

# Joshua Sells

St. Michael, MN | [joshuataylorsells@gmail.com](mailto:joshuataylorsells@gmail.com) | [joshuasells.com](http://joshuasells.com) | [linkedin.com/in/joshuasells](https://linkedin.com/in/joshuasells)  
[github.com/zeddiushq](https://github.com/zeddiushq)

## Summary

---

Software engineer with five years of experience building high-scale data and event systems on Google Cloud. I design systems end to end, tune them with measurements, and own them in production. Outside of work I build whole systems on my own, including a Rust API with authentication written from scratch, an OAuth 2.0 and OpenID Connect identity platform, and an on-chain arbitrage system. Most comfortable in Rust, Go and Java.

## Experience

---

**Engineer 2**, Best Buy – Richfield, MN May 2024 – present

- Architected and owned the Go rewrite of the event ingestion API, replacing the Java service. It takes thousands of requests per second and tens of thousands at peak. At the same traffic, instance-hours dropped 50% and mean latency fell from 7.65 ms to 1.60 ms. Cold start improved about 5x, which let the fleet scale in, and a bounded in-flight semaphore applies backpressure instead of dropping load.
- Built a stateful Apache Beam streaming pipeline that flags bot traffic at thousands of events per second on about one worker. I shipped a working first version, then followed up with an inverted index, so per-event cost no longer grows with the number of rules.
- Designed Data Surge, a config-driven ETL platform where new pipelines are added with configuration and a SQL file instead of a new service. It runs hundreds of executions a day at 99.99% success, and its output table is the production read path for a downstream export.
- Led cost and capacity reductions across the platform, including about 80% fewer BigQuery slots on the heaviest pipelines with identical output, unused Pub/Sub subscriptions removed, and half the instances needed to run the event ingestion API.
- Moved about 25 services from Jenkins to GitHub Actions reusable workflows. Wrote the Java workflow family and its documentation, and was the top reviewer in the shared repository.

**Engineer 1**, Best Buy – Richfield, MN Mar 2022 – May 2024

- Architected Egress V3, a batch delivery platform that now moves billions of records a week at 99.8% success.
- Built a JSON Schema data-contract system that validates live production events against inheritable contracts.

**Software Engineer (contract)**, Best Buy – Richfield, MN June 2021 – Mar 2022

## Projects

---

**Zeddius (in progress)** 2026 to present

- Solo build of a personal tracking platform. The Rust (Axum) API is live on Cloud Run with authentication written from scratch, including Argon2id password hashing, opaque tokens, email verification, password reset and Sign in with Apple. Web and iOS clients are in development. The source is public on GitHub.

**Rabbit** Oct 2023 to Feb 2026

- Co-founded and co-built an OAuth 2.0 and OpenID Connect identity platform with PKCE and role-based access control, plus a 25-component Material UI design system packaged as a private npm package.

## Crypto arbitrage system

Dec 2021 to Jun 2023

- Led a two-person build of an automated arbitrage system across Ethereum, BNB Chain and Polygon that captured two profitable automated trades, on Polygon in 2022 and on Ethereum in 2023. An offline pipeline narrowed 214,061 pools to 2,749 and enumerated 224,124 cyclic paths, so each swap event is one lookup.
- Wrote the optimizer, including an exact integer port of the Uniswap V3 swap math in Java, plus the Solidity executor. Transactions went to block builders so failed attempts cost nothing.
- Full write-up, source and trade record. [joshuasells.com/projects/arbitrage](https://joshuasells.com/projects/arbitrage)

## Education

---

Concordia University, St. Paul, BS in Computer Science

Aug 2019 – Aug 2021

North Hennepin Community College, AA in Liberal Arts

May 2017

## Certifications

---

Google Cloud Certified Professional Data Engineer

Feb 2026 to Feb 2028

## Skills

---

**Languages:** Rust, Go, Java, TypeScript, Swift, SQL, Solidity

**Frameworks:** Axum, Micronaut, Apache Beam, React, Next.js, SwiftUI

**Cloud and data:** GCP (Cloud Run, BigQuery, Dataflow, Pub/Sub, Workflows), PostgreSQL, Docker, Kubernetes

**Practices:** System design, load testing, performance tuning, CI/CD (GitHub Actions, Workload Identity Federation), OAuth 2.0 and OpenID Connect, role-based access control