RAINBIRD

Deterministic Graph-Based Inference for Guardrailing Large Language Models

An Approach to Compliance and Control in Financial AI

March 19th 2025

Contents

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

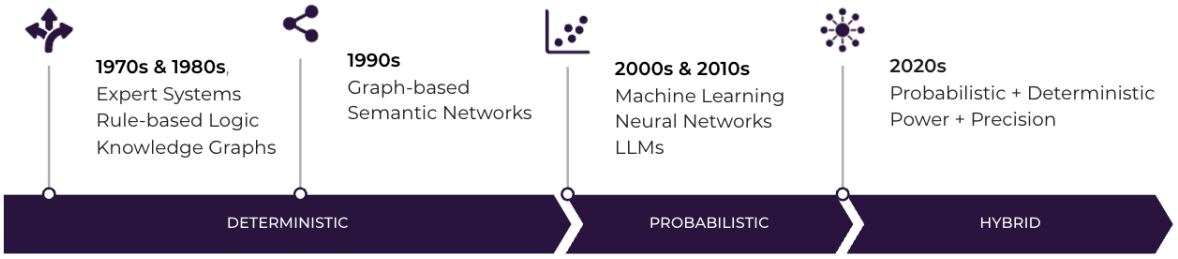
Introduction

The evolution of AI reasoning systems has swung across both probabilistic and deterministic paradigms. Early AI in the 1970s and 1980s was predominantly deterministic, relying on expert systems with explicit rule-based approaches to model human expertise. These systems evolved from simple if-then rules to more sophisticated knowledge representations including semantic networks and frames. By the 1990s, graph-based approaches emerged as a powerful way to represent more complex relationships between concepts.

The 2000s and 2010s witnessed a dramatic shift toward probabilistic models. Symbolic AI was replaced with sub-symbolic methods as statistical machine learning and neural networks gained prominence. These approaches sacrificed explainability for performance, leading to increasingly powerful but opaque systems.

The rise of Large Language Models (LLMs) marks the zenith of this probabilistic era, with remarkable word prediction capabilities but also significant limitations in consistency and precision.

We are now entering a third phase—the hybrid era—where organisations are combining the strengths of both paradigms. This new approach marries the flexibility and generative power of probabilistic models with the precision and explainability of deterministic systems. In regulated domains like financial services, this hybrid approach offers a compelling solution: leveraging LLMs for their linguistic capabilities while using deterministic graph-based inference to ensure compliance, consistency, and transparency.



LLMs are rapidly being adopted in financial services for tasks ranging from customer service chatbots to assisting with compliance reporting. Banks and fintech firms see the promise of AI in boosting efficiency, improving customer experience, and automating complex regulatory analysis. However, deploying LLMs in a highly regulated domain comes with significant challenges. By design, LLM outputs are probabilistic and non-deterministic -- the same query can yield different answers on different occasions -- which is problematic in high-stakes environments that demand consistency and accuracy. Moreover, these models often act as "black boxes," making it hard to explain why a particular output was generated.

A notable concern is the tendency of LLMs to hallucinate -- confidently producing false or fabricated information. In finance, even a single hallucinated detail (for example, an incorrect interpretation of a regulation or a fabricated compliance rule) can erode trust and cause legal trouble.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

All machine learning is based on the principle of finding correlations in data based on historic training. It predicts answers based on a “balance of probabilities” with no understanding of logic or causality.

Increasingly, developers are exploring an agentic configuration, where multiple LLM-powered agents take on different parts of a process, often challenging each other — usually following a Chain of Thought that itself is LLM generated. This approach can only “simulate” causal reasoning without the benefit of actually being causal.

Recognising these issues, financial institutions and regulators have begun advocating for stronger AI guardrails. The UK's Financial Conduct Authority (FCA), for instance, emphasises principles of fairness, accountability, and transparency in AI use, even as it maps existing rules to new AI contexts.

In this context, deterministic graph-based inference has emerged as a successful solution to guardrail LLM behavior. This approach uses a model of the domain built from first principles, leveraging knowledge graphs with explicit rules and causal relationships to verify or guide the outputs of an LLM. A graph-based inference engine produces decisions that are traceable, explainable, and consistent. This methodology enables organisations to encode legal and regulatory expertise into deterministic systems, creating a reliable framework where any advice or answer generated adheres to specific rules and regulations embedded in the knowledge graph.

In the following sections, we explore the risks LLMs pose in financial services, and how a deterministic inference layer can mitigate those risks — enabling the benefits of LLMs without compromising on compliance and reliability.

While other approaches to LLM guardrailing exist — including fine-tuning, retrieval-augmented generation (RAG), and prompt engineering techniques — none of these are precise and deterministic. This paper therefore focuses specifically on deterministic graph-based inference as a robust and lasting solution for financial services contexts. The emphasis on deterministic inference is motivated by the particular needs of financial institutions for determinism, explainability, consistency, and regulatory compliance. These are innate limitations of LLMs, and attributes that are lacking in other approaches that rely on LLMs to generate the final answer to a query.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Risks of LLMs in Financial Services

LLMs bring powerful language understanding capabilities, but in financial services they also introduce new risks.

Lack of Determinism: LLMs do not guarantee the same output for identical inputs every time. This non-reproducibility makes it hard to rely on them in compliance workflows that require consistent, testable results. An AI assistant might answer a client's regulatory query correctly one day and incorrectly the next, undermining predictable service. Such variability complicates auditing and violates the determinism expected in finance IT systems.

Hallucinations & False Information: Generative models can produce information that seems plausible but is entirely false. These hallucinations are especially dangerous in domains requiring factual accuracy like finance. An LLM-powered chatbot might *invent* a clause in an FCA regulation or cite a non-existent policy during a compliance check. In a financial context, a hallucinated compliance rule or misinterpreted guideline could lead to wrong decisions and regulatory violations.

Bias & Inconsistencies: LLMs learn from historical data, which typically embed societal or institutional biases. In finance, this can translate to discriminatory or inconsistent outcomes if not checked. Such bias isn't just unethical -- it can breach laws (e.g. the Equality Act 2010 in the UK) and lead to severe penalties.

Regulatory Failures: When LLMs are used for regulated tasks (like generating financial advice, performing AML checks, or preparing reports), errors can directly result in non-compliance. Unlike a controlled software system, an LLM might interpret rules loosely or overlook an important detail. Incorrect advice or mis-reporting to regulators could trigger enforcement actions.

Operational & Legal Risks: LLM-based systems can also introduce operational uncertainties—data privacy issues, lack of audit trail, and integration flaws. They often require large volumes of data (raising data protection concerns), and if that data includes sensitive financial information, improper handling could lead to breaches of GDPR or similar laws that come with big fines. Moreover, because LLMs lack explainability, it is hard to audit their decisions. This opacity is problematic in finance, where firms must document decision-making processes for internal risk management and regulators.

Prompt Injection Attacks: LLMs are vulnerable to specially crafted inputs designed to manipulate their behavior in unintended ways. These "prompt injection attacks" can bypass built-in safety measures and guardrails. In financial services, this presents a significant risk where malicious actors could trick an LLM into providing unauthorised financial advice, revealing confidential information, or bypassing compliance checks entirely. For example, an attacker might structure a query that instructs the LLM to "forget previous rules" and then proceed to ask for guidance on circumventing AML regulations. Unlike traditional software vulnerabilities that can be patched, prompt injection vulnerabilities are inherent to how LLMs function, making them particularly challenging to mitigate through training alone. Financial institutions deploying LLMs without appropriate guardrails could inadvertently create a new attack surface for regulatory breaches and security incidents.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

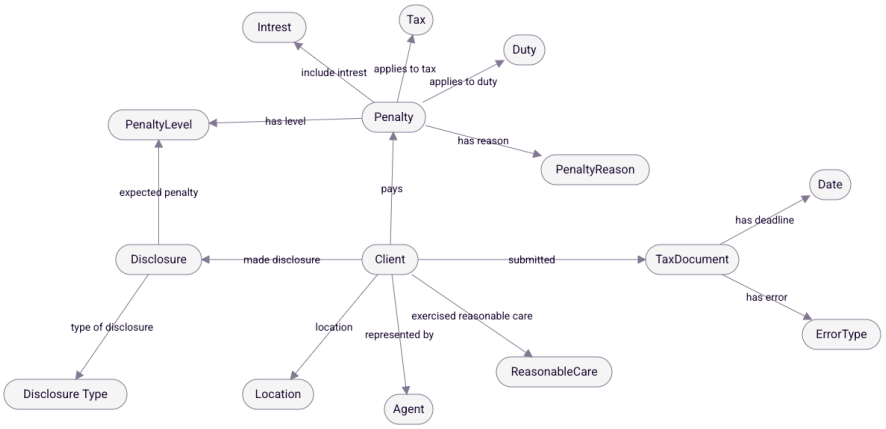
The Solution: Deterministic Graph-Based Inference

To mitigate the above risks, deterministic graph-based inference can be layered with LLMs as a form of AI guardrail. This approach combines the strengths of rule-based expert systems with the flexibility of LLMs, yielding an AI that is smart, accountable and safe.

What is Deterministic Graph-Based Inference?

Deterministic graph-based inference refers to AI systems that derive conclusions via fixed logical rules encoded in a graph or network structure. Instead of relying on statistical prediction, these systems reason over a knowledge graph of facts and rules. Because the rules are explicitly programmed or learned from authoritative sources, the inference is deterministic: given the same inputs, the system will always produce the same output by following the logical connections, regardless of complexity.

This deterministic approach allows domain experts to encode decision logic (like regulatory criteria, compliance checklists, product eligibility rules, etc.) into a graph-based knowledge model. The resulting framework creates a structured representation of complex regulatory requirements and business rules that can be systematically applied.



A small graph that derives reasons for tax penalties.

The symbolic inference engine then uses this model to answer questions through logical deduction. Importantly, every step in the reasoning chain is recorded — you can trace exactly which rule led to which conclusion. This is fundamentally different from how an LLM operates. An LLM generates an answer by predicting likely sequences of words based on patterns in training data, which can introduce bias, randomness and error. A graph-based system generates an answer by traversing a network of known truths and rules. The result is an explainable outcome backed by a clear chain of reasoning.

In summary, deterministic inference systems follow explicit rules rather than probabilistic patterns. They ensure compliance by design because the regulatory framework is built into the logic. This guarantees that any decision or recommendation the system makes is grounded in the rules that humans have vetted, rather than arising from an opaque machine learning process.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

How It Works with LLMs

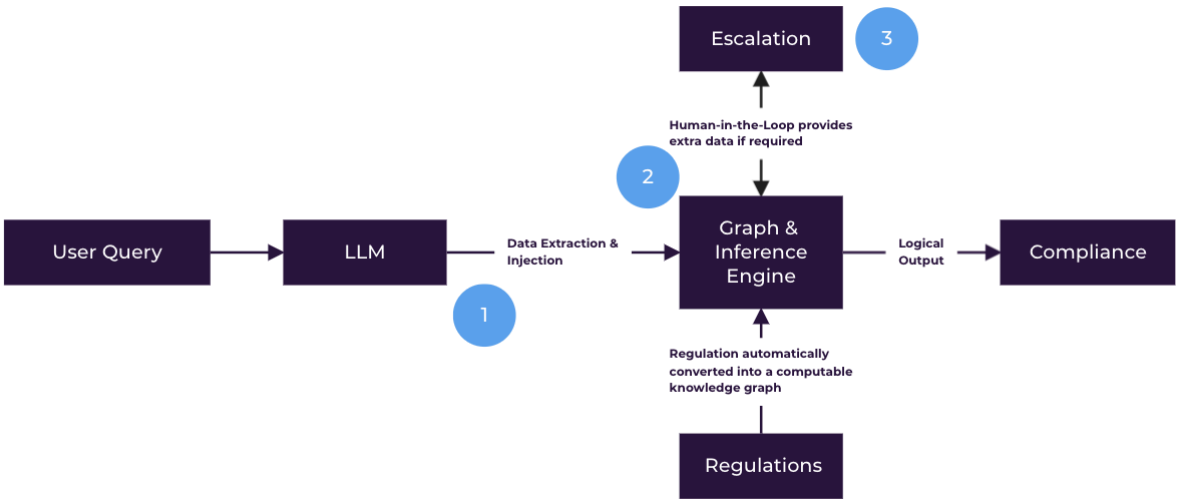
LLMs and deterministic graph-based inference systems can be integrated through two primary architectural patterns, each serving different use cases and offering distinct advantages. Below, we explore these patterns in detail.

Pattern 1: Graph-First Reasoning

Definition: The deterministic inference engine serves as the primary decision maker while the LLM acts as an interface layer, handling natural language understanding and communication. Critical decisions are made exclusively by the symbolic inference engine that traverses the knowledge graph.

Current implementations typically employ a two-stage architecture where LLMs handles natural language understanding while a separate deterministic layer manages reasoning over explicit decision logic in a graph. In this configuration, the LLM is not used as a proxy for reasoning. Instead, a deterministic graph serves as a structured knowledge source for inference using a causal reasoning engine.

When queries arrive, they are pre-processed by the LLM which understands and extracts context into the graph. Key assertions, recommendations, or decisions are then directly derived from the knowledge graph only, creating a synergistic system where the LLM provides flexibility and natural interaction, while the graph logically calculates accurate and compliant answers.



1. **Data Extraction & Injection:** The LLM processes the user query, extracts relevant information, and injects this data into the graph-based inference system.

2. **Causal Reasoning:** The deterministic inference engine processes the data, using the knowledge graph created from regulations to determine a logical, compliant output.

3. **Human in the loop:** When the system needs additional information that wasn't provided in the initial query or is not available from data sources, a human can intervene to supply any missing data or context.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

This architecture operates on a "deny all" principle, limiting answers to what can be generated safely from the graph. It is a "belt and braces" approach to guaranteeing precise and deterministic outputs and the total elimination of hallucinations, but one that is limited by the scope and complexity of the graph that can be produced.

While graph building was historically challenging due to the human burden of encoding all of the required knowledge into graphs — a discipline known as knowledge engineering — fine tuned LLMs are now adept at knowledge-engineering and can convert both explicit and tacit knowledge sources into sophisticated deterministic graphs. While a lack of precision, determinism and causal explainability are innate limitations of LLMs that cannot be overcome, the more powerful LLMs get over time, the larger and more complex the graphs they can create programmatically. Such an approach enables organisations to create enduring IP which acts as an effective and differentiated knowledge layer, and safely leverage LLMs as they evolve.

For exceptionally high-stakes use cases where you need to reason over structured data, financial institutions also have the option to bypass LLMs entirely, running pure deterministic inference directly over knowledge graphs that contain all of the expertise required to make a determination.

Graph based reasoning is extremely energy efficient and low latency. Modern graph inference engines are capable of reasoning over truly massive graphs — even extending to graphs with millions of nodes — this provides a promising deterministic future for computing over highly structured regulation in high-stakes scenarios.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

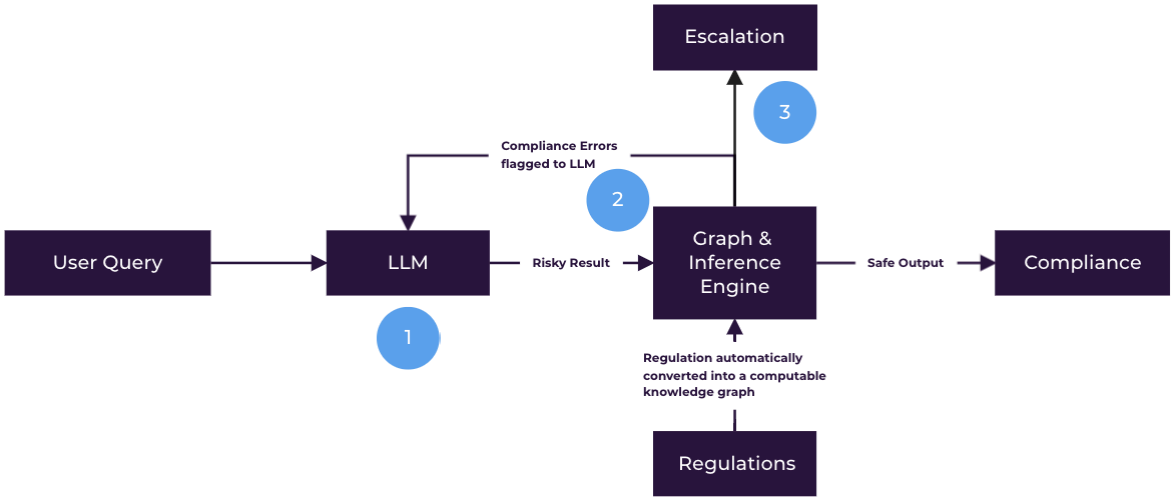
Pattern 2: Post-Generation Validation

Definition: The LLM generates complete responses that are subsequently verified and potentially corrected by the symbolic inference engine before being delivered to the user. The graph acts as a compliance checkpoint.

An alternative architecture for harnessing LLMs — while avoiding their pitfalls — is to enable an LLM to play a much greater role, generating provisional outcomes, but use a deterministic inference engine for verification.

In such a hybrid setup, the LLM handles what it's best at — understanding natural language, drafting fluent responses, and parsing broad queries, and the graph-based system does what it is best at — acting as a powerful guardrail ensuring factual and logical correctness against a set of rules.

While variations exist, a common workflow looks like this:



1. **LLM Drafts a Response:** The LLM receives a user's query or a task and produces an initial answer. For example, a customer might ask, "Can I open a stocks & shares ISA if I moved to the UK 3 months ago?" The LLM will generate a helpful-sounding answer based on its knowledge. Without oversight, however, there is a chance this answer could be wrong or incomplete (e.g. the LLM might *guess* the rules about residency requirements for ISAs).

2. **Deterministic Check or Augmentation:** Before the answer reaches the end-user, it is routed through the deterministic inference engine. The engine has encoded, say, the precise eligibility rules for a stocks & shares ISA as per HMRC/FCA guidelines. It will check the LLM's answer against these rules. If the LLM's answer said "Yes, you can open an ISA," but the rules graph knows that an ISA requires 12 months of UK residency (hypothetically), the engine flags a violation.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

It can either block that answer or correct it. In a guardrail mode, the knowledge graph might output a verdict like: "According to Rule 5.4, the client is not eligible due to residency under 12 months." The system can then prompt the LLM to revise its answer to include this correct information, or the deterministic system can directly provide the correct answer back to the user.

3. **Human or System Intervention:** If the deterministic engine flags that it ultimately cannot validate the LLM's response, or if the query falls outside known rules, it can escalate. For instance, it might ask for a human compliance officer to review, or it might instruct the LLM to say, "I'm sorry, I cannot provide that information." This ensures that when the AI doesn't *know* according to the rule set, it doesn't improvise dangerously.

This kind of feedback loop marries free-form generative ability with strict rule adherence.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Comparison: When to Use Each Pattern

Aspect	Graph-First Reasoning	Post-Generation Validation
Core Purpose	Precise and deterministic decision-making	Enable creative LLM outputs but with deterministic safety checks
Best For	High-stakes decisions, strict compliance requirements - scenarios requiring precision, determinism and explainability	Advisory content, narrative explanations, creative communications with regulated boundaries - scenarios where LLM flexibility adds value but needs appropriate safeguards
Process	LLM (optionally) structures data > graph and inference engine makes decisions > LLM (optionally) summarises results and rationale	LLM generates complete response > graph verifies for compliance > corrections applied if needed before delivery
Risk Control	Prevents errors by design (LLM cannot make key decisions)	Catches errors after generation (safety net approach)
Hallucination Management	Eliminates hallucinations entirely on critical decisions	Detects and corrects anticipated hallucinations before delivery
Implementation Complexity	Moderate - relies on a structured knowledge base built from documentation and expertise	Moderate - requires the creation of a robust verification rules to trap anticipated errors, and a correction mechanism
Response Flexibility	Narrowly constrained to what is modelled in the graph	More flexible and natural-sounding but influenced by public training data
Example Use Cases	Transaction screening, eligibility assessments, risk scoring, regulatory reporting, limit monitoring, compliance verdicts, automated approvals	Financial advice, complaint responses, investment research, product explanations, client communications
Trade-offs	Prioritises maximum safety while maintaining essential flexibility.	Balances LLM-generated creativity with compliance verification

For both patterns, successful implementation depends on the quality of the knowledge graph and the appropriate division of responsibilities between the LLM and deterministic components. The choice between patterns should be driven by:

Risk profile of the use case Complexity of regulatory requirements Need for natural language flexibility Performance requirements Explainability needs

Some implementations adopt a hybrid approach, using Graph-First Reasoning for high-risk decisions and Post-Generation Validation for supplementary information or explanations.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Benefits

Integrating deterministic inference guardrails with LLMs offers several compelling benefits for financial institutions:

Transparency & Auditability: Every decision made by the graph-based system can be traced back to a specific rule or data point. Unlike a pure LLM solution, a deterministic system can provide a clear rationale. A well-designed inference engine can output an evidence trail or decision tree showing a snapshot of data used, and how it applied various rules to reach an outcome. This level of explainability is crucial for audits and regulatory inspections. Graph-based reasoning enables "true causal reasoning from first principles," meaning the system can explain not just what decision was made but how and why. This satisfies the growing demand for AI explainability in finance.

Reduction in False or Non-Compliant Information: By design, a deterministic guardrail prevents many errors before they happen. Hallucinations and off-policy responses are caught by the rules filter. The LLM is essentially kept "in check" by the knowledge graph. It cannot easily introduce completely novel claims that aren't vetted, because any critical claims are verified against the graph. The knowledge graph serves as a grounding mechanism — the LLM's creativity is bounded by factual reference points. The outcome is much more reliable, and users (and regulators) can have greater confidence in what the AI says. In high-stakes use cases (like sanction screening or regulatory filings), this approach can prevent costly mistakes.

Regulatory Compliance Assurance: With regulatory knowledge encoded, the system actively enforces compliance. It acts like an automated compliance officer living inside the AI. Any advice or action the AI proposes is cross-checked with applicable laws and guidelines. This shifts the AI governance approach from reactive to proactive. Rather than finding out after the fact that an AI gave bad advice, the system is built to avoid the bad advice in the first place. Every answer is compliant by construction. This approach is likely to be viewed favorably by regulators, who have signaled the importance of fairness and accountability in AI.

Consistency and Reliability: Deterministic systems eliminate the randomness inherent in LLMs. This means customers and employees get consistent answers to the same questions, every single time. It builds confidence in the AI's reliability. Over time, the institution can develop a library of approved rules and responses, creating a stable knowledge base. This consistency also helps in training and change management —- staff know that the AI's behavior will be stable and won't produce surprises.

In short, deterministic graph-based inference brings precision, trust, and control to AI deployments in finance. It complements the generative and interpretive power of LLMs with a safety net of hard facts and rules. The result is an AI that not only automates and accelerates work, but does so in a way that is provably compliant and explainable.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Implementation

Introducing Rainbird

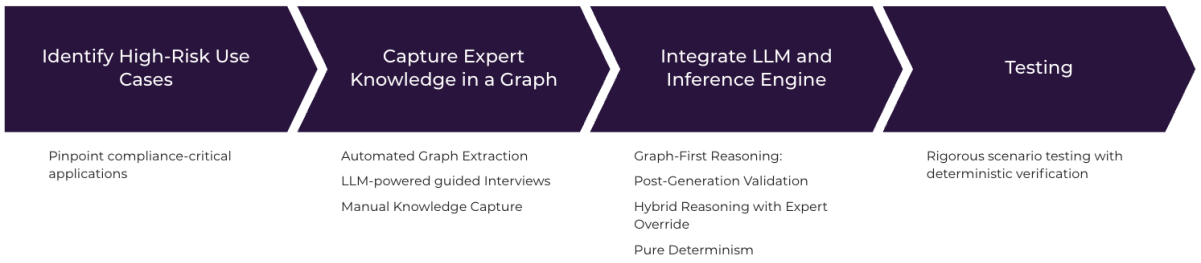
Rainbird stands as a leading implementation of the deterministic graph-based inference principles discussed in this paper.

Purpose-built for enterprise-grade knowledge graph reasoning, Rainbird's platform enables financial institutions to transform complex regulatory frameworks into executable, deterministic systems at scale. Major banks and financial services firms have deployed Rainbird to create robust guardrails around their LLM implementations, addressing the critical compliance challenges outlined earlier.

By encoding regulatory expertise into verifiable knowledge graphs, Rainbird ensures that AI-generated content and decisions remain fully compliant with intricate financial regulations while providing the complete traceability and explainability demanded by regulators and stakeholders. The platform's ability to rapidly build and reason over knowledge graphs makes it particularly well-suited for high-stakes financial applications where deterministic outcomes are non-negotiable.

Key Considerations

For financial firms looking to adopt a hybrid LLM + deterministic inference approach, there are several key considerations and steps to ensure a successful implementation.



For financial institutions looking to implement deterministic guardrails using technologies like Rainbird, a structured approach is essential.

Appendix A provides a detailed implementation framework covering the identification of high-risk use cases, knowledge capture methodologies, integration patterns, and testing strategies. These practical steps offer a blueprint for organisations seeking to operationalise the concepts discussed in this paper.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Conclusion

As financial institutions increasingly adopt LLMs to enhance customer experiences and operational efficiency, the intersection of probabilistic AI and regulatory compliance presents both opportunity and risk. Deterministic graph-based inference offers a robust solution that bridges this gap, providing the best of both worlds: the linguistic fluency and generative capabilities of LLMs paired with the precision, consistency, and explainability of rule-based systems.

The hybrid architecture described in this paper — whether implemented as graph-first reasoning, post-generation validation, or pure determinism — creates AI systems that can operate safely within highly regulated environments. These systems not only enhance compliance by design but also provide the transparency and auditability that regulators and stakeholders demand. By encoding regulatory frameworks and domain expertise into knowledge graphs, financial institutions can maintain control over AI outputs while still benefiting from the power of generative models.

Looking ahead, as LLMs grow more sophisticated, they will increasingly assist in the automated creation of comprehensive knowledge graphs, making this approach even more scalable and efficient. The future of AI in financial services will not be determined by choosing between probabilistic or deterministic approaches, but rather by thoughtfully integrating them to create systems that are both powerful and trustworthy.

By implementing deterministic guardrails around LLMs, financial institutions can move from merely exploring AI to confidently deploying it across their operations — knowing that every interaction remains firmly within the bounds of regulatory compliance and best practices. This approach transforms AI from a potential compliance risk into a literal compliance asset, one that can adapt to evolving regulations while maintaining the consistency and explainability that define responsible innovation in financial services.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Appendix A: Implementation Framework

Identify High-Risk Use Cases

Start by pinpointing where an LLM's output could pose compliance or accuracy risks. This typically includes any application that needs to provide complex reasoning, give regulatory advice, product recommendations, eligibility decisions, or generate reports for regulators. These are the areas most in need of guardrails. For each such use case, define clearly what rules or policies must be adhered to (e.g., "ensure investment advice complies with FCA suitability requirements" or "ensure all AML checks for transaction monitoring follow the defined procedure").

Capture Expert Knowledge in a Graph

Next, create a knowledge graph that encodes relevant rules. Rainbird offers two powerful approaches for this crucial step:

- **Manual Knowledge Capture:** Work with subject-matter experts (compliance officers, legal, risk managers) to encode their expertise into a structured knowledge graph. This traditional approach involves collaboration sessions where domain experts articulate rules, decision logic, and policies that are then formalised in the Rainbird platform. The resulting graph becomes a digital representation of the organisation's compliance framework—capturing complex regulatory nuances and internal guidelines in a machine-readable format. Significant LLM-powered tooling is now available to interview experts and structure that knowledge as draft graphs, accelerating the knowledge engineering process.

- **Automated Graph Extraction with Rainbird Noesis:** For organisations with extensive written regulations or documentation, Rainbird Noesis provides a revolutionary alternative. This specialised LLM can automatically extract and structure knowledge from regulatory documents, policies, and procedure manuals. It transforms dense regulatory text into verifiable knowledge graphs with minimal human intervention. Noesis identifies key concepts, relationships, conditions, and exceptions within lengthy documents, dramatically accelerating the knowledge engineering process. The system maintains traceability back to the source material, ensuring that each node and connection in the graph can be validated against the original documentation.

This dual approach gives organisations flexibility — leveraging both human expertise and existing document repositories to build comprehensive knowledge graphs — essentially codifying the firm's compliance handbook into a machine-readable form. This step also involves deciding the granularity: what will the graph cover and what will be left to the LLM? As a rule of thumb, the graph should cover anything that has a clear "right or wrong" per regulations, and the LLM can handle more open-ended conversation around it.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Integrate LLM and Inference Engine

Rainbird offers several sophisticated integration patterns to ensure the system architecture is designed such that the LLM and the deterministic engine communicate effectively:

- **Graph-First Reasoning:** Rainbird first determines the core decision (e.g. eligibility, required actions, compliance verdict) by traversing its knowledge graph with the relevant case facts which may have been extracted from unstructured sources aided by an LLM in the context of the graph. This authoritative determination — complete with reasoning chain — can then optionally be passed back to an LLM, which wraps the decision in natural language appropriate for the audience and context. The LLM can explain the decision in user-friendly terms without altering the underlying determination. This ensures the critical decision comes from the deterministic system while leveraging the LLM for communication.

- **Post-Generation Validation:** In this pattern, the LLM generates a complete response which is then passed to the Rainbird inference engine for verification before delivery to the user. Rainbird analyses the output for regulatory compliance, factual accuracy, and adherence to internal policies. When violations are detected, Rainbird doesn't just flag problems—it actively provides the correct information, allowing either the LLM to regenerate its response using this expert guidance or highlighting problematic sections for human review. For complex queries spanning multiple regulatory domains, Rainbird can decompose the question, verify different components separately, and synthesise a comprehensive response, creating a robust system with multiple layers of verification and correction. This approach creates a compliance checkpoint that catches errors after generation but before they reach users.

For all these patterns, the technical implementation typically involves secure API integration between the LLM and Rainbird platforms, with appropriate middleware to orchestrate the communication flow. The choice of pattern depends on the specific use case, risk profile, and performance requirements. Many financial institutions implement multiple patterns, applying stricter controls to high-risk functions while using lighter verification for lower-risk interactions.

Testing

Before going live, the combined system should be rigorously tested with a wide range of scenarios — both expected and edge cases. This means not only testing normal queries, but also strange or adversarial inputs to see how the system handles them. Because we have a deterministic component, we can create tests like: "For input X, the system *must* respond with Y (or contain Z in the answer)." This is much stronger verification than what's possible with an LLM alone. Simulate scenarios of compliance importance: e.g. a user asking something that should trigger an AML alert, a user trying to get unauthorised advice, or an employee query that requires absolute factual accuracy. Ensure the inference engine consistently catches any mistakes. If any slip through in testing, refine the rules or integration logic accordingly.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

It is also important to plan for how humans will interact with the system in exceptional cases. No matter how comprehensive the rules in a graph, there will be queries or situations that fall outside the automated knowledge. Determine triggers for escalation to a human expert. For example, if a query requires interpretation of a very new regulation that isn't in the graph yet, the system might respond with a polite deferral or route it to a human compliance officer. By defining these handoff points, you ensure that the AI does not operate beyond its competence. This also provides a safety valve: if the deterministic system ever encounters a conflict or ambiguity, it can flag a person to make the judgement call, thereby maintaining compliance.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Appendix 2: Glossary

Agentic Configuration: A system design where multiple LLM-powered agents take on different parts of a process, often challenging each other, usually following a Chain of Thought that itself is LLM-generated.

Causal Reasoning: A logical process of identifying the relationship between cause and effect, understanding why things happen rather than just observing correlations and predicting text.

Chain of Thought: A probabilistic technique used by LLMs where the model generates step-by-step processes that simulate logical reasoning, but remain non-deterministic and may produce different outcomes given the same input.

Deterministic System: A system where given the same input, it will always produce the same output, with no randomness or variability in its processing or decision-making.

Deterministic Graph-Based Inference: AI systems that derive conclusions via fixed logical rules encoded in a graph or network structure, ensuring consistent outputs for identical inputs by following logical connections.

Expert Systems: Early AI applications from the 1970s-1980s that captured human expertise in rule-based systems to solve complex problems in specific domains.

Fine-tuning: The process of further training a pre-trained language model on a specific dataset to adapt it to particular tasks or domains. For example, Rainbird Noesis is fine-tuned to create knowledge graphs from documentation.

Graph-First Reasoning: An architectural pattern where a deterministic inference engine serves as the primary decision maker while an LLM acts as an interface layer, handling natural language understanding and communication.

Guardrails: Protective constraints placed around AI systems to prevent unwanted behaviors, ensure safety, and maintain compliance with regulations and ethical standards.

Hallucinations: An innate limitation of LLMs where they confidently generate information that is entirely fabricated, false, or not supported by their training data or the real world. This is a structural limitation of probabilistic systems that predict content based on patterns rather than causal understanding.

Hybrid Era: The current phase of AI development that combines probabilistic models (like LLMs) with deterministic systems to leverage the strengths of both approaches.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public

Knowledge Engineering: The discipline of building knowledge-based systems, involving the development of knowledge graphs, rule sets, and frameworks that encode expert knowledge.

Knowledge Graph: A structured representation of knowledge as a network of entities, concepts, and the relationships between them, used to organise information in a way that machines can process.

Large Language Models (LLMs): Advanced AI systems trained on vast amounts of text data that can generate human-like text and perform various language-related tasks.

Post-Generation Validation: An architectural pattern where the LLM generates complete responses that are subsequently verified and potentially corrected by a symbolic inference engine before being delivered to the user.

Probabilistic Models: AI systems that generate outputs based on statistical patterns and probabilities learned from training data, introducing an element of randomness or variability.

Prompt Injection Attacks: Security vulnerabilities where specially crafted inputs manipulate an LLM's behavior in unintended ways, potentially bypassing safety measures and guardrails.

Rainbird: A platform for enterprise-grade knowledge graph reasoning that enables the transformation of complex regulatory frameworks into executable, deterministic systems.

Rainbird Noesis: A specialised LLM that can automatically extract and structure knowledge from regulatory documents, policies, and procedure manuals, transforming dense regulatory text into verifiable knowledge graphs.

Retrieval-Augmented Generation (RAG): A non-deterministic technique that enhances LLM outputs by retrieving relevant information from external knowledge sources before generating a response.

Sub-symbolic Methods: AI approaches (like neural networks) that don't use explicit symbolic representations but instead work with distributed representations or patterns across nodes.

Symbolic AI: AI systems that use explicit symbols and rules to represent knowledge and perform reasoning, as opposed to statistical or neural approaches.

Symbolic Inference Engine: A software component that applies logical rules to derive conclusions from a knowledge base, typically using techniques like forward and/or backward chaining.

©2025 Rainbird Technologies Limited. All Rights Reserved. Public