Turning Policy Documents into Evidence: Building Measurement Infrastructure for Operational AI Governance

Jaya Kandaswamy
President
The Institute for AI Measurement Science
Southlake, TX, USA
jaya.kandaswamy@iaimscience.org

Sowmya Podila
Technical Advisor
The Institute for AI Measurement Science
Southlake, TX, USA
https://orcid.org/0009-0007-4346-7750

***Abstract*—Artificial Intelligence (AI) governance frameworks such as the European Union (EU) Artificial Intelligence Act [1], the NIST AI Risk Management Framework (AI RMF) [2], and ISO/IEC 42001 [3] establish formal obligations for documentation, risk management, transparency, and monitoring. However, these frameworks are predominantly prose-based and non-executable, requiring organizations to manually interpret and operationalize requirements. Recent research on Policy-to-Tests (P2T) transformation [4] demonstrates that natural language policies can be converted into structured, atomic rules suitable for automated validation. In parallel, NIST's Open Security Controls Assessment Language (OSCAL) [5] demonstrates that compliance artifacts can be represented in machine-readable formats. This paper presents a reusable multi-agent governance measurement architecture - comprising a Standards Monitoring Agent, a Control Generation Agent, and an Evidence Collection Agent orchestrated by a modern stateful agentic framework. A multi-step agentic pipeline powered by Large Language Models (LLMs) converts governance prose into versioned, cross-framework evidence artifacts with mandatory human-in-the-loop (HITL) review at each stage. By separating evidence generation from compliance judgment, the architecture enables organizations to produce consistent, audit-ready governance evidence aligned with emerging regulatory requirements.**

***Index Terms*—AI governance; compliance-as-code; OSCAL; policy transformation; AI risk management; agentic architecture; executable regulation**

I. Introduction

The rapid deployment of autonomous and semi-autonomous AI agents has created a measurement gap across industries. While organizations increasingly demonstrate business value from AI systems, they often lack standardized, auditable methods to verify that these systems behave as intended, remain within authorized scope, and satisfy emerging regulatory requirements. Over the past several years, governments and standards bodies have produced extensive AI governance frameworks. While some frameworks, such as NIST AI Risk Management Framework (AI RMF), include measurement-oriented functions, most fall short of prescribing interoperable, operational mechanisms for generating verifiable evidence.

Building on P2T transformation [4] and NIST's OSCAL [5], which demonstrate respectively that policies can be converted to machine-readable rules and that compliance artifacts can be standardized, what remains missing is interoperable, audit-grade AI governance evidence artifacts reusable across audits, organizations, and time. This paper presents a reference governance measurement architecture for executable AI governance, independent of any specific institutional implementation.

Managerial relevance. Engineering teams, compliance officers, and technology leaders must translate broad policy requirements into implementation practices and evidence artifacts that can withstand internal audits and external scrutiny. This paper introduces a measurement-oriented architecture that converts governance policies into reusable control objects and standardized evidence artifacts collected automatically across engineering environments, reducing manual compliance work, improving traceability, and supporting practitioners implementing trustworthy AI systems and regulators auditing implementations at scale. The approach provides a path toward governance automation comparable to infrastructure-as-code in DevSecOps (Development, Security, and Operations) environments.

II. Background

The EU AI Act becomes generally applicable in 2026 and requires demonstrable compliance for high-risk systems [1]. Existing frameworks such as the NIST AI RMF [2] and ISO/IEC 42001 [3] define governance expectations but do not prescribe interoperable implementation artifacts. Organizations must therefore translate overlapping obligations into auditable engineering evidence. Two gaps emerge in this process: the first between regulatory intent and organizational implementation, and the second between implementation and regulatory confirmation. Without structured, traceable artifacts, inconsistencies can compound across both translations.

Academic research has begun examining technical approaches for operationalizing AI governance and assurance. Prior work on algorithmic auditing highlights the need for verifiable evidence linking governance policies to system behavior [7], [8]. Other research has explored structured evaluation

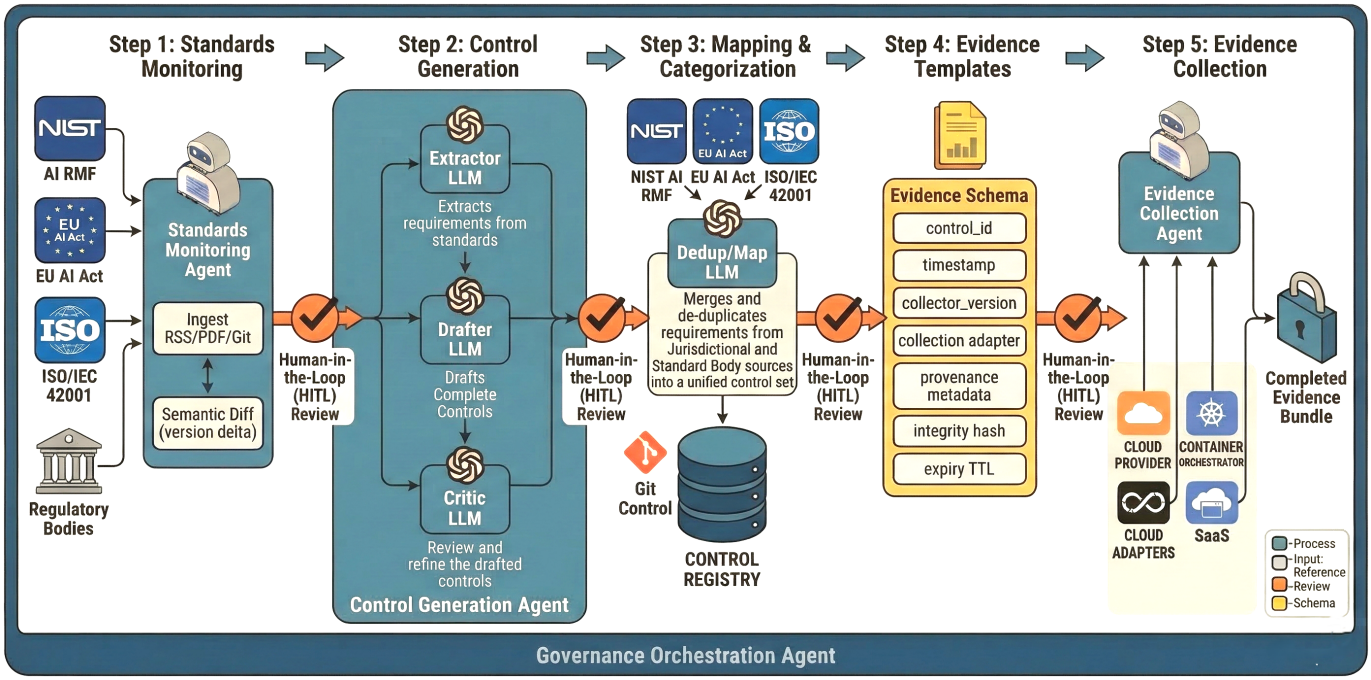

Fig. 1. Proposed AI governance measurement architecture showing the five-step governance orchestration pipeline including standards monitoring, control generation, control mapping, evidence schema generation, and automated evidence collection.

benchmarks and compliance testing environments for assessing whether AI systems meet safety and policy requirements [9]. Empirical studies of EU AI Act compliance identify data governance and transparency documentation as the most operationally challenging requirements, with no standardized tooling to generate the required evidence [10]. Standards efforts such as IEEE P2863, which defines governance criteria and performance auditing processes for organizational AI [11], confirm that the research community recognizes AI governance demands operational infrastructure, not only policy principles. However, most existing work focuses on behavioral evaluation or auditing methodologies rather than the generation of reusable, cross-framework governance evidence artifacts that can support audit workflows and compliance verification across organizations and regulatory frameworks.

The broader compliance-as-code ecosystem provides relevant foundations. Open Policy Agent (OPA) [12] and AWS Cedar [13] enable declarative policy enforcement, while regulatory technology research has explored encoding legal requirements in executable formats. These approaches address policy enforcement but do not generate the structured evidence artifacts needed for cross-framework governance verification.

In response to growing regulatory expectations, a rapidly expanding ecosystem of governance tools has emerged. Enterprise platforms such as Credo AI [14], Monitaur [15], Holistic AI [16], and Collibra AI Governance [17] provide AI system inventories, policy management, model monitoring, and regulatory mapping. Technical tooling such as Guardrails AI [18], NVIDIA NeMo Guardrails [19], OpenAI Evals [20] focus on runtime constraints, safety testing, and model evaluation. While valuable, most produce artifacts tightly coupled to individual tools or internal processes, limiting interoperability across governance frameworks, audit contexts, and organizations. Consequently, a structural gap remains between governance policy definitions and reusable operational evidence. This paper addresses that gap by proposing a measurement-oriented architecture that converts governance policies into reusable control objects and structured evidence artifacts supporting consistent governance verification over time.

III. Proposed Implementation

The proposed architecture consists of five steps: policy change detection, requirement extraction, structured control definition, reusable evidence schema design, and deterministic evidence collection. All artifacts require human review before release. The architecture is realized as a multi-agent system—Standards Monitoring Agent, Control Generation Agent, and Evidence Collection Agent—orchestrated by a Governance Orchestration Agent built on any stateful agentic framework such as LangGraph [6]. The overall architecture is illustrated in Fig. 1. AI agents assist in identifying regulatory updates and extracting candidate requirements.They suggest mappings and draft artifacts, but no controls are published and no code is modified without explicit human-in-the-loop (HITL) approval.

A. Step 1—Watch for Changes in Standards

The Standards Monitoring Agent monitors sources including NIST, ANSI, ISO, IEEE, the European Union (EU), the White House, and other government agencies, public regulations, and guidance documents via RSS, PDF, and Git connectors. Source documents are ingested using document parsing

libraries capable of paragraph-level text extraction, normalized into canonical segments, and compared against prior versions using structural diff techniques. A Large Language Model (LLM) assessment distinguishes material regulatory changes from cosmetic reformatting, preventing reviewer fatigue from false positives. When changes are detected, candidate requirements are surfaced for HITL review and documented with source references and timestamps. A confirmed material change triggers a targeted re-run of the Control Generation Agent (Step 2) scoped only to affected controls.

B. Step 2—Generate Controls

The Control Generation Agent converts raw governance prose into versioned control objects through a multi-step agentic pipeline powered by LLMs. The first step (Extractor LLM) identifies candidate requirement statements, filtering for *must* and *should* language and tagging scope qualifiers. The second step (Drafter LLM) writes a complete control YAML object including domain, lifecycle stage, evidence primitives, and a `recipe.yaml` stub that drives downstream collection. The third step (Critic LLM) challenges the draft: is intent accurately captured, is the scope too broad or too narrow, does it duplicate an existing control? A schema validation layer enforces structural correctness using typed schema enforcement libraries before any output reaches a human reviewer.

C. Step 3—Map and Categorize Controls

Controls are reviewed by a fourth LLM step (Dedup/Map LLM) that collapses overlapping controls across NIST AI RMF 1.0, ISO/IEC 42001, and the EU AI Act into single control objects with multiple source citations, eliminating redundant implementation work. The resulting catalog cross-references applicable frameworks by risk category and system component. A HITL domain reviewer spot-checks LLM-generated controls, approving, editing, or rejecting before promotion to the Git control registry.

D. Step 4—Define Evidence Templates

Rather than creating separate documents for each control, the system defines a small set of reusable evidence templates. Every evidence artifact follows the same outer structure - what control is this for, when was it collected, what system was checked, what tool collected it, what was observed, and what is the integrity hash, making evidence consistent across organizations. Each artifact follows a common envelope including `control_id`, `schema_version`, `collection_timestamp`, `collector_version`, `system_component`, `evidence_items`, provenance metadata, integrity hash, and lifecycle parameters. The `recipe.yaml` stub generated in Step 2 specifies which adapters and queries are required to satisfy each control, linking control definition directly to evidence collection.

E. Step 5—Automate Evidence Collection

The Evidence Collection Agent executes the `recipe.yaml` attached to each approved control and produces a structured evidence bundle. Open-source collectors run inside an organization's environment with read-only access. A unified query layer provides access across cloud platforms (AWS, GCP, Azure), Kubernetes, GitHub, databases, and SaaS systems through a consistent interface that maps directly to `recipe.yaml` adapter specifications. Secret and Personally Identifiable Information (PII) scanning tools handle DATA-LOG class controls. Where automation is not feasible, structured attestations and hashed configuration exports are supported. An LLM pass resolves ambiguous adapter results before the Evidence Assembler packages the final bundle. Evidence bundles include schema version, collector version, integrity hash, and expiration window, and can be serialized into OSCAL assessment-result layers [5]. All tools are open source; all changes go through review and code must pass tests; output must remain deterministic; releases are versioned and signed. No automatic publishing.

The tool stack is intentionally open. The Governance Orchestration Agent can be implemented using any modern framework such as LangGraph [6] that provides stateful multi-step workflows, human-in-the-loop interrupts, checkpointing, and parallel execution. The Control Generation Agent requires a vector store for the Retrieval-Augmented Generation (RAG) layer, document parsing libraries for PDF ingestion, and typed schema enforcement for output validation.

IV. Lifecycle Management

Evidence artifacts are time-bounded and include creation timestamps, defined validity windows, and re-measurement triggers (e.g., model retraining or configuration changes). Evaluation procedures are versioned to preserve longitudinal comparability. The governance layer operates at two levels. At the infrastructure level, the Governance Orchestration Agent coordinates agents and enforces HITL review gates before any artifact is published. At the institutional level, a neutral stewardship body could maintain open technical specifications: control schemas, evidence templates, and reference artifacts. Such a body would require explicit oversight mechanisms and would need to develop meta-standards governing schema versioning, evidence chain-of-custody, and cross-framework equivalence mapping capabilities that do not yet exist as formal standards. Organizations adopting the architecture maintain operational control over their own control catalogs and evidence collectors, but interoperability depends on shared schema conventions that only a coordinating body can establish.

A. Evidence Template

Control DATA-LOG-001 (*Sensitive personal data must not appear in application logs by default*) illustrates the evidence template design. This control is a cross-framework derived requirement synthesized from NIST AI RMF (MANAGE 4.1), EU AI Act (Articles 10 and 12), and ISO/IEC 42001 (A.8.5) illustrating the kind of operational control produced in Step 3. In a cloud-native deployment, an infrastructure query adapter

checks logging configuration, verifies encryption and retention settings, and scans logs for PII patterns via a secret and pattern scanning tool. It produces a `ScanResult` (log source, patterns checked, match count, tool version, timestamp) and a `ConfigurationSnapshot` (log level, redaction settings, encryption status, retention policy). In a legacy on-premises environment, administrators export configuration files and run the same scan locally. The evidence format remains identical across environments; only the collection adapter changes. Each bundle includes schema version, collector version, integrity hash, and a defined validity window. The architecture intentionally stops at structured evidence generation; interpretation of the evidence and compliance determination remain the responsibility of auditors, regulators, or adopting organizations.

V. Limitations and Threats to Validity

As a concept paper, this work identifies several anticipated limitations. First, LLM-generated controls may be plausible but technically incorrect; schema validation and the Critic LLM pass are proposed as primary defenses before human review, but their effectiveness at scale remains to be empirically validated. Second, ISO/IEC 42001 [3] is a paywalled standard; initial implementation is expected to target NIST AI RMF [2] and the EU AI Act [1], both freely available, before extending to ISO 42001. Third, the human-in-the-loop review step is the anticipated throughput bottleneck: reviewer expertise and availability will constrain pipeline output. Fourth, infrastructure query adapter coverage is strongest for public cloud environments; legacy and air-gapped systems rely on the structured attestation path, which introduces manual handling risk. Fifth, this architecture focuses narrowly on the measurement layer and does not reinterpret regulations or certify systems; downstream compliance decisions remain the responsibility of the adopting organization. Implementation and empirical validation are reserved for future work.

VI. Conclusion

Executable AI governance depends on more than policy interpretation. It requires structured controls and consistent evidence generation. By defining reusable control objects and deterministic evidence artifacts through a multi-agent architecture with human-in-the-loop review at every stage, organizations can demonstrate what they have implemented without relying on ad hoc documentation. This work focuses narrowly on the measurement layer. It does not reinterpret regulations or certify systems.

Future work will explore deeper alignment with OSCAL assessment layers, integration with model cards as evidence sources, and expansion of the control catalog to sector-specific AI regulations. Systematic evidence generation may also surface gaps in existing frameworks - controls that cannot be operationalized because the underlying regulatory text is ambiguous or silent on measurement criteria providing structured feedback to standards bodies such as NIST or the EU AI Office. For technology leaders, this approach provides a repeatable mechanism for linking policy obligations to observable system evidence, reducing manual compliance effort while improving audit transparency.

Acknowledgment

The authors used generative AI tools to assist with language clarity and figure preparation. ChatGPT (OpenAI) was used for limited language refinement and editing. Gemini (Google) was used to generate the architecture diagram based on the authors' conceptual sketches. All research design, technical content, analysis, interpretations, and conclusions were developed and validated solely by the authors.

References

[1] European Union, "Regulation (EU) 2024/1689 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)," *Official Journal of the European Union*, 2024.

[2] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, Jan. 2023.

[3] International Organization for Standardization and International Electrotechnical Commission, *ISO/IEC 42001:2023 Information Technology—Artificial Intelligence—Management System*, 2023.

[4] G. V. Datla et al., "Policy→Tests (P2T) for operationalizing AI governance," GitHub, 2025.

[5] National Institute of Standards and Technology, "Open Security Controls Assessment Language (OSCAL)," 2023. [Online]. Available: https://pages.nist.gov/OSCAL/

[6] LangChain, Inc., "LangGraph: Build Stateful, Multi-Actor Applications with LLMs," 2024. [Online]. Available: https://github.com/langchain-ai/langgraph

[7] I. D. Raji, A. Smart, R. N. White et al., "Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing," in *Proc. 2020 Conf. Fairness, Accountability, and Transparency (FAccT)*, 2020.

[8] S. Costanza-Chock, A. Raji, and L. Buolamwini, "Algorithmic impact assessments: a practical framework for public agency accountability," AI Now Institute, 2022.

[9] P. Guldimann, A. Spiridonov, R. Staab, and N. Jovanovic, "COMPL-AI framework: a technical interpretation and benchmarking suite for the EU AI Act," 2024, arXiv:2410.07180. [Online]. Available: https://arxiv.org/abs/2410.07180

[10] S. Feuerriegel, N. Hartmann, C. Janiesch, and P. Zschech, "Compliance challenges for high-risk AI systems under the EU AI Act: an empirical investigation," *Computers & Security*, 2026. [Online]. Available: https://doi.org/10.1016/j.cose.2026.056

[11] IEEE Standards Association, *IEEE P2863: Recommended Practice for Organizational Governance of Artificial Intelligence*, IEEE, 2022.

[12] Open Policy Agent, "OPA: Policy-based control for cloud native environments," CNCF, 2024. [Online]. Available: https://www.openpolicyagent.org/

[13] AWS, "Cedar: A language for defining permissions as policies," 2023. [Online]. Available: https://www.cedarpolicy.com/

[14] Credo AI, "Credo AI Responsible AI Governance Platform," OECD AI Policy Observatory, 2023.

[15] Monitaur, "Monitaur AI Governance Platform," OECD AI Policy Observatory, 2023.

[16] Holistic AI, "Holistic AI Governance Platform," Holistic AI, 2024. [Online]. Available: https://www.holisticai.com/ai-governance-platform

[17] Collibra, "Collibra AI Governance," Collibra, 2024. [Online]. Available: https://www.collibra.com/us/en/products/collibra-ai-governance

[18] Guardrails AI, "Guardrails: Adding guardrails to large language models," GitHub, 2024. [Online]. Available: https://github.com/guardrails-ai/guardrails

[19] NVIDIA, "NeMo Guardrails: An open-source toolkit for controllable and safe LLM applications with programmable rails," GitHub, 2023. [Online]. Available: https://github.com/NVIDIA-NeMo/Guardrails

[20] OpenAI, "Evals: A framework for evaluating large language models and LLM systems," GitHub, 2023. [Online]. Available: https://github.com/openai/evals