

Ivan Banov

Frontend Engineer – Design Systems & UI Architecture

TYPESCRIPT · STYLE ENGINES · DX TOOLING · PERFORMANCE · A11Y · TECHNICAL LEADERSHIP

I build the systems other engineers build on. Over *15+ years* I've turned fragmented UI into coherent, high-performance design systems and frontend platforms, working where design intent, render, and developer experience meet.

hi@ivanbanov.dev linkedin.com/in/ivanbanov

01 Experience

Tech Lead — Design System

Miro · Design System Team

OCT 2021 — PRESENT · BERLIN

I lead the architecture and evolution of Miro's design system, the shared UI foundation powering web, native and canvas.

- **Styling architecture across surfaces.** Designed and built the design system's styling library into a unified style layer to serve both DOM and canvas – aligned across the performance team, the canvas engine (Miro's in-house JSX-like renderer), and web. A full build-time style extractor compiles authored styles to static CSS, removing runtime cost and meeting the performance budget of the canvas's hottest paths.
- **Component architecture & Radix adoption.** Contributed to building the design system's component library and pioneered the adoption of Radix UI – wrapping primitives, defining the distribution strategy, and establishing how teams consume and extend components. The approach closely mirrored what shadcn/ui would later popularise, long before it existed.
- **Accessibility partnership.** Worked closely with the a11y team to cover their full spectrum of needs – from keyboard navigation and ARIA semantics to screen-reader edge cases. Iterating against non-obvious behaviors that sharpened both the components and my understanding of real-world assistive-technology constraints.
- **Figma to code.** Led the effort to organize and automate the icon library pipeline from Figma to code, and the design token distribution from Figma through DS packages – keeping design and code continuously in sync across consumers.
- **AI-ready design system.** Drove the move to an AI-ready pipeline where a single source flows packages to docs and MCP.
- **AI spec-driven development.** Studied and applied spec-driven development with clear constraints, making the spec and documentation the most important part of the process – enabling faster development, easier AI-assisted contributions, and consistent generation of high-quality code and tests.
- **Architecture & platform migration.** Stays current with the industry to keep the system on the right tools at the right time – adopted Stitches when the design system needed fast adoption and first-class design-token support, then architected the move away from it after deprecation, ensuring long-term maintainability without disruption.
- **Versioning & delivery infrastructure.** Owned how the design system reaches product teams: every package is versioned and released independently, and I rebuilt the build and publishing pipeline so consumers only ship the code they use. Cut CI from ~8 to ~2 minutes and shrank a heavy core package to under 1 KB, with no changes required from consumers. Written up on the Miro Engineering blog.
- **Footprint reduction & governance.** Researched and identified an ~83% reduction opportunity in Miro's CSS bundle size – formed and led a cross-team initiative to close that gap, handing the design system full styling control over the app, enforced with CSS handlers that block ad-hoc overrides of components and keep the UI consistent everywhere.
- **Automation.** Built visual-regression automation to catch UI drift before release, and automated the icon pipeline end-to-end from Figma to code.

Staff Software Engineer

Cobalt.io · Frontend Chapter Lead

AUG 2019 — OCT 2021 · BERLIN

Led the frontend chapter across four teams, setting technical direction from data fetching through to UI rendering for Cobalt's pentest-as-a-service platform.

- **Modular design system.** Built and scaled a modular design system that let four teams ship consistent UI independently and in parallel.
- **Ownership model.** Defined ownership boundaries and interfaces between frontend teams, removing overlap and unblocking parallel delivery.
- **Visual-regression testing.** Introduced a Jest + Puppeteer pipeline for automated UI validation, cutting regressions reaching production.
- **Design-system-driven PDF.** Architected a PDF-generation service via API that reused the design system, keeping print and web visually coherent from a single source.

Senior Software Engineer

Sauce Labs

JAN 2019 — JUL 2019 · BERLIN

Owned the frontend integration between Sauce Labs' real-device cloud and the testing services from the acquired TestObject.

- **Real-device cloud migration.** Helped migrate legacy real-device-cloud infrastructure to a modern, scalable platform.
- **Integration architecture.** Drove frontend optimizations that smoothed the integration between devices and testing environments.
- **SDK automation & state layer.** Designed an SDK-automation process from Swagger definitions and built the state layer that synchronized microservice data with the frontend — streamlining API consumption and maintainability.

Senior Frontend Engineer

Getsurance

JUN 2017 — DEC 2018 · BERLIN

Built Getsurance's frontend from the ground up in *Elm* — a type-safe, crash-free foundation for a regulated insurance product — and the automation around it.

- **Component library from scratch.** Created a fully reusable UI component system that let the team ship new features with minimal effort.
- **Email infrastructure.** Developed a responsive transactional-email system (Inky & Panini) for consistent branding across communications.
- **Headless CMS.** Integrated a static-site generator with a headless CMS for efficient content management.

Earlier

Across Loggi, Credits, Dafiti, UOL, and Think4 (2010–2017, São Paulo → Berlin), I grew from building brand storefronts and high-conversion checkouts into shared component libraries and app architecture — React, Redux, Ember.js, and React Native — standardizing frontend practices and shipping the products and internal systems that scaled each company.

02 Open Source

dunky.dev 🐾 — DX tools that unlock powerful UIs

2026 — PRESENT · OPEN SOURCE

A framework-agnostic foundation for design systems that solves the *wiring problem*: behavior and styles scattered across framework-specific code, forcing costly rewrites per platform.

- **State Machine.** Framework-agnostic behavior, described once. *Available.*
- **Style Engine.** Portable style schemas, resolved for performance. *Coming soon.*
- **Components.** Agnostic render layer over behavior and styles. *Coming soon.*

Links: Dunky: The Journey · [dunky.dev](#) · [GitHub](#)

03 Expertise

- **Architecture & Leadership:** Design systems · Frontend architecture · Technical leadership · Cross-team alignment · Platform migrations · API & contract design
- **Performance:** CSS optimization · Build-time extraction · Visual regression
- **Languages & Frameworks:** TypeScript · JavaScript · Elm · React · React Native · Redux · CSS
- **Systems & Tooling:** Style Engines · Design tokens · MCP / AI-ready docs · Icon pipelines · Component libraries

04 Speaking

- **Making Sense of CSS-in-JS** – Jun 2024. Why the ecosystem reached for it, how it actually works under the hood, and the runtime vs. zero-runtime trade-offs – a practical playbook for doing it right.
- **Packing a Design System** – May 2024. On architecting and shipping a design system at scale.
- **XSS – What It Is & How Attacks Work** – Mar 2017. A practical look at cross-site scripting.
- **Styleguide Concepts** – Jul 2016. From styleguide to a framework in code.

Education

- **FIAP – São Paulo** · Internet Systems, Associate Degree · 2015–2016

Languages

 Portuguese ·  Spanish ·  English