

Andrew Dodd

an Aussie, living in Nijmegen

andrew.john.dodd@gmail.com

+31 683 774 336

adodd.net

github.com/andrewdodd

Experience

Snr Software Engineer [Planon](#) - Nijmegen Jul 2019 – now

- Designing new features and implementing bug fixes in a meta-data driven Java enterprise application for facilities management.

Independent Consultant **Remote** Feb 2018 – Jul 2019

- Designing and developing systems for clients in the mining industry, both remotely (NL) and onsite (Zambia, Indonesia, Aus).

Snr Software Developer [Adimian](#) - Brussels Apr 2017 – Feb 2018

- Agilely developed and delivered, on-time, a new multi-user Python [web application](#) for energy traders at Engie Belgium. It was used to manage the risk and value of Engie's hydro power plant portfolio.

"you are curious, ... you accept to work on an undefined scope, ... you implement fast so that we can test and iterate, ... one of the best consultants we've seen ... this project is a great success", Engie PM

Lead Software Engineer [APC Global](#) - Remote May 2015 – Apr 2017

Device Architect [Mine Site Tech.](#) - Sydney Jul 2014 – May 2015

Product Engineer [Mine Site Tech.](#) - Sydney Jan 2012 – Jul 2014

Business Consultant [IBM](#) - Sydney Jan 2010 – Jan 2012

Skills

Languages

- Python (Django, Flask, Pandas)
- JavaScript (React, Redux)
- Java
- C
- Golang
- SQL (PostgreSQL, SQLite, MSSQL)

Tools / Engineering

- Integration technologies (REST APIs, websockets, sockets)
- IoT / Infrastructure technologies (MQTT, InfluxDB, RabbitMQ, Redis)
- Software engineering (TDD / unit & integration testing, Design Patterns / architecture, concurrency & multithreading)
- Git, VIM, Linux, Jenkins/Hudson, Ansible, Docker, SSH

Other cool stuff

- I made a circuit solver [[link](#)] that uses gradient descent optimisation. I explain why in a 'tutorial' [here](#).
- Frustrated by changing requirements and inspired by Golang's templating package, I created [docopt-uc](#), an opinionated CLI generation tool for microcontrollers.

Education

Bachelor of Engineering (Computer Systems) / 2004 – 2009

Diploma in Engineering Practice (Hons.)

University of Technology, Sydney

University Medallist, 1st in year

Andrew Dodd

an Aussie, living in Nijmegen

andrew.john.dodd@gmail.com

+31 683 774 336

adodd.net

github.com/andrewdodd

Highlights

Lighting Control System (2018-19)

Golang / JS / MQTT

- I designed and built control software from scratch for mining startup [IoT Automation](#). Their product controls 1200+ smart lights in the underground block cave at the world's largest gold mine [[link](#)]. The system reduced emergency notification time from 20 mins to 1 minute.
- I built the frontend in JavaScript (React/Redux), and the backend in Golang. Integration between the frontend and backend use a combination of RESTful APIs and websockets, while backend used JSON over MQTT to integrate with the smart lights.

"...the software is superb...beyond my expectations...", Mine manager email

Labour Utilisation System (2015-17)

Python / JS / Golang / InfluxDB

- I designed, implemented and commissioned the product suite for [APC Global](#)'s labour utilisation offering. It used a diverse tech stack ranging from Python web applications, single-page JS applications, Golang and Python integrations, and embedded Java for RFID tagging.
- This included planning and assigning work within a remote team (Australia, South Africa, Zambia & Belgium), that included software developers, network engineers, communications techs and mine staff.
- The system allowed [Mopani Copper Mines](#) (Zambia) to improve safety compliance and reduce costs by \$1M USD in the first year [[link](#)].
- Successfully delivered after 2+ years of failure by other vendors.

Proximity Detection System (2013-2015)

C / Python / 802.15.4

- I was the 'technical product manager' and the lead developer of the radio communications on [Mine Site Technology](#)'s product to detect the proximity of people and vehicles around heavy vehicles. I was responsible for ensuring the design "held together" across a product that used three different microcontroller platforms; developed custom RF communications; and had to deal with demanding power budget and latency requirements. Oh, it also had to be robust and easy to use.
- Video demo of the product [[link](#)], so simple right ;-)

<http://decoda.cc> (2019-2020)

Python

- I worked on understanding and decoding J1939 frame payloads while working for Mine Site Technologies. The standard encodes information in a pretty cool way, but is a pain to actually use. I had always wanted to make a web service to decode it (most of the 1000+ pages), and I completed it over winter this year.

Talks / People / Ideas that have been influential

- All of Rich Hickey's talks, in particular [Simple Made Easy](#)
- Gary Bernhardt on TDD, testing in general, "[Functional Core, Imperative Shell](#)", and how to code like a boss
- Eric Evans & Greg Young on "Domain Driven Design" and "CQRS"
- Rob Pike & Russ Cox, in particular on [Versioning in Go](#)
- Martin Fowler / Alastair Cockburn / Robert Martin / Dave Thomas / Andy Hunt / Allen Holub / Simon Brown / Dan Luu / Julia Evans et al.