FlowX.AI

TECHNICAL PAPER

OBSERVABILITY-DRIVEN RECURSIVE NODE ADAPTATION (ORNA)

Agent Self-Adaptation in Production

Closing the loop between observability, validation, and node-level refinement for LLM agents in regulated industries.

Bogdan Răduță
Head of Research, FlowX.AI
bogdan.raduta@flowx.ai

FlowX.AI · Applied Research

April 2026 · Version 1.0

ABSTRACT

Large language model (LLM) based agents deployed in regulated industries face a persistent tension: they operate in dynamic environments with evolving compliance constraints, yet manual agent tuning is slow, expensive, and fragile. Existing self-improvement frameworks assume benign operating conditions and largely ignore the governance requirements of production-grade systems. We present a framework for autonomous agent self-adaptation that closes the loop between structured observability, multi-tier validation, and node-level workflow refinement, without retraining or model weight updates. Our approach monitors production trace patterns through an observability layer (Observatory), identifies failure signatures at the node level, and applies targeted adaptations (prompt updates, recognizer extensions, threshold adjustments) subject to a three-tier validation protocol before safe rollout. We evaluate the framework on a cross-border loan classification use case with real regulatory compliance constraints, demonstrating measurable accuracy improvement (87% to 90%+ confidence) while reducing compliance violations from 15 to 0 in a single adaptation cycle. We discuss how the framework generalizes to KYC screening, fraud detection, and insurance claims processing, and we address the governance mechanisms (audit trails, rollback, human escalation thresholds) required for production deployment in regulated environments.

KEYWORDS agent self-improvement · LLM orchestration · production AI systems · observability · compliance · BFSI · governed AI · multi-tier validation

1 Introduction

Enterprise AI agents in banking, financial services, and insurance (BFSI) are increasingly deployed as multi-step orchestration workflows: sequences of LLM-powered nodes that extract entities, classify documents, apply regulatory rules, and produce structured outputs. Unlike research agents operating on benchmarks, these systems must satisfy hard compliance requirements. Every decision must be auditable, compliant with regulatory constraints, and explainable to regulators and internal risk functions.

A fundamental challenge emerges: these agents degrade over time. Entity vocabularies change. New regulatory guidance appears. Counterparty structures evolve. A loan classification agent tuned in Q1 may produce compliance violations by Q3 without anyone noticing until an audit. The response today is manual: a compliance team flags anomalies, engineers tune prompts or update extraction rules, a validation cycle is run, and a patch is deployed, typically over days or weeks.

This paper argues that this tuning loop can and should be automated, but only if automation is designed with governance at its core. Self-improvement for its own sake is insufficient and potentially dangerous in regulated environments. What is required is a closed-loop adaptation framework in which:

1. Structured observability surfaces failure patterns as actionable signals.
2. Adaptation targets node-level configurations (not model weights), ensuring interpretability and rollback capability.
3. Multi-tier validation enforces that adaptations improve accuracy without introducing new compliance violations.
4. An audit trail captures the full trace-to-adaptation-to-outcome chain, satisfying regulatory explainability requirements.

We call this framework **Observability-Driven Recursive Node Adaptation (ORNA)**, implemented atop a production LangGraph-based orchestration engine[1] and an enterprise observability platform.

1.1 Contributions

- A principled framework for autonomous agent self-adaptation in compliance-constrained environments.
- A three-tier validation protocol that decouples accuracy improvement from compliance regression.
- A concrete case study, cross-border loan classification, with measurable production metrics.
- A taxonomy of adaptation types (recognizer extension, prompt refinement, threshold adjustment) and their associated safety profiles.
- Governance patterns (audit trails, rollback mechanisms, human escalation thresholds) generalized across BFSI use cases.

2 Background and Related Work

2.1 Agent Architecture

Our framework operates on LangGraph-based directed graph workflows[1]. Each node encapsulates a bounded operation: entity extraction, classification, rule evaluation, output structuring. Nodes are parameterized by prompts, recognizer vocabularies[2], and validation thresholds, all of which can be adapted at runtime without altering the underlying model or graph topology. Edges define execution order, branching conditions, and parallel fan-out patterns. This node-level modularity is essential: adaptations remain scoped and interpretable.

2.2 Observability as a Feedback Mechanism

Prior work on agent observability has focused primarily on debugging and monitoring. AgentTrace[3] introduces multi-surface structured logging covering operational, cognitive, and contextual dimensions. AgentOps[4] proposes a hierarchical span model with guardrails as universal connectors across lifecycle phases. EDDOps[5] links evaluation outputs from production logs to continuous refinement. Our work extends this line: rather than treating traces as a monitoring artifact, we treat them as the primary input to an autonomous adaptation pipeline.

2.3 Self-Improving Agents

The self-improvement literature spans several approaches. SICA[6] demonstrates scaffolding-based improvement for coding agents without weight updates, noting observability as a safety mitigation. The Generator-Verifier-Updater (GVU) framework[7] formalizes improvement as a stability-bounded loop using variance inequality conditions. ARIA[8] achieves test-time learning through structured self-dialogue and human-in-the-loop corrections, targeted at regulatory compliance and domain knowledge drift.

What is notably absent from this literature is a framework that combines (a) production-grounded observability, (b) autonomous adaptation within guardrail boundaries, and (c) multi-tier validation enforcing compliance preservation. Prior frameworks either require human experts in the loop (ARIA[8]), operate in non-regulatory environments (SICA[6], GVU[7]), or address observability without autonomous adaptation (AgentTrace[3], EDDOps[5]).

2.4 Compliance and Governance in LLM Systems

LLM governance in regulated industries has been addressed from audit-trail perspectives[9,10], explainability requirements[11], and bias and fairness lenses[12]. None address the specific challenge of maintaining compliance during autonomous agent adaptation. This is our primary contribution gap.

3 Framework: Observability-Driven Recursive Node Adaptation

The ORNA framework consists of five interacting layers operating over a production agent deployment. Figure 1 provides an architectural overview.

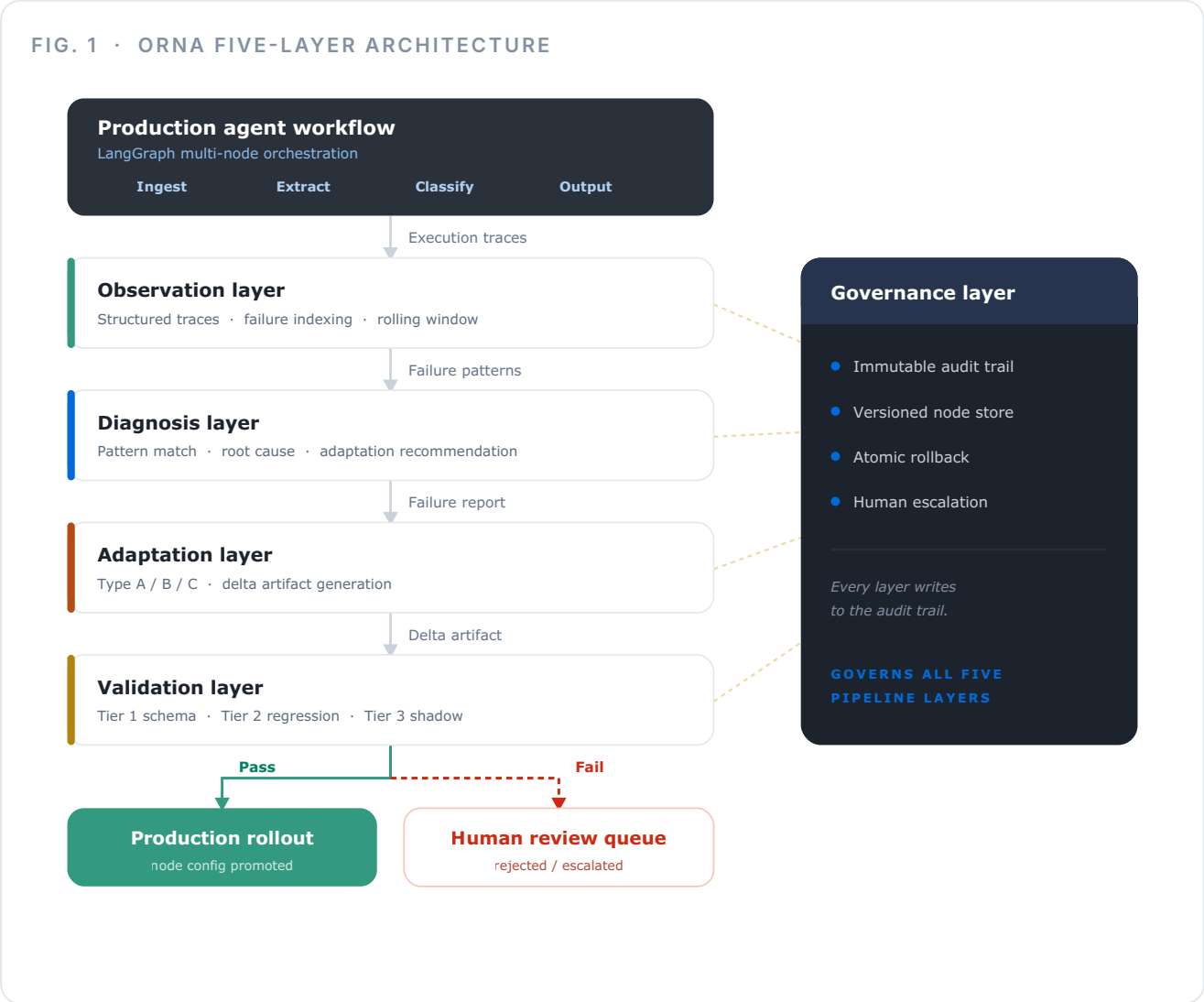

Figure 1. The five interacting layers of ORNA over a production agent deployment. Execution traces flow downward from the live workflow through observation, diagnosis, adaptation, and validation. Validated adaptations are promoted to production; rejected or escalated ones enter the human review queue. The governance layer spans the full pipeline, writing a signed, immutable record at every step.

3.1 Observation Layer

The observation layer collects structured traces from agent executions. Each trace captures:

- **Span-level metadata:** node identifier, execution time, input and output tokens, model invoked.
- **Structured output validity:** whether the node's output conformed to its declared schema.
- **Compliance flags:** violations of declared regulatory rules (for example, cross-border entity classification failures or missing required fields).
- **Confidence scores:** where the node produces probabilistic outputs, the distribution over classes.

Critically, traces are indexed by failure type rather than merely logged chronologically. The observation layer maintains a rolling window of recent execution traces grouped by failure signature, enabling the diagnosis layer to reason over patterns rather than individual failures.

3.2 Diagnosis Layer

The diagnosis layer applies a pattern-matching pipeline over indexed failure traces. Given a threshold of k failures of a consistent type within a rolling window W, the diagnosis layer produces a failure report consisting of:

- **Failure locus:** the specific node (or node pair) implicated in the failure pattern.
- **Failure type:** one of {schema violation, confidence degradation, compliance flag, extraction miss}.
- **Hypothesized root cause:** inferred from structural analysis of inputs and outputs. For example, repeated extraction misses on inputs containing subsidiary LLC structures would generate a "nested entity type gap" hypothesis.
- **Adaptation recommendation:** the class of adaptation estimated to address the root cause.

The diagnosis layer does not adapt agents directly. Its output is a structured artifact that the adaptation layer can act upon, and which is logged to the audit trail regardless of whether adaptation proceeds.

3.3 Adaptation Layer

The adaptation layer consumes failure reports and applies targeted modifications to the implicated node's configuration. We define three classes of adaptation with distinct safety profiles:

Type A Recognizer Extension

Adding entity types, vocabulary entries, or structural patterns to an extraction node's recognizer. *Safety profile:* low risk; additive change; prior extractions unaffected.

Type B Prompt Refinement

Modifying the system or task prompt of an LLM-invocation node to address systematic reasoning failures. *Safety profile:* medium risk; behavioral change; requires evaluation against the regression suite.

Type C Threshold Adjustment

Modifying confidence thresholds, classification cutoffs, or routing conditions. *Safety profile:* medium-high risk; affects classification distribution; requires compliance impact analysis.

For each adaptation, the layer generates a *delta artifact*, the exact diff between original and adapted configuration, and routes it to the validation layer before any deployment.

3.4 Validation Layer

The validation layer applies a three-tier protocol to every proposed adaptation before it is permitted to reach production:

Tier 1 Schema and Structural Validation

The adapted node configuration must parse correctly, conform to the declared node interface schema, and pass static analysis. Failures here are immediate blockers; the adaptation is rejected without further evaluation.

Tier 2 Regression Evaluation

The adaptation is evaluated against a held-out regression suite drawn from historical production traces (with known-good outputs). Acceptance criterion: no degradation in accuracy (measured by F1 over structured output fields) and no new compliance violations introduced. This tier runs against a sandboxed replica of the production workflow.

Tier 3 Shadow Deployment Validation

The adapted node is deployed in shadow mode, processing production traffic but not emitting outputs, for a configurable observation window (default: 200 production invocations). Shadow metrics are compared against the production node's concurrent metrics. Acceptance criterion: statistically significant improvement or equivalence in the target metric, with no compliance flag increase.

FIG. 2 · THREE-TIER VALIDATION PROTOCOL

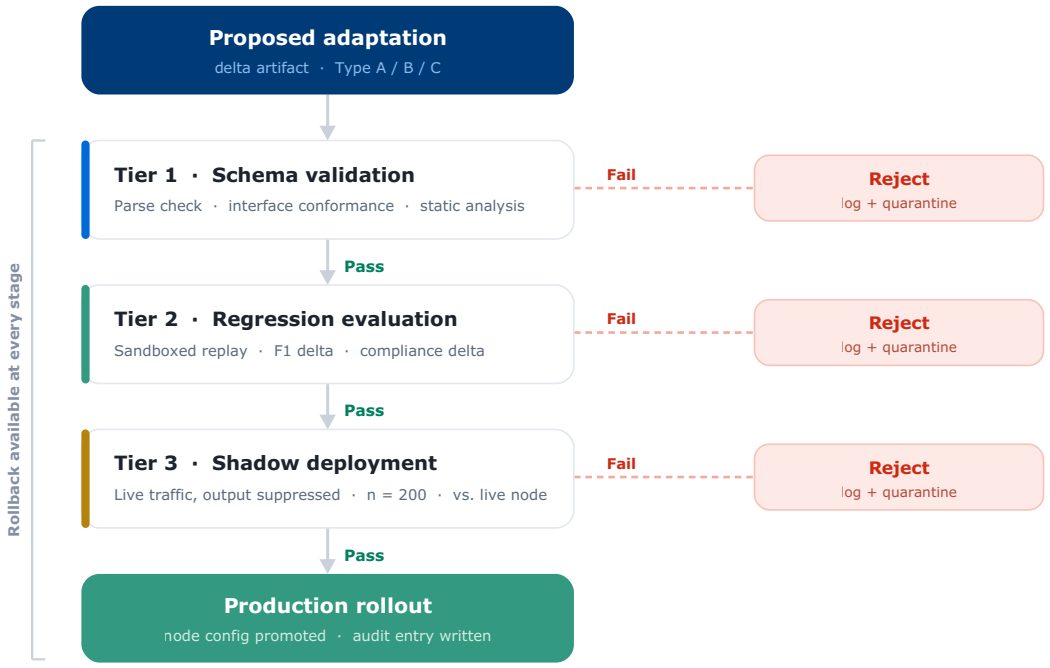

Figure 2. A proposed adaptation must clear all three validation tiers in sequence. Failure at any tier sends the candidate to a logged quarantine for human review; only candidates passing schema, regression, and shadow checks are promoted. Node-level versioning keeps rollback available at every stage.

Only adaptations passing all three tiers proceed to production rollout. Adaptations failing any tier are logged, quarantined, and flagged for human review.

3.5 Governance Layer

The governance layer maintains the audit trail and enforces human escalation thresholds. Every operation across all layers, including trace indexing, diagnosis report generation, adaptation proposal, validation outcomes, and production rollout, produces a signed, immutable log entry. This satisfies regulatory audit requirements for AI system decisions.

Human escalation is triggered when:

- An adaptation is of Type B or C and involves a node whose output feeds directly into a compliance-gated decision.
- Shadow deployment metrics show high variance, indicating the adaptation has significant but inconsistent effects.
- The adaptation cycle is recurring: the same node has been adapted more than n times within a rolling period, suggesting a structural problem beyond node-level tuning.

Rollback is available at node granularity. The adaptation layer maintains a versioned configuration store; any node can be reverted to a prior version atomically, without affecting other nodes in the graph.

4 Case Study: Cross-Border Loan Classification

4.1 Setup

We evaluate ORNA in a production loan classification workflow at a European financial institution. The workflow processes loan applications involving cross-border entities, applying regulatory classification rules to determine loan category, required reporting fields, and compliance flags. The workflow consists of six LangGraph nodes: document ingestion, entity extraction, jurisdiction resolution, regulatory rule evaluation, compliance flagging, and structured output generation.

The entity extraction node is parameterized by a recognizer vocabulary covering legal entity types, jurisdiction identifiers, and ownership structure patterns. The regulatory rule evaluation node applies a declarative rule set encoding the relevant reporting requirements.

4.2 Observed Failure Pattern

Over a three-week production window, Observatory traces flagged a degraded confidence pattern in the entity extraction node, coinciding with an increase in the compliance flag rate. Specifically, 15 of 28 cross-border loan applications processed in this window received compliance flags attributable to incomplete entity extraction. Entities involving nested LLC structures (subsidiary-of-subsidiary patterns) were being classified as unknown entity types, causing the jurisdiction resolution node to fail silently and the compliance flagging node to emit false positives.

The diagnosis layer identified this as a Type A failure (extraction miss) with the hypothesized root cause: the recognizer vocabulary lacked coverage for subsidiary LLC patterns introduced by a recent change in counterparty structuring conventions.

4.3 Adaptation and Validation

The adaptation layer proposed a recognizer extension: adding four new entity patterns covering subsidiary LLC variants (wholly-owned subsidiary, majority-owned subsidiary, LLC series structure, holding company wrapper). This constituted a Type A adaptation. Validation proceeded as follows:

- **Tier 1:** Configuration valid. Passes schema checks.
- **Tier 2:** Regression suite (n = 120 historical traces): accuracy improved from 87.2% to 90.4% (F1 over entity extraction fields). Compliance violations: 0 (down from 3 in the regression set).
- **Tier 3:** Shadow deployment (n = 200 production invocations): target metric improved 3.1 percentage points versus the concurrent production node. No new compliance flags introduced.

The adaptation was promoted to production. Post-deployment monitoring over the following two-week window confirmed: entity extraction confidence stabilized above 90%, the compliance flag rate on cross-border applications dropped to 0, and no regression was observed in standard (non-cross-border) applications.

4.4 Summary Metrics

Table 1. Production metrics before and after a single Type A adaptation cycle.

Metric	Baseline	Post-adaptation
Entity extraction confidence	87.2%	**90.4%**
Compliance violations (cross-border)	15 / 28	**0 / 28**
Adaptation type	–	Type A (recognizer extension)
Validation time	–	~4 hours
Human intervention required	–	None

5 Generalization to Other BFSI Use Cases

The ORNA framework is not specific to loan classification. We outline analogous adaptation scenarios in three additional domains.

KYC Screening

Entity extraction nodes may fail to identify new counterparty structure types (for example, trust layering or shell company indicators). Type A adaptations (recognizer extension) address coverage gaps. Type B adaptations (prompt refinement) address systematic misclassification of beneficial ownership relationships.

Insurance Claims Processing

Damage assessment nodes may exhibit confidence degradation when encountering new claim categories. Type C adaptations (threshold adjustment) recalibrate classification cutoffs. Type B adaptations refine prompts to handle edge cases in damage description language.

Fraud Detection

Feature extraction nodes may miss signals associated with emerging fraud patterns. Type A adaptations extend pattern vocabularies. Governance layer escalation thresholds are tightened for fraud-adjacent nodes, given the high cost of false negatives.

In each case, the three-tier validation protocol and the governance layer apply without modification; only the adaptation type and the escalation thresholds are domain-specific.

6 Safety, Governance, and Limitations

6.1 Safety Properties

ORNA provides three safety properties by construction.

- **Compliance preservation.** The Tier 2 regression suite includes compliance checks. An adaptation that improves accuracy but introduces compliance violations will be rejected.
- **Rollback atomicity.** Node-level versioning ensures that a regression in production can be reverted within minutes, without affecting the remainder of the workflow.
- **Auditability.** Every adaptation, including rejected adaptations, is logged with a full trace of the evidence that triggered it. Regulators can reconstruct the full decision chain for any period.

6.2 Limitations

- **Guardrail quality assumption.** ORNA assumes that the compliance rules encoded in the regulatory rule evaluation node are correct and complete. The framework strengthens existing guardrails; it does not detect missing or incorrect guardrails.
- **Novel pattern risk.** If a failure pattern is truly novel, not representable by any of the three adaptation types, ORNA will escalate to human review correctly but cannot adapt autonomously. This is a design choice, not a limitation to be engineered away.
- **Regression suite coverage.** Tier 2 validation is only as good as the regression suite. If historical traces do not cover relevant edge cases, a regression-passing adaptation may still fail in production. Ongoing trace collection and regression suite expansion are operational requirements.
- **Scope.** We evaluate on single-institution, single-workflow deployments. Multi-agent orchestration and cross-agent learning are out of scope for this paper.

6.3 When Not to Auto-Adapt

ORNA includes explicit human escalation thresholds. Auto-adaptation should not proceed when:

- The implicated node makes final compliance-gated decisions, as opposed to intermediate extraction or classification.
- The failure pattern has not been observed before, suggesting the agent may be encountering an out-of-distribution input distribution.
- The adaptation cycle is recurring on the same node, suggesting a structural misalignment between the node's design and its operating environment.

7 Discussion

The central claim of this paper is not that agents should self-improve. It is that agents in regulated environments *can* self-improve, but only within a governance architecture that makes adaptation safe, auditable, and bounded. The contribution is not the optimization loop per se; it is the design of the loop such that it strengthens rather than undermines the compliance properties of the system.

This distinction matters for the broader field. Much of the self-improvement literature assumes that accuracy improvement is the primary goal and that governance is a secondary concern. In BFSI, and increasingly in healthcare, legal, and infrastructure domains, this order is inverted.

> *An agent that is 95% accurate but unauditable is operationally unusable. An agent that is 87% accurate but fully auditable, with a validated path to 90%, is deployable.*
>
> ORNA DESIGN PRINCIPLE

ORNA operationalizes this priority ordering: compliance preservation is a hard constraint; accuracy improvement is the objective, subject to that constraint. The three-tier validation protocol enforces this ordering mechanically, not by convention.

Future work will address: (a) cross-agent learning, where adaptation evidence from one workflow informs adaptations in topologically similar workflows; (b) adaptive guardrail generation, where the framework proposes new compliance rules based on observed violation patterns; and (c) multi-institution federated adaptation, where anonymized failure patterns are shared across institution boundaries to accelerate vocabulary coverage expansion.

8 Conclusion

We have presented ORNA, a framework for autonomous agent self-adaptation in regulated environments. The framework closes the loop between structured observability, node-level adaptation, multi-tier validation, and governance, enabling agents to improve their own accuracy and compliance posture without human intervention, within bounded and auditable guardrails.

Empirical evaluation on a cross-border loan classification deployment demonstrates measurable improvement (87% to 90% confidence, 15 to 0 compliance violations) in a single adaptation cycle with no human intervention and a four-hour end-to-end loop. We have argued that this approach generalizes across BFSI use cases and that its governance properties make it suitable for production deployment in compliance-constrained environments.

The principal contribution is not a new self-improvement mechanism but a new design philosophy: autonomous adaptation as a *governed operation*, not a freeform optimization.

References

[1] LangChain, Inc. *LangGraph: A low-level orchestration framework for building stateful, multi-actor LLM agent applications.* github.com/langchain-ai/langgraph (accessed 2026).

[2] Microsoft. *Presidio: Context-aware, customizable PII detection and de-identification SDK.* github.com/microsoft/presidio (accessed 2026).

[3] *AgentTrace: A Structured Logging Framework for Agent System Observability.* arXiv:2602.10133, 2026.

[4] Dong, L., Lu, Q., & Zhu, L. *AgentOps: Enabling Observability of LLM Agents.* arXiv:2411.05285, 2024.

[5] Xia, B., Lu, Q., Zhu, L., Xing, Z., Zhao, D., & Zhang, H. *Evaluation-Driven Development and Operations of LLM Agents: A Process Model and Reference Architecture.* arXiv:2411.13768, 2024.

[6] Robeyns, M., Szummer, M., & Aitchison, L. *A Self-Improving Coding Agent.* arXiv:2504.15228, 2025.

[7] *Self-Improving AI Agents through Self-Play.* (Introduces the Generator-Verifier-Updater operator and the Variance Inequality stability condition.) arXiv:2512.02731, 2025.

[8] He, Y., Li, R., Chen, A., Liu, Y., Chen, Y., Sui, Y., Chen, C., Zhu, Y., Luo, L., Yang, F., & Hooi, B. *Enabling Self-Improving Agents to Learn at Test Time With Human-In-The-Loop Guidance.* Proc. EMNLP 2025 (Industry Track). arXiv:2507.17131, 2025.

[9] European Parliament and Council. *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).* Official Journal of the European Union, 2024.

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

[11] Barredo Arrieta, A., et al. *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI.* Information Fusion, 58:82-115, 2020.

[12] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. *A Survey on Bias and Fairness in Machine Learning.* ACM Computing Surveys, 54(6):1-35, 2021.

Appendix A Adaptation Type Taxonomy

Type	Target	Risk	Validation requirement	Example
A. Recognizer extension	Entity patterns, vocabulary	Low	Tier 1 + Tier 2	Add subsidiary LLC pattern
B. Prompt refinement	LLM system / task prompt	Medium	All three tiers	Clarify beneficial-ownership reasoning
C. Threshold adjustment	Confidence cutoffs, routing	Med-high	All three tiers + escalation review	Recalibrate fraud threshold

Appendix B Observatory Trace Schema (Abbreviated)

JSON

```
{
  "trace_id": "uuid",
  "workflow_id": "loan-classification-v2",
  "node_id": "entity-extraction",
  "timestamp": "ISO8601",
  "execution_ms": 840,
  "input_tokens": 1240,
  "output_tokens": 312,
  "structured_output_valid": false,
  "confidence_score": 0.61,
  "compliance_flags": ["MISSING_SUBSIDIARY_ENTITY"],
  "failure_type": "extraction_miss",
  "raw_output": "..."
}
```


FlowX.AI Technical Paper · ORNA v1.0 · Bogdan Răduță, Head of Research · bogdan.raduta@flowx.ai