A 42 CONSULTING.AI WHITEPAPER

The Runtime Authority Problem

Why AI Governance That Stops at Deployment Is Half a Governance Framework

Dr M Maruf Hossain, PhD, GAICD
Chief AI Strategist · 42 Consulting.AI
enquire@42consulting.ai · www.42consulting.ai

42 Consulting.AI
Scaling Intelligence

CONTENTS

Table of Contents

FOREWORD

From the Author.

Almost every AI governance framework I have examined does its most serious work at a single moment: the point of deployment. It assesses the system, documents the risk, approves the use, and records the decision. And then — for months, sometimes years, until the next scheduled review — it goes quiet.

This has always struck me as a strange place to concentrate governance. The deployment decision is a single act. The exercise of the authority granted by the decision is continuous. An AI system does not do its work at the moment it is approved; it does its work every day afterwards, under conditions that shift continuously and silently away from the ones under which it was approved.

A framework that governs the moment of approval, not the period of operation, is not a complete framework. It is half of one — and it is the wrong half, because the risk does not live at deployment. It lives in the long, ungoverned interval that follows.

This paper is about that interval, and about the two ideas that allow an organisation to govern it rather than merely hope through it. It is the most technical of the arguments I have set out, and the one I believe most clearly separates a governance framework that works from one that only appears to.

M Maruf Hossain

SECTION ONE

Executive Summary.

Most AI governance frameworks govern a moment. The risk lives in the interval that follows. This whitepaper establishes three propositions for the executives responsible for AI risk.

First, authorisation is a point in time; authority is exercised continuously. When an organisation approves an AI system for use, it makes a single decision at a single moment — call it deployment. But the authority granted by the decision is not exercised at that moment. It is exercised every day afterwards, across thousands of actions, under conditions that drift continuously away from those that existed at approval. Governance concentrated at the moment of approval addresses the smallest part of the system's actual operating life.

Second, a framework that stops at deployment is half a framework — and the wrong half. The prevailing pattern is to govern two moments: the deployment decision and a periodic review some months later. Between them lies the runtime — the long interval in which the system actually acts, binds the organisation, and drifts from its approved profile. Most frameworks govern the two moments and leave the interval between them effectively ungoverned. Because that interval is where authority is actually exercised, it is where the risk concentrates — and it is precisely what point-in-time governance misses.

Third, the runtime interval can be governed by two specific mechanisms. The first governs whether a delegation remains valid as conditions change: continuous admissibility, which monitors a system's operating conditions against the profile under which its authority was granted and requires reassessment when they diverge. The second governs how much authority is live across the estate at any moment: a portfolio-level autonomy budget, monitored continuously rather than reconciled annually. Together, they convert the ungoverned interval into a governed one.

This paper sets out the distinction between the moment and the interval, demonstrates why governing only the moment leaves the material risk unaddressed, and describes the two mechanisms that govern what happens between deployment and the next review. It closes with five questions that test whether an organisation governs the interval or merely the moment.

SECTION TWO

Two Moments, *One Interval.*

To see the problem clearly, separate two things that governance frameworks routinely conflate: the moment at which authority is granted, and the period over which it is exercised.

The grant of authority is an event. At an identifiable point — the deployment decision, which we can call T-zero — an organisation assesses an AI system and authorises it to operate. This event is discrete, documented, and, in most frameworks, well governed. A risk assessment is performed. An approval is recorded. A review date is set. Everything about the moment invites governance, because a moment is easy to govern: it has a date, an owner, and a decision.

The exercise of authority is not an event. It is a continuous process that begins at T-zero and runs until the system is decommissioned. Across that period, the system acts repeatedly, and with each action, it binds the organisation a little more. Nothing about this period is discrete. There is no single date, no single decision, no natural moment that invites a governance step. And so, in most frameworks, it receives none — until the next scheduled review, which is simply another moment, governing another instant and leaving the interval on either side of it untouched.

Figure 1 — Governance concentrates on deployment and on the next review. The runtime interval between them — where authority is actually exercised — is left ungoverned.

The figure makes the imbalance visible. Two thin moments are densely governed; the long interval between them, in which the system does all of its work essentially, is governed scarcely at all. This is not a marginal omission. The interval is not a gap *in* the framework — for most of an AI system's operating life, it *is* the framework's blind spot, the part of the system's life in which consequences actually accumulate.

SECTION THREE

Half a Governance *Framework.*

A framework that governs deployment and review, but not the interval between them, is not merely incomplete in a tidy, fill-in-the-gap sense. It is structurally mismatched to the risk it exists to manage.

Consider what point-in-time governance actually certifies. A deployment assessment certifies that, *as at the assessment date*, under the conditions then prevailing, the system was judged acceptable. That is a true and useful statement about a moment. It is not a statement about tomorrow, or about the ten thousand actions the system will take next quarter under conditions no one has yet examined. The certificate is accurate and, within days, no longer sufficient — because it speaks to a moment that has passed while the authority it granted continues to operate.

This is why the framework is not just incomplete but specifically the *wrong half*. If one had to choose between governing the deployment moment and governing the runtime interval, the interval is the more important of the two — because it is longer, because it is where conditions change, and because it is where the actions that bind the organisation are actually taken. A framework that governs only the moment has chosen the easier half and left the consequential half unattended. It is thorough about the part that carries little risk and silent about the part that carries most of it.

The deeper issue is that competitor frameworks treat governance as a *gate* rather than a *condition*. A gate is passed once: the system clears the assessment and proceeds. A condition must hold continuously: the system's authority is valid only while the conditions that justified it persist. Treating governance as a gate is what produces the blind spot. The system passes the gate at T-zero and is then, in governance terms, unsupervised until the next gate — no matter how far its operating reality has drifted from the basis on which it passed.

> *The interval is not a gap in the framework. For most of an AI system's operating life, the interval is the framework's blind spot.*

SECTION FOUR

What Changes *After Deployment.*

The case for governing the runtime interval rests on a simple observation: the conditions that made a delegation appropriate at deployment do not stay fixed. They drift — and the drift is rarely marked by any single, noticeable event.

Operating conditions shift. The volume of decisions a system handles grows. The mix of cases changes as the business changes. The financial exposure associated with each decision increases. The population the system affects widens. None of these is a malfunction; each is ordinary business evolution. But every one of them moves the system away from the profile under which its authority was granted — and a delegation calibrated to one profile is not automatically calibrated to another.

The system itself changes. Models are retrained. Prompts are revised. Upstream data sources are updated. Dependencies are swapped. Each change is individually reasonable and approved, yet the cumulative effect is that the system operating today is not, in any strict sense, the system assessed at deployment. The authorisation on file describes a predecessor.

The context evolves. The regulatory environment moves. The organisation's risk appetite is revised. Adjacent systems are deployed that interact with this one in ways no individual assessment anticipated. The system has not changed, and its conditions may not have changed — but the meaning of its authority has, because the world around the delegation is different.

Three things drift after deployment, and none of them announces itself. Set out together, they show why the transition from admissible to inadmissible is silent.

What drifts	How it shows up	Why no alarm sounds
Operating conditions	Decision volume grows, the case mix changes, financial exposure per decision rises, the affected population widens	None of it is a malfunction. Each shift is ordinary business evolution
The system itself	Models retrained, prompts revised, upstream data updated, dependencies swapped	Each change is individually reasonable and separately approved
The surrounding context	Regulation moves, risk appetite is revised, adjacent systems are deployed that interact with this one	The system has not changed at all. Only the meaning of its authority has

Table 1 — The three sources of drift, and why each passes unnoticed.

Read the third column and the governance problem is visible. There is no event to detect, because nothing discrete happens. The gap between the authorised profile and the operating reality simply widens until, at some unmarked point, it is too wide.

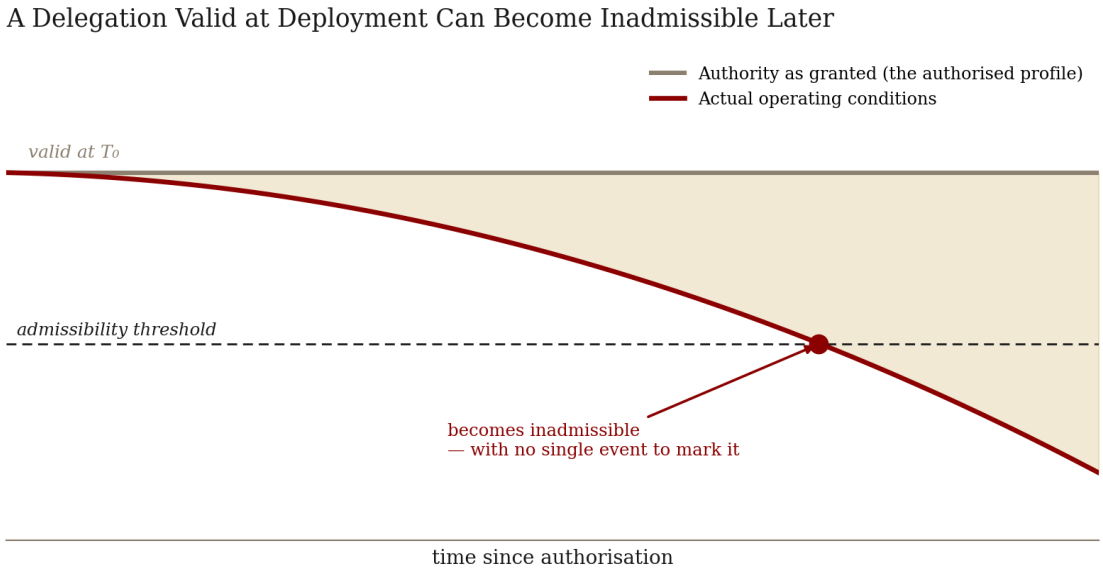

Figure 2 — The authorised profile holds constant while operating conditions drift. A delegation valid at deployment can cross into inadmissibility with no single event to mark the transition.

The crucial property, shown in the figure, is that the transition is silent. There is no alarm when a delegation crosses from admissible to inadmissible, because nothing discrete happens — the gap between the authorised profile and the operating reality simply widens until, at some unmarked point, the authority no longer fits what the system is doing. Point-in-time governance cannot detect a silent, continuous transition by definition. Only something that watches continuously can.

SECTION FIVE

Governing the Interval: *Admissibility.*

If the runtime interval is where authority drifts out of alignment, then governing the interval means continuously testing whether the authority still fits. This is the function of continuous admissibility.

The concept rests on a distinction between two questions that, in point-in-time governance, collapse into one. The first is *historical validity*: was this delegation correctly authorised when it was granted? The second is *current admissibility*: does this delegation remain appropriate now, given the current conditions? A deployment assessment answers the first question and is often mistaken for an answer to the second. But a delegation can be historically valid yet currently inadmissible — correctly authorised a year ago but no longer appropriate today.

Continuous admissibility (Hossain, 2026b) governs the second question as a live condition rather than a historical fact. It works by holding a record of the profile under which a system's authority was granted — the conditions, exposures, and limits that made the delegation appropriate — and monitoring the system's actual operating parameters against that profile on an ongoing basis. While the operating reality stays within the authorised profile, the delegation remains admissible. When it diverges materially, the delegation is flagged for reassessment, regardless of whether any scheduled review is due.

This inverts the logic of point-in-time governance. Under a gate model, authority is presumed valid until the next review disturbs that presumption. Under continuous admissibility, authority is valid only while it is demonstrably still fit, and divergence — not the calendar — is what triggers reassessment. The review date becomes a backstop rather than the primary control. The primary control is the continuous comparison between what was authorised and what is actually happening.

This is the mechanism that addresses the silent transition. Because admissibility is tested continuously against the authorised profile, the crossing point that no discrete event would announce is instead detected as a divergence — and the delegation is reassessed when it ceases to fit, not months later when a review happens to fall due.

SECTION SIX

Governing the Interval: *Aggregate Authority.*

Continuous admissibility governs whether an individual delegation still fits. A second runtime question concerns the whole: how much authority is live across the entire estate at any given moment — and whether that total is itself drifting.

Aggregate authority is not static, and it is not governed by approving each system in turn. Every new deployment adds to the total autonomous authority the organisation has delegated; every expansion of an existing system's scope adds to it further; and because each addition is individually modest and individually approved, the total can grow without any single decision that anyone would recognise as increasing the organisation's aggregate exposure. The estate's total authority drifts upward in the interval between the moments when anyone looks at the whole.

A portfolio-level autonomy budget (Hossain, 2026a) governs this. The organisation quantifies the decision authority each system holds, aggregates it into an enterprise total, and sets a board-approved ceiling on that total. Crucially for the runtime problem, the budget is monitored continuously rather than reconciled at review points: the live total is maintained as systems are deployed, expanded, and retired, so that the organisation can see at any moment how much of its authorised ceiling is committed — and is alerted as the total approaches the ceiling, rather than discovering an overshoot at the next annual reconciliation.

The two mechanisms are complementary because they govern different forms of runtime drift. Continuous admissibility governs the drift of an individual delegation away from its own authorised profile. The autonomy budget governs the aggregate's drift away from the board's approved ceiling. One watches each system; the other watches the whole. A framework that governs the interval needs both because authority can become inappropriate either when an individual system outgrows its grant or when the estate as a whole outgrows its mandate — and these are different failures requiring different instruments.

SECTION SEVEN

What Runtime *Governance Requires.*

Governing the interval is not a matter of reviewing more often. Reviewing more often simply adds more moments; it does not change the fact that the intervals between them remain ungoverned. Runtime governance is a different kind of control, with three operational requirements.

It requires a recorded authorised profile, not merely an approval. A point-in-time framework records that a system was approved. Runtime governance requires recording the profile under which it was approved — the conditions, exposures, and limits that the approval assumed — because that profile is the baseline against which ongoing operation is compared. Without a recorded profile, there is nothing to detect drift against; an approval alone gives a date, not a baseline.

It requires continuous comparison, not periodic inspection. The defining feature of runtime governance is that the comparison between the authorised profile and actual operation runs continuously, in the interval, rather than being performed afresh at each review. This is the requirement that most distinguishes it from point-in-time assurance, and the one that periodic methods — however frequent — cannot satisfy, because a process that runs at intervals cannot detect what happens between them.

It requires divergence to trigger action, not the calendar. In a runtime model, the event that prompts reassessment is a material divergence between authorised profile and operating reality — wherever in the interval it occurs — not the arrival of a scheduled review date. The calendar becomes a backstop. Divergence becomes the trigger. This is what allows governance to act when the risk actually changes, rather than when the diary says to look.

Taken together, these requirements describe a framework that treats governance as a condition that must hold continuously rather than a gate that is passed once. That is the whole of the runtime authority argument: a delegation's authority is not a possession granted at deployment and retained until revoked. It is a status that must remain continuously earned, against conditions that will not hold still.

SECTION EIGHT

Five Questions *for Your Board.*

The test of whether a framework governs the interval or merely the moment is not the rigour of its deployment assessment. It is whether anything governs the system between assessments. Each of the following five questions probes that distinction, and each should be answerable for any AI system in operation today.

1. **Do you govern the moment or the interval?** Once a system is deployed, what governs it before the next scheduled review — or is the interval between assessments effectively unsupervised?
2. **Did you record a profile, or just an approval?** For each system, did you record the conditions and limits under which its authority was granted — the baseline against which drift could be detected — or only the fact of approval?
3. **Would you detect a silent transition?** If a system's operating conditions drifted past the point where its authority remained appropriate, with no single event to mark it, would anything detect the crossing before the next review?
4. **Is your aggregate authority monitored or reconciled?** Do you know, continuously, how much autonomous authority is live across the estate and how close it sits to the board's ceiling — or is the total only reconciled periodically?
5. **Does divergence trigger reassessment, or only the calendar?** When a delegation ceases to fit current conditions, is it reassessed when the divergence is detected — or only when a scheduled review falls due?

> *A framework that governs only the moment is thorough about the part that carries little risk and silent about the part that carries most of it.*

ABOUT

42 Consulting.AI.

42 Consulting.AI builds constitutional AI governance for regulated enterprises. Its core work, the MANDATE Suite, is a comprehensive governance architecture comprising 64 normative instruments across five frameworks — governing the delegation of machine decision authority, the ethical and rights boundaries on its use, the management system that maintains it, and the regulatory extensions for EU and group structures. The Suite is designed for organisations in financial services, insurance, superannuation, healthcare, government, and other regulated sectors operating autonomous AI systems at scale.

The architecture is supported by an academic preprint series published on Zenodo, which establishes the theoretical basis for each governance mechanism, and by the forthcoming book *Governance-First AI*. The Suite's distinguishing conviction is that AI governance at the depth regulated enterprises require cannot be delivered by a compliance framework alone — it requires a constitutional architecture, adopted by the board, enforced by the infrastructure, and continuously tested against current conditions.

Dr M Maruf Hossain, PhD, GAICD · Chief AI Strategist
enquire@42consulting.ai · www.42consulting.ai

The preprint series is available at: zenodo.org/communities/mandate/records

REFERENCES

References *and Sources.*

The governance mechanisms described in this paper are set out in full in the MANDATE preprint series, published through Zenodo. The all-versions digital object identifiers below resolve to the current version of each paper.

REGULATORY CONTEXT

APRA (2026). *Letter to Industry on Artificial Intelligence (AI).* Australian Prudential Regulation Authority, 30 April 2026. Cited for its observation that point-in-time assurance methods are ill-suited to models that learn, adapt and degrade over time.

MANDATE PREPRINT SERIES

Hossain, M. M. (2026a). *The Autonomy Budget: A Portfolio-Level Framework for Governing Delegated Machine Authority in Regulated Enterprises.* Zenodo. https://doi.org/10.5281/zenodo.20480491

Hossain, M. M. (2026b). *Continuous Admissibility: A Temporal Governance Framework for Delegated Machine Authority in Regulated Enterprises.* Zenodo. https://doi.org/10.5281/zenodo.20580716

Hossain, M. M. (2026c). *Write-Before-Execute: A Governance Standard for Evidentially Defensible AI Decision Logging in Regulated Enterprises.* Zenodo. https://doi.org/10.5281/zenodo.20603712

Consulting.AI

Scaling Intelligence

enquire@42consulting.ai · www.42consulting.ai