

Konstantin Dolbilov

Full Stack Developer

Gijon, Spain (UTC+2) • cossamedia@gmail.com • github.com/konfrontend • linkedin.com/in/konstantin-dolbilov-94720b361

10+ years shipping web products end to end in TypeScript and Java: React and Vue.js frontends, 3D graphics in WebGL, Node.js and Spring Boot REST APIs, Electron apps, and workflows for AI coding agents.

SKILLS

- Languages: TypeScript, JavaScript, Java, SQL, HTML, CSS
- Frontend: React, Vue.js, Nuxt, Zustand, TanStack Query, Tailwind CSS, shadcn/ui, GSAP, Electron
- Web platform: Core Web Vitals, Accessibility (WCAG), SEO, SSG, i18n, PWA, Web Workers, WebSockets
- Graphics: three.js, React Three Fiber, WebGL, GLSL, SVG, Canvas API
- Backend and data: Node.js, Hono, Spring Boot, REST API, RBAC, PostgreSQL, SQLite, Drizzle ORM, Zod
- Testing, build and DevOps: Vitest, Playwright, JUnit, Vite, Turborepo, GitHub Actions, CI/CD, Docker
- AI tooling: Claude Code, Model Context Protocol (MCP), Agentic Workflows, Prompt Engineering
- Methodologies: Agile/Scrum, OOP, Domain Modeling, Architecture Decision Records (ADRs)

PROFESSIONAL EXPERIENCE

Loadster – Cloud SaaS for load testing and uptime monitoring of websites and APIs.

Full Stack Web Developer (Contract)

Mar 2019 – Present

- Built 314 of the 332 components in the Vue.js design system on a Tailwind CSS theme, shared by 2 product dashboards and the Hugo marketing site, and made the dashboard mobile-responsive.
- Shipped user-requested Playwright scripting for load tests end to end: the Spring Boot API, the Node.js runtime, a Monaco Editor with custom IntelliSense, and live traces of a running test.
- Built an MCP server in Spring Boot: 23 tools under RBAC that let AI agents call the private REST API.
- Designed the Claude Code workflow as engineering infrastructure: instruction files, per-module glossaries and path-scoped conventions; merged pull requests rose 40%, 22.5 to 31.5 a month (12-month averages).
- Introduced Vitest, writing 101 of its 207 test cases, plus 376 JUnit test cases across the Java backend.
- Rewrote the recorder extension in TypeScript on the WebExtension API and Chrome DevTools Protocol: one codebase generates Playwright Test code live in Chrome and Firefox.
- Tuned frontend performance: rendered 100,000-row trace reports in parallel pages, virtualized editors of thousands of rows with IntersectionObserver, and built custom Canvas charts for real-time metrics.

Jigawatt Solar – Software suite for solar installers: interactive 3D design and energy modeling.

Senior Frontend Developer (Contract)

Oct 2019 – Jul 2026

- Authored the 3D designer's roof modeling engine in React Three Fiber, WebGL and GLSL, 34 components tested by 64 Vitest test cases: procedural roof volumes, DCEL convex decomposition, and real-time editing.
- Built the shade simulation in a Web Worker: BVH raycasting computes each roof surface's yearly shade from 2,600 sun positions in under 15 s without blocking the editor, rendered as OffscreenCanvas heatmaps.
- Chose the 3D designer's stack alone (React, TypeScript, Vite, Zustand, three.js); authored 47% of its commits, 37 of 84 merged pull requests among 8 contributors, all 17 test files (140 test cases), and 11 ADRs.
- Extended the solar estimator's drawing engine into a photovoltaic (PV) array layout tool in the Vue.js PWA, an SVG editor over Google Maps imagery, writing 93% of its vector geometry core over 231 commits.

Nataliia Musikhina Website – Portfolio website for a tattoo artist.

Freelance Full Stack Developer

Jun 2026 – Aug 2026

- Built the site in Nuxt 4 on Sanity CMS, prerendered in English and French on Cloudflare Workers instead of per-request SSR, cut LCP from 3.6 s to 1.1 s, and reached WCAG AA contrast at Lighthouse accessibility 100.

PROJECTS

timestop – Local-first time tracker app with an optional self-hosted sync server.

Author & Sole Full Stack Developer

Aug 2026 – Present

- Built the Electron app in React, TanStack Query and shadcn/ui, a Turborepo monorepo with one shared Zod entity package: 70 pull requests in 5 weeks with AI coding agents, 2 public macOS and Windows releases.
- Designed the local-first model, SQLite as source of truth with an append-only change log, and chose it over PGLite in an ADR on resident memory: about 52 MB against 576 MB to 1.3 GB, under a 150 MB budget.
- Built the idempotent sync server in Hono on Node.js and PostgreSQL over a Drizzle ORM layer shared with SQLite, storing each change once so replays are no-ops, shipped as a Docker image with Docker Compose.
- Tested the app with 452 Vitest unit test cases and 11 Playwright end-to-end test cases in GitHub Actions CI.