Bin Hu

Senior Software Engineer

Perth, WA

+61 425 575 418 | elithrade@gmail.com | https://elithra.dev | elithrade | bin-hu-ab9035195

Summary

Software engineer with 20 years' experience, at RAC WA since 2022 and at Fugro for 16 years before that. At RAC I was tech lead on the digital membership card, where I decided the technical approach, wrote the solution intent, ran the threat modelling and led a team of three to launch. At Fugro I designed and built a live video streaming platform on AWS and led small teams on the Starfix.NG survey software. I treat programming as a craft. What I enjoy most is finding a simple, efficient solution to a problem, so I still write most of the code on the projects I lead, and in my own time I build side projects by hand and keep refining the Vim and terminal set-up I work in every day.

Skills

Languages	TypeScript, C#, C++, Python
Frontend	React, Next.js, Angular, Material UI, CSS
Backend	.NET, GraphQL, REST, Express, Entity Framework, SQL, DynamoDB
Cloud	Azure (Functions, API Management, App Service, Container Apps, Application Insights, Entra ID), AWS (Lambda, ECS/Fargate, S3, CloudFront, API Gateway, Elemental)
DevOps	Terraform, AWS CDK, Docker, GitHub Actions, Jenkins
Other	Contentful, Sitecore, Solr, Unity
Spoken	English, Mandarin

Education

University of New South Wales — *Sydney, NSW*

Bachelor of Engineering (Software Engineering), Honours — *2005*

Work Experience

RAC WA — *Perth, WA*

Senior Software Engineer — *April 2022 - Present*

- Tech lead of a team of three, and the myRAC team's technical representative, on the digital membership card that members add to Apple Wallet or Google Wallet. Decided the technical approach, wrote the solution intent, presented it for review by other teams' technical representatives, ran the technical alignment meetings and deployed the final solution. Within three months of launch, 96% of members who logged in to myRAC had added the card to a wallet.
- Produced the card's threat model and security design and ran the threat modelling sessions with the team and the other teams' representatives. In October 2024, taught the process to other RAC teams in an online session.
- Rebuilt the logged-in myRAC experience during RAC's move from Sitecore and ASP.NET pages to its Digital Experience Platform (Contentful, Next.js, GraphQL, Azure API Management, C# Function Apps). Designed the new product pages from UI components that hold no business logic, so the same components and page layouts serve roadside assistance, insurance, and finance applications and quotes.
- Started the move of the myRAC team's frontend, GraphQL layer and backend services into one monorepo.
- Led a team of three building the online cancel Roadside Assistance journey, which replaced a phone call to the contact centre, and wrote about 90% of its frontend and backend. Maintain and extend the team's other self-service journeys, digital roadside assistance (location-based patrol dispatch) and update my vehicle, with new features and accessibility fixes.
- Mentor the team's junior engineers: review the fixes and designs they propose and explain the trade-offs of the alternatives.
- On call 24/7 one week in every three as first responder to Azure P1 alerts, raised when an app's availability tests find a backend service it depends on is down: investigate, escalate to incident management and report status to them until it is resolved. When the public website, myRAC and the services behind them went down around midnight in October 2025, found the Azure portal unreachable too and traced the cause to a global Azure Front Door outage on the Azure status page. The post-incident review led RAC to revisit its disaster recovery plan.
- Reduced the Application Insights log volume, and with it the cost, of the self-service journeys app by removing unneeded info-level logs and using the request telemetry the platform already records (AppRequests) in place of console logging.
- Moved search on rac.com.au from Azure Search to SearchStax-hosted Solr: reindexed the site, changed the frontend to query the new engine, and secured access with Azure network rules and IP allowlists. Cached the first result set in the browser so that category clicks no longer re-run the search.
- Added registration lookup to the Autoservice booking app: upgraded the Angular app to a supported version, integrated a vendor's lookup API through the existing Azure Function App, and built the new step. Members now enter a rego instead of choosing make, model and year. Drop-off during booking fell by about 57%.
- Migrated the Roadside Assistance "change my cover level" journey to the new Azure backend: redesigned the UI and connected new upgrade and downgrade endpoints with the Microsoft Dynamics FinOps team, whose system holds the roadside products. Members make about 3,700 cover changes a year through it.

Fugro

Perth, WA

Senior Innovation Engineer, promoted from Software Engineer

June 2006 - March 2022

- Designed and built, as lead of a small team of three, a live video platform for remote ROV pipeline inspections. Owned it end to end: the design, most of the frontend and backend code, the AWS CDK infrastructure, the CI/CD pipeline, deployment and maintenance. Promoted to Senior Innovation Engineer after delivering it.
- The platform replaced a prototype that decoded video on a Node server in a local server rack and sent it over WebSockets, which could not host many live streams and had high latency. It runs entirely in AWS, and video reaches client browsers about 6 s after it arrives at MediaLive.
- Built it on AWS Elemental MediaLive/MediaStore with CloudFront signed cookies, a React/Material UI frontend, and an admin portal where operators manage projects and streams. Auth0 authenticated users, and a TypeScript/DynamoDB backend authorised them: which client organisation and projects each user belonged to, and which streams they could watch.
- Presented the platform's technology, lessons and results in a one-hour online talk to about 300 staff at the company-wide Fugro Innovators Conference, October 2020.
- Built the frontend of a data exchange for offshore survey data, my last project at Fugro: a dashboard that shows multibeam, pipeline inspection and other survey data as vessels upload it to S3 and lets data processors download it in batches, process it and upload the results (React, TypeScript). Started it from aws-serverless-spa-template, a personal React, Lambda and CDK starter published on GitHub, and set up its CI/CD pipeline.
- As Fugro modernised its stack from desktop software to the cloud from 2018, built the UI for a vessel activity reporting system, set up its CI/CD pipeline, and contributed to its API and AWS infrastructure. A daily Excel report of job progress and incidents, uploaded to S3, is parsed into a database through Entity Framework Core and served by a .NET Core API in containers (React, TypeScript, Docker).
- Worked with two other developers on the Unity application for a subsea pipeline inspection system developed by teams in Perth and Brisbane. ROVs recorded high-resolution photos and point clouds of the pipeline, an ML point classifier used both to identify freespans and damage, and the Unity app rendered the data for data processors to work through by hand.
- Worked on the Starfix.NG multibeam solution, which Fugro's survey vessels used to collect multibeam sonar data in the search for flight MH370 for the Australian Government from 2014.
- As Starfix.NG grew with plugins and new solutions, spent several years converting the monolith into a service-oriented architecture of core services such as hardware and geodesy. Led two to three developers, working mainly on the hardware service.
- One of the first developers on Starfix.NG, Fugro's in-house real-time survey data acquisition suite (C#, WPF), from 2007 to about 2012. It replaced Starfix, a set of small C++ programs linked by a message manager, with one package covering codecs, data monitoring, database logging and initial preprocessing. Took part in the initial design and owned sensor hardware initialisation, encoder and decoder set-up, and visualisation of the collected data.
- Started in 2006 building extensions to Starfix, the earlier C++ acquisition and processing software.

Personal Projects

url-shortener

github.com/elithrade/url-shortener

System design practice, written by hand

October 2026 - Present

- A URL shortener in ASP.NET Core (.NET 10) with Postgres and a Redis cache. I started it to stay sharp at writing code by hand, so I write the core logic myself and use AI as a reviewer and mentor.
- Handles code collisions, rate limiting with a token bucket, link expiry with a background sweep, click counting through a bounded queue with batched writes, and a cache that falls back to Postgres when Redis is down or slow.
- Load-tested with k6 on one machine: 10,000 redirects a second at a p99 of 1.5 ms with the cache and 5.3 ms without. The tests also showed that an app started cold under full load builds a backlog of requests, so I am now adding a warm-up step before the app reports ready.
- The in-memory and Postgres stores pass one shared contract test suite, and most tests were mutation-checked by breaking the code on purpose. Deployed to Azure Container Apps with Bicep and GitHub Actions. Each bug and experiment becomes a flashcard in LESSONS.md.

Development environment

github.com/elithrade

Dotfiles and Neovim configuration

2016 - Present

- Have kept my shell, tmux and editor configuration in git since 2016 and rebuilt it as my workflow changed: Vim, then Neovim, now LazyVim with C# and TypeScript language servers, debugging and test running.
- One command sets up a new WSL machine with zsh, tmux, Neovim and their tools, and is safe to re-run.
- This CV is LaTeX source in git, built and published by a GitHub Actions workflow.