Universal Commerce Resolver – Tech Stack & Step■by■Step Implementation

Architecture Overview

A multi-tenant autonomous ticket resolution platform. Core layers: Input (ticket ingestion), Understanding (intent/entities), Planning (playbook selection), Execution (tool calls via connectors), Verification (post-checks), Notification (customer updates), Policy/Risk (guardrails), Observability (logs, metrics), and Learning (playbook improvements).

Recommended Tech Stack

- Backend: FastAPI (Python), Pydantic, uvicorn.
- DB & Auth: Supabase (Postgres, Row Level Security), or Postgres + Prisma/SQLAlchemy.
- Queue/Workers: Celery or Dramatiq with Redis; or serverless queues (Cloud Tasks/SQS).
- LLM: OpenAI (function calling) with strict JSON schemas; optional reranker (Cohere/TEI).
- Connectors: Platform SDKs/REST (Shopify, WooCommerce, Stripe, PayPal, Shippo/EasyPost, Gorgias/Zendesk).
- Frontend: React (Next.js) admin panel; shadcn/ui; Tailwind.
- Observability: OpenTelemetry, Sentry, Prometheus/Grafana, ELK.
- Secrets: Cloud KMS/Vault; per-tenant OAuth.
- Deployment: Docker + Fly.io/Render/Vercel; CI/CD via GitHub Actions.
- Security: JWT (short■lived), RBAC, audit logs, idempotency keys.

Core Data Models (Universal)

Ticket(id, source, customer_email, text, status, platform, tenant_id, metadata)
AgentRun(id, ticket_id, status, role, risk_score, plan(jsonb), result(jsonb), timestamps)
Event(run_id, ts, type: thought|tool_call|tool_result|verify|note, payload(jsonb))
Tool(id, name, version, input_schema, output_schema, sensitivity, enabled)
Playbook(id, name, role, steps(jsonb), constraints(jsonb))
Policy(id, name, rules jsonb), PolicyBinding(policy_id, tool_id, role)
Tenant(id, name, connectors jsonb, policies jsonb, settings jsonb)

Connector Abstraction

Define a standard interface for commerce actions: get_order, cancel_order, issue_refund, update_address, get_tracking, create_return_label, create_replacement, apply_discount, send_notification. Each connector (Shopify, Woo, Custom) implements this interface and maps to native APIs.

Playbooks (Domain-Agnostic Format)

YAML/JSON that declares steps, inputs, expectations, risk, and approval thresholds. Example: refund_order → get_order → check_eligibility → create_return_label → issue_refund → notify → verify.

Agent Loop (Planner → Executor → Verifier)

Planner converts Task Object to a plan (playbook + filled params). Executor runs tools with policy checks and idempotency. Verifier checks acceptance criteria. If failure, auto-repair or escalate.

Policy, Risk, & Approvals

Per-tenant policies: limits (max_auto_refund), time windows, identity requirements. Risk score per step; pauses for approval above threshold. Enforce least privilege via role → allowed_tools mapping.

Observability & Audit

All events persisted. Attach correlation IDs, request IDs, and store raw tool I/O. Provide replay tooling and a timeline UI. Track metrics: automation rate, MTTR, success rate, human approval rate.

Security & Compliance

Encrypt secrets, RLS per tenant, principle of least privilege for connectors, signed webhooks, PII minimization, data retention controls, SOC2■friendly logging. Use dry■run/shadow mode before live actions.

Dev Environments & Testing

Local mocks for connectors (fake Shopify/Stripe/Shippo). Unit tests for tools & planners; integration tests for end■to■end flows; golden journeys + adversarial tests (fraud attempts, duplicated refunds, race conditions).

Step■by■Step Implementation Plan (12 Weeks)

Weeks 1–2: Foundations

- Set up repo, CI/CD, environments, secrets; scaffold DB schemas; create mock connectors; build ticket parser to Task Object.

Weeks 3–4: Tools & Playbooks

- Implement tool registry, input/output validation, idempotency keys; author playbooks for Refund, Order Status, Address Change.

Weeks 5–6: Agent Loop & Policies

- Planner → Executor → Verifier loop; risk scoring; human approval pauses; full event logging & audit traces.

Weeks 7–8: Real Integrations

- Shopify + Stripe + Shippo connectors; OAuth; signed webhooks; handle pagination/rate limits/backoff & retries.

Weeks 9–10: Admin Panel (MVP)

- Ticket view, plan preview, approval workflow, event timeline, metrics dashboard.

Weeks 11–12: Hardening & Pilot

- Shadow mode with 2–3 stores; measure automation rate; iterate on policies; prepare SOC2■friendly logging & on-call runbooks.