

Matt Duck

I'm a software engineer. Currently helping to run our team at [Way-bridge](#).

Since 2015 I've had various experience contributing to engineering teams, doing an IC -> team lead -> manager progression at three different B2B startups. My interests include:

- Product-focused software development, particularly web backend and distributed systems. I've worked a lot with **Python**, but can get by in other languages¹.
- Infrastructure / platform / "devops" engineering. I've primarily deployed on popular cloud environments like **AWS**², using **Linux**, **Docker** and **Kubernetes**. I enjoy monitoring production environments³.
- Relational databases, mainly using **Postgresql**.
- Building effective engineering culture and teams.

Updated: January 2022
Location: Remote / London
Email: hi@mattduck.com
Website: <https://www.mattduck.com>
Github: <https://github.com/mattduck>
Linkedin: <https://linkedin.com/in/mattduck>

¹ eg. Go, Javascript.

² usually automated by tools like Terraform.

³ using tools like Prometheus, Alert-manager, Grafana, Datadog, Sentry.

Waybridge 2020/04 - present

Waybridge⁴ is a New York/London startup using technology to transform the raw materials supply chain. I joined as the first **Software Engineer** on the UK side of the team, later worked as **Engineering Lead** for a year, and am now helping to run the Engineering org as **Director of Engineering**.

- I spent a lot of my first year building core components of the product across our data model⁵, API layer^{6,7} and cloud infrastructure⁸, contributing to hiring, and helping to establish good practices in Engineering⁹.
- In 2021 I was in a hybrid team-lead/manager role, leading one of our product engineering teams and managing five engineers across London, NY and Amsterdam.
- Since late-2021 I've been on the exec team and contributing to various aspects of running Engineering including org structure, working processes and hiring. On the technical side I'm leading our infrastructure efforts and backend architecture initiatives.

⁴ Started in Sept 2019 and originally named *CMDTY*.

⁵ Postgres and Python.

⁶ Python, Django, GraphQL.

⁷ also small frontend contributions using Typescript, React and Relay.

⁸ AWS provisioned using Terraform and deployed on Kubernetes (EKS) using Helm via Gitlab CI.

⁹ eg. I setup our core monitoring and alerting components using Python, Datadog, Opsgenie and Terraform.

Ometria 2017/07 - 2020/01

Ometria is a marketing SAAS product for online retailers. It drives revenue for brands like *Not On The High Street*, *Hotel Chocolat* and *Made.com*. I joined as a **Backend Developer** when the company had ~25 people, and saw it grow to more than 100 people. Through this time I was promoted to **Senior Developer**, **Lead Developer** and then **Engineering Director**.

- I was the main engineer on our data ingestion systems for 18 months. These were business-critical, realtime features¹⁰ that processed all the incoming ecommerce data points for all Ometria customers - up to 10 million records daily (orders, products, customers etc.).
- I worked on more than 50 separate production components or libraries - some as a contributor, and others as the main author. They used a variety of programming languages¹¹, databases¹², and message brokers¹³. Our systems ran in cloud environments¹⁴, and used modern monitoring tools and practices¹⁵. I ran various data and infrastructure migrations¹⁶.
- I was the first manager in Engineering other than the CTO, managing four engineers.

¹⁰ they used Python, Postgres and various AWS products, running on Kubernetes.

¹¹ primarily Python, but sometimes Go or PHP.

¹² a lot of Postgres (on RDS and Aiven). Some work on DynamoDB and Redshift.

¹³ Kinesis, SQS, Beanstalkd.

¹⁴ AWS, often configured using Terraform. Some GCP. Mostly on Kubernetes.

¹⁵ Prometheus, Alertmanager, Grafana, OpenCensus tracing.

¹⁶ eg. a zero-downtime migration of our public HTTP API to an ALB, which allowed us to route URL paths to be served by particular microservices.

- I met with the technical due diligence auditors for our [\\$21M Series B funding round](#). I managed their access to team accounts, and wrote documents that were specifically described in their reports as "thorough and excellent".

LETO 2015/02 - 2017/04

LETO were a small (~15 people) software agency for all kinds of technical projects. I joined as a **Web Developer**, and was later promoted to **Head of Engineering**. I worked for clients ranging from very small startups to bigger companies like *Camelot*, *Admiral*, and *Maxus Global*.

- I worked on a consumer mobile product for *Camelot*, which streamed real-time data from the lottery machines at Pinewood Studios during the *Lotto* show on BBC One. I automated infrastructure¹⁷, designed load tests¹⁸ and implemented a monitoring solution¹⁹.
- I was the main developer on backend systems for *Printt* (then named *AIWIP*), which at the time was an app providing printing services for students at UK universities. For about a year I was responsible for evolving and maintaining their on-site printing hardware²⁰, software and backend applications²¹, overseeing growth from 5 to ~35 printers at various locations in the UK. I helped them transition to an in-house team following an investment round.

Other experience

- I interned at **Musicmetric**, a music analytics startup who were later [acquired by Apple](#). I wrote a report analysing geographical trends in their bittorrent datasets²². 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 Emacs²³ (currently on hold).
- 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.

¹⁷ using Terraform to automate setup of AWS resources: VPCs, security groups, EC2, RDS, S3 buckets, Route53 records, Redis on Elasticache, etc.

¹⁸ I used loader.io to test Go HTTP endpoints.

¹⁹ using Prometheus, Grafana, Sentry and Newrelic.

²⁰ 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.

²¹ which used Python/Django, Go, PostgreSQL, RabbitMQ and Nginx, and ran on Ubuntu EC2 servers.

²² using Python and matplotlib

²³ Emacs is a programming/text-editing/computing environment and one of the "classic" programming text editors.