OvationCXM

2026

BUILD THE FOUNDATION FOR GOVERNED AI

A field guide to the foundational building blocks required to deploy applied AI across business and commercial banking



www.ovationcxm.ai

Build the foundation before expanding autonomy

The experimentation phase of AI in banking is evolving into deployment. Banks have spent the past several years testing AI, but scaling pilots into measurable value has been harder. An Ernst & Young-sponsored "2025 MIT Technology Review Insights" survey found that 52% of banks had piloted agentic AI, while only 16% had fully deployed use cases.[1] McKinsey describes the same gap as "pilot purgatory."[2]

The question is shifting from "What can we do with AI?" to "How do we put AI to work across the enterprise and generate meaningful ROI?"

That means embedding AI into real workflows, not layering assistants onto fragmented processes.

That shift changes the AI architecture, with the model no longer at the center. The AI application layer is now crucial because it binds models to the bank's context, tools, workflows and controls. The bank still decides which models can be used, how they may be adapted and how the value created by its data and workflows remains with the institution.

This guide outlines eight foundational building blocks for bank-provided AI across commercial banking middle- and back-office workflows. The final section shows how a common harness connects them. Client-owned AI agents entering the bank create a separate front-office control boundary.

Who Is This For?

- Heads of Commercial and Corporate Banking, Enterprise Payments and Treasury Management
- Heads of Servicing, Operations, Lending and Onboarding
- Product, Digital Transformation, Technology and Enterprise Architecture leaders
- AI, Risk, Compliance and Model Governance teams

Table of Contents

03 Building the enterprise knowledge foundation

04 Unifying the context AI needs to understand the customer

06 Making bank capabilities agent-ready

08 Defining the boundaries before granting autonomy

09 Separating the AI experience from the model

10 Making every AI interaction observable and auditable

11 Applying AI to workflows, not isolated features

12 Turning AI activity into compliance evidence

13 How OvationCXM connects the building blocks for governed AI

AI value increases with agency. So does the need for control.

As banks move into production, the value shifts from isolated productivity gains to end-to-end journeys. Accenture and McKinsey point toward conversational and agentic experiences that redesign work, while Deloitte emphasizes the AI-ready data required to support them. [2,3,4]

The message is clear: as AI moves closer to decisions and execution, the value opportunity grows, but so does the need for stronger context, controls and governance. We refer to this progression as:

The Spectrum of Automation



This is not a maturity ladder. Some workflows should remain assistive by design. Each step to the right increases what AI can do and the importance of the foundation it is built on.

Three principles run through the entire guide

1. **Context before intelligence.** AI needs the right institutional, customer and operational context to make a useful decision.
2. **Control before autonomy.** More agency requires explicit permissions, workflow boundaries, approvals and escalation.
3. **Governance must produce evidence.** The institution must be able to reconstruct what AI saw, decided, called and changed.

KEY TAKEAWAY

More agency requires more control. Governed Autonomy is achieved when AI operates with trusted context, controlled access, explicit boundaries and continuous governance.

Building the enterprise knowledge foundation

Large language models do not inherently know how a bank sells, implements, services or governs its products. That knowledge is fragmented across policies, product and partner documentation, internal repositories and, often, tribal knowledge held by experienced employees.

That is manageable when a tenured employee knows where to look. It becomes a production risk when AI must answer, recommend or act. The knowledge layer must ground AI in authoritative, current and permissioned information at the moment of use.[5]

> ***If the answer still depends on knowing who to call, that knowledge is not yet AI-ready.***

RAG is a mechanism, not the knowledge strategy

Retrieval-augmented generation, or RAG, retrieves relevant private content and places it into the model's working context. It helps AI use current bank information without retraining the underlying model.

RAG does not decide which source is authoritative, which version is current, who may use it or what happens when approved sources conflict.

Larger context windows do not solve those questions. More text can add cost and noise without adding authority. In commercial banking, the answer may also span both bank-owned and partner content.

What the knowledge layer should do

- Ingest approved bank and partner sources with clear ownership, version validation and approval metadata.
- Retrieve both plain-language and exact banking terms, while enforcing role and usage policy before content reaches the model.
- Return source attribution, abstain when approved evidence is missing and feed repeated misses into a human-approved knowledge update process.

Knowledge is an operating process. Product, operations, compliance or another designated owner remains accountable for what is official and which source takes precedence. AI should not become the authority that decides what is true. Recent guidance from Google Cloud and Microsoft reaches the same conclusion: trustworthy enterprise agents depend on authoritative business context and permission-aware grounding in the institution's own data.[5,6] Knowledge graphs help when answers depend on explicit relationships, but most first deployments do not need one.

ACTIONS TO GET STARTED

- Map one high-value journey and every knowledge source it depends on, including owner, approval status, freshness and access.
- Build a small evaluation set of real questions. Flag any answer that still depends on "ask the person who knows," and test retrieval before evaluating the model's prose.

KEY TAKEAWAY

AI should not determine what the bank knows. The bank must define what is authoritative, current, permitted and appropriate for AI to use.

Unifying the context AI needs to understand the customer

Knowledge tells AI how the bank operates. Customer and operational context tells AI what is happening right now.

In commercial banking, that context rarely lives in one system. CRM may know the relationship, a core or processor knows products and balances, onboarding knows implementation status, servicing knows the open issue and a partner may own the activity blocking the customer.

None of those systems has to be wrong for the bank to give a bad answer. The operational truth is distributed across the bank and its ecosystem, which is why Deloitte emphasizes data that is accurate, timely, broad and securely governed.[4]

If every system is right and the bank still cannot tell the customer what is going on, the context is not yet AI-ready.

This is not another MDM project

The goal is not to create another golden source. Systems of record should remain authoritative for the data they own.

The context layer has a different job: associate records across customers, accounts, products, cases, journeys and partner identifiers; preserve provenance; and retrieve current data from the system that owns it. Where mappings are uncertain, the system should preserve that uncertainty rather than let AI infer the relationship.

Resolve the relationship. Preserve the source.

Use the right context for the moment

1. **Indexed context**
   Relatively stable history (e.g., prior communications or closed cases).
2. **Live context**
   Values that come from the authoritative system at the moment of use (e.g., balances, entitlements or current status).
3. **Event context**
   Recent changes that alter the customer story (e.g., a failed file, new case or completed approval).

Not every piece of context has the same shelf life. Yesterday's product documentation may still be useful. Yesterday's balance may not be.

Across all three, preserve control

Preserve where each fact came from, how current it is, who may use it and the sequence in which events occurred. Give the model only the context required for the task.

Why ecosystem context matters

Commercial banking is often delivered across third parties, processors and outsourced servicing partners. The bank may own the relationship while critical onboarding, servicing or transaction activity lives with FIS, Fiserv, DXC or another provider.

AI that sees only the bank's internal systems can be confidently incomplete. The context layer must account for where the work actually happens, not just where the customer record is stored.

An anonymized example

A client asks why an implementation is delayed. CRM shows the sale is closed; onboarding shows active work; a processor shows a failed test; a partner ticket remains open; email shows the client already supplied the file.

Every system is correct. Unified context turns those fragments into the actual customer story.



ACTIONS TO GET STARTED

- Map one journey across every internal and third-party system, then identify the authoritative source and required freshness for each critical fact.
- Document how records are associated across systems and define access, confidence and model/provider usage rules before AI joins them.

KEY TAKEAWAY

AI does not need another system of record. It needs a governed way to assemble the right context from the multiple systems and partners where the relationship actually lives.

Making bank capabilities agent-ready

Context allows AI to understand. **Agent-ready access allows AI to interact.**

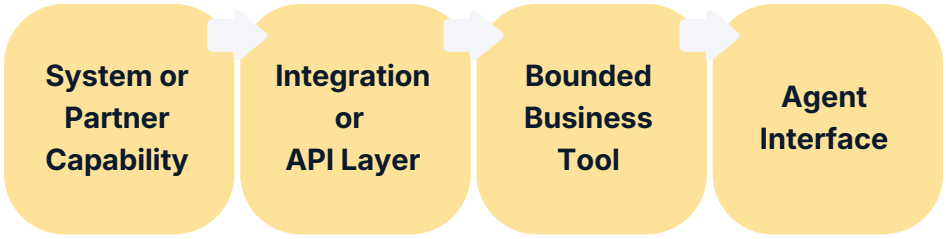
API coverage remains uneven across commercial banking. Modern APIs coexist with vendor connectors, file exchanges, legacy services and human-operated portals.

> ***AI does not eliminate that fragmentation. It exposes it.***

If a capability cannot be reliably accessed by a machine, it is not ready to become an AI tool. The target is not an API for everything; it is a reliable, governed machine interface for every capability AI may use.

APIs and MCP solve different problems

An API defines how a machine interacts with a system. MCP standardizes how approved resources and tools are presented to AI. MCP sits on top of the integration layer; it does not make a legacy system programmable or replace the bank's API gateway.[7]



Where no strong API exists, the bank may need an adapter, connector, file service or intermediary before a capability can become an agent tool.

Instead of exposing a broad processor API, publish bounded tasks such as get payment status, retrieve a rejection reason or open a servicing case.

Give the agent a business task, not the breadth of the underlying integration layer.

Not every capability is ready for AI

- **Ready:** a secure API or service already supports a bounded tool.
- **Needs work:** a connector, middleware layer, file interface or controlled bridge can make the capability machine-accessible.
- **Human-only:** the task still depends on a person operating a portal or coordinating across systems.

The first group can move quickly. The second creates an integration backlog. The third should remain outside greater agency until a reliable interface exists; for higher-risk actions, UI automation should not be the control boundary.

Bound the tool and enforce policy outside the model

An agent should not receive the full surface area of a backend API. Bound each tool by identity, customer or account scope, permitted inputs, read/write authority, limits and required approvals.

This is the difference between giving AI access to a system and giving AI permission to perform a task.

MCP can tell an agent which tools exist and how to call them. Deterministic controls should decide whether the call is allowed, denied or requires approval before execution.[7,8]

Start with read before write

Start with narrow, read-only capabilities that can be checked against the authoritative system. Get payment status is a different risk profile from release payment.

As the bank moves into write actions, tools should validate inputs, prevent unintended duplicate actions and return a clear execution result.

A denied request is evidence that the control worked, not noise to discard.



ACTIONS TO GET STARTED

- Choose one high-value, read-only task and document how it is accessed today. Reduce an existing machine interface to the minimum inputs, outputs and permissions AI needs or define the integration work required first.
- Define identity, entitlement and account scope, then test both permitted and denied paths before expanding to write actions.



KEY TAKEAWAY

AI readiness depends on integration readiness. Banks do not need an API for everything, but every capability AI is allowed to use needs a reliable, governed machine interface.

Defining the boundaries before granting autonomy

The design choice is not how clever the model is. It is what AI may reason about and what must remain deterministic. A model can interpret a request and propose an action; it should not be the final authority on whether a high-impact action is permitted.

Governed Autonomy means AI operates inside a defined envelope while identity, policy, journey state, approvals and system validations remain enforceable outside the model. Deloitte makes the same sequencing point: build controls into the operating model, not after deployment.[9]

A practical control pattern: before, during and after

- **Before the action:** validate the actor, customer, account, tool, parameters, limits and required approvals.
- **During execution:** enforce valid workflow state, handle retries and timeouts, prevent duplicates and stop or escalate when execution departs from the expected path.
- **After the response:** confirm what the downstream system actually did and trigger remediation when the outcome differs from the intent.

Guardrails belong outside the prompt

Prompts are guidance, not enforcement. A prompt that says “do not send a wire above $25,000” is instruction. A policy that blocks the call is a control. Anything protecting money, customer data or regulated decisions belongs in logic the model cannot talk around.

Any capability that moves money, changes beneficiaries or alters entitlements should inherit existing treasury limits, approvals and dual-control requirements, with deny-by-default when required authority is missing. Human-in-the-loop means a person must approve before completion. Human-on-the-loop means AI can proceed within defined limits while people monitor exceptions. Autonomy should increase by workflow only as controls and confidence are proven.[10]



ACTIONS TO GET STARTED

- Choose one workflow, separate judgment from deterministic control and define the required approvals, exception owner and emergency stop path.
- Test the happy path and key failures: denied policy, missing data, duplicate action, timeout and partial completion.



KEY TAKEAWAY

Autonomy is safest when AI has freedom inside clearly defined boundaries, not freedom from boundaries.

Separating the AI experience from the model

Models will keep changing. The workflow, context, tools and controls around them should not have to change with each provider or release.

The harness is the shared application layer between approved models and the business experience. It lets the bank change models and agent configurations without rebuilding the context, workflow, controls or customer experience around them.

Do not rebuild the operating layer for every use case

Approved model catalog and constrained routing:
Limit each task to models approved for its data class and risk tier.

Versioned prompts, agent specifications and promotion gates:
Know what is live and test changes before production.

Fallback and resilience:
Switch to another approved model, degrade to a deterministic path or stop.

Section 4 decides what is allowed. The harness manages which approved model and agent version performs the work.

Buying model access is not the same as owning the outcome. The bank should control model choice, how prompts, retrieval or fine-tuning are versioned, which data classes each provider may use and whether differentiating customer or workflow data may improve a provider's model. Hosted and BYOM approaches can both work if those rules stay with the bank.

KEY TAKEAWAY

The model should be replaceable. The bank's controls, context, workflow and customer experience should not be, and the value they create should stay with the bank.

Routing and evaluation are operating disciplines

Within the approved catalog, routing can optimize for quality, latency, cost and availability. Test against representative workloads rather than assuming the largest model is best. Changes to models, instructions, retrieval or tool schemas should pass evaluation gates before promotion and be monitored after deployment.[11,12]

ACTIONS TO GET STARTED

- Inventory approved models and providers, then create a common agent specification covering owner, purpose, instructions, context, tools, risk tier and evaluation.
- Define promotion gates and test one fallback scenario before production.

Making every AI interaction observable and auditable

Agentic AI creates a runtime governance problem: the system can retrieve different context, choose different tools and take different paths depending on the request. Governance therefore has to operate during execution, not only at design time.

The control plane is the runtime layer that knows what is deployed, records policy and execution decisions and preserves enough information to reconstruct what happened. Industry guidance increasingly points toward real-time oversight, identity, permission checks and durable activity records as AI scales.[3,13,14]

Traceability, observability and auditability are related, but not identical

- **Traceability:** what happened in an interaction, under whose authority, with which context, tools and outcome.
- **Observability:** how the system behaves over time, including completion, errors, tool use, latency and policy denials.
- **Auditability:** whether an independent reviewer can reconstruct the activity from durable, accessible records.

The control plane should know what is deployed

The bank needs a production inventory of AI capabilities with a named owner, purpose, risk tier, provider, connected sources, tools, permissions and current version. A day-to-day AI or AgentOps function should keep that inventory aligned to what is actually running.

A blocked call should remain visible. It is governance evidence, not noise.

Regulatory context is still evolving

Revised U.S. model risk guidance explicitly excludes generative and agentic AI from scope while directing banks to broader governance; the U.S. Department of the Treasury's 2026 "Financial Services AI Risk Management Framework" provides a complementary sector framework.[15,16]

The implication is risk-based governance connected to existing enterprise controls. A meeting summarizer and a beneficiary-change agent should not face identical requirements.



ACTIONS TO GET STARTED

- Create a production inventory and standard runtime record covering ownership, authority, permissions, calls or denials and downstream outcomes.
- Prove that risk or compliance can reconstruct an allowed action, a blocked action and a human override without relying on engineering.



KEY TAKEAWAY

You cannot govern an AI system you cannot see and you cannot audit an action you cannot reconstruct.

Applying AI to workflows, not isolated features

AI features do not create material value if the workflow stays the same. McKinsey argues that the larger value pool requires banks to rewire work and decisions end-to-end, not simply layer AI onto existing processes.[2]

Prioritize workflows using five questions

Volume	Friction	Context Readiness	Actionability	Risk & Reversibility
Does the workflow occur often enough for automation to matter?	How much time is spent searching, reconciling, re-keying or coordinating handoffs?	Can AI reliably access the knowledge and customer context required?	Can required systems be reached through reliable machine interfaces?	What happens if AI is wrong and can the action be reviewed or contained?

A servicing workflow across the Spectrum of Automation

A servicing request may begin with a summary and drafted response, then use agent-ready tools to retrieve live status or open a case. With governed orchestration, AI can run a bounded follow-up and escalate exceptions while stopping before actions that require separate approval.

If the knowledge is still tribal or the action still lives only in a portal, the workflow is not ready for greater agency.

Start with a workflow that can prove value and control

Define the baseline before development: cycle time, implementation days to live, repeat status contacts, exception aging, rework or SLA misses. Set the maximum allowed autonomy at launch.

A successful pilot is not an agent that "works." It is a workflow that improves a measurable business or customer outcome while proving the controls worked too.

ACTIONS TO GET STARTED

- Score three high-friction workflows, then select one with a clear owner, measurable baseline and maximum allowed autonomy.
- Measure value and control together; expand only after the bank can explain the gains, denials, overrides, errors and exceptions.

KEY TAKEAWAY

Do not deploy AI simply because a feature is possible. Deploy it where grounded intelligence and governed action can materially improve an end-to-end workflow.

Turning AI activity into compliance evidence

Governance defines how AI is supposed to operate. Evidence shows how it actually operated. Deloitte argues that auditability should be designed into the operating model rather than reconstructed after deployment.[9]

The runtime record from Section 6 should become evidence a controller, compliance officer or auditor can use months later without relying on engineering.

Evidence should be a byproduct of execution

For each material interaction, the evidence should show four things: who acted and under whose authority; what the system was allowed to see and do; what was executed, denied, approved or failed; and what the downstream system confirmed.

Generate that evidence as the workflow runs. Retrospective screenshots and narratives are weaker than records produced by the controls governing execution.

Evidence must be usable outside the AI platform

Different stakeholders may need different views, but all should use the same factual record. Evidence should be searchable, retainable and independently exportable, with access, retention, legal hold and tamper resistance defined by risk tier.

Reporting must not overstate compliance

A dashboard does not make an AI system compliant. Compliance depends on applicable laws, policies, use case and control environment; the evidence layer provides the factual record. Compliance evidence and reporting requirements should be validated against the institution's policies, regulatory obligations and control environment.

ACTIONS TO GET STARTED

- For one workflow, define what a controller, compliance officer or auditor must be able to answer, then generate a sample package with an allowed action, denied action, human approval and failed downstream call.
- Have the second line or internal audit review the package as if the event occurred six months ago, then close gaps in retention, access, tamper resistance, legal hold and independent export.

KEY TAKEAWAY

Governance tells AI how to operate. Evidence proves how it operated.



The building blocks for governed AI

How OvationCXM connects intelligence, context, workflows and controls for commercial banking

The governed harness

Banks should continue to build where they differentiate: proprietary products, policies, strategic interfaces, customer experiences and data. The challenge is assembling all of it into one consistent layer across every AI experience.

OvationCXM is designed to serve as that harness. CXMEngine is the system of action where customer context, journeys, integrations and controls meet. AI Studio embeds models and agents into that operating layer so bank-provided AI can use the same context and orchestration instead of creating a parallel stack.

How the building blocks map to CXMEngine®

- **Knowledge & Context:** Govern enterprise knowledge, close gaps through human-approved feedback and make role-aware context usable by employees and AI.
- **Customer, Operational Context & Integration:** Unify relationship activity across systems of record and partners, then connect APIs, files and services into bounded AI capabilities.
- **Journeys:** Define the stages, tasks, approvals, policies and exceptions that form the operating envelope for AI reasoning and actions.
- **AI Studio:** Provide the agent harness for approved models, instructions, data bindings, evaluation, and multi-model operation, including BYOM where appropriate.
- **Applied AI & Governance:** Apply shared context and orchestration across the Spectrum of Automation while connecting runtime activity to governance. Compliance Evidence & Reporting is an emerging capability.

Build what differentiates. Buy the harness.

The bank should own the assets that create differentiation: its products, data, policies, journeys and customer relationships. The harness lets those assets use changing models without surrendering control of model choice, adaptation rules or differentiating operating data. Model usage is a cost; the business value should remain with the bank.

For external, client-owned agents, OvationCXM's Agentic Transaction Control Layer (ATCL) extends the same principle to the bank's front-office boundary by governing agent identity, permissions, routing, approvals, revocation and audit while the bank retains execution.



KEY TAKEAWAY

The model provides intelligence. The harness connects that intelligence to governed banking context, workflows and controls.

References

Sources published in 2025-2026 were used wherever possible. Older sources are retained only when they remain the best or primary authority for a specific concept.

1. EY. (2025, September 23). Eight ways banks can move AI from pilot to performance. https://www.ey.com/en_us/insights/banking-capital-markets/eight-ways-banks-can-move-ai-from-pilot-to-performance

2. McKinsey & Company. (2026, February 27). The paradigm shift: How agentic AI is redefining banking operations. https://www.mckinsey.com/capabilities/operations/our-insights/the-paradigm-shift-how-agentic-ai-is-redefining-banking-operations

3. Accenture. (2026). Top Banking Trends for 2026. https://www.accenture.com/en/insights/banking/accenture-banking-trends-2026

4. Deloitte. (2025, October 30). 2026 banking and capital markets outlook. https://www.deloitte.com/us/en/insights/industry/financial-services/financial-services-industry-outlooks/banking-industry-outlook.html

5. Microsoft Learn. (2026). What is Foundry IQ? https://learn.microsoft.com/en-us/azure/foundry/agents/concepts/what-is-foundry-iq

6. Google Cloud. (2026, July 23). Your AI agents are ready. Is your data? https://cloud.google.com/blog/topics/ai-infrastructure/state-of-ai-infrastructure-report-and-the-agentic-data-cloud

7. Model Context Protocol. (2026, July 28). The 2026-07-28 Specification. https://blog.modelcontextprotocol.io/posts/2026-07-28/

8. Microsoft. (2026, April 22). Securing MCP: A Control Plane for Agent Tool Execution. https://developer.microsoft.com/blog/securing-mcp-a-control-plane-for-agent-tool-execution/

9. Deloitte. (2026, August 31). Agentic AI in Banking: Building the Auditable Bank. https://www.deloitte.com/uk/en/Industries/financial-services/blogs/agentic-ai-in-banking-building-the-auditable-bank.html

10. AWS. (2026, August 26). Closing the AI agent trust gap with graduated autonomy. https://aws.amazon.com/blogs/architecture/closing-the-ai-agent-trust-gap-with-graduated-autonomy/

11. Microsoft. (2026, August 20). Evaluate model router for your workload. https://learn.microsoft.com/en-us/azure/foundry/openai/how-to/evaluate-model-router

12. Microsoft. (2026, June 2). Build agents you can trust across any framework with open evals and a control standard. https://devblogs.microsoft.com/foundry/build-2026-open-trust-stack-ai-agents/

13. IBM. (2026, July 2). Agentic Control Plane in IBM watsonx Orchestrate: One place to control every AI agent. https://www.ibm.com/new/announcements/introducing-the-agentic-control-plane

14. Deloitte. (2026, March 5). Managing the new wave of risks from AI agents in banking. https://www.deloitte.com/us/en/insights/industry/financial-services/agentic-ai-risks-banking.html

15. Federal Reserve. (2026, April 17). SR 26-2: Revised Guidance on Model Risk Management. https://www.federalreserve.gov/supervisionreg/srletters/SR2602.htm

16. U.S. Department of the Treasury. (2026, February 19). Treasury Releases Two New Resources to Guide AI Use in the Financial Sector. https://home.treasury.gov/news/press-releases/sb0401

OvationCXM

The orchestration system for agentic banking

Schedule a briefing at www.ovationcxm.ai

OvationCXM provides the orchestration operating system for financial services. Its CXMEngine platform connects systems, data, teams and ecosystem partners to coordinate complex customer and operational workflows without requiring institutions to replace existing technology. OvationCXM's Agentic Transaction Control Layer (ATCL) extends that orchestration to client-facing AI agents, enabling institutions to register, authenticate and permission incoming agents and coordinate their approved activity across systems, workflows, partners and human approvals. Learn more about OvationCXM at www.ovationcxm.ai.