

Druhin Bhowal

(425) 362-0237 | dbhowal@cs.washington.edu | [linkedin.com/in/druhinb](https://www.linkedin.com/in/druhinb) | github.com/druhinb | U.S. Citizen

EDUCATION

University of Washington

Seattle, WA

B.S. in Computer Science | GPA: 3.93/4.0

Expected Jun 2027

- **Coursework:** Adv. Operating Systems, Distributed Systems, Compilers, Concurrency & Parallelism, Systems Programming, Computer Security, Algorithms, Data Structures, Machine Learning, Deep Learning, NLP, Autonomous Robotics

EXPERIENCE

Roblox

San Mateo, CA

Software Engineer Intern

June 2026 – Present

- Shipped server logs and post-mortem server history to the creator dashboard; **60%** of top 500 creators adopted them within a month.
- Built **Flink** pipelines processing and ingesting a combined **2M** events/sec of client and server logs into sharded **ClickHouse**.
- Owned the read path, designing an **Elasticsearch** cluster to serve server details in **under 200ms** and logs in **under 500ms** at p99.
- Led container connectivity, compute scheduling, and queue trigger design for Roblox's from-scratch **Lambda**-style platform.
- Built Lambda worker in **Rust & C**; **gVisor** isolation, custom Luau runtime, **SPIFFE** mTLS, sub-**100ms** in-game invocations.
- **Halved** payload sizes and root-caused duplicate delivery in Cross-Server Messaging, Roblox's **C++** inter-server bus.
- Won signoffs and SVP approval for designs and capacity estimates across **8+** partner teams, from engine security to infrastructure.

Paul G. Allen School of Computer Science & Engineering

Seattle, WA

Undergraduate Teaching Assistant

Jan 2026 – June 2026

- Taught **Operating Systems** to **150+** students, debugging kernels, hosting recitation for **40+** students, and managing logistics.

DropzoneAI

Seattle, WA

Research & Development Intern

Sept 2025 – Dec 2025

- Designed the agent's memory on **Postgres** and **pgvector**, carrying context across alerts to cut median investigation time **35%**.
- Shipped **Azure**, **Jira**, and **CrowdStrike** integrations, lifting alerts resolved without analyst input from **70%** to **90%**.
- Built adapters normalizing alerts, tickets, and EDR telemetry into a unified evidence model the agent reasons over.

Amazon Web Services

Seattle, WA

Software Development Engineer Intern

June 2025 – Sept 2025

- Built new shard-failover pipeline for **Kinesis Data Streams (Tier-0, 99.99% SLA)**, retiring a recurring class of long-tail outages.
- Modeled hardware telemetry in a **Java/Rust** service to proactively migrate at-risk shards, cutting heat-related disruptions **24%**.
- Designed real-time host-failure detection over fleet telemetry, triggering shard failover before customer latency degraded.
- Drove the rollout from canaried telemetry streams to every **us-west-2** host across heat, storage, and control-plane owners.

DevMatch

Seattle, WA

Software Engineer Intern

Oct 2024 – June 2025

- Built a browser IDE on **AKS** with Docker-in-Docker sidecars and per-tenant isolation, sandboxing **20,000+** untrusted code runs.
- Stood up an **OpenTelemetry**, **Prometheus**, and **Grafana** observability stack with alerting across the platform.

PROJECTS

dbinfer: LLM Inference Engine | github.com/druhinb/dbinfer | C++23, Metal, NEON

- Built an Apple Silicon LLM inference engine from scratch in **C++23**, writing bespoke CPU (**NEON**) and GPU (**Metal**) kernels.
- Achieved **147 tok/s** (Q8_0, M3 Max) GPU decode speed via successive optimization passes, **60%** of llama.cpp's Metal backend.
- Designed a custom compression format for model tensors, saving **15%** space vs. GGUF with a **10%** decode-time increase.

drubuntu: ARMv8 Multikernel OS | github.com/druhinb/drubuntu | C, ARMv8

- Built the memory system from bare metal: a buddy allocator managing RAM in **4KB** pages and a four-level page-table walker.
- Implemented same-core and cross-core message passing between processes, including **UMP**, a lock-free shared-memory channel.
- Wrote the **ELF** program loader, per-process address spaces, and multicore bring-up to run user processes across all **4** cores.

TECHNICAL SKILLS

Languages: C++, C, Rust, C#, Java, Python, TypeScript/JavaScript, SQL

Technologies: Flink, ClickHouse, Kafka, Kubernetes, Docker, gVisor, KVM, Postgres, AWS, Azure, Spark, Temporal, Metal, PyTorch, OpenTelemetry, Prometheus, Grafana, gRPC, React, Linux

Concepts: Distributed Systems, Operating Systems, Kernel Development, Performance Optimization, Concurrency, Compilers, Observability, Databases, ML Inference, Serverless, Workflows