Paul Amonson

Locations: Remote, Spokane, WA | Email: paulamonson@gmail.com

Summary

Accomplished Senior Software Engineer with over 20 years of experience designing and implementing software across diverse platforms, including Unix/Linux, Windows, and Mac/iOS. Proven expertise in middleware development, server-side components, REST API design, and observability/telemetry pipelines (OpenTelemetry), alongside Linux services and GUI tools. Adept at optimizing cloud software for performance, developing scalable HPC control systems, and contributing to industry standards. Proficient in Java, Python, Swift, C++, TypeScript, and more, with a strong focus on Agile methodologies and test-driven development. An early, hands-on adopter of AI-augmented development, using large language models to accelerate debugging, security review, and CI/CD optimization while applying disciplined human oversight to their output. Recognized for driving innovation, fostering cohesive teams, and delivering high-quality, secure, and efficient software solutions.

Core Proficiencies

- Use of AI in All Aspects of Development
- Multi-Language Use of OpenTelemetry (OTel)
- REST Backend Development Servers
- Programming Expertise
- Middleware Development
- Software Design & Architecture
- Team Collaboration & Leadership
- Data Engineering & Analysis
- Cross-Platform Development
- Personal Growth & Innovation
- Security Standards & Compliance
- Agile & Test-Driven Development
- Development Tools & Frameworks
- High-Performance Computing (HPC)

Professional Experience

Senior Software Engineer **4/2025 to 8/2026**
Anaconda Inc.

As a Senior Software Engineer, I was responsible for designing a data schema for telemetry collection and developing library wrappers for multiple languages for a common API across the company. I was also responsible for adding multiple new features to the core backend REST API used in the company for resource management (Python packages, AI models, common environment configurations, etc.).

- For the telemetry-related responsibilities, I built the schema for the backend storage, wrote the libraries for use in company applications, stood up OTel collectors to receive the telemetry, and exporters to save collected data into a database. Languages included Python, TypeScript, and Rust. Golang was on the roadmap. Early in my time at the company, I used AI to find bugs and security issues, and to generally review my code.
- On the resource manager project, I created new REST endpoints and integrated new SpiceDB permission checks into the code. SpiceDB was the most challenging part, since time constraints meant learning it on the fly. I was also analyzing a possible conversion from connexion (OpenAPI) to FastAPI in the resource manager. This task relied heavily on analysis and prototypes using Claude's Sonnet 4.6 (then 5) AI model.
- As part of the team on the resource manager, I felt that a 35-40 minute CI duration in GitHub Actions was unacceptable so I used AI to diagnose the slowness and, iterating multiple times, I was able to bring the CI time down to 8 minutes. This was accomplished by reducing test timeouts, using GitHub Actions environment caching, and rewriting a few poorly executing tests. I monitored and reviewed changes but let the AI do most of the work. AI makes mistakes — you have to know what to look for and where to find it.

Software Engineer and Senior Software Engineer **2002 to 10/2024**
Intel Corp
As a Software Engineer and Senior Software Engineer, I was responsible for designing, developing, and maintaining software applications, ensuring high performance and scalability.

- Enhanced cloud software performance by optimizing open-source libraries like PostgreSQL and Velox/Gluten for Intel architecture within the AWS team, driving improved performance and scalability. Employed profiling tools, including flame graphs, and developed test harnesses to ensure consistent and reliable outcomes.
- Designed and scaled a Unified Control System for High-Performance Computing (HPC) environments, overseeing data collection, transformation, RAS event management, and storage for a large number of nodes. Used Java and Python to implement control and analysis features, delivering a responsive and dependable system for internal use.
- Played a pivotal role in developing low-power data transfer standards by contributing to the Ecma working group, establishing technical requirements, and aiding in the creation of a ratified specification. Led the architecture and security evaluation of Intel's implementation, ensuring the platform's feasibility and compliance with security standards.

Personal Growth

This section describes what I do on my personal time for growth. This section only covers the last 1-3 years. These can be shared for review in separate review meetings (they are not open-sourced).

SwiftData/SwiftUI Multiplatform Daily Food Planner with Nutrition Details. This application was designed to run on iOS and MacOS. This was written for my personal use and will not be published. It leverages CloudKit for data access across all supported platforms.

CoreData/SwiftUI Shopping Application. There are 2 variations of this application. The first, published in November 2024, uses CloudKit to share data between devices on a single account. The second and newer version leverages a Redis client library and a server running on a Tailscale (https://www.tailscale.com/) private network to allow multiple users regardless of Apple IDs. It mimics CloudKit in that changes on one device show up moments later on other devices. This will not be released to the public due to Redis hosting costs.

Built a Golang backend service with PostgreSQL to replace CloudKit, then ported the Daily Food Planner and Shopping Application to use the new backend data store. The design used local-first data for offline use, syncing once a connection was established. The entire backend and the application ports were accomplished with Claude’s Sonnet and Opus models.

Education

BS in Computer Science (with a Minor in Math) from the University of Idaho