

Michael Williams

+64 22 621 3204 — michael.williams.nz@proton.me — mw.net.nz — github.com/mwilliamsnz

Software engineer and physics PhD experienced in turning open-ended technically difficult problems into working software and computational systems. Background in end-to-end software ownership, rigorous technical evaluation, large-scale data processing, HPC, and C/C++ systems programming.

Experience

Software Engineer, AI Training & Evaluation – DataAnnotation

Feb 2025–Present

Remote contracting work for a US-based company providing training data and human feedback to frontier AI labs. Mostly software, some data science work.

- Designed and delivered end-to-end software systems in complex domains, owning requirements, architecture, technical decisions, testing, and final quality as part of detailed evaluations of frontier models.
- Performed detailed technical reviews against explicit and unstated requirements to find subtle failures in correctness, architecture, maintainability, and implicit product requirements; communicated precise evidence-based technical judgements of issues and root causes.
- Deliberately took on projects outside my existing stack for technical growth, delivering working software using Rust, relational databases, JS/TS, web technologies, and more, taking advantage of my autonomy.
- Evaluated a high volume of unfamiliar codebases across systems, database, web, and other domains, quickly identifying relevant assumptions, conventions, and failure modes.
- Built tools to surface maintainability issues after I observed recurring problems; proactively automated repeated processes, shortening iteration cycles.

Doctoral Researcher – University of Canterbury

2021–2025

Mathematical Physics PhD using high-performance computing for terabyte-scale simulation and analysis.

- Developed analysis and data processing workflows from exploratory Python prototypes to repeatable large-scale runs on international HPC systems, operating with substantial technical independence; produced reusable software that was used by several other group members as core parts of their research.
- Designed and ran multidimensional scaling studies to determine optimal node/core configurations for simulations handling tens of terabytes of data; balanced speed against compute cost, and research needs against resource constraints.
- Owned a multi-year computational research project with five collaborators. Set computational and scientific direction in a poorly-charted field, translating open-ended scientific questions into a well-defined project scope with tractable simulation and analysis targets.
- Mentored two MSc students, acting as their primary technical advisor, continuing to support one project after completing my PhD; their projects reused and extended my research code and results.
- Managed storage and compute budgets for our wider research group, balancing competing research project priorities with compute, data retention and transfer constraints.
- Communicated technical findings through published work and conference talks, including to non-experts.

Projects

Selected Contract Software Projects

Paid work at DataAnnotation

Closed-source for confidentiality reasons, but showcased at mw.net.nz/projects. These were all end-to-end projects where I had **complete ownership**, from establishing requirements to final implementation and testing.

- Built a Python library to parse specific section structures out of Google Docs, configurable for different formats and usable through an API or stand-alone CLI tool. Used this to build a dashboard to extract, compare, and track sections from multiple documents, persisting extracted sections and metadata using SQLite, processing records, and detecting and reconciling stale, migrated, or out-of-sync records.

- Built a SQLite-backed media-tagging tool, with a configurable tagging attribute model, user-friendly triage workflows, Bayesian probabilistic tag suggestions and an expressive custom search DSL. Used for a 10,000-image database.
- Built a Qt application for visualising spatial data over partially-ordered timelines, where the absolute order of events is ambiguous. Designed a domain-specific data model using a DCEL graph model to support edge refinement and annotation-preserving region splitting, with JSON persistence and sharing.

Selected Personal Projects

mw.net.nz/projects

C++ real-time graphics/ECS framework – Custom performance-oriented rendering and engine infrastructure

- Built a SDL3_GPU/Vulkan rendering pipeline for efficient bulk data transfer to the GPU and low draw call overhead, with shader and GLB model loading and an integration layer for an external UI library.
- Designed a data-oriented ECS with custom memory allocation focused on cache-local iteration and predictable in-memory ordering for improved performance.
- Benchmarked critical paths with nanobench, tested core logical components with doctest.

Long-running Java Minecraft mod – Seven-year maintenance and third-party API integration

- Developed and maintained a large, structured project with over 100 source files, complex domain models and performance-sensitive systems over seven years of intermittent development and maintenance.
- Integrated deeply with a large upstream codebase where the integration surface changed often, writing code robust to upstream changes/migrations and changes in the third-party API or mod dependencies.
- Designed a performance-sensitive simulation of procedurally growing structures spanning thousands of world elements, centralising state and updates in one controller with failover, component tracking, and heuristic removal of ineffective components.

Skills

Programming Languages – Primarily C, C++, Python, Java. Professional exposure to **Rust** and JavaScript.

Software/API Design & Data Modelling – API and library design, relational schema design, SQLite, document-style and domain-specific data models, JSON serialisation, DSL design, graph and spatial data structures.

Systems & Performance – Concurrency and multithreading, MPI, HPC/SLURM, HPC scaling studies, memory-aware and cache-aware C++ design, algorithmic complexity analysis, performance benchmarking.

Testing & Reliability – Rust's libtest, pytest, doctest, JUnit; unit testing, integration testing, regression testing, edge case analysis, rigorous code evaluation and review.

Tools & Data – Git. Linux, shell scripting. GNU Make, CMake, Cargo. NumPy/SciPy & pandas. Large-scale scientific data processing.

AI – Professional experience evaluating coding agents with rigorous independent verification of AI outputs; AI accelerating delivery while retaining ownership of architecture, technical decisions, and testing.

Awards and Scholarships

Roper Scholarship in Science – Highest ranked UC science doctoral scholar

Aho Hīnātore | UC Accelerator Scholarship – PhD scholarship

AG McLellan Prize – Best honours project in physics/astronomy

More – Aho Hīnātore Accelerator Project Award, Prof. C C Farr Memorial Scholarship, Haydon Prize in Physics

Education

University of Canterbury

May 2021–2025: PhD – Mathematical physics. Thesis: *Numerical Simulations in Relativistic Cosmology*. Submitted 2024, examined 2025.

2020: B.Sc. (Hons) – Mathematical physics. First class honours. **Final year GPA 8.67/9.00.**

2017–2019: B.Sc. – Double major Mathematics/Physics. CS courses to 3rd year, CS/CEng GPA 9.00/9.00