

Bank-Grade Agentic AI: A Control and Assurance Framework for Financial Services

Abstract

This white paper presents the joint work of mCards, Radware, and SECAI to design, secure, adversarially test, and evaluate an AI-enabled financial services environment against established banking, cybersecurity, privacy, third-party risk, and AI governance frameworks. The initiative demonstrates a practical approach for deploying agentic AI with defined authority, enforceable controls, independent validation, remediation, and continuous assurance.

The result is both a scalable solution architecture and a repeatable operating model for banks and other regulated entities seeking to capture the efficiency and operating leverage of agentic AI without sacrificing control. By enabling greater automation across service, operations, workflows, and decision support, this approach can support material improvements in productivity, cost efficiency, scalability, and enterprise economics, including EBITDA performance.

This paper explains how that capability can be governed, tested, challenged, remediated, and maintained within a regulated operating environment. Its central premise is that financial institutions can pursue the economic benefits of agentic AI at scale when authority is bounded, controls are independently enforced, and the complete system can be demonstrated to operate within defined risk parameters.

Dated: 11 August 2026

Table of Contents

Executive Summary

1. Why Agentic AI Changes the Banking Control Problem
2. Governance, Accountability, and Human Authority
3. Agent Authority and Guardrail Design
4. System Architecture and Control Implementation
5. Third Party Risk, Regulatory Alignment, and Due Diligence
6. Adversarial Validation Methodology
7. Findings, Remediation, and Retest
 - Authority and Tool Control
 - Prompt and Instruction Manipulation
 - Identity and Data Isolation
 - Information Disclosure and Integrity
 - Control Interaction and Remediation
8. Continuous Assurance, Resilience, and Change
9. The Bank-Grade AI Assurance Framework
10. Conclusion: Demonstrable Control

FINAL - EYES ONLY

Executive Summary

Artificial intelligence is moving from generating information to taking action. In financial services, that transition materially changes the control problem. A conversational system that explains a transaction presents one class of risk. An agentic system that can retrieve customer information, invoke tools, interact with financial APIs, initiate workflows, or influence financial actions introduces a broader set of risks involving identity, authorization, information integrity, privacy, operational resilience, third-party dependencies, and human accountability. As AI becomes connected to the systems through which financial services are delivered, model accuracy alone is no longer an adequate measure of safety or readiness.

This white paper presents the results and lessons of a structured initiative undertaken by mCards, Radware, and SECAI to design, secure, adversarially test, and evaluate an AI-enabled financial services environment. The objective was not to evaluate an AI model in isolation. It was to examine the complete operating environment in which an AI agent interacts with customer identity, financial data, business logic, tools, APIs, security controls, third-party services, monitoring systems, and human decision makers.

The underlying premise is straightforward: **Bank-grade AI requires system-level assurance.**

The initiative began with a controlled integration model developed around defined permissions, infrastructure protection, auditability, staged deployment, and independent validation. That foundation was expanded through the Gold Standard validation campaign to incorporate identity-bound access, least privilege, transaction and workflow constraints, real-time monitoring, prompt and tool misuse protections, runtime containment, structured adversarial testing, remediation, and independent review.

Rather than treating the AI as an unconstrained conversational interface, the framework treats it as a controlled system actor whose purpose, authority, data access, permitted actions, prohibited actions, escalation requirements, and operating boundaries must be explicitly defined.

That distinction matters because instructions to an AI are necessary, but they are not sufficient control. An agent can be instructed not to access another customer's information, exceed a transaction threshold, bypass authentication, reveal protected information, or invoke an unauthorized tool. A financial institution should not rely exclusively on the model following those instructions. Critical boundaries must also be enforced independently by the surrounding system through identity controls, permissions, transaction limits, data segmentation, approved tool sets, policy enforcement, authentication requirements, and other deterministic mechanisms.

The AI should understand the rule. The institution should enforce the rule.

The same approach applies to information integrity. Financial institutions depend on authoritative information, and material financial facts should remain anchored to approved systems of record. Account balances, transaction status, account ownership, payment status,

fees, limits, and similar information should be retrieved and verified through controlled sources rather than generated from unsupported model inference. The AI may interpret, summarize, or explain authoritative information, but it should not create the authoritative financial fact when an approved source exists.

This separation between generation and authority is essential because a probabilistic model output should not silently become deterministic authority.

Human authority must therefore be designed into the operating model according to the risk and materiality of the activity. The framework does not assume that every interaction requires manual approval. Instead, autonomy is bounded according to factors such as customer impact, financial materiality, reversibility, data sensitivity, and the authority required to complete the action.

Low-risk informational activities can operate within established controls. Higher-risk recommendations may require confirmation or review. Material financial or account actions may require additional authentication, explicit approval, dual control, or other independent authorization. Activities such as unauthorized cross-customer access, modification of the agent's own permissions, or circumvention of security controls should remain outside the agent's authority altogether.

Meaningful human oversight also requires more than the presence of an approval step. The responsible person must have sufficient information and authority to challenge, stop, or escalate an activity. This principle fits within the broader Three Lines governance structure used throughout banking. Business and technology functions own and operate the system and its controls. Risk, compliance, privacy, security, and related oversight functions establish and challenge the relevant boundaries. Internal audit independently assesses governance and control effectiveness. Specialized external adversarial testing can supplement those functions by providing technical evidence about whether controls perform as intended when they are actively challenged.

The control environment also extends beyond the institution because agentic financial systems can depend on application providers, foundation models, security platforms, cloud environments, data services, identity providers, financial APIs, orchestration technologies, and downstream dependencies. As a result, traditional vendor diligence must expand beyond evaluating the security posture of individual companies.

The institution must understand the combined technology and authority chain, including what data each material component can access, what actions the integrated system can perform, which permissions exist, how material changes are communicated, how incidents are handled, what evidence is available, and how access can be revoked or services replaced without losing operational control. This is consistent with the federal banking agencies' third-party risk framework, which applies risk management across the lifecycle of third-party relationships and makes clear that use of a third party does not diminish the banking organization's responsibility for safe, sound, and compliant operations.

Control design, however, does not establish control effectiveness.

For that reason, the initiative incorporated structured adversarial testing designed to identify where intended controls and actual system behavior diverged. An initial benchmark established normal operating behavior, telemetry, response handling, and system characteristics while explicitly distinguishing that work from hostile testing. Subsequent campaigns moved into dedicated testing across Tools Misuse, Prompt Injection, Data Access, and Data Leakage.

The objective was not simply to accumulate successful attacks. It was to identify exploitable control weaknesses, preserve evidence of observed behavior, understand the cause and significance of those weaknesses, and create a basis for remediation and retesting.

The testing reinforced another important principle: **A technical control failure is not automatically a regulatory violation.**

A successful adversarial interaction can demonstrate that an operational or security control failed, but the regulatory consequence depends on the specific facts, applicable authority, information involved, authentication state, customer context, compensating controls, and resulting impact. Security findings, operational deficiencies, regulatory analysis, and legal conclusions should therefore remain distinct stages of the assurance process. AI can support classification and analysis, but material external regulatory and legal conclusions require appropriate human review.

This distinction is increasingly important because agentic AI does not fit neatly within a single traditional banking risk discipline. In April 2026, the Federal Reserve, OCC, and FDIC issued revised interagency Model Risk Management guidance that superseded SR 11-7 and expressly stated that generative AI and agentic AI are outside the scope of that guidance. The agencies nevertheless made clear that banking organizations should use their broader governance and risk management practices to determine appropriate governance and controls for systems outside its scope.

The implication is not that agentic AI sits outside banking risk management. It is that the control environment should be evaluated across the broader risks created by the use case, including governance, cybersecurity, privacy, third-party risk, operational resilience, consumer protection, human oversight, and independent challenge.

The resulting framework therefore extends beyond traditional model validation and beyond conventional penetration testing. It asks whether the institution can identify the agent, define its business purpose, bound its authority, control the information available to it, enforce prohibited actions outside the model, preserve meaningful human command, govern the third parties on which it depends, observe its material behavior, and independently test whether those controls continue to operate under adversarial conditions.

The evidence generated through that process should give business leadership, information security, risk, compliance, privacy, vendor management, internal audit, and other authorized stakeholders a common basis for decision making. It should show what the AI is authorized to

do, where its critical boundaries are enforced, how it has been challenged, what weaknesses were identified, what remediation has been implemented, what has been successfully retested, what residual risk remains, and what changes could invalidate the prior assurance conclusion.

Where retesting is incomplete or evidence has not yet been reconciled, that condition should remain visible rather than being absorbed into a broader claim that the environment has been fully validated.

That evidence also has to remain current. Agentic AI systems can change materially when the underlying model changes, when a new tool or API is introduced, when permissions expand, when a data source is modified, when a third party changes, or when security and authentication controls are reconfigured.

Assurance applies to a defined system configuration, not indefinitely to a product name.

Material changes should therefore trigger proportionate reassessment and, where appropriate, renewed adversarial testing so that the institution's assurance conclusion continues to reflect the system actually operating in production.

The work described in this paper leads to a direct conclusion for financial institutions considering agentic AI. The appropriate standard is not trust in the model. It is demonstrable control over the system. A bank should be able to show what the AI is designed to do, what it is prohibited from doing, what authority has been delegated to it, how critical boundaries are enforced, where authoritative information originates, when humans retain decision authority, how third-party dependencies are governed, what occurred during adversarial testing, how identified weaknesses were addressed, and how control effectiveness will be maintained as the environment changes.

Bank-grade AI is not established by what the system is told to do. It is established by evidence that its purpose is defined, its authority is bounded, its prohibited behavior is explicit, its critical controls are enforced outside the model, its information is grounded, its dependencies are governed, its actions are observable, its controls withstand adversarial challenge, and accountable humans remain in command.

1. Why Agentic AI Changes the Banking Control Problem

The transition from generative AI to agentic AI changes the nature of risk for financial institutions. A generative system primarily produces information for a person to consider. An agentic system can combine reasoning with access to data, tools, APIs, workflows, and other systems in order to complete a task. Once AI is connected to financial infrastructure, an error in model behavior can move beyond an inaccurate response and become an operational, security, privacy, or customer event.

For banks, this changes the control question. Model accuracy and output quality remain important, but they are no longer sufficient measures of readiness. Institutions must also determine what information an AI system can access, which systems it can interact with, what actions it can initiate, how those actions are authorized, and when human authority must intervene. The resulting risk is created not by the model alone, but by the combination of the model and the authority surrounding it.

Technical capability and institutional authority are not the same thing. An AI system may be technically capable of accessing an account, calling an API, retrieving customer information, modifying a workflow, or initiating a financial action. That capability does not mean the institution has authorized the system to perform those activities under every circumstance. Authority must be established independently through identity, permissions, data boundaries, transaction controls, policy, and approval requirements.

The OCC's Spring 2026 Semiannual Risk Perspective reflects this broader concern by identifying advanced AI as an increasingly important consideration in bank cybersecurity and operational risk, including risks related to privacy, security, data integrity, validation, and increasingly capable AI systems. The broader supervisory implication is that greater capability and autonomy require stronger governance and controls. (OCC)

The issue becomes more significant when an agent is given a legitimate business objective but has discretion over how to accomplish it. An AI may be asked to resolve an account issue, complete a customer request, investigate an exception, or execute an approved workflow. The objective itself may be appropriate. The risk arises when the system identifies a technically available method of accomplishing that objective that conflicts with the institution's security, privacy, authorization, or operational requirements.

For banking purposes, this creates three distinct concerns.

First, the AI itself can be manipulated through prompt injection, malicious context, poisoned information, or other adversarial techniques.

Second, an attacker can attempt to use the AI as a pathway to data, tools, or systems that would otherwise be unavailable.

Third, an authorized AI agent can pursue a legitimate objective through an action that exceeds the authority the institution intended to delegate.

The third concern is particularly important because it does not require a traditional external attacker. It is fundamentally an alignment problem between the objective assigned to the agent and the boundaries within which the institution expects that objective to be achieved.

Recent technical evidence illustrates why those boundaries matter. In August 2026, the United Kingdom's AI Security Institute reported that, during a deliberately permissive cybersecurity evaluation, agents took autonomous and unsanctioned actions on the live internet in 10 of 122 evaluation runs. The institute identified 19 such actions, including an attempt to introduce malicious code into an open source project and the use of fabricated online identities to influence a human maintainer. The institute also emphasized important limitations: the testing environment provided open internet access, some safety controls were disabled, and the results should not be generalized to ordinary production deployments. (AI Security Institute)

The relevance for financial institutions is not that AI systems should be expected to behave this way in normal operations. It is that capable agents can take unintended actions when objectives, operational access, and external controls are not sufficiently aligned. The appropriate response is neither to assume that the agent will behave incorrectly nor to assume that well-written instructions will always prevent inappropriate behavior. The operating environment must enforce boundaries that remain effective when the model produces an unexpected decision.

This makes the distinction between reasoning and authorization fundamental.

An AI can reason that an action would help accomplish a task, but that conclusion should not create the authority to perform the action. A conversational request should not establish identity. A model's interpretation of customer intent should not expand data access. A recommendation to use a tool should not create permission to invoke that tool. A generated conclusion that a transaction appears appropriate should not override an authentication requirement, approval condition, or transaction limit.

In banking, the model may participate in a decision process without becoming the source of authority for that decision.

The same principle applies when AI interacts with financial information. An inaccurate answer used only in conversation may create customer confusion. The same error used to populate a workflow, select a financial tool, or influence an account action can create a materially different consequence. Risk therefore increases as generated information becomes more closely connected to execution.

This is why the appropriate unit of risk is the complete AI-enabled system. The institution must evaluate the model together with its identity, permissions, data access, tools, APIs, operating instructions, security controls, third-party dependencies, monitoring, human oversight, and surrounding financial infrastructure. A model can perform well in isolation while the combined

system still contains excessive permissions, inadequate data boundaries, weak authorization controls, or unsafe operational pathways.

The mCards and Radware initiative was designed around this broader view. The Partnership framework contemplated an AI agent operating as a controlled participant within the mCards environment, subject to defined permissions, auditability, embedded security controls, governance, and independent validation. The Gold Standard extended that control model to include identity-bound access, least privilege, transaction and workflow constraints, monitoring, prompt and tool misuse protections, runtime containment, and structured adversarial testing.

The resulting banking control problem is therefore not whether AI can perform increasingly sophisticated tasks. The relevant question is whether the institution can demonstrate that the AI performs those tasks without exceeding the authority required to complete them.

The relevant unit of risk is the system, not the model.

Once authority becomes part of the risk, the next question is who defines, governs, challenges, and ultimately retains that authority. That is the role of governance, accountability, and human control.

FINAL - EYES ONLY

2. Governance, Accountability, and Human Authority

Once authority becomes part of the risk, the next question is who defines that authority, who challenges it, and who remains accountable when the AI acts.

For a financial institution, accountability cannot be assigned to the model. The institution remains responsible for the business activity, customer outcome, control environment, and risks created by the technology used to deliver the service. AI governance must therefore begin with clear ownership of the use case and explicit allocation of responsibility across business, technology, risk, compliance, security, privacy, and independent assurance functions.

This is consistent with established banking governance principles. The Basel Committee's corporate governance framework assigns responsibility for strategy, risk management, control functions, and safe and sound operation to the board and senior management. Within the Three Lines model, business and operational functions own and manage risk, independent risk management and compliance functions provide challenge and oversight, and internal audit provides independent assurance. (Bank for International Settlements)

Agentic AI does not change those responsibilities. It changes the technology through which they must be exercised.

The mCards and Radware initiative was structured around the same principle. The Partnership framework contemplated an AI agent operating as a controlled participant within the mCards environment, subject to defined permissions, monitoring, auditability, security controls, staged deployment, and independent validation. The Gold Standard further incorporated identity-bound access, least privilege, transaction and workflow constraints, monitoring, adversarial testing, and defined responsibilities for operation, oversight, and challenge.

For a bank operating model, those roles should remain precise.

Business and technology functions own the use case and operate its controls. Risk, compliance, privacy, security, and other second line functions independently establish or challenge the boundaries within which the system may operate. Internal audit remains the institution's third line and independently evaluates the effectiveness of the overall governance and control framework.

Specialized external testing, including the SECAI adversarial work described in this paper, can provide additional technical evidence to management, second line functions, and internal audit. It supplements those functions. It does not replace the institution's third line.

This distinction matters because effective AI governance requires both ownership and challenge. The team responsible for delivering the capability must understand and manage the risks created by it, while independent functions must be able to determine whether the use case, delegated authority, controls, monitoring, and residual risk remain appropriate.

Governance weakens when accountability is dispersed among vendors, technical teams, and committees without a clearly accountable owner.

NIST's AI Risk Management Framework supports a similar approach. Its GOVERN function emphasizes structures for establishing roles, responsibilities, policies, accountability, documentation, and communication throughout the AI lifecycle. The framework is voluntary, but it provides a useful reference for integrating AI governance into existing enterprise risk structures. (NIST)

For agentic systems, one of the most consequential governance decisions is the amount of authority delegated to the AI and the circumstances under which that authority returns to a human.

Human oversight should be proportionate to risk rather than applied uniformly to every interaction. Requiring manual intervention for routine, low-risk activity would reduce the value of automation without necessarily improving control effectiveness. The more useful approach is to align the form of oversight with factors such as financial materiality, customer impact, reversibility, data sensitivity, complexity, and the authority required to complete the action.

A low-risk informational task may operate without additional human approval where identity, access, and data controls are functioning as intended. A higher-risk recommendation may require customer confirmation or employee review. A material transfer, account change, exception, or other consequential action may require step-up authentication, explicit approval, dual control, or escalation.

Activities that would violate the institution's security, privacy, legal, or authorization boundaries should remain unavailable to the agent regardless of whether a human could otherwise choose to perform them through an authorized process.

This is the difference between human participation and human authority.

A human approval step is meaningful only when the reviewer has sufficient information, time, and authority to understand the proposed action, question it, reject it, escalate it, or stop execution. An approval process provides limited protection if the reviewer merely confirms a machine recommendation without meaningful ability to challenge it.

Emerging international financial sector work reflects the same direction. In its June 2026 consultation on responsible AI adoption, the Financial Stability Board emphasized human oversight proportionate to the materiality, risk, autonomy, complexity, and explainability of the use case, together with accountability, lifecycle risk management, cybersecurity, and third-party dependencies. Because the document is consultative and non-binding, it should be viewed as developing supervisory thinking rather than a regulatory requirement. (Financial Stability Board)

For banking purposes, the objective is to decide explicitly where authority may be delegated and where it must remain with the institution.

The AI may perform an authorized task, but it does not establish the institution's risk appetite. It may recommend an action, but it does not decide which controls are optional. It may operate within approved permissions, but it does not expand those permissions because additional access would make the task easier. When an assigned objective conflicts with a security, privacy, compliance, authorization, or human control boundary, the boundary must prevail.

Senior management therefore has a clear role in determining which AI use cases are sufficiently understood and controlled to proceed, which require additional safeguards, and which should remain outside the permitted operating scope. Those decisions should be supported by evidence about system behavior, delegated authority, control effectiveness, third-party dependencies, customer impact, and residual risk rather than by the capabilities or assurances of the model provider alone.

The result is a governance model in which autonomy is intentionally delegated rather than implicitly inherited from technology.

Business ownership establishes the purpose. Risk and control functions establish and challenge the boundaries. Human authority is retained where material judgment or approval is required. Internal audit evaluates whether the governance framework is operating effectively. Independent technical testing provides evidence about whether the controls withstand actual challenge.

With accountability established, the next question is more specific: what authority should the agent itself receive, and how should that authority be constrained? That is the purpose of agent authority and guardrail design.

3. Agent Authority and Guardrail Design

Once accountability is established, the next question is what authority the institution is prepared to delegate to the AI and how that authority will be constrained.

For an agentic system, permissions cannot be inferred from technical capability. The fact that an agent can call an API, retrieve a record, initiate a workflow, or construct a transaction does not mean it should have unrestricted authority to do so. The institution must define that authority explicitly and enforce it at the system level.

The Gold Standard was designed around this principle. It treats the AI agent as a privileged system actor subject to identity-bound access, role-based permissions, least privilege, transaction and workflow constraints, audit logging, real-time monitoring, prompt and tool misuse detection, and runtime containment. The Partnership framework similarly requires the agent to interact with mCards services only through approved interfaces and workflows, with defined boundaries for data access, permissions, and execution.

This paper refers to the resulting control principle as **Least Agency**.

Least privilege traditionally limits access to the minimum resources required to perform an authorized function. Least Agency extends that concept to the behavior of an autonomous system. It asks not only what information an agent can access, but what it can decide, initiate, modify, communicate, delegate, or execute without additional authorization.

For each material use case, the institution should define six elements: the agent's purpose, permitted activities, prohibited activities, operating constraints, escalation requirements, and fail-safe behavior. Together, these elements form a Behavioral Authority Specification that establishes the boundaries within which the agent may operate.

One overriding rule applies across all six elements:

An authorized business objective does not override a security, privacy, compliance, authorization, or human control boundary.

If an agent cannot complete an assigned task without crossing one of those boundaries, the appropriate behavior is not to find a path around the control. It is to stop, preserve the relevant context, and escalate according to the institution's defined process.

This approach is consistent with established financial sector access-control practices. FFIEC guidance emphasizes risk-based authentication, layered security, and access controls appropriate to the sensitivity and risk of the activity, including system-to-system access and third-party connectivity. For agentic AI, the practical implication is that the agent should be treated as a distinct non-human identity with explicitly defined authority rather than inheriting permissions implicitly from the customer, application, or developer operating around it. (FFIEC)

The guardrail model can be organized into four control areas.

Information integrity controls govern what the AI is permitted to state as fact and how material information is verified. Financial facts such as account balances, transaction status, account ownership, limits, fees, and payment status should be grounded in approved sources when those sources exist. The model may interpret or explain that information, but unsupported generation should not replace authoritative system state.

This is particularly important because NIST identifies confabulation, commonly referred to as hallucination, as a distinct generative AI risk. NIST's Generative AI Profile also addresses prompt injection, information security, privacy, and other risks that can affect model behavior and system integrity. The framework is voluntary, but it provides a credible technical reference for managing those risks. (NIST)

For financial services, information integrity should be evaluated according to consequence. An incorrect conversational answer is undesirable. The same incorrect output used to populate a workflow, determine an account action, or influence execution creates a materially different risk. The control environment should therefore prevent generated information from silently acquiring operational authority.

Data and privacy controls govern what information the agent may access, process, retain, and disclose. Customer and account boundaries should be enforced independently of conversational context. An authenticated interaction with one customer should not create technical access to another customer's information, and a prompt requesting broader access should not expand the agent's underlying permissions.

The adversarial testing described later in this paper demonstrates why these controls require direct validation. The Data Access campaign tested whether an authenticated user could cause the system to cross an authorization boundary, while the Data Leakage campaign examined whether the system could disclose information beyond the intended presentation boundary.

Operational controls govern which tools, APIs, workflows, and financial actions the agent can invoke and under what conditions. Permissions should be constrained by function, user context, account context, parameters, transaction thresholds, and other applicable policy conditions. Higher-risk actions may require additional authentication or independent approval, while prohibited actions should remain technically unavailable.

This is where the distinction between semantic and deterministic guardrails becomes most important.

A system instruction can tell an agent not to exceed a transaction limit. The system executing the transaction should independently enforce that limit. A system instruction can prohibit cross-customer access. The data layer should independently prevent the access. A system instruction can prohibit use of a restricted tool. The authorization layer should determine whether that tool can be invoked at all.

The AI should understand the rule. The institution should enforce the rule.

Human and governance controls define where machine authority ends. At the guardrail level, governance must translate into specific approval, escalation, intervention, and shutdown conditions. The agent should be able to recognize when it lacks authority to continue, while the surrounding system should independently prevent execution where required authorization is absent.

These control areas must operate together. Strong instructions do not compensate for excessive permissions. Strong authentication does not compensate for weak authorization. Strong API security does not compensate for unreliable financial information. Human approval does not compensate for a reviewer who cannot understand or challenge the proposed action.

The objective is therefore not to create a larger collection of prompts or policy statements. It is to create a layered authority model in which failure of one control does not automatically produce an unauthorized outcome.

That distinction was central to the Gold Standard validation design. The campaign was intended to test not only prompt behavior, but also tool misuse, workflow boundaries, data access, containment, monitoring, and the integrity of the surrounding control environment.

For a financial institution, the result should be a clear answer to three questions:

What may the agent know? What may it say? What may it do?

Those answers should be documented, technically enforceable, observable, and independently testable.

With the agent's authority and guardrails defined, the next question is where those controls operate in the actual system. The following section examines how the control model is implemented across identity, orchestration, financial APIs, data sources, runtime security, monitoring, and human intervention points.

4. System Architecture and Control Implementation

With the agent's authority defined, the next step is to determine where those boundaries are enforced in the operating environment. For a financial institution, this is where governance becomes architecture. The control model must be reflected in identity, data access, application logic, security enforcement, tool permissions, financial APIs, monitoring, and human intervention points.

The mCards environment was designed around this principle. The Partnership framework describes the AI agent as a controlled participant within the mCards platform rather than as a standalone system. The agent is intended to interact with mCards services through approved interfaces and workflows, with defined boundaries for permissions, data access, execution, logging, monitoring, and controlled deployment. The Gold Standard extends this design through identity-bound access, least privilege, transaction and workflow constraints, behavioral and contextual controls, prompt and tool misuse detection, runtime containment, and integrated security telemetry.

The architectural objective is to separate the AI's ability to reason from the systems that determine what it is permitted to access and execute.

The model should not become the source of its own authority.

Identity should be established independently. Access should be permissioned independently. Financial data should originate from controlled sources. Tool invocation should remain subject to application and policy controls. Material actions can be conditioned on authentication, transaction rules, or human approval. Activity should be recorded outside the conversational interface.

This approach is consistent with established security architecture principles. NIST Special Publication 800-207 applies zero trust concepts to both human and non-human entities and emphasizes explicit access decisions, least privilege, and protection of individual resources rather than broad implicit trust. FFIEC architecture guidance similarly emphasizes governance, security, resilience, interconnected systems, and third-party dependencies as part of technology risk management. (NIST; FFIEC)

Within the tested environment, the project documentation identifies a Synthflow-based conversational AI application connected to financial services through an MCP-based tool orchestration layer. The documented environment included Fiserv AllData and MasterCard-related capabilities, with tools supporting activities such as account information retrieval, balance inquiries, transaction retrieval, financial analysis, and funds movement.

These connections are material because each represents a boundary at which model output can potentially become data access or operational action.

The first architectural boundary is **identity and authorization**. The customer, application, agent, and downstream financial service should not be treated as a single security principal. The customer establishes who is requesting the service. The agent operates under a separately defined system identity and authority. The downstream service independently determines whether the requested operation is permitted for the relevant customer, account, resource, and action.

This separation supports attribution and prevents conversational context from becoming a substitute for authentication or authorization.

The second boundary is **tools and APIs**. The agent should reach financial functions only through approved interfaces that expose the minimum capability required for the use case. A read-only account function should not provide implicit access to unrelated records or transaction capabilities. A transaction function should accept only permitted parameters and remain subject to account ownership checks, transaction limits, authentication requirements, and other applicable controls.

The agent may formulate a request. It should not determine whether the request is authorized for execution.

The third boundary is **data access**. The AI may require financial information to complete an authorized task, but access should remain tied to the authenticated customer, approved business purpose, and permitted data set. Customer isolation, account scope, field restrictions, masking, and other protections should operate independently of what the model requests.

This boundary became particularly important during the adversarial testing because data access and information disclosure were identified as meaningful attack surfaces in the evaluated environment.

The fourth boundary is **runtime security**. The Partnership and Gold Standard contemplate Radware capabilities around the AI and surrounding infrastructure to evaluate behavior and context, inspect instructions, identify suspicious prompt and tool activity, generate telemetry, and support containment when activity exceeds established boundaries.

This layer can provide visibility beyond conventional authentication and API controls. A request may be technically valid while the sequence or context producing it is suspicious. Runtime security can therefore provide additional detection and containment capability around otherwise authorized interfaces.

The sequencing of the project must, however, remain precise.

The initial attack cycle occurred before the remediation phase identified in the project roadmap. That remediation phase included deployment of Radware API protection, strengthening of guardrails, and additional mCards hardening. Architectural intent and measured control effectiveness are therefore separate questions.

This section describes where the controls were designed to operate. It does not establish that every planned control was active or effective during the initial red team cycle. Evidence of effectiveness must come from the applicable post-remediation testing.

That distinction is essential to credible assurance.

The fifth architectural boundary is **observability**. Agentic systems require sufficient telemetry to reconstruct material activity across the user interaction, agent, tool layer, security controls, and downstream financial systems. The initial benchmark was used in part to establish telemetry and normal operating characteristics before hostile testing began.

For a financial institution, the objective is not to capture every internal model process. It is to preserve enough evidence to determine who initiated the interaction, which agent and configuration were involved, what data or tool was requested, what action occurred, which control permitted or denied the action, and whether human intervention was required.

The sixth boundary is **human intervention**. Where the governance model requires confirmation, escalation, or independent approval, the architecture must provide an actual enforcement point at which execution can be paused or prevented.

Human authority should not depend solely on the model deciding to ask for assistance. The surrounding workflow should know when approval is required and should prevent completion until that requirement has been satisfied.

These controls are most effective when they remain independent but coordinated.

Identity should not depend on the prompt. Data isolation should not depend on the model correctly interpreting customer context. Transaction limits should not depend on the agent remembering a policy. Monitoring should not depend solely on the application reporting successful actions. Human approval should not depend on the model deciding that an action appears important.

The objective is defense in depth without unnecessary complexity. Each material boundary should have an identified owner, an enforcement point, a source of evidence, and a defined response when the boundary is reached or violated.

The architecture should also distinguish between prevention and detection. Some actions should be technically unavailable to the agent. Others may be permitted only under defined conditions. Still others may be allowed but monitored for abnormal behavior. A financial institution should know which category applies to each material capability rather than treating all guardrails as equivalent.

The resulting architecture can be understood as a controlled path from customer request to financial action. Identity establishes the requesting party. The agent interprets the request within its defined authority. Policy and runtime controls evaluate the requested access or action. Approved tools and APIs connect the agent to authoritative financial services. Downstream

systems enforce their own controls. Monitoring records material activity, and human authority intervenes where required.

The model can reason, but it does not authenticate itself. It can formulate an action, but it does not grant itself permission. It can request financial information, but it does not define the authoritative record. It can recommend execution, but it does not determine which controls may be bypassed.

This architecture establishes the technical boundary of the AI-enabled system. The next question extends beyond that boundary because models, data services, security platforms, APIs, cloud services, and other external dependencies can materially influence the behavior, availability, security, and authority of the complete environment.

The following section addresses how those dependencies should be evaluated through third-party risk management, regulatory alignment, and due diligence.

FINAL-EYES ONLY

5. Third Party Risk, Regulatory Alignment, and Due Diligence

The architecture described in the preceding section extends beyond the institution itself. Models, cloud services, data providers, security platforms, APIs, orchestration technologies, identity services, and other providers can each influence how an agent operates. For a financial institution, this makes third party risk management part of the AI control environment rather than a separate vendor management exercise.

The governing principle is established in the Federal Reserve, FDIC, and OCC Interagency Guidance on Third Party Relationships. A banking organization's use of a third party does not diminish its responsibility to operate safely and soundly and comply with applicable laws and regulations. The guidance applies a risk based approach across the relationship lifecycle, including planning, due diligence and selection, contracting, ongoing monitoring, and termination.

For agentic AI, this principle has an important consequence. A bank cannot establish that an AI capability is sufficiently controlled simply because its providers have strong security programs, recognized certifications, or previous experience serving financial institutions. The institution must evaluate the risk created by the specific use case and the complete system supporting it. The interagency guidance similarly emphasizes that the level of diligence and oversight should reflect the risk, complexity, and nature of the relationship.

The relevant unit of diligence is the use case and the combined system supporting it.

This distinction becomes particularly important when an agent can access sensitive customer information, interact with financial systems, invoke tools, or support transactions. The same technology provider may present very different risk when its service is limited to summarizing public information than when it participates in customer authentication, account access, transaction processing, or financial execution.

Due diligence should therefore establish how responsibility and authority are distributed across the service chain. The institution should understand which party controls the model, system instructions, authentication, permissions, data access, tool orchestration, financial execution, security enforcement, monitoring, and evidence retention.

The analysis should also extend to material downstream dependencies. The interagency guidance recognizes that third parties may rely on subcontractors and that banking organizations should evaluate how those relationships are managed when they can affect sensitive information, operations, or higher risk activities. For agentic AI, these dependencies may include model providers, hosting environments, data services, external tools, APIs, and other components that do not have a direct contractual relationship with the bank but can materially influence the service.

The mCards initiative reflects this broader approach. Its Partnership and Gold Standard frameworks address the agent together with the surrounding financial infrastructure, security

controls, identity and access mechanisms, APIs, third party dependencies, monitoring, and independent validation rather than limiting assessment to conversational model behavior.

This creates an additional diligence requirement around delegated authority.

Traditional third party reviews often focus on the service a provider performs. Agentic AI requires the institution to also understand what authority the integrated technology receives. A capability that can retrieve information presents a different control problem from one that can initiate workflows, invoke transaction functions, or communicate with additional systems.

The diligence record should therefore make the agent's business purpose, data access, tools, permissions, transaction capabilities, prohibited activities, required human approvals, material dependencies, monitoring controls, and testing status visible to the functions responsible for approving and overseeing the use case.

Contracting should reinforce those boundaries. The interagency guidance identifies information security, confidentiality, regulatory compliance, business continuity, audit and reporting, incident management, subcontractors, and termination among the issues institutions should consider according to the nature and risk of the relationship.

For agentic AI, several questions become particularly important. The institution should understand whether it receives notice of material changes to the model, architecture, security controls, or subprocessors; whether relevant logs and testing evidence are available; whether independent testing is permitted; how incidents are investigated and communicated; how institutional and customer information may be used; and how credentials, data, access, and evidence are handled when the relationship ends.

These provisions matter because an AI service can change materially without changing its product name. A new model, provider, dependency, permission, or security configuration can alter the behavior or authority of the combined system and therefore change the institution's risk.

Regulatory alignment should be approached with the same precision.

The objective is not to describe an AI solution as broadly compliant with every banking regulation or technical standard. Applicability depends on the institution, activity, customer relationship, data, jurisdiction, and use case. A more defensible approach is to map the system's controls and evidence to the regulatory obligations and supervisory expectations that actually apply.

The April 2026 revised interagency Model Risk Management guidance reinforces this point. The Federal Reserve, OCC, and FDIC superseded SR 11-7 and expressly stated that generative AI and agentic AI are outside the scope of the revised guidance because they are novel and rapidly evolving. The agencies also stated that broader governance and risk management practices should guide the controls applied to systems outside the guidance's scope.

The implication is not that agentic AI falls outside banking risk management. It is that traditional model risk management should not be treated as the sole control framework for an agentic system.

FFIEC technology examination guidance provides part of that broader framework. Its Development, Acquisition, and Maintenance booklet addresses governance, development and acquisition, interconnected systems and third party services, security, resilience, maintenance, and change management. These are directly relevant to AI capabilities whose behavior depends on multiple technologies and external providers.

Privacy requirements must likewise be evaluated according to applicability. Regulation P, for example, governs the treatment of nonpublic personal information for financial institutions within its scope, including limitations on certain disclosures, redisclosures, and reuse. The practical control question for an AI system is therefore not simply whether privacy requirements have been documented, but whether the architecture and operating procedures enforce the institution's applicable data use and disclosure boundaries.

Other regulatory and standards frameworks may also be relevant depending on the institution and use case. Rather than expanding the principal narrative into a catalog of ISO, NIST, SEC, CMMC, cybersecurity, privacy, and industry requirements, those mappings are better maintained in a detailed applicability and control crosswalk. This preserves an important distinction: a framework may provide a useful control reference without being legally binding on every financial institution, and alignment should not be described as certification unless certification has actually occurred.

Third party diligence also does not end when the system enters production. The interagency guidance treats ongoing monitoring as part of the lifecycle and expects the level of oversight to remain commensurate with the risk and complexity of the relationship. Monitoring should be capable of identifying material developments such as security incidents, service interruptions, control deterioration, compliance concerns, and other changes that can affect risk.

Termination and transition planning are part of the same control model. For an agentic capability, the institution should be able to revoke credentials, disable tool access, protect or return required information, preserve necessary evidence, and transition the service without creating unmanaged operational dependence on a particular model or technology provider. The result is a more rigorous definition of AI due diligence.

The question is not whether a vendor has passed a security review. It is whether the institution has sufficient evidence to understand the complete capability, the authority delegated to it, the dependencies supporting it, the controls governing it, and the risks that remain. That evidence can establish how the system was designed and how its controls are intended to operate.

It cannot establish whether those controls withstand deliberate attempts to defeat them. That requires adversarial validation.

6. Adversarial Validation Methodology

The preceding sections describe how the AI-enabled environment was intended to be governed and controlled. Adversarial validation was used to determine whether those controls continued to operate when the system was subjected to behavior outside normal use.

The objective was not to test whether the model could answer difficult questions. It was to determine whether an authenticated or adversarial user could cause the combined system to access information, invoke tools, disclose data, or behave outside the boundaries established by the operating environment.

The validation program was conducted in stages so that ordinary behavior could be distinguished from hostile behavior and so that findings could be supported by preserved evidence rather than observation alone.

The first stage established a baseline. On May 12, 2026, SECAI conducted a 50-iteration benchmark against the Money Manager conversational banking agent. The benchmark established normal operating characteristics, telemetry, response handling, tool invocation patterns, and resource consumption. The underlying report explicitly states that the hostile attack loop was not implemented during this benchmark. The baseline therefore established normal behavior, not adversarial resilience.

That distinction is important. A system may operate normally under expected use while still containing weaknesses that become visible only when users deliberately challenge identity, instructions, permissions, tools, and data boundaries.

The subsequent red team program expanded testing across four principal attack domains: Tools Misuse, Prompt Injection, Data Access, and Data Leakage.

Tools Misuse testing examined whether otherwise approved capabilities could be invoked outside their intended operating conditions, including misuse of parameters, sequencing, or workflow logic.

Prompt Injection testing examined whether adversarial instructions could cause the AI to depart from intended behavior or circumvent operating constraints.

Data Access testing examined whether an authenticated user could cause the system to retrieve information outside the intended customer or authorization boundary.

Data Leakage testing examined whether the system could be induced to disclose information beyond what should be presented through the conversational interface.

The testing was intentionally broader than conventional prompt evaluation. Prompts were the attack mechanism, but the object under test was the complete system, including the application, data boundaries, tools, orchestration, and downstream functions available to the agent.

This distinction matters because an unusual or undesirable model response is not necessarily a material security event. The relevant question is what the interaction caused the system to do.

The campaign therefore relied on defined technical success criteria. An attack was not treated as successful simply because the model responded to it. Testing examined whether the interaction produced behavior such as unauthorized access, policy bypass, inappropriate tool use, information disclosure, or another defined departure from the intended control boundary.

The documented campaign volumes include 1,350 Prompt Injection probes, with 978 interactions meeting the campaign's successful breach criteria; 1,176 targeted Data Access probes, with 276 reported successful boundary crossings; and 1,988 Data Leakage probes, with 826 interactions meeting the campaign's leakage criteria. Tools Misuse testing was documented separately.

These figures should be interpreted by campaign and control objective rather than combined into a single headline number. The underlying project materials were produced at different stages and contain differences in aggregate reporting and some classifications. Consolidated statistics should therefore be used only after the evidence has been reconciled across campaign, configuration, test count, success criteria, finding, remediation state, and retest result.

Testing conditions are also important to interpretation. During portions of the red team exercise, controls including CAPTCHA, two-factor authentication, and per-user rate limits were intentionally relaxed to permit sustained automated probing. That approach can expose weaknesses more efficiently, but it can also increase the apparent scale of exploitation if the test environment is not clearly distinguished from production conditions.

The appropriate conclusion is therefore not that every successful test would occur at the same rate in production. It is that the testing identified control conditions that required analysis and, where appropriate, remediation.

Evidence preservation was another important component of the methodology. The baseline assessment documented structured telemetry and contemporaneous evidence, while the subsequent campaign volumes preserved attack prompts, system responses, classifications, and analysis at the individual finding level. This created an evidence chain that allowed findings to be reviewed after execution rather than relying solely on tester narrative.

The methodology also separated technical findings from regulatory analysis.

A successful technical attack first established an observed system behavior. That behavior was then evaluated for operational, security, privacy, or regulatory significance. This sequence is important because a control failure and a legal violation are not the same determination.

The Data Leakage campaign illustrates the distinction. Some successful leakage findings involved excessive or unmasked information presented to an authenticated user about that user's own information rather than another customer's data. Those conditions could still

represent meaningful control deficiencies without producing the same regulatory implications as unauthorized cross-customer disclosure. The Data Access campaign, by contrast, focused specifically on attempts to cross customer or authorization boundaries.

Regulatory classification was therefore treated as a separate analytical layer rather than as the definition of technical success.

The quality assurance process added another level of review. Dedicated QA artifacts were produced for the Prompt Injection, Data Access, and Data Leakage campaigns to determine whether the individual findings contained the required prompts, responses, classifications, reasoning, regulatory analysis, compensating-control information, and other evidence fields.

Across those QA artifacts, the documented review covered 978 Prompt Injection findings, 276 Data Access records, and 826 Data Leakage records. The QA process was designed to strengthen completeness and consistency of the evidence corpus before the findings were used in broader analysis.

The QA process should not be interpreted as regulator approval, legal certification, or proof that every regulatory classification was correct. Portions of the jurisdictional and regulatory analysis used Gemini 2.5 Flash as an analytical tool. For a bank-grade assurance process, AI-assisted regulatory analysis should be treated as supporting analysis and triage. Material legal or regulatory conclusions used for external representation should remain subject to qualified human review.

Taken together, the methodology followed a disciplined sequence: establish normal behavior, challenge defined control boundaries, preserve evidence, apply technical success criteria, assess operational and regulatory significance, perform quality assurance, remediate identified weaknesses, and retest the affected controls.

That sequence distinguishes adversarial assurance from ordinary functional testing.

Functional testing asks whether the system performs an intended task. Adversarial validation asks whether the system can be caused to perform a task the institution did not intend to authorize.

The value of the campaign is therefore not the number of attacks executed in isolation. Its value is the evidence it created about where the control environment failed, why those failures mattered, and what needed to change before the system could be evaluated again.

The next section examines those findings together with the remediation and retest process, connecting material control weaknesses to the response and residual risk that followed.

7. Findings, Remediation, and Retest

The adversarial validation program identified meaningful gaps between intended control behavior and observed system behavior under attack. Those findings were valuable not because of their volume alone, but because they showed where authority, data boundaries, instructions, and runtime controls could be manipulated and where the surrounding control environment required strengthening.

For a financial institution, the most useful way to interpret those results is by control theme rather than by test volume. Each material finding should be connected to the control that failed, the remediation required, the evidence produced, and the residual risk that remains.

One limitation is important. The project materials document the initial findings, quality assurance process, and remediation roadmap, but they do not yet contain a complete reconciled post-remediation dataset sufficient to support final before-and-after effectiveness claims across every attack class. Any conclusion about control improvement should therefore distinguish between remediation that was implemented or planned and remediation that has been independently validated through retesting.

Authority and Tool Control

The Tools Misuse campaign examined whether capabilities available to the agent could be invoked outside their intended operating boundaries, including misuse of parameters, workflow logic, sequencing, or other mechanisms through which a legitimate function could produce an unauthorized result.

This class of finding demonstrates that access to an approved tool does not establish authority to use that tool under every condition. An agent may be authorized to call a financial function while remaining prohibited from using particular parameters, crossing account boundaries, exceeding transaction limits, or combining capabilities in an unauthorized sequence.

The control response must therefore operate at the execution layer. Relevant measures include parameter validation, user and account scoping, least privilege, workflow restrictions, transaction controls, and independent authorization for higher-risk actions.

The Gold Standard anticipated this need by treating the AI as a privileged system actor subject to identity-bound access, least privilege, workflow and transaction constraints, monitoring, and runtime containment. The initial attack cycle provided evidence about where those intended boundaries required additional hardening.

Retesting should establish whether the same or equivalent misuse paths remain technically available and whether prohibited behavior is prevented, detected, or contained as designed.

Prompt and Instruction Manipulation

The Prompt Injection campaign documented 1,350 probes, with 978 interactions meeting the campaign's successful breach criteria under the tested configuration.

The significance of these findings is not simply that the model could be induced to produce undesirable responses. The more important question is whether manipulated instructions could influence access to data, use of tools, workflow behavior, or other operational capabilities.

This finding class reinforces the distinction between semantic and deterministic controls. System prompts and model guardrails should define permitted and prohibited behavior, but critical banking boundaries should not depend on the model reliably following those instructions.

The remediation approach therefore operates at two levels. Model and prompt controls can be strengthened to improve resistance to manipulation, while external controls should ensure that successful prompt injection cannot create permissions, broaden access, bypass transaction restrictions, or invoke capabilities the agent was not otherwise authorized to use.

The project roadmap reflects this layered approach. Following the initial attack cycle, the remediation phase included strengthening guardrails, additional mCards hardening, and deployment of Radware API protection before subsequent testing.

The relevant retest question is therefore not simply whether fewer prompts succeed. It is whether manipulated model behavior can still cross an operational boundary.

Identity and Data Isolation

The Data Access campaign tested whether an authenticated user could cause the system to retrieve information outside the intended customer or authorization boundary. The campaign documented 1,176 targeted probes and reported 276 successful boundary crossings under the tested configuration.

This was a material finding because it tested whether conversational context could influence actual data authorization.

The remediation principle is clear. Customer and account isolation should not depend on the AI correctly deciding whose information it should retrieve. Identity, account scope, resource ownership, and field-level permissions should be independently enforced by the surrounding application and downstream data services.

The testing conditions also require careful interpretation. CAPTCHA, two-factor authentication, and rate limits were intentionally relaxed during portions of the campaign. Those controls can reduce attack volume and strengthen session assurance, but they do not independently correct a condition in which an authenticated user can access another customer's data.

Retesting should therefore demonstrate that customer and account boundaries remain enforced regardless of how the request is phrased or which conversational path is used.

Information Disclosure and Integrity

The Data Leakage campaign documented 1,988 probes, with 826 interactions meeting the campaign's leakage criteria.

The findings illustrated why different forms of disclosure should not be treated as equivalent. Some involved excessive or unmasked information presented to an authenticated user about that user's own account. Others may implicate different privacy or authorization concerns depending on the information and context.

Cross-customer disclosure requires strong prevention because it represents a failure of authorization and data isolation. Excessive disclosure to the correct customer may instead require controls involving masking, minimization, field selection, response formatting, or restriction of unnecessary system detail.

The same principle applies to information integrity. An authenticated user may be fully entitled to receive information while the response itself is incomplete, unsupported, or incorrect. Material financial facts should therefore remain grounded in authoritative systems of record, and generated content should not silently become the basis for a financial action.

Retesting should address both dimensions: whether unauthorized information can still be disclosed and whether authoritative financial information remains accurate, appropriately limited, and separated from unsupported model generation.

Control Interaction and Remediation

The campaigns also showed that individual controls cannot be evaluated in isolation.

Authentication can succeed while authorization remains weak. A model can reject many injection attempts while a backend tool still accepts an unsafe request. Rate limiting can reduce attack volume while leaving a data-access defect unresolved. Strong instructions can coexist with excessive permissions.

Remediation must therefore be evaluated as a control system rather than as a series of isolated fixes.

The project roadmap reflects this broader approach. Remediation responsibilities were divided across mCards and Radware and included guardrail strengthening, application hardening, additional API protection, and further technical review before renewed testing.

A control should not be considered validated merely because a configuration changed or a new security capability was deployed. Validation requires evidence that the original weakness is no

longer exploitable under the tested conditions, that equivalent bypass paths have been considered, and that the remediation has not introduced an unacceptable new risk.

For each material weakness, the assurance record should therefore connect five elements:

1. The observed behavior and affected control.
2. The root cause and remediation.
3. The responsible owner.
4. The retest evidence.
5. The remaining residual risk and approval decision.

This creates a direct chain from adversarial evidence to control assurance.

The existing project materials are strong enough to support the initial findings and remediation rationale, but final before-and-after claims should be made only after the underlying campaign statistics, regulatory classifications, configuration states, and retest outcomes have been reconciled. Where retesting is incomplete, that condition should remain explicit.

That discipline is part of the assurance process.

The relevant question for a financial institution is not how many weaknesses were found during the initial campaign. It is whether material weaknesses were identified, understood, remediated, challenged again, and reduced to a level of residual risk that the institution is prepared to govern.

That is the transition from red teaming to assurance.

Once a control has been validated, the remaining question is how long that assurance remains applicable as the model, permissions, tools, APIs, providers, and surrounding security environment continue to change. The next section addresses continuous assurance, operational resilience, and change-triggered revalidation.

8. Continuous Assurance, Resilience, and Change

The preceding section establishes that remediation is not complete until the affected control has been challenged again and the remaining risk is understood. The next question is how long that assurance remains valid.

Agentic AI systems can change materially after testing. The model can be updated. System instructions can change. New data sources, tools, APIs, workflows, permissions, or third parties can be introduced. Authentication and security controls can be modified. A capability initially limited to information retrieval can later receive transaction authority.

Any of these changes may alter the system that was originally tested.

The appropriate principle is therefore:

Assurance applies to a defined system configuration, not indefinitely to a product name.

The mCards and Radware project design reflected this lifecycle approach through controlled deployment environments, change management, logging, access controls, security monitoring, remediation, and retesting. The project roadmap similarly treated the initial attack cycle, remediation, hardening, and subsequent testing as distinct stages of an assurance process rather than as a one-time certification exercise.

For a financial institution, the practical challenge is determining which changes are significant enough to require renewed assurance.

Not every software update requires a full red team exercise. Revalidation should be proportionate to the likelihood that a change affects authority, information integrity, privacy, cybersecurity, customer impact, or operational behavior.

Changes to the foundation model may affect reasoning, refusal behavior, tool selection, or susceptibility to manipulation. Changes to system instructions may alter behavioral boundaries. New data sources can affect privacy, accuracy, and customer isolation. New tools can expand what the agent is capable of executing. Changes to permissions can materially alter delegated authority. New providers or subprocessors can introduce additional security, availability, and data-handling dependencies.

Changes involving identity, customer data boundaries, transaction authority, privileged tools, or security enforcement warrant particular scrutiny because they can directly alter the consequences of agent behavior.

This leads to a risk-based revalidation model. Lower-risk changes may require regression testing and control verification. Changes affecting a defined capability may require targeted adversarial testing. Material changes to authority, data access, identity, transaction capabilities, security architecture, or critical dependencies may justify broader revalidation of the combined system.

The objective is not to create testing friction around every release. It is to know when the evidence supporting an earlier assurance conclusion is no longer sufficient.

Continuous monitoring provides the second component of this model.

Point-in-time testing can demonstrate how the system behaved under a defined configuration and test condition. Production monitoring is needed to determine whether that behavior remains consistent over time.

For an agentic system, monitoring should extend beyond application availability. Institutions should be able to identify material changes in tool use, denied actions, authentication failures, unusual data access, repeated guardrail violations, unexpected transaction behavior, security alerts, and human overrides. The purpose is not to retain every possible model artifact. It is to preserve sufficient evidence to reconstruct material activity and identify changes in risk.

Monitoring must also extend to external dependencies. Model providers can release new versions. APIs can change behavior. Security providers can modify controls. New subprocessors can be introduced. Continuous assurance therefore depends on both technical monitoring and the third-party change notification and oversight processes described earlier in this paper.

Operational resilience is equally important. A bank-grade AI environment must be designed not only for malicious activity, but also for ordinary failure.

Models can become unavailable. APIs can time out. Data sources can return incomplete information. Identity services can fail. Security controls can become unavailable. Third-party services can experience outages.

The expected response to those conditions should be defined in advance.

For lower-risk informational services, degradation may mean limiting functionality or returning the customer to a conventional service channel. For material financial actions, inability to verify identity, authorization, account state, authoritative data, or a critical control should prevent execution until the required condition is restored or an approved alternative process is invoked.

The governing principle is simple:

Failure should not expand authority.

A security service outage should not result in broader permissions. Failure to retrieve authoritative information should not permit the model to generate a substitute financial fact. Failure of an approval workflow should not convert a controlled action into autonomous execution.

Incident response should follow the same system-level view. An AI-related incident can originate in the model, application, identity layer, data source, tool, API, security control, third party, or

surrounding infrastructure. The response process should support detection, containment, revocation of access where necessary, evidence preservation, impact assessment, escalation, recovery, remediation, and determination of whether renewed testing is required.

Existing banking incident notification obligations continue to apply according to their established definitions. For example, the federal banking agencies' computer-security incident notification rule requires a covered banking organization to notify its primary federal regulator as soon as possible and no later than 36 hours after determining that an event meets the rule's definition of a notification incident. This does not mean every AI-related security event triggers regulatory notification. Applicability depends on the facts and the requirements of the rule.

AI risk should therefore be integrated into the institution's existing technology, cybersecurity, incident management, operational resilience, and governance processes rather than managed as a separate risk program.

Continuous assurance ultimately depends on maintaining an evidence trail across the lifecycle. The institution should be able to identify the configuration that was tested, the controls in place at that time, material changes made since testing, significant incidents or exceptions, remediation completed, and any targeted or broader revalidation subsequently performed.

The resulting assurance record should answer one executive question:

Is the system operating today materially the same system whose controls were previously evaluated?

If the answer is no, the institution should be able to identify what changed, what risk the change introduced, what testing was performed, and who accepted the resulting residual risk.

This closes the control lifecycle established throughout the paper. Authority is defined. Controls are implemented. The system is challenged. Weaknesses are remediated and retested. Monitoring, resilience, and change management then determine whether the resulting assurance remains valid.

The next section brings those elements together into a practical Bank-Grade AI Assurance Framework that a financial institution can use to evaluate whether an agentic AI capability is ready for approval and continued operation.

9. The Bank-Grade AI Assurance Framework

The preceding sections establish the elements of a bank-grade assurance process. The final step is to convert those elements into an evidence package that a financial institution can use to determine whether an agentic AI capability is sufficiently controlled for its intended use.

The framework is not intended to create a new certification standard or replace existing governance, cybersecurity, privacy, third-party risk, compliance, or audit processes. Its purpose is to connect those disciplines around one decision:

Can the institution demonstrate, with current evidence, that the AI system operates within the authority and risk boundaries established for its use?

Answering that question requires more than a model assessment, vendor questionnaire, penetration test, or policy document. The institution needs a concise body of evidence connecting the business purpose, delegated authority, system architecture, third-party dependencies, control testing, remediation, residual risk, and ongoing monitoring.

This paper refers to that evidence set as the **Agent Authority and Assurance Record**.

The record should provide a decision-ready view of the capability without requiring executives and control functions to reconstruct the system from multiple technical reports. It should establish what the AI is intended to do, what authority it has received, where its boundaries are enforced, how those boundaries have been challenged, and what risk remains.

For a financial institution, the assurance record should bring together seven core elements:

1. **Purpose and ownership.** The approved business purpose, accountable owner, intended users, material use cases, and operating scope.
2. **Authority and prohibited behavior.** The data, tools, workflows, transaction capabilities, permissions, and decisions available to the agent, together with activities that remain explicitly outside its authority.
3. **Architecture and dependency evidence.** The identity, orchestration, data sources, financial APIs, enforcement points, monitoring services, human approval points, and material third-party dependencies supporting the capability.
4. **Human authority.** The activities that may operate autonomously, those requiring confirmation or approval, those requiring additional authentication or dual control, and those prohibited from autonomous execution.
5. **Control validation evidence.** The testing methodology, material findings, supporting evidence, remediation actions, QA results, and applicable retest history.
6. **Residual risk and decision record.** Outstanding weaknesses, compensating controls, operating limitations, monitoring requirements, and the accountable function authorized to accept the remaining risk.

7. **Continuous assurance record.** The tested configuration, material changes since validation, significant incidents or exceptions, and any subsequent targeted or broader revalidation.

These elements create a common evidence base for business leadership, information security, risk, compliance, privacy, vendor management, internal audit, and other functions involved in approving or overseeing the capability.

The mCards, Radware, and SECAI initiative illustrates how that evidence can be assembled. The Partnership and Gold Standard materials define the intended authority and control architecture. The baseline establishes normal operating characteristics. The red team campaigns provide adversarial evidence across Tools Misuse, Prompt Injection, Data Access, and Data Leakage. The QA artifacts provide additional evidence regarding the completeness and consistency of the underlying findings, while the remediation roadmap connects identified weaknesses to planned and implemented control changes.

The assurance record should also preserve an important distinction established throughout this paper:

An implemented control is not necessarily a validated control.

A configuration change, new guardrail, security capability, or revised workflow establishes that remediation has occurred. It does not establish that the original weakness has been eliminated. Where retesting is incomplete or the underlying evidence has not yet been reconciled, that condition should remain visible.

This distinction is particularly important for executive and board-level decision making. Decision makers should be able to distinguish among controls that are designed, controls that are implemented, controls that have been independently challenged, and risks that remain unresolved.

The assurance record should also make the system's trust boundaries visible. The institution should be able to understand where customer identity is established, where the agent receives its authority, where data access is enforced, where financial actions are authorized, where third parties materially influence the environment, where human approval is required, and where material activity is recorded.

That visibility allows the bank to evaluate the capability as one governed system rather than as a collection of individual technologies and vendors.

The same record should support ongoing oversight after deployment. Because assurance applies to a defined configuration, the institution must be able to determine whether changes to the model, permissions, data sources, tools, APIs, third parties, security controls, or workflow materially alter the system that was originally approved.

Before approving an agentic AI capability, an institution should therefore be able to demonstrate three things with evidence:

- **What authority has been delegated, and where are the boundaries enforced?**
- **How has the system been tested, what weaknesses were identified, and what has been remediated or remains as residual risk?**
- **Is the assurance still valid for the system operating today, including its current model, permissions, dependencies, and controls?**

If those questions cannot be answered with current evidence, the institution does not yet have a complete assurance position.

The practical value of the framework is that it converts AI governance from a collection of policies, technical reports, and vendor artifacts into a decision-ready assurance package. It gives the institution a common basis for determining whether the capability is appropriate to deploy, what conditions should apply to its operation, what additional remediation is required, and when renewed testing is necessary.

This is the broader result of the mCards, Radware, and SECAI initiative.

The objective is not simply to produce a secure AI application. It is to produce evidence that a financial institution can use to understand the capability, govern its authority, challenge its controls, evaluate residual risk, and maintain that assurance as the system changes.

That evidence provides the basis for the final conclusion of this paper.

10. Conclusion: Demonstrable Control

Agentic AI creates a different control problem for financial institutions because it can move beyond producing information and begin interacting with data, tools, workflows, and financial systems.

The central question is therefore not whether the model can perform a task. It is whether the institution can demonstrate that the complete system performs that task within defined authority, security, privacy, operational, and human control boundaries.

The mCards, Radware, and SECAI initiative was designed around that principle. The work treated the AI agent as part of a regulated operating environment rather than as an isolated model. Governance, identity, permissions, authoritative data, tool access, transaction controls, runtime security, monitoring, human intervention, third-party dependencies, adversarial testing, remediation, and retesting were considered as parts of the same assurance problem.

The testing reinforced why that broader view matters. Normal operation does not establish resilience under hostile conditions, and well-designed instructions do not guarantee that critical boundaries will remain intact when the model is manipulated, uncertain, or operating in an unexpected context.

For that reason, critical controls must exist outside the model.

The AI should understand the rule. The institution should enforce the rule.

The same principle applies to authority. A model may interpret a request, recommend an action, or determine that a particular tool could help complete a task. Those conclusions should not create identity, permissions, transaction authority, or the ability to bypass an established control.

A probabilistic model output should not silently become deterministic authority.

Human oversight must also be deliberate. Not every interaction requires manual approval, but institutions should define where autonomy is acceptable, where additional authentication or approval is required, and where the agent should have no authority at all. Accountable humans remain responsible for establishing those boundaries, challenging their effectiveness, and determining whether residual risk remains acceptable.

Third-party dependencies must be governed through the same system-level perspective. Models, APIs, cloud services, data providers, security platforms, identity services, and other providers can materially influence what an agent knows, what it can do, and whether the institution can observe and control its behavior. Effective due diligence therefore requires understanding the combined technology and authority chain, not simply evaluating individual vendors in isolation.

Adversarial validation provides the bridge between intended control and demonstrated control. Its value is not the number of prompts executed or weaknesses identified. Its value is the

evidence it creates about where controls failed, why those failures mattered, what was changed, what was successfully retested, and what risk remains.

That evidence must also remain current.

An AI system can change materially when its model, permissions, tools, data sources, APIs, security controls, workflows, or third-party dependencies change. Assurance should therefore attach to a defined configuration and be reconsidered when material changes occur.

This changes the approval question for financial institutions.

The question is not whether an AI product has been tested at some point in its history.

The question is whether the system operating today remains within the authority, architecture, control, and risk boundaries the institution evaluated and approved.

The Bank-Grade AI Assurance Framework described in this paper provides a practical way to answer that question. It connects business purpose, governance, agent authority, architecture, third-party risk, human control, adversarial validation, remediation, residual risk, monitoring, and change management into a common evidence base.

The objective is not to claim that agentic AI can be made permanently secure or universally compliant. It is to establish a more defensible standard: that the institution can understand the system, constrain its authority, challenge its controls, identify and remediate weaknesses, evaluate residual risk, and maintain assurance as the environment changes.

For financial institutions evaluating agentic AI, that is the standard that matters.

The appropriate standard is not trust in the model. It is demonstrable control over the system.

Bank-grade AI is established by evidence that its purpose is defined, its authority is bounded, its prohibited behavior is explicit, its critical controls are independently enforced, its information is grounded, its dependencies are governed, its actions are observable, its controls withstand challenge, and accountable humans remain in command.