# Daniel Hines

Software Engineer

I like systems programming, blockchains, and type theory. I aim for software that is reproducible, correct, efficient, and fun to hack on.

d4hines@gmail.com | +1 (757) 705-5492 | Virginia Beach, VA | @danhines09 | github.com/d4hines

## WORK EXPERIENCE

### Tech Lead for Deku Sidechain
Marigold

*04/2022 - 01/2023*

*Deku was a framework for writing high-performance sidechains for Tezos in any programming language.*

*Achievements/Tasks*

- Lead all technical aspects of the Deku project, including architecture, implementation, and code review. Second most active contributor behind the original Deku author.
- Grew Deku's performance from 1000 to 300,000 transactions/second.
- Lead documentation, community engagement, and ecosystem integration. Built showcases like Deku-Plays-Pokemon.
- Supported and trained my team (8 members total) in Deku's design, blockchain fundamentals, OCaml, and Nix.

### Blockchain Protocol Engineer
Marigold

*11/2020 - 04/2022*

*Worked on the Tezos blockchain and the Deku sidechain project.*

*Achievements/Tasks*

- Implemented the global constants feature of the Michelson virtual machine, dramatically increasing the potential size and complexity of smart contracts on Tezos.
- Authored a framework and eDSL for testing and benchmarking the Michelson interpreter.
- Migrated the Deku codebase to the pre-released OCaml 5 compiler with multicore support. Used the Nix package manager for fully reproducible builds.
- Designed and implemented async state hashing, a distinguishing feature of Deku's consensus algorithm enabled by our use of multicore OCaml.
- Overhauled our automation for developing, testing, benchmarking, and deploying Deku clusters, dramatically improving our DevUX.
- Prototyped "parametric VM's" for Deku, which became a core value prop for the Deku project.

### Software Engineer
Finsemble (now interop.io)

*10/2018 - 11/2020*

*Trading desktop automation framework used by the world's largest banks.*

*Achievements/Tasks*

- Core engineer involved in every level of design, implementation, and testing of a novel desktop manage.
- Championed several significant refactors, including adoption of ESModules, Typescript, Redux, and the Elm/FRP architecture.
- Developed mathematical models of the product's core features using the specification language ALM. Helped transition these models to working software.

## SKILLS

OCaml

Distributed Systems

TypeScript

Web Dev

Nix and NixOS

## PERSONAL PROJECTS

### Flamingo Lang

- Inspired by Inclezan and Gelfond's research on the action language ALM, I wrote Flamingo, an ALM compiler and runtime for reactive systems. Flamingo lets users specify business logic in a fully declarative and modular manner, compiling to WASM or native NodeJS modules via Rust.

### Git Anger

- My role as tech lead for Deku required me to work on many features at once, so I wrote a tool in OCaml that implements "stacked PR chains" to help me maintain any number of clean, atomic PR's simultaneously with minimal overhead. My teammates re-implemented the tool in Rust and use it to this day.

### Tezos Place

- I made a clone of Reddit's r/place as a Tezos Smart Rollup (the first to be used in production). The system processed over 600K transactions from 184 players over 24 hours. I compiled the result into an interactive NFT that auctioned for 101ꜩ. Players described it as "the most fun they've had on Tezos since [the NFT boom of] 2021".