Egor Kravchenko

Senior Full-Stack & Mobile Engineer · Regulated Production Systems · Practical AI

Antalya, Turkey (Remote) | egorkravchenko.com | linkedin.com/in/kravchenko-egor | github.com/kravchenkoegor

PROFESSIONAL SUMMARY

Senior individual contributor who ships and stabilizes operationally complex, regulated workflow platforms end-to-end — FCA-regulated fintech serving 80+ financial institutions, then a bilingual medical-tourism platform owned solo as founding engineer. Working in TypeScript / Node.js, React / React Native, and PostgreSQL since 2017: secure multi-tenant APIs, auth and authorization, and performance work that protects contracts — a GraphQL N+1 fix cutting hot-path response times 250ms → 75ms for 300+ daily users.

The recurring signature is practical AI behind deterministic, human-reviewed controls: a 19-domain decision matrix reconciling three independent LLM audits, medical redlines enforced in code, cite-or-refuse retrieval grounding behind a hard relevance cutoff, and a pre-ingestion data-governance gate that caught real leaks — AI output treated as evidence to check, never authority. I drive coding agents under a codified design-and-review system and deliver small-team output solo.

Primary stack: TypeScript, Node.js, React / Next.js, React Native, PostgreSQL, Prisma, Redis, Docker, FastAPI / Python

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, Python, SQL

Backend: Node.js, Bun, Express, Fastify, GraphQL, REST API design, FastAPI, WebSocket, OAuth 2.0, JWT, Keycloak SSO, rate limiting, BullMQ

Frontend: React, Next.js, Blitz.js, Vue.js, React Native / Expo, Vite, Turborepo

Data & caching: PostgreSQL, pgvector, Prisma, SQLAlchemy, Supabase, Redis, Redis Streams, MySQL, Alembic migrations

AI / ML: LLM application and agent design, RAG retrieval over pgvector, multimodal LLM integration with structured JSON output, provider-agnostic AI layers, Whisper / WhisperX with speaker diarization, CLIP embeddings, local LLMs, MCP

Cloud & DevOps: Docker / Docker Compose, GitHub Actions, GitLab CI, AWS, GCP, Firebase Cloud Functions, S3-compatible object storage, Cloudflare Tunnel, Linux VPS hardening, systemd, encrypted offsite backups, HashiCorp Vault, Sentry

Testing: Jest, Cypress, Playwright E2E, integration tests against live Postgres, WCAG 2.1 AA accessibility auditing, solo CI quality gates

Architecture: Domain modeling, Strangler-Fig migration planning, monorepo consolidation, RBAC and authorization design, event-driven queue-worker systems, performance optimization, technical spec writing, team leadership and mentoring

PROFESSIONAL EXPERIENCE

Founding Engineer (Contract) — Medical Avenue

Remote (Seoul / Antalya) | January 2026 – Present

Sole engineer for a medical-tourism platform taken over from a departed agency team — mobile app, coordinator dashboard, backend, infrastructure, and AI automation end-to-end.

Full-stack product platform — *TypeScript, React Native (Expo), Next.js, Node.js, Prisma, PostgreSQL, Redis*

- Shipped a bilingual (EN/RU) platform from 2.0 alpha to beta in ~4 months as the sole engineer: rebuilt the React Native patient app to 2.0, owned the Next.js coordinator dashboard, and extended the Node/Prisma/PostgreSQL backend — inherited ~120 endpoints across 12+ domain modules, added ~70 new.

- Designed the core domain model — separated intake from fulfillment with a stable patient-anchor identity and append-only, point-in-time-versioned clinical form schemas for audit isolation.
- Built a schema-driven dynamic forms engine with conditional field visibility evaluated identically on client and server — root-caused dual-evaluation drift, extracted one shared visibility-context builder, and fixed 13+ visibility defects while keeping 789 unit tests green; Zod (mobile) + Yup (backend) validation, draft autosave, optimistic file-state reconciliation.
- Cut the inherited React Native app from a 1.2 GB memory peak with OOM kills on real devices to a stable ~350 MB — the root cause was the navigation architecture itself: three nested stacks retained every visited screen. Verified on a 2018 Android with 3 GB RAM; the same effort shipped error boundaries, crash-safe startup, and an end to spurious logouts. My first mobile project after 7 years of web.
- Built and enforced the product's design system — a tokenized, accessible React Native component library plus a machine-readable design contract — consolidating 4 divergent implementations into one and eliminating cross-session UI drift.
- Replaced a manual Word-template document workflow (15–30 min per document) with a validated one-click PDF generator: coordinators pick options, the system composes the document in seconds with live KRW→USD conversion snapshotted at generation time; end-to-end memo creation went from ~4h to ~30min and the tool is in daily coordinator use.

Backend APIs, auth & security — *Node.js, PostgreSQL, Redis, Supabase, Cloudflare R2, JWT/OAuth*

- Overhauled authentication: slim JWT payloads, a refresh flow that verifies signatures of expired tokens with machine-readable error codes, Redis-backed sliding-window sessions, Google/Apple OAuth, and rate limiting on sensitive routes.
- Designed a secure file backend on S3-compatible object storage: presigned upload → confirm → download → delete flow with metadata verification, a default-deny access registry, and append-only audit logging enforced by PostgreSQL triggers — replacing permanent public links with signed, time-limited URLs.
- Hardened the Supabase data layer — row-level security on every table, revoked default public grants — then moved live updates off client table-subscriptions onto backend-published, id-only Realtime Broadcast channels.

Architecture & technical direction — *NestJS, Next.js, OpenAPI, multi-agent LLM*

- Audited a ~290-route / ~90-model backend across three API surfaces and reframed a year-stale "greenfield rewrite" spec into a Strangler-Fig migration plan that preserved backward compatibility for live clients — refusing to re-scope shipped work as net-new.
- Scoped and specified a fixed 4-month, 2–3-engineer external vendor engagement end-to-end (domain model → current-state map → scope matrix → conflicts log → full spec → executive summary).
- Reconciled three independent LLM codebase audits into a 19-domain decision matrix under an explicit evidence-priority hierarchy — treating AI output as evidence to verify, not authority.

Quality & delivery — *Jest, GitHub Actions, i18n tooling*

- Held the test/quality gate solo: ~950 tests across ~100 suites, an 80% coverage threshold, CI on GitHub Actions, and pre-commit i18n parity checks; debunked 6+ over-claimed "critical" findings from an AI code review while keeping 868 tests green through a major release review.
- The human-judgment layer over coding agents, in production: drive Claude Code and coding agents under a codified design-and-review system to deliver the output of a small team solo — every diff reviewed, tested, and shipped under my own control.

Senior Software Engineer (Full-Stack) & Team Lead — NayaOne

London, UK (Remote) | May 2021 – December 2025

Enterprise SaaS platform serving 80+ financial institutions (Lloyds Banking Group, Barclays, Valley Bank, Saudi Awwal Bank) with FCA-regulated cloud environments for fintech experimentation. **Progression:** Senior Frontend (May 2021) → Senior Full-Stack (Feb 2022) → Team Lead (Nov 2024)

Performance optimization & security — *GraphQL, PostgreSQL, Prisma, Keycloak, OAuth 2.0*

- Cut hot-path API response times 250ms → 75ms (–70%) for 300+ daily users, eliminating a GraphQL N+1 bottleneck that threatened Tier 1 bank contract renewals — refactored 80+ queries with raw SQL joins and batched cross-

database resolvers, keeping every GraphQL contract byte-identical.

- Led Keycloak SSO integration with OAuth 2.0 flows and token refresh, enabling Lloyds Banking Group and 80+ enterprise tenants under financial-services compliance.
- Consolidated 3 codebases into a Turborepo/Vite monorepo — 3 apps + 12 shared packages, eliminating ~30% duplication.

Multi-portal React/Next.js platform — *TypeScript, React, Next.js, Blitz.js, Turborepo, Prisma, Docker*

- Architected 3 production Next.js / Blitz.js apps (Supplier Portal, Marketplace Manager, Backoffice); built 100+ reusable components in a shared monorepo package, cutting duplication ~40%.
- Implemented Keycloak-based RBAC with custom React middleware securing 50+ protected routes in a multi-tenant environment; built a type-safe API-client SDK standardizing auth and data fetching.

FCA–NVIDIA AI Sandbox & platform infrastructure — *Vue.js, Node.js, GraphQL, PostgreSQL, Docker, AWS, GCP, Azure*

- Delivered the FCA–NVIDIA AI Sandbox launch module end-to-end within a 1-week deadline — GraphQL API + data model, Vue frontend, tenant configuration — shipped dark behind feature flags and switched on for the fixed public launch date (London Tech Week, June 2025), establishing NayaOne as an official FCA technology partner.
- Cut sandbox deployment from 2–5 days to <1 day for 95% of projects by architecting automated provisioning — queue-based Terraform orchestration across AWS, Azure, and GCP serving a library of 36 pre-configured fintech sandbox templates.

AI-driven product intelligence — *Python, PostgreSQL, Gemini API*

- Completed a UX audit of 35K+ sessions in 1.5h vs an estimated 50–100h manual and exposed a 70% vendor-onboarding abandonment rate — sessions reconstructed in PostgreSQL with window functions over raw event streams, emails anonymized before leaving the DB, LLM classification of funnel stage and failure mode; took the findings to the CEO as a business case.
- Cut vendor-onboarding abandonment from 70% to <5% and eliminated 200–300 annual customer-support hours, leading the 5-person team that implemented the redesign.

CI/CD & DevOps — *Docker, GitLab CI/CD, HashiCorp Vault, Redis, Sentry*

- Cut pipeline duration 60% (20min → 8min) by building a complete GitLab CI/CD pipeline across Release/UAT/Production with multi-stage caching; integrated Vault, Redis, and Sentry.
- Built and operated event-driven background-job services on BullMQ + Redis — mail delivery and activity-event processing decoupled from the request path onto durable queues and workers.

Testing, accessibility & leadership — *Playwright, Cypress, Jest, Docker*

- Architected the client-app E2E framework across two generations — built the original Cypress suite of hundreds of specs, then migrated it to a containerized Playwright framework of 600+ specs across a 3-browser matrix with parallel execution on every MR.
- Found and fixed ~100 accessibility violations (WCAG 2.1 AA) across the client apps within a day — built the axe-core a11y test suite and ran the audit end-to-end.
- Led a 5-person team (standups, sprint planning, daily reviews); mentored a non-technical QA into a Mid-Level Developer over 18 months.

Full Stack Engineer — Synergy University

Moscow, Russia | October 2017 – May 2021

- Built a full-stack loyalty platform (TypeScript, Node.js / Express, Vue.js, MySQL) with enterprise SSO (Keycloak); mentored 2 junior engineers to mid-level within 6 months.

SELECTED PROJECTS

Medical-Tourism RAG Assistant — *Python, FastAPI, async SQLAlchemy, PostgreSQL + pgvector, LangChain · Dec 2025 – Jan 2026* Source-grounded retrieval assistant over a public medical-tourism corpus, built solo and by hand in ~2.5 weeks as pre-contract proof of competence in an unfamiliar domain. Ingestion framework (LLM-restructured transcripts, semantic chunking, SHA-256 idempotency), HNSW cosine retrieval with hand-implemented MMR reranking,

24 Postgres-backed integration tests — and grounding enforced at the deployment layer: mandatory citations, cite-or-refuse prompting, a real 0.35 relevance cutoff.

clinic-ops — lead-operations tracker for medical clinics — *Python 3.13, FastAPI, SQLAlchemy 2.0 async, PostgreSQL, Docker · 2026 (design-partner pilot)* Deterministic, no-LLM campaign lead-operations tracker built and deployed in three days with 136 tests green for a plastic-surgery clinic in Antalya. A 7-stage idempotent intake pipeline behind multi-source adapters, three medical redlines enforced as phrase-checks in code, and the AI automation phases (WhatsApp funnel, lead-ads webhook, PII tokenization) deliberately designed but kept behind the deterministic v0.1 — safety boundary as code, not convention.

Hermes + gbrain — self-hosted AI agent on an operated fork — *Python, PostgreSQL + pgvector, MCP, llama.cpp, Telegram · 2026* Self-hosted personal AI infrastructure: an open-source agent and web gateway on a personally provisioned and hardened VPS, with a maintained fork of an open-source knowledge-brain engine as the memory layer — a patch branch adding a local embedder recipe, embedding-dimension validation, federated MCP reads, and slug transliteration. Built the distillation pipeline that mines past sessions into linked, sourced memory notes, with a privacy-tiered cloud/local split and a pre-ingestion secret-scan gate that caught a real leaked token.

Aura — event-driven media-analysis backend — *Python, FastAPI, Redis Streams, PostgreSQL, WebSocket, Docker · 2025* Async multimodal pipeline built solo by hand: a FastAPI producer, two Redis Streams worker consumers with consumer groups and idempotency, authenticated per-task WebSocket status streaming, multimodal LLM analysis with structured JSON output and telemetry per invocation, deployed live via GitHub Actions to AWS.

Session Distiller — *TypeScript, Bun · 2026* CLI that ingests AI-coding-agent session logs across multiple harnesses, normalizes them into a unified schema, and exports ranked, sourced notes. Real data disproved the assumed "growing-snapshot" model — rebuilt reconstruction as a tree-based chain-merge that recovers history a naive latest-snapshot merge would silently drop.

LANGUAGES

English (Professional) · Russian (Native) · Ukrainian (Intermediate) · Turkish (Intermediate)