

ALEXANDER R. DAVIS

Toronto, Ontario, Canada (remote)

alex@name-industry.com | +1 416 524 8210

name-industry.com | github.com/name-industry

PROFILE

Senior developer with a systems orientation: architecture through to implementation. Rich clients on web, mobile and desktop, from Flash application development through React Native and Electron to React SPAs, and the backends behind them (Go, Node, API-first with live OpenAPI docs). Two shipped React Native apps, iOS and Android, with native modules in Swift and Java. Real-time data end to end: NATS, MQTT, websockets, time-series stores, canvas and D3 rendering. Design background (OCAD) and ad agency years (Cossette, Leo Burnett, MacLaren McCann, BIMM) building consumer-facing work for brands such as Bell, Ford and General Mills, before moving to startups and MVPs.

SKILLS

Front end: React, React Native (iOS and Android; native modules in Swift and Java), TypeScript, Next.js, Electron, D3, Apache ECharts, canvas, PixiJS

Back end and data: Go (Fiber), Node, PHP, REST and OpenAPI, MySQL, Elasticsearch, OpenSearch, GreptimeDB, InfluxDB, MongoDB (including Azure Cosmos DB's Mongo API), Typesense, NATS JetStream, MQTT, websockets

Auth and platform: Keycloak, Microsoft Entra, OIDC and OAuth 2, DPoP, OPA and RBAC, HIPAA-aware design, Docker, nginx, Ubuntu, Wolfi (Chainguard), Azure, CI/CD (GitLab, Forgejo Actions), NX monorepos

Design and tooling: UI/UX, Affinity Designer, Photoshop, OmniGraffle; AI-assisted development (Claude Code, local RAG over library docs)

EXPERIENCE

Anyon Systems, Frontend Developer (full stack), Jul 2024 to Aug 2026, remote

Quantum computing company. Brought in to speed up front end development on Thunderhead, the web app over the Go/Fiber API. External clients such as Calcul Québec use it to manage their users, API keys, job permissions and job results on MonarQ, Anyon's 24-qubit quantum computer; jobs themselves are submitted from the client's own code with the key. Anyon's internal teams use it to monitor the machine and control which parts of it are available to clients, for example taking most of the machine down for calibration while a few qubits stay available to students. Took over the Next.js front end from a bare scaffold and owned it through to a working handover. Added the Keycloak and JWT authentication integration to the Go backend.

- Built Thunderhead's client-facing UI: user onboarding and management, projects and groups, job list and status, and API key show, renew and reset, giving Calcul Québec self-service administration of its users, keys, submission permissions and job results on MonarQ
- Built the internal operations UI: a custom canvas graph for real-time TLS metrics, daily calibration metrics as lattice and linear graphs in pure CSS, Apache ECharts for the rest
- Added admin controls on the lattice graph that toggle individual qubits and couplers available or unavailable to clients, which the API enforces on job submission
- Built the Go authentication CLI for Thunderhead: OAuth login through Keycloak, which brokers transparently to each organisation's identity provider (CCDB for Calcul Québec, Microsoft Entra for Anyon), returning the user's API key for job submission to MonarQ from their own code
- Built Thunderhead's SDK pipeline: TypeScript, Python and Go SDKs generated from the Go/Fiber backend's OpenAPI spec, with a hand-maintained compatibility layer between the generated code and the public surface, so backend and dependency changes could ship without breaking consumers; the TypeScript SDK is what the Thunderhead web app runs on
- Built the developer portal and release pipeline for the SDKs: Next.js on GitLab Pages, GitLab CI that versions, publishes and writes the changelog for every release, with getting-started guides and the full consumer API reference
- Moved Thunderhead's CI images to Wolfi and Wolfi-based images for a minimal, continuously patched base, and began moving the Go/Fiber backend to a ports-and-adapters structure on that foundation
- Also created Anyon's style guide and built the new corporate website, business cards and trade show banners from it; set up Google Search Console, Mailgun and the email notifications for Thunderhead
- Mentored a junior developer on OpenAPI, CI setup and the first pass of the Python SDK, including practical use of AI coding tools

Name Industry, Principal, 2008 to present, Toronto

Design and development practice: contract work for clients and internal projects, open source where possible.

Contract work

- **Linewatch, Electron and Puppeteer software agent**, 2023. macOS and Windows desktop agent for a client that drives a browser session on the client's own sportsbook account, captures the live odds (moneylines, spreads) the site delivers to that session alongside ESPN schedule data, matches upcoming NBA and NCAA games to live ones with fuzzy search, and maintains a watch list with pinned events and auto-refresh. React UI over an Electron main process driving Puppeteer; IndexedDB storage; session logging.
- **MedusaJS parts catalogue**, 2023. Parts catalogue for a client, built on MedusaJS.
- **MEI, EHR schedule application and API** (via Aleph-1, later direct), 2019 to 2022. HIPAA-compliant (US clients), API-driven identity provider and data store in PHP and Elasticsearch, later moved to OpenSearch; OpenAPI v3 spec with live documentation for integration testing; Electron desktop app in React with natural-language search over the EHR; architecture documentation; all server containers (Ubuntu, nginx).
- **Maple Post, Questelevators mobile app**, 2018 to 2021. React Native app for iOS and Android that remote-controls a custom residential car elevator (four hydraulic posts, built by Questelevators) over MQTT with JSON payloads; custom Swift and Java modules for certificates and secure storage; UI/UX design.
- **Maple Post, condoadvisory.com**, 2020. Electron app for importing data into Wix collections, plus site design.
- **Aleph-1, CharityData.ca**, 2018. PHP and MySQL API ingesting years of Canadian charity tax data from CSV; denormalized schema for retrieval speed and low-cost schema changes.

Internal projects

- **White Room** (ni-app.ca), 2024 to present. Multi-tenant user management in Go and Fiber. Authorization with OPA (Open Policy Agent). Keycloak brokering social login (GitHub, Google), OIDC with DPoP. Real-time typeahead and semantic user search on Typesense. Time-series storage sits behind a package seam, with GreptimeDB chosen after running it beside QuestDB and InfluxDB 3. An experimental Arrow IPC path streams Greptime query results through the API to Perspective (FINOS) in the browser without re-parsing, for real-time system monitoring.
- **ni-rag**, 2026. Air-gapped hybrid documentation search: dense vectors in Chroma, BM25 in process, a concept graph in ArcadeDB, fused, reranked and boosted. Indexes are built on a connected machine and copied over with the containers; the deployed system runs and answers with no network. Returns ranked chunks with a calibrated confidence tier (high, medium, weak, where weak means the corpus does not cover the question) and every retrieval score per chunk, never a composed answer; debug mode exposes each stage's pool and timings, drawn as a pipeline graph in the UI. Served to Claude Code over MCP.
- **CCSM**, 2025. Climate-controlled enclosure on a Raspberry Pi 5: eight SHT35 sensors over I2C, four relays on GPIO, NATS JetStream on the bus, Telegraf into GreptimeDB, React SPA dashboard with the sensors as a live DAG and a 3D view of the enclosure; login through White Room.
- **ni-ina219**, 2022. Node module for the Raspberry Pi UPS HAT, published on npm.

Quartermaster, React Native Developer, 2016 to 2018, Toronto (partly remote from Vietnam)

Consumer app for homeowners: scan appliance receipts and nameplates to get manuals and service alerts; later a community app connecting homeowners with trades professionals. iOS and Android.

- Built the mobile app from zero as the sole mobile developer, against the company's existing API: authentication, onboarding, chat and group messaging, news feed and social features, push notifications and deep linking
- Set up cloud CI so iOS and Android builds shipped from one dashboard for stakeholders, plus over-the-air updates for releases between store submissions
- Carried the app through React Native's fastest-moving period, 0.17 to the 0.5x line, first alone and then with a second developer I mentored