ENTERPRISE GOVERNANCE · CANONICAL PUBLICATION

The Runtime Governance Thesis

Defining Governance for Autonomous Execution

Enterprise governance evolves whenever technology changes the object requiring trust. Autonomous systems change that object from principals and requests to autonomous execution. Runtime Governance is the discipline that evaluates whether execution remains a legitimate exercise of enterprise authority.

PUBLICATION RECORD

Version	1.0
Published	July 2026
Evidence Corpus	Schema v6 · 37 sources · 9 claims DERIVED
Framework Series	FW-001 – FW-003 DERIVED
Research Status	Active
Amendments	Publication & Amendment Policy

Executive Summary

For most of enterprise computing, the person who held authority was also the person who chose the action. Autonomous systems break that. An agent receives an objective, selects its own execution path, retrieves enterprise information, invokes tools, delegates to other agents, and adapts as it goes. The sequence that results was chosen by the agent, not by the principal whose authority it runs on.

Enterprise security answers three questions well, and answers them continuously: who is acting, whether they authenticated, and what policy permits. None of the three reaches that sequence. The principal is authentic. The credential is unexpired. Policy returns *permit*. And the action is still not something the organization conferred authority to do.

Runtime Governance is the discipline that makes the missing determination. It evaluates whether a specific execution remains a legitimate exercise of conferred authority at the moment work occurs, and records what that evaluation concluded as independently reviewable evidence. It names one evaluation object — the execution event — and defines five conditions under which execution can be considered legitimate.

The contribution is semantic rather than architectural. It is not an enforcement layer, a protocol, or a product category, and it replaces nothing: it depends on Identity and Access Management, Zero Trust, authorization, and AI governance to function at all. Of the five conditions, one is already well covered by continuous authorization and one is substantially covered by delegation standards. Three depend on information that no current standard specifies how to construct. That gap is the entire claim.

What follows states the thesis in full, establishes why existing governance no longer reaches the question, presents the evidence that a discipline is emerging rather than being invented here, describes the research program maintained around it, and addresses implementation. The Evidence Map catalogs the work that challenges the thesis alongside the work that supports it.

The Runtime Governance Thesis

The object of governance has changed

Every major advance in enterprise computing has required organizations to expand what they govern.

Early enterprise security governed networks: could a system communicate with another system? As enterprises interconnected, the object shifted to identities, and Identity and Access Management emerged to govern who was requesting access. Cloud, mobility, and distributed infrastructure then made static trust untenable, and Zero Trust extended governance to continuously evaluated requests.

Each transition followed the same pattern. Existing governance remained necessary and became insufficient, because the thing requiring trust had changed.

Autonomous execution is the next such change, and the reason is structural rather than rhetorical.

The coupling that broke is worth stating precisely. A person authenticated, requested an action, interpreted the result, and remained accountable for the outcome. Identity, intent, and execution travelled together because they originated in the same actor — and every governance model built since has quietly assumed they still do.

Autonomous systems break that coupling. An agent may receive an objective from a human, determine its own strategy, and produce a sequence of actions the initiating individual never anticipated. Three properties distinguish an agent action from a request: the requesting principal did not select it, the intent behind it is one or more delegation hops removed from its origin, and its causal trigger may be data the system ingested rather than an instruction a principal issued.

None of those three properties appears anywhere in the request tuple that authorization evaluates.

Agents are acquiring continuity across identity, memory, capability, and coordination. Governance must acquire the same continuity across authority.

A note on the name, since it appears to say otherwise. *Runtime* identifies the point at which authority becomes operational. It is not the defining characteristic of the discipline. The discipline is defined by what it evaluates — autonomous execution — not by when evaluation happens. Existing architectures already evaluate continuously. Evaluating a different object is the distinction.

The execution event

What Runtime Governance evaluates is the **execution event** as a composite object. At minimum it consists of the principal (who currently holds authority), the delegation chain (through what path authority was conveyed), the action, the target, the context (whether the organizational conditions that justified this authority still exist), the intent that initiated the chain, and the provenance of the instruction that triggered it.

Action and target are not tested against conditions of their own. They are what the evaluation is *about*. An execution event pairs an action against a target with the authority claimed to support it, and legitimacy is the determination that the pairing holds.

Why execution is the boundary

A reasonable objection: why govern execution rather than an earlier point — the decision an agent reaches, or the intent it forms? Three properties hold at execution and at no earlier point.

Authority is only exercised at execution. Authority is a relation between a principal and an effect on an enterprise resource. Before an effect occurs, authority is *held* but not *used*. Governance of held authority is authorization, and it is mature. Governance of used authority requires a moment at which use occurs. Earlier, the question is not merely harder to answer — it is not yet well-formed.

The evaluable object is only complete at execution. Intent and decision are internal states of a reasoning system: continuous, revisable, often not discrete artifacts at all. The target of an action, its parameters, and the context determining whether it is appropriate are frequently unresolved until invocation. Evaluation performed earlier operates on an incomplete object, and evidence attached to it records something the system may never do.

Earlier boundaries do not bind. Because autonomous execution is non-deterministic, an approved plan is not a commitment to execute that plan. Governing the decision yields a guarantee about an artifact that may be superseded before it runs.

This does not make earlier governance valueless. Model evaluation, tool scoping, guardrails, and human approval of objectives all reduce how often illegitimate executions are attempted, and they complement the discipline proposed here. The claim is narrower: earlier controls cannot discharge the question, because the question concerns an exercise of authority that has not yet occurred.

Execution legitimacy

If the evaluation object is the execution event, the discipline needs a precise account of what makes that object legitimate. Without one, legitimacy collapses into permission, and Runtime Governance collapses into authorization.

> *Authority answers who may act. Execution legitimacy answers whether this action remains a legitimate exercise of that authority.*

The definition is constructed to be testable. Each condition can be evaluated independently, can fail independently, and can be recorded as evidence. **An execution event is legitimate when all five conditions hold at the moment execution occurs.**

Authority Validity. Every authority in the chain — the originating principal's, each intermediate delegate's, and the executing agent's own — is currently valid rather than merely valid when granted.

Authority Provenance. The action traces to authority actually conferred, through an unbroken delegation chain in which no hop conveys more authority than the hop above it held. Authority may narrow across a chain; it may not widen.

Contextual Validity. The organizational condition that made the authority appropriate still holds — the matter is open, the project is active, the contract is in force, the lawful basis for processing still applies.

Intent Conformance. The action falls within the scope of the human or organizational objective that initiated the execution chain. It is not merely permitted by the principal's entitlements; it serves what was actually requested.

Causal Integrity. The action originates from principal intent rather than from content the system ingested. Instructions arriving through data channels do not inherit the authority of the system that read them.

Governance evidence

A determination that is not recorded is not governance. The discipline therefore produces **governance evidence**: a record of what legitimacy determination was made, which authority and contextual signals informed it, and why the execution was permitted, denied, escalated, or attested.

This is not the same as logging. Authentication logs show who signed in. Authorization logs show that policy permitted an operation. Audit logs reconstruct what happened. All remain essential, and none of them establishes governance reasoning. Logs may contribute to evidence; event records alone do not constitute it.

Why Existing Governance Is No Longer Sufficient

The previous section defined the discipline. This one establishes the gap it addresses, and states precisely how large that gap is — because the size of the claim is what makes it testable.

What existing disciplines already establish

Authority Validity is comprehensively addressed by continuous authorization and continuous access evaluation. This is what those architectures were built for, and they do it well.

Authority Provenance is substantially addressed in specification and partially in deployment. Delegation chains are structurally representable, and chain validation is active standards work — though non-widening is not generally enforced by construction.

Contextual Validity, Intent Conformance, and Causal Integrity are the domain proposed for Runtime Governance. The claim is not that existing systems are incapable of evaluating them; several are architecturally capable of consuming any input supplied to them. The claim is that no standard currently specifies how those inputs are *constructed*, and that in the absence of construction, evaluative capacity goes unused.

Of five conditions: one well covered, one substantially covered, three dependent on information no current standard produces. That is the entire contribution, and it is deliberately small.

Security correctness is not governance legitimacy

A system may be entirely correct and entirely illegitimate at the same time.

Security correctness asks whether systems behaved according to their configured rules. Governance legitimacy asks whether those rules still represented valid organizational authority for the action that actually occurred. The technical paper constructs a scenario in which every existing control returns the correct answer by its own criteria while three of the five conditions fail. It is offered as a falsification test: if such a scenario cannot be constructed, this discipline is unnecessary.

Authority drift

The divergence between conferred authority and current legitimacy is not static. It opens continuously as organizational conditions change — employment ends, matters close, projects conclude, contracts expire, classifications are revised — whether or not anyone observes it.

Authority drift is the term for that divergence. It is distinct from *entitlement drift* and *privilege creep*, which describe a principal accumulating excess entitlements over time. Authority drift is narrower and temporal: the gap at a given moment between the authority a system holds and the authority its circumstances would justify, irrespective of whether the entitlement itself is appropriate. A technically correct entitlement may still be illegitimate in current context.

Access certification detects drift episodically. Continuous evaluation detects the subset that manifests as identity-plane events. Neither observes drift originating in business systems, and neither expresses it as a measurable quantity.

Authority drift is developed as an operational condition in the Runtime Governance Blueprint and tracked as a recurring pattern in Accumulated Evidence #001.

The boundary of the claim

The argument depends on assumptions that may not hold, and states them rather than assuming them away.

It assumes that business systems can be consulted at execution latency, that intent can be represented well enough to evaluate conformance against it, that delegation chains can be correlated across organizational boundaries, and that governance effectiveness can eventually be measured rather than asserted. None of these is settled. Each is stated as an open question in the technical paper rather than treated as established.

Evidence for an Emerging Discipline

A proposed discipline should be falsifiable, and falsifiability requires that the supporting and opposing literature be visible rather than asserted.

The Runtime Governance Thesis is supported by an independently maintained Evidence Map cataloging primary research, standards work, vendor architectures, market developments, and adjacent research. Each source is classified by the claim it touches, its stance toward that claim, and its relation to the discipline proposed here. Inclusion does not imply endorsement. The map describes the state of the literature, not the state of AO Integrity.

Three properties of the map matter for how this thesis should be read.

It is maintained separately. Supporting evidence can be added, corrected, or withdrawn without amending the thesis, and the thesis can be revised without rewriting the corpus. The two move at different speeds because they answer different questions. Where the evidence contradicts a claim made here, the map says so, and the thesis is what changes.

It records work that narrows this claim. Several catalogued sources locate governance at the execution, execution path, action, or bind boundary independently of this work. At least one vendor markets the term *runtime governance* for a policy-enforcement layer, and at least one uses *runtime admissibility* directly. Where a standard or specification closes one of the five conditions, the map records that the corresponding claim has narrowed.

It is open to contribution. Sources that make the thesis less comfortable are the most useful additions to it.

That convergence is the substance of this section. A discipline that only one organization describes is a vocabulary; a discipline that multiple independent parties arrive at from different directions is emerging. The most useful response to this thesis remains a demonstration that it renames something that already exists.

Open the Evidence Map → · Submit a source →

The Runtime Governance Research Program

Runtime Governance is developed as a research program rather than a single document. Each component serves a different role, and the separation is deliberate.

Runtime Governance Thesis — this document. The canonical statement of the discipline, its evaluation object, and its bounded claim. Versioned and citable.

Evidence Map — the independently maintained corpus of research, standards, and vendor work bearing on the thesis, classified by claim and stance.

Framework Series — the operating models that translate the thesis into practice, beginning with the Runtime Governance Blueprint.

Research Briefs — short-form analysis of market signals, standards movement, and framework commentary as the field develops.

Accumulated Evidence — the ongoing series collecting proof points, recurring patterns, and operational examples.

Governance Questions — the specific questions the discipline exists to answer, each developed individually.

The thesis states the claim. The Evidence Map tests it. The frameworks operationalize it. The briefs follow the field as it moves.

Separating these artifacts is what allows the thesis to remain a claim rather than becoming a catalogue. A thesis that absorbed its own supporting evidence would have to be rewritten every time the evidence changed, and would gradually stop being falsifiable at all.

Canonical precedence

Separating artifacts creates a second requirement: when their definitions differ, something has to govern. This document does.

The order of precedence is:

Runtime Governance Thesis → Lexicon → Framework Series → Research Briefs → Governance Questions → Articles

A definition in a lower artifact that conflicts with this one is an error in that artifact rather than an alternative reading, and should be corrected. The single exception is an artifact that departs deliberately and says so, naming the thesis version it departs from and why. Silent divergence is not a variant; it is drift, and it is the same failure this discipline describes in a different register.

Reported inconsistencies are treated as defects. Tell us where you find one.

How this document itself changes — what constitutes a major revision, how superseded versions remain retrievable, and what a major revision obliges of derived artifacts — is set out in the Publication & Amendment Policy.

Implementation

Architectural neutrality is a property of the discipline, not an evasion of the engineering question. Organizations must still decide where evaluation occurs, and the available patterns differ in ways that determine which suits a given class of action.

Inline enforcement. Evaluation occurs in the execution path, and the action does not proceed until legitimacy is determined. This is the only pattern that prevents illegitimate execution rather than documenting it, which suits irreversible actions — funds movement, external disclosure, data deletion, production configuration change, physical actuation. It adds latency, places the evaluator on the critical path, and requires an explicit failure posture. That last point is frequently deferred and should not be: an organization that has not decided, per action class and in advance, whether unavailability halts execution has decided to fail open by default.

Independent attestation. Evaluation occurs out of band, observing execution rather than gating it. Attestation prevents nothing, but it can be positioned where the executing system cannot silently disable or bypass it, which makes its evidence considerably harder to dispute. It imposes no latency

and supports higher assurance precisely because the attesting party is architecturally separate from the executing one. Its limitation is inherent: an illegitimate action detected after execution has still occurred.

Hybrid. Most enterprises will require both — inline enforcement for defined high-consequence action classes, attestation for the remainder. The allocation should be made against consequence and reversibility rather than volume or convenience.

Evidence also varies in how much independence it carries, from self-logging, where verification requires trusting the producer entirely, through signed and tamper-evident records, to evaluation independently attested by a party distinct from the executing system. Most enterprise logging operates at the weakest level. The common failure is not falling short of the strongest one; it is describing a self-logged control as fully audited. Stating the level explicitly is itself a governance improvement.

The discipline does not require a platform

A reasonable objection to all of the above is that intent construction, business context construction, and delegation construction could be built inside an existing policy engine — and that if they were, no new architecture would have been introduced.

This thesis accepts that outcome. If an organization constructs those inputs and evaluates the five conditions within an existing policy engine, identity platform, or orchestration framework, it has implemented Runtime Governance. Nothing here requires the discipline to exist as a standalone platform.

This is the ordinary condition of a governance discipline. Zero Trust is not a product. Defense in depth is not a product. Both are implemented inside architectures that predate them, and both are useful because they establish what a system should be evaluated against rather than what component performs the evaluation. Runtime Governance is proposed on the same terms, and the argument stands or falls on whether the evaluation model is correct — not on where it runs.

The assurance scale, the decision receipt structure, and the verification requirements are specified in the technical paper and operationalized in the Framework Series.

Conclusion

Enterprise governance has evolved whenever technology changed the object requiring trust. Networks required governance of communication. Identity required governance of principals. Zero Trust extended governance to continuously evaluated requests.

Autonomous execution introduces the next such transition. Its distinguishing contribution is not evaluating at runtime — existing architectures already evaluate continuously — but evaluating a different object.

> ***Enterprise governance evolves whenever the object requiring trust changes.***
>
> ***Autonomous execution changes that object.***
>
> ***Runtime Governance is the discipline that evaluates whether autonomous execution remains a legitimate exercise of enterprise authority.***

Runtime Governance is proposed not as a competing architecture, but as a common vocabulary for evaluating autonomous execution. Whether the discipline ultimately becomes part of identity, Zero Trust, authorization, AI governance, or a distinct field will be determined by practitioners, standards bodies, and enterprise adoption. The purpose of this work is to provide a framework that can be challenged, refined, and tested.

Cite this work

Thornbladh, J. (2026). *The Runtime Governance Thesis: Defining Governance for Autonomous Execution.* Version 1.0. AO Integrity Research.

The complete technical paper — including the standards analysis, illustrative scenarios, falsification case, and full reference list — is available as a versioned PDF.