

Paul Amonson

Current Location: Hillsboro, OR 97124

Target Location: Remote or Spokane, WA

Email: paulamonson@gmail.com

Goals

I am seeking a position as a senior software engineer in a productive team environment or to help establish a productive team. I am a dynamic employee who can effectively work on all phases of software development including implementation, design, and architecture. I am looking for an opportunity to influence current project directions, start a new project, or provide new perspectives in existing projects.

Skills

- **Overall Summary:** Over 30 successful years designing and implementing software on multiple platforms including Unix/Linux and Microsoft Windows (all versions). My experience includes middleware/library code, server-side components, actual Linux services, command line tools, and GUI applications on Windows and MAC/IOS.
- **Dedicated Team Member:** Motivated to help foster a cohesive, inclusive and efficient team.
- **Self Motivated:** I complete tasks with precision and quality without the need for constant direction. I also proactively start and complete unplanned sub-projects that benefit the team or project to guarantee the project's success. More recently for personal growth I learned Xcode, Swift, SwiftUI, and SwiftData to write a calorie planning app.
- **Multiple Roles:** I effectively take on multiple roles (architecture, design, and implementation) on one or more projects and complete them consistently.
- **Expert Languages:** Java 8-17, C++, C, C#, Python, Swift, Groovy, Bash, other Linux shell scripting (sed, awk, and more), Ansible, Windows batch and some PowerShell, and a few others.
- **Proficient Technologies:** *Design/Implementation:* SDL best practices, Linux, SQL, Redis, JSON/YAML, XML, HTML, STL, ZeroMQ sockets, Kafka, VoltDB, HTTP, network sockets on both Windows and Berkley. *Build Environments:* Jenkins, Makefile, cmake, gradle/maven Java projects, meson with ninja, rpmbuild, DEB packages, and others. *IDEs:* IntelliJ, PyCharm, Visual Studio, VS Code, Xcode (Mac and IOS), and others.
- **Software Process:** Extensive experience with multiple development methodologies (Waterfall, Rapid Prototyping, Agile, etc...), multiple source control/automation/QA systems, conducting security analysis on designs (threat analysis), code and design reviews (bugs and security), comprehending specifications, and effectively writing specification documents. I am a strong believer in test driven development.

Work Experience

Intel Corp - 2002-10/2024 - Software Engineer and Senior Software Engineer

- *Cloud Software Optimizations (2 years);* Modifying open source libraries and tools to run optimized for Intel architecture. I was in the AWS group and modified PostgreSQL, Velox/Gluten, and some other OSS libraries with more minor fixes. This position required a working knowledge of flame graphs, and other profiling tools as well as a ability to design test harnesses to measure performance in a predictable manner.
- *DAI-DS (3+ years);* Based on the previous HPC projects, this project focused solely on monitoring and determining system health. Incoming data was normalized, analyzed and archived by this more targeted solution. This was targeted at an external customer.
- *Summary of All Previous HPC Control System Work (4+ years);* Started by prototyping control system control and monitoring operation by using Python. Then iterated and learned as much as we could about how scaling to large systems would work. Finally, the Unified Control System was started in Java 8. This software was designed based on the learnings from the prototyping. This was a large effort including data collection/transformation, data analysis, RAS event processing, and storage. The next major functionality was around controlling compute nodes with a Python CLI/Java REST server to perform common DB queries use cases and a UI for control operations. Finally, after control and analysis features were completed, we implemented a reactive system of control based on incoming data. For example: when thermal targets were exceeded for specified period would force a shutdown of the offending node(s) and send an alert to the administrator(s). The goal was to scale the software to handle up to 100,000 nodes in a HPC machine, a very high bar. This was targeted at an internal customer.
- *Tools Engineer for New Parallel Co-processor (>72 cores);* Wrote software requirements specifications, high level design specification, produced common team code (some in anticipation of need yet defined) while writing unit tests in parallel with implementation (Test Driven Development).

- *Low Power Data Transfer Specification Standards Work and Implementation*; Started by participated in the ECMA working group for a year defining technical requirements for Network Proxying (<https://ecma-international.org/publications-and-standards/standards/ecma-393/>). My goal was to help produce a specification that included Intel's implementation goals for the ME firmware. I succeeded and we had a group ratified specification finalized. Next I helped architect Intel's implementation. I was responsible for the technical feasibility research and translating the results to requirements in a capability architecture requirements specification document for this platform feature. Once the architecture was at a 0.50 maturity, I was then responsible for doing the security analysis to satisfy the employer security life cycle requirements for a product.
- *Design & Implement Reference Implementation for DLNA v.1.0*; over 18 months added multiple features to a middleware cross platform SDK (C/C++) for the UPnP networking protocol and made sure that they were DLNA compliant. This was targeted both to full system implementations and small devices with limited memory/processing power.

Previous Employers - 1990-2001 - Software Engineer

Details upon request.

Personal Growth (Not on-the-job)

This section describes what I do on my personal time for growth. This section only contains the last 1-2 years only. These can be shared for review in separate review meetings (they are not yet 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 a 2 variations of this application. The first which I published in November 2024 and uses CloudKit to share 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 published to the general population due to Redis management costs.

Education

BS in Computer Science (with a Minor in Math) from the University of Idaho year 1990. Details can be divulged upon request.

Personal

Interests include reading technical journals/blogs, developing OSS software (<https://www.github.com/paul-amonson>), bowling, watching good sci-fi movies, long drives, playing board games, and cooking.