

Artem Kuzko

Agentic-First Software Engineer / Client Solutions

I build AI-assisted solutions for the tedious work clients would rather not do by hand, and I build them by delegating to AI agents while keeping a human hand on every decision.

hello@kuzko.art
kuzko.art
github.com/akuzko
linkedin.com/in/akuzko
Based in Ukraine · Europe/Kyiv

Nearly 20 years of web development · 15+ years with Sphere · English (fluent), Ukrainian (native), Russian (native)

Now

Dearborn Labs · Engineer, via Sphere — May 2026 – present

Dearborn Labs builds AI-assisted solutions that take slow, error-prone work off clients' hands, mostly in insurance. I work on these projects agentic-first: AI agents plan, implement and review, while I own the architecture, the approval gates and the result.

- Anomaly and error detection in ledger spreadsheets with thousands of rows
- Vehicle damage recognition from photos
- Insurance document recognition and verification
- **Solution architect, AI-assisted.** On one project I also designed the full architecture and the Jira roadmap, with AI as a co-designer, while the team lead was on vacation.

Tech: Python · Django · Pydantic AI · Claude · OpenAI · TypeScript · Node.js · Fastify · Next.js · React · PostgreSQL · AWS · Terraform · OCR / vision

How I work — every ticket goes through the same loop, with a human gate where it matters

1. Plan
An agent reads the ticket and the codebase and drafts an implementation plan.

2. Approve
I review the plan and approve it, or send it back.

3. Build & review
One agent implements; a separate, read-only agent reviews until it signs off.

4. Ship
The PR goes up and CI is watched through to green.

Jira and GitHub are wired in through MCP, and plans and findings go out as shareable artifacts. I set the direction and judge the result; agents do most of the typing.

Experience

Sphere · contractor, all engagements below — Jul 2011 – present

Sphere is my long-term home: an outsourcing company I've worked with as a contractor from Ukraine for over 15 years, on engagements for its clients.

Dearborn Labs · Engineer, via Sphere — May 2026 – present

Agentic-first AI solutions for clients.

Clearcover · Fullstack engineer, Acquisition team — Mar 2024 – May 2026

Car insurance provider.

- Helped re-architect the Acquisition backend to support multiple insurance products and switch between them based on external criteria
- Refactored and re-implemented critical parts of the legacy code, paying down technical debt
- Built features across the backend and several frontends: customer-facing and agent-facing web applications

Tech: Ruby on Rails · TypeScript · React · PostgreSQL · Redis · Kafka · AWS · Kubernetes

BP Pulse · Architecture Consultant — Apr 2021 – Mar 2024

Electric vehicle charging provider in the UK.

- Analysed business and technical requirements for new features, designed solutions, and wrote documentation and implementation roadmaps
- Shaped the platform architecture and coordinated it across teams
- Designed and built new services in a multiservice architecture, and supported the existing ones

Tech: Ruby on Rails · Go · PostgreSQL · Redis · Kafka · AWS · Kubernetes

One Transport / Gett B2B · Solution Architect — Mar 2017 – Apr 2021

Logistics and transportation platform. It started as One Transport's B2B Rails and React monolith, and after its success moved into Gett's microservice infrastructure as Gett B2B.

- Led the project from scratch: repositories, servers, CI/CD, monitoring, logging and alerting
- Designed the system architecture: database schema, business-logic patterns and API
- Drove the migration from the Rails monolith into microservices on Gett's Kubernetes cluster
- Worked with cross-functional teams on technical requirements, and set the standards for code quality, testing and deployment

Tech: Ruby on Rails · JavaScript · React · Sidekiq · PostgreSQL · Redis · RabbitMQ · AWS · Kubernetes

Earlier with Sphere — 2011 – 2017

Projects older than these hold no technical value anymore. Their code is lost to me; their secrets will forever stay in the dreams of Mimir's head, and their memory lives only in the beat of Muninn's wings. Some details can still be shared in a direct conversation.

Before Sphere: Zone3000, Kharkiv — about five years, until 2011

Where it started: a PHP and JavaScript developer at a small Kharkiv company, then a switch to Ruby on Rails back in the Rails 2.3 days.

Skills

Agentic development	Daily: Claude Code: plugins, skills, subagents, review loops, MCP (Jira, GitHub), Claude API & Pydantic AI, OpenAI API, Plan → approve → build → review delegation
Languages & frameworks	Daily: TypeScript, React, Node.js, Fastify, Next.js Daily, in agentic work: Python, Django, Django Ninja, pandas Expert, no longer main focus: Ruby on Rails
Data & infrastructure	Daily or recent: PostgreSQL, Redis, Kafka, RabbitMQ, AWS, Terraform, Docker, Kubernetes, GitHub Actions, Cloudflare, Sentry, OpenTelemetry
Architecture & practice	Ongoing: Solution architecture & roadmaps, Monoliths & microservices, TDD / BDD, Agile & Scrum
Passion	Elixir / Phoenix: the language and framework that taught me the most about metaprogramming. No longer in daily use, never forgotten.

Projects & Open Source

Innkeeper.cc (commercial product, innkeeper.cc): Enterprise B2B SaaS for running restaurant operations, which I built end to end on Rails and React.

zen-service (ruby gem, rubygems.org/gems/zen-service): Service objects for Ruby applications. Used heavily in Innkeeper.

fieldwise (typescript library, npmjs.com/package/fieldwise): Form state and validation for React, built on zod. Used heavily in Innkeeper.