FinRegAgents: A Multi-Agent RAG Framework for AI-Assisted Financial Regulatory Audits with Confidence-Aware Validation

endvater

Independent Researcher

GitHub: github.com/endvater/finreg-agents

February 2026

Abstract

*We present **FinRegAgents**, an open-source multi-agent framework that simulates supervisory-style financial regulatory audits using retrieval-augmented generation (RAG). The system evaluates institutional compliance against four major European regulatory frameworks—GwG (Anti-Money Laundering), DORA (Digital Operational Resilience Act), MaRisk (Minimum Requirements for Risk Management), and WpHG/MaComp (Securities Trading Act)—across 94 audit fields organized in declarative JSON catalogs. Our key contribution is a **confidence-aware validation architecture** that addresses the critical gap between RAG retrieval and regulatory assertion: a retrieval quality gate prevents hallucination on thin evidence, a composite confidence score (retrieval relevance, evidence coverage, type matching, LLM self-assessment) quantifies assessment reliability, and structural validation detects phantom citations, placeholder artifacts, and logical inconsistencies before findings enter the audit report. We formalize the **coverage–verification gap**—the systematic discrepancy between a system's ability to produce assessments and its ability to verify them—drawing on parallels to ad-tech fraud detection. We describe the architecture, the confidence scoring framework, the validation pipeline, and report generation across three output formats. FinRegAgents is released under Apache 2.0.*

Keywords: retrieval-augmented generation, regulatory technology, multi-agent systems, confidence scoring, financial supervision, compliance automation, hallucination detection

---

1. Introduction

Financial regulatory audits represent one of the highest-stakes applications for AI-assisted document analysis. When a supervisory authority such as BaFin (Germany's Federal Financial Supervisory Authority) conducts a special audit (*Sonderprüfung*), the resulting findings carry legal consequences ranging from remediation orders to license revocations. This stakes profile creates an unusual constraint for RAG systems: the cost of a false positive (asserting non-compliance without evidence) and a false negative (missing genuine non-compliance) are both severe, yet qualitatively different.

Current RAG applications in legal and regulatory domains [1, 2, 3] have demonstrated strong information retrieval capabilities but typically lack mechanisms to distinguish between confident assessments backed by strong evidence and speculative assessments generated from marginally relevant context. In production regulatory settings, this distinction is critical: an audit finding without adequate evidentiary support is not merely unhelpful—it is professionally damaging.

We identify three interrelated problems that FinRegAgents addresses:

P1: The Hallucination-in-Compliance Problem. Standard RAG pipelines pass all retrieved chunks to the LLM regardless of relevance. When retrieval quality is low (all chunks score below a meaningful threshold), the LLM generates plausible but ungrounded assessments. In regulatory contexts, this produces audit findings that cite documents the system has not actually read closely, or worse, fabricates legal references.

P2: The Confidence Opacity Problem. Existing systems produce categorical assessments (compliant / non-compliant) without quantifying reliability. A human reviewer receives the same presentation for a finding backed by three relevant policy documents and one backed by a tangentially related spreadsheet. This opacity undermines the human-in-the-loop workflows that regulatory audits require.

P3: The Coverage–Verification Gap. Increasing the number of audit fields a system can evaluate (coverage) does not proportionally increase the system's ability to verify whether its evaluations are correct (verification). This gap, analogous to impression fraud in digital advertising [4], means that scaling regulatory AI without proportional verification investment creates systemic risk.

FinRegAgents addresses these problems through a layered architecture: declarative audit catalogs define the regulatory scope, multi-modal document ingestion builds the evidence base, specialized agents evaluate each audit field with confidence-aware validation, and a report generator produces findings that transparently distinguish high-confidence assessments from uncertain ones requiring manual review.

2. Related Work

2.1 RAG in Legal and Regulatory Domains

Retrieval-augmented generation for legal applications has been explored across contract analysis [5], case law retrieval [6], and regulatory compliance checking [7]. Savelka et al. [8] demonstrated that chunking strategies significantly affect legal RAG quality, finding that larger chunk sizes better preserve the argumentative structure of legal texts—a finding consistent with our choice of 1024-token chunks for regulatory documents. Louis et al. [9] identified hallucinated legal citations as a critical failure mode, with LLMs generating plausible but non-existent case numbers. Our structural validation layer directly addresses this class of error.

2.2 Confidence and Calibration in LLM Systems

LLM calibration research has shown that language models are systematically overconfident in their self-assessments [10, 11]. Kadavath et al. [12] demonstrated that while models can be prompted to express uncertainty, their verbalized confidence correlates poorly with actual accuracy. This finding motivates our composite confidence score, which weights LLM self-assessment at only 20% and derives 80% from observable signals (retrieval scores, evidence coverage, type matching).

2.3 RegTech and Automated Compliance

The RegTech literature distinguishes between rule-based compliance checking [13] and AI-assisted interpretation [14]. Arner et al. [15] survey the landscape, noting that most production systems operate on structured data (transaction monitoring, sanctions screening) rather than unstructured document analysis. FinRegAgents bridges this gap by operating on the unstructured evidence that feeds into regulatory examinations—policies, process documentation, interview transcripts, and system screenshots—rather than on structured transaction data.

2.4 Multi-Agent Architectures

Multi-agent LLM systems have been applied to research synthesis [16], software engineering [17], and debate-style reasoning [18]. Our architecture differs in that agents do not negotiate or debate; instead, each agent operates independently on a single audit field, with validation occurring as a post-processing step. This design reflects the regulatory audit structure where each finding must stand on its own evidentiary basis. The planned skeptic-agent extension (Section 7) would introduce adversarial review as an optional verification layer.

3. System Architecture

FinRegAgents implements a four-stage pipeline: (1) multi-modal document ingestion, (2) vector index construction, (3) per-field agent evaluation with validation, and (4) report generation. The architecture is designed around three principles: regulatory knowledge is declarative (stored in JSON catalogs, not in code), evidence assessment is transparent (every finding includes provenance and confidence), and failures are explicit (insufficient evidence produces "not auditable" rather than speculative findings).

3.1 Declarative Audit Catalogs

Each supported regulation is encoded as a JSON catalog containing sections (*Prüfsektionen*), each with audit fields (*Prüffelder*). An audit field specifies the regulatory question, expected evidence types, legal basis, severity classification, and a deficiency template for standardized reporting. Table 1 summarizes the four catalogs.

Regulation	Sections	Fields	Legal Basis
GwG (AML/CFT)	8	34	GwG, §25h KWG, BaFin AuA
DORA	5	18	DORA Art. 5–46, RTS
MaRisk	8	22	MaRisk AT/BT, §25a KWG
WpHG/MaComp	7	20	WpHG, MaComp, MAR, MiFID II

Table 1. *Regulatory audit catalogs. Each field includes: question, expected evidence, input types, assessment criteria, severity, and deficiency template.*

This declarative approach enables regulatory updates without code changes: when BaFin publishes a new circular or amends the GwG, only the JSON catalog requires modification. The audit logic, confidence scoring, and report generation remain unchanged.

3.2 Multi-Modal Document Ingestion

The ingestion layer handles five document types that correspond to real audit evidence: PDF documents (policies, process descriptions, prior audit reports), Excel/CSV files (transaction monitoring statistics, training records), structured interview transcripts (JSON/YAML questionnaires), system screenshots (KYC interfaces, alert dashboards), and system logs. Each document type receives specialized preprocessing. PDF and log files are chunked using sentence-aware splitting (chunk size 1024 tokens, overlap 128). Excel files are converted to text with aggregation statistics to preserve information beyond the first rows. Interview transcripts are parsed into question-answer pairs with cross-references to audit field IDs. Screenshots are indexed as placeholder documents flagged for human visual review—critically, the binary image data is *not* stored in the vector index to prevent memory bloat.

Deduplication operates via SHA-256 file hashing. If identical content appears in multiple directories (e.g., a policy document in both *pdfs/* and *logs/*), only the first occurrence is indexed.

3.3 Regulation-Specific Agent Prompting

Each audit is executed by a PrueferAgent configured for the specific regulation. Critically, the system prompt adapts to the regulatory context. A GwG audit receives an AML-specialist persona with relevant legal references (§25h KWG, AMLA Guidelines); a DORA audit receives an ICT-resilience persona referencing DORA Articles 5–46. This prevents the domain contamination observed in v1, where all four regulations were evaluated by an agent identifying as an AML specialist. The agent's retrieval query is optimized by concatenating the audit question, expected evidence keywords, and legal basis—a multi-signal query strategy shown to improve retrieval precision in domain-specific RAG [19].

4. Confidence-Aware Validation

The central contribution of FinRegAgents v2 is a three-layer validation architecture that intervenes between RAG retrieval and the final audit report. Each layer operates at a different abstraction level and cost point.

4.1 Layer 1: Retrieval Quality Gate

Before any LLM call, the system evaluates retrieval quality. For each audit field, the top-k (default: 8) chunks are retrieved from the vector index. If no chunk scores above a configurable threshold t_{min} (default: 0.35), the field is immediately classified as *nicht prüfbar* (not auditable) without invoking the LLM. This prevents the most common hallucination vector: the LLM generating a plausible assessment from irrelevant context.

The gate also serves as a cost optimization mechanism. With 34 audit fields in the GwG catalog, each requiring approximately 2,000 input tokens and 500 output tokens, the gate can save 15–40% of API costs in document sets with incomplete coverage.

4.2 Layer 2: Composite Confidence Score

Each finding receives a composite confidence score $C \in [0, 1]$ computed from four weighted signals:

$$C = 0.30 \cdot S_{\text{retrieval}} + 0.30 \cdot S_{\text{coverage}} + 0.20 \cdot S_{\text{type}} + 0.20 \cdot S_{\text{self}}$$

where $S_{\text{retrieval}}$ is the mean cosine similarity of retrieved chunks above t_{min}, S_{coverage} is the fraction of expected evidence terms matched in retrieved source filenames (fuzzy token-level matching), S_{type} is the overlap ratio between expected and retrieved document types (e.g., if the catalog specifies {pdf, interview} and only pdf was retrieved, $S_{\text{type}} = 0.5$), and S_{self} is the LLM's self-reported confidence (clamped to [0,1]).

The weight allocation reflects empirical observations from calibration research: LLM self-assessment (S_{self}) is systematically overconfident [12], motivating its lower weight (20%). Observable signals—retrieval score and evidence coverage—are weighted higher (30% each) as they provide ground-truth-adjacent measurements. Table 2 shows the escalation model.

Confidence Range	Action	Rationale
C < 0.40	Auto-reject: nicht prüfbar	Insufficient evidence for any assessment
0.40 ≤ C < 0.70	Finding marked: Review required	Assessment exists but needs human validation
C ≥ 0.70	Finding accepted into report	Sufficient evidence and consistency
>30% of section under review	Section-level escalation	Systemic evidence gaps in topic area

Table 2. *Confidence-based escalation model. Thresholds are configurable.*

4.3 Layer 3: Structural Validation

After the LLM produces a finding, five deterministic checks validate structural consistency at zero additional API cost:

Source cross-check: The agent's cited sources are compared against the metadata of actually retrieved chunks. Sources cited by the agent but absent from retrieval results are flagged as *phantom citations*—a reliable hallucination indicator. **Placeholder detection:** Unresolved template markers (e.g., *{paragraph}*) in the reasoning text indicate incomplete generation. **Assessment–evidence consistency:** A "compliant" rating without any cited text passages, or a "non-compliant" rating without a deficiency description, triggers a warning. **Positive–negative consistency:** A deficiency text accompanying a "compliant" rating is flagged as contradictory. **Legal reference validation:** Cited legal paragraphs are checked against known patterns per regulation

(e.g., §*n* GwG for AML, Art. *n* DORA for resilience).

Any structural validation warning automatically sets *review_required* = *true* on the finding, regardless of the confidence score. This ensures that even high-confidence findings with structural anomalies are flagged for human review.

5. The Coverage–Verification Gap

We formalize an observation from the system's development that has broader implications for regulatory AI. Define *coverage*(*n*) as the number of regulatory fields a system can evaluate given *n* audit fields in its catalog, and *verification*(*n*) as the number of evaluations the system can reliably confirm are correct. In naïve RAG systems, coverage(*n*) = *n* (the system always produces an assessment), while verification(*n*) ≈ 0 (no mechanism exists to validate any assessment).

This gap is structurally analogous to impression fraud in digital advertising [4], where ad-serving systems report billions of impressions (coverage) while the fraction confirmed as seen by real humans (verification) is systematically lower. The ad-tech industry developed three countermeasures: deterministic signal analysis, adversarial verification, and ground-truth calibration. FinRegAgents adapts all three:

Deterministic signals correspond to our retrieval quality gate and structural validation (Sections 4.1, 4.3). **Adversarial verification** corresponds to the planned skeptic-agent (Section 7), which reviews findings against the actual retrieved evidence. **Ground-truth calibration** corresponds to synthetic control fields—audit fields with known correct assessments embedded in the catalog to measure system accuracy on a per-run basis.

The key insight is that narrowing the coverage–verification gap requires investment proportional to coverage expansion. Adding 20 new audit fields without corresponding validation mechanisms does not improve audit quality—it degrades it by diluting the signal-to-noise ratio of the overall report.

6. Report Generation

FinRegAgents produces audit reports in three formats: machine-readable JSON (for API integration and downstream processing), Markdown (for version-controlled review workflows), and print-ready HTML (for presentation and PDF conversion). All three formats include the same information; the HTML format adds visual elements including confidence bars, color-coded severity indicators, and review markers.

6.1 Overall Assessment Logic

The overall assessment (*Gesamtbewertung*) applies a six-tier logic that explicitly incorporates evidence completeness. The critical design decision is that a high proportion of "not auditable" findings degrades the overall assessment, preventing the failure mode where 80% of fields cannot be evaluated but the remaining 20% are compliant, producing a misleading "COMPLIANT" overall result. Table 3 shows the mapping.

Condition	Overall Assessment
Any material deficiency (wesentlich)	SEVERE DEFICIENCIES
≥50% not auditable	INSUFFICIENT EVIDENCE
Deficiencies or ≥3 partial	DEFICIENCIES FOUND
≥30% not auditable	LIMITED RELIABILITY
Partial compliance findings	PARTIAL – REMEDIATION NEEDED
All compliant	COMPLIANT

Table 3. *Overall assessment logic with evidence-completeness awareness.*

6.2 Audit Trail

Every report includes an audit trail documenting the model version, catalog version, average confidence score, number of findings requiring review, and a timestamp. This enables reproducibility assessment: if the same document set produces different results with a different model version, the audit trail makes the cause immediately apparent. All dynamic content in the HTML report is escaped via *html.escape()* to prevent cross-site scripting vulnerabilities from document content propagating into the report.

7. Limitations and Future Work

FinRegAgents is an alpha-stage system with several known limitations that define the immediate research roadmap.

No empirical evaluation against human auditors. The confidence scoring framework is theoretically motivated and architecturally sound, but has not yet been validated against ground-truth audit findings from professional auditors. A controlled study comparing FinRegAgents findings with those of human BaFin auditors on the same document set would provide the calibration data needed to tune confidence thresholds.

In-memory vector index. The current LlamaIndex-based index is rebuilt on every run. For production use, a persistent vector store (ChromaDB, Weaviate) with delta-update capabilities is needed.

Skeptic-agent not yet implemented. The adversarial verification layer (Layer 2 in the coverage–verification framework) is architecturally specified but not yet implemented. The planned design sends each finding together with the actually retrieved evidence chunks to a cheaper model (Claude Haiku or Gemini Flash) configured as an adversarial reviewer. This creates a cost-effective verification layer without requiring Opus-level expenditure.

Screenshot analysis is stub-only. Screenshots are indexed as placeholder documents but not analyzed via vision models. Integrating Claude's vision capabilities for KYC interface screenshots and transaction monitoring dashboards would extend evidence coverage to visual compliance indicators.

Synthetic control fields for calibration. The ground-truth calibration mechanism—embedding audit fields with known correct assessments—is designed but not yet integrated into the catalogs. This would enable per-run accuracy measurement and confidence threshold auto-tuning.

8. Conclusion

FinRegAgents demonstrates that RAG-based regulatory audit systems require explicit verification architectures proportional to their coverage scope. The retrieval quality gate, composite

confidence score, and structural validation pipeline address the three core problems identified: hallucination on thin evidence, confidence opacity, and the coverage–verification gap.

The system's declarative catalog design, multi-modal ingestion, and transparent confidence reporting establish a pattern that generalizes beyond financial regulation to any domain where AI-assisted document analysis must produce findings with quantified reliability. The open-source release (Apache 2.0) enables the regulatory technology community to build on this architecture.

The coverage–verification gap framework suggests a broader principle: in high-stakes AI applications, the question is not "how many tasks can the system perform" but "how many of its outputs can the system verify." Systems that expand coverage without proportional verification investment create systematic risk—a lesson the ad-tech industry learned at scale and one that the regulatory AI community should internalize early.

9. Code Availability

FinRegAgents v2 is available at **github.com/endvater/finreg-agents** under the Apache License 2.0. The repository includes the complete pipeline, all four regulatory catalogs (94 audit fields), the test suite, and documentation.

References

[1] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS 2020*.

[2] Gao, Y., Xiong, Y., Gao, X., et al. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv:2312.10997*.

[3] Cui, J., Li, Z., Yan, Y., et al. (2024). ChatLaw: A Multi-Agent Collaborative Framework for Legal Applications. *arXiv:2306.16092*.

[4] Augustine, J., Xu, R. (2023). Click Fraud Detection: A Comprehensive Survey. *ACM Computing Surveys*, 55(13s), 1–38.

[5] Hendrycks, D., Burns, C., Chen, A., Spencer, B. (2021). CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. *NeurIPS 2021 Datasets Track*.

[6] Niklaus, J., Chalkidis, I., Stürmer, M. (2023). MultiLegalPile: A 689GB Multilingual Legal Corpus. *arXiv:2306.02069*.

[7] Savelka, J., Ashley, K. (2023). Explaining Legal Concepts with Augmented Large Language Models. *JURIX 2023*.

[8] Savelka, J., Ashley, K., Gray, M., Westermann, H., Xu, H. (2023). Can GPT-4 Support Analysis of Textual Data in Tasks Requiring Highly Specialized Domain Expertise? *ASAIL 2023*.

[9] Louis, A., van Dijck, G., Spanakis, G. (2024). Interpretable Long-Form Legal Question Answering with Retrieval-Augmented Generation. *AAAI 2024*.

[10] Xiong, M., Hu, Z., Lu, X., et al. (2024). Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. *ICLR 2024*.

[11] Tian, K., Mitchell, E., Yao, H., Manning, C., Finn, C. (2023). Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models. *EMNLP 2023*.

[12] Kadavath, S., Conerly, T., Askell, A., et al. (2022). Language Models (Mostly) Know What They Know. *arXiv:2207.05221*.

[13] Akhigbe, O., Amyot, D., Richards, G. (2019). A Systematic Literature Review of Requirements Engineering in Regulatory Compliance. *Journal of Systems and Software*, 149, 111–132.

[14] Becker, J., Delfmann, P., Dietrich, H., Steinhorst, M., Eggert, M. (2023). Business Process Compliance Checking: Current State and Future Challenges. *Business & Information Systems Engineering*, 65(4), 423–445.

[15] Arner, D., Barberis, J., Buckley, R. (2017). FinTech, RegTech, and the Reconceptualization of Financial Regulation. *Northwestern Journal of International Law and Business*, 37(3).

[16] Baek, J., Jauhar, S., Cucerzan, S., Hwang, S. (2024). ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. *arXiv:2404.07738*.

[17] Hong, S., Zhuge, M., Chen, J., et al. (2024). MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. *ICLR 2024*.

[18] Du, Y., Li, S., Torralba, A., Tenenbaum, J., Mordatch, I. (2023). Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv:2305.14325*.

[19] Ma, X., Gong, Y., He, P., Zhao, H., Duan, N. (2023). Query Rewriting in Retrieval-Augmented Large Language Models. *EMNLP 2023*.