Wilhelm Van Der Walt

Snr Python Backend Engineer + Engineering Lead

Strong Eng Leader | Deep Technical Knowledge | Assertive Self Starter

I Can:

Top Skill

Maintain backend infrastructure

- Terraform
- Digital Ocean, Heroku etc
- Ansible
- CI + CD (Gitlab / Github / CircleCI)
- Docker
- Prometheus + Grafana
- Database mgmt (Postgres / MySQL / Neo4j)
- Ubuntu, CentOS

Build webservers that are:

- **Scalable**: Python WSGI frameworks, CPU + memory usage optimisation
- **Performant**: Advanced Python data structures, algorithm selection, threading, multi-processsing
- **Stable**: Unit + Acceptance Testing
- **Modular**: SOLID Principles

Build front-ends that are:

- **Javascript Based**: React or Vue
- **Python Based**: Jinja Templates
- **Modern**: ES6 and ES7 webpack or vue-cli transpiled javascript
- **Modular**: Separate UI into components that can be extended and re-used using Vue or React.

Lead or Work in a Team

- **Mentor:** Ensure the growth of Junior Developers
- **Lead:** Take responsibility for Projects
- **Monitor:** Provide well informed estimates and provide comprehensive progress reports
- **Adapt:** Things change. I ensure that change is managed responsibly.

The Proof:

Recent Projects

The Meos Personal Cloud (July 2018 - Current)

What it does: A platform that commoditises cloud computing in a way that allows anyone to run their own cloud applications like private email and private file storage. (https://meos.app)

How it was built: Flask webserver with a Vue front end. Celery workers process payments and create private servers using Ansible and Digital Ocean. Each private server runs a Flask webserver with an API that allows a user to access and extend their private applications. The docker-python api is used to manage the running containers in each private server.

My Role: As the only engineer on this project, I did everything. This included setting up the CI + CD pipeline, writing the Personal Cloud OS, implementing both front-end and backend, infrastructure maintainance and monitoring, amongst much else.

The Babylon Digital Twin (May 2017 - July 2018)

What it does: Provides a comprehensive and accurate assessment of a user's overall health including liklihoods of developing certain diseases. (https://www.babylonhealth.com/product/healthcheck)

How it was built: Flask and Nameko micro-services deployed on docker. Plenty calculation parallelisation using multi-threading and multi-processing where appropriate. This project also involved building a modular and dyanmic conversation builder that gracefully handled errors as well as things like dynamic unit conversion and natural language understanding (NLU).

My Role: I built the team and then lead the backend development. I was responsible for the overall system design and api design. I also took on challenging aspects of the development work, like increasing performance of the calculation stage and handling incoming free text in different languages like Chinese.

The Babylon AI Chatbot (Feb 2016 - May 2017)

What it does: A chatbot that handles user queries with the aim of diagnosing their illness or directing them to a service that can. (https://www.babylonhealth.com/product/ask-babylon)

How it was built: Micro-services built predominantly with Flask and Nameko that did everything from entity extraction, classification, ontological positioning and error handling.

My Role: I built up and lead the backend team. I worked with many technical stakeholders in order to architect the backend. I did a lot of the early development while recruiting more team members. (Some lessons learnt building this product can be found in this talk I gave at Europython '17 https://www.youtube.com/watch?v=U8JO7QUxvyQ)

Distant Projects

Babylon Triage
Dec 2015 - Feb 2016

Lead the iOS team in building Babylons first decision tree backed triage system

Babylon POC's
Aug 2015 - Dec 2016

Worked closely with the CEO to develop proof of concepts that informed the early roadmap.

Pearlshare
July 2014 - Aug 2015

Developed the iOS app of an experience based social network using functional reactive principles.

Technical Disciplines

- If tests are hard to write first, the requirements probably need some work.
- Metrics. If it can't be measured it can't be optimized (This goes for team metrics as well).
- Always be teaching. I only truly learn something when I can teach it.

Formal Education

- MSc Clinical Engineering, **1st Class honours**, City University London
- BSc Bio-medical Engineering, **1st Class honours**, National University of Ireland, Galway
- Hilton College High school, **Straight A's**, 2004-2008

va3093@gmail.com @wilhelmvdwalt linkedin.com/in/wilhelmvdwalt