Assurance for Agentic Finance

When Autonomous Agents Hold Money: Closing the Governance Gap in Agentic Payments

Robert Jeffs

Version 1.4 — August 2026

Correspondence. robert.j.jeffs@qbrs-labs.com

Non-affiliation. This work is the author's own. It does not represent the views of any employer or client, and nothing in it should be read as investment advice, solicitation, or a performance claim.

Abstract

Autonomous software agents that hold funds and transact without human intervention moved into production between 2025 and 2026, on rails governed by major standards bodies and backed by the payments establishment. The assurance discipline that would let a regulated institution trust, examine, or answer for such an agent has not arrived with them. This paper locates the gap precisely — unverifiable counterparty software, unstandardized delegation of authority, and audit records that establish occurrence but not legitimacy — and argues that the missing methodology already exists in safety-critical systems engineering. It presents an Agentic Finance Assurance Framework translating that discipline into five elements (verifiable delegation, policy as enforceable constraint, consequence-graded assurance cases, runtime invariants with mandated halt semantics, and provenance-grade audit), a minimal graph-native reference model that makes the framework's central questions answerable by query, a worked example from a production autonomous trading system operated by the author, and a five-rung adoption path that docks into the model-risk, third-party-risk, and audit programs institutions already run. The framework is protocol-agnostic by design: the rails will churn; the questions an examiner asks will not.

Keywords: agentic payments, AI governance, assurance case, model risk, provenance, autonomous agents, safety-critical systems

---

1. Executive Summary

Autonomous software agents that hold funds and transact without human intervention are no longer prospective. Between mid-2025 and mid-2026, agentic payments acquired production protocols, foundation governance under the Linux Foundation and the FIDO Alliance, membership rolls spanning the payments establishment, and cumulative transaction counts above 150 million. The protocols are being extended deliberately toward greater autonomy, and exposure to agent-initiated flows will reach regulated institutions through their existing network relationships and vendor stacks whether or not they have an agentic strategy.

What has not arrived is the assurance discipline that would let a regulated institution trust such an agent, examine one, or answer for one. Three problems, routine in established payment systems, are unresolved for autonomous agents: counterparty risk from software whose provenance and behavioral envelope cannot be verified; authorization, where no settled standard expresses the mandate a principal actually delegated or guarantees its revocation; and accountability, where logs record that transactions occurred but not the chain of authority that made them legitimate. The existing governance estate does not reach the problem — model risk management governs software that informs decisions, not software that acts; third-party risk management governs vendors, not autonomous counterparties; smart-contract audit verifies code at a point in time, not conduct over a deployed life. The gap is methodological, not technological.

In April 2026 the International Monetary Fund published a note taking stock of agentic AI in payments, proposing a three-layer separation of intent formation, authorization and control, and settlement [1]. Its analysis is closely aligned with the argument advanced here, and it is explicit about its own scope: it "does not seek to draw definitive conclusions or propose prescriptive policy measures," aiming instead to frame design questions, architectural tensions, and risk channels. That boundary is where this paper begins. The framework presented below adopts the note's layered vocabulary and supplies the prescriptive discipline it declines to offer — a method for assembling, before deployment, the evidence that an agent's authority is bounded and its exercise of that authority reconstructible.

This paper argues that the missing methodology already exists. Civil aviation confronted the same structural problem — software that must be trusted with something irreplaceable, deployed at scale — and answered it not with perfect software but with an assurance discipline: rigor scaled to consequence, explicit claims backed by evidence, enforced behavioral envelopes, mandated responses to violation, and independent examination. The paper translates that discipline into an Agentic Finance Assurance Framework built on five elements: verifiable delegation, in which man-

dates are first-class, revocable artifacts and delegated authority can only narrow along a chain; policy as enforceable constraint, attached to mandates rather than agents; pre-deployment assurance cases graded by financial consequence; runtime invariant monitoring with pre-designed degradation and halt semantics, enforced by a watchdog the agent cannot reconfigure; and provenance-grade audit, captured as a condition of execution.

The framework's provenance core is presented as a graph-native reference model — six node labels and eight relationship types, with the delegation chain as its spine — small enough to be examinable and sufficient to make the central questions answerable by query: for any transaction, who authorized it, through what chain, under what policy, and was it inside its envelope; and for any revoked mandate or superseded policy, the full blast radius of affected transactions. The model doubles as an acceptance test an institution can put to any vendor of agentic capability: show that these queries are answerable from your records.

The disciplines are not hypothetical. The paper describes their operation in a production quantitative trading system built and operated by the author, where separation of duties, an independently enforced envelope, a gated assurance pipeline whose record contains more rejections than approvals, and a single authoritative knowledge model have governed autonomous components in live markets for several years. The closing section translates the framework into a five-rung adoption ladder for institutions — from exposure inventory through vendor demands to assured autonomy at consequence — docked into the model-risk, third-party-risk, and audit programs institutions already run. The framework's scope is stated honestly: it governs an institution's own assurance estate, and the paper identifies cross-party attestation — verifiable assurance about the unknown agent on the other side of a transaction — as infrastructure the ecosystem has yet to build.

The rails will churn; the framework is deliberately protocol-agnostic because the questions an examiner asks will not. What it offers is narrower and more valuable than a promise of safe agents: it makes trust in an agent examinable, which is the precondition for extending that trust at all.

2. The Agentic Finance Landscape

2.1 What has arrived

By agentic finance this paper means software agents that initiate, authorize, and settle financial transactions without human intervention at the point of transaction. The category is no longer prospective. Between mid-2025 and mid-2026 it acquired production protocols, institutional membership rolls, and cumulative transaction counts

above 150 million — and it did so along three distinct families of rails, whose differences matter less for our purposes than what they share.

The first family is stablecoin-native. The x402 protocol, introduced by Coinbase and Cloudflare in May 2025, embeds payment into the web's own fabric: an agent requesting a service receives an HTTP 402 "Payment Required" response and settles in stablecoin within the request flow, without accounts or cards [2]. Governance of the protocol was transferred to a foundation under the Linux Foundation in April 2026, with a launch membership that reads as a cross-section of the payments establishment — card networks, banks-adjacent processors, cloud providers, and exchanges among them [3]. Cumulative settlement on the protocol is discussed in Section 2.2, where the transaction data is read against its own methodology.

The second family is the mandate protocol. Google's Agent Payments Protocol (AP2), announced in September 2025 with more than sixty payments and technology partners, is payment-method-agnostic: its central construct is the *mandate*, a cryptographically signed artifact recording that a user authorized an agent to transact within stated limits [4]. Google donated AP2 to the FIDO Alliance in April 2026; the accompanying v0.2 release added explicit support for "Human Not Present" transactions — agents executing purchases autonomously under pre-authorized instructions [5]. The direction of travel is unambiguous: the protocols are being extended toward more autonomy, not less.

The third family is the card networks' own. Mastercard's Agent Pay extends network tokenization to agent-initiated flows through Agentic Tokens and a Verifiable Intent artifact; by 2026 it had grown a machine-to-machine payments offering for programmatic, fraction-of-a-cent transactions [6]. Visa's Intelligent Commerce program, with its Trusted Agent Protocol for agent identity and verification, reported over a hundred partners and shipped a consolidated merchant integration supporting four different agent-commerce protocols in a single product [7]. The networks have also moved toward interoperability with the mandate family: Mastercard's Verifiable Intent was co-developed with Google and is AP2-compatible, the two standards dividing the problem between them — AP2 defining how consent and delegation are expressed, Verifiable Intent defining how that consent is represented and verified as evidence [8].

2.2 Adoption, correctly read

The protocol layer has expanded faster than any account written a year ago anticipated, and it has done so in a direction that complicates simple adoption narratives.

Google's Universal Commerce Protocol launched in January 2026, providing a shared grammar for discovery and comparison and powering native checkout within Search's AI mode and Gemini [1][9]. The Agentic Commerce Protocol, from OpenAI

and Stripe, underpins in-conversation purchasing and introduced a four percent transaction fee for agent-led conversions in early 2026 — an explicit pricing of agency itself as a distinct economic function [1]. Stripe and Tempo Labs have published the Machine Payments Protocol, purpose-built for machine-to-machine settlement rather than extending HTTP [1]. Visa's Trusted Agent Protocol, Visa Intelligent Commerce, and Mastercard's Agent Pay — extended to machine-to-machine flows in the second quarter of 2026 [6] — converge on Know Your Agent: registration, cryptographic signature, and network tokenization sufficient to distinguish a legitimate agent from a malicious one. Amazon's delegated purchasing lets its assistant transact on external sites [1][10]. PayPal, following its 2026 acquisition of Cymbio, positions itself as a trust layer for independent agents while preserving merchant-of-record status for retailers [1][11].

Alongside these, a set of on-chain authorization primitives has matured to the point of institutional citation: verifiable off-chain claims consumable by policy engines, modular smart accounts enforcing spend limits and velocity and counterparty restrictions at the wallet layer, and registries for agent identity, validation and reputation [12].

Against that expansion, the transaction data rewards careful reading. The x402 protocol, which carries by far the most deployed agentic payment traffic, had accumulated roughly 157 million settled transactions by July 2026, across seven chains and eighteen tracked facilitators, moving approximately $41 million in stablecoin volume [13]. That is an average of under thirty cents per transaction. Monthly volume peaked in late 2025 and contracted sharply thereafter; on one chain, weekly transactions fell by more than ninety percent between late December 2025 and early February 2026 [14].

The contraction is easy to misread in either direction, and the reason is that the peak did not measure what a headline adoption figure implies it measured. On-chain analysis by Artemis, applying a filter for wallets repeatedly transacting with themselves, classified roughly 48 percent of x402 transactions and 81 percent of transferred value as non-organic as of December 2025 [15]. The trackers do not agree with each other. Published figures for the same periods differ materially between x402.org, Allium and Artemis, according to methodology — what counts as a transaction, which chains are included, and whether flagged activity is netted out. That disagreement is not a defect in the data so much as a demonstration of the point: a metric that varies by a factor depending on who computes it is not a metric an institution can govern against. Independent analysis of the same period attributes much of the surge to token speculation rather than agents purchasing services [16]. An adoption curve assembled from these counts is therefore measuring something other than institutional adoption, in both its rise and its fall.

Durability, where it exists, shows up elsewhere. The protocol has been absorbed natively into Google's AP2, Cloudflare's agent runtime, Stripe, and AWS Bedrock, and

the x402 Foundation moved to operational status under the Linux Foundation in 2026 with a membership spanning the card networks, the major acquirers, and the large cloud providers [3]. Institutional commitment of that kind is evidence that a protocol is becoming infrastructure. Transaction counts are not, and were never going to be. The question that matters to a regulated institution is not how many transactions crossed a rail last month but whether the authority under which any of them crossed can be reconstructed afterwards. That question is unaffected by the curve, in either direction [17].

National regulators have begun to address agentic deployment directly. Singapore's Infocomm Media Development Authority published a model governance framework for agentic AI in 2026, built on ex-ante risk assessment, explicit limits on agent autonomy and tool access, defined human approval checkpoints, and lifecycle technical controls [18]. It is the closest thing yet to a prescriptive national position, and it is structurally compatible with the framework proposed here rather than in competition with it. The Financial Stability Board has begun monitoring AI adoption and associated vulnerabilities in the financial sector [19], and the European Union's AI Act applies to a subset of these deployments by construction [20].

Supervisory bodies have moved further still. IOSCO finalized a Supervisory Toolkit for AI Use in Capital Markets in May 2026, covering the full lifecycle of an AI system and extending explicitly to agentic techniques [21]. The Financial Stability Board consulted in June 2026 on twelve sound practices for the responsible adoption of AI, with comments closing that July and a final report expected in October 2026 [22], and NIST's AI Agent Standards Initiative, announced in February 2026, gives its first pillar as facilitating industry-led development of agent standards and U.S. leadership in international standards bodies [23]. Read together, these state the requirement with increasing precision and leave the same thing unaddressed: what a firm produces to evidence that it has met them.

A further observation follows from the shape of the expansion rather than its size. Every one of the protocol efforts surveyed above addresses authorization — who may act, within what limits — and several provide for auditability as a recording function. None takes as its problem the question that follows: how a firm demonstrates, before deployment and to a third party that does not trust it, that the limits it has specified are completely enforced and continuously evidenced. That is a different discipline, and the gap is structural rather than incidental — it persists across the whole of that layer, which is what makes it worth a framework rather than a fix.

2.3 Institutions are already in the room

It would misread the landscape to treat agentic finance as a crypto-native phenomenon that regulated institutions can observe from a distance. The membership

of the protocol foundations, the card networks' product launches, and the processors' integrations place the payments establishment inside every rail family simultaneously. Cloud providers have shipped agent-payment capabilities into their enterprise AI platforms. Consumer-facing autonomous purchasing has been announced for mainstream commerce surfaces. The practical consequence for a bank, broker, or payments institution is that exposure to agent-initiated flows will arrive through existing network relationships and vendor stacks whether or not the institution has an agentic strategy — and its risk, compliance, and audit functions will be asked questions about transactions that current frameworks were not designed to answer.

2.4 Churn as a planning assumption

Finally, the protocol landscape itself is unsettled, and a governance approach must assume it stays that way. In eighteen months the field produced at least half a dozen overlapping standards — stablecoin-native settlement, mandate protocols, network token extensions, agent-identity schemes, and commerce protocols from AI platform vendors — with stewardship migrating between companies, foundations, and standards bodies, and interoperability bridges appearing between previously rival stacks [24]. The churn is no longer a prediction: one of these protocols is already contracting, and the mandate primitive has been reimplemented in at least three mutually incompatible forms. Some of these will consolidate; some will not survive. An institution that builds its governance around any single protocol inherits that protocol's fate. The framework developed in the remainder of this paper is therefore deliberately protocol-agnostic: mandates, envelopes, assurance cases, and provenance are requirements on *any* rail, and the reference model of Section 5 treats the protocol layer as an implementation detail beneath a stable governance vocabulary. The rails will churn. The questions an examiner asks will not.

3. The Assurance Gap

3.1 From software that recommends to software that transacts

The financial industry has three decades of institutional experience governing software that *informs* decisions. Model risk management as codified in SR 11-7 and its successors [25] rests on a structural assumption so fundamental that it is rarely stated: a model produces an output, and a human — or a process ultimately accountable to a human — decides what to do with it. Validation, ongoing monitoring, outcomes analysis, and effective challenge are all designed around this boundary between recommendation and action. The boundary is where accountability lives.

Agentic finance dissolves that boundary. An autonomous agent that holds funds, evaluates conditions, and executes transactions is not a model whose output a human re-

views; it is a principal actor in the payment system. The decision and the execution are the same computational event. When that event moves money between parties, every assumption underlying the recommendation-era governance stack — that there is a moment of human judgment to govern, a decision point to document, an override that precedes rather than follows the action — quietly fails.

This is not an argument that agentic payments are imprudent. The rails are real, the institutional commitment behind them is real, and the efficiency case is legitimate. It is an argument that a category of software has arrived for which the industry's assurance discipline was never designed, and that the gap between what the technology does and what the governance assumes is currently absorbed by the parties least equipped to price it: counterparties, principals, and ultimately the institutions that touch these flows.

3.2 Oversight as an engineering obligation

The gap described above is not only a technical one. In Delaware, whose corporate law governs most large US financial institutions, directors owe a duty of oversight defined in *Caremark* as the obligation to implement reporting and information-system controls, and to monitor operations once those controls exist [26]. The duty is not framed in terms of outcomes. It is framed in terms of whether a system capable of surfacing the problem was built and attended to.

That framing has been largely theoretical for thirty years, because oversight-duty claims are difficult to plead and were never tested at trial. That changed in July 2025, when shareholders of Meta Platforms brought the first *Caremark* claim to reach trial, arguing that directors had failed to oversee compliance with a 2012 consent order and had thereby exposed the company to a $5 billion regulatory penalty and roughly $3 billion in associated costs [27]. The trial was halted on its second day and settled for $190 million, paid by directors' and officers' insurance, without admission of liability [28].

Two features of that outcome are relevant here, and neither depends on any view about the merits.

The first is that the question at issue — what the board knew, when, and whether a system existed to tell them — took more than seven years to litigate and was never actually answered. It was reconstructed adversarially from documents and testimony, at a cost that consumed a meaningful fraction of what was recovered. An institution that can answer that question from its own records in an afternoon is in a materially different position from one that cannot, and the difference is not legal counsel. It is whether the controls emitted an auditable record while they were operating.

The second is the ordering of harm. The individuals whose data was taken received nothing from this proceeding; a derivative action compensates the company, not the people the company's failure reached. The regulatory penalty arrived years after the conduct. The shareholder recovery, at roughly three percent of the sum sought and funded by insurance, is unlikely to have altered any director's incentives. Harm to individuals came first, was largest, and was addressed last and least.

Agentic finance does not change this structure. It accelerates it. An autonomous agent transacting continuously against a policy nobody can reconstruct produces the same evidentiary problem at machine speed, distributed across counterparties who never agreed to bear it.

The institutional memory that regulated firms bring to this is not encouraging. The last decade's largest data-governance failures were not failures of intent or of technical capability; they were failures to build a system that could see what was happening in time to act on it. Agentic payment rails are being adopted at a pace that assumes this pattern will not repeat.

3.3 Three unresolved problems

The gap is concrete. Three problems, each routine in established payment systems, are unresolved for autonomous agents.

Counterparty risk from unverified software. When an institution transacts with a human-operated entity, centuries of legal and commercial infrastructure stand behind the interaction: identity, capacity, recourse. When it transacts with an autonomous agent, the counterparty is, in operational terms, a piece of software whose provenance, integrity, behavioral envelope, and operator are typically unverified and often unverifiable. There is no airworthiness certificate for a financial agent — no attestation that the software controlling funds on the other side of a transaction was built to any standard, behaves within any declared envelope, or will fail in any predictable way. Each party to an agentic transaction is implicitly extending trust to an artifact about which it can verify almost nothing.

Authorization and delegation. Agency law has well-developed answers to the question of what an agent may do on behalf of a principal. Those answers exist as legal doctrine, not as verifiable machine-readable form. Current agentic-payment implementations express authorization as, at best, protocol-level permissions: a key can sign, a wallet can spend. The mandate protocols of Section 2.1 go furthest, expressing transaction-scoped authorization in cryptographically signed form. What none of them expresses is the *governance mandate* — the scope, limits, conditions, and revocability of the authority the principal actually intended to delegate, bound to policy and durable across rails. The distinction is drawn precisely in Section 4.3. The

problem compounds when agents compose: an agent delegating a subtask to another agent creates a chain of authority in which the original principal's intent must survive translation across systems that have converged on no common representation of what was authorized. There is no settled standard for expressing delegation, verifying it at transaction time, or revoking it with confidence that revocation propagates.

Accountability and reconstruction. After an adverse event — an erroneous payment, a drained wallet, an agent transacting outside its intended scope — the governing questions are the same ones regulators and courts have always asked: who authorized this, under what policy, and was the action within the authority granted? Contemporary agent infrastructure produces logs, but logs are not provenance. A log records that a transaction occurred; provenance records the chain of delegation, policy, and authorization that made it legitimate or illegitimate. Without provenance-grade records, post-hoc reconstruction becomes forensic archaeology, liability allocation becomes contested, and the supervisory examination of an agentic-finance program has nothing rigorous to examine.

This is not a gap identified only by its critics. The IMF's April 2026 note classifies *authorization traceability failures* as a distinct risk category, attributing it to payment service providers and platforms that rely on mandate-based authorization without robust auditability. It records the cost as falling on those providers through liability exposure, on users through disputed payments, and on courts and regulators through legal uncertainty — and it marks the category as a market failure justifying policy intervention, describing the underlying condition as legal infrastructure mismatch and incomplete contracting [1].

The note's own prescription, in its discussion of the authorization layer, is that "broader and legally workable concepts of authorization (grounded in verifiable mandates, scope limitations, and auditability) are needed as agentic payment models evolve."

That sentence is a requirement specification. Verifiable mandates and scope limitations are largely supplied by the protocol layer surveyed in Section 2. Auditability is not, because auditability is not a protocol property. It is a property of what a deployment records while it is operating, and of whether those records can be assembled afterwards into an argument that a third party will accept. Supplying that is an engineering discipline, and it is the subject of the remainder of this paper.

3.4 Why the existing stack does not reach

It is tempting to assume the existing governance apparatus can be extended to cover autonomous agents. Examined closely, each candidate falls short of the problem.

Model risk management frameworks evaluate conceptual soundness, monitor performance, and analyze outcomes — all valuable, none sufficient. They govern the quality of a model's judgments, not the safety of an actor's conduct. Nothing in the model-validation lifecycle addresses delegation chains, transaction-time authorization, or the behavior of software that acts without an intervening decision-maker.

Third-party risk management, as framed in interagency guidance [29], governs vendors: due diligence, contractual controls, ongoing oversight of a known counterparty with legal personality. An autonomous agent encountered as a transaction counterparty is not a vendor. There is no diligence process, no contract negotiation, and frequently no identifiable party behind it at the moment of the transaction.

Smart-contract auditing, the crypto-native answer, verifies that code matches specification at a point in time. It says nothing about the operational assurance of an agent's behavior over its deployed life — under distribution shift, under adversarial pressure, under the accumulation of delegated authority the auditor never saw. Code correctness is necessary and radically incomplete.

Nor is the gap jurisdictional. The analysis above is anchored in U.S. supervisory guidance because that is where the largest share of early institutional exposure sits, but the pattern repeats across regimes. The UK PRA's supervisory statement on model risk (SS1/23) modernizes model-risk principles while retaining the model-as-advisor frame [30]; the EU's Digital Operational Resilience Act imposes ICT-risk and third-party oversight obligations that autonomous agents in the transaction path directly implicate [31], and the EU AI Act's high-risk provisions govern the development and oversight of AI systems generally, though the obligations bearing most directly on deployed agents do not apply until December 2027 [20]; Singapore's FEAT principles articulate fairness, ethics, accountability and transparency expectations for AI in financial services [32]. Each is serious work; none yet articulates an assurance discipline for software that autonomously transacts. The gap this paper addresses is common to all of these regimes, which is one reason the framework that follows is deliberately jurisdiction-neutral.

The common failure is structural: every one of these disciplines governs either *artifacts* (models, code) or *organizations* (vendors, counterparties). Autonomous agents are neither and both — deployed artifacts exercising ongoing agency. None of the disciplines surveyed above takes that category as its object. That is not an oversight any one of them could patch; it is a consequence of where each drew its boundary.

3.5 Adjacent work

The nearest neighbor to this work in the recent literature is AURA, an agent risk assessment framework cited in the IMF's mitigation discussion for its treatment of

human-in-the-loop supervision [33]. AURA scores individual agent actions across configurable risk dimensions, aggregating them into a normalized measure that gates approval, escalation, or human review. Its assessments are produced by language models at runtime, which places it and the framework presented here on opposite sides of the boundary described in Section 4.4: AURA supplies graded contextual judgment about whether an action is risky, where the framework supplies a constraint that holds irrespective of judgment. The two compose rather than compete — an action-level risk score is a plausible input to the consequence grading of Section 4.5 — but they are not substitutes, and the difference is not one of emphasis. AURA has no representation of authority: no principal, no mandate, no narrowing rule, and consequently no answer to the question of what a revocation invalidates. Its own survey of governance frameworks draws on NIST, ISO/IEC 42001 and the EU AI Act, and it identifies domain-specific specialization for finance as unfinished work.

Related work on agent identity and intent verification in trustless settings [34], and comparative analysis of inter-agent trust models across agentic web protocols [35], addresses the authorization layer this framework assumes rather than the assurance layer above it. Recent security analyses of x402 deployments document the vulnerability classes that a deployment-level assurance argument must account for [17][36].

3.6 The aviation precedent

Finance is not the first industry to confront software it must trust with something irreplaceable. Civil aviation faced the same structural problem half a century ago: flight-critical software whose failure kills people, deployed at scale, built by fallible organizations. Aviation's answer was not perfect software. It was an assurance discipline [37][38][39].

That discipline has a recognizable shape. Development rigor is scaled to hazard severity, so the effort invested in assurance is proportional to the consequence of failure. Safety is assessed at the system level, not the component level, because hazards emerge from interaction. Claims of safety are made explicit and structured — an assurance case that states what is claimed, on what evidence, under what assumptions — rather than left implicit in test results. Verification is independent of development. Behavioral envelopes are declared, and departure from the envelope has mandated, pre-designed consequences: degrade, revert, halt. And an authority external to the builder examines the case before the system carries passengers.

None of this made aviation software infallible. It made aviation software *trustworthy in a specific, examinable sense*: every operational aircraft flies inside a web of explicit claims, evidence, and enforced envelopes, and when something goes wrong, the discipline produces the records to determine why.

Finance now faces the same category of problem — software that must be trusted with money, deployed at scale, transacting autonomously — and does not yet have an equivalent discipline. The gap is not technological. The rails function; the cryptography holds; the protocols settle. The gap is methodological: the industry deploying autonomous financial agents has not built the assurance practice that lets a regulated institution trust one, examine one, or answer for one.

Finance has its own accident record, and it points where aviation points. On 1 August 2012 Knight Capital deployed new code to seven of its eight order-routing servers; on the eighth, a repurposed flag reactivated dormant code that had not been used in years and could not recognize when orders had been filled. In roughly forty-five minutes the router sent more than four million orders while attempting to fill 212 customer orders, and the firm took a loss of more than $460 million. The SEC's subsequent proceeding found no written deployment procedures and no requirement that a second technician review the change [40]. Seventeen years earlier, Barings Bank collapsed because one employee controlled both trading and settlement, so his positions were recorded by the party holding them; £827 million of unauthorized losses, more than twice the group's capital, ended a bank founded in 1762, and an internal audit had identified the dual-role problem the previous year without the finding being acted upon [41]. Neither was a failure of code correctness. Knight was a failure of deployment discipline and halt semantics. Barings was a failure of authorization independent of the party exercising it, and of records independent of the actor they described. These are the failure modes an autonomous agent reproduces, without the human latency that in both cases still left hours or years for someone to notice.

3.7 The ledger objection

A specific objection recurs wherever settlement occurs on-chain: that distributed ledgers already solve this. The record is immutable, cryptographically verifiable, and independently held. What could an assurance discipline add?

The naive form mistakes a category. A ledger establishes that a transaction occurred, that it was signed by a particular key, and that the record has not been altered. It establishes nothing about whether the party controlling that key was authorized to act, on whose behalf, within what limits, or whether those limits were current at the moment of execution. Integrity of record is not legitimacy of act. The decision to transact is made off-chain by software the ledger never observes; pseudonymous key control identifies a signer, not a principal; and auditability of a ledger is auditability of settlement, not of authority.

The sophisticated form is stronger, and is closer to a rival framework than to an objection: the claim is not that the ledger verifies the agent but that programmable

constraints do. Spending limits, multisig thresholds, time locks, and mandates encoded in contract make authorization deterministic and enforced at settlement rather than argued in a document. That deserves respect, because it implements this framework's own enforcement principle in a different substrate. The answer is scope rather than dismissal. Contract-enforced constraints bind what is expressible on-chain, and only on the payment leg. They cannot encode suitability, sanctions context, fiduciary obligation, or the cross-party boundary described in Section 4.7, and they do not constrain the reasoning that selects which permitted transaction to execute. An agent that stays perfectly inside its spending limit while being manipulated into the wrong counterparty produces a loss that is fully compliant, fully enforced, and fully immutable. Enforcement without an assurance argument is incomplete, and the strongest version of the objection demonstrates it.

The concession is real. Where settlement is on-chain, the ledger is a better evidence substrate than most institutional logging, and the reference model of Section 5 anchors evidence to it rather than duplicating it.

The last observation inverts the objection. Irreversible settlement removes the recourse layer that absorbs assurance failures elsewhere in finance: a mistaken wire can be recalled, a mistaken card payment charged back, a mistaken on-chain transfer neither. The environment its proponents cite as evidence that assurance is unnecessary is the environment in which this framework's demand is most acute. On-chain agentic settlement is not an exception to consequence-graded assurance. It is its maximum case.

3.8 What an adequate discipline must provide

Naming the gap defines the requirement. An assurance discipline adequate to agentic finance must provide, at minimum: a verifiable representation of delegated authority, so that mandate is checkable at transaction time and revocable in practice; policy expressed as enforceable constraint — spending limits, counterparty scope, halt conditions — rather than as prose that software cannot obey; a pre-deployment assurance case that states explicitly what is claimed about the agent's behavior and on what evidence; runtime monitoring of declared invariants, with pre-designed degradation and halt semantics when an invariant fails; and audit records at provenance grade, sufficient to reconstruct the chain from principal to policy to delegation to transaction after the fact.

These are not novel inventions. Each is a translation of a practice that safety-critical engineering has operated for decades. The remainder of this paper presents that translation as a coherent framework, and a reference model for implementing its provenance core.

4. The Agentic Finance Assurance Framework

4.1 Position within the layered model

The three-layer separation proposed by the IMF — intent and orchestration, control and authorization, settlement — is adopted here without modification, and it is worth stating plainly where the framework sits within it.

The framework is not a Layer 1 technology. It does not constrain how an agent reasons, plans, or forms intent, and it makes no claim to make probabilistic systems predictable. Nor is it a Layer 3 technology; settlement finality is a property of financial market infrastructures and is not this framework's to alter.

The framework operates at the **Layer 1 to Layer 2 boundary**: the point at which structured intent produced by an adaptive system is presented for deterministic authorization. The layered model establishes that this boundary must exist and must be rules-bound. It does not address how an institution demonstrates, in advance and to someone who does not trust it, that the boundary is correctly specified, completely enforced, and continuously evidenced.

That demonstration is an assurance case, constructed by the disciplines described in Section 3.6.

4.2 Design principles

The framework presented here is a translation, not an invention. It is intended to complement, not replace, general AI risk-management guidance such as the NIST AI Risk Management Framework [42], which it particularizes to the case of software that transacts. Every element adapts a practice that safety-critical engineering has operated, refined, and audited for decades, and the translation is governed by four principles.

Proportionality. Assurance effort scales with consequence. An agent authorized to spend small sums on reversible purchases does not warrant the assurance regime of an agent moving institutional funds across settlement finality. The framework therefore defines graduated assurance levels rather than a single bar, for the same reason aviation scales development rigor to hazard severity: uniform maximal rigor is unaffordable, and uniform minimal rigor is unacceptable.

Explicitness. Every claim about an agent's behavior is stated, not implied. What the agent may do, what it must never do, what happens when it approaches a limit, and what evidence supports each of these — all of it exists as examinable artifact rather than as institutional folklore or code comments.

Enforcement. Policy is expressed in forms software can obey and infrastructure can verify. A prose policy an agent cannot mechanically check is a hope, not a control.

Independence. The party verifying an agent's assurance case is not the party that built the agent. This is the oldest lesson in certification, and the one most cheaply discarded under commercial pressure.

The framework comprises five elements, corresponding to the requirements established in Section 3.8. They are presented individually below and as a lifecycle in Section 4.8.

4.3 Verifiable delegation

The foundation of the framework is the representation of authority. A *mandate* is a structured, signed, revocable artifact recording that a principal has delegated a defined scope of financial authority to an agent: what may be spent, with whom, under what conditions, until when, and subject to which policies. Mandates are first-class objects — created, versioned, suspended, and revoked as deliberate acts, each leaving a record.

A note on lineage and nomenclature is owed here. The Agent Payments Protocol introduced in Section 2 also centers a construct called a mandate — a signed artifact of user authorization [4] — and the convergence of vocabulary is no accident: both designs recognize that agentic payments fail without a verifiable record of delegated intent. The constructs differ in kind. A protocol mandate is a transaction-scoped payment artifact; the mandate defined here is an institutional governance object — bound to policy, graded by an assurance case, subject to a composition rule, and embedded in a provenance record that outlives any single rail. The two are complementary: a protocol mandate is one form of evidence a governance mandate can carry. Nor is the narrowing rule introduced below novel to this paper; it is the attenuation property long established in the capability-security literature [43], applied here to financial authority. The framework's contribution is the assembly, not the parts — which is the point of a translation.

Two properties give mandates their force. First, they are *checkable at transaction time*: before an agent's transaction executes, the chain of mandates from the acting agent back to a human or institutional principal can be mechanically verified — every link current, none revoked, the proposed action inside every link's scope. Second, they obey a *narrowing rule under composition*: when an agent delegates a subtask to another agent, the delegated authority can only be a subset of the authority the delegating agent itself holds. Authority may narrow along a delegation chain; it may never widen. This single rule eliminates a large class of failure in composed agent systems, in which accumulated delegation quietly exceeds anything a principal intended. Checkability

is doing real work in that claim: containment of one scope within another is trivial for numeric limits and undecidable for arbitrarily expressive conditions, so mandates must be written in a constraint language chosen for decidable comparison — a requirement modern authorization languages demonstrate is practical rather than aspirational [44].

Revocation is designed with the same seriousness as grant. A revoked mandate must fail verification immediately and everywhere — which imposes real requirements on how mandate state is stored and propagated, requirements taken up by the reference model in Section 5. One of them should be named now: a centralized provenance store makes revocation *recordable* and *checkable*; it does not by itself make propagation instantaneous across every enforcement point, and the residual window between revocation and universal enforcement is an implementation obligation the framework surfaces rather than solves. An assurance case for a consequential agent must state the bound on that window and the evidence for it.

4.4 Policy as enforceable constraint

Where a mandate records *that* authority was granted, policy defines *the envelope within which it may be exercised*. The framework requires policy to be expressed as machine-enforceable constraint: spending limits per transaction and per period; counterparty scope, whether as allowlist, verification requirement, or category restriction; temporal conditions; escalation thresholds above which human confirmation is required; and halt conditions under which the agent must cease transacting entirely.

Three rules govern policy in the framework. Policy *attaches to the mandate, not to the agent* — the same agent may operate under different envelopes for different principals, and an agent's capabilities are never confused with its permissions. Policy *composes restrictively*: where multiple policies apply along a delegation chain, the most restrictive binding governs, so no composition of policies can be more permissive than any of its parts. And policy is *versioned with provenance*: every change to an envelope is itself a recorded, attributable event, because the question "what policy was in force at the time of the transaction" must have exactly one answer.

4.5 The pre-deployment assurance case

Before an agent transacts with real funds, its operator produces an assurance case: a structured argument that the agent will behave within its declared envelope, decomposed into explicit claims, the evidence supporting each claim, and the assumptions on which the argument rests. This is the direct descendant of the safety case that certification regimes have required of flight-critical systems for decades — and its

nearest contemporary relative is the assurance-case practice now standardized for autonomous products generally [45] — and its value is identical: it converts diffuse confidence into an examinable artifact that a reviewer, a counterparty, a risk committee, or a supervisor can interrogate.

Proportionality enters through *agent assurance levels*. The framework grades required rigor by financial consequence, assessed on three axes: the value at risk under the agent's mandates, the reversibility of its transactions, and the scope of its autonomy — how long and how far it acts between human touchpoints. A low-consequence agent may warrant testing evidence and documented policy conformance; a high-consequence agent warrants independent verification, adversarial evaluation against its envelope, and demonstration of its halt behavior under fault injection. The graduation is the point: it gives institutions a defensible answer to "how much assurance is enough," which is otherwise decided by budget and optimism.

Consequence grading determines where a human sits relative to the transaction, not only how much evidence the case must carry. At low consequence, human involvement is retrospective: the agent acts and the record is reviewed. As consequence rises, approval moves in front of execution — a confirmation gate at a value threshold, a counterparty outside the standing allowlist, or a first transaction with a new party. At the top of the scale the boundary is categorical rather than numeric: actions that cannot be undone sit outside every standing delegation and require explicit, per-instance authorization. Reversibility, not value, is the cleaner criterion at that boundary, and it is the one distinction that does the most practical governance work. Where these gates sit is stated in the assurance case and enforced by the same layer that enforces the envelope, so that a gate is a constraint rather than a convention.

Evidence for an assurance case is broader than testing. Behavioral evaluation under distribution shift, adversarial probing of the policy envelope, verification that the enforcement layer cannot be bypassed by the agent it constrains, and review of the mandate and revocation machinery all belong in the case for consequential agents. What the case explicitly is not: a claim that the agent is intelligent, profitable, or correct in its judgments. It is a claim that the agent is *contained* — that whatever it decides, its actions remain inside a declared and enforced envelope. Two consequences of that position deserve plain statement. First, for the stochastic, learned agents actually driving this market — systems whose internal behavior resists the specification-and-verification treatment deterministic avionics software receives — the weight of the assurance case falls predominantly on the enforcement layer rather than the agent. Grading agents by assurance level is, in substance, grading envelopes; the paper regards this not as a weakness but as the honest center of the framework. Second, containment defines a residual attack class it cannot close: an adversary who manipulates an agent — through poisoned inputs, adversarial content, or prompt injection [46] —

into actions that are envelope-*compliant* yet harmful to the principal. Spending the full mandate on the wrong counterparty inside the allowlist violates no invariant. The assurance case for a consequential agent must therefore address this class explicitly: through provenance controls on the agent's inputs, anomaly detection tuned to intent rather than limits, envelopes sized to the harm a fully manipulated agent could do, and human confirmation gates at consequence thresholds. Where those mitigations cannot be evidenced, the honest assurance conclusion is a smaller envelope.

4.6 Runtime invariants and mandated degradation

An assurance case is an argument about the future made before deployment; runtime monitoring is the discipline of checking that argument continuously against reality. The framework requires each agent to operate under *declared invariants* — conditions that must hold throughout operation, drawn directly from its policy envelope and assurance case: cumulative spend within bounds, counterparties within scope, transaction rates within declared norms, delegation chains verifiable, enforcement layer responsive.

Invariant violation has *pre-designed consequences*. The framework mandates a degradation ladder specified before deployment: constrain (tighten the envelope and continue), suspend (cease initiating transactions, hold state, await review), and halt (cease all activity, preserve records, require explicit human re-authorization to resume). Which rung applies to which violation is decided at design time, in the assurance case — never improvised during an incident. A halted agent is not a failed system; it is the system working. This inversion, familiar from every safety-critical domain, is the single most transferable habit of mind the framework asks of its adopters.

Monitoring independence completes the element. The watchdog that evaluates invariants must be separate from the agent it monitors — a distinct process, with the authority to suspend or halt the agent and the inability to be reconfigured by it. An agent that monitors itself is unmonitored.

A question follows from the watchdog's independence requirement, and the FSB has already raised it. Its sound practices note that effective monitoring and detection of agentic errors may in some cases require augmentation with another AI agent or other forms of AI [22]. So: what if the monitor is itself an agent? Monitoring at machine speed is difficult to staff any other way. The framework's position is that an agentic monitor is permissible as an instrument and inadmissible as evidence. A model that scores, flags, or classifies may direct human attention and may trigger a deterministic gate; it may not be the thing that establishes the claim. The reason is not distrust of models but the structure of an assurance argument. Evidence must be checkable

by something that does not share the failure modes of what it evaluates, and a language model monitoring a language model shares them exactly: the same susceptibility to injection, the same distribution shift, the same non-determinism under identical inputs. Where an agentic monitor is used, the assurance case must state what its outputs are evidence of — which is its own operation — and the invariant that actually gates behavior must remain a deterministic check the monitored agent cannot reconfigure. Assurance of the assurer is not an infinite regress, provided the regress terminates in something that does not reason.

4.7 Provenance-grade audit

The final element makes the first four reconstructable. The framework requires that records be kept not as event logs but as *provenance*: a connected record linking each transaction to the mandate under which it executed, the policy in force, the delegation chain that authorized it, the invariant status at execution time, and the principal at the chain's origin. The test of adequacy is concrete: for any transaction, at any later time, an examiner must be able to answer — from records alone — who authorized it, under what policy, within what chain of delegation, and whether it was inside the envelope. And for any adverse event, the records must support the inverse query: every transaction whose authorization chain passed through a given revoked mandate, a given policy version, a given compromised agent.

These are graph queries in their natural form, which is why the reference model in Section 5 is graph-native. The point here is architectural: provenance is captured at transaction time as a condition of execution, not reconstructed afterward from logs. What is not recorded at the moment of action is, for assurance purposes, permanently unknown.

A log and a provenance record are different artifacts, and the difference is not one of completeness. A log records that something happened, in time order, usually emitted by the component that did it. Provenance records what an output depends on — which authority, which inputs, which prior decisions — as a dependency structure rather than a sequence. A log answers what occurred and when; provenance answers by what authority and on what basis. The operational test is the query: from a log you reconstruct a timeline, from provenance you run reachability. If this mandate is revoked, which actions become unauthorized? If this data source is found corrupt, which decisions rest on it? The revocation-propagation requirement stated above needs the second and cannot be satisfied by the first. The idea of a connected artifact chain is mature engineering practice rather than a novelty — aerospace programs have maintained one across the lifecycle for decades — and what makes it evidence rather than testimony is that each link references an artifact held by someone

other than the agent. A self-reported log is testimony. Barings failed on exactly that distinction: the positions were recorded by the trader who held them.

This is also the sharpest available statement of the cross-party boundary. At a party boundary an institution can generally obtain a counterparty's logs, its attestations, or a settlement record. What it cannot obtain is that counterparty's provenance — the authority chain inside the other organization that made its agent's action legitimate. The framework does not decline the cross-party problem as too hard. It identifies precisely which artifact class stops at the boundary, and why.

It follows that a distributed ledger record, whatever its integrity properties, is a log. It is an unusually good one — tamper-evident, independently held, a stronger substrate than most institutional logging. It has no provenance edges.

4.8 The framework as lifecycle

Assembled, the five elements form a lifecycle. At *design*, mandates and policy envelopes are specified as artifacts. At *assurance*, the case is built and independently verified to the level the agent's consequence demands. At *deployment*, enforcement and monitoring infrastructure are in place before the first funded transaction. In *operation*, invariants are continuously evaluated and every transaction extends the provenance record. On *violation or incident*, the mandated degradation ladder executes and the provenance record supports full reconstruction. Findings feed revised envelopes and a revised assurance case, and the cycle continues.

Three boundaries should be stated plainly. The framework does not make agents more capable, more profitable, or better at judgment; it constrains conduct, it does not improve cognition. It does not replace regulation or legal doctrine; it produces the artifacts — mandates, cases, envelopes, provenance — that give regulation and doctrine something rigorous to grip. And it governs an institution's *own* assurance estate: of the three problems posed in Section 3.3, delegation and accountability are resolved within these mechanisms, while counterparty risk — the unverified agent on the other side of a transaction — is addressed only where the counterparty can be made to answer, as a vendor or partner can and an anonymous agent on an open rail cannot. Closing that gap requires attestation infrastructure the ecosystem has not yet built: a portable, verifiable assertion that an agent operates under a stated envelope, backed by an examined assurance case. The framework defines what such an attestation would need to assert; building the infrastructure that carries it is deliberately scoped out of this paper and identified as the field's most consequential open problem. What it offers institutions is narrower and more valuable than a promise of safe agents: it makes trust in an agent *examinable*, which is the precondition for extending that trust at all.

5. A Graph-Native Reference Model

5.1 Why a graph

Section 4.7 established the framework's audit requirement: for any transaction, an examiner must be able to reconstruct — from records alone — the principal, the delegation chain, the mandate, the policy in force, and the envelope status at the moment of execution; and for any adverse event, the inverse query must be answerable. These are questions about *paths through connected records*: chains of delegation, versions of policy, transactions hanging off mandates. Asking them of tabular logs means reassembling the connections at query time, join by join, with the examiner supplying the structure the records failed to keep. A property graph keeps the structure. The chain of authority is not inferred from the data; it *is* the data.

This section presents a minimal reference model in property-graph form, with schema and illustrative queries in Cypher. It is a reference model in the strict sense: the smallest vocabulary sufficient to implement the framework's provenance requirement, intended to be extended in practice, and deliberately independent of any particular payment protocol, agent architecture, or vendor product.

5.2 Design commitments

Three commitments shape the model.

Append-only provenance. Nodes representing acts of governance — grants, delegations, policy versions, revocations, transactions — are immutable once written. Change is represented by new nodes and relationships, never by mutation. A superseded policy is not edited; a new version is created and linked. A revoked mandate is not deleted; a revocation event is attached. The record therefore holds its own history, and "what was true at time T" is a query, not an archaeology project.

State as events. The current status of any mandate or policy is derived from the events attached to it, not stored as a mutable flag. This removes the class of inconsistency in which a status field and the event history disagree — the history is the only authority.

Snapshot at execution. Every transaction node records, at write time, the specific mandate and policy version it executed under and the envelope status at that moment. Section 4.7's architectural rule appears here as schema: provenance is captured as a condition of execution. A transaction with unresolved provenance references is not a record of a governed transaction; it is evidence of an ungoverned one.

5.3 The vocabulary

Six node labels and eight relationship types are sufficient.

```
(:Principal)     A human or institutional authority at the origin of
                 all delegation. Properties: id, kind (human|institution).

(:Agent)         Deployed software capable of transacting.
                 Properties: id, operator, assurance_level.

(:Mandate)       A grant of financial authority. Immutable.
                 Properties: id, scope (structured constraint set),
                 valid_from, valid_to.

(:PolicyVersion) One immutable version of a policy envelope.
                 Properties: id, constraints (structured), created_at.

(:Event)         A governance act: revocation, suspension, halt,
                 resumption. Properties: id, kind, at, actor.

(:Transaction)   An executed financial action. Immutable.
                 Properties: id, amount, counterparty, at,
                 envelope_status.


(:Principal)-[:GRANTED]->(:Mandate)          origin of a chain
(:Mandate)-[:DELEGATED_FROM]->(:Mandate)     child derives from parent
(:Mandate)-[:HELD_BY]->(:Agent)              who exercises it
(:Mandate)-[:BOUND_BY]->(:PolicyVersion)     envelope in force
(:PolicyVersion)-[:SUPERSEDES]->(:PolicyVersion)
(:Event)-[:APPLIES_TO]->(:Mandate)           revocation, suspension
(:Transaction)-[:EXECUTED_UNDER]->(:Mandate)
(:Transaction)-[:UNDER_POLICY]->(:PolicyVersion)
```

The delegation chain is the spine: every mandate either is granted directly by a principal or derives from a parent mandate via `DELEGATED_FROM`, so every chain terminates, by construction, at a `(:Principal)`. The narrowing rule of Section 4.3 becomes a checkable structural property — each child mandate's scope must be contained within its parent's — verifiable when the delegation is written and re-verifiable by any later examiner. Policy attachment to mandates rather than agents (Section 4.4) is likewise visible in the schema: `BOUND_BY` runs from `Mandate`, and an agent's effective envelope is whatever the mandates it holds are bound by.

5.4 The questions, as queries

The model earns its keep in the queries a compliance officer, examiner, or incident responder would actually run. Three illustrate the pattern.

The examiner's question — for a given transaction, produce the full chain of authority:

```
MATCH (t:Transaction {id: $txn_id})-[:EXECUTED_UNDER]->(m:Mandate)
MATCH chain =
↪   (m)-[:DELEGATED_FROM*0..]->(root:Mandate)<-[:GRANTED]-(p:Principal)
MATCH (t)-[:UNDER_POLICY]->(pv:PolicyVersion)
```

```
RETURN p AS principal,
       [n IN nodes(chain) | n.id] AS delegation_chain,
       pv.constraints AS policy_in_force,
       t.envelope_status AS status_at_execution
```

One query answers who authorized it, through what chain, under what policy, and whether it was inside the envelope — the reconstruction that Section 3.3 observed is currently forensic archaeology.

The blast-radius question — after a mandate is revoked (or found compromised), find every transaction whose authority passed through it:

```
MATCH (rev:Event {kind: 'revocation'})-[:APPLIES_TO]->(m:Mandate {id:
↪  $mandate_id})
MATCH (child:Mandate)-[:DELEGATED_FROM*0..]->(m)
MATCH (t:Transaction)-[:EXECUTED_UNDER]->(child)
RETURN t.id, t.amount, t.counterparty, t.at,
       t.at > rev.at AS executed_after_revocation
```

Because delegation is a traversable structure, the query reaches not only transactions under the revoked mandate but transactions under every mandate *derived* from it — the exact set a flat log cannot produce without reconstructing the chains by hand. The final column isolates the most serious finding: anything executed after revocation should have failed transaction-time verification, and its presence indicates a propagation failure worth an incident of its own.

The stale-policy question — identify transactions executed under a policy version after it was superseded:

```
MATCH (newer:PolicyVersion)-[:SUPERSEDES]->(old:PolicyVersion)
MATCH (t:Transaction)-[:UNDER_POLICY]->(old)
WHERE t.at > newer.created_at
RETURN old.id AS stale_policy, newer.id AS superseding_policy,
       t.id, t.at
ORDER BY t.at
```

In a healthy deployment this returns nothing. A non-empty result is an early-warning signal that envelope updates are not propagating to enforcement — found by routine query rather than by post-incident review.

5.5 Scope and extension

The reference model is deliberately smaller than a production system. A deployment will extend it with protocol-specific transaction detail, counterparty verification records, invariant-check telemetry linked to transactions, richer event taxonomies,

and integration into existing GRC tooling. None of these extensions disturb the spine, and the spine is the point: six labels and eight relationships are enough to make the framework's central promise — trust that is examinable — operational. An institution evaluating an agentic-finance program, or a vendor's claims about one, can use the model as a concrete acceptance test: *show that these queries are answerable from your records.* Where they are not, the assurance gap of Section 3 is present, whatever the marketing asserts. Where settlement occurs on a public ledger, the transaction node carries the on-chain reference as evidence rather than duplicating it: the ledger is the stronger substrate for what happened, and the graph supplies what the ledger cannot — under whose authority, and whether that authority was current.

A closing note on enterprise semantics. The model is deliberately orthogonal to the industry's established semantic standards rather than an extension of them. FIBO, the Financial Industry Business Ontology, models the financial world — entities, instruments, contracts, obligations [47]; ISO 20022 standardizes the semantics of payment messaging [48]. Neither models delegated agentic authority, assurance cases, or provenance-grade audit, which is precisely why this model exists. Interoperation is nonetheless straightforward and, in production, should be made explicit: the model's `Principal` and counterparty concepts map onto FIBO's legal-entity and party constructs, and `Transaction` detail maps onto ISO 20022 payment semantics, so that the governance spine presented here attaches cleanly to an institution's existing semantic estate rather than competing with it.

5.6 Alignment with a standard provenance vocabulary

The reference model is not a new ontology. The W3C PROV data model supplies the general vocabulary this specializes: Entity, Activity and Agent, related by wasGeneratedBy, wasDerivedFrom and wasAttributedTo — and, for the case that matters here, actedOnBehalfOf, which models delegation as a first-class edge rather than an attribute [49]. The contribution here is not the idea of provenance but its specialization: which relations must be captured for a financial transaction to be answerable, and the requirement that capture be a precondition of execution rather than a reporting convenience. Aligning to a W3C Recommendation rather than inventing a vocabulary also answers a practical objection — an institution adopting this is extending a standard model, not carrying another one.

The delegation edge is what makes revocation answerable. Given any revocation event, the query reaches every mandate derived from the revoked one, the agent that held each, the transactions executed after the revocation, and the principal at the origin of the chain — the actedOnBehalfOf path, traversed:

```
MATCH (rev:Event {kind: 'revocation'})-[:APPLIES_TO]->(revoked:Mandate)
MATCH (descendant:Mandate)-[:DELEGATED_FROM*0..]->(revoked)
MATCH (descendant)-[:HELD_BY]->(a:Agent)
MATCH (t:Transaction)-[:EXECUTED_UNDER]->(descendant)
WHERE t.at > rev.at
MATCH chain = (descendant)-[:DELEGATED_FROM*0..]->(root:Mandate)
              <-[:GRANTED]-(p:Principal)
RETURN p.id             AS principal,
       a.id             AS acting_agent,
       revoked.id       AS revoked_mandate,
       t.id             AS transaction,
       length(chain)    AS hops_from_principal
ORDER BY hops_from_principal
```

A flat log cannot express this. It records each transaction and each revocation as separate entries and has no edge to follow between them; the relation that makes the second bear on the first exists only in the reader's head, and must be reconstructed by hand for every question asked.

6. Worked Example: Assurance in a Production Autonomous Trading System

The framework of Sections 4 and 5 is not a proposal awaiting its first implementation. This section describes its disciplines as practiced in a quantitative trading system built and operated by the author — a system constructed deliberately to institutional grade, with autonomous components authorized to act in live electronic markets under enforced constraint. The account is anonymized: no venue, instrument class, strategy, or performance detail is given, because none is needed. The subject is the governance architecture, and the lessons are in how the disciplines behave under the ordinary pressures of operating a real system.

6.1 Authority and separation of duties

The system operates under a single human principal, and the first discipline is that this fact is represented explicitly rather than assumed. Authority is separated into three roles: a decision authority that owns strategy, risk appetite, and the granting of mandates; an operational role that runs the system day to day within those mandates; and implementation agents — including AI coding and analysis agents — that execute defined tasks under explicitly scoped, revocable delegation. No role is permitted to widen its own authority; in particular, implementation agents receive narrow task-level mandates and cannot alter the constraints they operate under. The arrangement is Section 4.3 in miniature: even in a small organization, the chain from principal to acting software is written down, and the narrowing rule is enforced by construction — the party bound by a constraint is never the party that can modify it.

Irreversibility is treated as a mandate boundary. Actions that cannot be undone — deployments to live execution, changes to enforcement configuration, movements of funds — sit outside every standing delegation and require explicit, per-instance human authorization. Reversible work proceeds autonomously; irreversible work does not. The boundary described in Section 4.5, applied consistently at the scale of one operator.

6.2 The envelope

Policy exists as enforced constraint, not documentation. Exposure caps, scope restrictions on what the system may touch, rate bounds, and halt conditions are implemented in an enforcement layer distinct from the components it constrains, and the constrained components have no write access to it. The independence is the point, and it was designed in from the start on the reasoning of Section 4.6: an agent that can reconfigure its own limits has preferences, not policies.

Halt is a first-class state. The system is designed to stop — on invariant violation, on data it cannot validate, on any condition its assurance case did not anticipate — and to remain stopped until a human examines the record and explicitly re-authorizes operation. In several years of operation the halts that occurred were, without exception, cheaper than the incidents they preempted. The cultural adjustment mattered more than the engineering: learning to read a halted system as the discipline succeeding, rather than as an embarrassment to be resumed quickly, took deliberate effort — and the author's experience suggests institutions should expect the same.

6.3 The assurance case in practice

The system's equivalent of Section 4.5 is a gated research-to-deployment pipeline. No candidate strategy reaches capital exposure without passing an ordered sequence of falsification gates: statistical evidence of a genuine effect, diagnostic checks that the effect is not an artifact, explicit testing for overfitting and selection bias, and demonstration that the effect survives realistic costs and decays gracefully rather than vanishing out of sample. Each gate produces a recorded verdict against pre-stated criteria, and the accumulated verdicts constitute the assurance case for anything that deploys.

The most instructive property of this discipline is its output distribution: the record contains far more rejections than approvals. Candidates that survived early gates have been terminated at later ones, with the negative verdicts recorded in the same detail as any success, including the honest finding that an apparently genuine effect was uneconomic to exploit. This is what an assurance discipline looks like from the inside — most of its value is in what it prevents from deploying — and it is the property the

author would most urge institutions to examine when a vendor claims to practice one. Ask to see the rejections. A record containing only approvals is a record of a discipline that is not operating.

6.4 Provenance as the system of record

Every decision, invariant, gate verdict, calibration, and authorization in the system is recorded in a single authoritative knowledge model, maintained under the same source-of-truth principle the author's engineering lineage applies to safety-critical programs: the model is the record, and all working documentation is generated from it rather than authored beside it. The practical consequence is Section 4.7's reconstruction property at small scale. Questions of the form *why is this constraint set to this value, under what evidence was this component approved, what was known when this decision was made* are answered by query, years later, without dependence on the operator's memory.

The unglamorous lesson is that this record repeatedly outperformed recollection — the author's own included. Decisions reconstructed from the knowledge model differed, in unimportant and occasionally important ways, from how they were remembered. An assurance regime that depends on what its operators remember is a regime that degrades with staff turnover and time; one that depends on its records does not, but only if the records are captured as a condition of acting, in the manner Section 5 makes structural.

6.5 What the example does and does not show

A single-principal system operated by its builder is the easy case for governance — no committee, no vendor boundary, no contested authority. That is precisely why it is presented: the disciplines described here were adopted under no regulatory compulsion, at a scale where every shortcut was available, because they paid for themselves in prevented incidents and answerable questions. At institutional scale, with contested authority and real supervisory exposure, the case for them is strictly stronger, and the machinery of Sections 4 and 5 is what they look like grown up. What the example does not show is any claim about trading outcomes; the framework governs conduct, not returns, and the disciplines described would be worth their cost in a system that never earned a basis point.

7. An Adoption Path for Institutions

7.1 A maturity ladder

The framework of Sections 4 and 5 describes an end state. No institution reaches it in one step, and the attempt to do so — a comprehensive agentic-governance program launched before the institution has any agentic exposure to govern — is how frameworks become shelfware. The practical path is a ladder, each rung producing artifacts the next rung requires, and each rung defensible to a board or examiner as a complete position in its own right.

Rung one: visibility. The institution establishes, as a matter of fact rather than assumption, where agent-initiated flows already touch it — through network relationships, processor integrations, vendor AI features, and treasury or procurement tooling with autonomous capabilities. Section 2.3's observation applies: exposure arrives through the existing stack uninvited, and most institutions at this rung discover flows they did not know they carried. The deliverable is an inventory, and the honest version includes "unknown" entries.

Rung two: vocabulary and policy. Before any technology is deployed, the institution adopts the governance vocabulary — principal, mandate, envelope, invariant, provenance — and writes policy in those terms: which classes of agentic activity are permitted at all, what consequence tiers apply, and what any agent-facing system must be able to demonstrate. The assurance-level construct of Section 4.5 does its first work here, as a classification scheme for exposure the institution already has.

Rung three: demands on vendors. Most institutions will encounter consequential agents first as buyers, not builders. This rung converts the framework into procurement and third-party-risk requirements: vendors of agentic capability are asked for their assurance case, their envelope enforcement architecture, their halt semantics, and — the acceptance test of Section 5.5 — a demonstration that the reconstruction queries are answerable from their records. Ask to see the rejections; a vendor that can produce none of this has answered the diligence question.

One barrier deserves stating plainly rather than absorbing into a scope boundary. The framework assumes a deployer that can state what its agent may do and how it behaves. Many institutions cannot, because the reasoning layer belongs to a vendor and the contract does not entitle them to look inside it. The honest form of the objection is not that the framework is wrong but that its central artifact requires knowledge the institution is contractually unable to obtain.

Two things follow. The first is that this is what rung three exists for: an assurance case a deployer cannot construct alone is a procurement requirement before it is an engineering one, and the vendor demands described above are the mechanism by which the missing evidence becomes obtainable. The second is that where a vendor will not supply it, that is itself an assurance finding. A documented inability to evidence a con-

trol is a materially different position in front of a supervisor than an undocumented assumption that the control exists. The framework does not make an opaque vendor transparent. It makes the opacity visible, attributable, and priced.

Rung four: contained deployment. The institution's first own agents operate under the full discipline at deliberately low consequence: narrow mandates, reversible transactions, conservative envelopes, and an enforcement layer and independent watchdog stood up before the first funded transaction — because retrofitting enforcement under an agent already in production is the most expensive sequencing error available. The purpose of this rung is as much institutional as technical: it is where risk, audit, and operations acquire the operating habits that rung five assumes.

Rung five: assured autonomy at consequence. Mandates widen and consequence rises only as fast as the assurance case, the operating record, and the provenance infrastructure support. At this rung the institution can answer, from records, every question in Section 4.7's test — for any transaction, at any time, to any examiner — and its agentic activity is governed by the same disciplines, proportionally applied, as any other consequential automated system it operates.

The case for funding this work before it is required is easier to make than it once was. An oversight-duty claim now has a trial record behind it, and the question such a claim asks — was there a system capable of surfacing this, and did anyone attend to it — is answerable in advance or not at all. An assurance case assembled while the controls are being built is evidence. The same material assembled afterwards, under discovery, is reconstruction, and it is read as such.

This is the unglamorous version of the argument, and it is worth stating plainly because the glamorous version — that good governance protects the firm's reputation — has a poor record of releasing budget. What releases budget is the observation that the artifact is cheap to produce contemporaneously and expensive to produce later, and that the later production happens under conditions nobody chooses.

There is a convergence here worth noting once. The IMF's mitigation recommendations include board-level oversight of AI deployment and internal model risk management adapted for agentic systems [1]. Delaware's duty of oversight, meanwhile, is defined not by outcomes but by whether an information-system control capable of surfacing the problem was implemented and attended to [26]. The first says a board should be watching. The second says that failing to build what makes watching possible is itself the breach. An assurance case assembled during development satisfies both, and it is the only artifact discussed in this paper that does.

7.2 Fit with the existing governance estate

Nothing above creates a parallel governance universe. The framework docks into programs the institution already runs. Model risk management gains a new object class: the assurance case is the validation artifact for systems that act, complementing — not replacing — validation of models that inform. Third-party risk management gains rung three's requirements as an extension of existing vendor diligence. Internal audit gains the reconstruction queries as testable controls. Operational risk and resilience programs gain halt semantics as a designed capability rather than an incident outcome. The framework's contribution is not new committees; it is giving the existing ones artifacts rigorous enough to examine.

7.3 Where the framework docks

The most common objection an unaffiliated framework meets is not that it is wrong but that it is another one. A risk function already carrying supervisory guidance, an internal model risk policy, an ISO-aligned management system and a growing set of AI-specific expectations has a reasonable first question: does this map onto what I already hold, or is it a thirteenth vocabulary?

The answer is that this framework is not a competing statement of what to achieve. The official sector has stated that, and stated it well. What the sound practices, the supervisory guidance, and the emerging statutory regimes have in common is that they specify obligations without naming the artifact that evidences them. That is the layer this framework occupies, and the table below is the clearest available statement of it: for each of the twelve sound practices proposed by the Financial Stability Board [22], the framework element that produces examinable evidence, and the artifact itself.

Sound practice	Framework element	Artifact produced
1. Strategic direction and oversight	Consequence-graded assurance (§4.5)	Assurance level assigned per agent, tied to stated risk appetite
2. Governance and accountability	Verifiable delegation (§4.3)	Delegation chain terminating at a named principal; roles as mandate grants, not org-chart prose
3. Incorporation of AI risks into risk management framework	Pre-deployment assurance case (§4.5)	The assurance case as the validation artifact for software that acts
4. Organizational adaptability	The framework as lifecycle (§4.8)	Re-verification triggers on mandate widening or envelope change
5. Materiality and risk assessment	Consequence grading (§4.5)	Consequence tier recorded per use case, driving assurance rigor
6. Selection	—	The framework governs conduct after selection, not the selection itself
7. Data governance	Provenance-grade audit (§4.7)	Extends the report's own data-lineage requirement from inputs to authority
8. Explainability and transparency	Policy as enforceable constraint (§4.4)	An enforced envelope is examinable whether or not the reasoning is; the constraint is the compensating control
9. Performance management	Runtime invariants (§4.6)	Continuously evaluated invariants with pre-specified thresholds and recorded verdicts
10. Human oversight	Human gates and mandated degradation (§4.5, §4.6)	Pre-execution approval gates by consequence tier, and the degradation ladder: which violation escalates to which rung, decided before deployment
11. Cyber and ICT risk management	—	Addressed by existing security disciplines; the framework assumes rather than replaces them
12. Third-party AI risk management	The adoption ladder, rung three (§7.1)	Vendor assurance case, envelope architecture, halt semantics, and the acceptance test of §5.5

Two entries are deliberately empty. A crosswalk that claimed complete coverage would be advertising rather than mapping. The framework has nothing to add to model selection, which is a decision about fitness for purpose taken before any of this applies, and nothing to add to cyber and ICT risk management, which is a mature discipline this work depends on rather than extends.

The mapping holds against older anchors as well. Supervisory guidance on model risk management [25] requires conceptual soundness, ongoing monitoring and outcomes analysis, and independent validation proportionate to consequence; the assurance

case is the artifact that discharges the last of these for software that transacts rather than advises, and the sound practices' own insistence that audit, evaluation and testing remain functionally separate from development is the same independence requirement this framework applies to the watchdog in Section 4.6.

The statutory anchors map the same way. The EU AI Act's obligations on risk management systems, record-keeping, human oversight, and deployers of high-risk systems each specify a duty without naming the artifact that discharges it for software that transacts [20].

7.4 The cost of the ladder, honestly

The disciplines described are not free. Mandate and envelope machinery must be built or bought; assurance cases take senior engineering time; independent verification is a real budget line; provenance capture imposes requirements on every system in the transaction path. Two observations keep the cost in proportion. First, proportionality is load-bearing: the ladder prices assurance to consequence, and the institution controls consequence through mandate scope — nothing forces a maximal program on a minimal exposure. Second, the alternative is not zero cost but deferred cost: the institution that reaches an incident, an examination, or a litigation discovery without provenance-grade records pays for the gap at the worst possible price, at the worst possible time, in front of the least sympathetic audience. The ladder is cheaper than the archaeology.

8. Conclusion

The history of financial infrastructure is a history of trust being made examinable. Double-entry bookkeeping, the audit profession, clearinghouses, deposit insurance, model risk management — each arrived when a new capability outran the existing means of answering *why should this be trusted, and who answers when it fails*. Autonomous transacting agents are the current instance of that recurring moment. The capability is deployed; the examination discipline is not; and the interval between the two is where losses, litigation, and supervisory findings accumulate.

This paper's claim has been that the interval can be closed with methodology that already exists. The assurance disciplines of safety-critical engineering — translated here as verifiable delegation, policy as enforceable constraint, graded assurance cases, mandated halt semantics, and provenance captured at the moment of action — were developed for exactly this shape of problem, and they have the property regulated finance requires: they produce artifacts an examiner can interrogate rather than assurances an institution must take on faith. That is the layer this work occupies. Supervisory guidance, sound practices, and emerging statutory regimes

state with increasing precision what an institution must achieve; what none of them names is the artifact that evidences it. The graph-native reference model gives the provenance core a concrete, minimal, protocol-agnostic form, and the worked example demonstrates that the disciplines are operable in practice, at small scale, under no compulsion except that they pay for themselves.

None of this slows the technology down, and none of it needs to. The institutions that move first on assurance will not be the ones that adopt agentic finance last; they will be the ones able to adopt it at consequence, because they can answer for it. That has been the pattern in every safety-critical domain: the discipline is not the brake, it is the license.

About the Author

Robert Jeffs is a systems engineer with more than thirty years of experience in systems engineering and business automation development, principally in safety-critical aerospace and defense programs, and five years of applied experience in AI development and the cryptocurrency industry. His work spans model-based systems engineering, enterprise architecture, and graph-based knowledge representation, with a current focus on translating safety-critical assurance practice into governance frameworks for autonomous financial systems. He operates a production quantitative trading system governed by the disciplines described in this paper.

He holds an MBA from Heriot-Watt University and an MSc in Major Programme Management from the University of Oxford.

He advises institutions on assurance and governance for AI systems in regulated finance. Contact: robert.j.jeffs@qbrs-labs.com · www.qbrs-labs.com

A Note on Method

This paper was produced through an agentic AI development process: drafting, research, and revision were performed by AI systems operating under the author's direction, with section-gated review, independent adversarial peer review, and every claim subject to the author's verification. The process practiced the disciplines the paper describes — scoped delegation, gated assurance, and a single accountable principal. Responsibility for the content, and for its errors, rests entirely with the author.

The paper incorporates IMF Note 2026/004, which was published after the underlying research was completed; a corrected reading of the x402 transaction data; material on the corporate oversight duty and its first trial record; and a section on adjacent work. The references supporting this material were verified against primary sources.

References

1. Davidovic, Sonja, and Hervé Tourpe, *How Agentic AI Will Reshape Payments*, IMF Note 2026/004, International Monetary Fund, Washington, DC, 2026.
2. Coinbase / Cloudflare, x402 protocol specification and launch materials, May 2025. https://x402.org
3. Linux Foundation, x402 Foundation formalization announcement, April 2, 2026 (foundation first announced by Coinbase and Cloudflare, September 23, 2025).
4. Google Cloud, *Announcing Agent Payments Protocol (AP2)*, September 16, 2025. https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol
5. Google, *Google donates Agent Payments Protocol to FIDO Alliance*, April 28, 2026; AP2 v0.2 release adding "Human Not Present" payments. https://blog.google/products-and-platforms/platforms/google-pay/agent-payments-protocol-fido-alliance/
6. Mastercard, *Mastercard unveils Agent Pay*, April 29, 2025; *Mastercard launches Agent Pay for Machines*, June 10, 2026.
7. Visa, *Visa unveils new AI capabilities* (Intelligent Commerce), April 30, 2025; Trusted Agent Protocol, October 14, 2025; Intelligent Commerce Connect, April 2026.
8. FIDO Alliance, *Building the Trust Layer for Agentic Payments with AP2 and Verifiable Intent*, May 26, 2026, https://fidoalliance.org/building-the-trust-layer-for-agentic-payments-with-ap2-and-verifiable-intent/; Verifiable Intent co-developed with Google and contributed to the FIDO Alliance alongside AP2, as announced in ref. 5.
9. Google, *New tech and tools for retailers to succeed in an agentic shopping era* (Universal Commerce Protocol launch), January 11, 2026. https://blog.google/products/ads-commerce/agentic-commerce-ai-tools-protocol-retailers-platforms/
10. Amazon, *How Amazon is using generative and agentic AI to transform the shopping experience* ("Buy for Me" delegated purchasing completed on external merchant sites). https://www.aboutamazon.com/news/retail/amazon-agentic-ai-gen-ai-shopping
11. PayPal, *PayPal to Acquire Cymbio, Accelerating Agentic Commerce Capabilities*, January 22, 2026. https://newsroom.paypal-corp.com/2026-01-22-PayPal-to-Acquire-Cymbio,-Accelerating-Agentic-Commerce-Capabilities
12. See ERC-1812 (off-chain verifiable claims), ERC-6900 (modular smart accounts), and ERC-8004 (agent identity and reputation registries), surveyed in ref. 1; and ERC-4337 (account abstraction), which ref. 1 situates at the settlement layer.

13. Agent Economy, x402 on-chain settlement tracker (cumulative transactions and stablecoin volume across chains), accessed July 2026. https://agenteconomy.to
14. CoinDesk, *Coinbase-backed AI payments protocol wants to fix micropayments but demand is just not there yet* (citing Artemis on-chain analysis), March 11, 2026.
15. Artemis Analytics, wash-trading analysis of x402 activity, flagging wallets that repeatedly transact with themselves or cycle funds between linked addresses: approximately 48 percent of transaction count and 81 percent of transaction volume classified as non-organic as of December 2025. Reported in *Analysts temper hype on x402, agentic AI commerce growth*, Cryptopolitan, 2026. https://www.cryptopolitan.com/x402-agentic-ai-commerce-growth/
16. Chainalysis, *Inside x402: 100M Agentic Payments on Base*, June 2026. https://www.chainalysis.com/blog/x402-agentic-payments-adoption/
17. Ling, Shengchen, Yihang Huang, Yuefeng Du, Yuan Chen, Yajin Zhou, Lei Wu, and Cong Wang, *Free-Riding the Agentic Web: A Systematic Security Analysis of x402 Payments*, arXiv:2605.30998, 2026. https://arxiv.org/abs/2605.30998
18. Infocomm Media Development Authority, *Model AI Governance Framework for Agentic AI*, Singapore, launched 22 January 2026. https://www.imda.gov.sg/resources/press-releases-factsheets-and-speeches/press-releases/2026/new-model-ai-governance-framework-for-agentic-ai
19. Financial Stability Board, *Monitoring Adoption of Artificial Intelligence and Related Vulnerabilities in the Financial Sector*, October 2025. On the EU AI Act, see ref. 20.
20. Regulation (EU) 2024/1689 (Artificial Intelligence Act), as amended by Regulation (EU) 2026/1744 (Digital Omnibus on AI), in force 27 July 2026, deferring Annex III high-risk obligations to 2 December 2027 and Annex I embedded high-risk obligations to 2 August 2028.
21. International Organization of Securities Commissions, *Supervisory Toolkit for AI Use in Capital Markets*, Final Report FR/02/2026, May 2026.
22. Financial Stability Board, *Sound Practices for Responsible Adoption of Artificial Intelligence (AI): Consultation Report*, 10 June 2026.
23. National Institute of Standards and Technology, Center for AI Standards and Innovation, *Announcing the "AI Agent Standards Initiative" for Interoperable and Secure Innovation*, 17 February 2026.
24. Protocol landscape: OpenAI / Stripe Agentic Commerce Protocol, specification maintained by OpenAI and Stripe under Apache 2.0, https://github.com/agentic-commerce-protocol/agentic-commerce-protocol; Google Universal Commerce Protocol (January 2026); Stripe Shared Payment Tokens (March 2026).
25. Board of Governors of the Federal Reserve System / OCC, *Supervisory Guidance on Model Risk Management*, SR 11-7 / OCC 2011-12, April 2011.
26. *In re Caremark International Inc. Derivative Litigation*, 698 A.2d 959 (Del. Ch. 1996).

27. *In re Facebook Inc. Derivative Litigation*, Consolidated C.A. No. 2018-0307-KSJM (Del. Ch.), trial commenced July 16, 2025.
28. Settlement of $190 million approved by the Delaware Court of Chancery, April 7, 2026, reported in *Meta's $190 Million Settlement Gets Approval*, Bloomberg Law, April 8, 2026, https://news.bloomberglaw.com/delaware-brief/metas-190-million-settlement-gets-approval-delaware-brief; funded by directors' and officers' liability insurance, with no admission of liability. Accompanying governance undertakings included an expanded whistleblower program covering privacy-law violations and a new independent director code of conduct.
29. Federal Reserve, FDIC, and OCC, *Interagency Guidance on Third-Party Relationships: Risk Management*, June 2023.
30. Bank of England Prudential Regulation Authority, *SS1/23: Model risk management principles for banks*, 2023.
31. Regulation (EU) 2022/2554 on digital operational resilience for the financial sector (DORA), applicable January 2025.
32. Monetary Authority of Singapore, *Principles to Promote Fairness, Ethics, Accountability and Transparency (FEAT) in the Use of AI and Data Analytics*, 2018.
33. Chiris, Lorenzo Satta, and Ayush Mishra, *AURA: An Agent Autonomy Risk Assessment Framework*, arXiv:2510.15739, 2025. https://arxiv.org/abs/2510.15739
34. Acharya, Vivek, *Secure Autonomous Agent Payments: Verifying Authenticity and Intent in a Trustless Environment*, arXiv:2511.15712, 2025. https://arxiv.org/abs/2511.15712
35. Hu, Botao "Amber," and Helena Rong, *Inter-Agent Trust Models: A Comparative Study of Brief, Claim, Proof, Stake, Reputation and Constraint in Agentic Web Protocol Design*, arXiv:2511.03434, 2025. https://arxiv.org/abs/2511.03434
36. Stantchev, Vladimir, *Hardening x402: PII-Safe Agentic Payments via Pre-Execution Metadata Filtering*, arXiv:2604.11430v2, 2026. https://arxiv.org/abs/2604.11430v2
37. RTCA, *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*, 2011.
38. SAE International, *ARP4761A: Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment*, 2023.
39. Assurance Case Working Group, *Goal Structuring Notation Community Standard*, Version 3, 2021.
40. US Securities and Exchange Commission, *In the Matter of Knight Capital Americas LLC*, Administrative Proceeding File No. 3-15570, Release No. 34-70694, 16 October 2013.
41. Board of Banking Supervision, *Report of the Inquiry into the Circumstances of the Collapse of Barings*, Bank of England / HM Treasury, 18 July 1995.

42. NIST, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, January 2023.
43. C. Ellison et al., *SPKI Certificate Theory*, IETF RFC 2693, 1999; UCAN Working Group, *User-Controlled Authorization Networks specification* — the capability-attenuation lineage of the narrowing rule.
44. Amazon Web Services, *Cedar: an open-source policy language and evaluation engine with decidable, verifiable authorization semantics*, 2023.
45. ANSI/UL 4600, *Standard for Safety for the Evaluation of Autonomous Products*, 3rd edition, 2023.
46. OWASP Foundation, *OWASP Top 10 for Large Language Model Applications* (LLM01: Prompt Injection), 2025 edition.
47. EDM Council / Object Management Group, *Financial Industry Business Ontology (FIBO)*.
48. ISO 20022, *Universal financial industry message scheme*.
49. W3C, *PROV-O: The PROV Ontology*, W3C Recommendation, 30 April 2013.