

GOVERNED AI FOR INSTITUTIONAL CAPITAL

# Living Policy Architecture

A framework for institutions whose rules are written in prose but whose decisions must be executed, monitored, and defended in real time. On closing the gap between what an institution has committed to and what it actually does.

---

**Brendan Sibeth**, Founder & Managing Partner · v1.1 · 2026

## ABSTRACT

Every institution that manages capital operates against a body of rules: investment policy statements, mandates, risk limits, and the regulation that governs all of it. Those rules live in documents. The decisions they are meant to govern happen somewhere else, interpreted under time pressure by people who cannot hold every clause in working memory. The distance between the written rule and the executed action is where breaches, restatements, failed examinations, and the entire cost of manual oversight accumulate. **Living Policy Architecture** is the discipline of compiling written policy into executable, continuously evaluated logic that decides what it is entitled to decide, escalates what requires human judgment, adapts when a rule changes, and proves every outcome back to its source. The parts of this idea have a long lineage. What is new is the binding of them into a single discipline, aimed at the whole body of an institution's written policy rather than only the fragments that were ever easy to encode. This paper sets out what it is, the operational framework behind it, and why it belongs in the core of an institutional stack rather than at its periphery.

## 01

## The gap between the rule and the act

Consider how policy actually lives inside a pension plan, an asset manager, or an outsourced chief investment office. A mandate is negotiated and signed. An investment policy statement is approved by a board. A regulator publishes an obligation. Each of these is a document, written in careful prose, and each is then handed to people who are expected to honour it across thousands of subsequent decisions. The rule is static and textual. The activity it governs is continuous and numerical. Nothing automatically connects the two.

So the connection is made by hand. An analyst rereads the mandate before an allocation. A compliance officer reconciles, after the fact, whether what happened matched what was promised. A risk team assembles evidence for an examination by reconstructing decisions that were made months earlier under assumptions nobody wrote down. This is slow, expensive, and fragile, and it fails in a specific and costly way: the institution discovers that it breached a rule only after the breach has already happened, in a review, where the cheapest moment to have caught it has long passed.

The cost is not unknown. Compliance budgets are measured closely, and the headcount devoted to oversight is visible on every organisation chart. What is rarely done is to pull that cost apart and isolate the portion that exists only because policy sits in a form that cannot act. Separated out that way, a figure that looked like the fixed price of being regulated turns out to be, in large part, the price of a translation gap: the latency between a decision and its verification, the remediation that follows findings, and the reputational weight an institution carries each time it must explain, after the fact, why something was allowed to occur. The gap is treated as a fixed cost of doing business. It is not. It is an artefact of leaving policy in a form that cannot act.

*The rule is static and textual. The activity it governs is continuous and numerical. Nothing automatically connects the two, so the connection is made by hand.*

02

## What Living Policy Architecture is

Living Policy Architecture is the practice of treating policy as something that runs rather than something that is merely consulted. It begins by reading the rules as written, in the formats they already exist in, and compiling them into structured, executable logic. That logic then evaluates real activity against the actual policy, continuously, at the moment decisions are made rather than in a review weeks later. The word that matters most in the name is the first one.

### DEFINITION

**Living Policy Architecture** is policy compiled into executable logic that evaluates activity in real time, distinguishes the decisions it may take from the decisions it must escalate, regenerates its logic the same business day its governing rule changes as a versioned candidate the institution validates before it goes live, and writes an auditable record of every decision back to the source rule that produced it.

Three properties make it *living* rather than merely automated. First, it is **continuous**: it does not wait to be asked, it watches activity against policy as that activity occurs. Second, it is **adaptive**: when a mandate is amended or a regulation changes, the executable logic is regenerated from the new text the same business day, as a versioned candidate for validation, rather than waiting for a quarterly engineering cycle to catch up. Third, it is **accountable**: every evaluation it performs is recorded with the rule it applied, the data it weighed, the confidence it held, and the outcome it reached, so that the answer to the question every institution dreads, can you show why this happened, is a query rather than a forensic project.

It is worth being precise about what Living Policy Architecture is not. It is not a dashboard that reports breaches after they occur. It is not robotic process automation that repeats a fixed sequence regardless of whether the underlying rule still holds. It is not a large model asked to reinterpret a policy from scratch on every decision. It is a compiler and a runtime: the policy becomes governed, versioned code, and that code is what executes.

It is also worth being honest about where this comes from. Turning a written rule into logic that executes is not a new ambition. It runs from the early work on rendering statute as a logic program, the effort in the 1980s to express the British Nationality Act in executable form among the best known of it, through decades of business rules engines and decision models that let institutions hold their logic as governed, versioned configuration rather than as code buried in an application. What held the idea back was the cost and fragility of the first step, the compile, which depended on scarce specialists hand-encoding rules that grew brittle as they multiplied. What has changed is the economics of that one step. Reading structure out of prose is no longer the bottleneck it was. The discipline now is to use that capability without inheriting the old brittleness, which is why the logic that runs is deterministic and versioned, the genuinely ambiguous cases are returned to people rather than guessed, and the translation from text to logic is itself recorded rather than trusted.

03

# The operational framework

Living Policy Architecture operates as five stages over a single governed spine. Each stage is deliberately bounded, because the value of the system rests on every step being inspectable rather than on any step being clever.

---

## 1 Ingestion of the rule as written

Policy statements, mandates, regulatory text, and the institution's own precedent enter in their existing form. The system structures them and, critically, separates what can be compiled faithfully from what cannot. Some clauses are precise enough to translate without loss. Others carry the kind of open-textured judgment that resists any translation that would not quietly change their meaning. The first work of ingestion is to tell these apart, to compile what can be compiled, and to surface the rest to the policy owner for an explicit decision rather than guessing silently.

---

## 2 Compilation into executable logic

The structured rule becomes deterministic configuration: a set of conditions, thresholds, and required actions that can be evaluated against live activity. This is the step that converts prose into something that can run the same way every time it is invoked.

---

## 3 Continuous evaluation against activity

The compiled policy is applied to real decisions as they occur. In-policy actions that fall clearly within the rule proceed on their own. Anything that approaches a boundary, or that requires genuine judgment, is surfaced to the right person with the relevant clause and reasoning already assembled.

---

## 4 Attestation to the source

Every evaluation is written to a governed record: which rule was applied, which data informed it, what confidence accompanied it, what was decided, and what resulted. The record is designed to be produced on demand, in a form an examiner or a board can follow without translation.

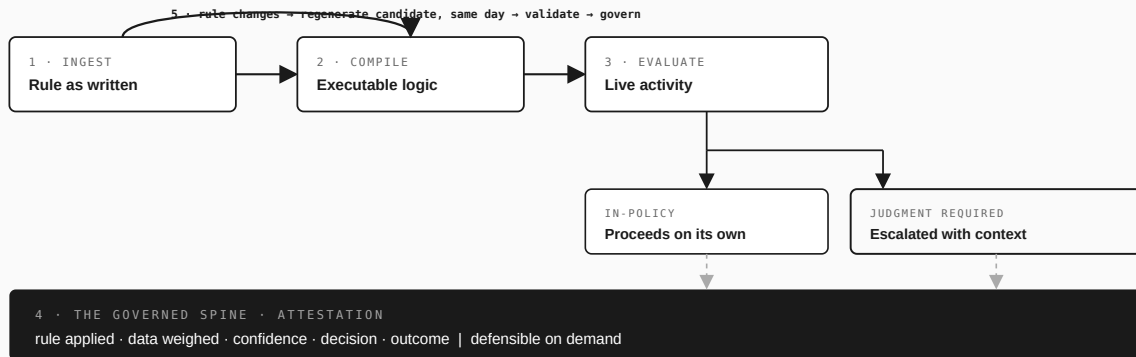
---

## 5 Regeneration on change

When the governing text changes, the executable logic is regenerated from the new version as a versioned, dated candidate. Generation can happen the same business day. What governs live decisions still passes the institution's own validation first, deliberately gated, so that the policy which runs is provably the policy in force, and the rules are never hot-swapped underneath a decision without review.

---

## EXHIBIT 1 · THE OPERATING MODEL

**From written rule to defended outcome**

**Exhibit 1.** Written policy is compiled into executable logic and evaluated against live activity. In-policy actions proceed; judgment calls escalate with their context assembled. Every evaluation writes to the governed spine, and a change in the governing rule regenerates the logic the same business day as a versioned candidate, validated before it governs.

04

## Why this belongs in the core of the stack

An institution's technology stack already contains systems that touch policy at the edges. There are governance and compliance tools that catalogue obligations. There is process automation that executes routine steps. There are reporting layers that summarise what happened. Each is useful, and none of them closes the gap described in the opening of this paper, because each sits beside the decision rather than inside it. Living Policy Architecture is positioned differently, and the difference is the entire argument.

### It is the layer where written commitment meets executed action

A compliance catalogue knows the rule exists. A reporting layer knows what was done. A policy layer that compiles the rule into runtime logic stands at the exact point where the two meet, evaluating the action against the commitment as the action occurs. That position is structurally load-bearing, and it is worth being precise about it. The institutions reading this already know one form of control that lives at the point of action: the real-time pre-trade compliance check that stops an order before it breaches a quantitative limit. That control is genuine, and it is also narrow. It governs what can be reduced to a number on a tradeable instrument, its rules are authored by hand, and changing them is a project. The argument here is for the same position in the decision, generalised: not only the constraints that were ever easy to encode, but the full body of written commitment, compiled from the text itself and regenerated when the text changes. A system that sits beside the decision can describe a breach after the fact. A system inside it can stop one. Pre-trade compliance proved that narrow case years ago. Living Policy Architecture extends it to the rest of what an institution has committed to.

### It converts oversight from a cost centre into an asset

When policy runs, the manual reconciliation layer shrinks, because the routine checks that consume expert time are absorbed by deterministic evaluation. The people who were rereading documents move to the judgment calls that actually warrant their attention. The institution does not merely spend less on oversight; it redeploys its most experienced people toward the decisions where experience compounds.

### It makes the institution's own commitments portable and durable

The judgment embedded in how an institution interprets and applies its policy is, today, tacit. It lives in the heads of senior people and walks out of the building when they leave. Compiling policy into governed logic externalises that judgment into a form the institution owns, versions, and carries forward, independent of any individual's tenure. This is the asset that does not depreciate.

| CONCERN                   | ADJACENT TOOLS (GRC, RPA, REPORTING) | LIVING POLICY ARCHITECTURE                                |
|---------------------------|--------------------------------------|-----------------------------------------------------------|
| <b>Position</b>           | Beside the decision                  | Inside the decision, at the point of action               |
| <b>Timing</b>             | After the fact, in review            | Continuous, at the moment it matters                      |
| <b>Response to change</b> | Manual reconfiguration cycle         | Same-day regeneration of a candidate, governed activation |
| <b>Evidence</b>           | Reconstructed for an examination     | Generated as a byproduct of operating                     |
| <b>Knowledge</b>          | Stays with the people                | Externalised, versioned, owned by the institution         |

The interest in compiling rules into logic is not ours alone, and a growing field is a sign the direction is real rather than a reason to claim it first. What distinguishes the position argued here is less the act of compiling than what surrounds it: logic the institution owns and can carry with it, evaluation at the point of the decision itself, and regeneration bound to a record that proves what ran and why.

Read across that table and the conclusion is not that adjacent tools are wrong, but that they were never designed to occupy the position that matters most. Living Policy Architecture is not another layer on the periphery. It is the connective core that the periphery has always implied and never supplied.

05

## Why Dartmouth Advisory Partners

A discipline this consequential should not rest on a single vendor's marketing, and we do not ask it to. What an institution should look for in a partner for Living Policy Architecture is a specific and demanding set of commitments, and we hold ourselves to them in the open.

The first is that the system is **built to be owned, not rented**. An institution's policy is among its most sensitive assets, and the logic compiled from it should belong to the institution, documented and transferable, not held hostage inside a black box. Our engagements are designed to end with the institution in possession of a system it understands and controls.

The second is that **provenance is the foundation, not a feature added late**. We treat the auditable record as the substrate of the system rather than a report bolted on at the end, because in regulated capital a decision that cannot be defended is a decision that cannot responsibly be made. Every output is built to be traced to the rule that produced it.

The third is **intellectual honesty about the boundary between what runs on its own and what requires a person**. We do not sell autonomy as an end in itself. In institutional finance the value was never that a machine decides unsupervised; it is that the routine is handled reliably, the consequential is escalated with its full context, and the line between the two is explicit, inspectable, and set by the institution. Drawing that line is the easy part. Making it hold is the hard part, and the failure modes are well understood. Automate the routine and the people left to judge the exceptions are the ones who now see them least often and are least practiced at them. Assemble the context for a reviewer and you risk leading them to the answer the system already reached. We treat these as problems to design against rather than risks to mention in passing, because an escalation channel that deskills its reviewers or nudges them into agreement is not oversight. It only looks like it.

We build governed AI for institutions that have to defend their decisions, and Living Policy Architecture is the centre of that work. The firm exists precisely because the gap between written policy and executed action is too important to leave to manual effort and too sensitive to hand to systems that cannot explain themselves.

## On the limits of this thesis

*Stated plainly, because a discipline built on provenance owes the same honesty about itself.*

### Compilation depends on the clarity of the rule.

There is a computable part of any policy and a remainder that resists computation, what philosophers of law have called the open texture of language, and policy that is genuinely open-textured cannot be compiled into something less ambiguous than itself. The discipline is in knowing which is which. The system surfaces the remainder for an owner to resolve; it does not pretend the ambiguity away.

### The hardest cases are meant to escalate.

Living Policy Architecture is not a promise to automate judgment. Its design goal is the opposite: to handle the routine cleanly so that human attention concentrates where it is genuinely required.

### Adaptivity is governed, not automatic trust.

Regeneration on a rule change is versioned and reviewable. The point is that the policy in force and the policy that runs stay provably aligned, with the change itself on the record.

### The system is itself something to be governed.

A discipline that compiles policy with the help of models cannot exempt those models from the scrutiny it asks of everything else, and it cannot lean on official guidance that has not yet caught up. The revised interagency guidance on model risk issued in April 2026 sets both deterministic rule-based processes and the newer forms of generative and agentic AI outside the formal model perimeter, the first because it is not a statistical model, the second because the regulators judged it too novel to scope. A system like this one lives in both of those gaps at once. So it governs itself: the compile is validated, the translation from text to logic carries its own provenance, and its outputs are open to the same challenge as any model an institution runs. Provenance that stopped at the boundary of our own work would not deserve the name.

---

**If the distance between your policy and your decisions is a cost you have been treating as fixed, it is the conversation we want to have.** Dartmouth Advisory Partners builds Living Policy Architecture for institutions that must defend every decision they make.

bsibeth@dap.solutions · dap.solutions · Toronto

#### NOTES

The lineage in Section 02 includes the formalisation of statute as a logic program (Sergot and colleagues, *The British Nationality Act as a Logic Program*, Communications of the ACM, 1986) and the longer tradition of business rules engines and decision models. The open texture of language in Section 05 is H. L. A. Hart's, from *The Concept of Law*, later sharpened for computation in work on the limits of computable contracts. The discussion of oversight failure modes in Section 05 draws on Lisanne Bainbridge's *Ironies of Automation* (Automatica, 1983). The model-risk guidance referenced in Section 05 is the *Revised Guidance on Model Risk Management* issued by the Federal Reserve, OCC, and FDIC on April 17, 2026 (Federal Reserve SR 26-2; OCC Bulletin 2026-13).