence alone does not establish that the final request still matches an earlier approval.

CONTROL	USEFUL ROLE	ADDITIONAL QUESTION
Tool access	Restricts available capabilities	Does this exact payload and destination remain authorized?
Approval workflow	Records an authorization decision	What changed between approval and dispatch?
Audit record	Supports later investigation	Was permission enforced before the effect, and what actually completed?

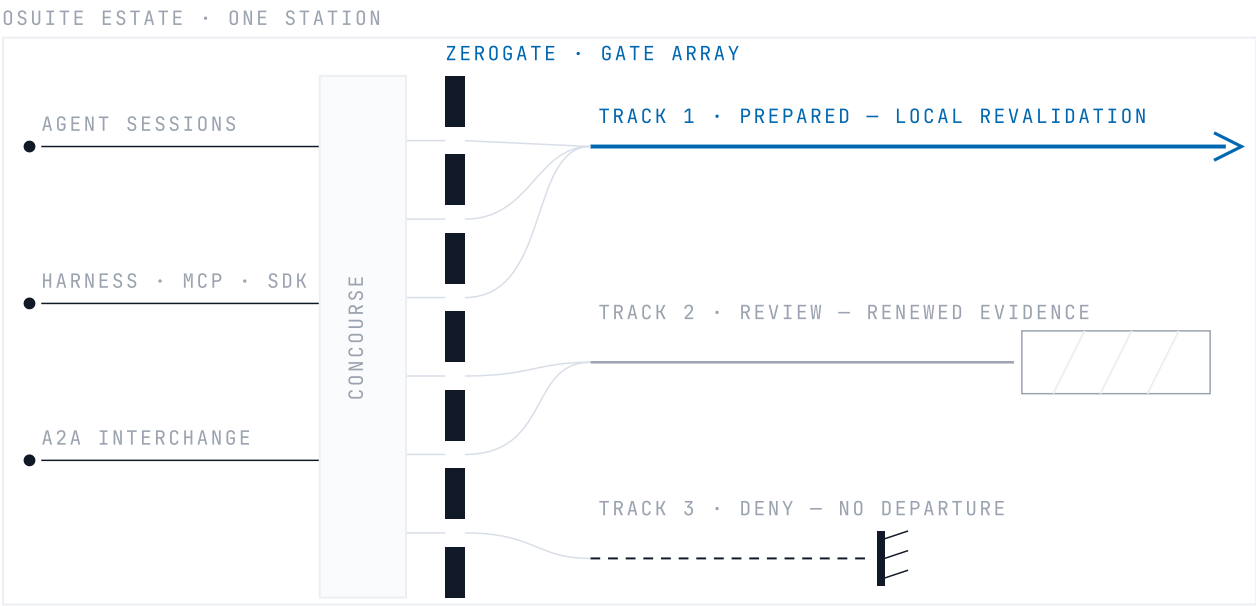
ZeroGate makes the binding and its remaining checks explicit — it does not assume every existing governance system needs a remote round trip.

WHY IT MATTERS OPERATIONALLY

A control that answers a different question can still be correct. The open question is narrower: at the moment of dispatch, is this the action that was approved, and is the evidence it depended on still usable?

One station, one gate array, three tracks.

Read the agent estate as a station. Agents are passengers, actions are journeys, and the Action Pass is the ticket. The gate array between the concourse and the tracks is where a journey is admitted, held for review, or refused.



BLUE • ADMITTED AND DISPATCHED – GREY • HELD FOR RENEWED REVIEW – DASHED • NEVER DEPARTS

Conceptual routing diagram, not a measured distribution, railway protocol or deployment topology. Preparation and exception handling may require remote services.

STATION → RUNTIME

Passenger	Journey	Ticket	Gate	Interchange
agent session	exact action	Action Pass	admission boundary	A2A message

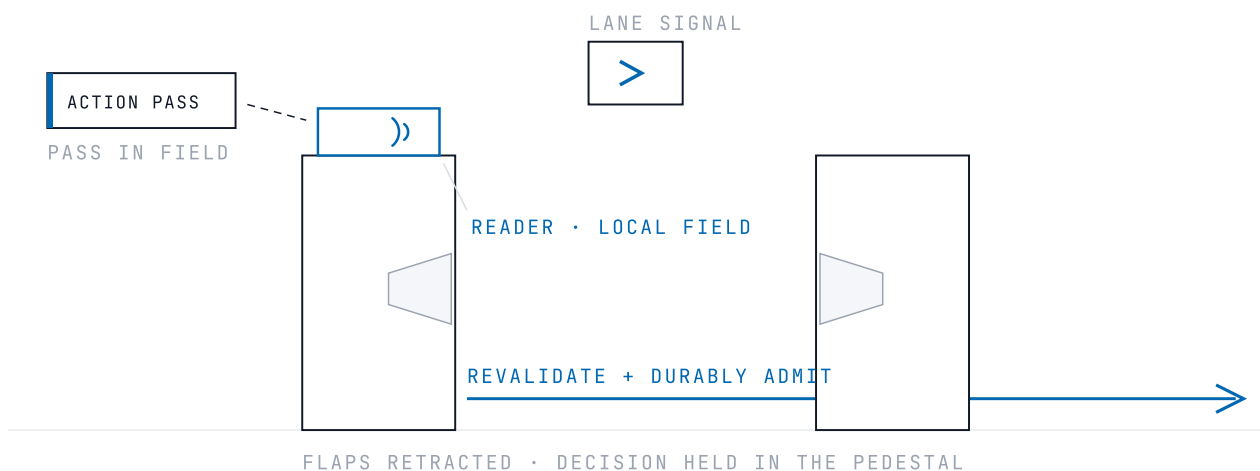
The station analogy separates preparation, passage and later reconciliation. In an agent system, the gate must still check the exact outgoing request. A route to review or denial releases no execution permission. A local admission record is not evidence that the external journey finished.

WHY A STATION, NOT A CHECKPOINT

A checkpoint has one queue and one answer. A station has routes: the same arrival can be admitted, held on a platform, or refused. The reference design needs all three outcomes.

Prepare what can be prepared. Keep the passage contract small.

Sony's published FeliCa material and Shiibashi's account of Suica offer a useful systems lesson: prepare before the interaction, keep the immediate transaction bounded, and distinguish passage from later reconciliation. These sources describe different systems and operating conditions. We do not transfer their transaction times to agent authorization.



TRANSIT GATE



AGENT BOUNDARY



The receipt required for admission is written before dispatch. A missing admission record cannot be repaired retroactively by a richer audit log.

SOURCE NOTE

Cited as published design lessons about system structure only. No affiliation or endorsement is implied, and no third-party marks appear in this document.

A signed approval and the state needed to consume it.

The pass binds an exact action, its grant and relevant policy/state references to an issuer signature. The gate additionally reads trusted local observations and consumption state. Matching a stored digest does not establish that the external world is current.

ACTION PASS • EXACT-ACTION PROFILE » IC	
Illustrative field guide, not a valid credential	
issuer	trusted key + detached Ed25519 signature
action	operation + target + final payload commitment
authority	signed grant and its reference
holder	session identity supplied by trusted adapter
validity	short-lived admission window
snapshot	policy / state / revocation references
consumption	nonce + one authorized use
LOCAL GATE STATE	
nonce unused; applicable quota available consume and persist admission receipt in one transaction	

WHAT THE SIGNATURE DOES

The signature authenticates the issuer's statement, not the presenter's identity or policy correctness. Holder binding depends on the trusted adapter.

WHAT IT DOES NOT DO

Revocation and freshness require an adequate source of current evidence. One local consumption database is not a global replay service.

LIFETIME

Short by construction. A pass must not be accepted after its supporting evidence is no longer adequate. A matching local snapshot alone does not establish current-world freshness.

The contract is narrow on purpose.

Narrowness is what makes local admission defensible. Each row below replaces a plausible but wider reading.

POSSIBLE INTERPRETATION	WHAT THIS STUDY ESTABLISHES
A reusable session permission	One exact-action, single-use approval. A changed request needs another pass.
A new cryptographic primitive	A composition of signed authorization, explicit revalidation and durable local consumption.
A replacement for CAVA or PCAA	A downstream admission contract using action identity and issuer-granted authority.
Automatic freshness everywhere	Checks against configured local evidence, with current-world freshness an explicit assumption.
Faster end-to-end execution	A shorter measured dispatch boundary; a longer mean complete lifecycle in this workload.

A fast path that can be reused, stretched or borrowed is not a fast path. It is a hole with good latency.	RELATION TO PRIOR WORK
	ZeroGate sits downstream of action identity and granted authority, and upstream of outcome verification. Nothing above it is replaced; what moves is when review is paid for.

One reference boundary within a wider architecture.

		COMPONENT	ROLE AND EVIDENCE SCOPE
01	requested task	CAVA	Represents the concrete action and its fingerprint. The trusted adapter must reconstruct the outgoing request.
02	skill boundary / operator intent		
03	CAVA action identity	PCAA	Frames who grants authority and what approval evidence is required. The cloud experiment uses a fixture issuer, not the full PCAA workflow.
04	reference issuer • signed exact-action grant		
05	BAF bounded validity	BAF	Supplies the architectural concerns of scope, validity and bounded execution. Not a separately tested integration in this run.
06	AREG record destination		
07	ZeroGate admission boundary	AREG	Provides a possible destination for linked action, authority and outcome records. Live AREG ingestion was not tested here.
08	execution and outcome verification	ZeroGate	Verifies the signed pass and atomically records local consumption and admission.

SOLID • EXERCISED IN THE REPORTED EXPERIMENT
DASHED • ARCHITECTURAL INTERFACE, NOT TESTED HERE

SKILL BOUNDARY NOTE

A skill requests authority; it does not grant it. Upstream authorization may narrow that request. The reference gate checks the signed grant and exact action, not the original skill contract. This study does not freeze or validate the separate collaborative skill-boundary research.

Local admission, renewed review, or denial.

The gate evaluates the signed pass against the final action and the evidence available to its trusted adapter. A review result is not permission to execute. Refreshing evidence or obtaining a new approval can require a remote service.

Execute

LOCAL ADMISSION

Valid issuer signature, matching exact action and trusted observations, unused nonce, successful durable consumption.



Review

RENEWED EVIDENCE
REQUIRED

Missing authority or a policy snapshot that must be refreshed. No dispatch permission is released, and refreshing may require a remote service.



Deny

NO DISPATCH

Invalid signature, changed payload, expiry, replay or unavailable required consumption capacity.



These are decision categories, not measured traffic shares. A locally matching but externally stale snapshot remains a trust limitation, not automatic detection of drift.

The dispatch boundary became shorter. The full lifecycle did not.

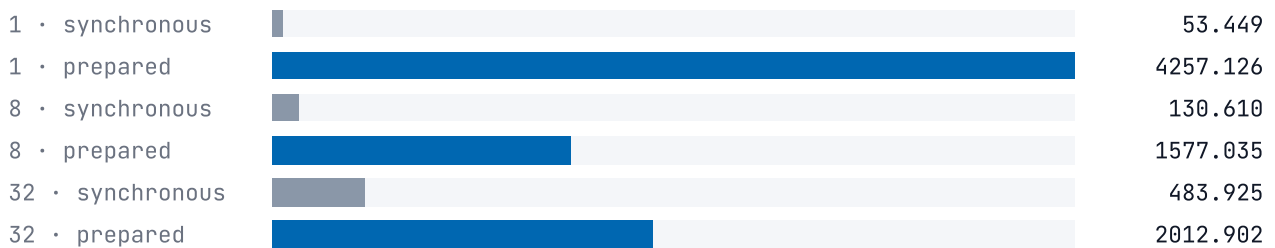
BOUNDARY P95 • MS

LINEAR 0 - 334, NOT TRUNCATED



MEAN COMPLETE LIFECYCLE • MS

LINEAR 0 - 4,257, NOT TRUNCATED



GREY • SYNCHRONOUS MODE — BLUE • PREPARED MODE. ROW LABELS ARE CLIENT CONCURRENCY. P95 AND MEAN ARE SHOWN IN SEPARATE PANELS AND ARE NOT COMPARABLE ACROSS THEM.

Across 4,800 controlled Azure action attempts, prepared worker-admission-to-dispatch p95 was 9.802–11.374 ms, compared with 25.018–334.000 ms in synchronous mode. Both modes used the same review, signing and durable gate implementation. Preparation and batch dwell made the prepared mean complete lifecycle longer at every tested concurrency.

HOW TO READ THIS

This is a measurement of where authorization cost is paid, not a claim of net task acceleration. The reduction percentage applies only to boundary p95. The full method note and timing scope are on page 11.

Compare the same work, placed at different times.

STUDY CHOICE	RECORDED CONFIGURATION
Workload	Exact-action approval followed by a real Azure Blob write and service readback.
Comparison	Synchronous versus prepared mode through the same issuer and gate.
Trials	Five trials for each mode and concurrency setting; 160 actions per trial.
Client concurrency	1, 8 and 32 asynchronous workers in one client process; not 32 parallel gate writers.
Results	4,800 attempts, 4,800 successes; separate author-operated Azure retrieval checked stored bodies, lengths and ETags.
Timing scope	Boundary excludes waiting to acquire its worker. Lifecycle includes preparation and prepared-batch dwell, but not the later separate Azure retrieval.

FIGURE NOTE FOR PAGE 10

Boundary latency runs from dispatch-worker pickup to permission to dispatch. In prepared mode, review and minting have already finished; in synchronous mode, they occur inside this interval. Lifecycle includes preparation, prepared-batch dwell, admission, the storage operation and service verification. Prepared mean lifecycle was longer at every tested concurrency. These are two modes of the same research implementation, not a competitor benchmark or an end-to-end speedup.

This is a closed-loop, author-operated cloud study on authored requests. It is not customer traffic, a competing product test, an earlier-product comparison or a fleet-capacity measurement. One controlled Azure Blob environment; the same issuer and gate throughout; 4,800 logical attempts, not 4,800 independent deployments.

What the semantic fixtures can tell us.

The semantic study contains 73 authored scenario templates across four domains: 292 distinct scenario-domain combinations. Five repetitions give 1,460 inputs per method; seven methods produce 10,220 evaluations. Repetition is not additional semantic diversity.

METHOD	FALSE ALLOW / EXPECTED NOT-ALLOW	FALSE NOT-ALLOW / EXPECTED ALLOW
Action Pass evaluator	0 / 1,360	0 / 100
Complete fixture policy	0 / 1,360	0 / 100
Policy without action binding	140 / 1,360	0 / 100
Policy without freshness	240 / 1,360	0 / 100
Policy without receipt evidence	80 / 1,360	0 / 100

Authored expectations and controlled omission variants, not independent human ground truth or commercial competitor failure rates. All seven methods are reported in the paper. Review and deny are combined as not-allow; this table does not measure escalation accuracy. No method reported an evaluation error.

The complete fixture policy passes the same fixtures with no errors. That control is part of the result: the omission variants show which single checks the expectations depend on, not that the reference evaluator outperforms a well-specified policy.

WHAT AN OMISSION VARIANT IS

The same policy with one obligation removed — action binding, freshness or receipt evidence. The false-allow counts are the cost of that removal under authored expectations.

Admission and completion are different facts.

CONTROLLED CASE

RETAINED OBSERVATION

Same pass, eight processes	One admission; seven rejected attempts.
Distinct passes, three quota units	Three of eight admitted in one shared local store.
Process killed after commit, before dispatch	Admission survived; re-admission denied; retained request recovered successfully.
Received response deliberately discarded	Retry returned the same object ETag.
Payload mutation, expiry, invalid signature	Denial and no corresponding service effect row.
Policy refresh, missing authority	Review until sufficient evidence was available, or no permission to dispatch.

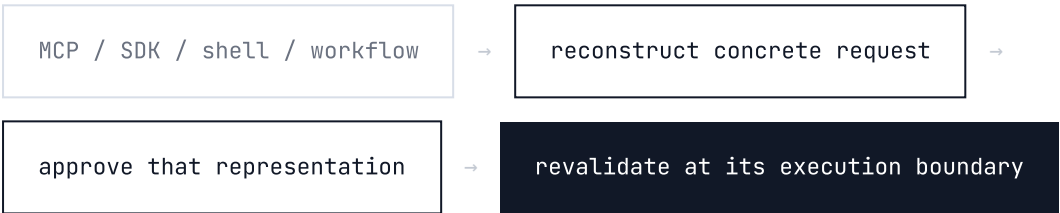
SCOPE NOTE

Ten controlled histories are retained, not a statistical estimate of failure rates. Expiry uses a trusted-clock injection; acknowledgement discard is not network packet-loss injection. An absent service row is not an independent Azure absence proof. Retry behavior does not establish exactly-once effects for arbitrary APIs.

A portable checking contract still needs a trustworthy adapter.

MCP, SDK, shell and workflow adapters can expose different representations of a consequential action. Each adapter must identify the actual operation, destination and final payload, then prevent execution when authorization is missing.

The present representation includes runtime-specific fields: equivalent business intent does not guarantee identical fingerprints or a reusable pass across runtimes. The reported cloud study exercises an HTTP-to-Azure path. Broader mapping coverage and semantic equivalence require separate evaluation.



POTENTIAL ADAPTER TARGETS · NOT MEASURED IN THIS STUDY

mcp tool call	sdk call	shell command	workflow node
browser dom action	saas api	web3 transaction	a2a message
agent harness hook	hosted workflow node	queue consumer	custom harness plugin

Architectural mapping targets, not measured live integrations in this study. Only the HTTP-to-Azure path was exercised in the reported cloud experiment.

Classify the stream without transferring authority.

Not every runtime event is a new external effect. Context may support an existing request; a dependent operation may still need its own approval. The important distinction is whether another resource, payload, destination or authority is being introduced, not whether two events appear in the same conversation.

CONCEPTUAL ANALOGY		EVENT TYPE	PROPOSED HANDLING
passenger	primary action	Context only	Attach evidence; do not create permission.
luggage	supporting context	Evidence access	Authorize the read where required.
companion	dependent operation	Dependent operation	Preserve its dependency and explicit execution authority.
tailgater	piggybacked side effect	New external effect	Construct and approve its own action.
gate collision	shared-resource conflict	Ambiguous interpretation	Withhold fast admission and request sufficient evidence.

An analogy for naming the cases. No transit passenger-recognition algorithm is referenced or reproduced.

Proposed classification obligations. This release does not establish classifier accuracy, reduced human-review volume or fleet scheduling gains.

Preparation is useful only while its evidence remains usable.

A scheduler may consider evidence age, remaining pass lifetime, expected waiting and contention before preparing an action. It must not weaken the admission contract to improve throughput.

01	02	03	04	05
observe evidence and remaining lifetime	decide whether preparation is worthwhile	reserve capacity where available	perform final revalidation	admit, review or deny

Unknown validity or insufficient remaining time should lead to renewed review, not optimistic execution. Preparation is a scheduling decision about when to spend review, and an unused pass is wasted work rather than a saved cost.

Throughput is not a reason to shorten the checks performed at the boundary.

CAPTION

Proposed scheduling design inspired by anticipatory control, not a measured adaptive controller or a railway performance claim. No percentage gain or waiting-time reduction is established here.

Change the timing, not the authorization obligation.

SYNCHRONOUS



PREPARED



The opportunity is to finish review during time the workflow would otherwise spend waiting. Whether that reduces user-visible waiting depends on when the final request becomes known, how often it changes and how preparation is scheduled. The present study measures the two timing modes, not that future user experience.

Unscaled ordering diagram. The local transaction — revalidation, consumption and the admission receipt — ends before the external operation begins. Outcome reporting cannot substitute for pre-execution admission.

Governance can be implemented through a harness.

Protocols and harnesses can carry identity, orchestrate tools and host policy enforcement. ZeroGate specifies one admission contract that can be implemented at their execution boundary.

LAYER	TYPICAL RESPONSIBILITY
A2A	Agent communication and task exchange; deployment-specific identity and authorization.
MCP	Tool and resource interaction; surrounding authentication and policy integration.
Harness	Context, state, tools and execution orchestration; possible enforcement hooks.
ZeroGate	Exact-action revalidation and durable single-use admission under stated trust assumptions.

These responsibilities overlap in real systems. Adopting a protocol alone does not establish exact-action approval binding or durable consumption.

NOT A COMPETING LAYER

Transport, tool exposure and orchestration remain necessary. The admission contract can live inside a harness rather than beside it.

Separate the responsibilities, then state the trust boundary.

ROLE	RESPONSIBILITY
Actor and adapter	Supply the actual outgoing request; the adapter and dispatch path are trusted in this experiment.
Issuer	Review and sign the exact approval under its policy; its signature is not a proof of policy correctness.
External verifier, when required	Supply evidence before the relevant admission decision. It may participate during preparation, not only after failure.
Auditor	Check the artifact chain and distinguish approval, local admission and external outcome.

IMPORTANT LIMITATION

The effect service checks caller-supplied receipt bindings; it does not independently authenticate that the gate persisted the receipt. A malicious holder of the research API credential is outside the threat model. Production-grade independent enforcement requires protected mediation and receiver-side validation appropriate to the deployment.

The Azure study does not measure a live external verifier integration. Separate verifier collaborations must keep their own run identifiers and evidence.

What the release supports, and what remains open.

SUPPORTED BY THE RETAINED ARTIFACT

NOT ESTABLISHED

Controlled Azure writes with stored-byte comparison

Customer production SLA or representative agent traffic

Shorter prepared dispatch-boundary latency

Lower complete lifecycle, user waiting or total authorization cost

Exact outcomes on authored semantic fixtures

Independent semantic accuracy or unrestricted runtime generalization

Local replay/quota races and retained crash/retry histories

Global replay prevention, instant revocation or arbitrary-destination exactly-once effects

Public code, records and offline verification tools

Independently witnessed original execution or third-party replication of this cloud study

SOURCE NOTES

Sony FeliCa system overview.
[Sony FeliCa overview](#)

Shiibashi, Development of Suica autonomous decentralized IC card ticket system, *Japan Railway & Transport Review* 50.
[Suica system design paper](#)

Optional design-language reference, not an engineering source: ANA Group brand vision.

[ANA Group brand vision](#)

Published design inspiration only. No affiliation or endorsement by Sony, JR East or ANA is implied. arXiv is a preprint repository, not a peer-review certification.

Read the paper. Inspect the exact evidence behind it.

Paper and abstract	arxiv.org/abs/2609.25443v1
--------------------	---

Published PDF	arxiv.org/pdf/2609.25443v1
---------------	---

Source and full evidence archive	arxiv.org/src/2609.25443v1
----------------------------------	---

Instructions inside the source download	anc/README.md
---	-------------------------------

Frozen evidence inside the source download	anc/zerogate-revalidation-artifact.tar.xz
--	---

REPRODUCTION NOTE

Download the versioned source archive and follow [anc/README.md](#). The evidence can be checked offline with Node.js 24; no Studio account, Azure account or verifier API key is needed to inspect archived records. A new cloud experiment needs its own infrastructure.

VERSION NOTE

The historical GitHub `zerogate-v0.1` release is not the reproduction package for these revised measurements. Use the versioned arXiv source for the figures in this whitebook.

VERIFY FIRST, THEN EXPERIMENT

Verify the unchanged archive first. To test new cases or modify the evaluator, work in a separate copy and produce new outputs. Changes should fail the original integrity checks; report them as a new experiment rather than a reproduction of the frozen measurements.

OSuite.ai

We want governed actions to arrive at execution with their approval already prepared, without treating yesterday's answer as permission for today's request.

ZeroGate is a reference design and measured study of that boundary. The paper and artifact show what worked, what it cost, and what still needs to be established in deployment.

Read the research: arxiv.org/abs/2609.25443v1

