

MBOYA DANROYLEX

Backend Engineer · Python (FastAPI) · Go · PostgreSQL · Redis · AWS · GCP
onyangoroylex@gmail.com · +254 768 384 224 · Nairobi, Kenya — Open to Remote (EU / UK / Global)
github.com/MboyaDan · linkedin.com/in/danroylex

PROFESSIONAL SUMMARY

Backend engineer with 5+ years shipping production systems that handle real financial transactions and serve real users at scale. Core stack is Python (FastAPI) and Go (Gin), with deep experience in REST API design, PostgreSQL optimisation, Redis caching, idempotent M-Pesa payment integrations, multi-tenant RBAC, and cloud infrastructure on both AWS and GCP. Founder of ELS Software Solutions; currently deploying Mezani, a multi-tenant Restaurant OS with LLM-powered operational insights, on AWS. Comfortable working async-first with distributed global teams across EU/UK timezones.

TECHNICAL SKILLS

Languages	Python · Go
Frameworks	FastAPI · Gin · Pydantic v2 · SQLAlchemy · Alembic
Databases	PostgreSQL · Redis · SQLite · BigQuery
AWS	EC2 · RDS (PostgreSQL) · ElastiCache (Redis) · S3 · Lambda · API Gateway · IAM
GCP	Cloud Run · Cloud SQL · Cloud Storage (GCS) · Pub/Sub · BigQuery · IAM · Cloud Composer (Airflow)
Infrastructure	Docker · Docker Compose · Nginx · Linux · GitHub Actions · Railway
API & Auth	REST · OpenAPI/Swagger · Webhooks · JWT · OAuth2 · RBAC · Rate Limiting
AI / LLM	Anthropic Claude API · OpenAI API · LLM-augmented services · AI-powered analytics
Data & ETL	Apache Airflow · Apache Spark · BigQuery · GCS · end-to-end pipeline design
Testing	pytest · pytest-asyncio · Go test · Postman · API contract testing

PROFESSIONAL EXPERIENCE

Backend Engineer & Founder · *ELS Software Solutions — Nairobi, Kenya* 2022 – Present

- Architect and own production backend systems processing real-time financial transactions across fintech, e-commerce, and SaaS verticals — from schema design through live deployment and incident response.
- Build high-concurrency REST APIs in Go (Gin) and Python (FastAPI); select Go for connection-heavy workloads (WebSockets, Pub/Sub consumers) and Python for rapid API iteration and LLM integration.
- Establish PostgreSQL performance baselines with EXPLAIN ANALYZE, add targeted composite and partial indexes, and run Alembic migrations concurrently to avoid table locks in production.
- Design multi-tenant RBAC with JWT/OAuth2, per-tenant data isolation, and refresh-token rotation to prevent replay attacks — the same pattern across three production systems.
- Lead async, remote-first client collaboration: OpenAPI/Swagger contracts agreed before implementation, structured handoff docs, and reproducible error traces for client engineering teams.

StylerStack — Fashion E-Commerce API — *FastAPI · PostgreSQL · Redis · Docker · GitHub Actions*

- Built microservices backend powering a Flutter mobile app and Next.js dashboard — order catalogue, cart state, fulfilment, and vendor management — handling concurrent reads from multiple storefronts.
- zero duplicate charges across all production transactions via idempotency keys, atomic DB upserts, and webhook signature validation on every callback.** M-Pesa Daraja integration:
- cut average API latency by ~40% under peak load; TTL and write-invalidation strategy prevent stale data without thundering-herd spikes.** Redis cache-aside on product catalogue and session data:

- ▶ JWT + OAuth2 auth with short-lived access tokens, refresh-token rotation, and RBAC across three roles (customer / vendor / admin); full OpenAPI/Swagger docs generated from code.
- ▶ Docker Compose for local parity; GitHub Actions pipeline builds image, runs pytest suite, and deploys to Railway on every merge to main — no manual deploys.

Mezani — Multi-Tenant Restaurant OS — *Go · PostgreSQL · Redis · WebSockets · FastAPI · LLM · Docker · AWS*

- ▶ Building a production multi-tenant Restaurant OS covering the full order lifecycle: QR ordering, kitchen display queues, staff management, and multi-location reporting.
- ▶ LLM integration (Claude API) exposed via a FastAPI micro-service: generates per-restaurant operational summaries and menu recommendations from live order data — interviewable AI-in-production story.
- ▶ Real-time kitchen updates via WebSockets + Redis Pub/Sub: order events published on write, kitchen displays subscribe per-tenant — sub-200ms delivery from waiter to kitchen screen.
- ▶ Tenant isolation enforced at the DB schema level; per-tenant RBAC (owner / manager / waiter / kitchen) with scoped JWT claims prevents cross-tenant data access at the ORM layer.
- ▶ Deploying on AWS: EC2 behind Nginx, RDS (PostgreSQL), ElastiCache (Redis); automated Alembic migrations on deploy; zero-downtime rollouts via rolling container restarts.

Restaurant Analytics Platform — GCP Learning Project — *Go · PostgreSQL · Cloud Run · Cloud SQL · Pub/Sub · BigQuery · Cloud Storage · GitHub Actions · OpenAI API*

- ▶ Designed and deployed a cloud-native analytics platform to internalise GCP production patterns — order management API, event-driven analytics pipeline, BI reporting, and AI-generated operational insights.
- ▶ Go + Gin REST API containerised and deployed to Cloud Run (serverless autoscaling to zero); Cloud SQL (managed PostgreSQL) connected via Auth Proxy — no public DB exposure.
- ▶ Event-driven order processing via Pub/Sub: order creation publishes an event, an async analytics worker consumes it, aggregates revenue and volume metrics, and writes results to BigQuery — decoupling ingestion from reporting.
- ▶ BigQuery analytical layer: partitioned tables by date, revenue trends, and order volume reports queryable via API — the GCP equivalent of the Airflow → Spark → BQ ETL pipeline built for a retail client in 2020.
- ▶ Menu images and documents stored in Cloud Storage with IAM service-account-scoped access; GitHub Actions CI/CD pipeline builds, tests, and deploys to Cloud Run on every push to main.
- ▶ AI analytics assistant (OpenAI API) transforms aggregated restaurant metrics into plain-language operational recommendations — same integration pattern as Mezani's Claude-powered insights service.

Mgari — Car Rental & Fleet Booking — *FastAPI · PostgreSQL · Redis · Google Maps API*

- ▶ Booking engine with SELECT FOR UPDATE locking to atomically reserve vehicle dates and credit owner wallets in a single transaction — zero double-bookings across all production reservations.
- ▶ Redis rate limiting (sliding window via INCR + EXPIRE) and session caching; Google Maps API for live vehicle geo-tracking with geo-fenced availability zones.

Health Insurance Disbursement System — *FastAPI · PostgreSQL · JWT · RBAC*

- ▶ Multi-party payment disbursement backend for county-level health insurance; RBAC across four roles (Super Admin, Hospital Admin, CHW, Patient) with per-role scoped API access.
- ▶ QR-code eligibility verification using short-lived RS256-signed JWTs — prevents replay attacks at the hospital point-of-care without a network round-trip to a central auth server.

Retail Data Pipeline — *Airflow · Spark · BigQuery · GCS · Docker · GCP*

- ▶ End-to-end ETL for a retail client: PostgreSQL → Airflow orchestration → Spark transformation → BigQuery warehouse via Cloud Storage staging; eliminated ~4 hours/day of manual reporting.
- ▶ Containerised Spark jobs with Docker for reproducible execution; Airflow DAGs scheduled daily with failure alerting and automatic retry logic.

Freelance Backend Engineer · *Independent Clients — East Africa 2018 – 2022*

- ▶ Delivered backend systems for fintech, logistics, and SaaS clients remotely; owned full lifecycle from initial API contract to production handoff and runbook documentation.
- ▶ Built the high-concurrency booking and ETL systems above (Mgari, Health Insurance, Retail Pipeline) as client engagements; integrated M-Pesa Daraja, Google Maps, Firebase Auth, and FCM across multiple projects.
- ▶ Debugged distributed system failures in production; authored reproducible error traces and incident runbooks adopted by client engineering teams.

EDUCATION & TRAINING

Software Development Bootcamp · *Skae Hub Technologies 2021*

Backend architecture, REST API design, databases, containerisation, and cloud deployment.

ADDITIONAL

Availability: Full-time remote · GMT+3 · Comfortable overlap with EU CET (-2 hrs) and UK GMT

Languages: English (fluent) · French (basic)

Portfolio: github.com/danroylex — Mezani OS · StylerStack API · Mgari · Restaurant Analytics Platform (repos on request)