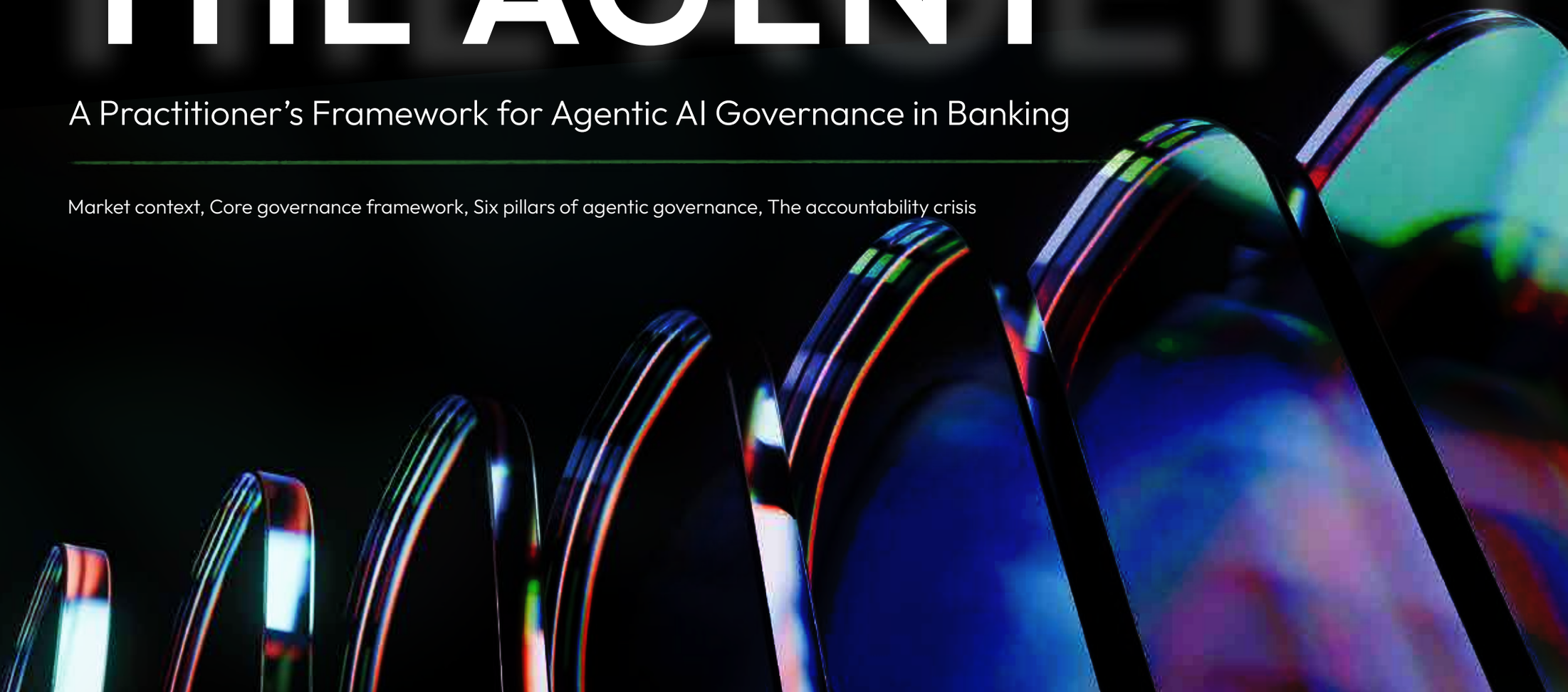
brillio

Brillio AI Governance Series | Banking Edition | 2026

# GOVERNING THE AGENT

## A Practitioner's Framework for Agentic AI Governance in Banking

Market context, Core governance framework, Six pillars of agentic governance, The accountability crisis



# EXECUTIVE SUMMARY

Banking is undergoing the most consequential technology shift in a generation. Institutions that are compressing credit decisions from days to seconds, reducing AML investigation backlogs by 70%, and delivering hyper-personalised customer experiences at scale are doing so through networks of autonomous AI agents. The governance frameworks built for the previous AI paradigm are not merely inadequate for this new one - they are structurally incompatible with it.

This white paper is organised as a practitioner's journey through the governance challenge. It begins with what is happening in the market, establishes the foundational governance framework that every banking AI deployment must be built on, and then examines the six governance pillars that agentic AI demands, as well as the accountability crisis that will define which institutions and partners lead in this era.

1

**Part I -**
**The Market:**

What is actually happening in banking AI, why the governance gap is structural not incremental, and why governance has become a hard buying criterion.

2

**Part II -**
**The Core Framework:**

A battle-tested four-layer AI governance architecture - built from practitioner experience - that defines the governance baseline, and how agentic AI stress-tests each of its four layers.

3

**Part III -**
**The Six Pillars:**

A deep technical treatment of the six governance pillars that agentic banking AI requires, including the technical architecture, control tower design, agent libraries, safety scoring, and key open-source tooling.

4

**Part IV -**
**Accountability Paralysis:**

The diagnosis of why accountability breaks down in agentic systems, its three structural mechanisms, and the architectural solution that eliminates it before it occurs.

# PART I:

# THE MARKET

## THE AGENTIC INFLECTION POINT

Between 2020 and 2023, enterprise AI deployments in banking were characterised by narrow, supervised, explainable systems. Fraud detection models flagged transactions for human review. Chatbots routed queries. Document processing pipelines extracted structured data. Risk was bounded and governance, while imperfect, was tractable.

By 2024, the architecture had transformed. Large language model-based agents with tool-calling capabilities, memory, and orchestration frameworks began replacing brittle rule-based systems. By 2025, multi-agent systems - networks of specialised AI agents collaborating autonomously on complex end-to-end tasks - had moved from proof-of-concept to production at leading financial institutions globally. A credit application that once required four human touchpoints and two days now completes in under four minutes with no human involvement for 83% of cases at certain digital lenders.

### The scale of the shift:

Gartner projects that by 2028, 33% of enterprise software applications will include agentic AI capabilities, up from less than 1% in 2024. IDC estimates financial services firms will account for the single largest share of enterprise AI spend globally, exceeding $100 billion annually by 2027. The Bank for International Settlements has flagged multi-agent AI as one of the top five systemic risk factors in global financial markets.

This acceleration has outrun governance by a significant margin. Most enterprise AI risk policies still reference model validation, human-in-the-loop requirements, and output monitoring - artefacts of a deterministic, single-model world. Agentic systems are non-deterministic, multi-model, and often self-directing. The governance gap is not a matter of degree; it is structural.

# FIVE MARKET SIGNALS THAT GOVERNANCE HAS BECOME A BUYING CRITERION

Governance in banking AI has moved from being a compliance function to becoming a commercial differentiator. The following five signals, each observable in competitive financial services mandates today, explain why:

01

### RFP-stage governance gates

Technology providers that cannot specify their AI governance architecture - including documented agent accountability, audit trail capability, and regulatory mapping - are being disqualified at the RFP stage in competitive financial services mandates. Governance is no longer a due-diligence question asked after vendor selection. It is a threshold criterion applied before shortlisting.

02

### Board-level risk register inclusion

AI risk - specifically the risk of autonomous agents making consequential errors in credit, AML, and customer-facing decisions - has moved onto the board-level risk registers of 74% of the top 50 global banks (Oliver Wyman, 2024). This elevates AI governance from a technology management responsibility to a fiduciary one.

03

### Regulatory examination of AI systems

Banking regulators - including the OCC, FCA, RBI, and MAS - are now including AI system examination in routine supervisory reviews. Banks that cannot produce structured governance evidence within a regulatory examination window face findings. Implementation partners whose architecture cannot support that evidence production are becoming a regulatory liability.

04

### The partner accountability covenant

When an AI-enabled engagement fails in production, the enterprise client does not cite an LLM hallucination in their board review. They cite their implementation partner. The reputational consequence accrues to the partner regardless of where in the technical stack the failure originated - because the client contracted with the partner. Governance is the partner's existential commercial imperative.

05

### ISO/IEC 42001 as procurement standard

ISO/IEC 42001 - the AI Management System certification standard published in December 2023 - is increasingly being written into banking procurement requirements in Europe and Asia-Pacific. In the same way that ISO 27001 became a non-negotiable baseline for information security vendors, ISO 42001 is tracking toward the same status for AI governance.

# WHY CURRENT GOVERNANCE FRAMEWORKS FALL SHORT

The governance crisis in banking AI is not caused by a lack of frameworks. SR 11-7 has governed model risk for over a decade. Banks have well-developed model validation practices, risk appetite statements, and three-lines-of-defence structures. The problem is not the absence of governance; it is the mismatch between frameworks designed for single-purpose statistical models and the operational reality of multi-agent autonomous systems.

| Governance Dimension | What frameworks assumed | What agentic AI requires |
|---|---|---|
| **Decision maker** | A model with defined inputs and outputs | A network of agents with dynamic reasoning, tool calls, and emergent inter-agent behaviour |
| **Accountability** | Model owner + risk team + business process owner | Ambiguous across model developer, orchestration designer, tool provider, deployer - accountability paralysis |
| **Audit trail** | Model inputs and outputs logged | Multi-hop reasoning chains, tool calls, inter-agent messages, memory accesses - full causal trace required |
| **Validation** | Statistical performance testing on representative data | Adversarial red-teaming, multi-step reasoning analysis, inter-agent interaction testing |
| **Monitoring** | Statistical output drift detection | Behavioural drift monitoring, tool call pattern analysis, escalation rate tracking |
| **Failure mode** | Predictable, bounded, attributable to a specific model | Emergent, cascading across agents, potentially attributable to orchestration logic rather than any individual model |

# PART II:
# THE CORE FRAMEWORK

The framework below represents the foundational governance architecture for enterprise AI: a four-layer structure that defines the minimum governance baseline for any AI deployment in a regulated financial institution. This framework predates the agentic era, which is precisely why it is instructive. Understanding where it holds and where it fractures under agentic pressure is the starting point for the governance upgrade that Part III delivers.

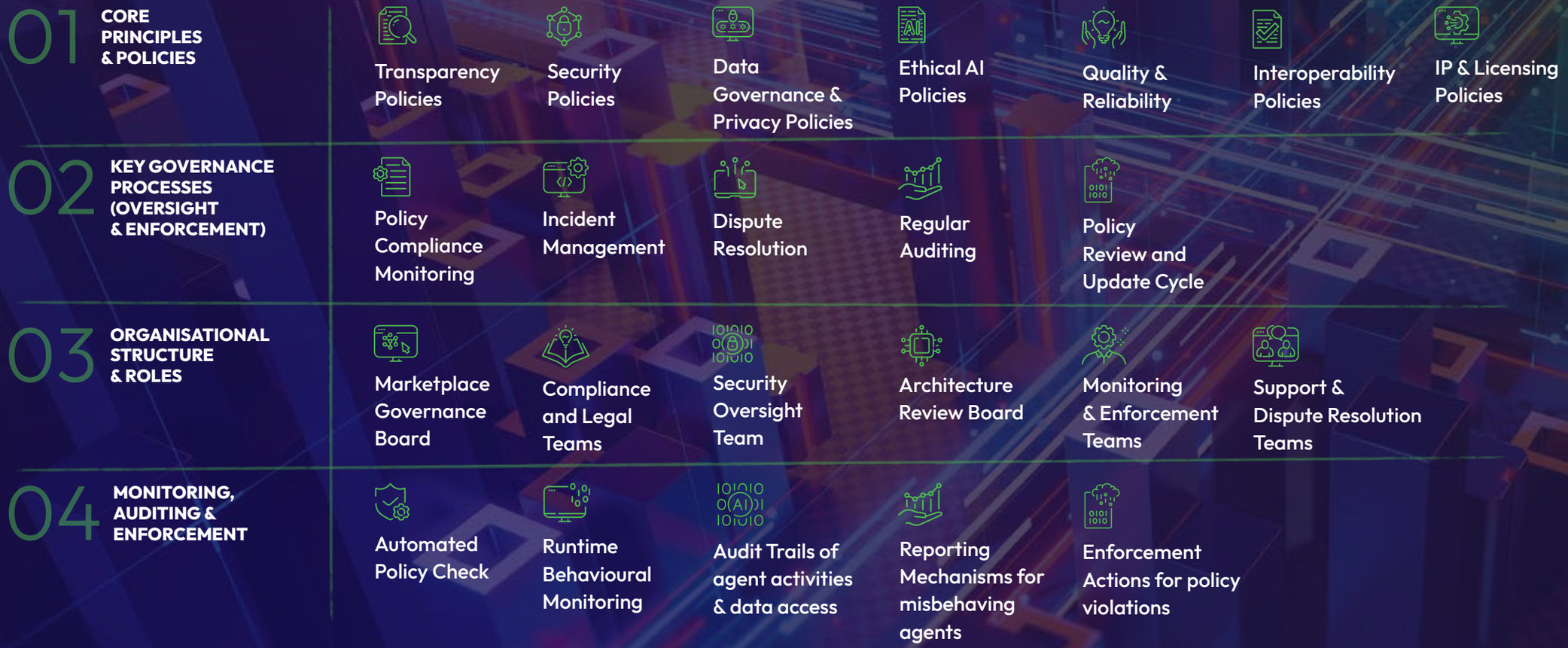



**Figure 1:** Foundational AI Governance Framework - Four-Layer Architecture

## LAYER 1:

## CORE PRINCIPLES & POLICIES

The policy layer defines the non-negotiable commitments governing every AI system: transparency (what the AI does and how it decides must be documentable), security (the system must not be exploitable), data governance and privacy (personal data must be handled in compliance with applicable law), ethical AI (the system must not discriminate or cause disproportionate harm), quality and reliability, interoperability, and IP and licensing compliance.

**Agentic AI impact:** The policy layer assumes that the scope of an AI system's actions is defined at design time. In agentic systems, the action space is dynamic because agents can use new tools and combine them in new ways. Policies must therefore address dynamic reasoning, tool use, and emergent patterns, not only fixed inputs and outputs.

## LAYER 2:

## KEY GOVERNANCE PROCESSES (OVERSIGHT & ENFORCEMENT)

The process layer defines the operational rhythms that keep governance alive: policy compliance monitoring, incident management, dispute resolution, regular auditing, and policy review cycles. These processes represent mature governance for model-based AI deployed on predictable timescales.

**Agentic AI impact:** Process-layer governance was designed for human-paced review cycles. Agentic systems can make hundreds or thousands of decisions per day, so auditing, escalation, and incident management must operate continuously and in real time.

## LAYER 3: ORGANISATIONAL STRUCTURE & ROLES

The process layer defines the operational rhythms that keep governance alive: policy compliance monitoring, incident management, dispute resolution, regular auditing, and policy review cycles. These processes represent mature governance for model-based AI deployed on predictable timescales.

**Agentic AI impact:** Process-layer governance was designed for human-paced review cycles. Agentic systems can make hundreds or thousands of decisions per day, so auditing, escalation, and incident management must operate continuously and in real time.

## LAYER 4: MONITORING, AUDITING & ENFORCEMENT

The monitoring layer defines the operational surveillance infrastructure: automated policy checks, runtime behavioural monitoring, audit trails of agent activities and data access, reporting mechanisms for misbehaving agents, and enforcement actions for policy violations.

**Agentic AI impact:** Output monitoring alone is not enough for agentic systems. Governance must capture inter-agent messages, tool calls, memory access, reasoning steps, and the final decision so that actions can be reviewed and paused when needed.

**Governance upgrade thesis:** The four-layer framework remains necessary, but it is no longer sufficient on its own. The six pillars in Part III extend the framework for autonomous, multi-agent banking AI.

# PART III:

# THE SIX PILLARS OF AGENTIC AI GOVERNANCE IN BANKING

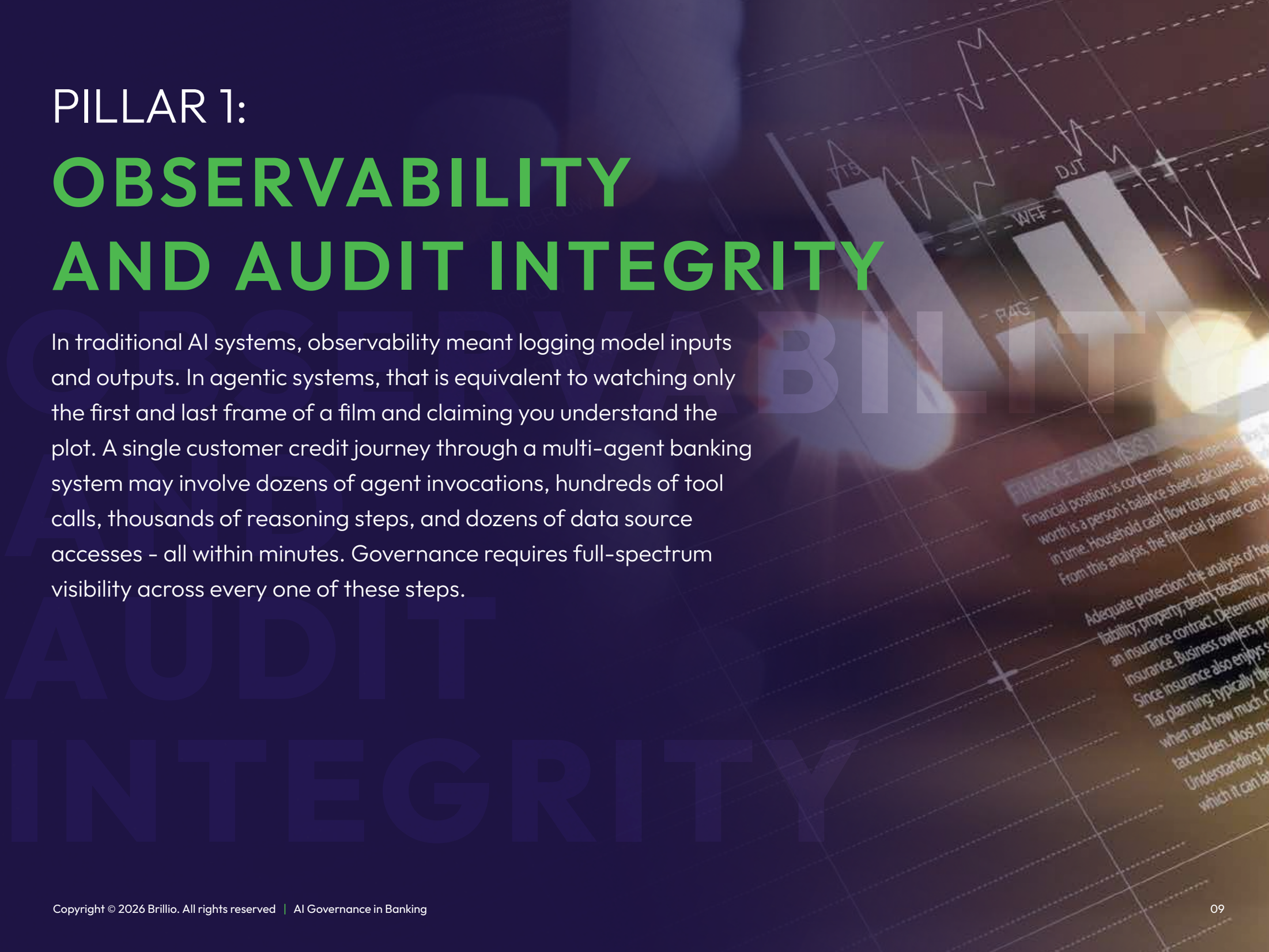
The six pillars below represent the governance capabilities that banking institutions and their implementation partners must build to govern agentic AI at production scale. Each pillar maps to one or more layers of the foundation framework and extends it in ways its original design did not anticipate.

| Pillar | Extends foundation layer | Core governance question answered |
|---|---|---|
| **1. Observability & Audit Integrity** | Layer 4: Monitoring, auditing, enforcement | Can we reconstruct, in full causal detail, why the agent made this decision? |
| **2. Policy-Driven Guardrails & Circuit Breakers** | Layer 1: Core principles;<br>Layer 2: Compliance monitoring | Can we guarantee the agent cannot take certain actions regardless of what it is told to do? |
| **3. Agent Safety Scoring & Risk Classification** | Layer 2: Regular auditing;<br>Layer 4: Enforcement | Do we know how dangerous each agent is, and are we governing it accordingly? |
| **4. Regulatory Compliance Mapping** | Layer 1: Policy layer;<br>Layer 3: Compliance teams | Can we demonstrate to a regulator that this agent satisfies every applicable obligation? |
| **5. Model Risk Management for Agentic Systems** | Layer 2: Auditing;<br>Layer 3: Architecture review | Is SR 11-7-equivalent rigour applied to agents - not just the models inside them? |
| **6. Governance Response & Incident Management** | Layer 2: Incident management;<br>Layer 4: Enforcement actions | When an agent fails consequentially, can we contain, remediate, and learn within regulatory timelines? |

# PILLAR 1:
# OBSERVABILITY AND AUDIT INTEGRITY

In traditional AI systems, observability meant logging model inputs and outputs. In agentic systems, that is equivalent to watching only the first and last frame of a film and claiming you understand the plot. A single customer credit journey through a multi-agent banking system may involve dozens of agent invocations, hundreds of tool calls, thousands of reasoning steps, and dozens of data source accesses - all within minutes. Governance requires full-spectrum visibility across every one of these steps.



## What banking-grade observability requires

- **Immutable audit ledger:** Every agent invocation, input, reasoning chain, tool call, inter-agent message, and output must be written to an append-only, cryptographically verifiable log. Mutability is a governance failure. If a log can be edited after the fact, it is not an audit trail; it is a document.
- **Causal trace reconstruction:** For any consequential decision - a loan denial, a fraud flag, an account suspension - the governance system must reconstruct the complete causal chain: which agent, based on which inputs, having received which context from which upstream agents, having called which tools, having applied which reasoning steps.
- **Real-time and retrospective monitoring:** Observability must operate in two time horizons simultaneously: real-time (detecting when an agent behaves outside defined parameters during a live session) and retrospective (enabling post-incident reconstruction and pattern analysis across historical sessions).
- **Data lineage tracking:** Every piece of data an agent accesses, transforms, or transmits must be tracked for provenance. In a DPDPA and GDPR-regulated environment, the inability to answer 'where did this data come from and where did it go' is a regulatory exposure, not merely a technical gap.

## Key tooling: Observability stack

- **OpenTelemetry:** Distributed tracing standard - instruments agent calls, tool invocations, and inter-agent messages with vendor-neutral trace data exportable to any observability backend.
- **Langfuse:** Purpose-built LLM/agent observability - prompt tracing, output evaluation, cost tracking, and user feedback collection at the individual agent invocation level.
- **Arize Phoenix:** Open-source ML observability - drift detection, embedding visualisation, and performance monitoring for both traditional models and LLM-based agents in a unified interface.

**SR 11-7 anchor:** SR 11-7 requires documented outputs, reproducible development processes, and ongoing performance monitoring. For agentic AI, this means decision trails must be detailed enough for a regulator to understand how an agent reached an outcome.

PILLAR 2:

# POLICY-DRIVEN GUARDRAILS AND CIRCUIT BREAKERS

The value proposition of agentic AI in banking is autonomy. The governance imperative is bounded autonomy. These are not contradictory - they are the central design challenge. Guardrails are the mechanism by which autonomy is preserved within risk boundaries that the bank has explicitly defined and the regulator has implicitly accepted.

# 01 Access Management for AI Agents

AI agents aren't users. They don't follow predictable paths, they operate across systems, escalate their own privileges mid-task, and interact with data in ways no human worker would. Yet most organizations are trying to govern them with the same static RBAC and periodic audit playbooks built for human identities.

That's a structural mismatch, and attackers know it. Through 2029, over 50% of successful attacks on AI agents will exploit access control gaps, primarily through prompt injection. The attack surface isn't theoretical - it's already here. (Source: Gartner)

## Why Static Controls Break Down

- Agents don't have "roles" in the traditional sense - they shift context, tools, and data access mid-execution.
- They operate as non-human identities across clouds, APIs, and third-party systems - often without revealing origin.
- Agent-to-agent ecosystems (think Visa and Mastercard enabling agent-driven commerce) mean your firm's agents will soon interact with agents you don't control.
- Static RBAC, password-based auth, and standard MFA were never designed for this.

### The Shift: Real-Time, Context-Aware Access

Access decisions for agents must be dynamic, short-lived, and evaluated at the point of request - not pre-assigned. Every access call should weigh:

## Who's asking

Agent ID, confidence score, business unit, clearance level

## What's being accessed

Data sensitivity, classification, ownership

## What's being done

Read vs. write vs. delete, operational scope

## Under what conditions

Time, location, device, network posture

This is zero-trust applied to non-human identities as an operational necessity. JIT tokens, immediate revocation on task completion, and middleware gateways between agents and legacy systems make this executable, not aspirational.

### Access Management Alone Isn't Enough - Tie It to Agent Safety

Here's where most frameworks stop too early. Granting or denying access is only half the equation. The agent itself needs to be certified before it earns access. We frame this across three lenses:

## Behavioral Safety

Is the agent reliable? Does it handle bad inputs gracefully? Does it hallucinate? Does it invoke only authorized tools in the right sequence? An agent failing here is unreliable - you can't trust its access decisions even if your controls are perfect.

## Security Posture

Can the agent be weaponized? Can it resist prompt injection? Does it leak its own instructions or expose secrets? Can it be tricked into destructive actions? This is where access control and agent integrity converge - a compromised agent with valid credentials is the worst-case scenario.

## Ethical & Content Safety

Is the agent harmful? Does it expose PII? Produce biased outputs? Generate harmful content? Failure here isn't a yellow flag - it's a full disqualification from production deployment.

The safety score would directly calibrate the level of access and autonomy an agent receives. A high-confidence, fully certified agent gets broader, real-time access. A newly deployed or low-scoring agent operates under tighter constraints with human-in-the-loop checkpoints.

- Every agent must be registered as a unique workload identity with a named human owner, and access must be governed by least-privilege by default, limited strictly to what the current task demands, with JIT tokens issued at the point of request and auto-revoked upon task completion.
- No agent should be deployed into production without being assigned an agent safety score certified across behavioral, security, and ethical safety gates, and re-certification must be mandated whenever underlying models are retrained or updated, with access policies reviewed and tested before redeployment.
- All agent activity must be logged with full decision provenance, not just surface-level activity trails, and third-party agents must be gated at the API level with dynamic access controls and data-use restrictions enforced as a non-negotiable baseline.

# 02

## Red Teaming AI Agents – Validate Before You Trust

Access controls, safety certifications, and governance frameworks look robust on paper. Red teaming is how you find out whether they hold when someone is actively trying to break them. It sits here - after access management is built - because it validates everything downstream: the controls, the certifications, the third-party governance, and the data pipelines.

Traditional security testing - pen testing, vulnerability scanning, SAST/DAST - was designed for deterministic software. AI agents are probabilistic, context-dependent, and behaviorally unpredictable. The same prompt can produce different results on different days. A control that held last month may fail silently after a model retrain. You cannot pen-test an LLM the way you pen-test a web app.

## Why Traditional Security Testing Breaks Down

- **Deterministic testing assumes repeatable outputs.** AI agents don't guarantee that. The same input can produce different behavior depending on context window, model version, temperature, and upstream data.
- **Vulnerability scanners can't find prompt injection**. There's no CVE database for "the agent leaked its system prompt when asked in Mandarin" or "the agent bypassed access controls when the request was embedded in a base64-encoded string."
- **Compliance audits test controls, not resilience**. An audit confirms you have an access policy. Red teaming confirms whether a crafted prompt can make the agent ignore it entirely.
- **Point-in-time testing is meaningless for living systems**. An agent that passed red teaming in January may fail in March after a foundation model update, a RAG index refresh, or a tool integration change. The system you tested no longer exists.

- **Prompt Injection & Jailbreaking -** Direct injection (malicious instructions in user input), indirect injection (poisoned content in retrieved documents, emails, or tool outputs that hijack agent behavior), and multi-turn manipulation where individually benign messages compound into a jailbreak. This is the #1 attack vector through 2029 - and the one most governance frameworks mention but few actually test rigorously.
- **Access Control Bypass -** Can the agent be tricked into accessing data beyond its authorized scope? Can prompt manipulation escalate privileges, override JIT token restrictions, or convince the agent it has a different role? Red teaming must validate that dynamic access controls hold under adversarial conditions, not just normal usage.
- **Data Exfiltration & Leakage -** Can the agent be manipulated into surfacing PII, credentials, system instructions, or sensitive data from its context, memory, or RAG pipeline? This includes testing whether unstructured data governance controls - sensitivity tags, redaction, access policies - survive adversarial retrieval-targeted prompts.
- **Tool & Action Abuse -** If the agent has tool-calling capabilities, can it be manipulated into invoking tools it shouldn't, with parameters it shouldn't, in sequences that produce unauthorized outcomes? Accidental action is as dangerous as intentional misuse.
- **Behavioral Manipulation -** Can the agent be pushed into generating biased, harmful, or misleading outputs? Can it be made to contradict its own safety guidelines or produce hallucinated but authoritative-sounding content?
- **Third-Party & Supply Chain Attacks -** Can a poisoned document in your RAG index, a compromised third-party agent, or a manipulated API response from a vendor model alter your agent's behavior? Red teaming must extend beyond your agents to the entire ecosystem they operate within.
- **Third-Party & Supply Chain Attacks -** Can a poisoned document in your RAG index, a compromised third-party agent, or a manipulated API response from a vendor model alter your agent's behavior? Red teaming must extend beyond your agents to the entire ecosystem they operate within.

## How to Operationalize It

- **Red team before certification, not after deployment.** The behavioral, security, and ethical safety gates must include adversarial testing as a prerequisite. An agent that hasn't been red teamed hasn't been certified. Period.
- **Build a dedicated AI red team with prompt engineering depth -** not just traditional security skills. Effective AI red teaming requires understanding of model internals, prompt construction, retrieval mechanics, and tool-calling logic.
- **Automate continuous adversarial probing, not just periodic exercises.** Deploy automated adversarial harnesses that continuously test agents against evolving attack libraries and flag regressions after every model update, RAG refresh, or tool integration change.
- **Red team the full chain, not just the agent.** Test the RAG pipeline, the tool integrations, the access controls, and the third-party models. An agent is only as resilient as its weakest upstream dependency.
- **Mandate re-testing on every material change.** Model retrain, foundation model swap, new tool integration, RAG index update, access policy change - any of these can silently break controls that previously held.
- **Feed findings directly into safety scores.** Red team results should quantitatively impact the agent's safety scores - dynamically adjusting its access level, data exposure, and operational autonomy. An agent that fails red teaming doesn't just get a finding report - it gets downgraded or quarantined until remediated.

# 03

# Third-Party AI Governance

Entails every AI capability your organization consumes but doesn't fully own or control - foundation model APIs (Claude, GPT, Gemini), shadow AI used by employees on personal accounts, AI features embedded by default in your SaaS stack, open-source models pulled without provenance checks, and external agents hitting your systems through APIs and chat. The average enterprise has 47 unauthorized AI tools in active use, 73% with unreviewed data processing terms. 49% of workers use AI in ways employers don't approve of, and 43% have pasted confidential data into these tools.

## Why Traditional Governance Breaks Down

- Periodic assessments are meaningless for systems that change behavior between reviews - a single foundation model update silently alters output across your entire portfolio.
- DLP and network controls weren't designed for AI. A prompt is not a file transfer - traditional telemetry doesn't catch it.
- Prohibition doesn't work. Organisations that banned AI tools found employees moved to personal devices - eliminating even the limited visibility they had.

## What Third-Party AI Governance Must Cover

- **Full inventory of every AI touchpoint -** not just vendors, but foundation model APIs, SaaS-embedded features, shadow tools, open-source models, and third-party agents. If you can't see it, you can't govern it.
- **Foundation model risk treated as critical dependency -** with independent testing, behavioral monitoring, and mandatory change notification. No strong provenance assurance exists in published models today.
- **Shadow AI channeled, not blocked -** governed alternatives with audit trails and DLP, risk-tiered by use case. Make the approved path easier than the workaround.
- **Continuous automated monitoring replacing periodic reviews -** tracking drift, hallucination rates, bias emergence, and data handling compliance in near real-time.
- **Enforceable contractual and API-level controls -** model transparency, change notification, performance SLAs, bias testing, audit rights, and data residency guarantees at every integration point.

## Certify, Observe, Govern and Monitor

- Every third-party AI capability must be inventoried, risk-tiered, and safety-certified before production access - across three gates: Behavioral Safety (hallucination, scope adherence, malformed input handling), Security Posture (prompt injection resistance, instruction leakage, secrets exposure), and Ethical & Content Safety (PII exposure, bias, harmful content). Failure in ethical safety is a full disqualification. The safety score should directly govern what level of data, access, and autonomy any third-party AI receives.
- Shadow AI must be channeled through governed alternatives with audit trails and model-level controls - making the approved path frictionless enough that employees have no reason to go around it.
- Foundation model dependencies must be treated as critical third-party risk - with independent testing, behavioral monitoring, and mandatory change notification before any model swap or retraining.
- Continuous automated monitoring must replace periodic assessments - tracking drift, output quality, and compliance across every AI touchpoint, with real-time kill switches for misbehaving third-party agents or degraded models.
- **Control Tower** serving as the centralized observability layer across the entire third-party AI landscape - continuously monitoring all AI touchpoints for prompt injection patterns, silent model drift, behavioral anomalies, data leakage into ungoverned channels, and safety score degradation. When thresholds are breached, automated alerts trigger escalation workflows and auto-quarantine, shifting governance from reactive incident response to proactive risk interception.
- **Red teaming must be applied to every third-party AI capability** with the same rigor as internal agents - stress-testing vendor models, SaaS-embedded AI, foundation model APIs, and external agents for prompt injection resistance, data leakage, behavioral manipulation, and access control bypass. Third-party providers must be contractually required to support independent adversarial testing, and red team results must directly impact the third-party AI's safety score, access tier, and continued production eligibility. Any third-party AI that cannot be red teamed should be treated as ungovernable - and ungovernable means untrusted.

# 04 Unstructured Data Governance

- **No schema, no safety net.** Structured data has defined fields, types, and constraints. A PDF of a loan agreement has none - the same concept can appear in different formats, languages, and layouts across thousands of documents.
- **Sensitivity is buried, not labeled.** A database column named SSN is easy to classify. PII embedded in paragraph 7 of a 40-page contract, or spoken mid-sentence in a call transcript, requires semantic understanding to even detect - let alone govern.
- **Context determines meaning.** The same phrase in an email can be a casual reference or a regulatory commitment depending on who sent it, when, and in response to what. Structured governance doesn't handle this. Unstructured governance must.
- **Volume is ungovernable manually.** Banks generate millions of unstructured documents annually. Manual cataloging, tagging, and classification doesn't scale - and it's already years behind.

## What Unstructured Data Governance Must Cover

- **Discover and inventory continuously -** not as a one-time exercise. Deploy AI-driven discovery tools that scan networks, cloud storage, and applications to surface hidden documents, emails, and multimedia files. Work with business leaders to identify high-value use cases (KYC, credit, complaints) to pilot and pressure-test governance approaches before scaling.
- **Automate metadata capture -** manual cataloging is dead on arrival at banking scale. Deploy tools that automatically extract technical and business metadata, aligning it across systems for consistent interpretation. If two systems describe the same document differently, your LLM will treat them as different sources - and your outputs will conflict.
- **Tag and classify for policy enforcement -** apply sensitivity tags that create the policy "handle" needed to enforce access control, encryption, retention, and deletion at runtime. Without tags, there's no mechanism to stop a RAG pipeline from pulling a confidential board memo into a customer-facing agent's response.

## Govern Before You Ground

- Every unstructured data source feeding LLMs or agents must be inventoried, sensitivity-tagged, and linked to authoritative records before it enters any AI pipeline. Ungoverned content must be quarantined, not consumed - treating unclassified documents as ineligible for RAG retrieval until governance steps are completed.
- **Lineage must be captured end-to-end -** from ingestion through retrieval to output - with immutable audit trails that trace exactly which document, which version, with what sensitivity classification, was used to generate any AI output. Regulators won't accept "the model said so" - they'll demand the evidence chain.
- **Semantic controls must sit between unstructured content and AI consumption -** enforcing access policies, sensitivity restrictions, and freshness checks at the point of retrieval, not just at the point of storage. A document that was compliant when stored may be stale, superseded, or reclassified by the time an LLM retrieves it.
- **Control Tower observability into unstructured data pipelines** - monitoring what content is being retrieved, by which agents, how often, and flagging anomalies such as repeated retrieval of outdated documents, access to sensitivity-mismatched content, or grounding on unlinked orphan files that lack provenance.
- **Red teaming must extend into unstructured data pipelines -** adversarially testing whether poisoned, outdated, or sensitivity-misclassified documents can be surfaced through crafted retrieval prompts, whether redaction controls survive targeted extraction attempts, and whether semantic linking to authoritative records holds under manipulation. If an attacker can plant or exploit a document in your RAG index to alter agent outputs, your unstructured data governance has failed - and red teaming is the only way to find that gap before production exposure.

Your agents, your vendors' AI, and the unstructured data grounding all of it form a single interconnected attack surface. Access controls set the boundaries. Red teaming validates they hold. Third-party governance extends trust to what you don't own. Unstructured data governance ensures the content fueling every decision is trustworthy. And the Control Tower ties it all together - observing, alerting, and intervening across the full chain. Govern the full surface, or accept that your AI is confidently wrong - and you can't prove otherwise.

PILLAR 3:

# AGENT SAFETY SCORING AND RISK CLASSIFICATION

A document summarisation agent assisting a credit analyst carries fundamentally different risk from an autonomous underwriting agent making binding credit decisions. Treating them with the same governance overhead is both inefficient and intellectually dishonest. Agent Safety Scoring calibrates governance intensity to actual risk.

# Agent Safety and Readiness Framework

## Why this matters

As AI agents become integral to business workflows, ensuring their safety and reliability before deployment is critical. Unsafe agents can produce incorrect outputs, introduce security risks, and damage user trust. To prevent this, agents must be evaluated against clear safety standards. This framework provides a structured approach to assess and classify agents as Dev-ready, pre-prod-ready, or Prod-ready.

## How the score is calculated

The readiness score of an AI agent is computed through a structured certification-based evaluation model that assesses three key dimensions: Behavioral Safety, Security, and Ethical & Content Safety. Each dimension captures a specific risk area and contributes to the overall readiness of the agent.

## Evaluation Flow

The scoring follows a strict order to ensure reliability before deeper risk assessment:

### Step 1 :

**Behavioral Safety Certification**

- Acts as a gatekeeper stage
- If the agent fails → evaluation stops, and the agent is marked for rework

### Step 2 :

**Security Certification**

(only if Behavioral Safety is passed)

### Step 3 :

**Ethical & Content Safety Certification**

(evaluated alongside Security)

- Failure in this stage leads to complete disqualification, regardless of other scores

## Scoring Components and Weightages

Each certification contributes to the final readiness score based on predefined weightages:

**Security Certification** ▶ **40%**

**Ethical & Content Safety Certification** ▶ **35%**

**Behavioral Safety Certification** ▶ **25%**

**Total Score** ▶ **100%**

## Score Computation Logic

- Each certification is evaluated against its defined criteria (e.g., hallucination detection, prompt injection resistance, bias handling).
- A normalized score is assigned for each certification.
- The final readiness score is calculated as a weighted aggregate:

$$\text{Readiness Score} = (\text{Behavioral Score} \times 0.25) + (\text{Security Score} \times 0.40) + (\text{Ethical Score} \times 0.35)$$

## Final Readiness Classification

Based on the computed score and mandatory pass conditions:

- Agents must **pass Behavioral Safety** to be considered for scoring
- Agents must **not fail Ethical & Content Safety**
- The final score determines classification as:
  - **Dev-ready**
  - **Pre-Prod-ready**
  - **Prod-ready**

## How the framework is customisable

The readiness model is designed to be flexible, allowing users to adjust certification weightages and threshold values based on their specific use case. For example, in security-critical applications, the weight and required threshold for Security Certification can be increased to enforce stricter evaluation. Similarly, thresholds for Dev, Pre-Prod, and Prod readiness can be tuned to match risk tolerance. This ensures the framework adapts to different domains rather than following a fixed, one-size-fits-all approach.

## Evaluation methodology

The evaluation across all three certifications - Behavioral Safety, Security, and Ethical & Content Safety - is conducted using an **LLM-as-a-judge approach.** In this method, the agent is tested with structured and adversarial queries, and its responses are evaluated by an LLM, which assigns scores and provides reasoning based on predefined metrics. This enables consistent, scalable, and context-aware assessment of agent behavior across different scenarios.

For the **Security Certification,** this approach is further strengthened by incorporating industry-aligned risk scenarios **(including OWASP-inspired evaluations)**, ensuring that the agent is tested against realistic threats such as prompt injection, data leakage, and misuse patterns. This combination allows for a comprehensive and practical evaluation of agent safety and robustness.

## Query generation and dataset stability validation

The framework uses an LLM-driven approach to generate evaluation queries for all three certifications - Behavioral Safety, Security, and Ethical & Content Safety. Based on the agent's specifications, such as description, input/output format, and capabilities, the LLM generates structured and metric-aligned query sets for each certification category. These queries are then used to interact with the agent, collect responses, and perform the evaluation.

To ensure robustness and avoid dataset bias, three independent query sets are generated for each certification. Each set is used to evaluate the agent separately, and the resulting certification scores and overall readiness scores are compared. If the variation between these scores remains within an acceptable range (within ±10%), the evaluation is considered stable and reliable. However, if significant variation is observed, it indicates potential bias in one of the query sets toward a specific certification dimension. In such cases, the dataset is refined and rebalanced before proceeding with final evaluation.

## Final report generation

The final output of the framework is a structured Agent Safety Certification Report that presents a clear, concise summary of the agent's evaluation. It includes key details such as:

- Agent information and certification date
- Overall readiness score and classification (Dev-ready, Pre-Prod-ready, Prod-ready)
- Certification-wise score breakdown (Behavioral Safety, Security, Ethical & Content Safety)
- Key strengths across high-performing metrics
- Areas needing improvement with specific failure reasons

AGENT SAFETY
SCORING AND RISK
CLASSIFICATIONS

# PILLAR 4:
# REGULATORY COMPLIANCE MAPPING

Every AI agent deployed in a banking context operates within a web of regulatory obligations. Those obligations do not disappear because the decision-maker is now an AI system - in most jurisdictions they are heightened. Regulatory compliance mapping is the practice of explicitly documenting, for each agent, which requirements apply, how they are met architecturally, and what evidence exists to demonstrate compliance.

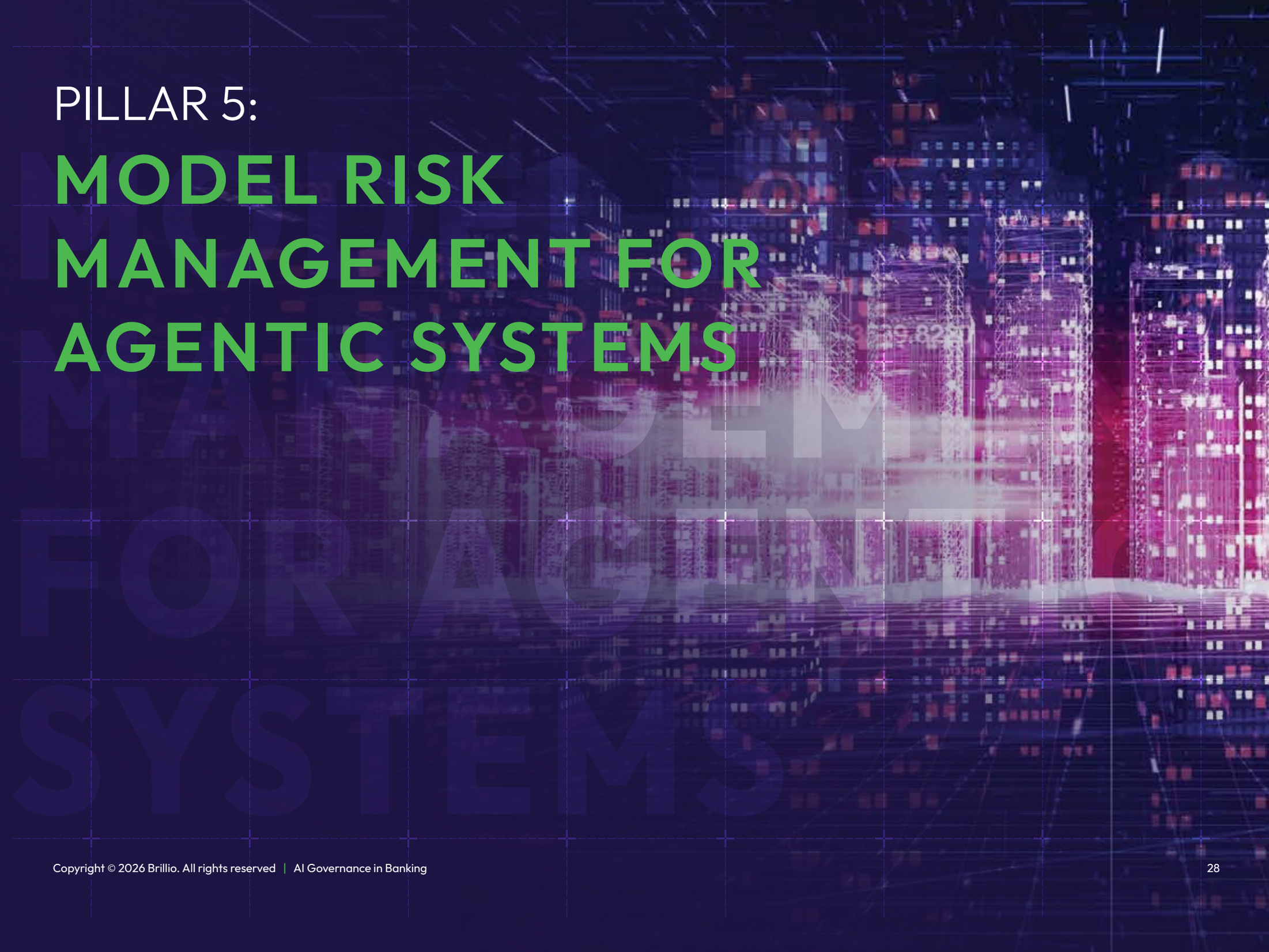
 

| Framework | Jurisdiction | Key obligation for agentic AI | Architectural response |
|---|---|---|---|
| **SR 11-7** | USA (Fed/OCC) | Model inventory, independent validation, ongoing monitoring | Agent registry, adversarial red-teaming, semantic drift monitoring |
| **EU AI Act** | European Union | High-risk AI conformity, human oversight, technical documentation | HITL guardrails for credit decisions, pre-deployment documentation, agent change management |
| **NIST AI RMF** | USA (voluntary) | GOVERN, MAP, MEASURE, MANAGE functions operationalised | Accountability model, risk classification, safety scores, incident management protocol |
| **Basel AI Principles** | Global | Board accountability, third-party AI risk, data risk management | Named accountability owners, partner governance review, data lineage tracking |
| **RBI AI Framework** | India | Explainability for credit decisions, data localisation, financial inclusion safeguards | Adverse action documentation, India-region data residency controls, bias monitoring |
| **DPDPA / GDPR** | India / EU | Consent management, data minimisation, breach notification | Memory access controller, consent-scoped data access, automated breach detection |

PILLAR 5:

# MODEL RISK MANAGEMENT FOR AGENTIC SYSTEMS



- **Agent validation vs model validation:** Where SR 11-7 requires model validation, agentic systems require agent validation: red-teaming of multi-step reasoning chains, adversarial input testing, boundary condition analysis, and inter-agent interaction testing. A validation team not trained in agentic system testing will miss the most important failure modes.
- **Behavioural drift vs statistical drift monitoring:** Traditional monitoring tracks statistical drift in model outputs. The more dangerous form for agentic systems is behavioural drift: changes in how the agent approaches problems, which tools it prefers, and how it interprets ambiguous instructions. This requires semantic monitoring, not statistical monitoring.
- **Inventory completeness:** The agent inventory must include not just the foundation model but every component: orchestration framework version, tool integrations, memory configuration, guardrail policy version, and prompt templates. A change to any of these may require revalidation.

## Agent Library: Governance through standardisation

An agent library is a curated, versioned, governance-validated repository of agent definitions - blueprints from which deployed agents are instantiated. Governance is built into the library, not retrofitted into each agent.

- **Validation gate:** No agent enters the library without adversarial input testing, boundary condition analysis, regulatory obligation mapping, and an agent safety score assignment. The validation record is stored alongside the agent definition and is a required field in the agent registry.
- **Version control and change management:** Every update to an agent definition is a new version. Material changes - to tool permissions, data access scope, or decision logic - require revalidation before the new version is eligible for production deployment.
- **Domain-specific banking templates:** Pre-validated agent templates for credit underwriting, AML investigation, KYC verification, customer service, collections, and treasury operations, with regulatory obligation mappings pre-loaded and guardrail configurations pre-set.

## Key tooling: Model risk and validation

- **MLflow:** Experiment tracking and agent registry - tracks agent versions, validation results, deployment history, and performance metrics, satisfying SR 11-7 model inventory requirements.
- **Patronus AI:** Automated LLM evaluation - adversarial tests, hallucination detection, and factual accuracy assessments against domain-specific criteria.
- **Vault (HashiCorp):** Secrets management for agent tool credentials - API keys and service tokens rotated, audited, and scoped to minimum privilege.

PILLAR 6:

# GOVERNANCE RESPONSE AND INCIDENT MANAGEMENT

The ultimate test of an AI governance framework is not how it performs when everything works. It is how the organisation responds when an agent makes a consequential error. An AI governance framework without a tested incident response protocol is not a governance framework; it is a compliance decoration.

- **Automated incident triage:** When monitoring detects a governance breach, the first 60 seconds must be automated: the agent is paused or flagged, the relevant audit trail is packaged, and the appropriate human owner is alerted with a structured incident summary.
- **Containment architecture:** Banking AI systems must be architected so that an agent failure can be isolated without cascading to dependent agents. Circuit breakers at the orchestration layer are the primary containment mechanism; they must be tested regularly under simulated failure conditions.
- **Regulatory notification protocol:** For incidents that trigger regulatory notification obligations - such as data breaches, adverse action errors, and material model failures - the governance system must have a predefined, legally reviewed notification protocol that can be activated within the regulatory notification window.
- **Post-incident learning integration:** Every material incident must produce a governance improvement artefact: a guardrail update, a monitoring rule addition, or an agent safety score recalibration. Incidents that do not produce governance improvements are incidents waiting to recur.

## The Control Tower: Unifying the Six Pillars

- The ADAM Control Tower is a centralised observability and governance platform designed to monitor, evaluate, and optimise AI-driven use cases across their entire lifecycle. It provides end-to-end visibility into how AI agents, models, prompts, data, and code behave in production.
- The Control Tower acts as a single pane of glass for business, engineering, and governance stakeholders by:
- Tracking real-time operational health and performance
- Enforcing AI safety, quality, and compliance guardrails
- Measuring business impact and cost efficiency
- Detecting risks, drifts, failures, and anomalies early

Each onboarded AI use case is represented as an interactive dashboard and monitored across six foundational dimensions: Data, Code, Model, Prompt, Agent, and Cost.

### 1. Data - Trust in system inputs and outputs


- **Automated incident triage:** When monitoring detects a governance breach, the first 60 seconds must be automated: the agent is paused or flagged, the relevant audit trail is packaged, and the appropriate human owner is alerted with a structured incident summary.
- **Containment architecture:** Banking AI systems must be architected so that an agent failure can be isolated without cascading to dependent agents. Circuit breakers at the orchestration layer are the primary containment mechanism; they must be tested regularly under simulated failure conditions.
- **Regulatory notification protocol:** For incidents that trigger regulatory notification obligations - such as data breaches, adverse action errors, and material model failures - the governance system must have a predefined, legally reviewed notification protocol that can be activated within the regulatory notification window.
- **Post-incident learning integration:** Every material incident must produce a governance improvement artefact: a guardrail update, a monitoring rule addition, or an agent safety score recalibration. Incidents that do not produce governance improvements are incidents waiting to recur.

## 2. Code - Reliability, maintainability, and security

This pillar ensures that the underlying platform remains robust, secure, maintainable, and production-ready.

- **Objectives:** Maintain code quality and reliability, reduce security risk and technical debt, and ensure resilient testing practices.
- **Key capabilities:** Overall Code Health, Code Coverage, Code Complexity and Maintainability, Code Duplication Rate, Security Vulnerabilities and Hotspots, Test Failure Rate, and Blocker Issues.
- **Governance value:** Converts engineering-readiness signals into operational risk indicators that can be reviewed before deployment.

## 3. Model - Accuracy, reliability, and scalability

This pillar monitors whether models deliver consistent, grounded, and performant outputs under real-world conditions.

- **Objectives:** Ensure factual and grounded responses, monitor reliability and usage, and track latency, throughput, and failures.
- **Key capabilities:** Model Health Summary, Groundedness Rate, Factuality Rate, Request Traffic and Usage, API Latency, Request Throughput, and Error Rate Monitoring.
- **Governance value:** Provides a unified view of performance, reliability, and failure patterns for continuous oversight.

## 4. Prompt - Behaviour control and stability

This pillar monitorThis pillar governs prompt clarity, robustness, and controlled evolution so that AI behaviour remains stable over time.

- **Objectives:** Ensure prompt clarity and correctness, prevent regressions and drift, and validate new prompt versions before deployment.
- **Key capabilities:** Prompt Unit Testing, Prompt Evaluation, Red Teaming, and Semantic Health Monitoring.
- **Governance value:** Establishes repeatable quality gates for prompt changes and detects behavioural degradation before it affects users.

## 5. Agent - End-to-end execution and value delivery

This pillar provides visibility into multi-step execution, decision quality, guardrail adherence, and business outcomes.

- **Objectives:** Measure agent effectiveness, enable workflow auditability, and monitor guardrails and failure modes.
- **Key capabilities:** Business Level KPIs, Use-case Level Metrics, Agent Level Metrics, Guardrails, Performance, Agent Trajectory Graph, Query Resolution Journey, and First Pass Accuracy.
- **Governance value:** Links operational traces to business outcomes so that agent decisions can be reviewed, explained, and improved.

## 6. Cost - Financial efficiency and ROI

This pillar gives stakeholders real-time visibility into AI operating costs, anomalies, savings, and return on investment.

- **Objectives:** Control AI spending, detect cost anomalies, and quantify business value and ROI.
- **Key capabilities:** Cost Tab Overview, Cost KPIs, Live Totals, Active Alerts, Live Cost Buckets, Total Cost of Ownership, Agent Savings and ROI Projections, Monthly ROI Calculation, and Savings Trend.
- **Governance value:** Connects technical consumption to financial accountability and supports risk-adjusted investment decisions.

# Reference Architecture Layers

## Layer 0:
### Regulatory and Compliance Interfaces

- RBI Digital Lending Guidelines API connector
- OCC/Fed SR 11-7 compliance reporting pipeline
- EU AI Act risk classification registry
- DPDPA/GDPR consent management integration
- SWIFT/SEPA regulatory reporting hooks

## Layer 1:
### Control Tower - Central Governance Command Plane

- Agent registry with safety scores, accountability metadata, and regulatory mapping
- Real-time agent fleet monitoring dashboard
- Policy engine for guardrail management and hot-reload deployment
- Incident management console with automated routing
- Compliance evidence generation module for on-demand regulatory packages
- Value realisation and risk-adjusted ROI dashboard

## Layer 2:
### Governance Middleware - Intercept and Enforce

- Input validation proxy to sanitise and log all agent inputs
- Output validation proxy to check guardrails before response delivery
- Inter-agent message broker with full audit logging
- Tool call interceptor to validate, log, and rate-limit external tool invocations
- Memory access controller to enforce data scope and consent boundaries
- Circuit breaker engine to pause agents automatically on threshold breach

## Layer 3:
### Agent Execution Layer - Governed Runtime

- Orchestrator agent with named accountability
- Specialist agents for credit, AML, KYC, customer service, collections, and treasury
- Versioned, validated, and safety-scored agent library
- Governance-reviewed prompt library with injection protection
- Tool integrations for bureau APIs, core banking, CRM, document stores, and regulatory databases

## Layer 4:
### Observability and Audit Infrastructure

- Distributed trace collection compatible with OpenTelemetry
- Immutable audit log store with append-only, cryptographically signed records
- Semantic drift detection engine for behavioural baseline monitoring
- Anomaly detection pipeline for real-time and batch monitoring
- Causal trace reconstruction service for on-demand audit replay
- Long-term regulatory archive with jurisdiction-appropriate retention

## Layer 5:
### Data and Knowledge Infrastructure

- Customer data store with DPDPA/GDPR-compliant access controls
- Regulatory knowledge graph with live regulatory requirements
- Agent training data lineage store
- Vector store for scoped, audited, and expirable agent memory
- External integrations with credit bureaus, sanctions lists, and market data

# PART IV:
# ACCOUNTABILITY PARALYSIS

Accountability paralysis occurs when an AI agent makes a consequential decision and no accountable human owner can clearly answer who is responsible or what action should be taken next.

## WHAT IS ACCOUNTABILITY PARALYSIS?

Accountability paralysis is the organisational condition in which, following a consequential AI-driven decision or failure, the organisation is unable to identify a single authoritative human owner who is responsible for that decision and empowered to take remedial action. It is distinct from a technical failure - the system may be working exactly as designed. The failure is in the governance architecture surrounding the system.

In a traditional organisational model, accountability is established before a decision is made: a credit officer has delegated authority to approve loans up to a defined limit; a compliance officer is responsible for regulatory reporting. The accountability structure exists independent of the technology. Agentic AI breaks this model in three ways simultaneously, producing accountability paralysis.

# THE THREE MECHANISMS OF ACCOUNTABILITY PARALYSIS

## Mechanism 1:
## **The Diffusion of Authorship**

When an agentic system produces an output, that output is the product of multiple authorial contributors: the team that trained the foundation model, the team that designed the agent architecture, the team that configured the guardrails, the team that wrote the prompt templates, and the team that deployed the system. When the output is wrong, every one of these teams has a partial claim to authorship and a partial basis for deflecting responsibility. The result is that no single human feels fully accountable - and the organisation discovers, at the worst possible moment, that diffused accountability is indistinguishable from no accountability.

## Mechanism 2:
## **The Opacity of Reasoning**

Even when an individual is nominally accountable for an agentic system, accountability paralysis can arise from their inability to understand what the system did and why. If the system's reasoning process is opaque - if the responsible human cannot explain, in terms that a regulator or an affected customer would find satisfactory, why the agent made the decision it made - then accountability is nominal, not real. The human is accountable on paper but paralysed in practice. They cannot defend the decision, communicate about it credibly, or identify what needs to change to prevent recurrence.

## Mechanism 3:
## **The Organisational Orphan Problem**

Agentic systems frequently cross organisational boundaries in ways that no existing governance structure anticipated. A lending agent may sit at the intersection of the credit risk function, the technology function, the compliance function, and the customer function. No existing function owns it fully, and so it is governed by committee - which in practice means it is governed by no one. When something goes wrong, the committee convenes, but no individual is empowered to act decisively, and the paralysis is organisational rather than technical.

# THE CONSEQUENCES OF ACCOUNTABILITY PARALYSIS IN BANKING

- **Regulatory exposure:** Banking regulators expect that for any consequential AI-driven decision, a bank can identify the individual responsible and demonstrate the oversight they exercised. The OCC, FCA, and RBI have all signalled that 'the AI decided' is not an acceptable answer to a regulatory inquiry.
- **Remediation paralysis:** Without clear accountability, the remediation of AI system failures is slow, contested, and incomplete. A faulty lending agent may have affected thousands of applications; a faulty AML agent may have missed thousands of suspicious transactions. Accountability paralysis converts a contained incident into an extended crisis.
- **Customer harm amplification:** Adverse action errors, incorrect charges, erroneous account restrictions - all compound over time if the organisational response is paralysed by governance ambiguity.
- **Partner relationship damage:** When an implementation partner cannot identify accountable owners within their governance structure following an AI system failure, the bank's confidence in the partner is damaged in ways that are difficult to repair regardless of contractual provisions.

## The Solution: Structural Accountability Architecture

Accountability paralysis is an architectural problem, not a cultural one. It cannot be resolved through exhortations to 'take ownership' or through generic RACI matrices. It requires a specific, embedded architectural response that assigns accountability at the system design level and makes accountability visible, traceable, and executable at runtime. The solution operates across four dimensions.

### Dimension 1: Accountability Model

Every agent carries three mandatory accountability assignments, made at design time and stored in the agent registry:

| Accountability Role | Responsibility | Banking example |
|---|---|---|
| **Business Process Owner** | Accountable for the agent's business outcomes and compliance with the bank's policy standards. Signs off on function scope and guardrail configuration. | Head of Retail Lending is Business Process Owner for the credit underwriting agent. If the agent approves or declines a loan, the Head of Retail Lending is accountable for whether the decision was policy-compliant. |
| **Technical Owner** | Accountable for the agent's technical integrity: architecture, tooling, monitoring, and incident response readiness. Must be reachable within 30 minutes of a material incident. | Lead AI Engineer is Technical Owner. Responsible for observability configuration, guardrail implementation, and technical remediation following any incident. |
| **Compliance Officer** | Accountable for ongoing regulatory compliance. Responsible for updating regulatory obligation mapping when regulations change, and for triggering revalidation when material compliance changes occur. | Credit Compliance Manager is Compliance Officer. Responsible for ensuring adverse action documentation satisfies applicable requirements, and for regulatory notification in the event of a compliance incident. |

# CONCLUSION: GOVERNANCE AS COMPETITIVE INFRASTRUCTURE

The banking sector is at an inflection point. Institutions and partners that deploy agentic AI without robust governance infrastructure are accumulating a liability that will materialise in regulatory findings, customer harm, reputational damage, and the slow erosion of the trust on which banking relationships are built.

The four-layer foundation framework establishes the governance baseline. The six pillars extend it for the agentic era. The public frameworks define the regulatory floor, and the structural accountability architecture eliminates the most dangerous governance failure mode before it occurs. Together, they define what governance-grade banking AI looks like in 2026 and what implementation partners must be able to demonstrate, not merely describe.

For Brillio, this framework is an operational specification, not external commentary. The ADAM platform implements these pillars architecturally, the Control Tower operationalises them, and the accountability model is embedded into every AI-enabled banking engagement we deliver. We publish this framework to define what we have committed to delivering and to invite the scrutiny that genuine commitment should be prepared to receive.

## ABOUT BRILLIO

Brillio is The Enterprise AI Accelerator helping Fortune 1000 companies move from AI ambition to scaled impact, faster. Powered by our AI accelerator platform – Agentic Data and Application Management (ADAM), Brillio is one of the fastest-growing digital technology service providers, delivering transformation across five core workstreams: business-led transformation, customer experience transformation, AI and data engineering, digital engineering, and infrastructure engineering.

With 14 delivery locations across North America, Europe, and Asia and a team of over 6,000 customer-obsessed professionals, Brillio combines deep industry expertise, modern engineering, and accelerators to deliver measurable outcomes. Headquartered in Dallas, Texas, Brillio serves clients globally with a commitment to speed, scale, and measurable impact.

https://www.brillio.com/
Contact Us: info@brillio.com

brillio