ROY THANDE

+254 790 906 416 | roy.thande@gmail.com | Nairobi, Kenya

roythan.de | linkedin.com/in/roy-thande-trf44 | github.com/thanderoy

SUMMARY

Engineering team lead with 5+ years building production identity and security systems, specializing in authentication infrastructure. Architected and led a greenfield OAuth2/OpenID Connect platform (token issuance, JWT and session handling) for a health insurance exchange serving 7 insurers and 2,500+ providers, open-sourced an OIDC authorization server, and built a custom Attribute-Based Access Control engine in Rust. Comfortable at the systems level across Go, Rust, and Python. Recently shipped a Go 1.23 service with bounded-channel concurrency, SKIP-LOCKED Postgres workers, and Prometheus observability. Introduced a team-wide RFC process and design security and tests in from the start rather than bolting them on at review.

CORE COMPETENCIES

- Python / Django / DRF
- Go Concurrency & Systems
- High-Volume Data Pipelines
- Multi-Tenant Architecture
- REST API Design
- Oauth2/OIDC
- Postgres
- Written RFC Design Process
- Docker / Kubernetes / AWS
- CI/CD Automation (GitLab)
- Distributed Systems Observability
- Cross-Language (Rust)

EXPERIENCE

Senior Software Engineer & Team Lead · Savannah Informatics, Nairobi, Kenya *July 2024 – Present*

- **Product Engineering Leadership**:Led the product engineering team of 6+ for a health insurance Electronic Data Interchange (Slade360 EDI) serving 7 private insurers and 2,500+ providers across Kenya. Subsequently joined engineering work on a national-scale variant of the same platform, serving as the digital claims infrastructure for Kenya's universal health coverage programme launched in October 2024.
- **OAuth2/OIDC Auth Platform**: Architected and delivered a multi-tenant platform extension in Python/Django with OAuth2/OpenID Connect authentication - token issuance, JWT and session handling - enabling a health informatics product to be white-labelled by third-party platforms and fuelling a West Africa market expansion partnership.
- **Market expansion:** Architected and delivered a multi-tenant platform extension in Python/Django with OAuth2/OIDC authentication, enabling a health informatics product to be white-labelled by third-party platforms – fueling a West Africa market expansion partnership.
- **Integration Layer Refactor:** Replaced a monolithic integration layer with an Adapter Pattern in Python, decoupling the platform core from its external dependencies. Each connector to a health insurer or hospital information system became an isolated adapter. New integrations that previously required weeks of core codebase changes shipped in days.
- **RFC Process:** Introduced a lightweight RFC process requiring written design discussion before any significant architectural code was committed. Reduced recurring rework, improved async cross-team alignment, and left a documented decision trail for the team.
- **ABAC Rule Engine:** Identified that off-the-shelf permission libraries couldn't express the access rules required by a multi-insurer, multi-system platform. Built a custom Attribute-Based Access Control engine in Rust that cut permission evaluation latency by ~70% and enabled arbitrarily complex, auditable access policies.
- **AI-Assisted Engineering**: Actively integrate AI tooling (Claude Code) into daily engineering work for architecture review, code analysis, and accelerating debugging cycles, meaningfully reducing time-to-solution on complex problems.

Software Engineer · Savannah Informatics, Nairobi, Kenya *June 2022 – July 2024*

- **Pipeline Reliability & Observability:** Diagnosed and resolved silent failures in a Celery task pipeline processing millions of member balance, benefits, and insurance records daily — covering 1.2M+ insured members across a health insurance exchange. Profiled the pipeline, identified unmonitored edge cases in high-volume task execution, implemented custom Sentry instrumentation, and cut the task failure rate by 60%.
- **Deployment Pipeline Overhaul:** Designed and built a Kubernetes/Docker/Ansible CI/CD pipeline from scratch. Deployment time dropped from 3 hours to 18 minutes; release success rate hit 99%.

- **Test Automation as a Deployment Gate:** Built GitLab CI/CD pipelines with automated unit and integration testing blocking every merge to production. Reduced time-to-production by 40% and made test coverage a structural property of the codebase.
- **Workflow Automation:** Built a Python automation via the Microsoft Graph API that eliminated a 15+ engineer-hours/week manual document processing workflow entirely.
- **Testing Culture:** Kept test-first development as the standard across all production services. Unit and integration tests with pytest shipped alongside every feature as a delivery criterion, written from day one.

PROJECTS

Markline - **Scope-creep recovery tool** markline.app

Built a continuous order-matching engine in Rust with a modular crate architecture - per-symbol sharded order books (BTreeMap, price-time FIFO), a pre-trade risk pipeline, write-ahead-logged trade persistence, and a real-time market-data stream over actix-web REST/WebSocket, backed by a 70+ test suite and benchmarks.

Rust, Tokio, Postgres, Websockets, actix-web

ais-tracker - Maritime Intelligence Platform (Go)

Built a maritime-intelligence service in Go 1.23 that ingests live vessel positions and serves analytics over an HTTP + WebSocket API - bounded-channel ingest with backpressure, River (Postgres SELECT ... FOR UPDATE SKIP LOCKED) background workers, and a custom Prometheus collector for ingest throughput, queue health, and per-route API latency. Concentrates capability in a single PostgreSQL 16 (PostGIS/TimescaleDB/pgvector) backend and ships as a single go:embed binary.

Go, Postgres, Websockets , River

Authy IDP - OAuth2 & OpenID Connect Authorization Server

Production-grade OAuth2 and OpenID Connect Authorization Server built in Python/Django. Designed test-first with full pytest coverage, modelled on Microsoft Entra security principles, and validated under concurrent authentication load.

Python, Django, OAuth2/OIDC, pytest

EDUCATION

Bachelor of Business Information Technology *2018 – 2022*
Jomo Kenyatta University of Agriculture and Technology, Kenya

CERTIFICATIONS & TRAINING

Developing AI Applications with Python and Flask · Google / Coursera, *2025*
Test-Driven Development with Django · TestDriven.io, *2023*
Leadership in Business and AML/CFT in Finance · UNITAR-AGFUND, *2022*

TECHNICAL SKILLS

Languages: Go | Python (Django, FastAPI, Flask) | Rust | SQL (PostgreSQL, MySQL) | JavaScript (React)
Identity & Security: OAuth2 | OpenID Connect (OIDC) | JWT | Sessions | Microsoft Entra| IAM | ABAC | OWASP Top 10
Backend Ecosystem: Django REST Framework | Celery | SQLAlchemy | Django ORM | Multi-tenant architecture
Cloud & DevOps: AWS (EC2, S3, RDS) | Kubernetes | Docker | Terraform | GitLab CI/CD | Ansible | Linux
Observability: Sentry | Prometheus | Grafana (Alloy) | Loki
Testing: pytest | TDD | Unit & Integration Testing | CI-gated pipelines