Charles Hampton

Alexandria, VA · 703-776-0200 · jesse@hamptontech.it.com

EDUCATION

Southern New Hampshire University — B.S. Computer Science (in progress) 2025 – 2028 (expected)
Junior standing; transferred from Bridgewater College.

Bridgewater College — Computer Science 2023 – 2025

EXPERIENCE

Convergent Data LLC — *Software Engineering Intern* Dec 2025 – Present · Alexandria, VA

- Engineered and maintain a 60-patch GitLab CI/CD pipeline that cross-compiles RTEMS for the NXP T4240 (PowerPC e6500), produces bootable ITB images, and archives build artifacts to a Nexus repository.
- Debugged RTEMS RTL (run-time loader) dynamic module system at the assembly level — resolved PowerPC64 REL24 relocations and TOC pointer setup enabling dynamically loadable `.o` modules over NFS.
- Diagnosed a DPAA2 Ethernet driver stall where ARP/ICMP frames entered the hardware QMan DQRR queue but never surfaced to the BSD network stack; traced root cause to a Frame Queue stuck in Active Held state.
- Designed hypervisor configurations (NXP HV) for running Linux and RTEMS guest partitions concurrently on the same T4240 hardware, supporting a dual-RTEMS shared-image pipeline path.
- Built a T4240 fleet management web application (Python/Flask + JavaScript) with AUTO_IP board detection, real-time serial WebSocket, multicast configuration delivery to 7 guest partitions, and an NFS-based app install system.
- Implemented a Mongoose HTTP server embedded into a running RTEMS image for in-band runtime diagnostics and application delivery.

Taurus Telesys — *Software & Compliance Engineer* 2024 – 2025 · Newport News, VA

- Developed a Digital Twin+ distributed energy simulation product in EPICS (Experimental Physics and Industrial Control System): containerized solar testbed with photovoltaic panels, inverters, and thermal tanks instrumented via Prometheus and visualized in Grafana dashboards; deployed on RHEL 8 via Podman with STIG compliance requirements.
- Built and maintained an air-gapped build environment (ProjectNexus): Sonatype Nexus 3 artifact repository, offline Maven builds in Docker, and a VxWorks 6.9 PowerPC cross-compilation container using Wine + proprietary Wind River GCC — all on a direct-ethernet-linked Intel Mac with no internet access.
- Set up TFTP/NFS boot infrastructure (RHEL 8.8, FIPS-enabled) serving T4240RDB SDK images as part of a 500-node production deployment at Newport News Shipbuilding; managed DTB modifications and FMC policy files for the Fedora Solution production TFTP server.
- Administered two VPN-segmented Proxmox environments (TaurusHQ + TGate); performed VM lifecycle operations including vzdump backups, qmrestore migrations, and FIPS-compliant SSH key provisioning.
- Supported NIST SP 800-171 and CMMC Level 2 compliance: mapped controls to evidence, authored SSP sections, and prepared artifacts for CMMC assessment including access control, audit logging, configuration management, and incident response.

Analytica — *CI/CD Intern* 2022 (5 months)

- Built and maintained CI/CD pipelines for development teams; streamlined build and release workflows.

PROJECTS

Federal Opportunity Intelligence Platform · *Python, Flask, PostgreSQL, SAM.gov API, LanceDB*

- Built a full-stack contract intelligence web app: SAM.gov API integration with keyword matching across 47 phrases and 7 capability groups, contract pipeline UI (Tracking → Pending → Confirmed) with bid/no-bid scorecard and win-rate dashboard.

- Implemented semantic resume-matching against federal opportunities via PDF/DOCX extraction; integrated USAspending.gov for historical award pricing; added a RAG semantic search layer over all contract posts using vector embeddings.
- Deployed with hourly cron + flock, launchd plist, JSON API with API-key auth, and nightly Google Drive backup via rclone.

CMMC Compliance SaaS Platform · *Python, React, PostgreSQL, Ollama, LanceDB, Proxmox*

- Built a multi-tenant CMMC/NIST compliance SaaS: JWT/RBAC auth, 1,456 seeded controls translated to plain-English IT requirements, Incident Response "Panic Button" with 72-hour DoD notification countdown, and role-based audit log.
- Deployed an AI layer on Proxmox LXC with NVIDIA GTX 1060 GPU passthrough: LanceDB vector database (1,538 chunks), Ollama with Qwen2.5 + Nomic embeddings (sub-second inference vs. 30s CPU), three AI endpoints with conversation history.
- Wrote `master-nvidia-deploy.sh` automating full LXC lifecycle: free-ID provisioning, active GPU conflict resolution via hardware scan, and NVIDIA library wiring inside the container.

Terminus — Terminal Code Editor · *Rust, ratatui, tree-sitter, ropey, crossterm*

- Built a terminal code editor in Rust: rope-based O(log n) text editing, incremental syntax highlighting via tree-sitter across 10+ languages, multi-cursor, split panes, fuzzy file finder (Ctrl+P), find/replace, and git status indicators.
- Wired live AI chat panel to Claude/Codex/Gemini CLI tools; built a background Gemini Oracle daemon with full codebase context.

Dead Drop — Multi-Agent Communication Platform · *Python, MCP (Model Context Protocol), HTTP*

- Designed and built an MCP server enabling real-time AI-to-AI message passing across Claude Code sessions: HTTP push transport, team-scoped inboxes, Drift Protocol v2 (task handshakes, health pings, file ownership), and compaction recovery for context loss.

Local RAG Desktop Application · *Python, PyQt6, HuggingFace Transformers, Ollama, SQLite-vec*

- Iterated from a CLI tool to a full TUI to a native PyQt6 desktop app for semantic search and Q&A over arbitrary codebases; HuggingFace embeddings stored in SQLite-vec, Ollama Q&A wired into a side panel with visualization.

Detroit — Multi-Agent Code Factory Pipeline · *Rust, GPUI, SQLite, TypeScript, Vite, React*

- Built a native desktop application in Rust using GPUI (the framework powering the Zed editor) with an assembly-line UI: six reorderable stations (INTAKE → RESEARCH → PLAN → BUILD → REVIEW → SHIP), each assigned a specific AI agent — Gemini for research, Claude for planning and QA review, Codex for code generation.
- Implemented real-time JSONL stream parsers for all three AI CLI tools (Claude, Codex, Gemini), rendering live token-by-token output in a live feed panel; wrote per-agent cost tracking against real published pricing (Claude: $15/$75 per M tokens; Codex: $2.50/$10; Gemini: $0.10/$0.40) with a running cost display per job.
- Persisted full job history to SQLite: sources indexed, implementation plan, review result, cost in cents, and all generated output files — browsable in an output browser panel with file-level navigation.
- Supported custom pipeline stages (insertable anywhere in the assembly line), parallel build lanes, per-job rules, model picker per station, and a source index panel showing research citations.
- Structured as a Cargo workspace (`detroit-core` library + `detroit-cli` binary) sharing pipeline logic with the GUI; companion web dashboard built in Vite + React + TypeScript with a Node.js/WebSocket backend mirroring all pipeline events.

AI Video Generation Lab · *Python, FastAPI, CogVideoX (diffusers), Ollama, ffmpeg*

- Built a local text-to-video pipeline: FastAPI job queue accepts a screenplay, Ollama (llama3) parses it into scenes with visual prompts, CogVideoX-2B (HuggingFace `diffusers`, Apple Silicon MPS backend) generates one video clip per scene, and ffmpeg assembles clips into a final MP4.
- Single-page web UI with progress tracking (scene N of M), live status polling, and in-browser video playback — entire pipeline runs locally with no cloud API dependency.

Raspberry Pi Face & Object Recognition · *Python, OpenCV, Raspberry Pi, camera module*

- 5-phase learning project: hardware setup → real-time face detection → face recognition → object detection → output/alerting; documented hardware procurement across three budget tiers.

SKILLS

Languages: Python, C, Rust, JavaScript / TypeScript, Bash / zsh, SQL, PowerPC assembly (inline)

Frameworks & Libraries: React, Vite, FastAPI, Flask, Tailwind CSS, Bootstrap, ratatui, GPUI, PyQt6, Recharts, React Flow, tree-sitter

AI / ML: Ollama (Qwen2.5, Nomic), LanceDB, HuggingFace Transformers, SQLite-vec, RAG pipeline design, MCP server development, Claude API, multi-agent orchestration

Embedded / RTOS: RTEMS (PowerPC64), VxWorks 7, NXP T4240 (DPAA2, QMan, FMAN), hypervisor configuration, RTL dynamic loading, ITB/FIT image format, U-Boot, NFS boot

DevOps / Infra: GitLab CI/CD, Docker / Compose, Proxmox VE (LXC + KVM), Kubernetes, NVIDIA GPU passthrough, TFTP / PXE, NFS, Nginx, Git

Security / Compliance: CMMC Level 2, NIST SP 800-171, DFARS, JWT / RBAC, multi-tenant SaaS auth, SSP authoring, control mapping & evidence collection

Data & APIs: PostgreSQL, SQLite, LanceDB (vector), SAM.gov API, USAspending.gov API, Google Drive (rclone), OAuth