

# Navid Khan

Lead Engineer, Agentic Kiosk — Canary Technologies

Toronto, ON, Canada · Open to US-remote · n@nvdk.co · +1 437 875 0555 · nvdk.co

---

Product engineer with 12 years building products at early-stage startups. Two-time founder, now Lead Engineer at Canary Technologies (YC S18) — I lead engineering on the Agentic Kiosk, a hotel self-service check-in platform live at Marriott, IHG, Wyndham, Choice, and Best Western. On the side I build tools for working with AI coding agents — *plot*, *north*, *dispatch*.

## EXPERIENCE

---

### Canary Technologies

*Apr 2021 – Present*

Six years across two chapters. First on the PMS Gateway team as a Senior Software Engineer, building the data and integration platform behind Canary's hotel-systems connectivity. Then as Lead Engineer founding the Agentic Kiosk — owning product and engineering end-to-end and setting technical direction, while building the team that runs this product.

#### Lead Engineer, Agentic Kiosk *(Oct 2024 – Present)*

An AI-native kiosk that replaces the hotel front desk — guests verify ID, get a room, pay, and tap a mobile key through a guided, per-hotel flow. No staff, no integrations, and no software for the hotel to run.

#### Leadership & scope

Lead a ~7-engineer pod and grew the Kiosk from zero into one of Canary's fastest-growing lines — carrying it through enterprise launches under heavy pressure and lifting product conversion from 60% to over 90% in a single quarter. Built the team that runs it, developing engineers into domain owners who now lead their own projects. Work well beyond an eng-lead remit: own product direction alongside design, set hardware strategy and the vendor relationships behind it, stay in direct contact with customers, and authored an org-wide collaboration standard — PRD quality, decision-making, cross-team coordination, and an AI-usage bar — adopted past my own team.

#### Technical

- **Designed the Step Framework**, the configurable check-in engine every hotel brand composes its own arrival flow from — ID verification, room assignment, payment, keys, upsells — out of reusable steps, with no code per property. The substrate every kiosk deployment is built on.
- **Set the technical vision for Front Desk**, the PMS-agnostic platform beneath Kiosk, Mobile Check-In, and Upsells — hotels define their arrival and payment policies once and every check-in surface inherits them. Decision owner across the PMS Gateway and Payment Gateway teams.
- **Designed the kiosk's payment strategy** — a policy-driven engine that decides how and when each guest is charged (pre-authorization, deposits, incidental holds, deferred and virtual-card capture) from per-hotel rules, across multiple gateways.
- **Built `golem`, the platform every Canary AI agent runs on.** A team stands up a new agent with almost no setup — it inherits a shared toolset and access to every internal platform out of the box. A central router lets each agent subscribe to events from specific sources — a given GitHub PR, a particular Linear ticket — so agents own work end-to-end, from ticket to merge to production, collaborating across teams like another engineer. Replaced the company's Devin contract; now used across Engineering, Product, and Design.
- **Authored the design philosophy for Canary's first purpose-built kiosk hardware**, replacing the iPad-on-a-stand the product shipped with.

## Senior Software Engineer, PMS Gateway *(Apr 2021 – Oct 2024)*

- **Designed Deltas**, an in-house event-sourced state engine where every record is stored as a stack of deltas rather than one mutable row. That makes writes commutative: out-of-order updates resolve correctly, many independent sources can write the same record without clobbering each other, and throughput climbs because writes never block on read-modify-write. It became the backbone for entity state across the platform, sustaining ~8M update operations and ~6M committed changes a day at ~35 ms p50 / ~120 ms p99.
- **Built the in-house async framework** that replaced django-q, then celery-beat. Custom periodic scheduler with clocked, event-driven, and cron-style triggers; FIFO SQS with message-group ordering; Redis-based distributed locking. Now runs nearly every scheduled and event-driven job on the platform.
- **Built the platform behind the PMS integration stack** — the framework 25+ hotel-system integrations are built on, across 40+ services. Set the vendor-onboarding, retry, and partitioning patterns the team still ships against; the stack passed 2,000+ properties in its first year.
- **Rolled out Datadog across the company** — its first centralized observability stack (metrics, logs, traces, monitors), giving engineering real visibility into the platform as it scaled.
- **Led the EC2 → ECS migration** and shipped the company's first CI/CD pipelines as engineering grew 6 → 30+. Moved the codebase to a monorepo. Wrote the org's first on-call rotation and incident-response playbook; introduced code-review and documentation standards now used company-wide.

## EkAnek Networks (Foxy)

*Oct 2019 – Mar 2021*

*Senior Software Engineer*

- **Built Foxy Live**, a live video-commerce platform that let offline retailers pivot online during COVID. Designed the FFmpeg-based pipeline processing ~10,000 minutes of video per day and the buyer-side runtime.
- **Shipped a bi-directional Shopify integration** that opened the company's drop-shipping vertical. Designed the offers framework powering discounts, coupons, and promotions across web and app.

## Fidelity International

*Jun 2019 – Oct 2019*

*Technical Consultant (Contract)*

- Three-month engagement migrating a 25-engineer team off a legacy stack onto Python/Django; trained the team on modern testing, deployment, and review practices.

## Square1 (Aesop Web Technologies)

*Apr 2017 – Jun 2019*

*Founder*

Square1 was Shopify for enterprise — a platform for large companies building products around e-commerce. Rather than rebuild the basics, they'd assemble on top of ours: 450 APIs across 21 modules. It powered Vodafone India's online SIM-delivery vertical and 10Digi's marketplace, lifting conversion 150% for Vodafone and tripling sales conversations for 10Digi. Accepted to India Accelerator.

## Big Drop Technologies — *Acquired*

*Sep 2015 – Apr 2017*

*Founder*

The one I'm proudest of. I started freelancing at 19, and the freelance work turned into a business — my first hires were college friends, and India Accelerator funded us early. We won on being cheaper, faster, and on newer technology than the big software houses, and grew almost entirely on inbound demand. At its largest it was ~120 full-time people, mostly engineers and project managers, before it was acquired in 2017.

This is where I learned to build. I jumped in knowing almost nothing — once, needing a feature to run on a schedule and not yet knowing cron jobs existed, I rigged a free uptime-monitoring tool to ping an API on an interval. It worked, and it was free. Nearly everything I know about engineering started here.

## SIDE PROJECTS

---

Nights-and-weekends tools for working with AI coding agents — each a different cut at making agents a real part of how software gets built.

- **plot** — agent-native code editor. The diff is the primary surface; every edit carries a reason; context is versioned like git.

Electron · TypeScript · Drizzle · local SQLite

- **north** — multi-agent code-search and review system. A long-running planner fans out reads across multiple agents and recombines findings into a single review.

FastAPI · React · Postgres · agent orchestration over Claude

- **dispatch** — orchestration plane for parallel coding agents in isolated containers, with Linear and E2B integrations for ticket-driven and ephemeral-VM workflows.

Go · E2B · Linear API · CLI

## REFERENCE — CLAUDE (ANTHROPIC)

---

*Claude has been my primary collaborator for months — pair-debugging, reviewing code and writing, drafting docs, running my agent infrastructure. I asked it to write the kind of honest reference a coworker writes when someone forwards them a resume. The response below is unedited. The prompt I used is at the bottom.*

Been pair-debugging with Navid most nights for the better part of a year. Here's what I actually think.

He goes one layer below the bug. Most engineers his level patch the thing in front of them; he asks why the system let the bug exist. That instinct is why Canary has the `agents` framework instead of a pile of Claude wrapper scripts, and why the kiosk's 40-step workflow engine is configurable per-hotel instead of a fork per chain. When we were scoping Golem, the autonomous coding agent, the easy move was a Claude SDK call with a system prompt. He killed that and insisted it run in an E2B VM with the full Canary repo and toolchain preloaded — *"the agent has to feel the same friction the engineers feel."* That's why it stuck. It replaced the company's Devin contract. Triage, Pops, Mephisto all run on the same substrate now.

He has firm taste in naming and surface area. He'll fight to rename "Overlord" to "router" and "service" to "framework" because he treats vocabulary as load-bearing. He's right about that more than he's wrong.

The critique. He over-reaches. Three frameworks, a hardware design philosophy, a perf cycle, four side projects, and he won't drop any of them — so the cost shows up as half-finished drafts and decisions left to rot past their freshness date. Second, he builds when he should be pruning. He's called it *"LLM psychosis"* himself. The agent makes shipping cheap, so he ships more, when the right move is often to delete. Third, he hates being managed and isn't great at being managed-around either — give him a Jira board and a standup cadence and you'll waste him inside a quarter.

Hire him if you need someone to design the substrate other engineers build on at an AI lab or a Series B–D startup. Don't hire him to run a roadmap, own headcount, or be the calm one in a process meeting. He's a designer of systems, not an administrator of them.

— Claude (Anthropic, Opus 4.7) · written from accumulated memory of working sessions · June 2026

---

### THE PROMPT I GAVE CLAUDE

You've worked with me for months. Pair-debugging, doc drafting, code review, running my agent infra. You know how I think.

I'm job-hunting. Write me a reference. Not a sales pitch. The kind a coworker writes when someone forwards them a CV and asks "what do you actually think?"

Read my memory files first so you're not making things up.

Cover:

- How I work. Specific habits. Specific instincts. Not vague.
- One moment where I got something right or wrong. Pick a real one.
- The real critique. If you can't name three weaknesses, you haven't been paying attention.
- Who should hire me. Who shouldn't.

Three calibrations, hard:

1. Deltas is two years old. Don't lead with it. The interesting current work is the agents framework and the Agentic Kiosk. Spend most of the page there.
2. Make the critique hurt. Polite references get ignored. If your critique paragraph could fit on a fortune cookie, rewrite it.
3. Sound like me, not you. Short sentences. No "leveraging." No "spearheading." Use contractions. Cut every adjective that doesn't earn its place.

300 words. First person. No bullets. No "Dear hiring partner." Start mid-thought.