LUICE SALVATO

Senior Software Engineer | Reliable Systems, Safety Infrastructure & Empirical Research

Nairobi, Kenya | salvatoluice5@gmail.com | +254 113 794 219 | github.com/salvatoluis | linkedin.com/in/salvato-luis

PROFILE

Senior software engineer with more than three years of progressively responsible, full-time engineering experience and intensive technical training, providing a strong equivalent professional qualification in place of a bachelor's degree. Primary builder of WasaaChat's current platform, now supporting 75,000 users and growing rapidly. Led ambitious production work across distributed systems, security-sensitive infrastructure, native applications, financial systems, and operational tooling. Experienced at turning underspecified questions into testable systems, tracing failures across complex shared codebases, and shipping reliable implementations quickly. Now transitioning into empirical AI safety research, with particular interest in safeguards, adversarial robustness and AI control, scalable oversight, and the infrastructure required to evaluate frontier models safely.

EQUIVALENT PROFESSIONAL QUALIFICATION

- Demonstrated senior-level capability through ownership of a rapidly growing, 75,000-user communications and financial platform rather than through an academic credential alone.
- Built and operated consequential systems spanning real-time communication, identity, security, payments, native mobile applications, distributed data, testing, observability, migrations, and production release engineering.
- Progressed from Software Engineer to Senior Software Engineer while leading architecture, implementation, diagnosis, technical planning, code review, and mentoring across complex shared codebases.

RESEARCH DIRECTION

- Apply production security, reliability, and distributed-systems experience to empirical questions about how advanced models fail under unfamiliar, adversarial, or high-stakes conditions.
- Build reproducible evaluation harnesses, observability, experiment infrastructure, and operational safeguards that help researchers measure model behavior and identify failures before deployment.
- Develop the research judgment needed to move from robust engineering evidence to publishable empirical work, with a long-term commitment to making advanced AI systems safe and beneficial.

EXPERIENCE

Webmasters KE Ltd | Nairobi, Kenya

Senior Software Engineer | 2026 - Present

Software Engineer | May 2023 - 2026

- Architected and built the core production platform for WasaaChat, owning systems across backend services, native iOS and Android clients, consumer web, administrative tooling, financial workflows, and release readiness.
- Delivered a modular TypeScript and NestJS backend spanning identity, real-time messaging, calling, feeds, creator systems, commerce, media, notifications, and operational support, with explicit domain boundaries and extensive automated testing.
- Implemented distributed real-time communication with presence, delivery and read state, media processing, group conversations, multi-instance WebSocket fan-out, and SFU-backed audio and video calling.
- Diagnosed difficult cross-layer failures involving network state, concurrency, device lifecycle, media publication, background execution, distributed signaling, and server configuration; translated observations into regression tests and durable fixes.
- Built reliability and evaluation seams through structured logging, tracing, metrics, health checks, contract safeguards, unit and integration suites, migration gates, and CI pipelines, making system behavior measurable and failures reproducible.

Webmasters KE Ltd | Continued

- Engineered transaction-safe wallet services in Go and PostgreSQL using exact money representation, an append-only double-entry ledger, idempotent operations, compensating reversals, risk checks, payment callbacks, and reconciliation workflows.
- Designed security-sensitive controls including OTP and multi-factor authentication, step-up verification, role-based access, audit trails, KYC workflows, privacy controls, and hardened administrative actions.
- Built native SwiftUI and Kotlin/Jetpack Compose applications with secure credential storage, offline queues, background synchronization, biometrics, media systems, notifications, and platform-specific lifecycle controls.
- Led technical planning and architecture decisions, reviewed code, mentored engineers, and worked across product, design, operations, and DevOps to convert ambiguous requirements into scoped, testable, production-ready work.
- Owned build quality, safe schema evolution, security scanning, container readiness, and staged promotion workflows while partnering with DevOps on deployment and infrastructure configuration.

SELECTED TECHNICAL IMPACT

Reliable experimentation and diagnosis
Created observable, testable system boundaries across large codebases so hypotheses about failures could be checked against logs, metrics, contracts, and regression tests. Applied this approach to concurrency, real-time media, authentication, offline synchronization, and distributed delivery problems.

Security and adversarial thinking
Designed controls around identity, authorization, sensitive transactions, administrative privilege, privacy, and auditability. Treated malformed inputs, replayed operations, stale state, compromised workflows, and partial failure as design conditions rather than exceptional cases.

Complex systems implementation
Took broad product questions from architecture through implementation across services, data models, native clients, asynchronous workers, operational tooling, and release processes. This end-to-end ownership supports rapid prototyping without losing sight of downstream behavior or deployment reality.

Data integrity under failure
Applied idempotency, append-only records, transactional outboxes, compensating operations, offline replay, explicit conflict handling, and safe migrations to workflows where duplication, loss, or inconsistent state would have material consequences.

TECHNICAL LEADERSHIP

- Lead technical planning, architecture, code review, and delivery trade-offs across a broad product surface and complex shared repositories.
- Mentor final-year Information Technology students at Strathmore University, helping them turn academic ideas into maintainable systems and communicate technical decisions clearly.
- Collaborate across disciplines and explain system behavior, risk, evidence, and implementation choices to both technical and non-technical partners.

TECHNICAL TOOLKIT

Languages: TypeScript, Go, Swift, Kotlin, Dart, SQL

Backend and distributed systems: Node.js, NestJS, REST, WebSockets, PostgreSQL, Redis, Kafka, Cassandra-compatible data models, background workers

Infrastructure relevant to ML systems: Docker, Kubernetes, Terraform, GitOps, GitHub Actions, shared service infrastructure

Evaluation and reliability: Unit, integration and contract testing; structured logging; OpenTelemetry; metrics; health checks; reproducible regression testing; migration validation

Security and architecture: Authentication, MFA, RBAC, auditability, privacy controls, transaction integrity, event-driven workflows, offline synchronization, safe schema evolution

Mobile and web: SwiftUI, Swift Package Manager, Jetpack Compose, Room, WorkManager, Flutter, React, Next.js

TECHNICAL EDUCATION

Software Engineering & Artificial Intelligence | Moringa School | 2025

Software Engineering | Moringa School | Jun 2022 - Feb 2023