Joe Gornick

Staff Software Engineer

25 years of making systems simpler and building what teams need next, so they can focus on delivering customer value.

Minnesota joe@joegornick.com linkedin.com/in/jgornick github.com/jgornick

SUMMARY

I build the shared foundations other teams build on: design systems, front-end architecture, developer experience, APIs, data models, standards, and now the context AI agents work from, mostly in TypeScript and Node.js. I find the common problem behind what the business, its teams, and its customers ask for, and build one solution every team can use and extend. The same approach has worked at startups, agencies, and enterprises, in industries from video streaming and energy to data protection and retail. At Best Buy, I co-founded and lead the architecture of BRIX, the design system its digital product teams use.

EXPERIENCE

Best Buy

Staff Engineer Feb 2020 – Present

Senior Digital Engineer (contract) Sept 2018 – Feb 2020

BRIX · Enterprise cross-platform design system 2021 – Present

- **Championed one cross-platform design system.** During the mobile app's move to React Native, separate teams were building the same experiences for web, iOS, and Android. In 2020 I made the case that one design system, built the same way on every platform, would let one team design, build, and ship each experience everywhere. I co-founded BRIX to build it, and in 2022 we merged the platform-specific design system teams and codebases into one 10-person team. Best Buy then consolidated its teams too, with one team per experience instead of one per platform.
- **Architected BRIX's components.** Components had to look and behave similarly on web, iOS, and Android without separate logic for each. I designed its 50 accessible React and React Native components to share their logic and render each platform's own UI elements. I also left them open to extension, formalized in 2.0 as a theming and slot model, so teams changed a component only where they had to and kept the rest of BRIX.
- **Defined one specification for BRIX.** Code, docs, and design files had drifted out of sync. I designed one source for every contract, foundation, and doc, and an MCP server that exposed it to AI agents. The same contracts became the basis for generating BRIX's native iOS and Android components, now in progress.
- **Established isolated component development in Storybook.** Building components inside the full web and mobile apps meant slow feedback and components tied to app state. I standardized Storybook stories for every BRIX component, covering each variant and state and doubling as documentation and as interaction, visual, and accessibility tests. Engineers saw changes in seconds and caught regressions before components reached an app, and other front-end teams adopted the practice.
- **Led the team's move to an agentic SDLC.** AI-generated code, tickets, and docs were adding review load because agents had no project context. I set the direction and built the first pieces: a plugin of agent skills that gave agents the team's context, with a person approving every external action. Pull requests drew fewer comments on simple issues, and refinement took less time because tickets arrived with the context to start work.

Smart-home device management (IoT) · Consumer IoT platform · Tech lead 2018 – 2021

- **Integrated vendor data into the cross-platform experience.** This new product let customers monitor and control their smart-home devices from their Best Buy account. As tech lead, I integrated the vendors' data into one experience for web and mobile, and our team proved experiences could be written once for every platform, the approach BRIX later standardized.
- **Built a Device Simulator for vendor integrations.** Testing depended on slow, rate-limited vendor APIs and physical hardware. I built the simulator on HAProxy to record live requests and responses from ~10 vendors' APIs. When switched on, it answered each request with its recorded match, so test runs became reliable and reproducible.
- **Launched Store Busyness, one of Best Buy's first cross-platform experiences.** During the COVID-19 pandemic, customers needed to know how busy a store was. I turned store-sensor data into busyness and curbside wait times for every store on the website and app, falling back to 28 days of history, so customers could plan trips around crowds.

Social Patch

socialpatch.app

Founder 2019 – Present

- **Built a social media coach instead of a blocker.** Blocking apps rarely changed how people used them, teens especially. I designed an app that opened social platforms such as Instagram, Reddit, and X inside it and paused single actions, such as liking or posting, for a moment of reflection. It coached with an on-device AI model grounded in 350+ research sources and launched on the App Store.
- **Decoupled platform fixes from app releases.** Social platforms changed their pages constantly, and an App Store release for every fix was too slow. I made each platform an adapter: a signed JavaScript bundle, built from a data manifest of ranked fallback selectors and network patterns, delivered over the air and run in the platform's page. Nightly CI tests against the live sites caught drift and an AI agent skill drafted the fix, so platform fixes shipped without an app release.

Veritas Technologies (now Cohesity)

Principal UI Engineer Nov 2016 – Sept 2018

- **Architected the NetBackup Web UI.** NetBackup, Veritas's flagship enterprise backup product, could only be managed from desktop consoles. On a 10-person team, I designed its web replacement as a component-based single-page app and set the early direction for adopting the Veritas Design Language. The Web UI shipped in 2018, letting administrators manage NetBackup from a browser, and it has stayed in use since.
- **Set the API standard for NetBackup.** Each team designed APIs their own way, so the Web UI needed an integration per team. Working with the API steering committee, I wrote the specification every team adopted, built on open API standards, so the Web UI integrated with every API the same way. The first standardized APIs shipped in 2018.

PREVIOUS EXPERIENCE · 2001 – 2016

Power Over Time, Senior Software Engineer (2015 – 2016): for its grid-interactive controllers, which let utilities manage home appliances, I built the device software: a captive portal for setup, device-to-cloud APIs, and Wi-Fi-to-cellular failover that kept the controllers online when home Wi-Fi dropped.

Revation Systems, Software / IT Architect (2014 – 2015): for LinkLive, a HIPAA-compliant cloud contact center, I replaced reports that queried raw events with a streaming analytics pipeline and a report builder where each report was a configuration, making report generation much faster. I also set up infrastructure as code to manage data centers across multiple locations.

Packet Power, Senior Software Engineer (2013 – 2014): for EMX, a data-center power-monitoring portal, as its main developer I modeled each rack's power chain, linking each sensor to its place in the data center and to what powered it, and added versioned releases and migrations so cloud and on-prem installs upgraded alike.

Irdeto, Senior Software Engineer (2010 – 2013): for MediaManager, a video content management system, as lead engineer I moved its storage from a document database to a relational one and replaced its hard-coded content model with a type-driven entity framework, so new content types became definitions rather than new code.

space150, Senior Creative Technologist (2010): for this digital agency's national brands, I led a team building front ends, including a keyboard- and screen-reader-accessible statement guide a major card issuer needed for compliance.

Swarmcast, Lead Web Developer (2007 – 2010): for its HD video delivery platform, I built a scheduler that provisioned live-event encoders automatically, so staff no longer ran them by hand. I also built the HD video client's libraries, a subscription site, a streaming TV app, and the MLB.TV NexDef installer for every browser plug-in.

Patterson Companies, Senior Programmer Analyst (2004 – 2007): for this dental distributor, I replaced the one-off scripts importing all transactions with reusable ETL applications, so a new file layout took minutes instead of hours. I also built the web service that moved legacy systems off an aging network protocol with minimal changes.

MiniMax Corporation, Web / Software Developer (2001 – 2004): for ScadaCam, infrared and thermal cameras that utilities used to inspect substations remotely, I built its web interface for live views, camera control, and setup, so crews could check a site before driving out.

SKILLS

Front end	Component frameworks (React, React Native, Angular), design systems and tokens, accessibility
Data and APIs	Relational, document, and key-value stores (PostgreSQL, MySQL, MongoDB, Redis), message brokers (RabbitMQ), REST APIs (JSON:API, OData)
Languages	TypeScript, JavaScript (Node.js)
Testing and delivery	Monorepos (pnpm workspaces, Lerna), unit, visual, and accessibility testing (Jest, Storybook, axe), CI/CD and GitOps (GitHub Actions), infrastructure as code (Terraform)
AI development	Agent skills, MCP, on-device models, human-in-the-loop workflows