

The Story

I have told my story to many people. Someone recently suggested I write it down. So here it is.

Youth

I started my first web development company when I was 15 years old. Search [archive.org](https://archive.org/details/omnisite.net) for [omnisite.net](https://archive.org/details/omnisite.net) or [click here](#). Yes. It's very 1996.

During my university years, I was more interested and excited in hacking on my own things and "real jobs" than I was in my degree. I started a couple more little companies: MonoHost, which sold ASP.net webhosting on Linux, and Citygroup Consulting. I worked five cooperative "work terms" (internships, to non-Canadians) and, by the time I started my first fulltime jobs at the RCMP and Adxstudio as a programmer, I was a bit disillusioned with software development. Even as a recent graduate, I had plenty of experience watching software teams struggle to deliver software in Java and C#. I started writing Ruby and experimenting with Rails before 1.0, but I wasn't confident a little web framework could survive in a world of Microsofts and Oracles. I quit Adxstudio, thinking maybe I'd go teach English in China.

ThoughtWorks

Instead, I joined ThoughtWorks. ThoughtWorks appealed to me for two reasons: First, they were already a massively international company when I joined in 2005 — they had offices in India and China that I was very excited to visit. I was a small town kid. I wanted to see the world. Second, they seemed to *get it*. Developers at ThoughtWorks obsessed about quality. At ThoughtWorks, pair programming, TDD, and CI weren't just novel ideas. (And they were still somewhat novel back in 2005.) They were the company's way of getting ahead.

I hacked away in C#, Ruby, and Java for companies like Enmax, Shaw Cable, Suncor, rockthetvote.org, and Liquidnet. I scoped and planned projects, coached other developers, and learned about sales and capacity planning... it was a pretty intense couple of years. I got to work alongside the people who normalized Lean, kanban boards, unit tests, continuous integration, distributed version control, build pipelines, and continuous delivery.

With a competing offer in hand, I managed to convince ThoughtWorks to send me to India to help start the Pune office, near Mumbai. After spending a year in India I desperately wanted to stay. But the 2008 financial crisis had other plans and I was sent home to Canada.

DRW Trading

Almost no one does well in high-volatility, high-uncertainty markets. Almost. But algorithmic trading firms do very well in precisely those kinds of markets and when DRW offered me nearly three times what I was making at ThoughtWorks, I couldn't turn them down.

I moved to Chicago, joined the team that would ultimately rewrite DRW's entire trade capture and positions management stack, lead the team that wrote a completely new instrument platform, and then... got bored, I guess.

High finance has many things to offer: great people, technology, work, and money. But rarely does it ever feel like you're making much of a difference in the world. Some friends I'd made during my first year in India kept pestering me to join their little startup. Finally, I relented and moved back to India.

C42 Engineering

I thought that, perhaps, I'd kick around India for a year. Before that year was out, though, C42 closed doors and let go of all their (young but exceptionally talented) staff. While C42 wound down, the rest of us went for long lunches and talked about what we would want a company to look like, if we started one.

Nilenso Software

My only stipulation for starting Nilenso with these folks was that it must be *a worker cooperative*. No one should own any more than anyone else and it shouldn't suffer the fate of so many other little companies, where the people who own the business don't even work there.

We had other fundamental rules, too. We refused to build weapons. We refused to spy on people. We refused to create software that manipulated human behaviour. Integrity was key.

Beyond that, we wanted to build a business where any one of us would be happy to work for the rest of our lives.

We wanted to work with great tech. Over the years we wrote Clojure, Elixir, PureScript, Haskell, Java, C, Go, and other languages. We built large ML models, dynamic pricing systems, experimentation platforms, streaming video services, large distributed systems, embedded systems, and some of the world's first local-first software.

We wanted to work with great clients. We built software for Staples Labs, Flipkart (India's Amazon), PayTM Insider (a large Indian entertainment company, if you're not from India), GoJek, MapBox, and Simple.org.

We also experimented a lot with product development at Nilenso, producing a lot of open source as a consequence: <https://github.com/nilenso>.

In 2014, I had a bicycle crash in California while visiting a client. Thanks to the joys of the United States healthcare system (or lack thereof), I ended up with a crushed optic nerve and a permanent scleral buckle, after a couple botched surgeries. I couldn't program for a few years, so I took on even more of a leadership, mentoring, sales, and hiring/HR role at Nilenso.

Around 2015, an inside joke taken too far led me to start wearing nothing but custom-tailored black suits with white shirts. To this day, that's all I wear in public... unless I'm doing yoga or riding a bicycle.

I left in 2018 (as explained below) but Nilenso still exists, it is still owned by its employees, and it remains a very successful and well-regarded company.

Travel and Covid

After many years, my partner and I were finding Bangalore's pollution and traffic noise to be too much for us. We decided to leave. After leading Nilenso for the better part of a decade, I hired my replacement and spent a year traveling, trying to decide what was next (and whether or not that thing even involved software). I spent a lot of time considering meditation research but, as you can see from my resume, I'm not a neuroscientist so this was really a non-starter. We had actually decided we would move to the Himalayas when the Covid lockdown hit.

While stuck in the Himalayas — Himachal and Kashmir — for most of 2020, I began volunteering with Pariyatti (a 501(c)3 nonprofit from Washington state) to help them build a digital library. My eyesight was getting better, so I thought this was a good way to get back into a little bit of programming. In a fit of over-engineering, I decided this digital library needed a graph database. I'd begun with Neo4j, but ultimately landed on XTDB (née "Crux"). After spending significant time with the XTDB team as a user, they offered me a job.

By this point, I was back in Canada for Covid's Third Wave because I couldn't stay in India on my visa. The XTDB team was spread across Germany, Sweden, and the UK so moving to Halifax was a good fit for timezones. I've since made friends here and gotten involved with a local nonprofit (Atlantic Canada Vipassana Meditation), so this is home for a while.

XTDB

I started on the XTDB team by writing long-form articles, explaining exactly what an "immutable, bitemporal, time-traveling graph database" was — and why anyone would want one. I hired the team that redsigned opencrux.com (the old brand, which was sued out of existence) into xtdb.com. I led the team that defined the brand and hired a designer to consult with us on the new XTDB logo. I spent months on market research with potential customers — a move that coincided with our shift from XTDB 1.x (a row-oriented, datalog queryable OLTP store) to XTDB 2.x (a columnar, disaggregated HTAP store). XTDB was receiving a lot of interest at the time, including interest from VCs. Håkan Råberg, XTDB's architect, and I contended that XTDB needed to become its own company, separate from the consultancy (JUXT Ltd.) which spawned it. The owners of JUXT disagreed, so we left to begin Endatabas.

Endatabas

We built, marketed, and pre-sold Endatabas throughout 2023 and 2024. By the time we felt comfortable looking for the venture capital to take Endatabas to market, however, the VC world had changed a great deal. 2021 was a great time to find funding for a funky, open source database. 2024? Not so much.

Some VCs even went so far as to say things like "This is a great idea. If you brought me this

two years ago, I could write you a cheque for \$2 million today. But... it's 2024." I at least admire their honesty.

As we wound up Endatabas, I spent significantly more time working with the three non-profits I contribute to: Pariyatti, The International Vipassana Technology Association Inc., and Atlantic Canada Vipassana Meditation. This was a wonderful way to spend the latter part of 2024.

What are you looking for, exactly?

Good question! From someone like me, there isn't a single answer. I've led software teams as large as 30 and as small as 2. I've designed databases, distributed systems, and SQL dialects. I've built teams, companies, and products.

There are two spectrums I think about the kind of work I'd like to do.

On one spectrum, there is *meaning*: is the thing I'm doing really important? Over the past 5 years, I've really enjoyed working more and more with nonprofits and open source. But I also interviewed for a Staff Engineer role at Stripe recently. Arguably, India's UPI is the meaningful equivalent of something like Stripe, but I still think what Stripe builds is useful.

The second spectrum is one of the role's *responsibility*. I'm quite happy to work in any of the roles I've been doing over the past 10 years: Program Manager, (Technical) Product Manager, Engineering Manager, or CEO. I love working with people and helping teams build a great product, so all of these make sense to me. But I'm also quite happy to step back into an IC role. Especially if I found a job where I could work on open source or open data or work in the tech stack I use in my volunteer work (predominantly Rails and Flutter), I might find that really gratifying. Sometimes it's fun just to write code.

Now what?

I'm happy to answer any questions you have. If you live in or near Halifax, I'd rather meet you in person. If you don't, I'm happy to spend a couple hours on a no-obligations call just to get to know each other.