OmniBank Agents Architecture (v1.3)

An AI Agent Operating System for Regulated Banking: Control-Plane Design, Policy-Grounded Reasoning, and Auditable Execution

Author: Ganesh Prasad Bhandari

ORCID: 0009-0002-7308-4279

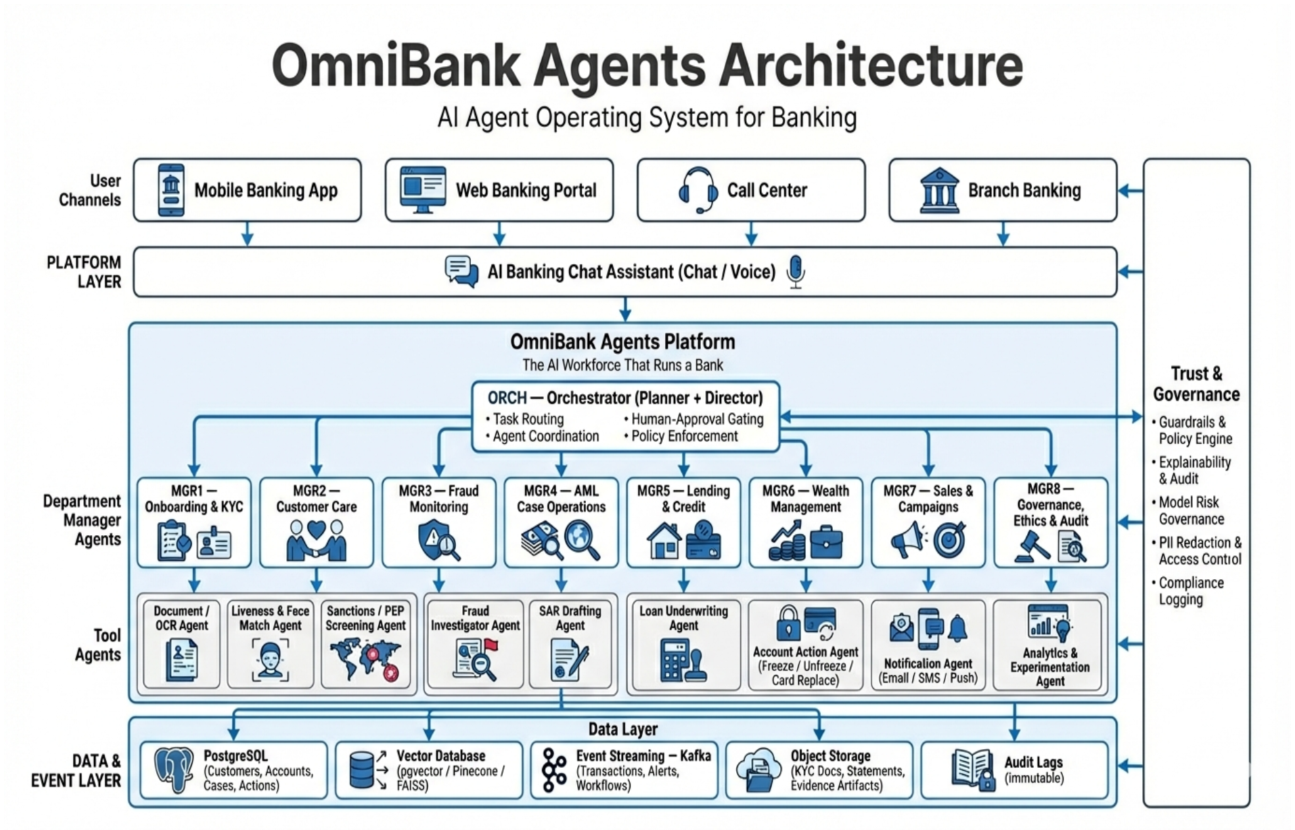
DOI: 10.5281/zenodo.18423410

Brand: AIInovateHub | LinkedIn Newsletter: AI Vanguard

Release date: January 29, 2026

Publication links

DOI:	https://doi.org/10.5281/zenodo.18423410
YouTube:	https://www.youtube.com/watch?v=nMdX96puPxM
GitHub:	https://github.com/GaneshPrasadBhandari
ORCID:	https://orcid.org/0009-0002-7308-4279
Medium:	https://medium.com/@ganeshprasadbhandari79
LinkedIn:	https://www.linkedin.com/in/ganesh-prasad-bhandari-b165b9187/


OmniBank Agents Architecture
AI Agent Operating System for Banking
User Channels
Mobile Banking App
Web Banking Portal
Call Center
Branch Banking
PLATFORM LAYER
AI Banking Chat Assistant (Chat / Voice)
OmniBank Agents Platform
The AI Workforce That Runs a Bank
ORCH — Orchestrator (Planner + Director)
• Task Routing
• Agent Coordination
• Human-Approval Gating
• Policy Enforcement
Department Manager Agents
MGR1 — Onboarding & KYC
MGR2 — Customer Care
MGR3 — Fraud Monitoring
MGR4 — AML Case Operations
MGR5 — Lending & Credit
MGR6 — Wealth Management
MGR7 — Sales & Campaigns
MGR8 — Governance, Ethics & Audit
Tool Agents
Document / OCR Agent
Liveness & Fece Match Agent
Sanctions / PEP Screening Agent
Fraud Investigator Agent
SAR Drafting Agent
Loan Underwriting Agent
Account Action Agent (Freeze / Unfreeze / Card Replace)
Notification Agent (Email / SMS / Push)
Analytics & Experimentation Agent
DATA & EVENT LAYER
Data Layer
PostgreSQL (Customers, Accounts, Cases, Actions)
Vector Database (pgvector / Pinecone / FAISS)
Event Streaming – Kafka (Transactions, Alerts, Workflows)
Object Storage (KYC Docs, Statements, Evidence Artifacts)
Audit Lags (immutable)
Trust & Governance
• Guardrails & Policy Engine
• Explainability & Audit
• Model Risk Governance
• PII Redaction & Access Control
• Compliance Logging

Abstract

Regulated banks are adopting large language models (LLMs) for customer support and internal productivity, yet most deployments remain “informational” and stop short of controlled execution. This creates a gap between natural-language interaction and bank-grade action: approvals, policy enforcement, auditability, security, and model risk governance. This white paper proposes **OmniBank Agents**, an **AI Agent Operating System (AOS)** that separates probabilistic reasoning from deterministic execution. A centralized **Orchestrator** manages routing, approvals, and policy gates, while departmental **Manager Agents** provide policy-grounded reasoning over constrained retrieval corpora. Execution is performed only by **Tool Agents** (API-bound, permissioned, and observable) with immutable audit trails. We detail the control-plane design, data/event substrate, trust & governance mechanisms, and a validation approach aligned to common model risk and security expectations for banking.

Keywords: agentic AI, regulated banking, orchestration, policy-as-code, model risk management, auditability, security, KYC, AML, fraud, retrieval-augmented generation (RAG).

1. Introduction

Banks operate under strict constraints: customer safety, regulatory compliance, privacy, security, and operational resilience. LLMs provide powerful natural-language interfaces, but they are stochastic and can produce plausible outputs that are incorrect or unauthorized. In regulated settings, the core question is not whether an AI can “answer,” but whether an AI can act under explicit controls and produce artifacts that withstand audit scrutiny.

We target an architecture that converts intent into an approved, policy-compliant work order; executes actions through signed, least-privilege APIs; and emits a replayable evidence trail. OmniBank Agents mirrors a bank’s org chart and enforces control points at the boundaries between reasoning, planning, and execution.

2. Design principles

- **P1 — Separation of concerns:** LLMs propose and explain; tools execute.
- **P2 — Policy before action:** every action passes through explicit policy gates and approvals.
- **P3 — Domain isolation:** each department agent uses a constrained retrieval corpus and prompt scope.
- **P4 — Evidence by default:** decisions store inputs, retrieved policy, tool outputs, and rationale.
- **P5 — Least privilege:** tools are permissioned per action; no broad “god-mode” agents.
- **P6 — Observability and rollback:** measurable latencies, error modes, and circuit breakers.

3. System architecture overview

OmniBank Agents is organized into three planes: **(i) Interface** (channels + conversational capture), **(ii) Control** (orchestration, routing, approvals, policy enforcement), and **(iii) Execution** (tools, data access, immutable logging). This structure makes the system governable: the control plane constrains what may happen; the execution plane guarantees what did happen.

A work-order boundary converts free-form conversation into a normalized object that ORCH can route and validate. Departmental Manager Agents reason only over scoped policy and case context, while Tool Agents execute only permissioned API calls under signature, idempotency, and logging.

4. Data, memory, and event substrate

PostgreSQL stores validated facts (customers, accounts, cases, actions). A vector store provides contextual retrieval (policies, procedures, case notes) but is not treated as authoritative. Kafka carries real-time streams (transactions, alerts, workflow state transitions). Object storage holds evidence artifacts referenced by immutable audit entries. A key design decision is **scoped retrieval**: each manager agent queries only its domain index.

5. Trust & governance

• **PII/PCI controls:** redaction/tokenization before LLM processing; role-based access to raw fields.
• **Policy engine:** policy-as-code checks for every tool invocation; hard blocks for violations.
• **Human gating:** approvals for high-risk actions (freeze/unfreeze, limit changes, SAR submission).
• **Immutable audits:** append-only logs binding work order → plan → tool calls → outputs → approvals.
• **Model risk governance:** version models, prompts, routing graphs, and policy bundles; document validation and change history.

5.1 Threat model (selected)

• **Prompt injection:** mitigated by the work-order boundary and tool allow-lists.
• **Data exfiltration:** mitigated by PII redaction, scoped retrieval, and strict access control.
• **Unauthorized actions:** mitigated by approval gates, policy engine checks, and signed tool invocations.
• **Hallucinated rationale:** mitigated by storing retrieved policy snippets and tool outputs as evidence.
• **Model drift and regression:** mitigated by monitoring, regression test suites, and staged rollout.

6. Workflow case studies

KYC

OCR + liveness + sanctions/PEP screening; policy thresholds decide if human review is required; all evidence bound to audit.

Fraud

Kafka alert → fraud manager retrieves context → proposes action; ORCH requests approval when required; tool agent executes freeze/unfreeze; notifications sent.

AML

Case assembly + evidence packaging; SAR drafting tool generates draft; submission remains human-gated; standardized artifacts improve audit readiness.

Lending

Underwriting tools compute factors; lending manager produces explanation trace with policy references; borderline/high-value cases routed to human approval.

7. Validation and evaluation approach

Evaluation must cover both reasoning quality and control effectiveness: retrieval precision and citation correctness; hallucination rate; policy-block rates; approval latency; tool error rates; security outcomes under red-team suites; and end-to-end audit replayability.

8. References (selected)

• NIST AI Risk Management Framework (AI RMF 1.0).
• Federal Reserve / OCC model risk management guidance (SR 11-7).
• ISO/IEC 27001 Information Security Management.
• OWASP LLM Top 10.

Copyright and license

How to cite

Ganesh Prasad Bhandari. (2026). *OmniBank Agents Architecture (v1.3): An AI Agent Operating System for Regulated Banking*. AIInovateHub. Zenodo. https://doi.org/10.5281/zenodo.18423410

Disclosure