04 / PIPELINE

Tornike Kalandadze

Automation & Platform Engineer

Event pipelines · Reliability · Replay · Human gates

tornikekalandadze.work@gmail.com
+995 599 258 808
Tbilisi, Georgia · GMT+4
github.com/stariik
linkedin.com/in/tornike-kalandadze-997701365

3	~467/s	0
RUNTIMES · ONE PROTOCOL	MEASURED WORKER THROUGHPUT	REAL SINKS FIRED IN DRY-RUN

01 PROFILE

Automation engineer who treats side effects, failure recovery, and auditability as first-class design problems. My work spans production AI-generation workflows and independent systems prototypes: append-only journals, dead-letter recovery, cross-language RPC, native Windows event collection, backpressure, deterministic replay, and human approval gates.

02 EXPERIENCE

Lead Developer & Partner
AI Academy
Apr 2026–Present
Launched product · four-person team

- Designed a three-stage syllabus-to-course pipeline—parse, expand, generate—with contextual carry-over, schema validation, progress tracking, and a stop path that prevents orphaned token spend.
- Built auditable usage and cost tracking, a second-model review gate, automated enrollment/payment reconciliation, and admin workflows around the pipeline.
- Lead architecture, review, deployment, and product delivery across the team.

Independent Systems Engineer
Automation prototypes
2026
Event systems, distributed runtimes, local automation

- Built Relay, a self-hosted event engine with an append-only journal, backpressure, dead-letter recovery, and side-effect-free time-travel replay.
- Built Polyglot Bridge, a zero-dependency C#/Java/Rust RPC mesh whose protocol is enforced byte-for-byte by a Python conformance suite.
- Built Sentinel, a local Windows event pipeline that combines native watchers, SQLite, IsolationForest scoring, and a human decision gate.

Freelance Software Engineer
Independent client work
2019–Present
Remote product delivery

- Automate product workflows and integrations inside client systems while keeping state explicit, failures inspectable, and irreversible actions gated.

04 OPERATING STRENGTHS

- Model state transitions and failure paths before adding orchestration.
- Prefer replayable, inspectable systems over silent scheduled scripts.
- Automate execution while preserving human authority over risky outcomes.

05 TECHNICAL SKILLS

SYSTEMS
Java 21 · Spring Boot · C#/.NET 8 · Rust · Python · TypeScript

PIPELINES
Event journals · Backpressure · Dead-letter queues · Replay · Cron · Webhooks

DATA & TRANSPORT
Redis · SQLite · PostgreSQL · TCP/UDP · REST · SSE · WebSockets

OPERATIONS
Docker · Audit logs · Idempotency · Failure recovery · Human approval gates

06 EDUCATION & CREDENTIALS

IT STEP ACADEMY · 2016–2018
Software-development studies with course certificates in PHP, HTML/CSS, JavaScript, databases, and C#.

07 LANGUAGES

Georgian — native · English — C1 · Russian — B1/B2

03 SELECTED ENGINEERING WORK

Relay
Independent prototype

github.com/stariik/RELAY

Laptop-scale event engine for files, cron, HTTP, and RSS with a visual pipeline editor and immutable journal.

- Historical events can be replayed through edited YAML with sinks stubbed and network untouched.
- Backpressure blocks producers instead of dropping data; dead-letter replay resumes at the exact failed node without repeating earlier side effects.

Java 21 · Spring Boot · React · Redis · JSONL

Polyglot Bridge
Independent prototype

github.com/stariik/POLYGLOT-BRIDGE

Zero-dependency RPC mesh: C# control surface, Java scheduler, and Rust workers share a hand-designed binary protocol.

- Python vectors enforce byte-identical conformance across three implementations; correlation IDs trace every request.
- Heartbeats reroute work after a worker dies; four workers reached approximately 467 image tiles/second.

C# · Java 21 · Rust · TCP/UDP · Python

Worldwide remote · employment, contract, or freelance

Portfolio evidence and source code available through the links above.