

Enterprise AI System Architecture & Delivery Lifecycle

Phase 1: Platform Foundation & Shared Substrate (Build Once)

Before writing any use-case code, establish the core platform, runtime controls, and operational instrumentation.

- **1. Data Architecture & Shared State Engine**
 - Define global entity schemas (e.g., LoanApplication, CustomerProfile) via strongly typed models (Pydantic / TypeScript).
 - Formalize state machines with explicit transition guards (e.g., DRAFT → SUBMITTED → UNDER_REVIEW).
- **2. Semantic Metadata & Asset Registry**
 - Catalog enterprise APIs, shared utilities, data access layers, and ML models.
 - Index them with function signatures, schemas, and descriptions in a vector store so coding agents can semantically search existing code before writing duplicates.
- **3. Enterprise RBAC & Security Guardrails**
 - Centralize roles, token scopes, and document-level ACLs.
 - Implement standard input/output guardrail layers (PII/PCI masking, prompt injection defense, schema validation).
 - Build reusable route decorators (e.g., @require_role(...), @enforce_acl(...)).
- **4. Governed RAG Ingestion Pipeline & Retrieval SDK**
 - Build ETL pipelines for knowledge documents (semantic chunking, hybrid BM25 + dense search, re-ranking).
 - Provide a standardized internal client (enterprise_rag.search(query, user_acl)) that enforces document-level permissions automatically.
- **5. Central Telemetry, Observability & FinOps Chassis**
 - Standardize distributed tracing across LLM calls, vector lookups, and agent steps (e.g., OpenTelemetry, Langfuse, Arize Phoenix).
 - Set up token-spend attribution (cost tracking per persona, per department, and per model tier) and semantic caching layers (e.g., Redis).

- **6. The Agent Contract (AGENTS.md & Golden Repo)**

- A template repository pre-wired with Docker containers, linters, base SDKs, observability wrappers, and explicit instructions governing AI coding behavior.

Phase 2: Persona-Driven Slice Design (Prepare Per Use-Case)

Break the target business process into atomic, state-bound persona envelopes. Each envelope includes both functional logic and quality benchmarks.

- **1. Persona Boundary Specification**

- **Target State Window:** Allowed entry and exit states (e.g., Underwriter Persona: SUBMITTED → APPROVED/REJECTED).
- **Authorization Scope:** Role permissions, accessible data fields, and permitted tool calls.
- **Escalation Triggers:** Conditions that kick a persona session to a fallback state (e.g., 3 consecutive validation failures trigger Support Persona).

- **2. Golden Benchmark Dataset Curation**

- Build an evaluation test suite (20–50 representative inputs, edge cases, and ground-truth outputs).
- Define acceptable quality thresholds for downstream evaluation (e.g., Faithfulness score > 0.90, Latency < 2.5s).

Phase 3: The Autonomous AI Coding Loop (Execute Per Persona)

Feed one persona spec and its test harness to the AI coding agent. The agent operates within a closed-loop generation and verification cycle.

- **1. Discover Before Build (Deduplication Check)**

- The agent semantically queries the Metadata Registry for existing tools, calculators, or API wrappers.
- *Hit:* Agent reuses the canonical asset.
- *Miss:* Agent builds a new, typed implementation conforming to platform standards.

- **2. Implement Persona Logic with Embedded Observability**

- Agent writes the routes, workflow transitions, and UI components.
- Agent instruments every LLM and tool invocation with the platform's distributed tracing spans and token-tracking hooks.

- Newly created reusable functions are queued into a staging/ candidate registry.
- **3. Automated Testing, Evaluation & Benchmarking**
 - **Deterministic Tests:** Unit and integration tests verify state transitions, schema validation, and RBAC isolation.
 - **Non-Deterministic Eval:** The golden benchmark dataset runs through an automated evaluation harness (measuring context recall, groundedness, and hallucination rates).
 - The agent cannot conclude until both unit tests and benchmark score thresholds are satisfied.
- **4. Human Review & Asset Promotion**
 - Engineers review the pull request, verify benchmark metrics, and approve newly staged tools into the permanent enterprise catalog.

Phase 4: Production Operations, Monitoring & Continuous Feedback

Once deployed, the persona slice enters the enterprise operational flywheel.

- **1. Real-Time Telemetry & Alerting**
 - Monitor latency budgets (Time-to-First-Token, tool round-trips), cost-per-transaction anomalies, and guardrail violation rates.
- **2. Human-in-the-Loop (HITL) & Escalation Queues**
 - Provide review workflows for low-confidence outputs, agent retries, or manual overrides.
- **3. Evaluation Flywheel & Continuous Improvement**
 - Capture user feedback (thumbs up/down, field edits, support resolution notes).
 - Automatically triage flagged interactions to expand the golden benchmark dataset and refine prompt or fine-tuning pipelines.

Phase	Functional Engine	Security & Governance	LLMOps & Telemetry
1. Platform	Shared models, State machine, Metadata registry	PII redaction, Document ACLs, RBAC decorators	Central tracing collector, Token cost router, Semantic cache
2. Slice Spec	Persona entry/exit states, Action contracts	Field-level permissions, Tool access policies	Golden benchmark datasets, Quality acceptance thresholds
3. Coding Loop	Tool reuse check, Workflow & API scaffolding	Auth tests, Prompt injection red-teaming	Automated evaluation pipeline, Latency & token unit tests
4. Operations	Live workflow orchestration, Agent fallback loops	Guardrail tripwire logs, Audit trail compliance	Live trace dashboards, Cost attribution, HITL feedback loops