AML Observability: Rethinking Transaction Monitoring as a Debuggable System

A Position Paper on Compliance System Architecture

Jürgen Schiller García
FinCrime Watchdog
ORCID: 0009-0001-5370-4972

April 2026 · v1.3 · Submitted to aixiv.science

Abstract

AML transaction monitoring systems are widely deployed but often difficult to reconstruct and evaluate in production. This paper proposes that a key limitation is not only insufficient detection logic, but insufficient system observability. It defines *AML Observability* as the capability to reconstruct detection-relevant decisions across the full processing lifecycle. The paper proposes a five-layer architecture, introduces a *Transformation Spiral* linking governance, observability, and AI, and uses an OBASHI-informed lens to interpret public enforcement cases. It concludes that observability is a significant enabling condition under which governance can become evidence-based and AI more readily governable.

Keywords: AML, Transaction Monitoring, Observability, Compliance Architecture, Data Lineage, OBASHI, Transformation Spiral, Strangler Fig Pattern, AMLR, AMLA, RegTech

---

1 Introduction

AML monitoring is typically framed as a detection problem: build better rules, train sharper models, reduce false positives. However, many documented failures suggest a deeper issue: institutions often cannot reconstruct how their systems actually behave. They know *what* was flagged, but not *why*—and, more critically, they cannot systematically determine what was *missed*.

This paper argues that AML weaknesses are frequently better understood as observability deficits rather than detection deficits. The distinction matters: detection can be improved incrementally through tuning, but observability requires architectural decisions about how systems describe themselves. Without that foundation, every downstream improvement—whether rule refinement, model deployment, or AI integration—operates on uncertain ground.

The paper draws on concepts from distributed systems engineering—where observability has been a first-class design principle for over a decade—and applies them to the AML domain. It builds on a four-part article series published on FinCrime Watchdog (April 2026), consolidating the market analysis, layer model, OBASHI case interpretation, and legacy migration strategy into a single, peer-reviewable position paper.

2 Related Work

This paper draws on three distinct bodies of literature: observability in distributed systems, architectural patterns for financial transaction processing, and regulatory technology (RegTech).

Positioning the contribution requires engaging with each.

2.1 Observability in Distributed Systems

The foundational distinction between monitoring and observability was operationalized for software systems by Majors et al. [Majors et al., 2022] and Sridharan [Sridharan, 2018], building on Kalman's control-theoretic definition [Kalman, 1960]. The OpenTelemetry project [OpenTelemetry, 2024] has since standardized the instrumentation model around logs, metrics, and traces. Recent work has extended observability into multi-cloud and federated environments: Jha [Jha, 2025] proposes federated learning for cross-cloud observability with privacy-preserving model aggregation, while Garnimitta [Garnimitta, 2025] develops a federated AI observability framework for multi-cloud microservices with differential privacy guarantees. These contributions demonstrate that observability architectures can operate under data sovereignty constraints—a finding directly relevant to GDPR-constrained AML systems (see Section 11).

2.2 Event-Driven Architectures for Financial Systems

Event-driven architecture (EDA) has been widely adopted for real-time financial transaction processing. Chilukala [Chilukala, 2025] demonstrates that EDA-based payment systems achieve sub-200ms latencies and 99.99% success rates, while Thompson and Davis [Thompson & Davis, 2023] propose observability-driven development as a design paradigm for distributed financial systems. These architectures share structural similarities with the five-layer stack proposed in Section 6.1: both decompose processing into discrete stages with inter-stage telemetry. However, existing EDA literature focuses on *operational* observability (latency, throughput, availability) rather than *compliance* observability (detection lineage, decision reconstruction, regulatory explainability). Our contribution is the explicit extension of observability from operational concerns to governance and AI governability.

2.3 RegTech and Compliance Automation

The RegTech literature has produced frameworks for standardizing compliance practices across institutions [Arner et al., 2017], and recent work applies advanced monitoring techniques to transaction integrity [Liu et al., 2025]. However, these contributions typically assume that the underlying monitoring infrastructure is adequate and focus on detection logic rather than system-level reconstructability. Schiller García [Schiller García, 2026c] identifies a related structural limitation in the context of RAG-based regulatory audits: the *coverage–verification gap*, where a system's ability to produce compliance assessments outpaces its ability to verify them. AML Observability addresses the architectural precondition that FinRegAgents assumes: without observable data pipelines, even well-designed verification architectures operate on uncertain inputs.

2.4 Positioning This Contribution

Table 1 summarizes how this paper relates to and extends prior work. The core novelty is not the application of observability *per se*, but the argument that observability is the *sequencing precondition* for both governance effectiveness and AI governability in AML—a claim not made in any of the cited works.

2.5 Conceptual Demarcation: What AML Observability Is Not

A legitimate concern is whether "AML Observability" merely relabels established concepts. Table 2 addresses this by distinguishing observability from six adjacent concepts and identifying what each covers and where it falls short.

Table 1: Positioning relative to related work.

Prior Work	**Focus**	**This Paper Extends By**
Majors et al. (2022)	Observability for distributed systems	Applying observability to compliance/AML domain
Jha (2025), Garnimitta (2025)	Federated observability with privacy	Connecting to GDPR/banking secrecy constraints in AML
Chilukala (2025)	EDA for financial transaction processing	Shifting from operational to compliance observability
Liu (2025)	Advanced transaction monitoring techniques	Adding system-level reconstructability as prerequisite
Arner et al. (2017)	RegTech standardization frameworks	Focusing on architectural preconditions, not policy
Fed/OCC (2011)	Model Risk Management (SR 11-7)	Extending from model validation to full-pipeline observability

Table 2: AML Observability vs. adjacent concepts: what each covers and where it falls short.

Concept	**What It Covers**	**What It Does Not Cover**
Data Lineage	Where data originates and how it flows through systems	Detection logic, model behavior, case outcomes, feedback loops
Audit Trail	Retrospective record of who did what and when	Diagnostic capability: cannot answer *why* a decision was made or what was missed
Explainable AI (XAI)	Post-hoc interpretation of individual model predictions [Fed/OCC, 2011]	Upstream data quality, ingestion completeness, transformation correctness (Layers 1–3)
Process Mining	Discovery and conformance checking of actual process flows [van der Aalst, 2016]	Requires structured event logs that AML batch systems rarely produce; focuses on process, not data/decision quality
Model Risk Mgmt (SR 11-7)	Validation, governance, and controls for quantitative models [Fed/OCC, 2011]	System-level reconstructability; treats model as isolated component, not as part of a five-layer pipeline
Operational Monitoring	System availability, latency, throughput, error rates	Compliance correctness: a system can be "up" and processing transactions while silently missing detections
AML Observability	**End-to-end reconstructability of detection-relevant decisions across all five layers**	**Integrates the above as layer-specific capabilities within a unified diagnostic framework**

The key argument is that AML Observability is not a seventh concept alongside the six above, but the *integration principle* that connects them across the full processing lifecycle. Data lineage serves Layer 1–2, XAI serves Layer 4, audit trails serve Layer 5—but without an architecture that spans all five layers simultaneously, each operates in its own silo. Observability is the architectural property that makes the whole greater than the sum of its parts.

3 Core Propositions

The paper rests on three propositions that together form the argumentative spine.

P1: Observability deficits create epistemic uncertainty in AML systems.
When a system cannot describe its own internal state—which inputs it consumed, which transformations it applied, which signals it ignored—the institution operating it faces epistemic uncertainty, i.e. uncertainty about what the system actually does in production. This is not a theoretical concern: it manifests in the inability to explain false positives, the invisibility of false negatives, and the difficulty of validating detection effectiveness under audit.

P2: Governance without observability remains under-evidenced.
Regulatory frameworks—MaRisk, BAIT, DORA, the forthcoming AMLA regulatory technical standards—increasingly require institutions to *demonstrate* effectiveness, not merely *assert* it. Without system-level observability, governance remains declarative: institutions describe what *should* happen rather than evidencing what *does* happen.

P3: AI without observability is difficult to govern.
The integration of machine learning and AI into AML systems introduces additional complexity. If the system surrounding them is itself not observable, the challenge compounds: the institution cannot explain the model's behavior *or* the data pipeline that feeds it. Observability is therefore a precondition for responsible AI deployment in compliance, not an afterthought.

4 The Conceptual Gap: Monitoring vs. Observability

The distinction between monitoring and observability has not yet been systematically applied to AML. Table 3 makes the mapping explicit:

Table 3: Monitoring vs. Observability—conceptual mapping for AML systems.

Dimension	**Monitoring**	**Observability**
Design premise	Known failure modes are anticipated	Unknown failures must be discoverable
Instrumentation	Predefined dashboards, thresholds	Rich telemetry: logs, metrics, traces
Core question	"Did something happen?"	"Why did it happen—and what did we miss?"
Output	Alerts	Reconstructable understanding
Epistemology	Confirmation of expected patterns	Exploration of system state space

The concept of observability originates from control theory [Kalman, 1960] and was operationalized for distributed software systems by practitioners including Charity Majors, Cindy Sridharan, and the OpenTelemetry project.

5 Structural Limitations of Current AML Systems

5.1 Fragmented Data Flows

Multiple upstream systems deliver data through inconsistent transformations. End-to-end traceability from source record to detection decision typically requires manual reconstruction.

5.2 Opaque Detection Logic

Rules and models operate with limited visibility into feature engineering and decision boundaries.

5.3 Missing Feedback Loops

Case outcomes are rarely reintegrated systematically into model tuning or rule calibration.

5.4 No System-Level Debugging

False negatives remain structurally invisible. This is an architectural consequence of systems designed for detection output rather than self-description.

6 AML Observability: A Conceptual Architecture

We define **AML Observability** as:

> *The capability to reconstruct detection-relevant decisions across inputs, transformations, feature derivations, model inferences, and disposition outcomes—across the full processing lifecycle.*

6.1 The Five-Layer AML Observability Stack

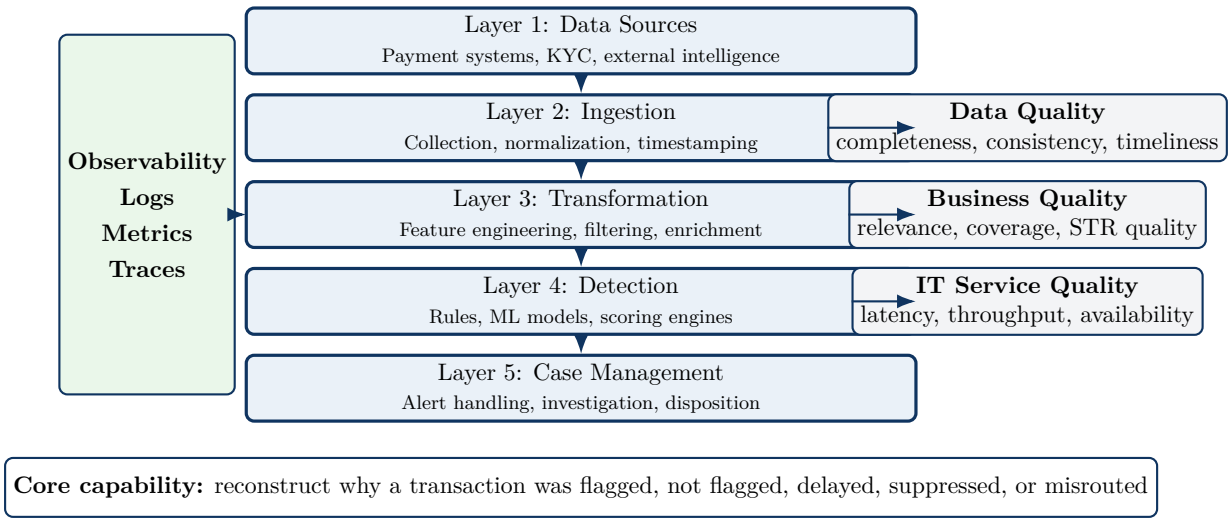

Figure 1: The Five-Layer AML Observability Stack.

6.2 Table View of the Stack

6.3 Observability Pillars

Analogous to distributed systems, AML observability relies on three pillars: **Logs** (structured records of what events occurred within the processing pipeline), **Metrics** (time-series data on

Table 4: The Five-Layer AML Observability Stack.

Layer	Function	Observability Requirement
1. Data Sources	Payment systems, KYC, external intelligence	Source completeness, freshness, schema validation
2. Ingestion	Collection, normalization, timestamping	Provenance tagging, deduplication metrics, latency
3. Transformation	Feature engineering, filtering, enrichment	Transformation lineage, data quality scores
4. Detection	Rules, ML models, scoring engines	Decision explanation, confidence scores, coverage
5. Case Management	Alert handling, investigation, disposition	Outcome tracking, feedback integration, SLA metrics

system behavior—alert volumes, processing latency, data quality scores, model drift indicators), and **Traces** (end-to-end reconstruction of how a specific transaction flowed through all five layers from source to disposition).

The system must be able to answer four questions for any given transaction: Why was it flagged (or not)? Which data contributed to the decision? What transformations were applied? What signals were ignored or lost?

6.4 A Hypothetical AML Trace

To make the five-layer architecture concrete, consider a hypothetical transaction trace for a single wire transfer flagged by the AML system. Table 5 illustrates the data points that would be captured at each layer in an observable system.

In a non-observable system, only layers 4 (the alert) and 5 (the disposition) are typically visible. Layers 1–3 are opaque: the institution cannot reconstruct which data fed the decision, whether the PEP status was current, or whether the feature derivation was correct. This trace structure illustrates why observability must span all five layers to answer the four core questions posed in Section 6.3.

The trace also demonstrates the technical requirements: each layer must emit structured telemetry (log entries, metric updates, trace spans) that can be correlated via a shared transaction identifier. This is architecturally analogous to distributed tracing in microservice systems (e.g., OpenTelemetry's `trace_id`/`span_id` model [OpenTelemetry, 2024]), applied to the compliance processing lifecycle rather than the request-response lifecycle.

6.5 Three Quality Dimensions

The critical insight is that these three dimensions are not independent: a data quality failure may manifest as a business quality failure, which in turn may appear as an IT service quality failure.

Table 5: Hypothetical AML trace: a single wire transfer through all five layers.

Layer	**Observable Data Points**	**Diagnostic Question Answered**
1. Data Sources	Transaction ID, amount (€48,200), originator (Customer A, PEP status: false), beneficiary (Entity B, jurisdiction: UAE), source system: SWIFT gateway, timestamp: 2026-04-06T09:14:22Z	Was the transaction received? Was PEP/sanctions data current at ingestion time?
2. Ingestion	Normalization applied (ISO 20022 mapping), dedup check (no duplicate), ingestion latency: 340ms, provenance tag: `swift-gw-prod-01`	Did the transaction arrive intact? Was it deduplicated? How fresh is the data?
3. Transformation	Features derived: beneficiary_jurisdiction_risk=high, amount_percentile=97.2, customer_avg_monthly=€12,400, cross-border=true. Data quality score: 0.91 (1 field missing: beneficiary LEI)	Which features were computed? Were any inputs missing or stale?
4. Detection	Rule R-17 triggered (high-risk jurisdiction + amount $>$ 3$\times$ average). ML model score: 0.78. Combined alert score: 0.82. Confidence: high. Alternative rules evaluated but not triggered: R-04, R-11	Why was this flagged? What was the model's contribution? What rules did not fire?
5. Case Mgmt	Alert assigned to analyst, priority: medium. Investigation time: 2.4h. Disposition: escalated to MLRO. STR filed: yes. Feedback: true positive	What happened after the alert? Did the outcome feed back into model tuning?

Table 6: Three quality dimensions of AML Observability.

Dimension	**Scope**	**Example Metrics**
Business Quality	Alert relevance, detection coverage, STR quality	True positive rate, typology coverage ratio
Data Quality	Completeness, consistency, timeliness of inputs	Missing field rate, cross-source consistency
IT Service Quality	Processing latency, availability, throughput	P95 latency, batch completion rate, error rate

7 The AML Transformation Spiral

A recurring question in the industry is sequencing: should institutions invest in AI, in governance, or in infrastructure first? We propose a **Transformation Spiral** as a heuristic sequencing model—not an empirically validated maturity framework, but a tool for making implicit dependencies between governance, observability, and AI explicit and discussable.

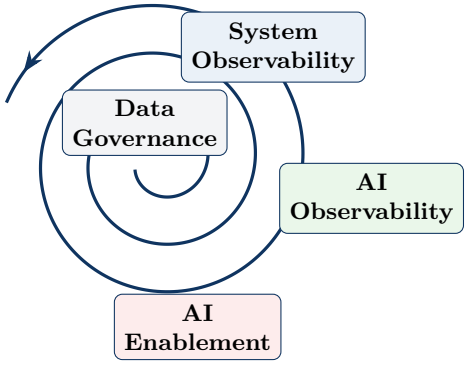

Figure 2: The AML Transformation Spiral.

Institutions may enter the spiral at any point. However, entry flexibility does not imply equal stability: the further institutions progress without establishing observability, the greater the accumulation of opacity, control debt, and rework. Maturity requires closing the loop.

8 Case Interpretation: An OBASHI-Informed Lens

Methodological note: These interpretations are illustrative, not causal proof. The selected cases represent widely documented, high-impact enforcement actions with publicly available descriptions of control breakdowns, allowing cross-case comparison despite limited architectural detail.

Table 7: OBASHI-informed observability gap interpretation of five enforcement cases.

Case	Fine	Primary Failure Layer	Observability Gap (Interpreted)
Danske Bank Estonia	$2B	Data Sources / Ingestion	No provenance for non-resident flows across subsidiary boundary
Westpac (AUSTRAC)	$1.3B AUD	Detection / Transformation	IFTI reporting gap in batch processing; silent failure
Capital One (FinCEN)	$390M	Case Mgmt / Feedback	SAR backlog without systematic triage or feedback
Swedbank	$386M	Data Sources / Transformation	Baltic subsidiary data not integrated into group monitoring
NatWest	£265M	Detection	Cash deposit alerts suppressed by static thresholds

The recurring pattern is that each enforcement action can be mapped to a failure at a specific layer where observability was absent or insufficient. Danske Bank and Swedbank share a common pattern: subsidiary data did not flow into group-level monitoring—a Layer 1–2 failure that no amount of detection tuning at Layer 4 could have compensated. Westpac's IFTI gap represents a silent transformation failure: the system processed transactions but silently dropped a reporting obligation, invisible without Layer 3 instrumentation. Capital One illustrates a feedback failure: the SAR backlog was not a detection problem but a case management bottleneck that never triggered upstream recalibration. NatWest's threshold suppression is perhaps the most straightforward: a detection-layer configuration that actively prevented alerts, discoverable only through cross-layer trace analysis. These are not merely isolated implementation failures; they are architectural gaps that an observability-first design would likely have made more visible.

9 Migration Strategy: The Strangler Fig Pattern

Table 8: Strangler Fig migration phases for AML observability.

Phase	Action	Value Created
1. Instrument	Add observability probes without changing behavior	Immediate visibility; baseline for improvement
2. Extract	Move processing steps into observable microservices	Parallel validation; gradual capability transfer
3. Replace	Redirect traffic once parity is validated	Full observability; legacy decommission

Phase 1 creates immediate value—visibility into the current system's actual behavior—before any replacement occurs.

10 Implications

10.1 Regulatory Compliance

AML observability directly supports traceability, data lineage, and explainability requirements from AMLA's RTS and the AMLR effectiveness mandate. These expectations are consistent with broader international supervisory trends, including FATF guidance on effectiveness and risk-based supervision.

10.2 Operational Efficiency

Observable systems enable systematic false positive reduction through root cause analysis, faster investigation through transaction-level tracing, and evidence-based model improvement.

10.3 Risk Management

An observable AML system can identify blind spots, detect systemic data flow weaknesses, and answer: "How do you know your system works?"

11 Discussion and Limitations

The proposed framework carries inherent limitations that must be addressed transparently.

System complexity and cost. An observability layer adds infrastructure, operational overhead, and engineering expertise requirements. Full-trace observability generates significant

storage and processing demands. However, recent work on federated observability [Jha, 2025, Garnimitta, 2025] suggests that privacy-preserving aggregation techniques can reduce the data volume that must be centralized, potentially making observability viable even under strict data minimization constraints.

GDPR and data minimization. A tension exists between the observability goal (reconstruct everything) and the GDPR principle of data minimization (store only what is necessary). We suggest two design strategies: (1) *selective trace retention*, where full traces are retained only for transactions that trigger alerts or that fall into statistical sampling windows, with summary metrics retained for all transactions; and (2) *on-demand trace reconstruction*, where raw telemetry is retained in encrypted, time-limited storage and full traces are assembled only when needed for investigation or audit—analogous to the "right to explanation" under GDPR Art. 22. Both strategies require careful legal analysis, but they demonstrate that observability and data minimization are not inherently contradictory.

Vendor lock-in and closed-box systems. The Strangler Fig migration strategy (Section 9) assumes that institutions can instrument their existing systems. In practice, many AML platforms (e.g., SAS, NICE Actimize, Oracle FCCM) are proprietary and offer limited API access to internal processing states. Phase 1 instrumentation in such environments must rely on *peripheral observability*: capturing inputs to and outputs from the closed-box system, monitoring batch job completion, and comparing expected versus actual alert volumes. This provides partial but immediate visibility. Full observability requires either vendor cooperation (exposing internal telemetry via APIs) or eventual replacement of opaque components—which is precisely what Phases 2 and 3 of the Strangler Fig pattern address.

Cultural shift. Moving from compliance-as-checkbox to engineering-driven approaches requires organizational transformation. The Transformation Spiral acknowledges this: maturity is iterative, not binary.

Open questions. How can observability standards be harmonized across institutions, potentially through AMLA's supervisory convergence mandate? Can federated observability architectures—building on the cross-cloud frameworks proposed by Jha [Jha, 2025]—enable cooperative compliance across institutions without centralizing sensitive data? What role should AI play not only in detection but in interpreting observability data, moving toward autonomous compliance diagnostics? And finally: can the coverage–verification gap formalized by Schiller García [Schiller García, 2026c] be narrowed systematically through observability-driven feedback loops?

12 Conclusion

The sequencing principle is clear:

> *First make the system legible. Then govern it. Then make it smarter.*

The five enforcement cases analyzed suggest that several high-impact AML failures of the past decade can plausibly be interpreted, at least in part, as observability failures—failures to know what the system was actually doing.

Observability First is not a competing paradigm. It is a sequencing principle—and arguably one the AML industry has not yet fully adopted.

References

[Majors et al., 2022] Majors, C., Fong-Jones, L., Miranda, G. *Observability Engineering*. O'Reilly Media, 2022.

[Sridharan, 2018] Sridharan, C. *Distributed Systems Observability*. O'Reilly Media, 2018.

[Alder & Wallis, 2017] Alder, F., Wallis, P. *OBASHI: Business & IT Alignment and Governance*. Van Haren Publishing, 2017.

[Fowler, 2004] Fowler, M. "StranglerFigApplication." martinfowler.com, 2004.

[Kalman, 1960] Kalman, R. E. "On the General Theory of Control Systems." *IFAC Proceedings*, 1(1), 1960.

[FATF, 2021] FATF. *Guidance on Risk-Based Supervision*. 2021.

[EU, 2024a] Regulation (EU) 2024/1624 (AMLR). Official Journal of the EU, 2024.

[EU, 2024b] Regulation (EU) 2024/1620 establishing AMLA. Official Journal of the EU, 2024.

[BaFin, 2024] BaFin. *Auslegungs- und Anwendungshinweise zum GwG*. Updated 2024.

[OpenTelemetry, 2024] OpenTelemetry Project. "OpenTelemetry Specification." opentelemetry.io, 2024.

[Schiller García, 2026a] Schiller García, J. "AML Observability Series (Parts I–III + OBASHI Analysis)." *FinCrime Watchdog*, April 2026. watchdog.endvater.de

[Schiller García, 2026b] Schiller García, J. "272 Datenpunkte und ein Paradigmenproblem." *BKR* (forthcoming), C.H. Beck Verlag, 2026.

[Jha, 2025] Jha, N. N. "Federated learning for cross-cloud observability: Privacy-preserving model aggregation across distributed cloud platforms." *World Journal of Advanced Research and Reviews*, 2025.

[Garnimitta, 2025] Garnimitta, B. "Federated AI Observability in Multi-Cloud Microservices: A Secure and Scalable Federated Learning Perspective." *International Journal of Engineering and Advanced Technology Studies*, 13(3), 20–31, 2025.

[Chilukala, 2025] Chilukala, S. "Event-Driven Architectures in FinTech: Enabling Real-Time Payment Processing and Settlement." 2025.

[Thompson & Davis, 2023] Thompson, H., Davis, F. "Observability-driven development: A new paradigm for distributed financial systems." 2023.

[Liu et al., 2025] Liu, Y. et al. "Advanced Techniques in Real-Time Monitoring for Financial Transaction Integrity." 2025.

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

[Schiller García, 2026c] Schiller García, J. "FinRegAgents: A Multi-Agent RAG Framework for AI-Assisted Financial Regulatory Audits with Confidence-Aware Validation." *aiXiv*, aixiv.260228.000002, February 2026.

[van der Aalst, 2016] van der Aalst, W. M. P. *Process Mining: Data Science in Action*. 2nd ed., Springer, 2016.

[Fed/OCC, 2011] Board of Governors of the Federal Reserve System / Office of the Comptroller of the Currency. "Supervisory Guidance on Model Risk Management (SR 11-7)." April 2011.

[Baesens et al., 2021] Baesens, B., Hillard, R., Soetemans, M. *Analytics and AI for AML and Financial Crime Detection*. Wiley, 2021.

This paper was developed with AI-assisted drafting (Claude, Anthropic) under human editorial oversight by the author. The underlying research, conceptual framework, case interpretation, Transformation Spiral model, and all editorial decisions are the author's own work.