JONATHAN FARRO

Senior Software Engineer · Backend Systems & Data Platforms

San Diego, CA · Open to remote | 352-231-2504 | Jon.farro@gmail.com
linkedin.com/in/jonathan-farro | github.com/jarquet | jarquet.net

SUMMARY

Senior software engineer with 15+ years designing and running backend services and data platforms in Python, C#, and Rust on AWS. First engineer and technical lead of a solar-analytics team that grew to eight, delivering the reporting platform behind 130 utility-scale plants and 3–5 GW of generation. Most recently built production Temporal.io workflow services handling up to 20,000 requests a day for a ticket-resale marketplace. Owns systems end to end, from architecture and data modeling through APIs, deployment, and on-call, and now builds with Claude Code, MCP, and agentic workflows.

TECHNICAL SKILLS

Languages	Python, C#, Rust, TypeScript/JavaScript, SQL, Bash
Backend & Data	FastAPI, Django, ASP.NET, Temporal.io, REST API design, Pandas, NumPy, SciPy, PostgreSQL, SQL Server, DynamoDB, Redis, ZeroMQ
Cloud & Infra	AWS (EC2, ECS, EKS, Lambda, SQS, SNS, S3, RDS, DynamoDB, Cognito, CloudFormation, CodePipeline, CodeBuild, CloudWatch), Kubernetes, Kong, Azure DevOps, Datadog, Linux
Front End & Tools	React, Django templates, Retool, Puppeteer, Git
AI-Assisted Dev	Claude Code, Cursor, GitHub Copilot, MCP, agentic workflows

EXPERIENCE

Independent Software Projects
Feb 2026 – Present

Self-directed · San Diego, CA

- Built two **algorithmic trading applications** paper-trading through the Alpaca API: one applies machine-learning models to price structure to emit trade signals; the other uses Claude Code scheduled tasks to run fundamental and pattern-based analysis across S&P 500 tickers three times each trading day, grades its own past calls, and keeps a lessons-learned history in the repository.
- Building a **training journal for ADA service-dog handlers**: shipped as a React 19 / TypeScript single-page app with local storage, now being rebuilt as a native Android app in Rust.
- Wrote a Texas Hold'em poker game (Python CLI and web) with full betting rounds, hand evaluation, and a computer opponent.

Senior Software Engineer
Mar 2022 – Feb 2026

Victory Live

Ticket-resale marketplace integrating primary and secondary ticketing platforms; 70–100 person engineering organization.

- Designed and built two greenfield **Temporal.io** workflow services (Python, FastAPI) to replace unstable legacy integrations, deploying them to Kubernetes through the shared CI/CD pipeline and supporting them in the on-call rotation:
  - **Ticket fulfillment API:** an async translation layer that lets internal systems retrieve mobile-wallet (PKPass) tickets from a third-party ticketing API; 1,000–20,000 requests per day.
  - **Account password-reset service:** Puppeteer automation that completes the vendor's reset flow, retrieves one-time codes through an internal email-monitoring API, rotates to a random password, and stores it in the password manager; 20–2,000 resets per hour for downstream systems that depend on valid credentials.
- Built an automated **event-matching pipeline** that joins catalog data from 10–12 ticketing sources (thousands of events per month, 2–8 source IDs each), raising the automatic match rate from the low 20s to the mid-40s percent and reducing manual matching across thousands of event IDs; also delivered a Retool interface and backing API for analyst-assisted matching.

- Cut EC2 spend by more than **$100K per year** with no service interruptions by replacing fixed worst-case provisioning for an SQS-fed ECS worker fleet with a Lambda autoscaler that sizes the Auto Scaling group from queue depth and CloudWatch load history.
- Maintained and extended Python and C#/ASP.NET ticket-exchange and marketplace-integration services deployed across AWS and Azure.

  Tech: Python, FastAPI, Temporal.io, C#, ASP.NET, Puppeteer, Kubernetes, AWS (ECS, EKS, Lambda, SQS, S3, RDS, DynamoDB, CloudFormation, Parameter Store), Azure DevOps, Datadog, Retool

Senior Software Developer & DevOps Engineer · Team Lead
Oct 2014 – Mar 2022

SOLV Energy

Operations and maintenance provider for utility-scale, NERC-compliant photovoltaic power plants.

- First software engineer on the team and its **technical lead** as it grew to eight: interviewed every engineering hire (including two more experienced engineers), set technical direction on languages, frameworks, and build-versus-buy decisions, mentored junior engineers, and made nearly all architecture decisions.
- Architected and built the **SOLV reporting platform** from the ground up, growing automated report delivery from dozens of hand-built reports per month to thousands: daily reports to 250 field technicians, multiple daily reports to 10 clients, and monthly reports to 30 clients across 130 utility-scale PV plants totaling 3–5 GW.
- Built ingestion and analytics pipelines processing **hundreds of GB of raw sensor data per day** into PostgreSQL and SQL Server for real-time and post-processed analytics (Python, Pandas, SciPy, ZeroMQ, Redis).
- Replaced a 20-minute Python resampling prototype with a **Rust Lambda** triggered on S3 upload that downsamples 10 Hz data for thousands of sensors per plant-day to 5-minute intervals in under 20 seconds.
- Delivered web-based diagnostic and situational-awareness tools (Django) that combine multiple REST APIs and relational sources for plant operators and technicians.
- Owned deployment, CI/CD (CodePipeline, CodeBuild, CloudFormation), and on-call for the services built.

  Tech: Python (Pandas, SciPy, Django, FastAPI), Rust (tokio, serde), AWS (EC2, Lambda, S3, SQS, SNS, RDS, DynamoDB, Cognito, CloudFormation, CodePipeline, CodeBuild), PostgreSQL, SQL Server, ZeroMQ, Redis

Bioinformatics Programmer & Web Developer
Aug 2011 – Oct 2014

MATRR.com · Baylor University (remote)

Monkey Alcohol Tissue Research Resource: research-data platform and tissue bank for an alcohol-research consortium of 3 institutions and 12 investigators.

- Led all development of the MATRR platform as its sole engineer (at times with one graduate student): a **Django** application running tissue request, review, shipment, and inventory workflows plus a Pandas-based research-data platform, including system administration, QA, and releases.
- Built the ingestion, statistical-analysis, and visualization pipeline (Pandas, NumPy, SciPy, Matplotlib) for longitudinal experiment data: 12 cohorts of 8–15 subjects with weekly blood-ethanol measurements and event-level consumption data over 2–3 years each.
- Co-authored two peer-reviewed articles in Alcoholism: Clinical and Experimental Research (2014), contributing the data analysis and figures, and presented the platform at the Research Society on Alcoholism annual meeting (2011).

  Tech: Python, Django, Pandas, NumPy, SciPy, Matplotlib, PostgreSQL, Linux

Co-founder & Chief Engineer (part-time)
2009 – 2013

FBS Holdings, LLC

- Co-founded an exempt commodity pool operator trading currencies and derivatives; owned all engineering part-time while completing a degree and working at Baylor.
- Implemented real-time algorithmic trading strategies and a black-box execution platform in C#, a distributed backtesting framework (C#, Django), and web-based analysis tools (Python, Django).
- Scaled backtesting from a handful of runs per day on laptops to a cost-constrained EC2 compute farm running genetic walk-forward portfolio optimization behind REST services.

EDUCATION

B.S., Business Administration (Finance) · University of Florida 2011