Daniel Cooke

About me

I'm a software engineer from a small seaside town in Ireland. I've been writing computer programs for over 10 years now, my first language was Pascal, but throughout school and university I explored many different paradigms from Java to Prolog to x86 Assembly.

My "professional" experience is in web development, I have been working in-depth with React frontends and Node backends for the past 6-7 years.

Open Source work

- remix-sse
- remix-wizard
- omni
- cypress-autostub

Where I've worked

Fullstack Engineer

eXRt Intelligent Healthcare

January 22, 2022 - Present almost 2 years

React NextJS NodeJS AWS TypeScript Docker MongoDB

I was contracted by the founder and CEO to develop their MVP MedTech app ReSynk which assists in the rehabilitation of stroke patients.

I worked closely with the founder and their VR team to deliver a comphrensive and *reliable* API with accompanying dashboard application.

One of highlights of the project was the implementation of a *WebRTC* video conferencing feature that allows doctors to monitor their patients VR sessions remotely.

Due to the sensitive nature of the domain, *security* was an upmost concern from day one. I implemented a custom authentcation system that considers both the accesibilty needs of the users, as well as the mission critical data flowing through the system every day.

The app is currently live in production with several hospitals around the UK, and I continue to provide support and maintenance to eXRt.

Fullstack Engineer

FundamentalVR

4 months (2023 - 2023)

React Electron NodeJS AWS TypeScript

At FundamentalVR I was part of a team of 3 developers in charge of delivering a Electron (React) application to replace a legacy Unity application.

We had ownership of our serverless REST API along with several supporting services. We worked under intense time pressure and managed to deliver a high quality product in 3 months, complete with a comphrensive test suite and a reliable and available REST API.

Utilising async event systems wherever possible to reduce load on key webhooks and services.

Frontend Engineer

Poppins Agency

3 months (2023 - 2023)

React NextJS Tailwind CMS

As a Frontend Contractor at Poppins Agency, I crafted two visually striking marketing sites from scratch over three months. These sites featured intricate Figma designs and utilized CSS animations extensively.

My role encompassed setting up the Storyblok CMS and designing a monorepo of libraries to accelerate future app development. Noteworthy is my proposal to the core Storyblok team for an SDK aimed at expediting CMS configuration for developers.

Key technologies I employed include Next.js for dynamic web functionality, TailwindCSS for efficient styling, and TypeScript for code robustness. This setup facilitated smooth content management and site maintenance.

Throughout this engagement, I showcased an ability to deliver complex design concepts and optimize development workflows for lasting efficiency.

Solutions Architect

Point Topic

7 months (2022 - 2023)

Technical Leadership Architecture AWS

As a Solutions Architect, I lead a team of 3 junior developers to replace an outdated system with a brand new platform within a year. The main goals from my involvement were as follows.

1. To upskill the 3 junior developers so the business had an in-house team capable of building further products and delivering value without outsourcing.
2. To ensure a smooth delivery of their new subscriber platform within a timely fashion.

I provided ongoing support, architected the system on AWS and worked closely with C-levels and key stakeholders to ensure a timely delivery.

I introduced the team to a basic MERN stack, utilizing AWS for static hosting, Auth0 for authentication, and NestJS with MongoDB for the backend. This architecture was chosen to support future expansion and familiarize the team with foundational concepts.

I held a daily technical scrum where we focused on any blocker, I remained in close contact with the team over Slack and delivered bi-weekly end of sprint demos to the CEO.

Overall the outcome was extremely positive. In-fact the team have just recently developed an internal application for the business without my assistance!

Frontend Engineer

TerraQuest

about 1 year (2021 - 2022)

React NextJS GatsbyJS Azure Storybook styled-components

During my tenure at TerraQuest, I was an integral member of the frontend team responsible for creating a land referencing portal for the Northern Irish government. This initiative was part of a broader project spanning the entirety of the UK.

Recognizing the importance of **SEO for public-facing websites**, I proposed a strategic overhaul by transitioning the existing GatsbyJS frontend to Next.js (they had a lot of dynamic content that would be painful to deliver statically). To advocate for this transition, I spearheaded the development of multiple Next.js prototypes, effectively showcasing the advantages to stakeholders.

Additionally, I played a pivotal *role in enhancing the organization's component library.* This involved contributing complex components such as a headless Data Table and a Date Picker, both of which significantly enriched the library's capabilities.

Senior Software Engineer

IBM

almost 3 years (2018 - 2021)

React NextJS NodeJS Cypress Docker GraphQL Rust

IBM was my first major role after graduating from university, and what a learning experience it was. I was surrounded by some incredible engineers, on the QRadar UI team.

During my time working on the QRadar SIEM I worked with NextJS and NodeJS extensively. We mantained a complex GraphQL API gateway that connected the various QRadar Java services.

QRadar uses its own subset of **SQL called AQL (Ariel Query Language)**. One of my most notable achievements was building the QRadar Event viewer, a dashboard which allows users to perform AQL queries against millions of events ocurring across their organisation's network. The implementation had to gracefully handle failed requests. We use **Apollo Client to cache certain AQL GraphQL queries on the client for rapid response times.**

Another complexity with the application was the requirement for almost every screen/state to be shareable via URL. We had to encode complex UI state in the URL, I achieved this with a custom built react state → URL "compiler" and "decompiler". This meant that no matter how complex the current UI state was, the analyst could share their view with colleagues.

I used **Rust** only briefly to create a few CLI tools that allowed non technical users to spin up development versions of the app.

I received multiple promotions at IBM and had the opportunity to travel around the world and talk at several events on topics such as Docker and Cypress.

Working on a data-intensive system like a SIEM, our UI had to be very cognant and efficient with its data fetching using tools like graphql/dataloader to ensure we aren’t overfetching and implementing exponential backoff for failed requests. Testing was paramount on a system like this, and I developed several testing dashboards and utils to assist with our Cypress e2e testing. I had several placement students under my mentorship by the end of my tenure and thoroughly enjoyed every moment of this role!