

Product Engineer

Years of creating products at small tech startups, has given me the privilege of working intimately with brilliant founders and colleagues from a wide range of backgrounds. I do everything from interaction design to scalable infrastructure. I'm obsessed by context; I love speaking with customers and aligning product roadmaps with their needs. I believe people come first.

Work

[Salamander](#) ML-ready servers

2018-now founder

Bootstrapped, designed, built, and onboarded 1,000 users to my compute platform for AI professionals and students. As **first-to-market** with persistent storage for spot servers, I **reduced costs by 60%** compared to the previous industry standard. As I transitioned from minimum viable product to maturity, Google, Paperspace, and Crestle started making similar improvements to their own offerings.

Salamander is the only ML-ready cloud to include advanced features like complete usage audits, and to offer free credits to students via the GitHub education pack. According to customers, Salamander "addressed a lot of pain points that other platforms couldn't", is "a god-tier setup", also "**the web interface is super intuitive and it is really easy to get started**". Over half of customer support emails include a note of appreciation.

I sold Salamander to fastai. It will become a vital resource for their community (100k students).

[VenueScanner](#) event planning

2018 senior full-stack

I inherited a codebase abandoned partway through a rewrite from Node to Ruby, **reduced the scope of engineering projects**, and identified which planned features were essential. Each leader at VenueScanner had a unique perspective, which I helped guide toward an **affirmative product direction**. I created a messaging platform with integrated contract negotiation & payments.

In addition to brownfield development, I also:

- Identified and fixed critical security issues
- Responded to service outtages; and improved uptime, from 90% to 99%
- Reduced loading time for customer support tickets, from 2 minutes to 5 seconds

[Beyond Labs](#) enterprise budgeting

2016-2018 full-stack

Beyond Labs nurtured my (then) nascent technical leadership skills. I migrated years of prior work to React. I helped the team **adopt practices like continous integration** and automated testing. I built common libraries to help scale the product team from one to four developers and **redesigned parts of the codebase**. I also **wrote feature proposals and planned sprints**. My eternal thanks to Xavier Wilders for the trust he invested in me, and whose mentorship helped me become the engineer I am today.

Some completed projects at Beyond include:

- Internationalisation tooling, with spreadsheet integration
- Rules engine for matching accounting records with projects
- Datagrid library which can load and scroll through **100k+ rows without any perceptible delay**

MyGravity loyalty rewards

2015-2016 front-end

Given sole responsibility for isolated parts of MyGravity. I chose React & Redux, to build the reporting dashboard and campaign management tools.

Movidiam film production

2014-2015 junior front-end

Transformed completed designs into a minimum viable product under supervision. Created the first interactive prototypes for most features available on the website today.

Education

- Taught myself programming with online resources in 2013
- Taught myself Machine Learning (including Deep Learning) in 2017
- Reading 35 books every year

Favourite Things

This section was originally called "Skills", but there's more to engineering than "React and 7 other frontend frameworks" (Ember, Angular 1, Angular 2, jQuery, Backbone, Elm, Cycle). Engineers like me can explain their work to anyone, help teams ask good questions, and navigate an array of useful mental models. I can explain how computers work from transistor chemistry upto the v8 internals. I can narrate 150 years of computer history in detail and recognise when trends repeat themselves. I can pick context-specific algorithms and data structures.

- web framework: React; approachable team, and always raising the bar for UI excellence
- deep learning framework: PyTorch; versatile, and enables easy experimentation
- language: JavaScript; it is ugly and beautiful, adapts to new ideas, and helps you build things fast
- runtime: Node; it's JavaScript, but would be nice to build with something new (like Go or Rust)
- datastore: Prisma; a productivity revolution that just works and can adapt
- SOLID principle: dependency inversion; did this for years, but only learned the name recently
- software pattern: bowtie; distinguish between core functionality and business logic
- VCS: git; remember `git merge -Xignore-all-space` when introducing an autoformatter
- fiction: The Poetry Cloud; Liu supports philosophical exposition with technical detail
- non-fiction: Making Learning Whole; how to share skills/knowledge effectively
- philosophy: ascendency; predict the future, and explain the present
- absurdity: I cannot decide between Dadaism and evaluating variables at extreme values
- sport: dance; it makes others happy to see me dancing, and I like spreading joy
- podcast: The Knowledge Project; Shane demonstrates how to listen actively and ask good questions
- cellular automata: Brian's Brain; the emergent structures are interesting