

ANDREW BOLAJI

Forward Deployed Engineer | LLM Applications, RAG & Evals | 9 Years Shipping to Enterprise Customers

Spring, TX (Houston area) | (404) 437-4881 | andrewbolaji@gmail.com
linkedin.com/in/andrewbolaji | github.com/andrewbolaji | andrewbolaji.com

Professional summary

Nine years in production systems, the last five embedded with enterprise financial and SaaS customers as the person they escalate to when something is broken and nobody knows why: run discovery, reproduce it, own it through the permanent fix. I moved that instinct into building. Seven products designed, built, and deployed solo, including a private AI assistant over a company's own documents with access control enforced at retrieval, a voice agent gated by its own eval harness, and an AWS platform provisioned entirely in Terraform. Separately, a system for a private healthcare practice, built to the customer's regulatory constraints rather than my own, and still in production.

Technical skills

Customer-facing & delivery: Technical discovery and scoping with enterprise stakeholders | Solution design and functional specs | Decomposing ambiguous problems into shippable scope | POCs and live demos | Client communication during production incidents | On-call and incident response | Root-cause analysis | UAT | Agile/Scrum

AI & LLM: Retrieval-augmented generation (RAG) | Agent orchestration and tool use | LLM application development (Claude / Anthropic SDK, OpenAI) | Eval design: deterministic assertions, LLM-as-judge with logged reasoning, recorded-replay CI gates, adversarial suites | Prompt engineering and function calling | MCP server development (Model Context Protocol, Claude Desktop integration) | Prompt-injection and PII defenses | Voice agents (Retell AI, Twilio) | Agent development tooling (committed permission policies, test-gated Stop hooks, adversarial reviewer loops)

Cloud & data platform: AWS (EKS, Redshift, S3, IAM, VPC, EC2, ECR) | Terraform (EKS, Redshift, VPC and IAM from scratch) | Kubernetes and Helm | Docker (multi-stage, pinned images) | Dagster (software-defined assets, freshness policies, partitioned backfills) | dbt | Qdrant | Prometheus, Grafana, Loki, Alertmanager | GitHub Actions CI/CD, Jenkins

Languages & data: Python | SQL (PostgreSQL, MySQL, SQL Server, Redshift) | TypeScript / JavaScript | Java | Swift | Dart | XML

Backend & app: FastAPI | Supabase (Postgres, RLS, Edge Functions) | Firebase (Firestore, Auth, Cloud Functions) | Stripe Connect | REST APIs, webhooks, SAML/SSO | React 19 | Flutter | Node

Selected projects

Designed, built, and deployed end to end, solo. Two were built for other people's businesses, one in production today; the rest are demo and pilot scale, where the work is in correctness, security controls, and proving both. Repos at github.com/andrewbolaji.

Custos: private AI assistant over a company's own documents. Staff ask questions in plain language and get answers cited back to the source document and the exact span. Python and FastAPI, local BGE-small embeddings into a self-hosted Qdrant vector store, Claude for generation, React chat UI, Docker. Security is the product: per-user access control enforced at retrieval rather than in the prompt, because a prompt is not a security boundary; retrieved content treated as data and never as instructions; PII redaction at index time and again at answer time; tools read-only by default with side-effectful actions gated behind explicit confirmation. Eight documented threats, each with a control and a passing eval, and 61 adversarial evals gating every commit in CI. In red-team testing a poisoned document convinced the model to send an email, and it tried. The action gate blocked it, because a tool with real-world effects cannot execute itself.

Client work: patient intake, scheduling and payments for a private telehealth practice. Built for the practice owner, a solo psychiatric nurse practitioner. Patient intake, calendar and booking, payment collection, and transactional email, integrated with SimplePractice, the EHR the practice already ran on. The constraints were the customer's, not mine: intake and clinical wording written against the practice's HIPAA obligations, and coverage limited to the plans it actually accepts. In production and still in use.

Porter: AI voice agent, live at 515-608-4723. Retell AI for voice, Claude for reasoning, Twilio for telephony. Holds context across a call, classifies urgency the way a dispatcher would, captures a qualified lead (name, callback number, address, problem, urgency), and walks a booking flow with SMS confirmation, which writes to a stub calendar and says so on the call. Guardrails keep it from quoting prices it was not given or diagnosing over the phone; when confidence drops it hands off to a human with a summary instead of inventing an answer. The system prompt and agent config are version-controlled and exported from the vendor dashboard, so the product itself lives in git.

Porter Evals: the CI gate for the agent above. A system prompt is the product, and a one-word edit changes it silently. 43 scripted call scenarios covering extraction, urgency classification, guardrail adherence, escalation precision and recall, and adversarial prompts (injection, system-prompt extraction, persona hijack, social engineering). An early run scored 34/43 and exposed three false positives in my own scorer, including a regex that flagged "we don't do landscaping" as an invented service because "don't do" contains "do." The agent was right and the eval was wrong, so the suite now verifies the verifier: every assertion ships with known-good and known-bad fixtures and scoring refuses to run until the scorer passes. Every push replays recorded model responses, so the CI gate costs nothing per run; a secret-gated live job re-records on prompt or model changes. The published stability pass runs all 43 live five times: 43, 43, 42, 43, 43, median 43/43, with one guardrail scenario tracked as genuinely flaky at 4/5.

Roster: compliance and operations platform, deployed with a pilot operator. React 19, TypeScript, Supabase. Operations software for regulated transitional-housing operators, built to a municipal ordinance with mandatory incident reporting and a five-year retention minimum. Hard DELETE is blocked at the database level for residents, documents, and incidents; soft-delete with retention windows is

the only path. Public form IPs are SHA-256 hashed, check-in photos EXIF-stripped in the browser, and every anonymous endpoint runs through a SECURITY DEFINER function that whitelists the exact fields it returns. Sentry with PII scrubbing and structured request-ID logging in production, routed to me.

Reckon: AWS data platform with orchestration and observability. Terraform-provisioned stack (EKS, Redshift, S3 data lake, IAM, VPC; 52 resources), ingest-to-warehouse pipeline with dbt models gated by data tests, and Dagster orchestration: software-defined assets for every ingestion and dbt model, freshness policies matching the platform's 24h/48h trust gate, retries with backoff, and real date-partitioned backfills (window-aware ingest, so a backfill touches only the requested days). kube-prometheus-stack observability with dashboards as code. The demo: break a source on purpose, watch Dagster halt downstream models, the alert fire, and the dashboard report its own staleness instead of serving old data as fresh. Includes an MCP server copilot that refuses to answer from data failing the trust gate.

Junta: two-sided marketplace with escrow. React 19, TypeScript, Supabase (Postgres, row-level security, 37 migrations, 16 Edge Functions). Homeowners post one standardized scope; vetted contractors submit sealed bids; funds are held via Stripe Connect destination charges with manual payouts, so the platform controls release timing without taking custody of user money. Full job-lifecycle state machine through dispute resolution, EXIF/GPS stripped client-side, address withheld until a lead is paid. Tenant isolation is proven, not assumed: 46 row-level-security tests run against the live database. Built for the site owner, deployed at pilot scale, payments in Stripe test mode.

Also shipped: Vouch and Chants, two Flutter/Firebase iOS apps with entitlements enforced server-side rather than in the client, including 79 Firestore security-rules tests proving a modified client cannot unlock paid content.

Professional experience

Senior Application Support Engineer (Tier-3 escalation and solutions engineering)

October 2022 - Present

Vestmark | Wakefield, MA (Remote)

The technical owner enterprise financial institutions escalate to on the VestmarkONE wealth management platform, from the first ambiguous conversation through the permanent fix.

- Run discovery directly with client stakeholders on ambiguous, high-stakes problems and decompose them into technical specifications engineering ships against
- Built Python automation adopted into the support organization's daily workflow, replacing manual triage steps and cutting time to first response on escalated tickets
- Debug production issues at the database and integration layer: SQL query optimization, XML configuration, REST API integrations, and SAML/SSO authentication flows on mission-critical financial systems
- Run root-cause and impact analysis across portfolio management and trading workflows, surfacing systemic issues and driving the permanent fix with engineering rather than the patch
- Validate fixes across dev, staging, and production, supporting user acceptance testing ahead of release

Senior Software Support Engineer

February 2021 - October 2022

Veracross | Wakefield, MA (Remote)

Backend support and solutions engineering for 300+ institutional customers across 20+ countries on an enterprise EdTech SaaS platform.

- Traced application bugs through the full stack, from client-reported symptom to database root cause, and triaged into engineering by priority
- Designed and wrote custom SQL reporting and Python tooling to fulfill bulk data and reporting requests
- Troubleshoot client API integrations end to end, isolating failures across authentication, payload, and data-mapping layers
- Mentored 3 junior engineers on debugging, SQL optimization, API troubleshooting, and client communication

Software Support Engineer

March 2019 - February 2021

SSI Schäfer | Atlanta, GA

- Maintained and debugged Java-based warehouse management systems processing millions of daily e-commerce transactions for enterprise clients including Walmart
- Wrote and ran SQL for production diagnosis, data migrations, corrections, and optimizations across live databases
- Selected for on-call second-shift production support, trusted with off-hours incident response on mission-critical systems

Software Developer

September 2017 - March 2019

Honda Manufacturing of Indiana | Greensburg, IN

- Proposed, built, and debugged Java changes to a production assembly-line control application running live manufacturing operations at a target of 1,000 cars per day
- Built real-time plant signboard systems (Moxa, DataPipe) and SQL-backed business intelligence dashboards in Tableau and SAP BusinessObjects, turning live production data into visibility for the floor and for operations leadership

Education

Bachelor of Science, Computer Science | Georgia State University, Atlanta, GA

2017