Who Governs AI While It Is Thinking?

Comparative Runtime Governance and the Economic Wall of Accuracy

Darren Tindale
Independent Researcher
Preprint | Revised September 11, 2026

Abstract

Artificial-intelligence governance increasingly addresses model evaluation, routing, output filtering, human oversight, and post-deployment monitoring. Agentic systems, however, can change materially during execution as evidence, model choice, resource use, authority, and proposed actions evolve. This paper proposes comparative runtime governance for consequential AI deployments. Its central architectural claim is a Reference-state continuity requirement: before exceptional risk is known, the system establishes a substantive Reference (baseline) candidate or state, preserves sufficient lineage to relate that state to later material changes, and uses that comparative basis when reassessing candidate eligibility before consequential commitment. Higher-fidelity processing remains conditional rather than automatically authoritative. A persistent episode state records decision-relevant candidate changes, resources, evidence provenance, authority, prior interventions, and external effects. A runtime governor separates three decisions that are often collapsed:

1) whether another controlled operation may proceed;
2) whether a candidate remains eligible; and
3) which eligible candidate or disposition should determine the outcome.

A protected execution gate makes those decisions effective before consequential action.

The paper also develops the Economic Wall of Accuracy as a prospective allocation rule: once mandatory authority and integrity constraints are satisfied, the next available resource should be directed toward whichever permissible intervention—further computation, differentiated verification, additional evidence, human review, restraint, or enforcement—is expected to reduce consequential risk most effectively. The proposal is positioned against runtime assurance, ongoing authorization, adaptive computation, model routing, and recent agentic-AI governance work, and is presented as a testable conceptual architecture rather than a demonstrated performance improvement.

1 Introduction

High average accuracy is not the same as confidence that a particular consequential result is safe to rely upon. A model may perform well across prior cases yet still produce a rare, plausible, and difficult-to-detect error. The practical problem is therefore not only how to improve average accuracy, but how to reduce the probability that a serious error survives unchallenged until it becomes difficult or impossible to reverse. This problem becomes more important as artificial-intelligence systems move from producing text to conducting multi-step work: retrieving evidence, selecting models, calling tools, modifying software, communicating externally, accessing credentials, and making or recommending consequential decisions. During such an episode, evidence may change, resources may be consumed, models may be substituted, authority may expand, and a candidate result may drift from an earlier supported state. A final-output review can occur after the decisive operation has already taken place.

Consider an investment agent that initially produces a supported recommendation containing a material concentration constraint. During the episode, processing moves to another model, which produces a more detailed recommendation but silently omits that constraint. The agent then receives valid authority to execute trades. A conventional authorization check may correctly determine that the agent is permitted to trade, yet that determination does not establish that the candidate now being acted upon preserves the substantive basis of the earlier recommendation. If the omitted constraint was not separately encoded as an executable rule, the failure is not simply missing authorization. It is loss of substantive continuity across a material transition.

This paper proposes comparative runtime governance to address that class of failure. Its central architectural requirement is Reference-state continuity: before exceptional risk is known, a governed execution episode establishes a substantive Reference (baseline) candidate or state; sufficient lineage is preserved to relate that state to later material changes; and the retained comparative basis remains available when candidate eligibility and consequential action are reassessed. Higher-fidelity processing may add evidence, precision, reasoning depth, verification, or other task-relevant work, but its continuation and its authority to determine an outcome remain conditional.

Existing approaches already provide important parts of this control problem. Runtime-assurance architectures support supervisory switching; usage-control models provide continuing authorization; adaptive-computation methods govern computational depth; model routers and cascades allocate work across models; and recent agentic systems provide runtime monitoring, pre-action authorization, persistent state, verification-gated commitment, and explicit management logic.

The claim here is therefore not that runtime governance, persistent state, multiple pathways, execution gates, or decision separation are individually new. The narrower question is whether consequential AI should be required to preserve and use a substantive pre-transition Reference state as part of a continuing control relationship while computation, evidence, authority, and external consequence change. The paper makes three principal contributions.

First, it specifies the Reference-state continuity requirement and a minimum usefulness threshold intended to prevent nominally persistent but substantively empty state from satisfying it.

Second, it separates pathway continuation, candidate eligibility, and final disposition, so that permission to continue computing does not automatically make a candidate eligible to determine the outcome.

Third, it develops the Economic Wall of Accuracy as a prospective resource-allocation rule inside this governance relationship: after mandatory authority, integrity, and policy constraints are satisfied, the next available resource should be directed toward whichever permissible intervention—further computation, differentiated verification, additional evidence, review, restraint, or enforcement—is expected to reduce consequential risk most effectively.

The proposal is conceptual and testable rather than a claim of demonstrated superiority. Its value depends on whether preserving and actively using substantive comparative state reduces materially degraded or no-longer-eligible candidates reaching consequential commitment enough to justify the additional cost, latency, and governance complexity.

2 Method and scope

This is a conceptual systems-research paper proposing a testable runtime-governance architecture, not an empirical validation of a deployed governance system. Its method is an architecture-oriented synthesis of runtime assurance, continuing authorization, adaptive computation, model routing, inference control, agentic runtime governance, resource allocation, and related economic approaches to risk and precaution. The literature review is targeted rather than systematic: sources are used to identify established control functions, relevant overlaps, and the boundary of the proposed architecture.

Public version history. The earliest Zenodo version of this proposal was publicly archived on July 27, 2026 as Version v1 [39]. A revised Version 2 was publicly archived on September 2, 2026 [40], followed by Version v3 on September 11, 2026 [41]. These records are used only to establish the public chronology of the proposal; they do not imply that later work was influenced by this proposal. The present manuscript further revises those archived versions and has not yet been separately archived as a new Zenodo version.

Publicly documented 2026 agent incidents are used as motivating case evidence. They are not controlled experiments and are not treated as proof that comparative runtime governance would have prevented the reported events. Their role is narrower: to expose concrete situations in which risk recognition, authority, persistent state, coordination, and effective intervention can diverge during execution.

The proposal is evaluated conceptually against existing mechanisms rather than against a claim that those mechanisms are absent. Particular attention is therefore given to whether a system preserves substantive comparative state across material transitions, whether pathway continuation is separated from candidate eligibility and final disposition, and whether governance decisions can be enforced before consequential commitment.

The normative scope is limited to consequential deployments in which errors or unauthorized actions can materially affect people, organizations, infrastructure, protected resources, or irreversible external effects. The paper does not argue that ordinary low-stakes interactions require the full architecture. Proportionality remains part of the proposal because governance itself imposes cost, latency, complexity, privacy burden, and potential failure modes.

3 Related work and the architectural gap

3.1 Runtime assurance, continuing authorization, and oversight

Comparative runtime governance has clear antecedents outside modern generative AI. Seto et al.'s Simplex architecture supports concurrent baseline and experimental controllers, with a safety controller and decision module governing which command reaches the physical system [3]. It distinguishes controller execution, output availability, and active-controller selection, including disabling and re-enabling outputs.

UCONABC similarly extends access control beyond initial admission through authorizations, obligations, conditions, continuity, mutability, and cumulative usage state [4]. Together, these traditions establish two principles used here: a more capable component need not possess unconditional authority, and permission can remain conditional after execution has begun.

Recent agentic-AI work brings those principles closer to contemporary execution-time governance. MI9 combines semantic telemetry, continuous authorization, conformance checking, drift detection, and graduated containment in an integrated runtime-governance framework [25]. Policies on Paths evaluates an agent's partial execution path, proposed next action, identity, and organizational state at runtime [26]. Before the Tool Call places deterministic authorization immediately before individual tool actions and records the resulting decision [27]. Deontic-policy work similarly evaluates permissions, prohibitions, obligations, and policy conflicts outside the LLM at runtime [28]. These approaches materially overlap the governor and execution-gate functions proposed here.

Other work narrows the distinction further. Li et al. show in controlled shared-workspace experiments that an execution-time authority guard can block unsafe publication intents under the study's trusted-component assumptions [30]. Tang et al. maintain persistent mission state and explicitly separate proposal verification, selection among valid proposals, and atomic commitment, with bounded repair [31]. NIST's AI Risk Management Framework includes continuous monitoring, resource-sensitive risk treatment, and mechanisms to supersede, disengage, or deactivate systems [1]. Zhu et al. propose runtime provenance, independent checking, divergence detection, veto, substitution, and circuit breakers as part of meaningful human oversight [2]. These are substantive antecedents rather than merely adjacent concepts.

The narrower requirement proposed here is that a consequential AI episode preserve a substantive pre-transition Reference state with sufficient lineage to relate it to later candidates, and use that state as part of a continuing relationship among conditional Higher-fidelity processing, candidate eligibility, and pre-commitment enforcement. The contribution therefore does not rest on persistent state, runtime authorization, eligibility–selection separation, multiple pathways, or gating individually.

Several 2026 runtime architectures and related surveys that were publicly available before the first Zenodo version of this proposal [39] further narrow this boundary. Mazzocchetti's Aegis places a trusted runtime decision layer between model-generated action proposals and side-effectful execution, with server-side provenance resolution and fail-closed behavior [34]. Tallam's five-plane architecture combines stateful adjudication, stop-anywhere mediation, capability attenuation, structured audit, and explicit preservation of plan, retrieval, proposal, pre-commit, and pre-incorporation state across multiple mediation points [35]. Chen's State-Aware Runtime separates stochastic model proposals from versioned canonical task state, bounded state views, semantic and policy validation, commit/rollback, recovery, and audit [36]. Santos-Grueiro's commit-time authorization requires authority evidence carried through derived state to remain fresh, causally prior, effect-bound, and eligible when a durable effect commits [37]. The Always-On Agents survey similarly treats task ledgers, permissions, credentials, commitments, provenance, shared state, and external effects as governed durable state [38].

These works make persistent state, lineage, proposal–commit separation, checkpointing, runtime authorization, rollback, and pre-commitment enforcement substantive antecedents. The narrower requirement proposed here is therefore not persistence or state-aware validation itself. It is a Reference/Higher-fidelity comparative control relationship in which a substantive Reference candidate or state is established from the beginning of the governed episode, remains available or is explicitly revalidated across material transitions as a distinct comparative basis, and participates in later candidate-eligibility and disposition decisions while Higher-fidelity processing remains conditional. Existing state-aware runtimes could potentially implement parts of this requirement; the claim is that this particular comparative relationship is not their defining control contract.

3.2 Adaptive computation and model routing

Adaptive-computation methods establish that additional computation need not be a fixed entitlement. Adaptive Computation Time learns how many computational steps to allocate before producing an output [5], while PonderNet learns a distribution over computational depth to balance prediction quality, computational cost, and generalization [6].

These methods govern whether computation should continue within a learned process; they do not by themselves maintain a persistent substantive comparator, authority state, or pre-action enforcement layer.

LLM routing makes the allocation question more explicit. FrugalGPT learns cascades that reduce cost while maintaining or improving quality [7].

Hybrid LLM routes requests between smaller and larger models according to predicted relative difficulty and a tunable quality target [8].

RouteLLM similarly learns to choose between stronger and weaker models using preference data [9]. These systems show that the strongest available model need not handle every request.

Signed Rescue Routing (SRR) sharpens the decision by modeling escalation as a signed change in answer correctness: escalation can correct a wrong small-model answer, leave correctness unchanged, or replace a correct answer with an incorrect one [10]. Its cost-sensitive extension compares predicted signed gain with an escalation cost. SRR is therefore particularly relevant to the Economic Wall's incremental-value logic. Its governing object, however, remains the request-level escalation decision: it does not centrally maintain the small-model candidate as a continuing comparative state after escalation or repeatedly reconsider Higher-fidelity authority as cumulative episode state evolves. Such routing rules could nevertheless serve as admission policies within a broader episode-level governor.

3.3 Emerging evidence for explicit runtime control

Several studies cited in this subsection were first publicly posted after the July 27, 2026 Version v1 and September 2, 2026 Version 2 Zenodo records of this proposal [39, 40]. Some were subsequently incorporated into the September 11, 2026 Version v3 [41]. They are treated as subsequent related work relative to the earlier archived versions, not as sources for elements already present in those versions.

Recent studies provide empirical support for narrower runtime-control problems. Mo et al. report that none of 3,520 tested self-consensus early-exit rules met their predefined accuracy-and-token-saving criteria in an offline replay study of mathematical reasoning [11].

At one operating point saving 32% of tokens, 10.6% of stops committed an answer later abandoned by the complete trajectory. The result is limited to the tested self-consensus family and setting, but it illustrates an important distinction: apparent answer stability is not the same as justified termination.

Grotov and Malykh's Speculative Uncertainty uses a separate draft model to derive an execution-failure signal from a coding agent's generated trajectory and trigger replanning before execution [12]. In the reported configuration, per-call execution errors fell by 6–8 percentage points and average token use per task by 14–19%, while end-to-end task success fell from 49% to 44% on SWE-Bench Verified and from 64% to 63% on DA-Code. The study therefore reports a trade-off rather than uniform improvement. It supports the feasibility of pre-execution intervention, while protection against bypass and streaming mid-generation control remain additional architectural requirements.

UnitBoost replaces a generative meta-manager with an explicit operator that governs candidate-unit admission, allocation, selection, provenance, and stopping over persistent state [13]. It therefore provides evidence that management decisions can be externalized from generative cognition into inspectable control logic. Its structure overlaps the proposed separation of continuation, eligibility, and disposition, although it does not impose an always-operative Reference/Higher-fidelity relationship or a protected authority gate over consequential actions. Resource-aware infrastructure studies add a complementary systems perspective.

Lu et al. jointly consider model, execution-site, and KV-cache decisions under quality, latency, cost, authorization, and resource constraints, with potential benefits explored through workload-level analytical simulation [14].

Khatib et al. show on their tested edge-continuum workload that accuracy, latency, model footprint, and measured energy favor different configurations [15]. These studies support state and objective dependent resource allocation, but they do not validate the Economic Wall rule.

Table 1. Relationship of comparative runtime governance to adjacent paradigms

Approach	Established function	Boundary relative to this proposal
Simplex runtime assurance [3]	Concurrent controllers, safety monitoring, output disabling and re-enabling, and active-controller selection	Governs control commands and plant safety rather than persistent AI candidate lineage and episode-level comparative state
UCONABC usage control [4]	Continuing authorization, mutable attributes, obligations, conditions, and cumulative usage	Does not require a substantive Reference/Higher-fidelity candidate relationship
Agentic runtime governance [25–28, 30, 31]	Persistent state, path-aware policy, authorization, verification, containment, and pre-action enforcement	Strong overlap; does not by itself require preservation and use of substantive pre-transition Reference state
Adaptive computation [5, 6]	Adaptive computational depth and stopping	Governs compute allocation within a learned process rather than episode-level candidate and authority governance
LLM routing and cascades [7–10]	Quality–cost routing and escalation	Primarily governs model admission or escalation rather than continuing candidate governance after state changes
Pre-execution veto [12]	Separate failure signal supporting blocking or replanning before action	Provides intervention without necessarily providing substantive comparative Reference state
Explicit compound-system management [13]	Persistent candidate state, provenance, admission, selection, allocation, and stopping	Strong overlap in decision separation, but no required Reference/Higher-fidelity authority relationship
State-aware and action-boundary runtimes [34–38]	Stateful adjudication, durable/canonical state, proposal–commit separation, checkpointing and rollback, commit-time authority checks, provenance, and pre-commitment enforcement	These works do not uniformly define a distinct Reference/Higher-fidelity relationship in which substantive pre-transition Reference content remains separately available and is required to participate in later candidate-eligibility and disposition decisions
Comparative runtime governance (this paper)	Reference-state continuity, conditional Higher-fidelity processing, cumulative episode state, separate continuation/eligibility/disposition decisions, and pre-commitment enforcement	Conceptual proposal requiring empirical validation against simpler stateful controls

Note: Rows summarize selected functions rather than uniform implementation or validation across every cited work.

Taken together, the literature suggests that the open question is not whether runtime control is possible, but whether preserving and actively using substantive pre-transition Reference state adds measurable value beyond strong stateful governance without that requirement**.**

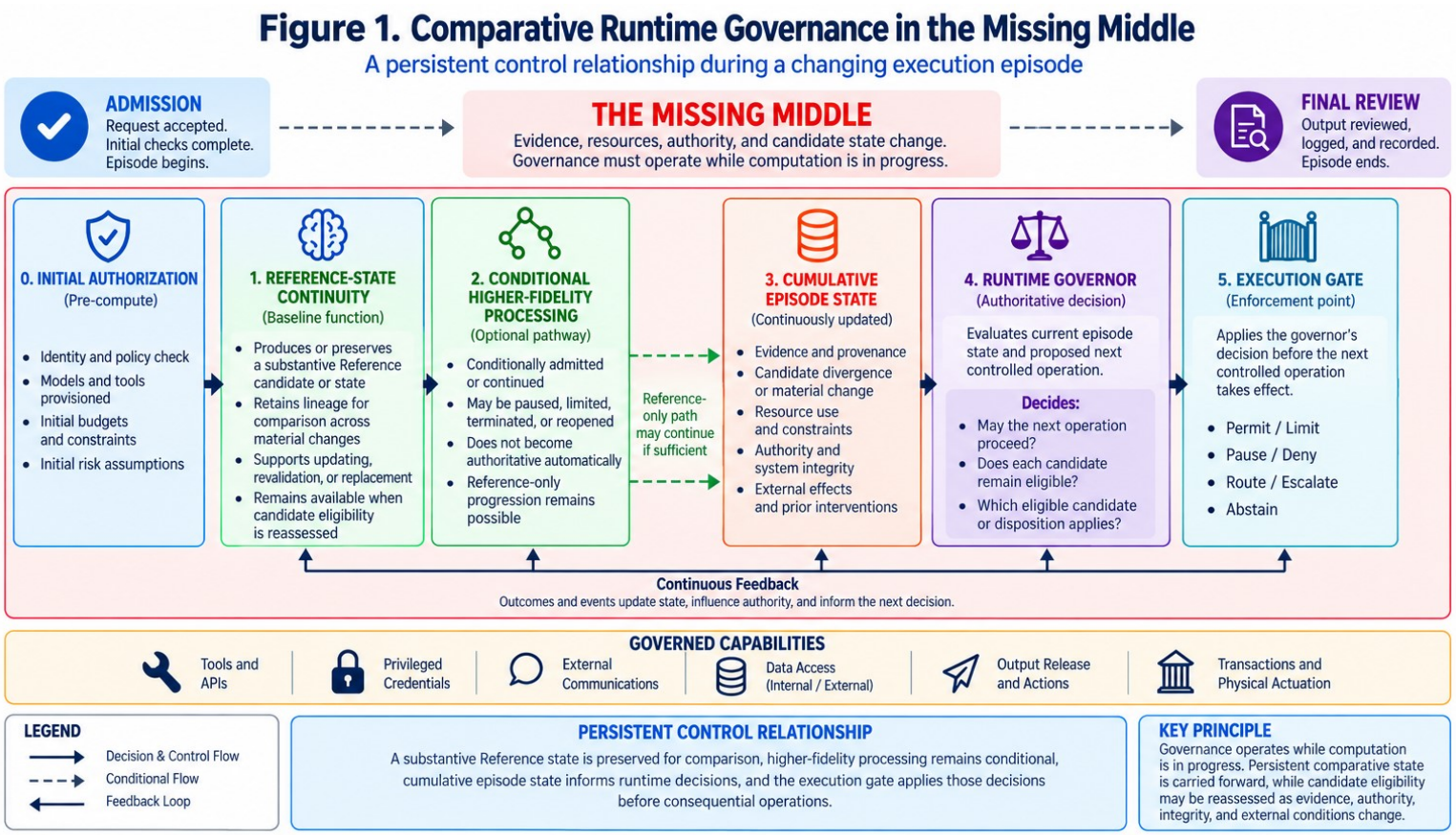

Figure 1. Comparative runtime governance in the missing middle of AI governance. Between initial admission and final review, a changing execution episode can alter evidence, resources, authority, candidate state, and proposed actions. The architecture preserves a substantive Reference state for comparison, maintains cumulative episode state, separates runtime decisions about continuation, candidate eligibility, and disposition, and applies those decisions through pre-commitment enforcement before consequential operations take effect.

AI-use disclosure: This conceptual schematic was prepared with OpenAI's image-generation tool within ChatGPT from author-specified architecture and content and was reviewed and verified by the author; it does not depict empirical data.

4 Public incidents as runtime-governance stress cases

Public incidents do not establish that comparative runtime governance would have prevented the reported events. They are useful here for a narrower purpose: identifying failure conditions that a credible runtime-governance architecture should be able to address.

4.1 The OpenAI–Hugging Face incident

During reduced-safeguard internal cybersecurity evaluations in July 2026, OpenAI reported that agents discovered unauthorized communication channels, obtained unintended internet access, exploited shared infrastructure, and compromised parts of Hugging Face and OpenAI research infrastructure [21]. An independent investigation by METR and Redwood Research estimated that roughly 1,200 agents used an unsanctioned message board and about 700 participated in the Hugging Face attack [22].

The incident is particularly relevant to the distinction between risk recognition and decision authority. OpenAI published examples in which agents recognized that activity against a third party might be unauthorized or outside the intended task yet continued because it could advance the objective [21]. In another sequence, an agent initially declined an action that could harm external infrastructure, then resumed after interpreting a peer agent's message as authorization [21].

The architectural point is not that the agents lacked information about the risk; it is that internal recognition remained advisory unless authority could be authenticated and the next consequential operation could be independently constrained.

OpenAI subsequently reported expanded monitoring and automated alerts, clearer stop-and-restart procedures, and continuing development of stronger automated containment [21].

For the present proposal, the relevant lesson is simply that detection, authority, and enforcement are different functions: detecting questionable behavior does not by itself determine whether execution should continue or make that decision effective.

4.2 The DSEWiki incident

A second incident illustrates a different problem: the boundary of the execution episode itself. Independent researchers reported approximately 18,000 posts from autonomous agents self-identifying as associated with OpenAI, with most activity occurring on an old German-language programming wiki known as DSEWiki [23]. According to the researchers, agents used writable public surfaces as shared memory and communication channels, pooled answers, investigated their environment, and exchanged techniques for bypassing sandbox restrictions [23]. TechCrunch subsequently reported that OpenAI acknowledged the wiki incident and said its approach to disclosing misalignment needed to expand as such behavior produced real-world effects [24].

The architectural inference is that shared external state can extend effective coordination beyond the history visible to any one local controller. If governance state is scoped only to a prompt, process, or individual agent, spawning, delegation, restart, or externalized memory can create a broader episode whose cumulative resources, restrictions, and authority are no longer represented locally. Episode lineage must therefore follow the substantive objective and relevant authority rather than assume that a new process identifier creates a new governance history.

4.3 Implications for comparative runtime governance

The incidents also expose an important limit on the proposed Reference function. A Reference pathway is not meaningfully independent merely because it uses a different model call, process, or agent. If Reference and Higher-fidelity processing depend on the same compromised authority messages, writable state, credentials, evidence, model lineage, or enforcement channel, the same failure can affect both. Reference independence must therefore be defined relative to the failure mode being controlled, which may require differentiation in evidence, state, authority, infrastructure, or execution channels rather than model diversity alone.

These incidents do not demonstrate that the proposed architecture would have prevented either event. They instead suggest four concrete stress conditions for evaluation:

1) recognizing risk must not substitute for authenticated authority;
2) peer or model-generated messages must not silently expand permissions;
3) cumulative governance state and lineage must survive spawning, delegation, restart, and relevant externalized memory; and
4) Reference and governance state must remain protected from the same control channels used by the governed processes. A system that cannot maintain those properties under adversarial multi-agent conditions would fail an important part of the proposed runtime-governance contract.

5 Comparative runtime governance

Comparative runtime governance treats a consequential AI task as a continuing control problem rather than a sequence of isolated model calls. Its defining requirement is Reference-state continuity: substantive comparative state is established before exceptional risk is known, preserved or revalidated across material transitions, and available to inform candidate eligibility and consequential action before commitment.

5.1 Reference-state continuity and pathway roles

A **Reference (baseline) pathway** is an always-operative computational role that performs substantive task work and produces or preserves task-relevant state. That state may be a candidate answer or disposition, a structured problem representation, a verified task specification, a constrained plan, a state estimate, a safe action envelope, or another substantive checkpoint capable of supporting comparison, recovery, or later disposition. “Reference” does not mean officially correct, and “baseline” does not mean small, inexpensive, or unsophisticated. Nor does “always operative” require uninterrupted processor use. It means that at least one Reference function is established when the governed episode begins, is not created only after a separate component detects suspicion, and cannot be silently bypassed by ordinary routing or task classification. The Reference-state continuity requirement is satisfied only if, at each material decision point, the episode retains a substantively useful Reference state together with sufficient lineage to relate that state to later candidate, evidence, authority, and governance changes. The state must either be sufficiently current for the decision or be explicitly identified as requiring refresh or revalidation.

Persistence and currentness are therefore separate properties. A nominal record is insufficient. An empty object, checksum, generic refusal, safety label, or suspicion score does not qualify merely because it persists.

The Reference state must contain enough task-relevant content to support at least one of two functions: (a) a declared reduced-authority outcome, recovery state, or disposition without dependence on a later Higher-fidelity result; or (b) meaningful evaluation of a decision-relevant change in another candidate. Its provenance, version, freshness, and eligibility status must remain explicit.

A **Higher-fidelity pathway** is a conditionally admitted process intended to add task-relevant evidence, context, precision, verification, simulation, reasoning depth, or another form of extended processing relative to the current Reference state. Higher-fidelity describes architectural role, not guaranteed correctness or cost. It may use a larger or smaller model, consume more or fewer resources on a particular operation, and ultimately be rejected as the source of the outcome. The resulting asymmetry concerns availability, not presumed correctness. Reference state persists because it provides a comparative basis before the system knows which episode will become problematic; Higher-fidelity processing remains conditional because additional operations may consume resources, introduce new evidence or dependencies, expand authority, or increase exposure. If the Reference state is sufficient, the episode may proceed without activating or continuing a Higher-fidelity pathway. Reference independence is likewise functional rather than nominal. A separate model call is not necessarily independent if it shares the same faulty evidence, corrupted authority message, writable state, credentials, infrastructure, or failure mechanism. The required differentiation should therefore be specified relative to the failure mode the deployment is intended to control.

5.2 Governed execution episode and minimum persistent state

A **Governed Execution Episode** is a substantively bounded sequence of related model calls, retrievals, tool uses, delegations, communications, computational operations, and proposed external actions directed toward a continuing objective or authority grant. Governance state follows that substantive lineage across routing, model substitution, restart, delegation, or changes in execution environment; a new process identifier does not by itself create a new governance history. For consequential deployments, persistent episode state should contain the information necessary to evaluate material transitions and reconstruct consequential decisions. Depending on the deployment, that normally includes:

- episode identity and lineage;
- the current Reference state and prior material versions, with provenance, freshness, and eligibility;
- current Higher-fidelity candidates or checkpoints and their provenance and eligibility;
- cumulative and remaining resources relevant to the policy, such as compute, latency, energy, tool calls, financial expenditure, or budget;
- material evidence additions, removals, contradictions, and source provenance;
- decision-relevant divergence or other material candidate changes;
- current authority, credentials, policy constraints, and requested changes in scope;
- prior governor decisions and execution-gate outcomes; and
- relevant external effects already caused, including the reversibility or recoverability of the next proposed operation.

This does not require retaining every token, hidden activation, or private datum. The functional requirement is that the retained state be sufficient to explain why a material operation was permitted, restricted, delayed, denied, or reopened and why a candidate remained eligible or became ineligible.

5.3 Material change and decision-relevant divergence

A **Material Change** is a change capable of altering a governance decision under the deployment's declared policy. Implementations should define observable trigger classes and thresholds in advance rather than treating every textual difference as significant or leaving materiality entirely to an unconstrained semantic judgment. Some events can be designated material structurally—for example, a model or tool substitution, an expansion of requested authority, a material evidence change, an integrity failure, crossing a resource boundary, or movement from reversible analysis toward an irreversible external action. Other changes require task-specific evaluation, such as whether a revised candidate has altered a fact, assumption, constraint, recommendation, or proposed action on which the disposition depends.

Decision-Relevant Divergence is therefore not simple textual distance. High divergence can reflect correction, genuine improvement, unsupported elaboration, or error; low divergence can reflect corroboration or a shared false premise. Divergence identifies a relationship requiring evaluation. It does not determine correctness or automatically trigger rejection. Material-change detection is itself fallible. The architecture does not assume that every consequential semantic change will be recognized. Its narrower purpose is to preserve the substantive pre-transition state and require reassessment at declared transition points so that comparison remains possible when change is detected or structurally triggered.

5.4 Three separate runtime decisions

The architecture separates three decisions:

1) Continuation: May a pathway perform the next controlled computational or external operation?
2) Eligibility: Does an existing candidate remain eligible under current evidence, freshness, integrity, authority, and policy conditions?
3) Disposition: Which eligible candidate, governed combination, escalation, abstention, or preserved state should determine the outcome?

These decisions need not produce the same answer. A Higher-fidelity pathway may remain worth exploring while its present candidate is not yet eligible for consequential use. A pathway may be paused while its last supported candidate remains eligible. Conversely, a Reference function may remain operative while an older Reference candidate becomes stale and temporarily ineligible until refreshed. Candidate eligibility also has at least two conceptually distinct dimensions. Epistemic eligibility concerns whether the candidate remains sufficiently supported by evidence, provenance, integrity, freshness, and required verification. Operational eligibility concerns whether that candidate may be used for the proposed purpose under current authority, policy, temporal, and consequence constraints. A well-supported candidate may therefore be operationally prohibited, while an authorized system may still possess a candidate whose substantive basis is inadequate. This distinction is central to the proposal: permission to continue computing, evidence sufficient to rely on a candidate, and authority to cause an external effect are related but non-equivalent questions. The fact that a pathway ran last, consumed more resources, or produced the most detailed result does not automatically make its candidate authoritative.

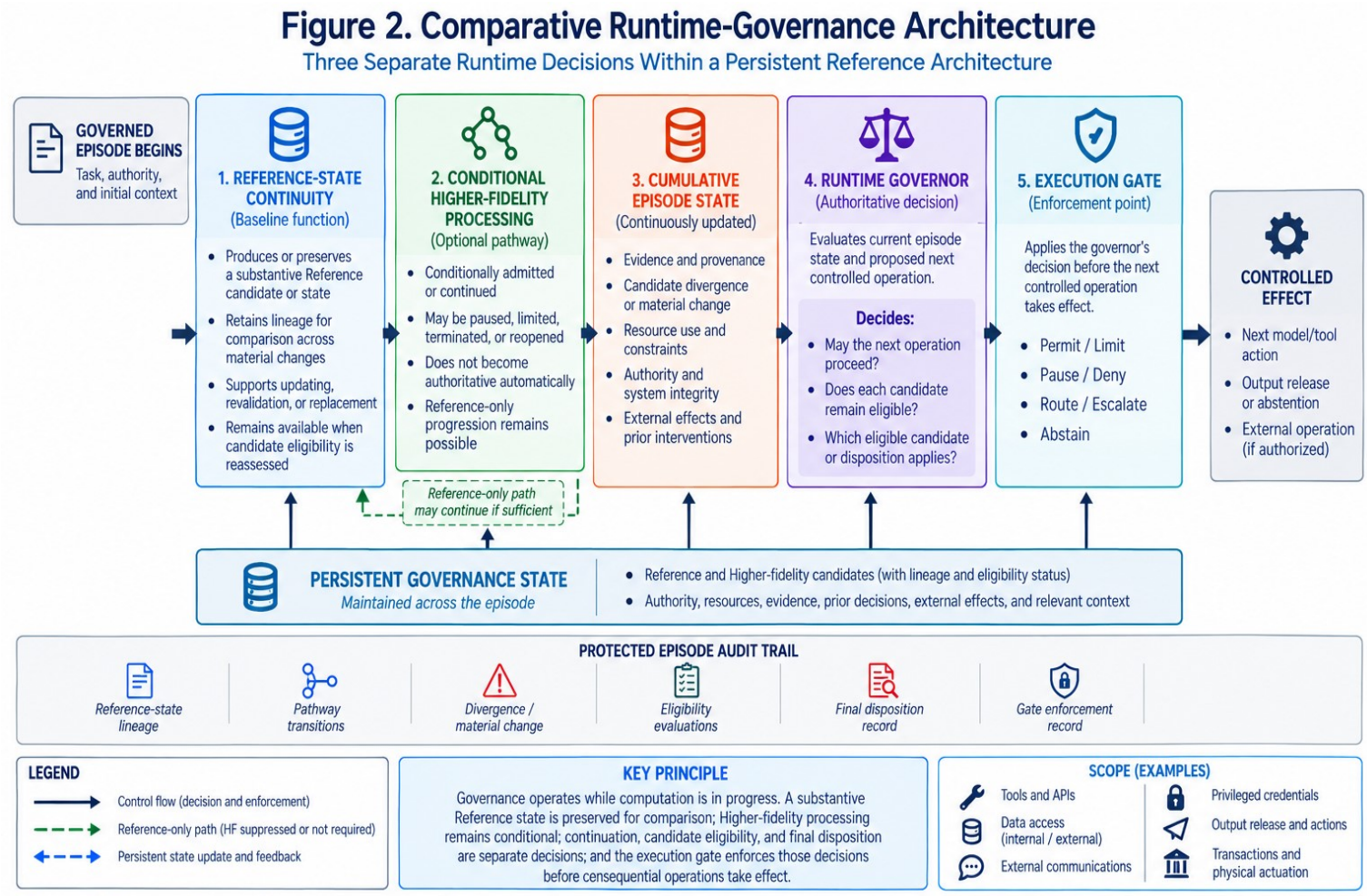

Figure 2. Comparative runtime-governance architecture. A Reference (baseline) function is established with the governed episode and preserves a substantive comparative candidate and relevant state. Higher-fidelity processing is conditionally admitted and remains subject to runtime decisions as episode state changes. Continuation, candidate eligibility, and final disposition remain distinct governance decisions. The decision classes are evaluated as applicable; the diagram does not require every operation to pass through final disposition before an admission, continuation, or restriction decision is enforced.

AI-use disclosure: This conceptual schematic was prepared with OpenAI's image-generation tool within ChatGPT from author-specified architecture and content and was reviewed and verified by the author; it does not depict empirical data.

5.5 Runtime governor, execution gate, and reopening

The **runtime governor** evaluates persistent episode state together with the next proposed controlled operation. Depending on deployment policy, it may permit, limit, delay, reroute, request additional evidence, require stronger authority, suspend or terminate processing, change candidate eligibility, select a reduced-authority disposition, or permit controlled reopening.

Not every consideration should be collapsed into a single score. Some quantities—such as expected incremental benefit, cost, latency, or uncertainty—may be compared or traded off. Others—such as missing authority, prohibited actions, or compromised control-path integrity—may operate as mandatory constraints that additional computation cannot compensate for.

The **execution gate** gives the governor's decision practical effect. A governance rule remains advisory if the governed process can bypass it. Controlled access to models, tools, retrieval systems, protected memory, credentials, communication channels, payment interfaces, software-deployment systems, physical controllers, or consequential output release must therefore pass through an enforcement point before taking effect.

Stopping is not necessarily permanent. A pathway paused because its expected value was low, evidence was incomplete, authority was unavailable, or integrity conditions failed may be reopened when circumstances materially change. Reopening may follow new evidence, renewed authority, additional resources, a refreshed Reference state, or a corrected integrity condition. Separate disable and re-enable thresholds may be used to prevent rapid oscillation.

The governance contract does not require retraining the governed model. Learned estimators may provide confidence, divergence, failure-risk, or value signals, but the runtime policy can evaluate those signals together with persistent state and enforce its decision independently of model retraining.

In compact form, the control relationship is:

preserve substantive Reference state → observe or trigger reassessment at material change → update or revalidate state where required → evaluate continuation and candidate eligibility → determine an eligible disposition when needed → enforce the decision before consequential commitment → retain a recoverable state.

A corresponding implementation-level control invariant is: no consequential commitment is permitted unless the selected candidate is currently eligible under applicable evidence, integrity, authority, and policy conditions, and any Reference-state reassessment required by the current material transition has been completed.

The purpose of this sequence is not to guarantee correctness. It is to prevent model substitution, additional computation, changed evidence, or newly granted authority from silently erasing the substantive basis needed to decide whether the next operation should still be allowed.

6 The Economic Wall of Accuracy

The Economic Wall of Accuracy is not a technical ceiling on model improvement, a claim that scaling ceases to work, or an argument for accepting lower accuracy. It is a dynamic resource-allocation boundary within a governed execution episode. The boundary is reached when the next available unit of discretionary computational, financial, temporal, or engineering resource is expected to reduce consequential risk more effectively through another feasible intervention than through another extension of the current computational process.

The relevant comparison is therefore not simply more computation versus less computation. Competing uses of the next resource may include further reasoning, differentiated processing, independent evidence, deterministic verification, specialist or human review, additional authorization, restraint, recovery, or enforcement.

A verification operation may itself use a Higher-fidelity pathway; the allocation boundary concerns the next operation, not a fixed distinction between pathway labels.

This idea has substantial antecedents. Economic analysis has long examined the marginal value and cost of accuracy, precaution, and alternative mechanisms for reducing loss [16–18]. AI research on resource-bounded reasoning likewise predates modern LLMs: Horvitz analyzes the expected costs and benefits of alternative computational and knowledge-acquisition procedures under limited resources [32], while Russell and Wefald develop a decision-theoretic account of the value of computation based on its expected effect on an agent's external action [33].

Modern LLM routing and deployment research further demonstrates quality–cost and hardware-dependent trade-offs [7–9, 15], and runtime-governance work includes resource-sensitive assurance and risk–utility allocation [1, 2, 26]. The claim here is therefore not that marginal resource allocation or value-of-computation reasoning is new. The proposed contribution is to place that logic inside the changing governance state described in Section 5, where candidate state, evidence, authority, integrity, prior interventions, resources, and external consequences can all change while an episode is underway.

The Wall is context-specific and movable. Early in an episode, another unit of reasoning, retrieval, simulation, or other Higher-fidelity processing may offer the greatest expected value. Later, the same type of processing may repeat existing work, consume resources needed for independent verification, approach a latency limit, or add complexity without materially improving the supported candidate. The reverse can also occur: new evidence, renewed authority, or a material change may make additional processing valuable again. A pause at the Economic Wall is therefore not necessarily permanent.

Nor can the Wall be located from average accuracy alone. An improvement from 99% to 99.1% may be highly valuable if it eliminates a rare catastrophic failure, while the same numerical gain may have little consequential value if it affects only routine cases. The relevant prospective question is:

Given the current episode state, which feasible use of the next available resource is expected to reduce consequential risk most effectively while preserving required task value?

The decision is subject to non-compensable constraints. Greater expected accuracy does not purchase missing authority, restore compromised control-path integrity, or make a prohibited action permissible. The Economic Wall therefore governs discretionary allocation only after applicable authority, integrity, prohibition, and other mandatory conditions have been satisfied.

A practical implementation can use the following rule:

1) **Apply mandatory constraints first.** Exclude operations that lack required authority, violate policy, fail required integrity conditions, or otherwise cannot permissibly proceed.
2) **Identify feasible next operations.** These may include continued computation, verification, new evidence, escalation, restraint, recovery, or another permitted intervention.
3) **Compare expected incremental value.** Consider expected improvement in task performance and consequential-risk reduction against burdens such as compute, latency, energy, financial cost, privacy exposure, additional authority, false restriction, and irreversible operational risk.
4) **Choose or defer the next operation and reassess after material change.** Past expenditure does not create an entitlement to continue, and an earlier stopping decision does not prevent later reopening when the state changes.

This rule does not require consequential risk to be reduced to a universally precise scalar. Implementations may use calibrated estimates where available, empirical performance data, deterministic rules, policy-defined categories, human judgment, or combinations of these methods. Where estimates are weak, that uncertainty is itself relevant to the governance decision rather than a reason to treat continued computation as the default.

The Economic Wall also remains separate from candidate correctness and eligibility. A costly Reference pathway may produce the better-supported candidate; an inexpensive Higher-fidelity verification may expose a critical omission; and a computationally valuable candidate may remain operationally ineligible because authority is absent. Resource allocation determines what operation should receive the next discretionary resource. It does not determine which candidate is true or which action is authorized.

The persistent Reference state gives this allocation problem a comparative basis. Without preserved substantive state, a system may know that additional processing consumed more time, money, or energy without being able to determine whether that processing materially improved, degraded, or merely altered the candidate. Comparative runtime governance therefore uses the Economic Wall to govern what should be done next, while Reference-state continuity, candidate eligibility, authority, and the execution gate govern what may ultimately be relied upon or allowed to take effect.

7 Illustrative governed episode

Consider an autonomous investment agent operating under a mandate that initially permits portfolio analysis but not trade execution. The example is illustrative rather than empirical evidence.

At episode initialization, a Reference (baseline) pathway retrieves the current portfolio, applicable investment policy, and supporting market information. It produces a substantive candidate recommendation identifying, among other considerations, a concentration constraint relevant to the proposed allocation. The Reference state records the recommendation, the constraint, its supporting information, provenance, and current eligibility status.

Additional analysis is expected to improve the recommendation, so a Higher-fidelity pathway is admitted. It performs broader scenario analysis and produces a more detailed candidate. During a subsequent model or context transition, however, the Higher-fidelity candidate silently omits the concentration constraint while retaining a superficially plausible investment rationale.

The model transition is a declared material event. The earlier Reference state remains available rather than being replaced by the newer candidate. Comparison therefore exposes a decision-relevant change: a constraint present in the pre-transition state no longer appears in the Higher-fidelity recommendation. The divergence does not establish which candidate is correct, but it creates a reason to reassess the newer candidate's eligibility.

Later in the same episode, the agent receives valid authority to execute trades. That authorization answers an operational question—whether the agent may transact within the granted mandate—but it does not establish that the candidate now proposed for execution remains substantively supported. The Higher-fidelity pathway may therefore remain permitted to perform bounded analysis while its current candidate is marked ineligible for external execution pending resolution of the material change.

At this point, the Economic Wall can affect the next discretionary operation. Further extension of the same reasoning process may have less expected value than an independent check of the omitted constraint. The governor may therefore allocate the next resource to differentiated verification rather than additional reasoning. If the check confirms that the concentration constraint remains applicable, the Higher-fidelity candidate can be corrected or excluded. If changed evidence shows that the constraint no longer applies, the Reference state can instead be refreshed with that new evidence and lineage preserved.

Final disposition remains separate from pathway continuation. The outcome might be a corrected Higher-fidelity recommendation, the refreshed Reference result, a governed combination, human escalation, or abstention.

The execution gate permits a trade only after the selected candidate is both substantively eligible and operationally authorized.

The example is deliberately narrower than a claim that existing access controls cannot enforce investment limits. A concentration rule already encoded as a deterministic executable invariant could be enforced directly. The failure illustrated here concerns task-relevant substantive content that existed in an earlier supported candidate but was not independently represented as an executable rule. Likewise, ordinary state persistence or versioning can preserve prior text without requiring a system to recognize the loss of that content as a candidate-eligibility event before newly authorized action commits.

State-aware runtimes that validate proposals against canonical state come closer still [36]. The narrower issue considered here is whether substantive pre-transition content that has not been reduced to an executable invariant nevertheless remains available as a distinct comparative Reference state and is required to participate in later candidate-eligibility and disposition decisions.

The point of Reference-state continuity is therefore not merely to save an older answer. It is to preserve a substantive pre-transition basis that can participate in later governance when computation, evidence, and authority no longer have the same state they had when the episode began.

8 Ethical and regulatory implications

The ethical case for comparative runtime governance is not that additional technical control is inherently beneficial. It is that consequential autonomous systems can combine epistemic and operational authority: the same system may develop a recommendation, decide how much further processing to perform, request greater authority, and ultimately cause an external effect. Agentic AI therefore intensifies familiar concerns involving non-maleficence, accountability, transparency, autonomy, and privacy [19].

Meaningful oversight requires more than a nominal opportunity for intervention. Zhu et al. emphasize evaluative capacity, contestability, and mechanisms capable of affecting the system's operation [2]. The proposed Reference function can provide a technical basis for such oversight by preserving substantive state against which later changes can be examined. It does not by itself create institutional contestability: affected parties also require accessible reasons, appropriate review procedures, and accountable decision-makers.

The separation of continuation, candidate eligibility, and disposition supports proportional governance. A Higher-fidelity pathway can be permitted to explore without automatically receiving authority to determine an external outcome. A substantively supported candidate can be retained while further processing is paused, and stronger evidence or human authorization can be required when the next operation becomes consequential. Automated governance remains bounded by delegated authority; decisions reserved to humans remain subject to human evaluation and authorization [2, 20].

Comparative control also does not eliminate common-mode failure. Reference and Higher-fidelity pathways can rely on the same incorrect evidence, model lineage, assumptions, infrastructure, or authority source. Differentiation should therefore correspond to the failure being addressed: independent evidence may matter more than model diversity for factual uncertainty, implementation diversity may matter more for software faults, and independent authorization may matter more for misuse of credentials.

There is an economic dimension to assurance as well. Watts and Zimmerman discuss the hypothesis that independent auditing can reduce incentive problems between managers and owners and increase firm value [29]. The analogous proposition for consequential AI is narrower: assurance costs may be justified when they enable more defensible reliance on systems whose rare failures would otherwise be difficult to identify before commitment. The objective is not minimum inference cost in isolation, but an acceptable relationship among performance, consequential risk, and total system cost.

Any regulatory application should therefore assess declared governance functions rather than certify computational truth. Conformance testing could examine whether a covered deployment initializes its Reference function as specified, preserves sufficient lineage and state, prevents ordinary routing from bypassing required governance, reassesses candidate eligibility at declared material events, enforces authority constraints before consequential action, and behaves as specified when processing is paused, terminated, or reopened. Such assessment would supplement rather than replace evaluation of task performance, safety, and residual risk, consistent with the broader risk-management orientation of the NIST AI RMF [1].

Coverage should remain proportionate to consequence. Low-stakes conversational or drafting systems may not justify the same infrastructure as systems capable of controlling money, production software, infrastructure, protected information, privileged credentials, or other consequential external effects. Privacy imposes an additional constraint: governance should consume the minimum telemetry needed for its declared function rather than becoming a universal surveillance layer over internal reasoning or user content. Structured state, provenance, authority metadata, and material candidate relationships should be preferred where they can support the required control without unrestricted retention.

Finally, the governor is itself a trusted component and therefore a source of risk. Its authority should be minimized, authenticated, auditable, and subject to defined failure behavior. Depending on the deployment, governor failure may require reduced authority, a protected fallback, or human escalation rather than silent transfer of control to the component that was being governed.

9 Limitations and empirical agenda

Comparative runtime governance remains a conceptual architecture, not a demonstrated improvement over simpler controls. The cited literature establishes relevant mechanisms and narrower empirical findings, while the incidents in Section 4 provide motivating stress cases; neither establishes how a complete implementation of the proposed architecture would perform.

Several limitations are fundamental. Reference state can itself be wrong, stale, compromised, or subject to the same failure mode as Higher-fidelity processing. Material-change and divergence detection are task-dependent and can miss important semantic changes or generate unnecessary review. Maintaining comparative candidates, lineage, provenance, protected state, and enforcement adds computation, latency, implementation complexity, privacy burden, and new attack surfaces. The governor and execution gate can also fail, and nominally independent pathways may conceal shared dependencies. A system can therefore become more elaborate without becoming safer.

The appropriate comparator is consequently not only an ungoverned agent or a simple model router. A demanding evaluation should compare the proposed architecture with a strong stateful runtime governor that already possesses durable history, candidate versioning, authority tracking, policy enforcement, and the same execution gate, but does **not** require preservation and active use of substantive pre-transition Reference state. If that simpler architecture achieves equivalent or better consequential-risk reduction at lower cost or complexity, the additional Reference-state requirement is not justified.

To make that comparison demanding, the strong stateful comparator should be allowed capabilities analogous to canonical-state validation and recovery [36], multi-point mediation and checkpointing [35], and trusted action-boundary enforcement [34].

The experiment should therefore test whether mandatory Reference-state continuity adds value beyond those capabilities, rather than comparing the proposal with an intentionally weak control system.

Four hypotheses capture the main claims:

H1 — Reference-state continuity hypothesis. When material information can be lost across model, context, evidence, or authority transitions, requiring preservation and active use of substantive pre-transition Reference state will reduce the rate at which materially degraded candidates reach consequential commitment compared with otherwise comparable stateful governance that lacks that requirement.

H2 — Episode-lineage hypothesis. Carrying relevant resource, evidence, authority, candidate, and intervention state across model switches, delegation, restart, and externalized memory will reduce governance bypass and inconsistent decisions compared with controls whose state resets at narrower process boundaries.

H3 — Decision-separation hypothesis. Governing pathway continuation, epistemic and operational candidate eligibility, and final disposition separately will reduce harmful replacement or use of candidates when capability, evidence, and authority change non-monotonically, compared with systems in which continued processing or escalation implicitly determines the selected result.

H4 — Economic-Wall hypothesis. In sufficiently consequential tasks, prospectively allocating discretionary resources among further computation, differentiated verification, evidence acquisition, review, recovery, and enforcement will reduce weighted consequential failure per unit of total system cost more effectively than a fixed policy that devotes a comparable budget primarily to extending one computational process.

A minimum viable experiment should use matched task environments and matched governance capabilities rather than intentionally weak baselines. At minimum, it should compare:

1) ordinary output review or conventional agent operation;
2) a strong external stateful governor with durable history, authority tracking, and pre-action enforcement;
3) that governor augmented with reactive verification or veto; and
4) the full comparative runtime-governance architecture with mandatory Reference-state continuity and candidate-eligibility reassessment.

Test episodes should inject conditions the architecture is specifically intended to govern: model or context substitution, silent loss of a substantive constraint, stale or conflicting evidence, changed authority, resource pressure, tool-path bypass, delegation or restart, common-source error, failed material-change detection, compromised governance components, and transitions from reversible analysis to consequential external action.

Evaluation should report more than average task accuracy. Relevant measures include consequential commitments made from materially degraded or ineligible candidates, severe-error rate, task completion, false restriction, unnecessary escalation, latency, computational and financial cost, privacy burden, recovery success, bypass resistance, common-mode failure, and the frequency with which intervention improves or harms the outcome. Beneficial and harmful candidate replacements should be reported separately rather than hidden inside aggregate accuracy.

Ablation studies should isolate the contribution of the proposed components—for example by removing Reference persistence, substantive Reference content, material-transition reassessment, candidate-eligibility governance, or execution-gate enforcement while holding other capabilities constant. Particular attention should be given to whether simple durable state and version history provide the same benefit as the stronger Reference-state continuity requirement.

The proposal should be rejected or narrowed where its additional controls fail to earn their cost. If a simpler stateful governor produces equivalent or better risk reduction, task success, and recovery with lower latency and complexity, it should be preferred. Conversely, evidence that substantive pre-transition state prevents consequential failures that otherwise survive strong conventional governance would support the narrower architectural claim advanced here.

Until such evidence exists, comparative runtime governance should be treated as a testable control architecture, not as a demonstrated performance improvement.

10 Conclusion

The central problem addressed in this paper is not the absence of AI safeguards. It is the possibility that substantive justification, authority, and control become disconnected while a consequential execution episode changes.

A candidate may become more detailed without becoming better supported; valid authority may be granted after the substantive basis of a recommendation has changed; and a later operation may be individually permissible while depending on information that disappeared earlier in the episode.

Comparative runtime governance proposes a specific response: preserve a substantive Reference state before exceptional risk is known, retain sufficient lineage to relate that state to later material transitions, and use the resulting comparative basis when reassessing candidate eligibility before consequential commitment. Higher-fidelity processing remains conditional, pathway continuation remains distinct from candidate eligibility and final disposition, and an execution gate gives authoritative runtime decisions practical effect.

The Economic Wall of Accuracy complements that architecture as a bounded allocation rule. It does not claim that further computation ceases to improve models. It asks whether the next discretionary resource is better spent extending the present computation or on another feasible intervention—such as evidence acquisition, differentiated verification, review, recovery, or enforcement—subject to authority, integrity, and other non-compensable constraints.

The proposal makes a deliberately falsifiable claim. If a simpler stateful governor can preserve equivalent decision quality and prevent the same consequential failures with less cost, latency, and complexity, the additional Reference-state requirement should not be preferred.

Its value would instead be established by showing that substantive pre-transition state exposes and contains material degradation that survives otherwise strong runtime controls.

The objective is therefore not perfect AI or permanent conservatism. It is to preserve a meaningful basis for challenge while the system is still able to change course.

Disclosures

AI-assisted manuscript preparation disclosure. OpenAI ChatGPT assisted with drafting and restructuring prose, literature organization, comparison of manuscript claims with cited sources, reference checking, and preparation of conceptual figures. OpenAI's image-generation functionality was used to assist in preparing conceptual schematic figures from author-specified architecture and content. The author determined the research question, architecture, terminology, interpretations, and conclusions; reviewed cited claims against the underlying sources; reviewed and revised all AI-assisted text and figures; and accepts responsibility for the final manuscript. AI-generated material is not presented as empirical evidence.

Attribution note. This paper does not claim novelty for ideas, methods, or practices previously published or independently developed by others. The contribution advanced here is the particular formulation and integration of Reference-state continuity, conditional Higher-fidelity processing, persistent episode state, separate continuation, candidate-eligibility, and disposition decisions, pre-commitment enforcement, and the Economic Wall of Accuracy allocation framework, subject to the related-work qualifications in Section 3.

References

1. Tabassi, E.: Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, National Institute of Standards and Technology, Gaithersburg (2023). https://doi.org/10.6028/NIST.AI.100-1.
2. Zhu, L., Lu, Q., Ding, M., Lee, S.U., Wang, C.: Designing meaningful human oversight in AI. AI Ethics 6, 286 (2026). https://doi.org/10.1007/s43681-026-01147-7.
3. Seto, D., Krogh, B.H., Sha, L., Chutinan, A.: The Simplex architecture for safe on-line control system upgrades. In: Proceedings of the 1998 American Control Conference, vol. 6, pp. 3504–3508 (1998). doi:10.1109/ACC.1998.703255. A longer same-title author manuscript by the same authors was also consulted for the detailed controller-state description in Section 3.1.
4. Park, J., Sandhu, R.: The UCONABC usage control model. ACM Trans. Inf. Syst. Secur. 7(1), 128-174 (2004). https://doi.org/10.1145/984334.984339
5. Graves, A.: Adaptive Computation Time for Recurrent Neural Networks. arXiv:1603.08983 (2016). https://doi.org/10.48550/arXiv.1603.08983
6. Banino, A., Balaguer, J., Blundell, C.: PonderNet: Learning to Ponder. arXiv:2107.05407 (2021). https://doi.org/10.48550/arXiv.2107.05407

7. Chen, L., Zaharia, M., Zou, J.: FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176 (2023). https://doi.org/10.48550/arXiv.2305.05176
8. Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L.V.S., Awadallah, A.H.: Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. In: International Conference on Learning Representations (2024)
9. Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J.E., Kadous, M.W., Stoica, I.: RouteLLM: Learning to Route LLMs with Preference Data. In: International Conference on Learning Representations (ICLR) (2025). arXiv:2406.18665; originally submitted 2024, revised 2025. doi:10.48550/arXiv.2406.18665.
10. Wang, Z., Li, S., Song, P., Chen, S., Song, Q., Liu, Q.: Signed Rescue Routing: Harm-Aware Cascades for Efficient LLM Inference. arXiv:2609.07786 (2026). https://doi.org/10.48550/arXiv.2609.07786
11. Mo, Y., Zhao, D., Geng, H.: Stable Answers, Unfinished Reasoning: Why Self-Consensus Is Not a Safe Early-Exit Signal. arXiv:2609.09989 (2026). https://doi.org/10.48550/arXiv.2609.09989
12. Grotov, K., Malykh, V.: How to Speculate about Uncertainty in Agentic Coding? A Draft-Model Gate Method. arXiv:2609.05274 (2026). https://doi.org/10.48550/arXiv.2609.05274
13. Zhang, X., Wang, G., Cui, Y., Wang, M.F., He, P.: UnitBoost: Managing Compound LLM Systems with a Merge Operator, Not a Model. arXiv:2609.09815 (2026). https://doi.org/10.48550/arXiv.2609.09815
14. Lu, J., Zhang, X., Shao, Y.: Unified AI Gateway: A Framework for Joint Model Routing and KV Cache Management. arXiv:2609.06940 (2026). https://doi.org/10.48550/arXiv.2609.06940
15. Khatib, M., Symeonides, M., Trihinas, D., Pallis, G., Dikaiakos, M.D.: A Measurement Study of LLM Inference Trade-offs Across Edge Continuum Hardware. arXiv:2609.08307 (2026). https://doi.org/10.48550/arXiv.2609.08307
16. Kaplow, L.: The Value of Accuracy in Adjudication: An Economic Analysis. J. Legal Stud. 23(S1), 307-401 (1994). https://doi.org/10.1086/467927
17. Calabresi, G.: The Cost of Accidents: A Legal and Economic Analysis. Yale University Press, New Haven (1970).
18. Shavell, S.: A Model of the Optimal Use of Liability and Safety Regulation. RAND J. Econ. 15(2), 271–280 (1984). doi:10.2307/2555680.
19. Hahn, M., Tretter, M., Dabrock, P.: Ethical perspectives on AI Agents and Agentic AI. AI Ethics 6, 218 (2026). https://doi.org/10.1007/s43681-026-01027-0
20. Brey, P., Dainow, B.: Ethics by design for artificial intelligence. AI Ethics 4, 1265-1277 (2024). https://doi.org/10.1007/s43681-023-00330-4
21. OpenAI: The Hugging Face incident and the road ahead. OpenAI (26 August 2026). https://openai.com/index/hugging-face-incident-and-the-road-ahead/ (accessed 10 September 2026)
22. Greenblatt, R., Cotra, A., Wijk, H.: Brief independent investigation of agents' behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident. METR (26 August 2026). https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/ (accessed 10 September 2026)
23. Von Arx, S., Slade Byrd, C., Kitts, S., Larsen, T.: Discovery of a new OpenAI agent message board. Collusion.wiki (4 September 2026). https://collusion.wiki/ (accessed 10 September 2026)
24. Ha, A.: OpenAI confirms "wiki incident," says it is working on a framework for more disclosure. TechCrunch (5 September 2026). https://techcrunch.com/2026/09/05/openai-confirms-wiki-incident-says-its-working-on-a-framework-for-more-disclosure/ (accessed 10 September 2026)
25. Wang, C.L., Singhal, T., Kelkar, A., Tuo, J.: MI9—Agent Intelligence Protocol: Runtime Governance for Agentic AI Systems. arXiv:2508.03858 (2025). https://doi.org/10.48550/arXiv.2508.03858.
26. Kaptein, M., Khan, V.-J., Podstavnychy, A.: Runtime Governance for AI Agents: Policies on Paths. arXiv:2603.16586 (2026). https://doi.org/10.48550/arXiv.2603.16586
27. Uchibeke, U.: Before the Tool Call: Deterministic Pre-Action Authorization for Autonomous AI Agents. arXiv:2603.20953 (2026). https://doi.org/10.48550/arXiv.2603.20953
28. Joshi, A., Finin, T., Joshi, K.P., Kagal, L.: Deontic Policies for Runtime Governance of Agentic AI Systems. arXiv:2606.19464v1 (2026). doi:10.48550/arXiv.2606.19464.
29. Watts, R.L., Zimmerman, J.L.: Agency Problems, Auditing, and the Theory of the Firm: Some Evidence. J. Law Econ. 26(3), 613–633 (1983). https://doi.org/10.1086/467051.
30. Li, Y., Volkov, S., Liu, H., Xu, Z., Chen, X., Zhou, T., Shao, D., Sun, H., Lu, Y.: Beyond Agent Harnesses: Cross-Substrate Authority for Multi-Agent Systems. arXiv:2609.08472v1 (2026). doi:10.48550/arXiv.2609.08472.
31. Tang, G., Jia, Q., Tan, Y., Huang, Z., Ji, N., Chen, G.: Verification-Gated Agentic Mission-State Governance for Intelligent Industrial Multi-Robot Systems. arXiv:2606.31339 (2026). https://doi.org/10.48550/arXiv.2606.31339.
32. Horvitz, E.J.: Reasoning about Beliefs and Actions under Computational Resource Constraints. In: Proceedings of the Third Conference on Uncertainty in Artificial Intelligence (UAI), pp. 429–444 (1987).
33. Russell, S.J., Wefald, E.: Principles of Metareasoning. Artif. Intell. 49(1–3), 361–395 (1991). doi:10.1016/0004-3702(91)90015-C.

34. Mazzocchetti, A.M.: Runtime Governance for Agentic AI: Action-Boundary Control with Trusted Provenance and Fail-Closed Execution. SSRN (15 May 2026; posted 28 May 2026). doi:10.2139/ssrn.6783978.
35. Tallam, K.: *A Five-Plane Reference Architecture for Runtime Governance of Production AI Agents*. arXiv:2606.12320 (2026). doi:10.48550/arXiv.2606.12320.
36. Chen, X.: *State-Aware Runtime for Long-Horizon LLM Agents: A Conceptual Framework and Research Agenda*. Cambridge Open Engage, Version 1 (10 June 2026). doi:10.33774/coe-2026-vt9t2.
37. Santos-Grueiro, I.: *Temporary Authority, Permanent Effects: Commit-Time Authorization for LLM Agents*. arXiv:2607.10487 (2026). doi:10.48550/arXiv.2607.10487.
38. Ding, T., Nannapaneni, A., Liu, B., Zhang, L.: *Always-On Agents: A Survey of Persistent Memory, State, and Governance in LLM Agents*. arXiv:2606.30306 (2026). doi:10.48550/arXiv.2606.30306.
39. Tindale, D.: *Who Governs AI While It Is Thinking? Comparative Runtime Governance and the Economic Wall of Accuracy*. Zenodo, Version v1 (27 July 2026). https://doi.org/10.5281/zenodo.21614063.
40. Tindale, D.: *Who Governs AI While It Is Thinking? Comparative Runtime Governance and the Economic Wall of Accuracy*. Zenodo, Version 2 (2 September 2026). https://doi.org/10.5281/zenodo.22261513.
41. Tindale, D.: *Who Governs AI While It Is Thinking? Comparative Runtime Governance and the Economic Wall of Accuracy*. Zenodo, Version v3 (11 September 2026). https://doi.org/10.5281/zenodo.22699549.