Andrew Hartman

256-513-9447

andrew.hartman@hartmansoftware.com

Keywords

Software Engineer	**Backend**	**Embedded**	**Models & Sims**

Technical Skills

Languages:	C++, Go, Python
Frameworks:	Kafka, RabbitMQ, FlatBuffers, Protobuf, gRPC, Qt, FastAPI, Postgres, PostGIS
Tools:	Docker, Git, Gitlab & Jenkins CI/CD, Jira, CMake, Valgrind, Perf, Perforce, SVN

Work Experience

Software Consultant **Hartman Software Solutions** Current

Responsibilities: Read multiple messages from different vendor hardware (**ARINC-429, MIL-STD-1553, etc**) and translate them to a common **Protobuf** format. The messages were then compressed with **ZStd** and made available to send to different sinks (**MQTT, Kafka, TCP, etc**) based on configuration.

- Architected and implemented a core library in modern **C++23** to provide common interfaces to collect messages from multiple hardware vendor APIs.
- Optimized main message collection hotpath, resulting in **82%** increased throughput.
- Implemented wrapper around **Opentelemetry** to compress and write logs, traces, and metrics to disk with **ZStd**, stream to an **OTLP** collector, or output in plaintext to the console.
- Implemented robust **CICD** pipeline on legacy **C++03** libraries, exposing and fixing **1000+** warnings, static-analysis findings, and sanitizer findings.
- Identified and fixed multiple data-race and threading bugs.
- Set up multiple dashboards and graphs to get insights from **OpenTelemetry** via **Signoz**
- **Mentored** multiple junior and mid-level developers.

Sr. Software Engineer **Rebellion Defense** Jul 2022 - Oct 2023

Responsibilities: Design and create the foundation for a cloud based modeling & simulation platform. I was tasked with architecting and leading the development of the framework that controlled the models, data flow, timing, and more.

- Architected and implemented an ABI stable **C++** lib reducing time to implement new language support by **90%** and supporting simulation upgrades without requiring clients to recompile their code.
- Architected and implemented language bindings with the above lib to enable **Go & Python** simulation models.
- Implemented **OpenTelemetry** instrumentation interface for logging, tracing, and metrics in the **C++, Go, & Python** codebases.
- Optimized **OpenTelemetry gRPC** exporters to be async resulting in **198%** increased throughput.
- Set up multiple dashboards and graphs to get insights from **OpenTelemetry** via **AWS OpenSearch**
- Architected and implemented multiple backend **REST** api layers using Python's **FastAPI** framework.
- Migrated prototype / legacy **Python** codebase to **Go** resulting in 102% performance increase.
- Migrated **Postgres** database / queries to use **PostGIS** resulting in a 61% faster response time.
- **Tasked and mentored** multiple junior and mid-level developers.

Software Lead **MTSI** Apr 2021 - Jul 2022

Responsibilities: Partner with NASA to design and create a framework that enabled AI & ML on safety critical devices. Our focus was enabling autonomy on drones to allow certification of non-human-in-the-loop systems. Essentially, by running the framework on a drone you would be cleared to fly without an operator and could have the autonomous system take full control completely outside of radar range.

- Architected and implemented a robust **C++** microservice framework utilizing **Kafka and Protobuf** to actively monitor and rectify erroneous UAV and Aircraft behavior.
- Selected on-vehicle hardware and software components to align with the defined architecture and product offerings.
- Architected and implemented diverse interfaces to hardware sensors and autopilots in **C++**, ensuring seamless integration and data exchange.
- Developed clear company-wide engineering strategies for observability, testing, and code quality.
- Migrated legacy **C** libraries into modern **C++20** libraries.
- Implemented custom **Yocto** linux image for use in embedded companion computer.
- Architected and implemented an in-flight status monitor for flight testing using **Qt**.

Software Lead **SAIC** Oct 2020 - Apr 2021

Responsibilities: Modify the MLRS / HIMARS launcher to be operable remotely for range safety while testing experimental munitions.

- Architected and implemented modifications of HIMARS / MLRS launcher to be operable remotely via **UDP, TCP, & Qt** over radios.
- Worked closely with the customer and systems team to ensure the **UX** was intuitive.
- Concentrated on reducing operator cognitive load regarding latency and launcher state.
- Optimized and translated old **C++03** code to modern **C++17** equivalent.
- **Tasked and mentored** team of 4 other software engineers.

Software Engineer **Torch Technologies** May 2018 - Oct 2020

Responsibilities: Extend a high-fidelity PATRIOT simulation to work in realtime with an HWIL ICC and create a flight test planning tool for hypersonic missiles with a focus on satellite sensor participation.

- Architected and implemented Python analysis scripts with **Pandas & numpy** to analyze simulation data.
- Translated IR sensor models and radar fusion algorithms from **Matlab** to **C++**.
- Architected and implemented a cross platform C++ **Qt** desktop application for data scientists to rapidly iterate on flight test planning for hypersonic missiles.
- Worked closely with the customer regarding **UX** to ensure the UI was intuitive.
- Set up **Jenkins CI/CD** jobs to automate code sanition, analysis, and regression tests.
- Debugged C++ middleware to allow communication between PATRIOT **HWIL** and a simulation via **DIS & UDP**.

Education

Bachelor of Science, Computer Science 2015 - 2018
University of Alabama Huntsville

Last updated: March 21, 2024