

Alireza (Parsa) Ghadimi

Senior Software Engineer

San Francisco, CA

✉ hey@parsa.wtf ☎ +1 917 420 2781 🌐 parsa.wtf 🐙 [qti3e](https://github.com/qti3e)

Writing code for about 15 years. Currently at Deno (the JavaScript runtime), working on Deno Deploy. Before that, CTO at Fleek, a decentralized hosting platform where I led 30+ engineers for 4 years (raised \$25M). My first job was working for Ryan Dahl in 2018 on PropelML—machine learning tooling for JavaScript. That year I was an early contributor to Deno before it went public.

Work Experience

System Engineer

Deno

02/2025 – Present

Remote

Working on Deno Deploy.

Chief Technology Officer (CTO)

Fleek

05/2024 – 12/2024

New York, USA

Decentralized app hosting platform that raised \$25M in Series A.

- Led migration of backend infrastructure off cloud platforms.
- Shaped technical strategy and company direction.
- Led a team of 30+ engineers.

Lead Software Engineer

Fleek

02/2023 – 05/2024

New York, USA

Led the development of Fleek Network.

- Designed and built distributed systems architecture.
- Created high-performance incremental Blake3 protocols.
- Implemented Raft consensus algorithm.

Senior Software Engineer

Fleek

02/2021 – 05/2022

New York, USA

Decentralized infrastructure platform for developers building Dapps.

- Contributed to product ideation and roadmap planning.
- Developed open-source tooling to solve internal engineering challenges.
- Led the development of several projects including XTC, Cap, IC-Kit, Sly, etc.

Software Developer

PropelML

03/2018 – 07/2018

Remote

Machine Learning in JavaScript. Worked directly with and reported to the original creator of Node.js, Ryan Dahl.

- Refactored the online notebook (Preact).
- Ported numpy's tensor formatter from Python to TypeScript.
- Created a value inspector like the one in Chrome Devtools.
- Created a data-serialization protocol to transfer JS objects with circular references across web workers.

Compiler Engineer

Truebase

12/2018 – 07/2019

Toronto (Remote)

Platform aiming to revolutionize software creation.

- Implemented compiler backend and developer tooling.
- Contributed to project roadmap and technical direction.
- Researched graph databases, with focus on DGraph.

Technical Skills

Languages: Rust, Go, C, Python, TypeScript, JavaScript, x86-64, Bash, ...

Runtimes & Internals: V8, rustc, rust-analyzer, Go internals, Deno, Node.js

WebAssembly: Binary format, Wasmtime, Cranelift

Linux & Virtualization: eBPF, io_uring, rseq, KVM, Firecracker, gVisor, libublk, NixOS

Low-level: ELF binary format, linkers, SIMD (AVX2, AVX-512)

Compilers & Analysis: E-graphs, abstract interpretation, symbolic execution, SAT solvers, LLVM, tree-sitter

Cryptography: Threshold ECDSA, zero-knowledge proofs, FFTs, erasure coding

Distributed Systems: Raft, CRDT, P2P networking, anti-entropy protocols

Specs: ECMAScript, HTTP caching (RFC 9111)

Currently Exploring

Machine learning infrastructure, CUDA, PTX, NCCL, llama.cpp

Selected Projects

Dfinity tECDSA Optimization

2022

- Collaborated with Victor Shoup at Dfinity to improve threshold ECDSA performance.

Deno (Early Contribution)

2018

- Contributed to the initial development of Deno before it was published on GitHub.
- Acted as a moderator and helped maintain the repository in its early weeks.
- Fun fact: Made the first non-Ryan Dahl commit to Deno at late 16.

Cranelift PTX Backend for Rust

Ongoing

- Building a PTX backend for Cranelift and integrating it into rustc as part of exploring GPU toolchains.
- Currently compiles basic Rust functions to PTX text for NVIDIA GPUs.
- <https://github.com/qti3e/wasmtime/commit/dc3141b> — <https://github.com/qti3e/rust/commit/671a0e2>

JavaScript Optimization Engine (JOE)

Ongoing

- Long-running personal experiment: a JavaScript AOT compiler written in Rust.
- Zero dependencies—no std, no alloc, no libc, no libm, no compiler-rt, no Cargo.
- Features: symbolic execution, e-graph-based IR, aggressive static analysis.
- Goal: if your server-rendered React app doesn't compile down to string concatenation, that's a bug in React.

Blake3 JS

2024

- A WebAssembly JIT written in JavaScript that runs the Blake3 hash function.
- Achieves 2.2x the speed of native WASM execution.
- <https://parsa.wtf/blake3>

IC Kit

09/2021 – 12/2022

- A SDK for Dfinity’s Internet Computer, built to facilitate automated testing and make Smart Contract development safer.
- Attracted community adoption and is actively used across many projects.

Dank’s Cycles Token (XTC)

05/2021 – 10/2021

- A utility token for the Internet Computer that wraps the native Cycles token.
- Heavily used service allowing users to hold +\$1M worth of cycles tokens.

Cap Open Internet Service

09/2021 – 12/2024

- An actively used open internet service for Dfinity’s Internet Computer.
- Allows smart contracts to hold transaction history/logs.

Markus

12/2019 – 08/2021

- Designed a declarative query language and universal ORM for safe & secure data-driven software.
- Created the parser on top of tree-sitter (Rust + C), implemented an SAT-solver with IR optimizations.
- <https://github.com/Markus-Studio/Markus>

Slye

05/2018 – 07/2019

- Open-source cross-platform 3D presentation software written in TypeScript using React and Three.js.
- <https://github.com/qti3e/Slye>

Interests & Philosophy

Rust. TypeScript when necessary. Making things fast.

I like Go’s philosophy more than I like writing Go. I like Nix because I want my computer to be a deterministic function of its configuration. I like reading source code—I have V8, LLVM, the Linux kernel, glibc, musl, GHC, and four SAT solvers cloned locally. Just in case.

Code I Study

awk, BLAKE3, bloaty, busybox, carbon-lang, courgette, ecma262, egg, engine262, esmeta, firecracker, ghc, glibc, gmp, go, gvisor, jemalloc, kissat, libc, librseq, libublk-rs, liburing, linux, llama.cpp, llvm-project, lmdb, mold, musl, nanos, oxc, plan9, qemu, quickjs, regalloc2, rust, rust-analyzer, splr, tcmalloc, test262, tokio, v8, varisat, wasmtime, z3

Awards

South Korea Science & Engineering Fair Medalist

09/2019

Korea Science Service – “WaterScript”: A JavaScript optimization engine and runtime simulator.

Best Lecture in Computer Science & Math

2019

Avicenna Research Center – Lecture on “Dead Code Elimination in Dynamic Languages using Execution Simulation”.

3x Gold Medal – Festival of Student Achievements

2017–2019

Tebyan – First-place award 3 years in a row, each with a different project.