Paul (PJ) O'Leary Jr.

FULL STACK ENGINEER

peej@oleary.com | (415) 450-7038 | San Francisco, CA
linkedin.com/in/pjol | github.com/pjol

TECHNICAL SKILLS

Languages: GoLang, TypeScript, Solidity, Rust
Databases: PostgreSQL, MySQL, Supabase, MongoDB
Client-Side: React.js, Next.js, HTML & CSS
Cloud & Deployment: AWS, GCP, Vercel, Bare Metal, Privy
Server-Side: REST API, MCP, OAuth 2.1 / PKCE, Applied ZK
CS Fundamentals: DSA, Cryptography, Blockchain

PROFESSIONAL EXPERIENCE

SFLUV | Founding Engineer — **Apr '23 – Present**

A USD-backed community currency rewarding volunteerism in San Francisco

- Shipped an MCP server giving AI agents live access to financial, merchant, and tax reporting — 11 read-only tools spanning three PostgreSQL databases, hardened with statement timeouts, row caps, and curated column selection
- Implemented a spec-compliant OAuth 2.1 authorization server (PKCE, dynamic client registration, hashed single-use refresh-token rotation, per-request admin re-checks) bridging the app's existing Privy identity to MCP clients
- Architected and executed a zero-user-action migration of the currency from Berachain to Celo: upgradeable UUPS contracts (Solidity/Foundry), snapshot-and-reissue distribution to all holders, and deterministic CREATE2 address parity so every user's smart wallet kept its address across chains
- Built a resumable, idempotent Go migration orchestrator with a Next.js operator console, dry-run production audits, an anvil-fork rehearsal harness, and a cross-chain history backfill preserving ~$100K of transaction records for continuous tax reporting
- Designed and implemented fully integrated merchant portal with wallet and admin functionality; deployed Ponder instance for blockchain activity indexing & webhook triggers

OPEN SOURCE & PROJECTS

go-chi/chi | Open Source Contributor | GoLang — **Jul '26**

Merged contribution to chi, a 22k+ star Go HTTP router used at Cloudflare and Heroku · github.com/go-chi/chi/pull/1129

- Authored the merged implementation of routing support for the new HTTP QUERY method (IETF RFC 10008), designed for a clean handoff once native support lands in Go's net/http

Bracket | Founder / Full Stack Developer | Next.js – TypeScript — **Jun '26**

Configurable tournament operations for local events · bracket.sh

- Built and launched a tournament operations platform with a public tournament board, competitor registration, and live bracket management
- Ran a successful live tournament end-to-end through the platform

BirdFeeder | Full Stack / Smart Contract Dev | JS – GoLang – Solidity — **Mar '25**

A marketplace for delivery services, powered by Self Protocol · devfolio.co/projects/birdfeeder-6ef6

- Utilized Self Protocol to create authenticated Order Book contract, and devised a market protocol incentivising participating actors to compete as reputation providing services
- Won 1st prize for EthSF Hackathon Self Protocol bounty, and finalist in main track

Cali Creds | Full Stack / Smart Contract Dev | Rust – Solidity — **Oct '24**

An on-chain identity solution using Succinct's SP1 and the California DMV's mDL · github.com/pjol/Cali-Creds

- Devised mdoc credential validation scheme using SP1 to generate proofs, with microservice architecture for scalable proof generation

EDUCATION

Hack Reactor | Advanced Software Engineering Immersive Program — **Sep '23 – Jan '24**

- Coding training in JavaScript, React, Express, PostgreSQL, MongoDB, and MySQL in an Agile environment, including solo coding, pair programming, and team collaboration

University of British Columbia | Undergraduate Study — **Sep '20 – Dec '21**

- Coursework in psychology & information studies