

From Policy to Pipeline: Operationalizing Automated Data and AI Governance

How public-sector organizations can turn governance policy into executable, auditable controls across the data and AI lifecycle.

By Gopichand Mannava, M.Eng., MBA, PMP | Chief Data Architect, State of Connecticut | Independent Researcher

The Governance Gap Between Policy and Practice

Every mature data organization I have worked with has a governance policy. Far fewer have governance that *runs*. The gap between the two is where trust in artificial intelligence quietly erodes.

DAMA International places data governance at the center of the DMBOK wheel for a reason — every other knowledge area, from Data Architecture to Data Quality to Metadata, depends on it¹. Yet in practice, governance too often lives in policy documents, committee minutes, and quarterly attestations. Meanwhile, data pipelines run every hour, models retrain overnight, and generative AI features ship on two-week sprints. The cadence mismatch is the problem.

Trust in AI is built upstream through disciplined data lifecycles and federated governance operating models. The practical next step is to translate governance policy into automated, executable controls that keep pace with the systems they are meant to govern. In public-sector environments, where accountability, transparency, cybersecurity, and public trust are central obligations, this question is not academic. Auditors, oversight bodies, program leaders, and residents increasingly expect an answer that can be demonstrated in production—not reconstructed after an incident or audit request.

Why Manual Governance Cannot Scale to AI

Three structural forces make traditional, document-driven governance unsustainable in an AI-era data estate:

- **Volume and velocity.** A single mid-sized agency easily runs thousands of pipelines, hundreds of datasets, and dozens of models. No governance council can review each one at the speed they change.
- **Composability.** Modern AI systems chain retrieval, embeddings, foundation models, and downstream applications. Risk emerges at the seams between components, not inside any single one, and manual reviews rarely see the whole graph.
- **Regulatory concurrency.** The NIST AI Risk Management Framework², the EU AI Act³, sector guidance from the FDA and financial regulators, and — in Connecticut — Public Act 26-15 on artificial intelligence

¹DAMA International, DAMA-DMBOK: Data Management Body of Knowledge, 2nd Edition, <https://dama.org>

²National Institute of Standards and Technology, AI Risk Management Framework (AI RMF 1.0), January 2023, <https://www.nist.gov/itl/ai-risk-management-framework>

³European Parliament, Regulation (EU) 2024/1689 (EU AI Act), Official Journal of the European Union, June 13, 2024, <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>

and automated decision-making⁴ each impose overlapping but non-identical control expectations. Reconciling them by hand across a live portfolio is not a defensible operating model.

Automation is not a substitute for governance. Automation is what turns governance from an aspiration into a control environment.

Manual governance runs on policy time. Systems run on machine time.

Annualized cadence of governance activities vs. the systems they are meant to govern.

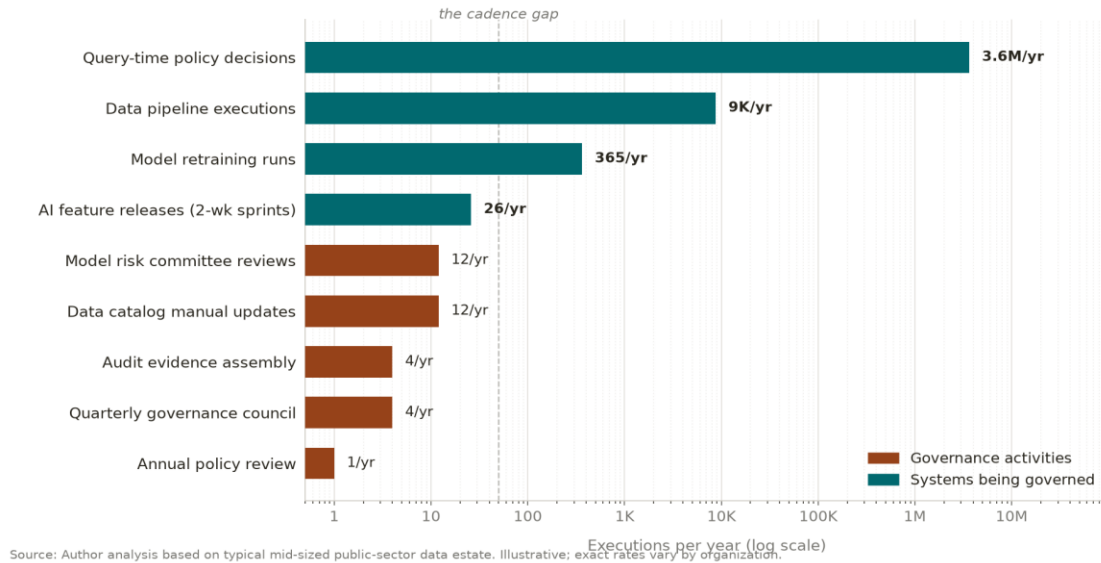


Figure 1. The cadence gap. Governance activities are measured in months and quarters; the systems they govern are measured in seconds and minutes.

A Working Definition of Automated Governance

I use a deliberately narrow definition: **automated governance is the practice of expressing data and AI policies as code, executing them against live systems, and producing audit-ready evidence as a byproduct of normal operations.**

That definition has three practical implications. Policies must be machine-readable, not only human-readable. Controls must run continuously, not on request. And evidence must be generated automatically, not assembled retroactively for the next audit cycle.

Nothing here is exotic technology. What is new is the discipline of treating governance as a software product — with versioning, tests, deployment pipelines, and ownership — rather than as a set of PDFs on a SharePoint site.

Five Governance Controls That Belong in the Pipeline

Not every DMBOK activity should be automated, and not every automation is worth the engineering cost. Five controls consistently offer a practical starting point because they create visible, testable signals of whether governance is operating as intended. Each maps cleanly to a data-governance knowledge area.

⁴State of Connecticut Public Act 26-15, An Act Concerning Artificial Intelligence and Automated Decision-Making, May 27, 2026, <https://www.cga.ct.gov/>

Control	DMBOK Knowledge Area	What Automation Looks Like in Production
Metadata capture	Metadata Management	Every pipeline emits schema, ownership, classification, and lineage on run — no manual catalog entry required.
Policy-as-code	Data Governance	Access, retention, and use-case restrictions expressed as executable rules and enforced at query time and model-serving time.
Continuous data quality	Data Quality	Quality rules attached to datasets as contracts; failures block downstream consumption and open incidents automatically.
Model risk registration	Data Governance + Data Security	Every deployed model is registered with its intended use, training data lineage, evaluation metrics, and risk tier before it can serve production traffic.
Evidence generation	Data Governance (cross-cutting)	Control executions, exceptions, approvals, and remediations are logged in an immutable evidence store aligned to NIST AI RMF and applicable statutes.

Five automated controls, mapped across the DMBOK

Each control anchors in one knowledge area (★) and reinforces several others (●, •).



Figure 2. Each automated control anchors in one DMBOK knowledge area and reinforces several others — which is why automation is a governance strategy, not a tooling decision.

Metadata as a Byproduct, Not a Chore

The most common reason data catalogs fail is that keeping them current is somebody’s side job. The fix is not more discipline; it is emitting metadata from the systems that already know it. When ingestion frameworks,

transformation tools, and BI platforms are instrumented to publish schema, ownership, sensitivity classification, and lineage on every run, the catalog stops being a manual artifact and starts being a mirror of reality.

This matters for AI because model behavior cannot be explained without knowing where training and inference data came from. Lineage is not a nice-to-have; it is a prerequisite for explainability, incident investigation, and accountable oversight.

Policy-as-Code, Not Policy-as-PDF

Access policies, retention schedules, cross-border restrictions, and permitted-use rules are all decidable questions. They belong in a policy engine that data platforms and AI applications call at runtime — not in a document that assumes humans will apply it correctly under deadline pressure.

The operating model shift is significant. Governance stops writing policies and starts owning policy source code, tests, and release pipelines, in partnership with platform engineering. Reviews focus on the code, the test coverage, and the change history — evidence that auditors can inspect directly.

Data Contracts and Continuous Quality

Data quality rules expressed only in dashboards create observers. Data quality rules expressed as contracts between producers and consumers create accountability. When the contract fails — a null-rate breach, a schema drift, a distribution shift — downstream consumption pauses, an incident opens, and the producing team owns the remediation. Quality moves from a monthly report into an operational discipline.

A Model Registry That Is Actually Enforced

A model registry that lists models is a spreadsheet. A model registry that is enforced — where the serving infrastructure refuses to route traffic to unregistered or expired models — is a control. The registration record captures the model's intended use, training data lineage, evaluation metrics, human-oversight design, and risk tier. Risk tiering is what allows the same framework to handle both a low-risk internal summarization tool and a high-impact benefits-eligibility model, with proportionate scrutiny for each.

Evidence as a Byproduct of Operations

The final control is the one that most changes how governance is perceived by auditors and executives: every policy check, every exception, every approval, and every remediation writes to an immutable evidence store, tagged to the applicable framework — NIST AI RMF functions, EU AI Act obligations, or state-specific requirements. When the audit request arrives, the answer is a query, not a scramble.

The Operating Model: Federated, Not Centralized

Automated governance succeeds when it is federated with clear central guardrails, echoing the data-product operating model that Zhamak Dehghani has articulated for data mesh⁵ and that Trust Continuum Part 4 will

⁵Zhamak Dehghani, Data Mesh: Delivering Data-Driven Value at Scale, O'Reilly Media, 2022.

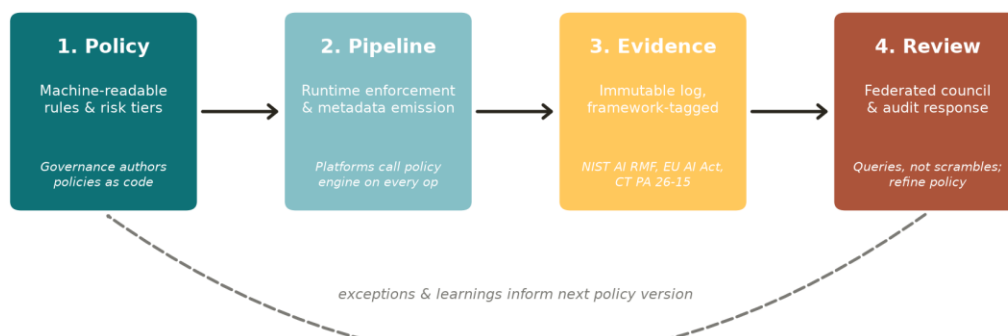
explore in depth. In my experience, three responsibilities cannot be delegated to individual domains, and everything else can:

- **Central:** the policy catalog itself, the risk-tier taxonomy, the evidence schema, and the identity of the systems of record for each. These are the invariants.
- **Central:** the shared platform capabilities — metadata emission libraries, policy engine, model registry, evidence store — offered as products with documented contracts.
- **Central:** the escalation path for policy exceptions, disputes between domains, and material control failures.

Everything else — which datasets get catalogued first, which quality rules matter most, which models to prioritize for tier assessment — should live with the domain teams closest to the data. This is a federated operating model expressed in engineering practices rather than only policy structure.

The automated governance loop: policy becomes code, evidence closes the loop

Every stage produces auditable output; the review stage feeds refinements back into policy.



Framework synthesis by the author; aligned to DAMA-DMBOK Data Governance function.

Figure 3. The automated governance loop. Policy becomes executable code, execution generates evidence, and evidence feeds the next policy version.

A Note From the Public Sector

Public-sector environments intensify the case for automation for three reasons. First, our data touches residents whose choices are limited — a citizen cannot switch away from their state government the way a customer can switch away from a bank. That raises the bar for evidence of responsible use. Second, our workforce cannot grow to match our data. Automation is how a governance team of ten oversees a portfolio built by thousands. Third, transparency obligations are stronger. When the public can request the record, the record has to exist — completely, contemporaneously, and without heroic effort.

Across jurisdictions, public agencies face growing expectations for responsible AI, transparency, security, and accountable decision-making. Agencies that industrialize their governance evidence will be better prepared to adopt AI responsibly at scale. The alternative is a false choice between speed and defensibility—one public institutions should not impose on residents.

Where to Start on Monday Morning

Automated governance is not a program to launch; it is a set of disciplines to introduce, one control at a time. If you are early in this journey, three practical starting moves have consistently paid back:

- **Pick one high-value dataset and instrument it end-to-end.** Emit lineage and quality signals from the pipeline that produces it, register the consuming model or report, and route evidence into a store you can query. One worked example is worth ten reference architectures.
- **Convert one policy from prose to code.** Choose a policy that is currently enforced by memory and diligence — a retention rule or an access restriction — and express it in a policy engine. Measure how often it fires. The number will surprise you.
- **Publish a risk-tiering rubric and apply it to your current AI portfolio.** Even a simple tiering scheme separates the systems that need heavy oversight from the ones that need lightweight oversight, and immediately makes your governance capacity go further.

Each of these moves is small enough to complete in a quarter and concrete enough to demonstrate to leadership, auditors, and delivery teams. The most durable governance programs are not designed only on a whiteboard; they are composed from a series of controls that each earned their place through operational use.

Closing the Loop

The DAMA-DMBOK framework has always insisted that governance is the hub, not a spoke. Automated governance is what makes that assertion operational in an AI-era data estate. It closes the loop between the policies we write and the systems we run — turning trust from a claim into a control, and turning the Trust Continuum from a series of concepts into a system that residents, regulators, and colleagues can rely on every day.

DAMA Northeast is one of the few communities positioned to lead this conversation across sectors. I look forward to continuing it with the chapter.

About the author

Gopichand Mannava, M.Eng., MBA, PMP, is Chief Data Architect for the State of Connecticut and an independent researcher focused on enterprise data architecture, responsible AI governance, cybersecurity, analytics, and public-sector digital transformation. His work examines how public institutions can build secure, trustworthy, and accountable data and AI capabilities at enterprise scale. The views expressed are his own and do not necessarily represent those of the State of Connecticut.