

# SAI KRISHNA RAVI

Senior Full-Stack Engineer

ravi.saikrishna487@gmail.com | +1 (551) 587-9002 | github.com/skrishna87

Los Angeles, CA — **Open to relocation; visa sponsorship required**

## SUMMARY

---

Senior full-stack engineer with 7+ years architecting and shipping internal platforms end-to-end — from database to deployment. Work in Go and Python (50/50 split) on the back end and Next.js / React on the front end, with the past three new services built in Go for performance and operational simplicity. Owned the architecture for an internal AI chatbot platform and a centralized JWT-based identity service used across the application portfolio. Routinely deliver systems from greenfield design through Azure deployment and production support.

## SKILLS

---

**Languages:** Go, Python, JavaScript / TypeScript, SQL

**Backend:** Go (stdlib, SSE streaming), Django, Django REST Framework, REST APIs, JWT (HMAC & RSA / JWK), LLM tool-calling, RAG patterns

**Frontend:** Next.js, React, PWA (offline support, QR scanning), iframe SDK design, Tailwind CSS

**Databases:** MS SQL Server, PostgreSQL, Neo4j / Cypher, Redis (cache, session store, queues), Azure Blob Storage Tables

**Cloud & DevOps:** Azure (Container Apps, Container Registry, Front Door, Blob Storage, Cache for Redis, Entra ID), Azure DevOps Server (CI/CD), Docker, Nginx, Linux

**Architecture & Practices:** RBAC design, SSO (Entra ID), short-lived SAS tokens, async task patterns, caching strategies, HTML email pipelines, testcontainers, evaluation harnesses for LLM apps

## ENGINEERING PRACTICE

---

Rigorous adversarial review workflow for AI-assisted development: dual-model code generation and review cycle, comprehensive test coverage across unit / integration / smoke tiers, and manual validation gates. Architecture decisions, judgment calls, and final verification remain human-owned. Measure twice, cut once.

## EXPERIENCE

---

**International Medical Corps** — Senior Full-Stack Engineer Aug 2018 – Present

*Initially engaged via 6COM Inc. as a contractor; converted to direct employment Feb 2021. Title reflects scope; on-record title is Python Developer.*

- Architect and lead developer on 10+ internal web platforms used by 5,000+ employees across 15+ country missions, spanning finance, operations, IT, and HR workflows.
- Drove the team's adoption of Go for new services — built the last three greenfield platforms in Go (stdlib only) for lower latency, simpler deployments, and fewer dependencies.
- Own end-to-end delivery: database design, API, front end, containerization, and deployment via Azure DevOps pipelines to Azure Container Apps.
- Standard stack: Go or Django on the API, Next.js for the UI / proxy layer, Redis for cache and session storage, MS SQL Server for primary data, RBAC and Entra ID SSO across all apps.

## SELECTED PROJECTS

---

**Internal AI Chatbot Platform (“Luma”)** — *Embedded LLM assistant deployed across 5+ internal apps.*

- Architected the platform end-to-end: Go backend with Server-Sent Events streaming, Next.js admin UI for agent and tool configuration, and a React iframe-based SDK for drop-in chat embedding.

- Designed a client / tenant model where each embedding app gets a dedicated client ID and signing secret, with JWTs scoping tool access to that app's knowledge base.
- Built an automated evaluation harness for regression detection and to streamline migration to new model versions with minimal config changes.
- Integrated with a sibling document-ingestion / graph service developed by another team, exposing its retrieval tools through Luma's per-client tool config layer.

**Internal Authentication Service** — *Self-initiated centralized JWT-based identity service for the application portfolio.*

- Identified duplicated LDAP / auth boilerplate across applications and proposed a single auth service as a self-driven initiative; took it from prototype to production.
- Migrated v1 (Django + HMAC shared secret) to v2 in Go with RSA-signed JWTs and exposed JWK endpoints — eliminating shared-secret distribution and improving rotation hygiene.
- Replaced SQL-backed refresh-token tracking with Azure Blob Storage Tables for refresh JTI persistence, removing a database dependency for the service.
- Added an endpoint to issue container-scoped Azure SAS tokens so client apps upload directly from browser to Blob Storage — removing API hops and reducing backend memory pressure.

**Asset Management Portal** — *Bi-annual asset verification for 5,000+ employees across 15+ country missions.*

- Migrated the API from Django to Go (stdlib, no framework) for better performance and simpler async task handling under bi-annual peak verification load.
- Built a PWA front end with offline support and in-browser QR scanning so field staff in low-connectivity locations can verify assets without a network connection.
- Multi-database design: MS SQL for primary transactional data, PostgreSQL for audit logs and DLQ; Redis for caching and queues.
- Companion Python script handles org-wide HTML email alerts to custodians (pending verifications) and managers (mission-level progress).

**Fleet Management Portal** — *Cloud-first vehicle, driver, and trip management.*

- Go + Next.js platform for trip requests, driver / vehicle assignment, availability and maintenance / fuel tracking; single API serves both the web admin and a mobile app maintained by another team.
- Diagnosed high-latency reports from offshore UAT users and led the resolution effort — tuned Azure Front Door routing and upstream caching to materially reduce perceived latency for distant geographies.

**Bank Balance Tracking & Executive Reporting** — *Multi-country balance entry, Citi API sync, and exec dashboards.*

- Django + Next.js portal where 400+ finance users across global offices enter daily balances and reconcile against an automated Citi Bank API sync.
- Same UI surfaces aggregated dashboards by sub-org and country for C-suite and executive consumption — collapsing what had been a manual spreadsheet pipeline.

**Additional Platforms** — *Other apps owned end-to-end on the same Django / Go + Next.js + Azure stack.*

- Facility / portfolio management (interactive map UI, automated monthly executive snapshots); IT request tracker with department / global admin RBAC; proposal planning, finance / donor tracking, and workforce planning portals for budget holders and dept heads.

## EDUCATION

---

**M.S., Electrical Engineering** — Fairleigh Dickinson University, Metropolitan Campus 2018

**B.Tech, Electronics & Communication Engineering** — JNTU Kakinada 2016