

Safeguards for Agentic Finance at Runtime

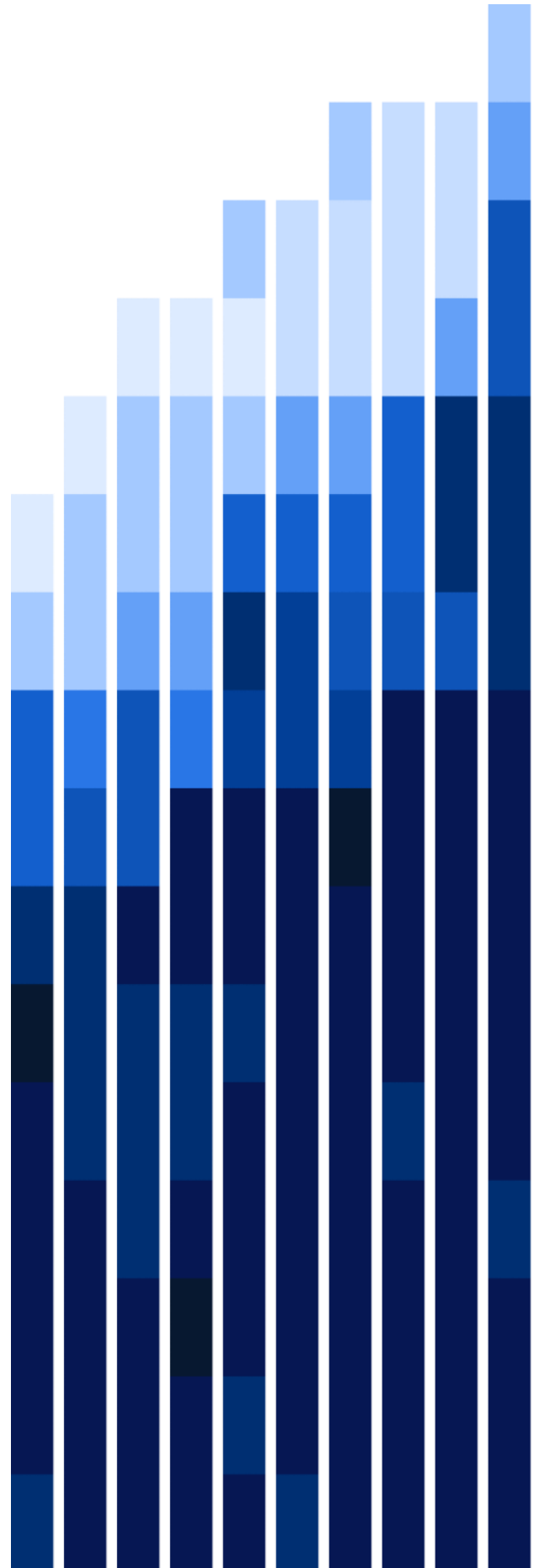
White paper

Version 1.1, October 2026

Disclaimer

This report and its contents are made available on an “as-is” basis without warranties of any kind. The content in this report does not constitute regulatory, financial, legal or any other professional advice and should not be acted on as such.

None of its authors and contributors shall be liable for any damage or loss of any kind howsoever caused as a result of the use of the information contained or referenced in this report.



SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

Document Version

Version	Date	Rationale
1.0	3 July 2026	First publication
1.1	9 October 2026	Clarify scope and implementation approach; Additional case studies

Contents

Introduction	4
Background and Motivation	5
The SAFR Framework.....	8
SAFR and Broader Ecosystem	15
How Institutions Implement SAFR.....	17
SAFR in Action: Case Studies	20
Conclusion	25
References.....	26

Introduction

Financial institutions are increasingly deploying AI agents that can initiate actions rather than merely support decisions. These systems may initiate payments, submit trading orders, approve credit applications, file regulatory reports, and settle insurance claims, in some cases at high frequency and without human review of each individual action. As these systems move from recommendation to execution, they can improve operational efficiency, reduce processing times, and support more continuous service delivery. At the same time, this shift introduces governance challenges that existing control frameworks were not designed to address. Those frameworks were developed for human decision-making and are less well suited to autonomous systems operating at machine speed and scale.

The shift is structural. Agentic systems are capable of pursuing specified objectives by planning intermediate steps, selecting tools, and initiating actions without continuous human direction. This differs in a material respect from earlier AI deployments in financial services, which were generally limited to generating outputs for subsequent human review and execution. Where a system is able to initiate consequential actions directly in live operating environments, the associated risk profile also changes¹. Risks may arise from actions taken outside the intended scope, from actions that are difficult to reverse, or from incomplete records of the basis on which actions were taken. These considerations give rise to corresponding governance requirements, including clearly defined authority, conditions of use, operational limits, accountability, and traceability.

The Financial Context Raises the Stakes

These concerns are heightened in financial services. Actions in this sector are regulated, have direct effects on customers, carry legal and economic consequences, and are often difficult to reverse. This is the case for agentic applications in payments², liquidity management, compliance triage, credit assessment, and post-transaction processing.

A further systemic concern arises where multiple institutions deploy similar models or rely on a small number of common AI service providers: behaviours may become correlated under stress conditions, a dynamic the Financial Stability Board (FSB) has identified³ as a potential source of systemic vulnerability. Layered regulatory

¹ Kasirzadeh and Gabriel (2026) characterise AI agents across four dimensions: autonomy, efficacy, goal complexity, and generality. They argue that these dimensions, individually and in combination, shape an agent's risk profile and provide a basis for calibrating governance measures to the characteristics and deployment context of different agents

² Davidovic & Tourpe (2026) discuss issues such as traceability, opacity, systemic effects, cybersecurity, and legal uncertainty in agent-mediated payment flows. Separately, Aldasoro & Desai (2025) use prompt-based experiments to assess whether a generative AI agent can support intraday liquidity management in wholesale payment settings, including payment prioritisation and liquidity-delay trade-offs.

³ The FSB identifies correlated AI model behaviour across institutions as a specific systemic vulnerability, noting that concentration in a small number of AI service providers amplifies the potential for correlated exposures and market stress (FSB, 2024), and that reliance on common AI models or infrastructure can amplify risks like herding and procyclicality (FSB, 2026).

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

obligations spanning prudential, conduct, operational resilience, AML/CFT, data protection, and consumer protection requirements compound this picture. The volume and speed of agent decisions make traditional per-action oversight operationally impossible.

What is needed is a governance layer that operates at the point of action. For each agentic system and action type, institutions need to establish what the system is authorised to do, how proposed actions are assessed at runtime before execution, and what records are retained to support review, accountability, and remediation when outcomes diverge from intent.

This paper introduces SAFR (Safeguards for Agentic Finance at Runtime) as a reference approach for that runtime governance layer. Developed through collaboration among financial industry members, SAFR supports a more consistent way of designing, governing, and implementing agentic AI systems across use cases. It does not constitute regulatory guidance or supervisory expectations, nor does it prescribe or anticipate future directions for such guidance or expectations. Each institution remains responsible for determining how its deployment aligns with applicable supervisory expectations and internal governance requirements. SAFR defines data structures, evaluation logic, and escalation contracts for agentic AI deployment. It sits between the agent and the systems it acts on, evaluating proposed actions before execution, while working alongside existing payment rails, settlement protocols, compliance engines, and core banking systems. By operating at the level of individual agent actions, SAFR's controls are designed to limit the scope of harmful actions that may correlate and propagate across institutions or amplify system-wide stress.

Background and Motivation

For most of the past decade, AI functioned as decision support in financial services. Models produced outputs in the form of recommendations, scores and flags that humans reviewed before action was taken. That boundary is dissolving. Institutions are now deploying AI agents that act autonomously, without per-decision human review, across the full range of financial operations.

Existing Efforts

The collaborative work between MAS and a consortium of leading financial institutions across banking, insurance, and capital markets through Project MindForge⁴ produced

⁴ Phase 2 of Project MindForge, launched November 2024 and published at the Singapore FinTech Festival in November 2025. The Executive Handbook (MindForge Consortium, 2025a) sets out 17 Considerations across four sections (Scope and Oversight, AI Risk Management, AI Lifecycle Management, and Enablers); the Operationalisation Handbook (MindForge Consortium, 2025b) provides implementation guidance including specific practices for agentic AI governance -- the principle of least privilege for agent tool and data access, agent certification for reusable components, accountability division at design time, kill switches and timeouts for autonomous action containment, and traceability through searchable logging.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

an AI Risk Management toolkit, at the centre of which is an operationalisation handbook that explicitly addresses agentic AI as a distinct category. It recognises that autonomous systems operating across multiple platforms introduce risk characteristics that differ from those of traditional or generative AI models.

The MindForge AI Risk Management Handbook extended that risk taxonomy into a full governance framework covering traditional AI, generative AI, and agentic architectures, including concrete practices for enterprise oversight, use case risk tiering, and agentic system containment.

MAS has since established a dedicated work stream under BuildFin.ai to develop implementation resources for agentic AI risks. This will support ongoing efforts by the industry to manage AI risks as institutional deployment of agents accelerate in financial services.

The broader governance landscape points in the same direction. Both the IMDA Model AI Governance Framework (MGF) for Agentic AI⁵ and NIST's AI Risk Management Framework⁶ point to the importance of institutions being able to demonstrate, with evidence, that autonomous AI systems operate within defined boundaries and that humans remain accountable for consequential decisions.

A growing focus on guardrails has accompanied this shift, with controls intended to shape what AI systems say or do in individual interactions. These controls can reduce certain classes of error, but they are typically probabilistic and may not, on their own, enforce the precise structural, numerical, or policy constraints that many financial actions require. In contexts such as transaction thresholds, or covenant checks, an output that is plausible but incorrect can still carry material consequences.

Technical work has begun to address the operational dimensions of this challenge. Recent work on agentic payments has specified layered authorisation requirements for autonomous financial action⁷. Complementary research has explored compliance-aware payment architectures that combine signature-based authorisation with programmable compliance controls⁸, and runtime governance approaches that shift oversight from model-level review toward bounded capabilities, telemetry, continuous authorisation, and tiered containment⁹. Taken together, this body of work points toward

⁵ Identifies four core components of agentic systems — models, memory, tools, and actions — and proposes a four-part governance model covering use-case bounding, access limitation, human oversight, and principal hierarchy accountability (IMDA, 2026).

⁶ Identifies three core characteristics of AI risk—complexity, emergence, and systemic impact—and proposes a four-part core governance model (Map, Measure, Manage, Govern) designed to cultivate a culture of trust and embed continuous risk evaluation throughout the AI lifecycle (NIST, 2023).

⁷ Proposes a three-layer model for agentic payments governance — intent and orchestration, control and authorisation, and settlement — identifying the control and authorisation layer as the critical governance point (Davidovic & Tourpe, 2026).

⁸ Implements mandate-based authorisation for agentic payments using x402 cryptographic signatures and programmable compliance constraints on stablecoin rails, demonstrating that bounded agent authority can be enforced at machine speed without degrading settlement efficiency (See & Tan, 2026).

⁹ Proposes decomposing agentic workflows into reusable 'capabilities' — bounded classes of actions with explicit authority, constraints, and evidence requirements — and extending model risk management into runtime through governance-semantic telemetry, continuous authorisation, and tiered containment (Szpruch et al., 2026).

a common conclusion: that governing agentic AI in financial services requires controls that operate at runtime, not just at the model or policy level.

The Case for SAFR Approach

What the current landscape does not yet provide is a runtime standard: a specification for how governance objectives are enforced at the moment an agent proposes to act, in production, at the speed and volume at which agents actually operate.

Despite the pace of agent deployment, the governance infrastructure available to financial institutions has not always kept up. Three gaps are particularly consequential.

The first is the absence of *pre-execution assurance*. Model risk management frameworks are calibrated to pre-deployment validation. They assess the model before it goes live, not the decisions it makes in production. Audit processes are retrospective, reviewing sampled transactions hours or days after execution. Neither catches a problematic agent decision before it executes. By the time an issue appears in the audit log, the action has already occurred and its consequences are already in motion.

The second gap is the absence of a *human-agent governance interface*. There are no consistent standards for defining when a human should be in the loop. The mechanism is typically ad hoc: a notification, an email alert, or a dashboard flag, with no defined deadline for response, no standard format for what needs to be decided, and no audit record of the decision made. This gives the appearance of human oversight without the substance of it.

The third gap is *fragmentation*. Where institutions have built agentic guardrails, they have done so independently for each deployment. The controls are not interoperable, not auditable in a consistent format.

A misconfigured data feed, a corrupted instruction, or an adversarially crafted input can trigger a cascade of autonomous actions across multiple agents that is detected only after the harm has occurred. At the volume and speed at which agents operate, the window for human intervention without a structured, pre-execution governance mechanism is small.

These gaps reflect the fact that existing governance infrastructure was designed for a different kind of AI system. SAFR is proposed as an industry reference approach to fill them as a runtime governance layer that enforces authority, calibrates human involvement, and produces a complete audit record at the point of every consequential agent decision. It provides a shared structural base — a common governance framework, a defined set of runtime components, a consistent set of disposition outcomes, and a standard audit record format — that enable interoperability across deployments while permitting each institution to configure its own controls.

The SAFR Framework

SAFR introduces a governance checkpoint between material agent decision and its execution. It consists of four runtime components interacting through the Governance Envelope as a single, structured record of what the agent proposes to do, the action trace supporting the proposal, and the evidence needed to assess it:

- **Agent Identity** – Binds each proposed action to a recognised, registered agent, verified against that agent's registry entry before any other evaluation proceeds.
- **Controls Repository** – The institution's configurable rulebook: the controls — drawn from sources such as organisational policies, regulatory requirements, product rules, and user-provided mandates — against which a proposed action is checked.
- **Disposition Engine** – Evaluates each in-scope action against the controls retrieved from the Controls Repository and resolves it to one of the possible outcomes.
- **Audit Log** – A tamper-evident, append-only record of every governance decision.

Together, these ensure that no agentic action reaches execution without having been declared, authorised, and assessed. SAFR is an industry reference model where each mechanism defines a condition that should hold for every consequential agentic action, leaving the form of implementation to the deployment context.

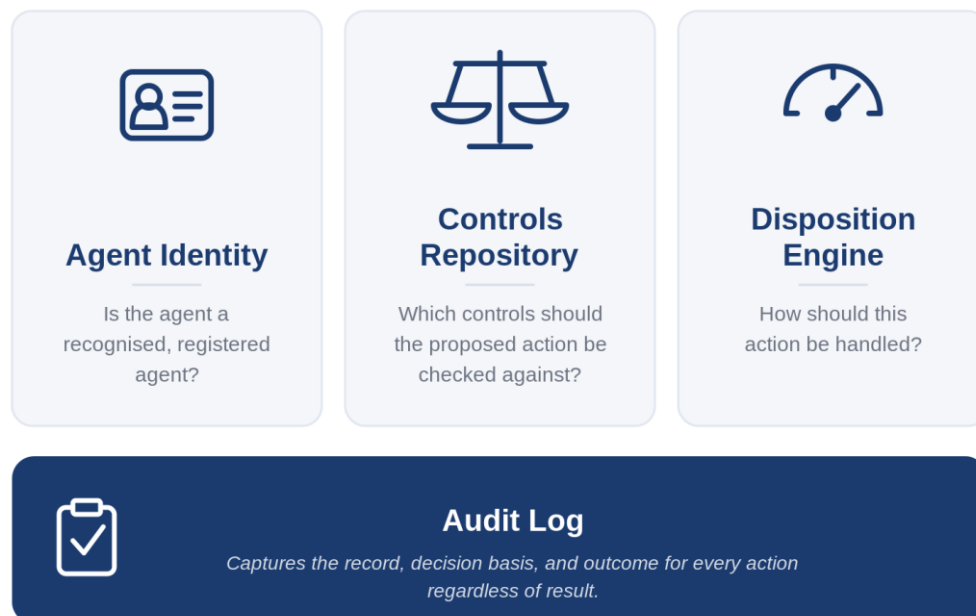


Figure 1. Components of SAFR.

The Governance Envelope

Before any in-scope agentic action can be executed, it should be packaged with the context needed to assess it. This package is the governance envelope. The envelope captures three classes of information:

- Action — the action type, and the scope and parameters of the proposed action.
- Action trace — the observable steps the agent executed in arriving at the proposed action, for example, the tool calls made, data retrieved, checks performed, and evidence relied on.
- Context metadata — the agent identity, applicable mandate, current account or system state, and operative policy constraints.

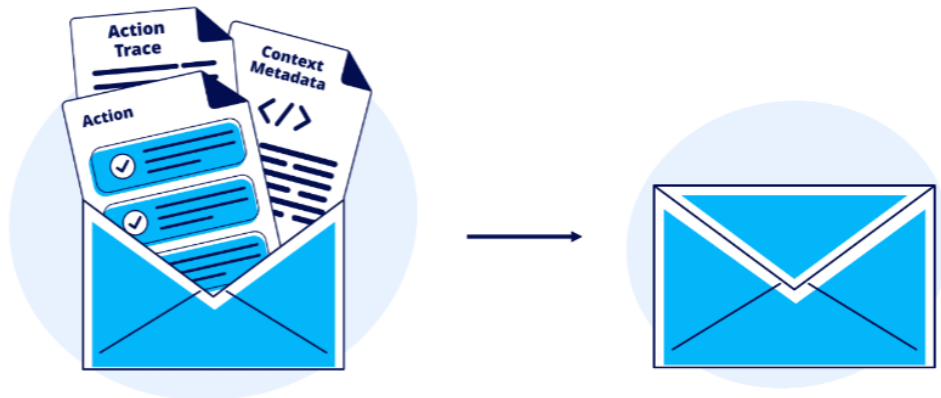


Figure 2: Contents of the Governance Envelope

Capturing the action trace makes the envelope a governance artefact: the observable basis for the proposed action is preserved at the point of proposal, available for validation and subsequent review without reconstructing or relying on the agent's internal reasoning.

The governance envelope should not be treated as self-verifying. Where the action details and action trace are supplied by the agent, consistency between them does not establish that they accurately reflect the action proposed or the steps actually performed. Where feasible, these fields should therefore be checked against records captured independently of the agent, such as the initiating instruction, the actual action payload, and system-generated records of tool calls and data access. The envelope should also be bound to the verified agent, session and applicable mandate, and protected against subsequent alteration.

The governance envelope is intended to provide a complete and coherent representation of the proposed action at the point it is submitted for governance. It should contain the fields needed to identify the agent, describe the action, capture the action trace, and supply the relevant context for assessment. The action trace should document how the agent arrived at the proposal, including the steps taken, tools used, data retrieved, and checks performed, and evidence relied on. The proposed action should be understandable from that record, so that the envelope functions as a faithful

declaration of what the agent intends to do and the basis on which it reached that proposal.

The Agent Identity

For governance to be meaningful, the agent proposing an action must be who it claims to be. An outcome produced by the Disposition Engine is only as reliable as the identity that was presented to it. If the agent's identity cannot be verified, the Controls Repository cannot be correctly consulted, and the Audit Log cannot capture an accountable record. Agent Identity verification is therefore the condition that allows the Controls Repository to retrieve the applicable controls, as the identity of the agent shapes what it may be permitted to do.

At runtime, the Agent Identity component resolves and confirms the identity of the agent behind each proposed action. When a Governance Envelope is received, the component determines which registered agent identity corresponds to the claimed agent, retrieves the relevant registry entry, and verifies that the identity matches. An envelope that fails this check results in the proposed action being rejected at this stage.

The nature of this resolution depends on the network context in which the agent operates. In closed-loop environments — where the agent acts entirely within a single institution's infrastructure — the relevant registry may be the institution's own, and identity resolution is a direct lookup against it. In open networks, where an agent may interact with counterparties, payment systems, or other institutions outside the deploying organisation, the context is more complex. An agent may be registered with multiple identity databases (e.g., the deploying institution's internal registry, a payment network's agent registry, or an inter-institutional directory). In these settings, the Agent Identity component must also determine which registry is authoritative for the context of the proposed action and retrieve the appropriate identifier accordingly. The output in either case is the same: a confirmed identity binding that can be carried forward to the Controls Repository, or a rejection recorded in the Audit Log.

Agent Identity should not be understood as a mechanism for trusting the agent's reasoning or output. SAFR assumes that an agent's proposed action still requires independent governance assessment. The purpose of Agent Identity is to bind the proposed action to a recognised agent, determine which controls apply to that agent and action type, verify whether the agent is authorised to invoke the relevant capability, and preserve an accountable audit record.

The Controls Repository

The Controls Repository holds the controls against which proposed actions are checked: a configurable rulebook drawn from sources such as organisational policies, regulatory requirements, product rules, and user-provided authority. Controls take different forms depending on their source (e.g., regulatory requirements impose

categorical constraints that apply broadly across action types; product rules define operational boundaries for specific classes of action; and user-provided authority captures the scope that a user has delegated to a specific agent).

Controls may be broadly categorised by their nature. Generic controls (e.g., authorisation checks, exposure limits) apply to any actor and are typically deterministic as they evaluate against fixed thresholds or categorical rules. AI-specific controls (e.g., evidence quality, envelope integrity checks) apply specifically to agentic actions and may involve probabilistic or semantic assessment.

A mandate is the mechanism through which delegated authority is scoped and expressed in machine-readable form. Where a principal authorises an agent to act on its behalf, the mandate records that delegation explicitly, capturing what the agent may do, within what limits, and under what conditions¹⁰. This draws on a well-established paradigm in access control: capability-based security¹¹, in which authority is held in a discrete, bounded token that confers specific rights on its bearer without requiring consultation of a global permissions registry. An agent cannot extend the scope of a mandate through its own reasoning or inference. Authority is explicit, structured, and defined by the mandate, not inferred by the agent.

Controls encode the parameters of an agent's permitted authority, such as:

- Permitted action types — the action types the control governs.
- Decision logic — the rules and thresholds to be evaluated against.
- Escalation conditions — the circumstances under which review is required before execution.
- Validity period — the timeframe within which the control is operative.
- Principal authority — the hierarchy from which the control derives its force, and the authority retained to update or revoke it.

The Disposition Engine

The Disposition Engine evaluates the proposed action deterministically against the controls retrieved from the Controls Repository. Not all in-scope actions warrant the same treatment: an instruction to retrieve a document carries materially different consequences from an instruction to execute a payment or suppress a compliance alert. The Disposition Engine makes this differentiation explicit and systematic, producing a defined, binding outcome for every proposed consequential action

¹⁰ A prominent example is the Agent Payments Protocol (AP2), which uses cryptographically signed "mandates" to establish an AI agent's spending authority. These mandates capture user instructions and operational constraints, providing counterparties with verifiable, non-repudiable proof that a machine-to-machine transaction is fully authorized and within scope.

¹¹ The capability-based security model was formalised by Dennis and Van Horn (1966), where a capability identifies an object and the permitted operations over it, and remains the foundation for contemporary authorisation frameworks. The core principle, that authority should be expressed in an unforgeable token or reference that carries authority directly rather than resolved against a global permissions registry at runtime, underlies modern access delegation standards including OAuth 2.0 (Hardt, 2012).

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

calibrated to the specific risk it presents, whether realised as a fixed threshold lookup or a multi-factor risk assessment.

It produces one of four outcomes for every action:

- Deny — the proposed action violates a hard regulatory or policy constraint, or presents a risk profile above defined thresholds. The action is rejected before execution, with a specific reason recorded.
- Escalate — the proposed action is within scope and below hard constraints, but above the threshold for autonomous execution. The action is held pending human review before proceeding.
- Auto-Execute — the proposed action is within scope, below hard constraints, and within defined risk thresholds. The action proceeds without requiring human intervention.
- Observe — the proposed action is permitted to proceed but is flagged for monitoring; the action executes while a structured observation is logged for subsequent review. This outcome is used where the action does not meet the threshold for escalation but presents a pattern or signal the institution has configured as warranting attention.

All four outcomes feed into the Audit Log, which operates independently of which outcome was reached. Outcome assignment is calibrated to the risk profile of the proposed action. Factors that may drive the calibration:

- Action reversibility — whether the action can be undone if it produces an unintended result.
- Financial materiality — the monetary value and potential loss exposure of the action.
- Customer impact severity — the potential effect on customers if the action is erroneous or unauthorised.
- Regulatory sensitivity — whether the action type is subject to specific regulatory obligations or heightened scrutiny.
- Novelty or anomaly — whether the action departs from established patterns within the mandate.

Higher-risk profiles shift outcomes toward Deny or Escalate. Lower-risk profiles receive Auto-Execute or Observe. The calibration is set at design time through the governance parameters of the controls, ensuring that threshold-setting is subject to appropriate review before deployment.

The Audit Log

Every governance decision produces an entry in the Audit Log, which serves as an immutable, tamper-evident record that captures the complete history of every in-scope action an agent takes, and every decision made about it. Once written at the point of decision, an entry cannot be modified by the agent or by any downstream system

without detection, supported by appropriate integrity and confidentiality controls¹². The governance history is preserved exactly as it occurred.

Each log entry should capture the essential elements of a governance decision: the governance envelope as submitted, the mandate against which the action was checked, the outcome produced by the Disposition Engine, the specific rules applied, the basis for that outcome, and the time elapsed at each stage, subject to appropriate data minimisation, access controls, and retention requirements. Beyond these essentials, institutions have flexibility in how they structure and extend their log to fit their own compliance infrastructure and operational context.

The Audit Log serves two distinct purposes. Operationally, it provides the structured governance record that existing compliance platforms, AML/KYC systems, transaction monitoring, and sanctions screening can draw on in order to assess agent-initiated actions. For accountability, it is designed to ensure that every action an agent takes, and every decision to allow, deny, escalate, or observe that action, can be fully reconstructed and reviewed without relying on the agent's own account of what occurred. The log is the authoritative record, independent of any party with an interest in how events are characterised after the fact.

Action Lifecycle Across the Four Components

Figure 3 maps the sequential control flow. A proposed action is packaged in a Governance Envelope and validated for completeness and coherence. Agent Identity is then verified against the relevant agent registry. An envelope that fails this check is rejected immediately and the failure logged. Envelopes that pass are then verified against the Controls Repository to identify which controls need to be checked. The Disposition Engine then determines the appropriate thresholds to test against, and returns one of four outcomes. The Audit Log records the full governance trace for every outcome.

In multi-step agentic workflows, each consequential action should be evaluated independently. An Auto-Execute or Observe outcome at one step carries no authority into the next (i.e., prior authorisation should not carry forward as the agent adapts to intermediate results and changing conditions).

¹² Types of integrity and confidentiality technical controls include cryptographic hashing, digital signatures, append-only storage, timestamping, encryption, key management, and permissioned access.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME



Figure 3. Process flow through SAFR.

SAFR and Broader Ecosystem

SAFR provides the runtime mechanism to operationalise governance of agentic AI at the point of an agent's decision.

Where SAFR Sits in the Stack

To understand where SAFR fits, it helps to trace how an agent-initiated financial action travels from decision to settlement, and what role each layer plays in that journey. No layer is a substitute for any other. SAFR operates after model-level guardrails and before the proposed action is executed, delivered, or otherwise relied upon in a financial-services workflow. The following walks each layer in sequence, from model output to downstream execution or delivery.

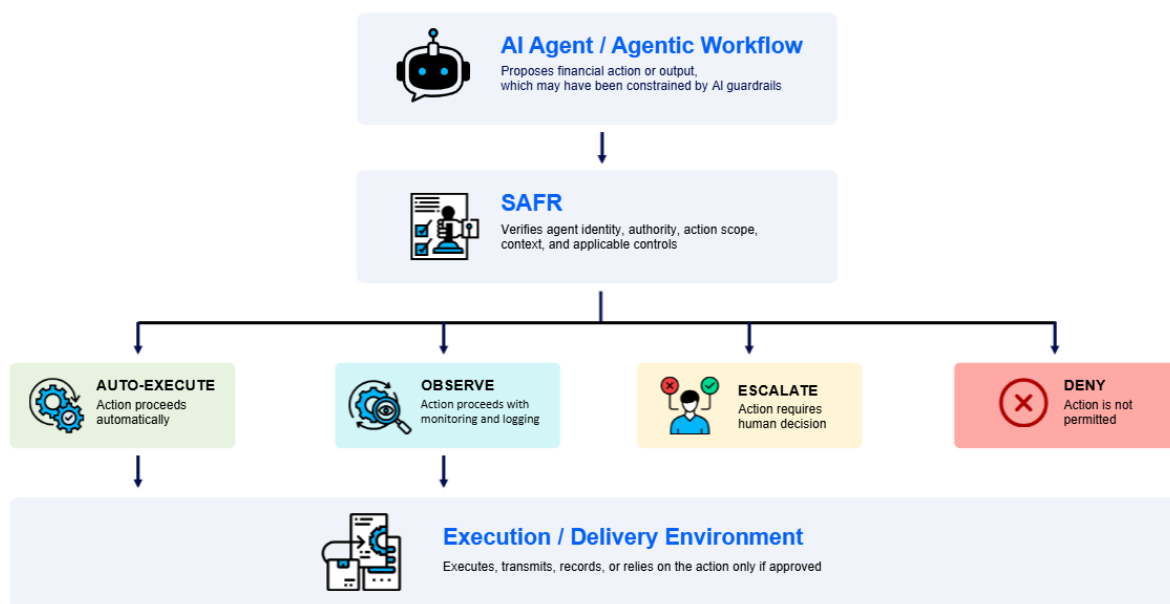


Figure 4. SAFR within the financial technology stack.

At the top of the stack, the AI agent proposes a financial action or output. What the agent can propose may already be constrained by controls built into the application and the agent itself, including capability restrictions, content filtering, prompt defences, and other guardrails that limit available actions and screen out harmful, malformed, or policy-violating outputs before they leave the model. These controls are important, but they do not by themselves determine whether a proposed financial action should proceed. Many financial actions require additional evaluation against delegated authority, institutional policies, regulatory requirements, client-specific constraints, or supporting evidence. An action may fall within an agent's permitted capabilities yet still require assessment as to whether it is authorised, appropriate, and sufficiently justified in its particular context. Runtime governance therefore provides an additional layer of evaluation for consequential financial actions before execution.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

The proposed action is then passed through SAFR components. SAFR sits between the agent and the execution environment, taking the proposed action and evaluating it deterministically against the relevant controls to decide whether it can be executed by the agent, must be escalated for human decision, or must be rejected. This is the point at which the institution determines, before execution, whether the agent is permitted to do what it proposes to do and under what conditions it may proceed.

The downstream layer depends on the nature of the financial outcome. Where value moves, settlement layer enforcement, network rules, and scheme compliance come into effect. At that point, the action is governed by the rules of the relevant rail, network, or scheme, together with any controls embedded in the settlement layer itself. Card network rules, SWIFT message standards, ACH operating rules, machine-to-machine execution protocols like the Machine Payments Protocol (MPP)¹³, and programmable settlement conditions such as Purpose Bound Money (PBM)¹⁴ or stablecoin controls determine whether and how the transaction is processed once submitted. However, not all consequential financial outputs involve payment rails. Where the agent produces an investment recommendation, credit assessment, compliance disposition, insurance claims decision, regulatory filing, or customer communication, the relevant downstream environment may instead be an advisory platform, trading system, credit engine, compliance case-management system, claims platform, regulatory-reporting system, or client communication channel. These downstream controls remain essential, but they operate after SAFR has determined whether the agent may take, transmit, or rely on the proposed action in the first place.

What SAFR is not

To avoid conflating SAFR with adjacent initiatives, it helps to be explicit about scope boundaries.

- Payment schemes and settlement rails move money; SAFR governs the decision to move it. Systems such as SWIFT, card networks, and multilateral payment platforms handle transmission, while SAFR performs the pre-execution evaluation that comes first.
- Existing compliance platforms serve a different function. SAFR may incorporate checks such as sanctions screening and AML triggers, but its purpose is to generate the structured governance record needed to assess agent-initiated actions, not to replicate the compliance assessments those platforms already perform.

¹³ MPP is an open standard, co-developed by Stripe and Tempo, that allows agents and services to coordinate fiat or stablecoin payments programmatically.

¹⁴ PBM is an approach that allows conditions to be programmatically added to the transfer of tokenised assets (Monetary Authority of Singapore, 2023). Programmable conditions may be extended to regulatory compliance requirements (Pham et al., 2026).

- The same distinction applies to LLM safety tools. Content moderation, prompt-injection defences, and output filtering operate at the model's input/output layer, governing what the model produces. SAFR operates at the execution layer, governing whether the proposed action is authorised and executable before it proceeds downstream. These layers are complementary and not interchangeable.
- Nor is SAFR a managed service. It serves as an industry reference for institutions to implement within their own infrastructure, using their own rule configurations and governance arrangements.

How Institutions Implement SAFR

Institutions adopt SAFR by integrating its components into their existing technology and governance infrastructure. The degree of change required depends on where an institution is starting from.

SAFR does not require implementation as a single central platform within an institution. It is a reference architecture for runtime governance, and its components may be implemented centrally, embedded within individual agent deployments, exposed as shared institutional services, or combined through a hybrid model. What matters is not the topology of implementation, but that each in-scope agentic action is subject to the required SAFR functions before execution or downstream reliance.

For some institutions, a central SAFR service may provide consistent policy management, common audit records, and reusable escalation workflows across multiple agents. For others, especially where agents operate in different business lines, technology stacks, or latency-sensitive environments, SAFR functions may be implemented at the agent or workflow level, provided they conform to common institutional standards for governance envelopes, disposition outcomes, control semantics, and auditability.

Integration patterns

SAFR supports two primary deployment patterns. The choice depends on whether the institution is governing new agent capabilities or bringing existing deployments under governance.

1. Native Integration

The agent is instrumented to emit a Governance Envelope before each proposed action. The SAFR validator evaluates the envelopes against the institution's controls repository and returns an outcome decision before the agent takes any action. This pattern is recommended for new agent deployment. It provides the tightest integration, the most granular governance record and the cleanest audit trail. Any agent built on a modern API or developed in-house can be instrumented with minimal engineering efforts.

2. Gateway Integration

A SAFR gateway intercepts outbound API calls at the infrastructure layer, wrapping each call in a Governance Envelope and evaluating it without any changes to the agent code itself. This pattern is designed for legacy systems, third party agents or existing deployment where modifying agent code is not feasible. It allows an institution to bring its existing agents under governance without re-engineering individual systems or blocking development teams. For institutions with many existing agents, the gateway pattern serves as a practical starting point by establishing coverage first and native instrumentation can follow for new builds.

Controls Configurations

The Controls Repository has several categories of controls, each independently configurable. Configuration of the controls should be owned by the institution's risk and compliance leadership. Table 1 describes some of the applicable categories of controls.

Table 1: Control categories

Control category	What the institution decides
Authorisation	Which agents are permitted to act for which principals. Who can delegate authority to whom, and to what depth. Which action types each agent class is permitted to initiate.
Exposure Limits	Per-action and aggregate value threshold. Below the threshold, autonomous actions are permitted. Above it, human review is required or the action is blocked. These mirror existing delegated authority frameworks and should be set to match them.
Rate Limits	Maximum action rate per time window. Protects against runaway agents, data feed errors, and adversarial prompt injections that cause agents to act at abnormal speed or volume.
Evidence Quality	Minimum confidence threshold and required evidence for autonomous execution. Actions where the agent's stated confidence falls below the threshold are routed to human review regardless of value.

Activating Human Reviewer Escalation

When defining the controls to determine the thresholds required for escalation to a human reviewer, institutions must assess three operational dimensions to ensure that escalation is substantive rather than nominal.

- **Escalation volume.** Institutions should assess the volume of actions it can manually assess. An escalation function that generates more reviews than the institution can meaningfully process defeats its own purpose.
- **Review turnaround.** Escalations to a human reviewer should include a timeout window. If no decision is made within that period, the action should default to block or be escalated to a senior reviewer. The window should reflect realistic reviewer availability, including overnight and weekend coverage for time-sensitive domains.
- **Reviewer authority.** Reviewers must have clear authority to approve, modify, or decline escalated actions. Their decisions should carry the same institutional weight as the original agent decision. Escalating to someone without authority creates the appearance of oversight without making it effective.

SAFR in Action: Case Studies

This section illustrates how SAFR operates across different financial domains through industry case studies.

Treasury and Payments

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
Ant International <i>Agentic Treasury Protocol</i> AI agents autonomously execute treasury functions within policy limits set by a human principal	Each agent holds a Digital Agent Passport that identifies it, links it to a named human principal, and records its permitted capabilities. The level of verification required scales with the size and risk of the transaction.	The human principal authors a mandate specifying what the agent may do, up to what financial limits, and under what conditions. These rules are encoded digitally as signed, machine-readable policy rather than held as manual procedures, and remain operative for a defined period.	Policy rules are checked before any payment or trade reaches settlement. Where limits are exceeded, the action is blocked. Circuit breakers can halt agent activity at the agent, principal, or counterparty level. Where instructions are ambiguous, the agent defaults to inaction.	Every transaction produces a tamper-evident record carrying the authorising mandate, the rules applied, and the outcome reached.
Mastercard <i>Agent Pay</i> A system enabling registered AI agents to transact on behalf of consumers, with agent identity bound to the payment token	AI agents must undergo a rigorous registration and identity verification process analogous to KYC. Each registered agent is assigned an Agentic Token: a unique digital identity credential cryptographically bound to the agent's operator and the end-user it represents. Only registered, vetted agents may operate on the Mastercard network.	Each Agentic Token is bound to the scope of authority the consumer has granted (e.g., specific merchant or category, maximum amount, and time frame) under a principle of least privilege. The consumer's consent is required for each use of the agent's authority, and the agent cannot exceed the limits encoded in its token.	At the point of transaction, the agent presents its full scope of intent for validation. The transaction then flows through Mastercard's standard network authorisation and fraud screening, augmented by the Agentic Token's identity and intent data — visible to both the network and the issuer — enabling real-time, risk-informed approval.	The system creates a tamper-resistant record of the consumer's authorisation when an AI agent acts on their behalf, aligned with Mastercard's co-developed Verifiable Intent framework. This shared record supports dispute resolution and audit across the ecosystem, while preserving existing consumer protections such as chargeback rights.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
Visa <i>Visa Intelligent Commerce (VIC)</i> AI agents help consumers and businesses discover, initiate, and complete transactions securely, while retaining the trust, controls, and protections of traditional Visa payments.	Agents are formally registered in the Visa ecosystem, with business and legal checks applied at onboarding. Agent-initiated transactions are identified at the network level through Token Requester Identifiers, enabling issuers to recognise a registered agent acting on behalf of a consumer and apply risk-based approval decisions accordingly.	The consumer's instruction is confirmed via Passkey — a FIDO-standard biometric — and converted into a digital rule stored on the Visa network specifying permitted amount, merchant scope, and validity period. The agent cannot modify these limits; any attempt to exceed the authorised amount is declined at the network level before the transaction is processed.	At payment, Visa checks the transaction in real time against the stored instruction. Transactions within the mandate (including amount and date) proceed automatically; those outside it are declined without human intervention.	All key moments are recorded by design, from instruction capture and mandate validation through to disposition outcome and settlement. Post-transaction, the agent submits merchant acknowledgements to Visa as Signals, providing a structured record that supports dispute resolution across the ecosystem.
Circle An AI agent autonomously discovers and pays for API services on a per-use basis, covering fraud scoring, FX data, and KYC checks, holding and spending digital currency through an Agent Wallet.	Each agent holds a portable, verifiable identity registered on-chain under the ERC-8004 standard and bound to its Agent Wallet. The agent's identity is resolvable by the institution's own registry or, in open-network transactions, through the Circle Payments Network, making the acting agent traceable across both closed and open settlement environments.	Agent Wallet policy sets per-transaction and aggregate spending caps and defines the permitted recipients and contracts at build time, so the agent acts only within pre-authorised bounds. Each call carries a signed per-call authorisation. Separately, institutional compliance controls (e.g., Circle's Compliance Engine) screens transactions	Before each action executes, the Agent Wallet's policy layer checks each action against its caps and permitted recipients, and compliance screening (e.g. Circle's Compliance Engine) conducts checks against sanctions and other applicable requirements. In-policy, approved-list actions proceed automatically; flagged patterns are monitored; those above value	Each action produces an append-only record of the intent submitted, the rules applied, and the outcome reached. Circle's Compliance Engine and identity validation records feed into this trail, reconciled against on-chain settlement. Legal terms and recourse are captured via the Legal Context Protocol, providing a complete governance trail for each autonomous transaction.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
		against sanctions and other applicable requirements.	or novelty thresholds are escalated; and sanctions hits, blocklisted addresses, expired mandates, or cap breaches are declined.	

Wealth Management and Advisory

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
OCBC <i>Source of Wealth Assistant (SOWA)</i> ¹⁵ An AI agent parses client financial documents and drafts a source of wealth memo to support compliance review	Agents are narrowly defined to perform specific task in the document analysis and memo drafting workflow. Agents are identified to ensure that the acting agent is bound to the actions that it is permitted to perform.	Agents perform narrowly scoped tasks (extraction, drafting, checking), not end-to-end decisions. Agents can only act when they have been authorised to do so based on pre-defined workflow triggers.	There is human review at critical decision points, including after the income and financial documents have been assessed to ensure plausibility. Ultimately, outputs are advisory and subject to human validation.	Each assessment produces a standardised, structured memo capturing the documents reviewed and conclusions drawn against internal guidelines, which serves as the institution's record of the source of wealth assessment for compliance review.

Corporate Banking

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
An internal AI agent prepares market and client intelligence briefs for senior	Each agent is uniquely identified within the institution's systems. Agents performing critical	The agent's permitted scope is defined by predefined constraints — limited to	The workflow distinguishes between tasks that the agent may execute autonomously and	Agent actions are subject to enterprise-level monitoring, with clear visibility and traceability of each

¹⁵ Developed together with Bank of Singapore, a wholly owned private banking subsidiary of OCBC.

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
bankers ahead of client meetings	tasks autonomously are assigned a named, accountable human owner with clear responsibility for the agent's output.	accessing designated news sources — and the principle that impactful and material actions require human approval.	material actions (e.g., delivering the brief), which requires explicit approval from a human.	agent's actions, enabling the institution to trace what was done back to a named, accountable human owner.

Insurance

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
Manulife A GenAI-powered sales enablement tool to support preparation, insight generation, and customer engagement	Each request records the adviser identity, linking every agent action to a known, identified human principal.	Only approved enterprise content can be used by the agent for retrieval and synthesis. Outputs are validated through content filtering, retrieval controls, and LLM-as-a-Judge evaluation against answers curated by subject matter experts.	Once proposed agent actions are evaluated, out-of-scope or low-confidence responses are blocked or escalated. System provides no autonomous execution pathway into financial systems.	Material events across each interaction — request, retrieval, generation, evaluation outcome, and adviser usage — are systematically captured in a structured audit trail.

Risk and Compliance

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
AI agents investigate flagged transactions and propose whether a suspicious transaction	Each investigation agent is registered in the institution's agent registry and assigned to a named compliance owner. Its registry entry records the business lines and alert types it may	Authorisation controls set which agents may investigate alerts in each business line. Investigation controls, defined by operational risk and compliance, set the mandatory	The proposed disposition is checked against these controls. Its action trace is cross-checked against system-generated records of the tool calls and data access	Each case produces a tamper-evident record. It covers the originating alert, the action trace, the controls applied, the outcome and its basis, and any

SAFEGUARDS FOR AGENTIC FINANCE AT RUNTIME

	Agent Identity	Controls Repository	Disposition Engine	Audit Log
report (STR) should be filed	investigate. Each proposed disposition is bound to the verified agent, so the applicable controls can be retrieved.	steps for each alert type and the minimum evidence needed for each conclusion. Mandatory steps include customer due diligence review, sanctions and adverse media screening, and review of related accounts and prior alerts. Tipping-off restrictions bar any customer-facing action while a case is open.	the agent actually performed. Dispositions from agents not authorised for the business line, or with missing steps or insufficient evidence, are denied and returned to the queue. Proposals to file an STR are escalated to a compliance officer, who approves the filing. Ambiguous, high-risk or low-confidence cases are escalated for additional human review. Low-risk cases with complete evidence are closed as false flags and monitored through quality-assurance sampling.	human decision. Decisions not to file are kept alongside filed STRs to support internal audit, model validation and supervisory review. Access is restricted because STR information is confidential.

Conclusion

Autonomous AI agents are already being deployed across financial services. For that deployment to scale responsibly, governance infrastructure must keep pace so that risks are managed effectively and boards and senior management retain meaningful oversight.

SAFR is a practical industry reference approach for what runtime governance of agentic AI could look like. It provides a structured way to define what an agent is authorised to do, assess proposed actions before execution, and produce a consistent record for accountability and review. At this stage, it is intended as a starting point to support industry discussion and give institutions a foundation to build on. It should evolve through industry input and implementation experience.

Any specification of this kind depends on practitioner knowledge. Institutional experience of how agents behave in production, where governance gaps emerge, and which controls are operationally realistic will determine how the framework develops in practice.

FinTechs and financial institutions are invited to contribute to SAFR and the BuildFin.ai working group by sharing pilot findings, identifying gaps in the specification, and raising domain requirements not yet addressed. The aim is a standard that is technically rigorous, operationally grounded, and practically useful for financial institutions seeking to govern agentic AI in their use cases.

References

- Aldasoro, I., & Desai, A. (2025). *AI agents for cash management in payment systems* (BIS Working Papers No. 1310). Bank for International Settlements.
- Davidovic, S., & Tourpe, H. (2026). *How agentic AI will reshape payments* (IMF Note 2026/004). International Monetary Fund.
- Dennis, J. B., & Van Horn, E. C. (1966). *Programming semantics for multiprogrammed computations*. Communications of the ACM, 9(3), 143–155.
- Financial Stability Board. (2024). *The financial stability implications of artificial intelligence*. FSB.
- Financial Stability Board. (2026) *Sound practices for responsible adoption of artificial intelligence (AI): Consultation report*. FSB.
- Hardt, D. (Ed.). (2012). *The OAuth 2.0 authorization framework* (RFC 6749). Internet Engineering Task Force.
- Infocomm Media Development Authority. (2026). *Model AI Governance Framework for Agentic AI* (Version 1.5). IMDA.
- Kasirzadeh, A., & Gabriel, I. (2026). *Agentic profiles for effective AI governance*. *Nature*, 656, 320–328.
- MindForge Consortium. (2025a). *AI Risk Management: Executive Handbook*. Monetary Authority of Singapore.
- MindForge Consortium. (2025b). *AI Risk Management: Operationalisation Handbook*. Monetary Authority of Singapore.
- Monetary Authority of Singapore (2023). *Purpose Bound Money (PBM) Technical Whitepaper*. MAS.
- Monetary Authority of Singapore. (2025). *Guidelines on artificial intelligence risk management*. MAS.
- National Institute of Standards and Technology. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). U.S. Department of Commerce.
- Pham, K. U., Davidovic, S., Toh, W., See, K., Chang, C., & Vicary, S. (2026). *Programmable compliance: Compliance architecture for regulated tokenised assets*. Global Layer One.
- See, K., & Tan, X. W. (2026). *Compliance-aware agentic payments on stablecoin rails*. arXiv:2605.00071.
- Szpruch, L., Sudjianto, A., Bhatti, T., & Ang, G. (2026). *Scalable runtime governance for agentic AI in financial services*. Working paper.

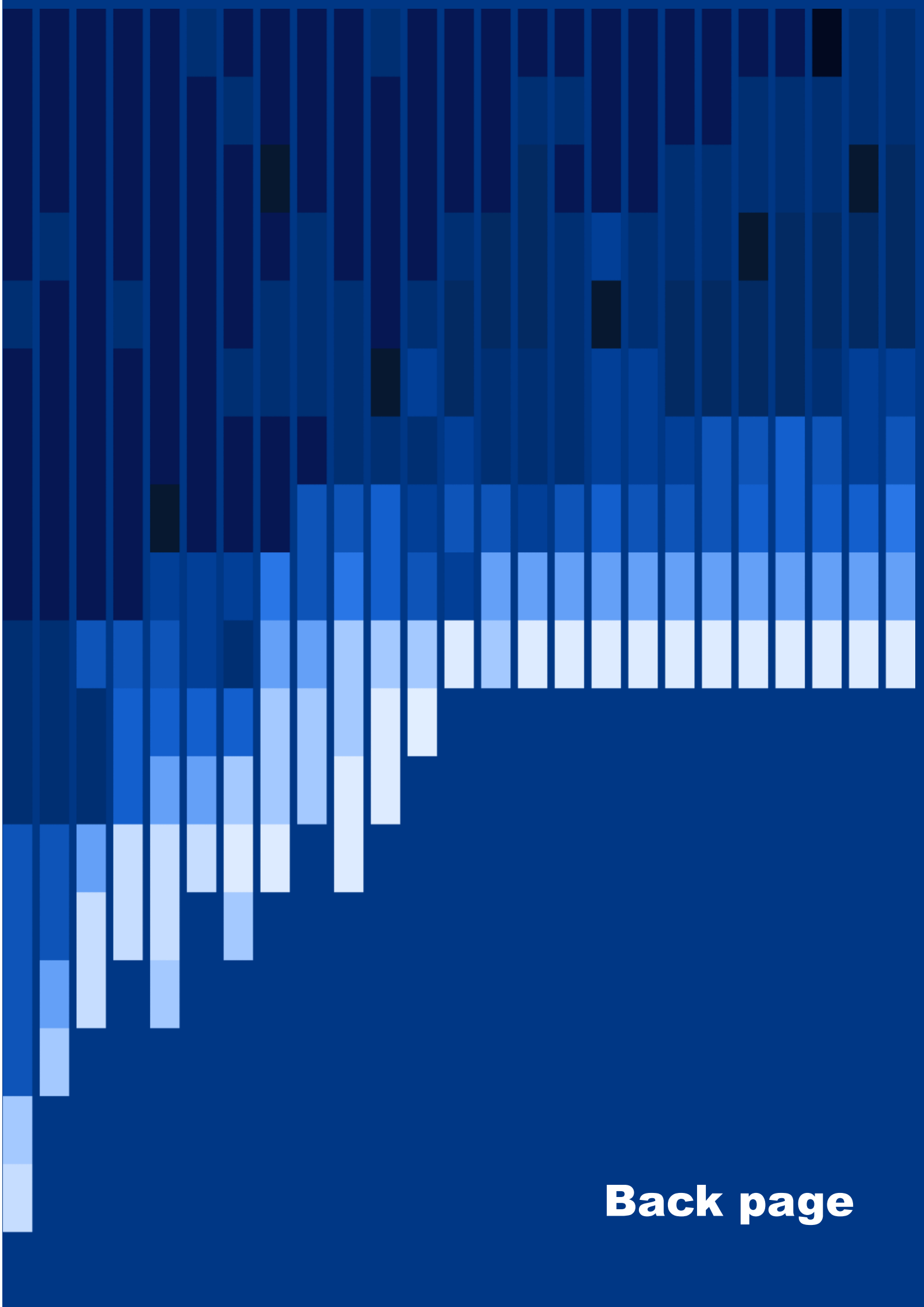
Acknowledgement

This paper is written with contributions from

- Ant International
- Circle
- DBS
- Google
- HSBC
- J.P. Morgan Chase
- Manulife
- Mastercard
- OCBC
- UOB
- Visa

With special thanks to

- Gary Ang, *Quantitative*
- Maxim Afanasyev, *National University of Singapore*

An abstract geometric pattern composed of vertical bars of varying heights and widths. The bars are colored in shades of blue and white, creating a sense of depth and movement. The pattern is most dense on the left side and fades towards the right.

Back page