

I'm passionate about pragmatic, rigorous software craftsmanship. In particular developer tooling and foundational systems (build systems, text editors, programming languages, databases, etc). I also enjoy photography.

Experience

Senior Software Engineer, Mercury Technologies (Portland, OR)

2022 - today

- Maintained ACH and check payment rails.
 - Designed and lead initiative to migrate vast majority of internal transfer transaction volume from wires to cheaper and more stable ACH payment rail.
 - Saved over a million dollars (and counting), decimated interrupt load, delivered ahead of schedule.
- Worked on data engineering for Risk.
 - Maintained our query scheduler and feature library storage layer.
 - Collaborated with Risk strategists to productionize analytical queries.
- Contributed to internal developer tooling (Haskell, Rust, Nix).

Software Engineer, Arista Networks (Remote)

2020 - 2022

- Created CI runtime to marry Nix build graph with Buildkite CI service (Haskell).
- Created [nix-graph](#) library to reify Nix build graph into in-memory graph data structure (Haskell).
- Maintained internal merge queue service (Haskell).
- Maintained internal package manager (Haskell).
- Maintained other developer tools, NixOS systems, and the Nix monorepo.

Intern Developer, Panoramic Software (Irvine, CA)

2018 - 2020

- Maintained data migration tool (Haskell).
 - Wrote migrations to manipulate JSON data in PostgreSQL databases.
 - Simplified business logic by factoring out effects from ad-hoc IO code.
 - Converted codebase to "[Three Layer Haskell Cake](#)".
- Built automated release tagging tool (Haskell).
- Worked in large codebase for production web app (Elm).
- Wrote web crawler to test properties of said web app (TypeScript, Puppeteer).

Projects

All these projects have been developed in my free time, and are published as open source software on GitHub.

- **indigo / blue** : Work-in-progress terminal-based modal text editor, inspired by Kakoune and Neovim (Rust)
- **fsm-playground**: Prototype finite state machine library, to define workflows for payment rails at Mercury (Haskell)
- **affection**: Toy effect system based on the freer monad (Haskell)
- **lambo / lang**: Work-in-progress interpreted programming languages, based on lambda calculus (Haskell)
- More examples on [my GitHub](#)

Skills

- **Languages**: Haskell, Rust, SQL, Nix, PureScript, TypeScript, Shell
- **Tools**: PostgreSQL, Nix, Git, Docker, Unix, Terraform

Education

2016 - 2019 | **Saddleback Community College**

Studied Computer Science

2012 - 2016 | **San Juan Hills High School**