

David Purkiss

Software Engineer

The systems a business runs on, built with agentic engineering

Open to staff and principal engineering roles in backend, platform, data and AI infrastructure. Remote in the UK or Europe, or on-site in Edinburgh. Also open to relocating to the US, which would need visa sponsorship.

📍 Edinburgh, UK · 🏠 Indefinite Leave to Remain, no UK sponsorship required

✉ me@davidpurkiss.info

🌐 [Website](https://www.davidpurkiss.info) (https://www.davidpurkiss.info) [in LinkedIn](https://www.linkedin.com/in/david-purkiss-a8a26519) (https://www.linkedin.com/in/david-purkiss-a8a26519)

🐙 [GitHub](https://github.com/davidpurkiss) (https://github.com/davidpurkiss)

PROFILE

Nineteen years building software, most of it on the systems a business actually runs on: the ones where getting it wrong is expensive and hard to undo, and where separate systems have to agree with each other. The same pattern keeps repeating. I build the thing that doesn't exist yet, then own it once everything else depends on it. Data is the constant, from industrial time-series early on through analytics at Skyscanner to ingesting and matching messy data at volume now. Payments and ledgers came later, and they are where I learned what being wrong actually costs. I spent my first decade mostly in the front-end, moved back through full-stack into backend and platform work, and more recently into the agent infrastructure my teams work inside. We don't run a title ladder where I am, so I still carry the title I was hired with in 2022. It describes less of the work than the rest of this page does.

SKILLS

SYSTEMS & ARCHITECTURE

System design Distributed systems Event-driven architecture Microservices

DATA & SEARCH

PostgreSQL Elasticsearch Redis Apache Spark Apache Druid

PLATFORM & DELIVERY

AWS Kubernetes Docker Infrastructure as code CI/CD

AI & AGENTS

MCP & agent tooling

Agentic workflows

ML pipeline orchestration

LANGUAGES & FRAMEWORKS

TypeScript

Node

Python

C# / .NET

GraphQL

React

React Native

Vue

Nuxt

SvelteKit

COMMERCE & PLATFORMS

Stripe

Sage Intacct

Directus

Supabase

EXPERIENCE

HENI

August 2022 - Present

Senior Software Engineer

TypeScript, Node, Postgres, Directus, Elasticsearch, Redis, Kubernetes, Stripe, Sage Intacct

HENI sells artworks, books and merchandise, mostly through drops. An artist releases a body of work and it goes on sale at a fixed moment. When I joined there was no way to take a payment at all. I designed the order system and the Stripe integration it started from, and almost four years on I'm still the commerce backend's largest contributor, owning the data model, the money paths and the deployment substrate. Since 2023 I've split my time with ArtEye, our art-market data platform, which is deliberately the opposite problem. Commerce has to be exactly right because the mistakes are hard to undo; ArtEye is about volume and messy data.

- Grew that first checkout into the system the business runs on. It now reaches finance and accounts, warehousing and logistics, customer care, sales operations and drop allocations, with Stripe, Sage Intacct, HubSpot and Easyship behind it.
- Over 100 drops have gone out on the platform, for artists including Damien Hirst, David Hockney, Peter Doig and Brian Clarke, and for Ed Sheeran and Jack White. I built and run the release process behind them. Automating the data import and the migration between environments, and pulling what had been per-drop decisions into configuration, took a drop from three or four hours to put live down to about thirty minutes.
- Built the internal application 15 to 20 people across four teams operate in: sales raise orders directly, warehousing sees what is paid and ready to ship, and customer care can change addresses, contact details and line items, or refund through Stripe, without an engineer in the loop.
- I'm responsible for the money being right, from checkout all the way through to the ledger: invoicing, credit notes, cross-border VAT, AR reconciliation and deferred-revenue recognition into Sage Intacct. What finance needs is to trace any order to a booked transaction and be told when one doesn't reconcile, so I built the tooling that continuously verifies payments against ledger records, with the remediation paths to close what it finds. The data model is designed for that,

with an event-sourced order lifecycle and transaction tables mirroring the accounting ledger, so reconciliation falls out of the schema rather than being bolted on afterwards.

- The second platform is ArtEye, our art-market data system, and it is deliberately the opposite problem: artworks, artists, auctions, exhibitions and galleries, collected from scrapes, dealer feeds, PDFs and gallery visits, then matched, normalised and served. I designed the split it is built on. Two services with separate databases, so noisy high-churn ingestion never touches the read side, plus the activity-centred model the market data hangs off. It holds artworks and activities in the tens of millions, and grows daily.
- Built the pipeline over it: entity resolution against existing artists and organisations, plus the orchestration that lets ML stages run at ingestion volume, including OCR, image embeddings, AI tagging and vision-based cropping. Search is Elasticsearch, and reindexing it is zero-downtime and zero-loss. A versioned index builds while incoming updates collect on a paused secondary queue, then aliases switch atomically and the backlog drains.
- Made AI operational for the team rather than incidental: an MCP tool server exposing the production ERP to agents, agent access across Postgres, production logs and job queues, and an agent-oriented documentation system both repositories run on. The agent skills and MCP workflows that came out of it have since been taken up by two or three other teams.
- I'm who people bring designs to, including on work I'm not part of. That is a standing arrangement with the data engineering team from the ArtEye work, and with whoever owns a project elsewhere: they bring a design document or just a conversation, and often code review with it.
- Mentored two engineers over the past two to three years. One now owns most of what I used to and stands in when either of us is away, which was the point: too much of this used to be work only I could do. I also laid the foundations everyone works to, covering CI/CD, code review, the release process and the architecture, and those conventions have carried across the other teams. A fair amount of the job is removing blockers so everyone else can keep moving.

Truepill

November 2021 - July 2022

Senior Software Engineer

React, TypeScript, NestJS, GCP, RabbitMQ, Kubernetes, Netlify, Contentful, CircleCI

Eight months on a platform team for the [Optum Store](#), a high-traffic US pharmacy e-commerce site selling prescription medication, over-the-counter products and virtual care. React and TypeScript at the front, NestJS services behind it, running on GCP and Kubernetes. I left for HENI, where the commerce platform had not been built yet.

Skyscanner

September 2019 - October 2021

Senior Software Engineer

React, Redux, Java, Python, AWS, Apache Druid, Apache Spark, Experimentation

Skyscanner is where I saw what a startup looks like once it has scaled, and where I got my appetite for data back. I worked mostly on Travel Insights, a partner-facing analytics product for airlines, airports and destination marketing organisations: a

small number of customers, each of whom cared a great deal about the answer.

- Owned and delivered the initial release of Travel Insights Vision, our visual analytics product, over six months. A React and Node front-end onto data served by Apache Druid and Apache Spark.
 - Moved deliberately into a support role, helping others deliver, once it was clear I was becoming a knowledge silo. I had taken on a large share of my team's roughly 30 services when they were inherited from another squad during a re-organisation, learned them quickly, and become the team's go-to for technical and domain questions. Being the answer to everything is a bottleneck, not an achievement.
 - Built and migrated the flights and hotels checkout flows onto a new payment abstraction layer, a Java service fronting Braintree, ramping from a trickle to 100% of traffic through experiments over a few weeks. The booking flow it sat in was handling in the region of 100,000 bookings a month.
 - Took over optimisation and maintenance of the Apache Druid cluster the product's queries ran against.
 - Led the redesign of the Partner Portal's most-used page, introducing Redux for state management and bringing team members who hadn't used it up to speed.
- A team of seven owned all of the above. Alongside it I picked up Python properly for the first time, along with a good deal of AWS: Lambda, EMR, ECS, S3, IAM and CloudFormation, with infrastructure as code used across every service.

The Sun Exchange

December 2017 - August 2019

Senior Full-stack Engineer

NodeJS, TypeScript, GraphQL, Vue, Docker, PostgreSQL, AWS, Blockchain

I joined The Sun Exchange to take charge of the front-end and ended up owning a good deal more. The business crowdfunded solar cells on a blockchain: buyers own individual cells in a school or factory's array and get paid for the power those cells generate.

- Led front-end development of the company's ICO. We started building our own solution three weeks before launch, in parallel with a supplier we had stopped believing could deliver, and launched on ours. It raised in the low millions of dollars.
- Architected and built the rewrite of thesunexchange.com in Vue, backed by GraphQL and a headless CMS, once the ICO had proved the stack.
- Refactored the existing React and Next front-end into reusable components and onto a proper CSS framework, which is what made the fully automated crowdfunding rewrite possible.
- Built integrations across the stack, including Stripe, Paygate, HubSpot, GraphCMS and S3, through 20 to 40 solar crowdfunding rounds, each drawing a few thousand participants.

Forgood

November 2016 - November 2017

Technical Lead

.NET, C#, Cosmos DB, Aurelia, Azure, Team leadership

Taking over a platform that was being built by an outsourced development house, with no in-house team to take it over, was the hardest job I have had. I hired and ran a team of four while keeping business-critical delivery going.

- Cut hosting costs by 70% by moving the data layer off a self-hosted Couchbase VM onto Azure Cosmos DB. Before Couchbase 4.5 there was no ad-hoc query support, so application code pulled whole collections into memory and filtered them there; even once N1QL landed, common queries needed an explicit index to perform. Between that and shedding the IaaS, the move was the obvious call.
- Rebuilt the buggy ASP.NET MVC front-end in Aurelia, converting MVC controllers to Web API as I went, and addressed the load-time and SEO problems that surfaced after the first release.
- Recruited and managed the team of four that took the platform in-house.

Blue Nickel Solutions

December 2014 - November 2016

Lead Software Engineer

.NET, C#, Akka.NET, MongoDB, Distributed computing

A distributed computation engine and data store for the mineral processing and manufacturing industry, built to compete with Gensym's G2, Wonderware and GE's product suite. We folded several existing products into a new one built from the ground up on Aurelia, ASP.NET Web API, Akka.NET and MongoDB, with a microservice back-end. I managed a team of two and supported the rest of the business across its other products.

Flowgear

July 2014 - November 2014

Full-stack Software Engineer

SaaS, .NET, C#, WCF, Backbone, Silverlight, Integration

Blue Nickel Solutions

October 2010 - June 2014

Full Stack Engineer

.NET, C#, ASP.NET Web API, WPF, Silverlight, Cassandra, MVVM

AD Game Programming Courses

January 2010 - June 2013

Partner/Director

C#, XNA, Game Design, Game Programming

A hobby company teaching game programming in C# and XNA, run alongside my day job. We moved on from it in 2013.

Tao-Tech

November 2009 - September 2010

Software Development Consultant

.NET, C#, Syspro ERP, Joomla

Fusion Software

March 2007 - October 2009

Software Developer

VB6, VB.NET, C#, .NET, ERP, MS SQL, CRM

PROJECTS**NestTube**

Shipped on iOS. A parent-controlled replacement for YouTube: you whitelist the channels, kids request new ones for approval, and per-category time limits replace a single screen-time cap. Videos play through a private proxy, so no ads and no tracking. Built end to end: app, Supabase backend, subscriptions and release.

React Native, Expo, TypeScript, Supabase, RevenueCat, Sentry, NativeWind

[🔗 nesttube.app](#)

host-solo

A self-hosted application platform for a single VPS. Multi-environment deploys with network isolation, automatic Let's Encrypt certificates through Traefik, DNS record management and S3-compatible backups. Published to PyPI, and it runs my own production: the Directus backend behind Artilly deploys through it.

Python, Docker, Traefik, Let's Encrypt, DNS, S3

[🔗 github.com](#)

davidpurkiss.info

This CV. All content in one JSON file behind a Zod schema, with a local editing studio, a PDF exporter, and Markdown, JSON and TOON exports for anything reading it programmatically.

Nuxt 4, Vue 3, TypeScript, Zod, Playwright

[🔗 davidpurkiss.info](#)

Artilly Studio

My wife's art studio in Edinburgh, running classes for children. I built and maintain the site, including class bookings, payments and the invoicing behind them.

SvelteKit, Tailwind, Directus, Stripe, Brevo, Vercel

[🔗 artilly.co.uk](#)