

Govern financial agents through meaning, not prompts alone

Prompt controls are too narrow for agents that cross systems, interpret policy and produce consequential financial work.

The control problem changes when software can act

A chatbot answers inside one conversation. An agent can assemble context, call tools, delegate work, update analysis and publish an output across several systems. Its risk surface includes identity, memory, tools, data movement, delegation, approvals and the meaning applied to the task.

Prompt guardrails address only part of that surface. They may tell an agent not to exceed authority, but they do not define the institution's authority structure. They may request citations, but they do not identify which evidence is acceptable for a particular decision. They may specify a calculation, but they do not carry the approved definition, scope and exception logic behind it.

Financial institutions therefore need a control layer that remains outside any single prompt or model. The layer must describe what the firm means, what the task is allowed to do and how the result will be evaluated.

Ontology can become part of the policy plane

A governed ontology represents the concepts, relationships, policies and precedents that agents must use. It can connect those objects to owners, evidence requirements, permissions and effective dates. That makes it part of the policy plane for agentic execution rather than a passive knowledge graph.

When an agent receives a task, the system can resolve the relevant concepts and retrieve only the approved context. The workflow can determine which tools are permitted, which calculations are authoritative and which conditions require escalation. The output can retain provenance linking its claims to sources and the institutional meaning applied.

Meaning: approved concepts, calculations and relationships.

Evidence: source requirements and provenance for claims and figures.

Authority: who may view, change, approve or publish.

Behavior: tools, actions, delegation and prohibited operations.

Evaluation: competency questions, golden answers and policy tests.

Give every agent a bounded role

A dynamic financial workflow may need a researcher, policy interpreter, modeler, critic and presentation specialist. Those roles should not receive identical context or authority. The researcher may collect evidence but not approve a conclusion. The modeler may execute an approved calculation but not redefine the measure. The critic may challenge source coverage without gaining access to unrelated confidential material.

Role boundaries should be explicit and inspectable. They include the identity under which the agent acts, the sources it can access, the tools it can call, the data it can retain, the agents it can delegate to and the events that require a person. This reduces the chance that an apparently helpful workflow quietly accumulates excessive privilege.

Human review should also be specific. "Human in the loop" is not a control unless the institution knows which human, reviewing what evidence, against which standard and with what authority to approve, amend or stop the work.

Govern the interfaces, not only the native workspace

Institutional work will occur across many surfaces: internal applications, enterprise copilots, developer environments, messaging channels and third-party agent frameworks. API, MCP and A2A interfaces allow capabilities to move across those surfaces, but they also create new boundaries that must be governed.

An API contract should specify schemas, identity, permitted operations, versioning and evidence returned. An MCP surface should expose only the context and tools approved for the connected harness. An A2A workflow should define how agents discover one another, what task information can be delegated and how results carry identity and provenance back to the orchestrator.

The ontology provides continuity across these interfaces. It prevents each channel from inventing its own definition of the firm's concepts and gives the institution one place to govern changes in meaning even as execution technology evolves.

Evaluate the decision path, not only the prose

A fluent answer can still be institutionally wrong. Agent evaluation should inspect whether the workflow selected the correct definition, gathered required evidence, applied the approved calculation, respected permissions, escalated the right exception and represented uncertainty honestly.

Competency questions and golden answers provide a test set grounded in the domain. Source-coverage checks can detect unsupported claims. Constraint tests can identify prohibited actions. Adversarial cases can probe boundary conditions and conflicts between policies. These evaluations should run when the ontology, agent logic, integration or model provider changes.

Failures should improve the right layer. A missing concept belongs in ontology stewardship. A wrong tool call belongs in execution policy. A poor synthesis may belong in the model or prompt. Separating these causes prevents endless prompt tuning around structural defects.

Operate meaning as a living control

Financial meaning changes. Policies are revised, products launch, regulations shift, committees establish new precedent and source systems are replaced. The ontology therefore needs ownership, review cadence, versioning, release management and impact analysis.

A governed change should show who proposed it, who approved it, which competency questions were rerun and which workflows or prior outputs may be affected. High-impact changes may require staged release or dual operation. Retired definitions may need to remain available for historical interpretation even when new work uses the current version.

This operating model can be client-run, Penomic-managed or shared with a delivery partner. What matters is that responsibility remains explicit and that ontology operations are treated as part of production, not as a one-time modeling exercise.

Design for the harness after the current harness

No institution can select a permanent model, copilot or agent framework today. New execution environments will emerge, and existing ones will change their interfaces and economics. Hard-coding institutional meaning into each harness creates repeated migration work and inconsistent controls.

Separating governed business meaning from any single model lets the Penomic deployment evolve its model mix while preserving definitions, provenance, policy and evaluation history. The same governed context can serve Alex, Penomic applications and approved third-party harnesses through controlled interfaces.

That is the architectural purpose of knowledge-first agent governance: not to predict which agent framework wins, but to ensure the institution's meaning and authority remain inspectable and controlled in Knowledge Architect whichever approved interface invokes Penomic.

Bring us one consequential decision.

Penomic makes a financial institution's definitions, policy, precedent and judgment usable by people, models and agents, with provenance on every output.

Request a private briefing at penomic.ai/contact · More research at penomic.ai/insights