Oscar L. Downing

oscar.lewis.downing@gmail.com | 0400 533 516 | www.linkedin.com/in/oscar-downing | Git: **qozymandias**

EDUCATION

UNIVERSITY OF NEW SOUTH WALES

B.E. (Hons) in Computer Engineering

Feb. 2016 — Aug. 2021 | Sydney, NSW

Honours thesis: "Enhancements to the Cogent Property-Based Testing framework".

Stephen Robjohns Memorial Rural Engineering Scholar.

SKILLS

PROGRAMMING

Proficient

Rust • C++ • Python • Bash
Typescript • Javascript • C

Intermediate

Java • Kotlin • Haskell • VHDL
Verilog • LaTeX • Isabelle

WEB & CLOUD

HTML/XML • JSON/YAML
CSS • React • WASM
AWS Boto • EC2 • RDS
JNCIA-Junos • JNCIA-DevOps

TOOLS & FRAMEWORKS

MongoDB • RocksDB • CUDA
GNU/Linux • Git • Docker
CMake • Cargo • Conan • GDB
pytest • async.io • Node.JS
SpringBoot • GTest • JUnit
Cucumber/Selenium • Ansible

EXP. CONT.

SOPRANO DESIGN | Graduate Software Engineer

Aug. 2021 — Dec. 2021

• Contributed to the development and QA of the MEMS platform, a Java web application (Apache Struts), tested with JUnit/Cucumber/Selenium.

JUNIPER NETWORKS | Software Engineer Intern

Feb. 2019 — Aug. 2020

• Developed modules (Python/Ansible) for Network Implementation and Test Automation (NITA) tool. Configured/deployed Warp17 traffic gen tool for scale test.
• Developed JSnapy scripts for automated testing. Upgraded static JunOS config files into dynamic Jinja2/YAML files.

EXPERIENCE

DELPHINUS LAB | Software Engineer

Dec. 2023 — Present

- Contributed extensively to the design, development, and optimisation of the zkWASM Playground, a cloud-based platform for zero-knowledge proofs of WebAssembly programs. Improved backend reliability, performance, and system observability (Rust/ActixWeb), and enhanced frontend integration (React/Typescript).
- Designed and delivered key features including Proof Continuations, Task Archiving, integration of new zkWASM versions, and RocksDB based Merkle storage. Developed associated handlers, implemented robust cleanup routines, and enabled graceful shutdown and recovery.
- Resolved critical MongoDB performance and memory issues through query optimization, index strategy redesign, client connection pooling, and efficient client resource management. Introduced request throttling, dynamic fee scaling for large proofs, and structured log rotation support.
- Led major architectural refactors to core components such as the session update mechanism, dry-run service, and auto-submit logic. Integrated cargo clippy across the codebase, eliminating unsafe patterns, panics, and duplication, substantially improving maintainability and code health.
- Strengthened testing infrastructure and documentation, expanding and improving the end-to-end test framework and unit tests. Added test coverage for edge cases (e.g., multi-prover synchronization, timeout handling), and authored scripting utilities for testing and debugging.

DOLBY.IO | Software Engineer

Dec. 2021 — Nov. 2023

- Contributed to the Dolby Voice Conferencing Server (DVCS) project, a highly performant multi-threaded audio mixing server written in C++ (std-17/Conan/CMake) for VOIP communications; used by the dolby.io platform, designed for conferencing and virtual worlds communication at high-scale — thousands of real clients connected and talking. Designed/developed/tested critical new features (Dist Attens).
- Developed supporting Python/Bash scripts for testing/CI/CD, including unit/system tests (GTest) and debugged complex multi-threaded problems in C++ with GDB. Improved OATs end-to-end testing framework (Python/async.io)
- Improved CI scripts (as DVCS shifted its CI to GitLab) and AWS scripts (Python/Boto) which saved thousands of dollars in EC2 computing costs. Additionally, built a Node.JS Typescript application for viewing CI/CD pipelines which interfaced with GitLab REST APIs.
- Improved documentation for onboarding new engineers. Mentored interns, actively learned new technologies, and presented talks internally.
- Upgraded build/production environment to Debian, rewriting toolchains (Conan/CMake) and fixing bugs uncovered by the new compiler; including rewriting packaging scripts and supporting Python scripts.

ATLASSIAN | Site Reliability Engineer Intern

Nov. 2020 — Feb. 2021

- Worked with Shard Capacity Management team on the Tenant Placement Service project: an internal AWS automation microservice, being developed to manage the RDS fleet (which hosts data from Jira and Confluence cloud products).
- Designed/developed automation of manual tasks relating to RDS life-cycle management. Creating new/deleting RDS instances and auto-scaling low storage RDS. Implemented tracking and reporting for internal audits of RDS events. Kotlin/SpringBoot, with a Postgres DB (using JPA/Hibernate) and REST APIs; Used in prod, saving the team manual effort and impacting 150 000+ customers.