

Izaak Walton

Los Angeles, CA · izaakw@protonmail.com
linkedin.com/in/izaak-walton · github.com/Izaakwltn

SUMMARY

Software engineer who builds tools, compilers, and infrastructure that make hard technical work easier — from compiler backends that run experiments on real hardware to ML training stacks, full-stack apps, test frameworks, and developer tooling. Comfortable across the stack and close to the metal, an open-source maintainer, and focused on building things other engineers actually rely on.

EXPERIENCE

Software Engineer — Kreisleriana LLC

April 2026 - Present

- Built Alkan, a self-hostable reading app for EPUBs and PDFs from self-hosted libraries: a universal SwiftUI iPhone/iPad app shipped via TestFlight, backed by a from-scratch HTTP/JSON service in Common Lisp with books stored content-addressed and deduplicated by SHA-256.
- Developed an OMR that reliably reads sheet music: trained a CTC-CNN-BiGRU recognizer — fed by a synthetic-data factory that repurposes the project's music engraver for perfectly-labeled training data — that cut held-out Symbol Error Rate from ~ 0.55 to 0.24 and replaced a heuristic classifier producing empty output on every input.
- Optimized the in-house Common Lisp autodiff/ML library's training hot path to $\sim 3.9\times$ faster per step (9.0s $\rightarrow$ 2.3s/sample) through profiling-driven, type-specialized fp32 kernels and tape-graph optimization, and wired cuDNN batch-norm into the autodiff tape.
- Implemented three coexisting authentication modes over a single session layer — anonymous, an OIDC relying-party flow (PKCE, JWKS-validated JWTs), and local accounts with argon2 hashing and login throttling.
- Provisioned a three-node NixOS homelab (Colmena, disko, nixos-anywhere) and joined it to a cloud fleet as a WireGuard hub-and-spoke member with subnet routing across a 10.0.0.0/8 supernet.

Software Engineer I, II — HRL Laboratories

March 2024 - April 2026

- Built a compiler backend in Common Lisp, Coalton, and Python that generates assembly for a QICK (Quantum Instrumentation Control Kit) FPGA, enabling scientists to run generically-defined solid-state spin-qubit experiments on real hardware via VISA/SCPI.
- Implemented a compiler backend emulating hardware output to provide compile-time debugging information to experimenters, written in Common Lisp and Python.
- Diagnosed and resolved time-sensitive bugs across multiple layers of a large internal software stack written primarily in Python.
- Mentored an intern, setting goals, steering work, and helping them prepare for their final presentation.

Maintainer, Contributor — Coalton Lang (Open Source)

May 2023 - February 2025

- Added file and system tooling to the standard library, native stream utilities, byte-code generation, and environment-variable access replacing awkward Common Lisp interop.
- Added docstring support for structs, type constructors, and class methods, improving documentation across the standard library.

- Wrote benchmark and example programs, including a subset of the Gabriel Lisp benchmarks and a minimal esolang interpreter, to document the language for new developers and quantify ongoing efficiency work.
- Performed code reviews, debugged and triaged issues, and supported newcomers as an active maintainer.

Software Engineering Intern — HRL Laboratories

May 2023 – March 2024

- Built a testing framework for predicting floating-point error in quantum simulation, written in Coalton, Python, and Bash.
- Wrote a quantum circuit simulator in Coalton, using the language's type-class system to simulate circuits in parallel across multiple numeric types.
- Developed an Emacs mode for an internal pulse specification language, added to the team's default config.

EDUCATION

Master of Music, Violin Performance — University of Denver, Denver, CO

September 2017 – June 2019

Bachelor of Music, Violin Performance — University of Georgia, Athens, GA

August 2011 – May 2015

LPS Qubit Collaboratory's Quantum Computing Summer Short Course, 2025

SKILLS

Languages: Python, Common Lisp, Swift, Coalton, Bash

Machine learning: reverse-mode autodiff, CNN/BiGRU, CTC, CUDA/cuDNN, ONNX, data augmentation

Systems & tooling: Compiler backends, test frameworks, simulators, performance profiling, code review, NixOS, PostgreSQL, WireGuard, continuous integration, Git

Hardware / instrumentation: VISA/SCPI instrument control, QICK FPGA, hardware-in-the-loop emulation