

I’m a software engineer with varied IC and management experience at startups since 2015.

I’ve twice transitioned from IC to management roles, which most recently led to me heading up the Engineering org at Waybridge through a transition period after we were acquired by a competitor¹. While I’ve enjoyed that experience and I expect to do whatever a business needs, I still consider myself a hands-on engineer. My interests include:

- Product-focused software development. I’ve mostly worked on backend systems², using Python³ and Postgresql for a lot of the last 10 years. I’ve deployed on popular cloud environments like AWS⁴, using Linux, Docker and Kubernetes.
- All things internal tooling, team processes and developer experience. I like making the environment better for myself and the people around me.
- Production monitoring: I think it’s vital to have good insight into your programs, and I’ve set up monitoring approaches a couple of times using tools like Datadog, or Prometheus and Grafana.

Personal details

Last updated:	March 2024
Location:	Remote/London
Email:	hi@mattduck.com
Website:	www.mattduck.com
Github:	github/mattduck
Linkedin:	linkedin/mattduck

Work history

2	Waybridge 2020 - 2023
3	Ometria 2017 - 2020
3	LETO 2015 - 2017
4	Other experience

¹ Eng was only 9 people post-acquisition.

² Sometimes monoliths, sometimes distributed / services architectures.

³ also Go, JS/Typescript, PHP.

⁴ usually orchestrated by tools like Terraform or Ansible.

Waybridge is a B2B SaaS supply chain product for enterprises that buy or sell physical commodities (eg. metals). In 2022 customers used the platform to manage material worth more than \$7 billion. I joined as an early **Software Engineer** when we were pre-customer, was promoted to **Engineering Lead**, and then later to **Director of Engineering**. In 2023 we were acquired, and I led the Engineering org through an initial post-acquisition period.

- I wrote code for hundreds of contributions across our backend⁵ and infrastructure⁶. I rolled out our production monitoring tech⁷ and processes⁸, designed our customer data import automation system⁹, and built the user permissions and authorization layer. As we grew I drove improvements related to API latency and database usage¹⁰.
- I took on additional management responsibilities and worked on various team initiatives¹¹. After an internal re-org I was asked to join the Waybridge leadership team, spending a year co-running Engineering, managing half our engineers¹², and helping the company get to the next stage of investment or M&A.
- I led Engineering for five months after our acquisition by Minehub Technologies. I advised the new exec team on technical strategy, risks and hiring. I promoted a new engineering manager, and brought in a talent & people consultant to support Engineering and the wider business.

⁵ Mostly a Python/Django/Celery monolith, using GraphQL, Relay and Graphene for the API layer.

⁶ AWS, EKS/Kubernetes deployed via Helm/Helmfile using Gitlab CI, Terraform, Postgres, Redis, Cloudflare Workers.

⁷ Datadog for metrics, logs, traces, etc. Alerts configured as code using Terraform. Pager rota via Opsgenie.

⁸ eg. our incident management process, on-call rota and bug reporting tools.

⁹ Python, Postgres, S3, SFTP. Standard problems like matching inconsistent object IDs across customers, replaying records etc.

¹⁰ eg. I drove a transformation in attitude towards SQL where it made sense over ORMs, educated on topics like N+1 problems and separating logic from I/O, and built various tooling to support this.

¹¹ eg. sprint processes, guilds, our Engineering progression framework, hiring processes, performance reviews, SDLC practices, SOC2 compliance efforts.

¹² I managed 7 engineers across London, San Francisco, NY and Amsterdam.

OMETRIA

2017/JUL - 2020/JAN

Ometria is a VC-backed marketing SaaS product that drives revenue for brands like *Not On The High Street*, *Hotel Chocolat* and *Fred Perry*. I joined as a **Backend Developer** when the company had ~25 people, and saw it grow to more than 100 people. As we grew I was promoted to **Senior Developer**, **Lead Developer** and then **Engineering Director**.

- I was the lead engineer on our data ingestion systems for 18 months. These were business-critical OLTP workloads¹³ that formed the endpoint for processing Ometria's inbound ecommerce data (orders, customers, browser sessions etc.) - up to 10 million records daily and total hundreds of millions of rows.
- I authored or worked on ~50 production services and libraries that powered the Ometria product. They used various programming languages¹⁴, databases¹⁵, and message brokers¹⁶. We deployed in cloud environments¹⁷, and used modern monitoring tools and practices¹⁸. I ran data and infrastructure migrations¹⁹.
- I was the first manager in Engineering other than the founding CTO, managing four engineers. I facilitated one of my direct reports becoming the first employee in the company to transition to full-time remote work. I met with the technical due diligence auditors for our **\$21M Series B round**, managing their access to team accounts and providing various documents and materials.

¹³ they used Python, Postgres and various AWS services, running on Kubernetes.

¹⁴ primarily Python, sometimes Go or PHP.

¹⁵ a lot of Postgres (on RDS and Aiven), PgBouncer, Redshift for OLAP, DynamoDB.

¹⁶ Kinesis, SQS, Beanstalkd.

¹⁷ AWS, configured using Terraform. Some GCP. Mostly on Kubernetes.

¹⁸ Prometheus, Alertmanager, Grafana, OpenCensus tracing.

¹⁹ eg. a zero-downtime migration of our customer API to an Application Load Balancer, which allowed us to route URL paths to be served by particular microservices.

LETO were a small (~15 person) London-based software agency for all kinds of technical projects. I joined as a **Web Developer**, and was later promoted to **Senior Engineer**. I worked for clients ranging from brand new startups to established companies like *Camelot*, *Admiral*, and *Maxus Global*.

- I worked on a consumer mobile product²⁰ for *Camelot*, which streamed real-time data events from the lottery draw machines at Pinewood Studios during broadcast of the *Lotto* TV show on BBC One. I automated infrastructure²¹, designed load tests²² and implemented a monitoring solution²³.
- I was the primary engineer on backend systems for *Printt*, which at the time was an app providing printing services for students at UK universities. For about a year I was responsible for their on-site printing hardware²⁴, software, and backend applications²⁵, scaling from 5 to 35 printers at various locations in the UK. These efforts helped them secure an investment round, after which I led a handover process for their first in-house engineering hires²⁶.

²⁰ iOS and Android clients, but I worked on the backend. The app was a game that used audio watermarks in the TV stream to sync game state with the user's TV.

²¹ using Terraform to automate setup of AWS resources: VPCs, security groups, EC2, RDS, S3, Route53 records, Redis on Elasticache, etc.

²² I used loader.io to test Go HTTP endpoints.

²³ using Prometheus, Grafana, Sentry and Newrelic.

²⁴ these were Raspberry Pis running our Python applications using Supervisor. I wrote scripts that used Ansible to provision the Pi images. They used SSH tunnels to communicate with our servers.

²⁵ which used Python/Django, PyPy, Go, Postgres, RabbitMQ and Nginx, and ran on Ubuntu EC2 servers.

²⁶ LETO was their outsourced Engineering team in the early stages of the business.

OTHER

- I interned at *Musicmetric*, a music-industry data analytics startup who were later **acquired by Apple**. I wrote a report analysing geographical trends in their bittorrent datasets²⁷. They later expanded on this work as part of a marketing piece, which saw them receive **coverage on BBC news**.
- Pre-covid I spent a year **organising Emacs London**, a small monthly meetup focused around the programming/text-editing environment Emacs.
- Before I worked in software I received a **first-class BSc in Music Technology at the University of Kent**, which is where I started programming.

²⁷ using Python and matplotlib.