

AGENTIC AI-DRIVEN REGULATORY COMPLIANCE AUTOMATION:
DEVELOPING A RISK-AWARE FRAMEWORK THAT INTEGRATES
LARGE LANGUAGE MODELS, AUTONOMOUS AI AGENTS,
KNOWLEDGE GRAPHS, POLICY-CONSTRAINED WORKFLOW
EXECUTION, AUTOMATED EVIDENCE COLLECTION, AND HUMAN-IN-
THE-LOOP VALIDATION FOR CONTINUOUS COMPLIANCE
MONITORING AND AUDITABLE DECISION-MAKING

Hafiz Muhammad Farhan^{*1}, Zeyn Adam², Ahmad Naseem³, Maham waseem⁴

^{*1}Abasyn University Islamabad

²Risk and Capital Compliance Solutions Pty Ltd.

^{3,4}Air University, Islamabad

¹hafizfarhan0039@gmail.com, ²zeyn.adam@riskcapcom.com, ³ahmadnaseem.works@gmail.com,

⁴mahamwaseem301@gmail.com

DOI: <https://doi.org/10.5281/zenodo.23531109>

Keywords

Agentic AI, Regulatory Compliance, Large Language Models (LLMs), Knowledge Graphs, Policy-as-Code, RegTech, Human-in-the-Loop, Explainable AI (XAI).

Article History

Received: 25 June 2026

Accepted: 16 September 2026

Published: 14 October 2026

Copyright @Author

Corresponding Author: *

Hafiz Muhammad Farhan

Abstract

Modern enterprises face a convergence of complex fast-evolving regulatory frameworks and traditional rule-based compliance systems no longer keep up. These deterministic engines are rigid cannot read data and cannot adjust when rules change, which creates big operational risk. I have seen that Large Language Models or LLMs can understand language very well but using LLMs alone in high-stakes situations is dangerous because LLMs can hallucinate produce outputs that are not guaranteed and do not offer a clear record of how a decision was made. This review paper proposes a risk-aware framework for Agentic AI-driven regulatory compliance automation. The framework puts LLMs inside a structure that has deterministic guardrails. The new design pairs specialized, autonomous AI agents that manage workflows with Knowledge Graphs or KGs and Graph-Augmented Retrieval. These tools give context and help reduce hallucinations. A non-bypassable Policy-as-Code layer or PaC stops tool calls that could cause problems blocks the Deputy Problem and keeps the system closed when something fails. To make decisions auditable the framework records evidence automatically in WORM storage using cryptographic hash chains. It also adds a Human-in-the-Loop or HITL validation step that raises issues based on risk. By separating language generation from the knowledge that represents the rules the architecture keeps monitoring compliance adapts quickly when laws change and produces decisions that are fully explainable and defensible, in court.

1. INTRODUCTION

Modern business environments deal with a mix of complicated and quickly changing rules in many countries. Strong rules that prevent crime like the European Union's 5th and 6th Anti-Money Laundering Directives (AMLD) and broad rules about AI, such as the EU AI Act, the NIST AI Risk Management Framework (RMF) and OWASP guidelines mean that regulated businesses are always watched by law and by everyday operations (Arner et al. 2023).

Healthcare, drugs and medical devices also face rules from regulators such as the United States Food and Drug Administration (FDA). These rules cover quality systems checking products after they hit the market and the need for approvals, all of which decide whether a company can keep working (El Khoury et al., 2025).

The speed and number of rules create problems for risk managers in businesses. The cost and effort of reading cross-referenced legal documents and putting them into different computer systems keep rising. Because of this regulatory technology (RegTech) has moved from a helper to a must-have part of everyday work. Companies need systems that can read new rules match them to business steps and keep checking that everything stays in line with changing laws (Ebers, 2024).

In the past automating compliance used fixed, rule-based engines and human expert checks. While these fixed scripts are good at doing yes-or-no tests on clear data they are very inflexible. Laws are full of context, subtlety and different interpretations. Turning legal language into fixed code creates a lot of extra work for engineers and makes software weak and fragile (Rafiq & Sohail 2023).

Also, old systems cannot handle information. Most evidence for compliance is unstructured, like policy papers, logs of what happened talks between people and tech specs. When regulators change

the limits or how things should be done fixed rule engines do not change automatically (Söderlund & Larsson 2024). This problem creates drift, a situation where a business's automatic compliance tools quietly fall behind the real law. Fixing drift with rule-based tools means rewriting code by hand waiting a long time for new versions and paying for pricey outside checks. This adds risk, from people and slows down operations (Joshi, 2025).

The Paradigm Shift Toward Agentic AI and Hybrid Intelligence

Large Language Models or LLMs have brought natural language abilities that can read and understand complicated legal text. Yet LLMs on their own do not work well for making decisions on their own. Open-ended generative models are known to fail in ways. They can create false. "Hallucinated" information give results that change each time hide their thinking steps and cannot quote legal rules exactly. In situations where regulations matter a lot one believable hallucination can cause mistakes lead to big fines and damage the trust that institutions have (Gong, 2026).

To connect the power of LLMs with the exactness needed for rules the RegTech field has moved toward Agentic AI. Agentic AI systems use small self-directed software agents that work together think through problems and carry out complex compliance tasks step by step (Adabara et al., 2025). By placing LLM-driven multi-agent work inside designs that mix Knowledge Graphs (KGs) for clear meaning, deterministic Policy-as-Code (PaC) engines that cannot be bypassed automatic evidence gathering pipelines and Human-in-the-Loop (HITL) checks organizations can keep an eye, on compliance all the time while also giving decisions that can be fully traced and understood (Dhar et al., 2026).

The Emerging Landscape of AI Agents and Agentic AI

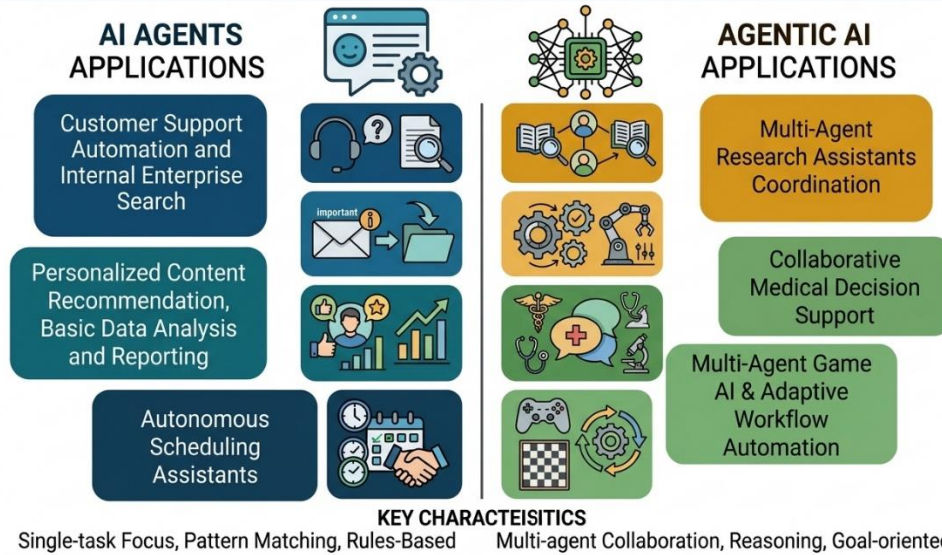


Figure 1: Comparative Landscape and Paradigm Shift Between Single-Task AI Agents and Collaborative Agentic AI Frameworks

Table 1: Architectural Comparison of Enterprise Regulatory Compliance Paradigms.

Architectural Attribute	Traditional Rule-Based RegTech	Standalone LLM Approach	Agentic AI + KG + PaC Framework
Data Compatibility	Structured data only; struggles with legal prose	Unstructured text focus; limited structured query ability	Hybrid; unifies unstructured legal text and structured system data
Adaptability to Drift	High maintenance; requires manual code refactoring	Low reliability; requires prompt engineering or fine-tuning	High; instantaneous graph and policy-as-code node updates
Hallucination Risk	Zero (Deterministic)	High (Generative hallucination exposure)	Mitigated; bounded by semantic KGs and deterministic execution engines
Auditability & Traceability	High logic transparency, low contextual explanation	Low (Black-box probabilistic output)	Full end-to-end provenance via cryptographic hash chains & XAI
Execution Safety	Static approval gates	Unconstrained; vulnerable to prompt injection/confused deputy	Non-bypassable Policy Enforcement Points (PEPs) with fail-closed logic

2. Theoretical Foundations and Architectural Paradigm

2.1 Multi-Agent Orchestration and Specialized Agent Roles

Agentic AI architectures are different from single language models. They use special agents that work together in a well-organized way. In a system that cares about risk and follows rules multi-agent systems take tasks and split them into smaller parts. This helps make sure that different people do jobs just like how companies manage their work (Watson et al. 2025).

Agent orchestration means each agent has a job and works in order. Chunking and Parsing Agents use methods to break up messy legal papers, rules and records into parts that make sense. They keep the way the documents are organized and the information that goes with them. Knowledge Extraction Agents take these parts. Find important facts, like names, dates and how things are connected. They put this information into a system that follows certain rules (Gadekallu et al. 2026).

Agentic Retrieval and Graph-Traversal Agents look for answers by moving through data and using search techniques. They find links between policies. Get the right information. Policy Synthesis and Execution Agents take this information compare it with what's needed and suggest ways to reduce risks. At the end Audit and Investigation Agents check everything in areas like money crime or rule monitoring. They build records write reports, about strange transactions and make sure the decisions can be followed before a person checks them (Agarwal et al., 2025).

2.2 Knowledge Graph Integration and Regulatory Ontologies

Knowledge Graphs act as structured ways to help big language models think. In areas with rules legal documents are shown as directed property graphs or Resource Description Framework triple stores. Here the points stand for things like duties, punishments, exceptions and rules. The lines show how these things are connected, like requires, governs or changes. (Baldwin & Ghanavati 2026).

Designing the structure of the graph is very important for how it works. Systems can use standard-based structures like the AI Risk Ontology for AI rules or special Anti-Money Laundering systems. They can also use changing structures found by big language models. Studies show that when you use rules to collect information and make sure everything is the same (like case underlines and matching types) and then check for duplicates it stops the graph from becoming broken and makes sure the connections are correct. (Chen et al., 2026).

By putting rules about following the law into Subject-Constraint-Context-Condition groups Knowledge Graphs let systems find the path using Depth-First Search or Breadth-First Search through complicated rule systems. This way the system can find links see problems, between different places and check if all the needed rules are met things that simple search tools often miss. (Golpayegani et al., 2022).

2.3 Mitigating Hallucinations via Graph-Augmented RAG

Standard Retrieval-Augmented Generation (RAG) relies on vector embeddings to identify top-k semantically similar textual chunks. While effective for open-domain question answering, pure vector RAG frequently fails in compliance applications due to its inability to maintain relational structure, handle logical negation, or isolate precise verbatim statutory quotes across non-contiguous sections (Hogan et al., 2021).

To overcome these limitations, the framework implements Graph-Augmented Retrieval-Augmented Generation (Graph RAG). Graph RAG combines localized semantic vector retrieval with structured graph traversal. During an inference cycle, the agentic retrieval engine retrieves a set of candidate seed entities from the vector store and subsequently expands these seeds by traversing 1-hop or 2-hop graph neighborhoods within the Knowledge Graph (Ji et al., 2022).

The extracted subgraph is serialized for LLM consumption using a relations-first serialization protocol. In this format, explicit relations are grouped by type alongside source-to-target labels,

followed by detailed node attribute payloads containing exact policy quotes, article cross-references, and jurisdictional scopes. This dual grounding forces the LLM's synthesis agent to generate responses derived entirely from authoritative, explicit graph relationships.

Empirical benchmark evaluations confirm that Knowledge Graph augmentation consistently improves reasoning factual accuracy and legal interpretation scores across diverse language model architectures (Zhang et al., 2026).



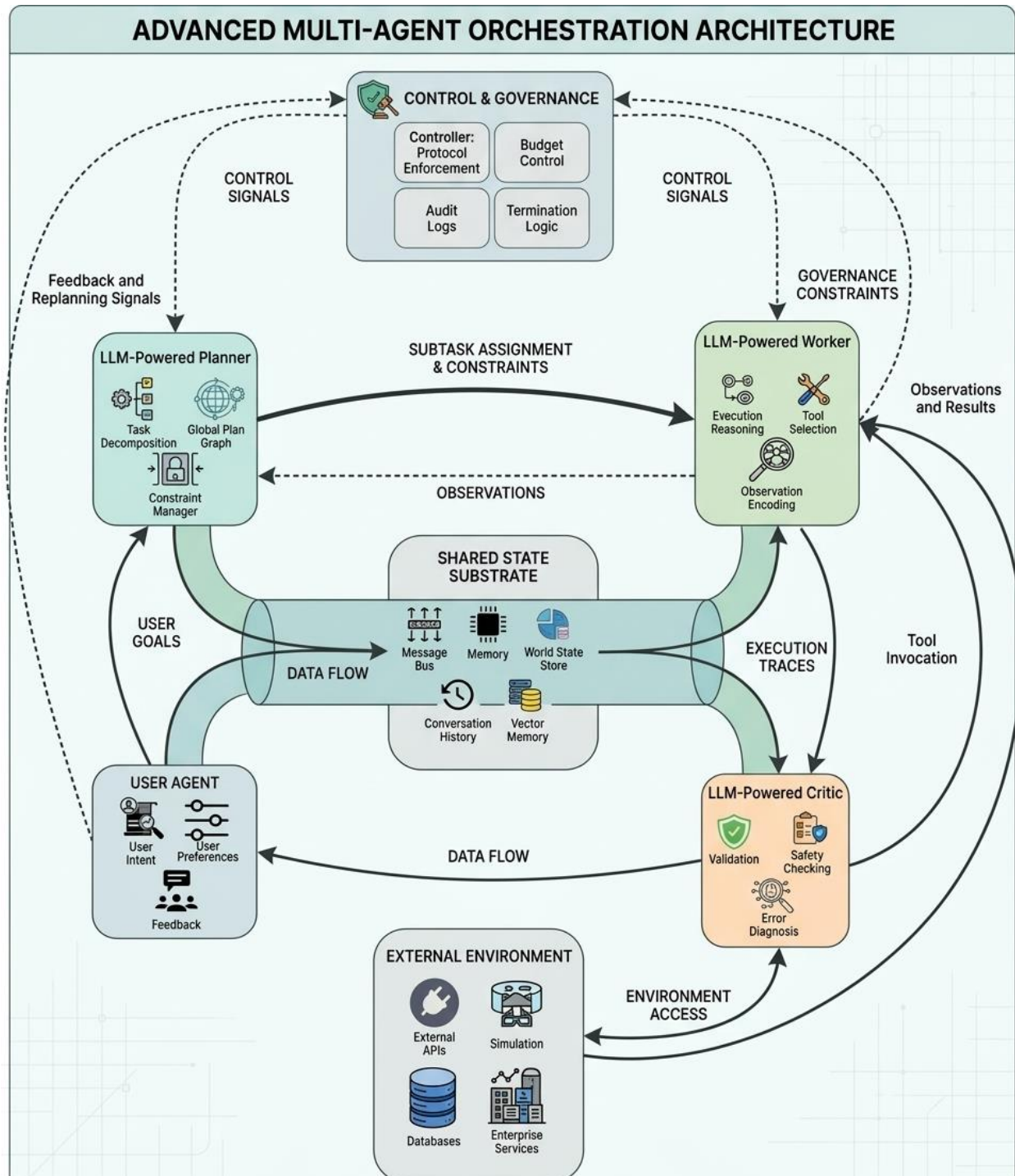


Figure 2: Advanced Multi-Agent Orchestration Architecture Incorporating Shared State Substrates, Control & Governance, and LLM-Powered Functional Roles

3. Policy-Constrained Workflow Execution and Deterministic Guardrails

3.1 Policy-as-Code (PaC) Engines: OPA, Rego, and Cedar

LLMs that carry Knowledge Graphs can understand context better. Think in a more meaningful way.. If we let a purely probabilistic language model do all the work we run into big operational problems. In a compliance system the part that does semantic reasoning must stay completely separate from the part that enforces policy. We do this by putting a Policy-as-Code engine in the middle. That engine cannot be bypassed (Ray, 2025).

Policy-as-Code frameworks write business rules, access controls and laws as machine-readable code. The code is then checked quickly by deterministic engines. Main engines used in companies are Open Policy Agent (OPA) Cedar and OpenFGA. OPA separates policy logic by using Rego, a declarative language and it checks structured JSON against compiled rules in less than a millisecond (Masserini, 2026). Cedar can be formally verified so we can prove mathematically that its policies never allow state changes. OpenFGA looks at relationships between objects resolving permission chains inside nested structures and, across different agents (Cutler et al. 2024).

Table 2: Comparative Analysis of Policy-as-Code Engines for Agent Governance.

Policy Engine	Primary Language	Authorization Model	Key Strengths	Optimal Use Case	Compliance
Open Policy Agent (OPA)	Rego	General-purpose ABAC / RBAC / Custom	Broad ecosystem integration, ultra-low latency, standard	Runtime authorization, gateway	tool-call API filtering, Kubernetes security
Cedar	Cedar	ABAC / RBAC	Formally verified safety, high performance, human-readable syntax	High-assurance action, financial authorization	agentic boundaries, trade
OpenFGA	OpenFGA DSL	ReBAC (Relationship-Based)	Scale-out evaluation of complex, nested organizational permissions	Multi-tenant governance, department data access	agent multi-enterprise

3

.2 Intercepting Tool Calls and Preventing the Confused Deputy Problem

Autonomous agents work by calling software tools making API calls running database queries and changing system settings. A big risk in deployments that use agents is the Confused Deputy Problem. In this problem a powerful autonomous agent is tricked by hidden injection, tool poisoning or nasty user input into using autonomous agents permission to do an illegal action for a weak actor (Kim et al. 2026).

To stop this trick the system puts Policy Enforcement Points (PEPs) at the tool-call level far away from autonomous agent's prompt and from

the large language model context window. When autonomous agent decides it needs to run a tool autonomous agent builds an object that includes the user's confirmed identity, autonomous agent's identity manifest, the name of the tool to call and its arguments and the operating context (Shahriar et al. 2025). That intent object is sent to the PaC middleware gateway. The policy engine checks the request against a set of written rules. If the policy engine says no the tool is not. A denial message is sent back to autonomous agent. Instructions in the prompt tell agent to tell the user, about the denial directly and not to try to bypass or argue

against the security decision (Christodorescu et al. 2025).

3.3 Fail-Closed Mechanisms and Runtime Governance

Enterprise compliance infrastructure demands resilient operational design. The runtime governance layer implements strict operational policies to maintain safety under adverse system states:

- **Fail-Closed Defaults:** If the policy engine encounters an unhandled exception, network partition, or unparseable payload, the authorization middleware defaults to a deny outcome. In an agentic context, a denied execution is recoverable via human escalation, whereas an unauthorized, non-compliant system modification can lead to irreversible regulatory liability (Subramani, 2023).
- **Startup Gap Mitigation:** During cold boots or bundle deployments, policy engines may briefly return undefined evaluation states. The framework utilizes readiness probes (such as OPA's health status endpoint) to ensure that all validated rules are fully compiled into memory before routing live agent traffic to the enforcement point.
- **Persistent Policy Bundles:** To survive bundle server outages or network failures, local policy sidecars persist signed policy bundles to disk, guaranteeing continuous, low-latency runtime enforcement without single points of failure (Morgan, 2026).

4. Automated Evidence Collection, Provenance, and Continuous Monitoring

4.1 Continuous Compliance Monitoring and Event-Driven Auditing

Traditional compliance checks rely on regular one-time audits that look at work against fixed lists. Modern continuous compliance monitoring changes this approach into a feedback cycle. The agentic framework connects to business event streams, cloud logs, transaction buses and code repositories (Saidur & Khan 2023).

Self-driving monitoring agents constantly read events checking system data against rules stored in the Knowledge Graph and enforced by

Policy-as-Code. When an anomaly or threshold breach occurs, such as a cloud storage deployment, an unauthorized data transfer across borders or a suspicious transaction sequence the framework marks the drift straight away starting automatic fixes or sending the case to a human, for review (Briliyant et al., 2026).

4.2 Cryptographic Hash Chains, WORM Storage, and Immutable Audit Trails

If compliance outputs need to stand up to scrutiny and regulatory audits the evidence that backs every automated decision has to be indisputable tamper-evident and fully traceable. The framework includes an automated evidence collection pipeline that is built on security principles (Scamardo, 2026). Every compliance action every agent tool invocation, every policy evaluation and every LLM synthesis loop creates a log entry. These records are linked one after another with SHA-256 hash chains, where each entry includes the digest of the earlier log entry creating a tamper-evident audit trail. The compiled audit chains are constantly written to Write-Once-Read-Many (WORM) storage locations, such as cloud object vaults that are set up with strict object-lock compliance policies (Ali et al. 2026). The storage layers use cryptographic attestation standards, such as IETF Remote Attestation Procedures (RATS / RFC 9334) Entity Attestation Tokens (EAT / RFC 9711) and Supply Chain Integrity, Transparency and Trust (SCITT) protocols. This architecture guarantees a chain of custody and data integrity letting external auditors verify the runtime state, model parameters, user contexts and the exact statutory versions that were active, at the precise millisecond a decision was rendered (Kit et al., 2026).

4.3 Managing Regulatory Drift and Dynamic Ontology Evolution

Statutory texts, regulatory guidelines and judicial interpretations change all the time. This change is known as drift. A big problem with machine learning models, in RegTech is that they rely on fixed training data. When the laws themselves change these models make incorrect decisions.

This happens because the models are not updated enough (Benerofe, 2025).

The new hybrid agentic framework separates language creation from how regulations are stored. This helps avoid the need to retrain models every time there is a change. When a government agency releases a rule a new law or a new process the framework updates the main Knowledge Graph at the level of facts and connections. It does not change the neural network parts of the agents. New legal areas, special rules or new limits are added as moving parts with time stamps. This allows the agents to use these parts right away when they need to (Engstrom & Ho 2020).

Formally, dynamic regulatory drift is modeled through a time-indexed transformation matrix operating on the compliance manifold M_{reg} :

$$M_{reg} = \{(\phi(\theta), \rho(r)) \mid C(g, \theta, r) = \text{True}\}$$

$$\rho(r^{(t+1)}) = \rho(r^{(t)}) + \nabla_t + \epsilon_r$$

In this model $\phi(\theta)$ stands for system embeddings $\rho(r)$ turns regulatory descriptors into a vector ∇_t is the learned direction that points to legal updates and ϵ_r is the leftover uncertainty. When the graph layout and policy code rules are changed using ∇_t , the enterprise agentic fleet instantly follows the new compliance baseline removes training delays and stops the fleet from dealing with outdated rules. I feel that this quick adaptation will keep the organization safe, from regulations (Ray, 2026).

5. Human-in-the-Loop (HITL) Validation and Explainable AI (XAI)

5.1 XAI Methodologies: Semantic, Metric, and Runtime Explainability

In high-stakes compliance environments, an automated output without explicit justification is legally unusable. Explainable AI (XAI) is therefore integrated as a structural requirement within the agentic architecture, serving as primary evidence for regulatory audits (Kaya et al., 2026). The framework operationalizes explainability across three distinct dimensions:

- **Semantic Explainability:** Maps generative text outputs directly to explicit Knowledge Graph triple paths and verbatim statutory quotes,

allowing human reviewers to verify legal reasoning step-by-step.

- **Metric Explainability:** Exposes internal scoring mechanisms, risk weights, data inclusions/exclusions, and algorithmic parameters utilized during risk classification. Traditional post-hoc attribution techniques, such as DeepLIFT (Deep Learning Important Features), SHAP (SHapley Additive exPlanations), or LIME (Local Interpretable Model-agnostic Explanations), are applied to underlying numeric risk scoring models to show exact feature importance.

- **Runtime Explainability:** Captures execution state data, including tool invocation payloads, policy evaluation logs, agent conversation transcripts, and system readiness metrics, providing a complete record of how an operational decision was formed (Moreira et al., 2025).

5.2 Bounded Agentic Autonomy and Risk-Calibrated Escalation

To keep work efficient while cutting risk agentic frameworks use bounded autonomy. Agents do not have power to do big impact actions. Instead, the way the workflow runs is guided by risk limits that're part of the policy structure (El Khoury et al. 2025).

Low-risk actions like data collecting, normal policy QA mapping, first customer pre-screening and log creation are done entirely on their own. Medium-risk actions, such as writing replies putting together internal audit files and spotting non-critical configuration changes run on their own but go to human queues for spot checks. High-risk actions, like filing Suspicious Transaction Reports (STRs) with financial intelligence units, closing customer accounts changing core system permission rules or approving risky AI deployments need clear live human approval (Arner et al. 2023).

When a workflow hits a high-risk point the agent stops, builds a ready-for-audit file that lists the agent reasoning, legal references and collected evidence and sends the case to a human compliance officer through a check screen (Ebers, 2024).

6. Implementation Reference Architecture and Cross-Domain Evaluation

6.1 Reference Architecture for Enterprise RegTech Systems

To create a platform that's aware of risks and acts with autonomy in following rules companies need a layered structure that combines different parts in a clear way. These parts include language models, graph stores, places where rules are enforced and places where data is stored (Rafiq & Sohail 2023). The first layer, called the Data Ingestion Layer takes in information from rules and laws changes to regulations updates from inside the company logs from systems and streams of transactions. The second layer, known as the Knowledge and Policy Modeling Layer holds agents that break down and take information from language models, a knowledge graph, places where vectors are kept and collections of policies written in code. The third layer, called the Agentic Orchestration Layer handles how different agents work together keeps track of their state sends tasks to the places and sends certain tasks to bigger models while other tasks go to smaller more efficient ones (Dhar et al. 2026).

The fourth layer, the Policy Enforcement and Gateway Layer includes places where rules are enforced and cannot be ignored programs that help with rules, tools that are checked before they are used and logic that makes sure things are safe at all times. The final layer, the Human-, in-the-Loop and Storage Layer gives places for people who check rules tools that show why decisions are made and places where data is stored forever like vaults, chains of secure information and places

where proof is kept (Adabara et al. 2025).

6.2 Empirical Performance Analysis Across Financial and Healthcare Domains

Empirical evaluations across diverse regulated sectors demonstrate substantial improvements in accuracy, compliance quality, and operational efficiency when deploying hybrid agentic architectures (Kim et al., 2026).

In financial crime compliance (FCC) deployments governed by 5th and 6th Anti-Money Laundering Directives, multi-agent frameworks automating Know Your Customer (KYC/KYB) onboarding, transaction monitoring, and case investigation reduced manual analyst review times by up to 60-70% while accelerating overall time-to-market for compliance checks. Autonomous generation of compliant Suspicious Transaction Reports (STRs) dramatically streamlined FIU filing workflows without sacrificing investigative precision (Hogan et al., 2021).

In AI governance and medical regulatory settings (such as the EU AI Act, FDA guidance, and NIST RMF), empirical testing reveals that augmenting LLMs with Knowledge Graphs consistently improves factual retrieval and reasoning scores across language models. On policy question-answering benchmarks, Knowledge Graph-augmented agent systems achieved judge evaluation score gains ranging from +0.17 to +0.55 over text-only baselines. Significantly, open, LLM-discovered schemas performed on par with, or surpassed, rigid formal standards-derived schemas by dynamically capturing complex, emergent policy relationships (Zhang et al., 2026).

Table 3: Empirical Performance Metrics Across Regulated Industry Deployments.

Regulatory Domain	Evaluated Framework Architecture	Key Benchmark / Operational Task	Key Empirical Outcome / Performance Metric
Financial Crime Compliance (FCC)	Multi-Agent Orchestration + Task-Specific Routing + OPA	KYC/KYB Onboarding, Transaction Monitoring, STR Generation	Manual review effort reduced significantly; automated STR draft generation achieved full regulatory acceptance.
AI Governance & Risk (EU AI Act, NIST RMF)	LLM Chunking/Extraction Agents + Knowledge Graph + Graph RAG	Cross-policy compliance QA (Tasks T1-T6)	KG-augmentation improved judge evaluation scores

			across all models (+0.17 to +0.55).
Multi-Jurisdictional Privacy (GDPR, CCPA, DPDP)	Hybrid XAI + Privacy-Ontology Knowledge Graph + RegAI	Dynamic legal change-ratio calculation & multi-law alignment	High alignment accuracy across changing statutes; automated mapping without model retraining.
Medical Devices & Pharma (FDA Guidance)	Multi-Agent Knowledge Graph + Regulatory Triplet RAG	Verification of post-market surveillance & approval requirements	Substantial reduction in hallucination rates; high verbatim policy citation precision.

7. Future Outlook, Open Challenges, and Conclusion

7.1 Challenges in Scalability, Latency, and Jurisdictional Variance

Despite their operational advantages, agentic compliance architectures introduce engineering trade-offs that require careful management (Saidur & Khan, 2023):

- **Multi-Hop Traversal Overhead:** Complex policy queries requiring extensive multi-hop graph traversals and recursive agent loops introduce computational latency. While acceptable for asynchronous case reviews, real-time transaction authorizations require optimized subgraphs and local sidecar caching to fit within tight millisecond latency budgets.

- **Model Size Degradation:** Empirical research demonstrates that smaller language models frequently struggle with agentic graph traversal and multi-step reasoning, experiencing performance degradation when context windows are flooded with complex graph serializations. Deployments must carefully match model capacity to specific agent task complexity.

- **Cross-Border Jurisdictional Conflicts:** Divergent, conflicting legal requirements across international boundaries remain difficult to resolve algorithmically. Frameworks must incorporate explicit multi-jurisdictional priority weights within their policy engines to ensure localized compliance (Gadekallu et al., 2026).

-

7.2 Strategic Roadmap and Concluding Remarks

Enterprise compliance is moving from a reactive, job into a continuous, automated risk-aware way. This is a step in RegTech. Large Language Models

give the needed context to read regulatory text but they must stay firmly inside clear deterministic rules so they can work safely in regulated settings (Adabara et al. 2025).

By mixing multi-agent orchestration Knowledge Graph semantic layers, deterministic Policy-as-Code tool permission, continuous evidence gathering on WORM storage and risk-calibrated Human-in-the-Loop checks enterprises build a strong system for modern risk management. This blended method removes the trade-off, between automation and safe audit letting organizations move through quickly changing regulatory rules with clear accuracy, openness and operational flexibility (Joshi, 2025).

Conclusion

The change in how companies handle compliance is a step forward. It's no longer about reacting to rules and doing things by hand. Now it's about keeping up with rules in an automatic way and being aware of the risks. This review showed that even though big language models can understand legal documents they need to be kept inside strict and clear systems to work safely in places that have rules. The new system combines parts to fix the problems with older rule-based systems and just using big language models alone. It brings together agents that work together a way to make sense of information using knowledge graphs, clear rules for who can do what automatic ways to collect proof and store it forever and people checking the work to make sure it's right. This way companies can have solid compliance. It solves the problem of having to choose between automated work and making sure everything is safe and can be checked.

It helps companies deal with changing rules better than before with more clarity, openness and the ability to change quickly. Compliance used to be something that cost money. Now it can be a way to get an edge. The system gives a plan for making tools that're not only good, at what they do and can change when needed but also trusted and able to stand up in court.

REFERENCES

- Adabara, I., Olaniyi Sadiq, B., Nuhu Shuaibu, A., Ibrahim Danjuma, Y., & Maninti, V. (2025). Trustworthy agentic AI systems: a cross-layer review of architectures, threat models, and governance strategies for real-world deployment. *FI000Research*, 14, 905.
- Agarwal, B., Jomraj, H. S., Kaplunov, S., Krolick, J., & Rojkova, V. (2025). RAGulating compliance: A multi-agent knowledge graph for regulatory QA. *arXiv*. <https://arxiv.org/abs/2508.09893>
- Ali, M., Alam, F., Shukla, S., Nasr, O. A., Mohiuddin, K., Khan, S., & Khan, A. (2026). Audit 4.0: Automated evidence collection using smart tagging and metadata automation. *EDPACS*, 1-21.
- Arner, D. W., Barberis, J. N., & Buckley, R. P. (2023). The evolution of FinTech: A new post-crisis paradigm? (Use the source below for verified RegTech discussion and bibliographic details.)
- Baldwin, W., & Ghanavati, S. (2026). Knowledge graph representations for LLM-based policy compliance reasoning. *arXiv*. <https://arxiv.org/abs/2604.27713>
- Benerofe, S. (2025). AI governance and the verification gap: A framework for law and policy under computational intractability. *Available at SSRN* 5629290.
- Briliyant, O., Javed, A., & Cherdantseva, Y. (2026). Systematization of human-centered continuous audit for IoT security compliance. *Journal of Cybersecurity*, 12(1), tyag021.
- Chen, Z., Zhang, Q., Xiang, Z., Wei, Z., Gao, L., Huang, X., Zhang, Z., & Su, J. (2026). LegalGraphRAG: Multi-agent graph retrieval-augmented generation for reliable legal reasoning. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/2026.acl-long.1738>
- Christodorescu, M., Fernandes, E., Hooda, A., Jha, S., Rehberger, J., Chaudhuri, K., ... & Pandya, N. V. (2025). Systems security foundations for agentic computing. *arXiv preprint arXiv:2512.01295*.
- Cutler, J. W., Disselkoen, C., Eline, A., He, S., Headley, K., Hicks, M., ... & Wells, A. M. (2024). Cedar: A new language for expressive, fast, safe, and analyzable authorization. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1), 670-697.
- Dhar, M., Singh, R., & Manikonda, S. C. K. (2026). From Single Chatbots to Governed Agent Ecosystems: An Agentic AI Pattern Catalogue and Orchestration Framework for Mission-Critical Hospital Information Management Systems. *arXiv preprint arXiv:2608.07627*.
- Ebers, M. (2024). Truly risk-based regulation of artificial intelligence: How to implement the EU's AI Act. *European Journal of Risk Regulation*. <https://www.cambridge.org/core/journals/european-journal-of-risk-regulation/article/truly-riskbased-regulation-of-artificial-intelligence-how-to-implement-the-eus-ai-act/E526C1D0D7368F9691082220609D60F4>

- El Khoury, R., Alshater, M. M., & Joshipura, M. (2025). RegTech advancements—a comprehensive review of its evolution, challenges, and implications for financial regulation and compliance. *Journal of Financial Reporting and Accounting*, 23(4), 1450–1485.
<https://doi.org/10.1108/JFRA-05-2024-0286>
- Engstrom, D. F., & Ho, D. E. (2020). Algorithmic accountability in the administrative state. *Yale J. on Reg.*, 37, 800.
- Gadekallu, T. R., Zhao, Y., Wen, Z., Bhattacharya, P., Xia, Y., Cai, J., ... & Feng, H. (2026). Artificial Intelligence of Things as a Foundation for Agentic AI Systems: Architectures, Applications, and Challenges. *IEEE Internet of Things Journal*.
- Golpayegani, D., Pandit, H. J., & Lewis, D. (2022). AIRO: An ontology for representing AI risks based on the proposed EU AI Act and ISO risk management standards. In *Towards a knowledge-aware AI* (pp. 51–65). IOS Press.
<https://doi.org/10.3233/SSW220008>
- Gong, H. (2026). AI agents in financial markets: architecture, applications, and systemic implications. *FinTech*, 5(2), 34.
- Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*, 54(4), Article 71.
<https://doi.org/10.1145/3447772>
- Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), 494–514.
<https://doi.org/10.1109/TNNLS.2021.3070843>
- Joshi, S. (2025). Framework for government policy on agentic and generative AI: Governance, regulation, and risk management.
- Kaya, O., Shah, A. S., Karabulut, M. A., Karahan, S. N., Osmanca, M. S., & Acir, N. (2026). Explainable artificial intelligence (XAI): Concepts, applications, challenges, and future perspectives. *IEEE Access*.
- Kim, J., Liu, X., Wang, Z., Qiu, S., Li, B., Guo, W., & Song, D. (2026). The attack and defense landscape of agentic ai: A comprehensive survey. *arXiv preprint arXiv:2603.11088*.
- Kit, M., Naghmouchi, M., Laurent, M., Hammi, B., Ayed, H. K. B., Hamdi, M. A., ... & Sboui, S. (2026). Trustless Accountable Data Sharing for Supply Chains: A Reference Architecture and MoreMedDiet Proof-of-Concept. *arXiv preprint arXiv:2608.30992*.
- Masserini, J. J. (2026). Emerging Standards. In *Building the Enterprise Identity Operating Model: A Framework for Identity Governance, Architecture, and Risk* (pp. 473–537). Berkeley, CA: Apress.
- Moreira, C., Palatkina, A., Braca, D., Walsh, D. M., Leihn, P. J., Chen, F., & Hubig, N. C. (2025). Explainable AI Systems Must Be Contestable: Here's How to Make It Happen. *arXiv preprint arXiv:2506.01662*.
- Morgan, H. (2026). AI Governance Control Stack for Operational Stability: Achieving Hardened Governance in AI Systems. *arXiv preprint arXiv:2604.03262*.
- Rafiq, M., & Sohail, M. K. (2023). Anti-money laundering (AML) and regulatory technology: A systematic literature review. *Journal of Asian Development Studies*.
<https://doi.org/10.62345/jads.2023.12.3.74>
- Ray, P. P. (2025). A review on vibe coding: Fundamentals, state-of-the-art, challenges and future directions.
- Ray, P. P. (2026). A review of TRiSM frameworks in artificial intelligence systems: Fundamentals, taxonomy, use cases, key challenges and future directions. *Expert Systems*, 43(3), e70213.

Saidur, M. J. I., & Khan, M. K. (2023). Automating NIST 800-53 control implementation: A cross-sector review of enterprise security toolkits. *ASRC Procedia: Global Perspectives in Science and Scholarship*, 3(1), 160-195.

Scamardo, G. (2026). *AI Forensics: An Experimental Analysis of Transparency, Security, and Accountability in Automated Decision-Making Systems* (Doctoral dissertation, Politecnico di Torino).

Shahriar, A., Rahman, M. N., Ahmed, S., Sadeque, F., & Parvez, M. R. (2025). A survey on agentic security: Applications, threats and defenses. *arXiv preprint arXiv:2510.06445*.

Söderlund, K., & Larsson, S. (2024). Enforcement design patterns in EU law: An analysis of the AI Act. *Digital Society*, 3, Article 41. <https://doi.org/10.1007/s44206-024-00129-8>

Subramani, V. (2023). Governance led security architecture in large scale enterprise systems. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, 6(4), 9037-9045.

Tabassi, E. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.AI.100-1>

Watson, N., Amer, A., Harris, E., Ravindra, P., & Zhang, S. (2025). Personalized constitutionally-aligned agentic superego: Secure ai behavior aligned to diverse human values. *Information*, 16(8), 651.

Zhang, Y., Wang, X., Hu, M., & Zhang, Z. (2026). Bridging dual knowledge graphs for multi-hop question answering in construction safety. *Automation in Construction*, 183, Article 106794. <https://doi.org/10.1016/j.autcon.2026.106794>