

Smart Bank – AI Powered Banking Assistant

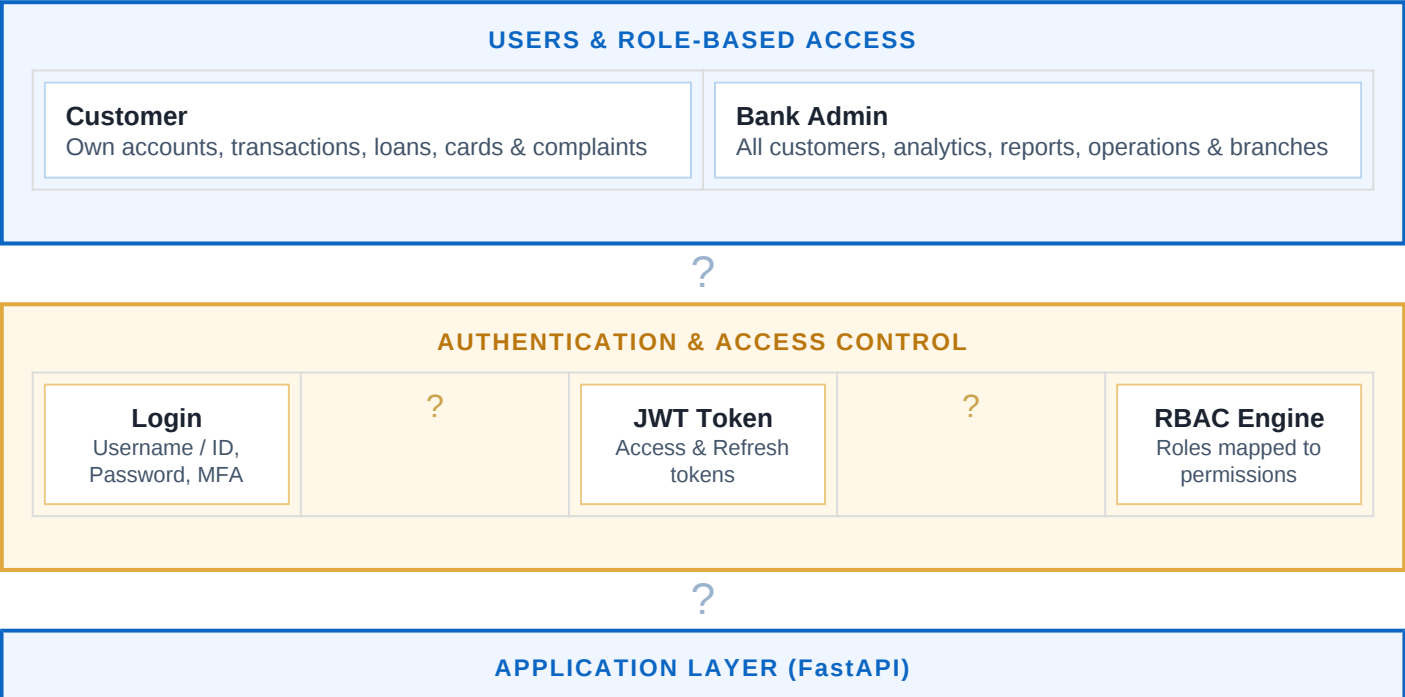
Smart Bank - AI Powered Banking Assistant

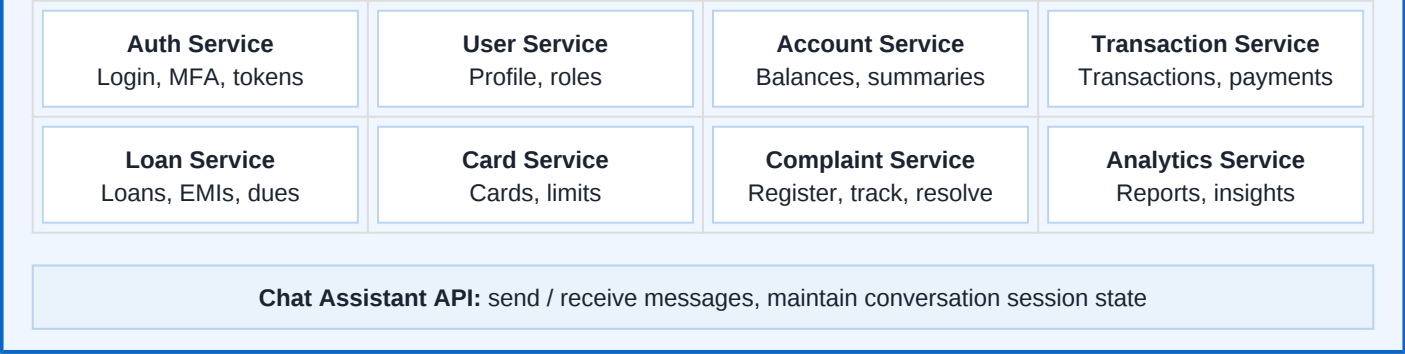
Role-Based Dashboards using Semantic Kernel, Azure OpenAI, MySQL & OpenTelemetry

A Reference Architecture for Intelligent, Secure, and Observable Digital Banking

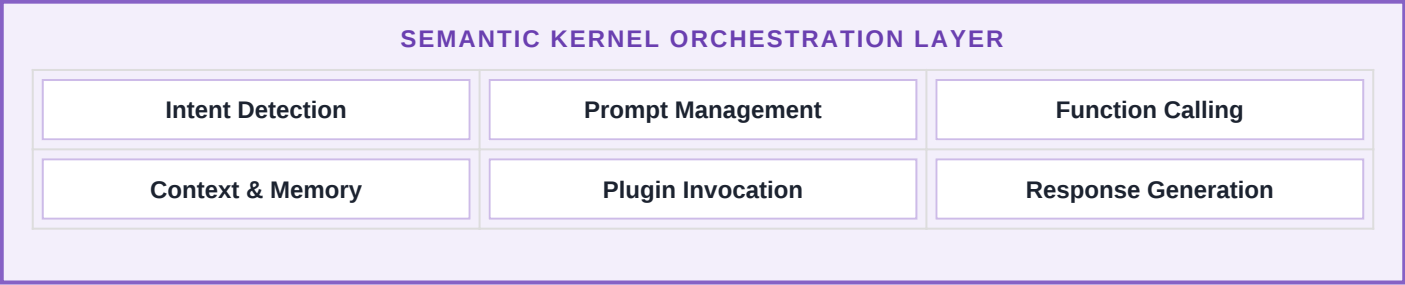
Banking customers now expect instant, conversational, and personalized service, while banks must keep every interaction secure, auditable, and compliant. **Smart Bank** answers both needs: an AI-powered banking assistant built on **role-based dashboards** for customers and administrators, orchestrated by **Semantic Kernel**, reasoning with **Azure OpenAI**, backed by a **MySQL** core data store, and observed end-to-end with **OpenTelemetry**.

Smart Bank: AI Powered Banking Assistant Architecture





?



?

PLUGINS (BANKING CAPABILITIES)

Account, Transaction, Loan, Card, Complaint, Analytics, Customer & Payment. Typed operations mapped to application services.

KNOWLEDGE & SEARCH (RAG)

Azure AI Search (vector) over policy docs, FAQs, statements & guidelines. Grounds answers in the bank's own content.

AZURE OPENAI SERVICE

GPT-4o / GPT-4.1 with embeddings: chat completion, function calling & response generation.

MYSQL DATABASE (CORE DATA STORE)

users, roles, customers, accounts, transactions, loans, credit_cards, complaints, branches.

AZURE BLOB STORAGE

Statements, KYC files, loan agreements, policies & forms.

EXTERNAL INTEGRATIONS

Payment Gateway, SMS / Email, KYC / AML, Credit Bureau & Core Banking.

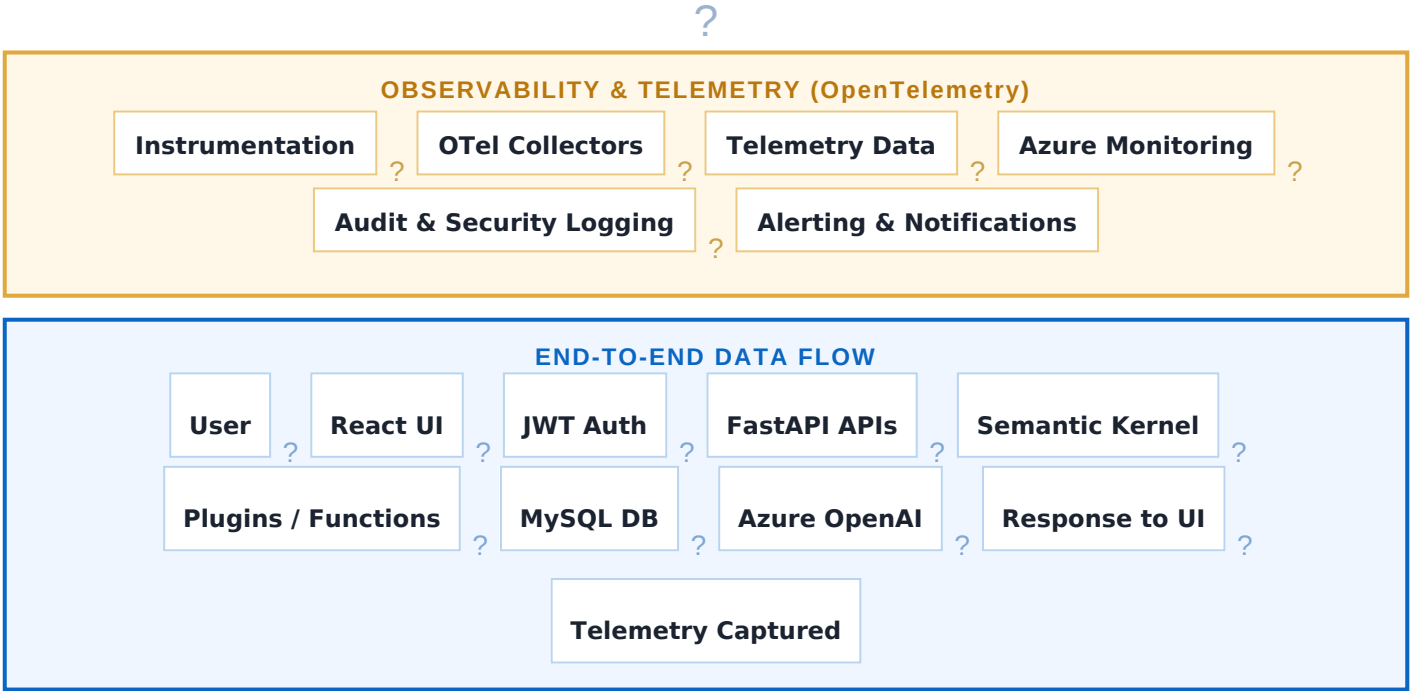


Figure 1 — The complete Smart Bank architecture, from role-based user access through the FastAPI application layer, Semantic Kernel orchestration, Azure OpenAI reasoning, MySQL persistence, and full-stack observability.

Smart Bank Architecture — Layer Overview

Layer	Component(s)	Role
Users / Access	Role-Based Dashboards	Separate Customer & Bank Admin experiences, enforced by RBAC
Authentication	JWT + MFA + RBAC Engine	Verify identity, issue tokens, map roles to permissions
Application Layer	FastAPI Services	Auth, User, Account, Transaction, Loan, Card, Complaint, Analytics, Chat APIs
Orchestration	Semantic Kernel	Intent, prompts, function calling, memory, response generation
AI Reasoning	Azure OpenAI (GPT-4o / 4.1)	Language understanding, function-calling decisions, replies

Knowledge	Azure AI Search (RAG)	Grounds answers in policy docs, FAQs, statements, guidelines
Plugins	Banking Capability Plugins	Typed banking operations mapped to application services
Core Data	MySQL Database	System of record for users, accounts, transactions, loans
Documents	Azure Blob Storage	Statements, KYC files, loan agreements, policies, forms
Integrations	External Services	Payment, SMS/Email, KYC/AML, Credit Bureau, Core Banking
Observability	OpenTelemetry + Azure Monitor	Traces, metrics, logs, alerts, audit & security logging

1. What Is Smart Bank?

Smart Bank is a reference architecture for an intelligent banking assistant that lets users converse naturally with their bank instead of navigating dozens of screens. It is not a single product but a composition of cloud-native services that turn natural-language requests like "show my last five transactions," "what is my EMI due date," or "raise a complaint" into safe, governed actions against real banking data.

Principle	What it means
Conversational	A chat assistant replaces complex navigation for everyday banking tasks.
Role-aware	Distinct experiences for Customers and Bank Admins, enforced by RBAC.
Grounded	Answers are based on the bank's own data and documents, not guesswork.
Secure	JWT authentication, MFA, and least-privilege permissions throughout.

Observable	OpenTelemetry traces, metrics, and logs feed Azure monitoring and alerting.
------------	---

2. Users and Role-Based Access

Two primary roles drive the entire experience. The architecture deliberately keeps their capabilities separate so that a single platform can serve very different needs without compromising security.

Role	Scope of Access
Customer	Own accounts, transactions, loans, cards, and complaints (self-service only).
Bank Admin	All customers, analytics, reports, operations, and branch data (organization-wide).

3. Authentication & Access Control

Every session begins at the security boundary. Credentials are verified, a token is issued, and a role-based engine decides what the authenticated identity is allowed to do.

Stage	Purpose
Login	Username or ID, password, and multi-factor authentication (MFA) for identity assurance.
JWT Token	Issues a short-lived access token and a refresh token for stateless, scalable sessions.
RBAC Engine	Maps roles (Customer and Admin) to a granular set of least-privilege permissions.

4. Role-Based Dashboards

Once authenticated, each role lands on a tailored dashboard. Both dashboards embed the same AI Banking Assistant, but its scope and verbs differ by role.

Customer Dashboard	Bank Admin Dashboard
Account summary & balances	Customer & account management

Transactions history	Transactions & analytics
Loan details and EMIs	Loan & credit card management
Credit cards and limits	Complaint management
Complaints register & tracking	Branch performance
AI Banking Assistant: chat with the bank	Reports & operational analytics
Profile management	AI Banking Assistant: ask, analyze, act

5. The Application Layer (FastAPI)

A set of focused, independently scalable services, built with FastAPI, exposes the bank's capabilities as clean APIs. Each service owns a single domain, making the system easier to reason about, test, and evolve.

Service	Responsibility
Auth Service	Login, MFA, and token issuance & validation.
User Service	Profile, preferences, and role management.
Account Service	Accounts, balances, and summaries.
Transaction Service	Transactions and payments.
Loan Service	Loans, EMIs, and dues.
Card Service	Cards, limits, and payments.
Complaint Service	Register, track, and resolve complaints.
Analytics Service	Reports, insights, and dashboards.
Chat Assistant API	Send/receive messages and maintain conversation session state.

6. MySQL Database: The Core Data Store

A relational MySQL database is the system of record. A normalized schema links identities, roles, and financial entities through primary and foreign keys, keeping data consistent and queryable.

Table	Key Fields	Purpose
users	user_id (PK), username, password_hash, role_id (FK)	Authentication identities.
roles	role_id (PK), role_name, description	RBAC role definitions.
customers	customer_id (PK), name, email, phone, address	Customer master data.
accounts	account_id (PK), customer_id (FK), account_type, balance	Bank accounts & balances.
transactions	transaction_id (PK), account_id (FK), amount, status	Money movement records.
loans	loan_id (PK), customer_id (FK), loan_amount, emi_amount	Loan lifecycle & dues.
credit_cards	card_id (PK), customer_id (FK), credit_limit, available_limit	Card limits & usage.
complaints	complaint_id (PK), customer_id (FK), type, status	Complaint tracking.
branches	branch_id (PK), branch_name, location, manager_id	Branch & operations data.

7. Semantic Kernel Orchestration Layer

The intelligence of Smart Bank lives in the Semantic Kernel orchestration layer. It sits between the chat interface and the bank's capabilities, turning a free-form request into a precise, governed sequence of operations.

Kernel Stage	What it does
Intent Detection	Interprets what the user actually wants from natural language.

Prompt Management	Builds and templates the prompts that guide the model's reasoning.
Function Calling	Selects and invokes the right banking function for the intent.
Context & Memory	Maintains conversation context so multi-turn dialogue stays coherent.
Plugin Invocation	Routes the request to the correct banking capability plugin.
Response Generation	Composes a clear, grounded answer to return to the user.

Semantic Kernel Orchestration Flow

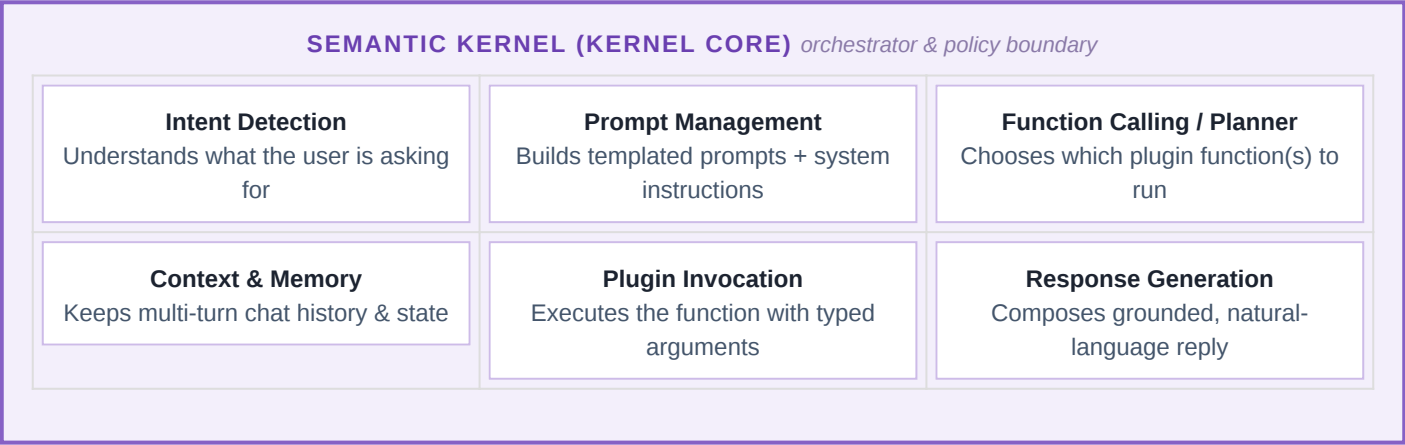
Intent ? Plan ? Ground ? Invoke ? Reason ? Respond

Customer / Admin
React UI · Chat Message

? 1

JWT Auth · Chat Assistant API (FastAPI)
Validates role & session, forwards message + identity

? 2



?

PLUGINS (BANKING CAPABILITIES)

5

Account · Transaction · Loan · Card · Complaint · Analytics · Customer · Payment. Typed functions call FastAPI services, which query / update the **MySQL Database**.

KNOWLEDGE & SEARCH (RAG) 4

Azure AI Search · vector search over policy docs, FAQs, statements & guidelines. Grounds answers in the bank's own content.

AZURE OPENAI SERVICE 6

GPT-4o / GPT-4.1 · embeddings. Chat completion & function-calling decisions, reasoning over context + retrieved knowledge.

?

OBSERVABILITY · OpenTelemetry (every step is traced)

Traces · Metrics · Logs · AI Token Usage ? OTel Collector ? Azure Monitor / Application Insights

Request path: steps 1 to 6. Response path returns through the Chat Assistant API to the user (steps 7 to 8). Telemetry is captured throughout.

Figure 2 — The Semantic Kernel orchestration flow, from chat message to grounded response, with OpenTelemetry tracing every step.

8. Plugins, Knowledge Search & Azure OpenAI

Three capabilities power the assistant's reasoning: a library of banking plugins for actions, a retrieval-augmented knowledge base for grounding, and Azure OpenAI for language understanding.

Capability	Role in the assistant
Plugins (Banking Capabilities)	Safe, typed operations the assistant can call: Account, Transaction, Loan, Card, Complaint, Analytics, Customer, and Payment. Each maps to an application-layer service.
Knowledge & Search (RAG)	Retrieval-Augmented Generation over Azure AI Search (vector search) across policy documents, FAQs, statements, and guidelines, grounding responses in the bank's own content.
Azure OpenAI Service	GPT-4o / GPT-4.1 with an embeddings model provide chat completion, function calling, and response generation: the linguistic engine behind every conversation.

9. External Integrations & Document Storage

Smart Bank does not operate in isolation. It connects to the broader banking ecosystem and stores documents durably in the cloud.

Component	Purpose
Payment Gateway	Processes payments and settlements.
SMS / Email Service	Delivers alerts, OTPs, and notifications.

KYC / AML Service	Identity verification and anti-money-laundering checks.
Credit Bureau API	Credit scores and history for lending decisions.
Core Banking System	Authoritative ledger and account operations.
Azure Blob Storage	Statements, documents, KYC files, loan agreements, policies, and forms.

10. Observability & Telemetry (OpenTelemetry)

Observability is the feedback loop that keeps the platform healthy. OpenTelemetry instruments the entire stack and pipes signals into Azure monitoring, audit logging, and alerting.

Pipeline: Instrumentation (traces, metrics, logs, events) → OTel Collectors → Telemetry Data → Azure Monitoring → Audit & Security Logging → Alerting & Notifications

Stage	What it captures
Instrumentation	Request/response traces, DB query performance, API latency, and AI token usage.
Collectors	The OTel Collector gathers and forwards telemetry to backends.
Azure Monitoring	Application Insights dashboards, workbooks, alerts, and performance views.
Audit & Security Logging	Login attempts, RBAC changes, data-access logs, and compliance trails.
Alerting & Notifications	Email, Teams/Slack, SMS alerts, and incident escalation.

11. End-to-End Data Flow

Bringing every layer together, a single request travels a clear, traceable path from the user interface to the AI and back, while telemetry is captured at every hop.

Flow: User → React UI (Dashboard / Chat) → JWT Auth → FastAPI APIs → Semantic Kernel (Orchestration) → Plugins / Functions → MySQL DB (Data Retrieval) → Azure OpenAI (Response Generation) → Response to UI → Telemetry Captured

Conclusion

Smart Bank shows how modern AI can be woven into banking without sacrificing security or control. Role-based dashboards keep experiences tailored and safe; Semantic Kernel and Azure OpenAI turn natural language into grounded action; MySQL provides a trustworthy system of record; and OpenTelemetry ensures the whole platform is observable and auditable. Every component is a managed, cloud-native service that integrates natively with the others, eliminating the friction of stitching together disparate tools and positioning the bank to deliver intelligent service that is **fast, secure, and reliable**.

A reference architecture for intelligent, secure, and observable digital banking.

Revision #6

Created 2026-07-03 06:45:33 UTC by Mohammad M Javid

Updated 2026-07-09 05:25:40 UTC by Mohammad M Javid