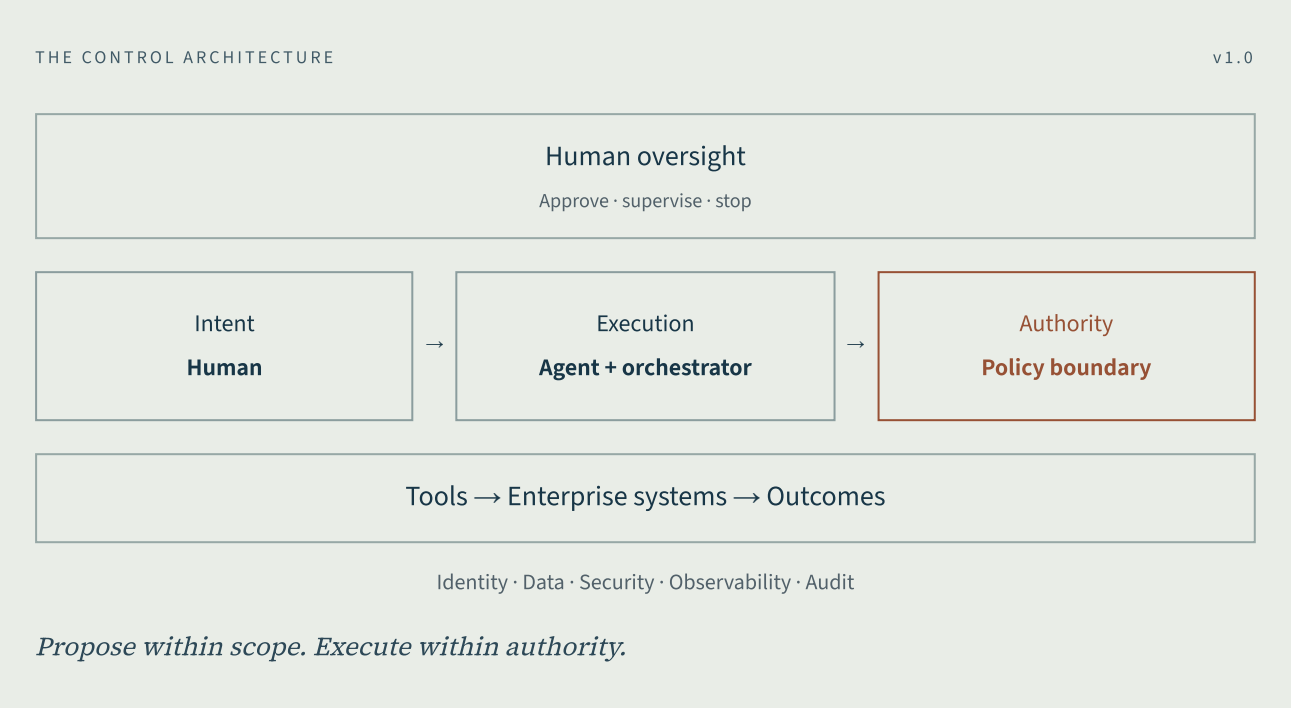
ARCHITECTURE / VERSION 1.0

Reference Architecture for Agentic Systems in Regulated Environments

A control-oriented architecture for delegated work: bounded authority, accountable human oversight, observable execution and recoverable enterprise actions.

By Balaji Krishnan Published September 6, 2026 Updated 2026-09-06 6 min read



Abstract

An agent's instructions are only one part of its control system. This reference architecture puts identity, authorization, approval, execution and evidence outside the model's discretion. It describes a vendor-neutral pattern for enterprise workflows in which a language model proposes or selects actions while deterministic services decide whether those actions may execute.

The architecture is a conceptual design by Balaji Krishnan. It is not a certified product, a regulatory checklist or a claim of authorization. Organizations must map it to their own legal obligations, system boundaries and assurance processes.

System boundary and trust model

Define the business workflow before drawing the agent boundary. Identify users, data owners, system owners, human reviewers and the operations team. Record which systems are authoritative for identity, records and transactions. Retrieved documents, user messages and tool outputs are data; they cannot grant permissions or change approval policy.

A request enters through an authenticated interface. The orchestrator associates it with a scoped work item and a correlation identifier. The model selects a proposed action using permitted context. A policy-enforcement service evaluates that proposal against the initiating user's authority, the agent's identity, the workflow state and the requested resource. Only then can a tool adapter invoke an enterprise API.

The execution plane

Component	Responsibility	Boundary to enforce
Human interface	Capture intent and present decisions requiring review	Authenticate users; make pending effects visible
Orchestrator	Manage work state, retries, budgets and stop conditions	Persist state; limit steps, time and spend
Model gateway	Select approved models and apply data routing rules	Restrict providers and destinations; track model versions
Context service	Retrieve authorized, relevant information	Enforce resource-level access before retrieval
Policy enforcement	Evaluate proposed actions	Deny by default; do not accept model-supplied approval
Tool adapters	Execute typed, validated operations	Use narrow interfaces and scoped credentials

Component	Responsibility	Boundary to enforce
Enterprise systems	Maintain authoritative records and transaction rules	Recheck authorization and business invariants

Prefer durable work identifiers and explicit state transitions. A retried request must not become a second payment or a duplicate case update. Where the enterprise API supports idempotency keys, derive one from the authorized business operation. Where it does not, add reconciliation and a human exception path before permitting retries.

Identity and authorization

Give an agent or workload a distinct identity. Avoid shared administrator tokens. Record whether an action executes on behalf of a user or under a service identity, and ensure the effective permission cannot exceed the approved delegation. Credentials should be short-lived where the platform supports it and unavailable to the model as text.

Separate read, propose, approve and execute privileges. The actor requesting an action should not be able to manufacture its own approval. At execution time, verify that the approval covers the exact resource, parameters and validity window. If any of those change, request a new decision.

Human oversight patterns

Human in the loop (HITL) means the action waits for a person before a defined consequential step. The review screen should show the proposed effect, relevant evidence, uncertainty and an explicit reject path. A summary generated by the same agent is insufficient as the only basis for approval.

Human on the loop (HOTL) means a person supervises execution and can intervene under defined conditions. This is useful only when detection and intervention are fast enough relative to the possible harm. Measure reviewer capacity and response latency rather than assuming an available dashboard is effective oversight.

Choose oversight per action class. Read-only retrieval and an irreversible financial instruction should not inherit the same review policy because they happen in the same conversation. Test absence, overload and disagreement among reviewers.

Governance and assurance plane

Maintain separate but linked records for model governance and agent governance. Model review concerns the model and its use context. Agent review also covers instructions, tool permissions, action sequences, memory, escalation, integrations and changes in delegated authority.

Keep an agent register with owner, purpose, environment, data classes, model versions, tools, approval tier and retirement criteria. A model substitution, new tool or expanded data source may change the risk of the whole workflow. Evaluate the changed system before expanding authority.

The NIST AI RMF is a voluntary source for structuring risk management. This architecture uses it as background, without claiming conformity. OWASP's excessive-agency guidance motivates narrow functionality and permissions; this design's implementation boundaries are the author's recommendations.

Observability and audit

Capture request and work identifiers, actor identity, approved scope, model and instruction versions, tool calls, policy decisions, approvals, outcomes and error states. Log enough to reconstruct an action without routinely copying sensitive payloads into a second uncontrolled store.

Do not require private model chain-of-thought. Preserve the evidence and observable events that explain what the system did, together with user-facing decision rationale where appropriate. Apply retention, access control and redaction to logs. Test whether an investigator can reconstruct a sampled failed transaction from the retained record.

Failure containment and recovery

Set limits on iterations, tool fan-out, execution time and spend. Fail closed when authorization or mandatory approval is unavailable. When a downstream system times out after a write, treat the outcome as uncertain and reconcile before retrying.

Provide a kill switch that disables new consequential actions while preserving evidence and recovery access. Test its propagation time. Define compensating actions where reversal is possible and an escalation path where it is not. Rollback of an agent version does not undo effects already committed to enterprise systems.

Regulated and public-sector use

Map the complete workflow to the organization's applicable requirements, including information classification, retention, residency, accessibility and human decision responsibilities. Record the deployment environment and every external service receiving information.

For U.S. federal use, obtain a boundary-specific determination from the responsible security and authorization teams. Do not infer that an application is authorized merely because an underlying cloud service has an authorization. This publication makes no FedRAMP certification, authorization or endorsement claim and does not prescribe an authorization pathway.

Design review walkthrough

Consider an illustrative internal case-assistance workflow. The agent can retrieve authorized case records and propose a draft response. Sending the response requires a named reviewer. The approval binds the recipient and exact text. A tool adapter revalidates both before delivery, records the provider's receipt and refuses to retry an uncertain send automatically.

This is a hypothetical design exercise, not a reported deployment or case study. To adapt it, identify your irreversible step, make its approval inspectable and test failure immediately before and after execution.

Implementation acceptance criteria

A reviewer should be able to demonstrate denied access, revoked identity, expired approval, duplicate execution, malformed tool arguments, injected instructions in retrieved content, model unavailability and a working stop action. Document the expected result, observed result, evidence location and unresolved risk for each test.

Do not approve production solely because these tests pass. Add domain-specific requirements and realistic operating loads, including the workload imposed on human reviewers.

Version history and citation

1.0 — 6 September 2026. Initial conceptual reference architecture. Cite: Krishnan, B. (2026). *Reference Architecture for Agentic Systems in Regulated Environments* (Version 1.0). AgenticTransformation.ai. Use the versioned URL; no DOI has been assigned.

Stable identifier: https://agentictransformation.ai/frameworks/regulated-agentic-reference-architecture/v1.0/

References

1. NIST: AI Risk Management Framework ↗
   Published 2023-01-26. Reviewed 2026-09-06. Voluntary framework for incorporating trustworthiness into AI design, development, use and evaluation. Framework publication date; landing page is maintained.

2. Anthropic: Building effective agents ↗
   Published 2024-12-19. Reviewed 2026-09-06. Distinguishes predefined workflows from model-directed agents; recommends increasing complexity only where justified. Vendor engineering perspective, not an independent benchmark.

3. OWASP Gen AI Security Project: LLM06:2025 Excessive Agency ↗
   Publication date not stated. Reviewed 2026-09-06. Excessive functionality, permissions and autonomy are identified as causes of excessive agency. Page belongs to the 2025 edition; exact publication day is not stated.