Alfred Dwan

New York, NY • +1 5513768692 • alfred.dwb@gmail.com

EDUCATION

New York University GPA: 4.0/4.0 Expected May 2026
Master of Computer Science & Engineering *New York, NY*

City University of Hong Kong 2020 - 2024
Bachelor of Computer Science *Hong Kong*

EXPERIENCE

PredictX (Startup) May 2025 – Sep 2025
Software Engineer Intern - AI track *Hong Kong*

- **In collaboration with the DeepSeek AI team, shipped a user-facing intelligent Q&A feature that increased the self-service success rate by 25%** by providing instant, accurate answers to complex user questions.
- Engineered a **user-facing** AI backend service using **FastAPI**, **Docker**, and **Kubernetes** (**HPA**, readiness/liveness probes), successfully maintaining a p95 latency of <500ms under load, supporting 10K+ monthly requests via async I/O and backpressure controls.
- **Architected the underlying agentic RAG workflow** for this feature using **LangChain** and **FAISS (HNSW/IVF)**, which automated query processing and cut internal support's task resolution time by 40%.
- Optimized LLM inference costs by 35% with no performance regression by implementing **LoRA fine-tuning**, **quantization** (**BitsAndBytes**), and packaging models with **TensorRT** for efficient serving.
- Established the end-to-end MLOps pipeline that accelerated the delivery of new **AI features**, reducing the model deployment cycle from days to hours using **GitHub Actions** and integrating a monitoring stack with **Prometheus**, **Grafana**, and **OpenTelemetry** for proactive alerting.

ByteDance Jun 2024 – Aug 2024
Software Engineer Co-op Program (Industry Collaboration) *Beijing*

- **Improved the listening experience for users on low-quality networks** by reducing audio streaming latency by 22%, using an **agent-driven** framework in multi-threaded **C++** and **Swift** (**AVFoundation**) that optimized buffer pre-warming and adaptive chunking.
- Designed a real-time, multi-agent orchestration layer using **Kafka**, **Redis**, and **WebSockets** to handle concurrent live-stream events, improving system stability by 25% during peak traffic.
- Shipped an AI-powered content discovery feature that increased Daily Active Users (**DAU**) by 12%, using **PyTorch** and **FAISS** to reduce user navigation time by 20%.
- Cut model inference latency and memory usage by over 30% by deploying **HuggingFace** agents with **ONNX Runtime** on **Kubernetes**, managed via an **Istio** service mesh for intelligent traffic shifting and fault tolerance.
- Architected an automated canary and rollback system for the CD pipeline using **Prometheus** time-series telemetry, cutting incident recovery time from hours to minutes and safeguarding production SLOs.

Greenhouse Data (Startup) Feb 2023 – May 2023
Software Engineer Intern *Hong Kong*

- Built a **distributed** web crawler in **Node.js** microservices using **JavaScript** and **TypeScript**; achieved **10K+ pages/min** with **Express.js**, **PM2**, **RabbitMQ** (<5 ms); deployed on **AWS EC2** via **Docker Swarm**, **Jenkins CI/CD**, and **Prometheus** monitoring with **Unix/Linux** environments
- Designed **MongoDB** + **Mongoose** for **1TB+ data** with **SQL** query optimization; validation/indexes ensured 99.99% consistency and **5 ms queries**; added full-text search via **Elasticsearch** and implemented **PostgreSQL** for relational data management.
- Boosted throughput with **Redis Cluster** caching and **Lua** atomic ops, reaching **10K+ req/sec** and cutting API latency by **20%**; secured access with **OAuth 2.0**/session auth and HTTP/HTTPS protocols through security software development practices.
- Shipped a real-time data monitoring dashboard for clients using **React**, allowing users to track job status and performance from the web application.

Siemens Sep 2022 - Jan 2023
Software Engineer Intern *Hong Kong*

- **Built a full-stack monitoring dashboard** for station control systems using **React**, **FastAPI**, and **PostgreSQL**, enabling real-time visualization of sensor and telemetry data across 20+ IoT nodes.
- **Shipped public-facing RESTful APIs** that enabled self-service data access for field engineers, standardizing communication between IoT devices and the cloud.
- **Engineered the CI/CD pipeline** using **Jenkins**, **Docker**, and **Google Test**, which accelerated the feature delivery lifecycle by 40% for critical mobility projects.
- **Led the modernization of legacy C++ modules** by migrating them to containerized microservices, improving system scalability and reducing maintenance downtime by 5%.

Dolby Mar 2022 - Aug 2022
Software Engineer Intern *Beijing*

- Built responsive **React** + **TypeScript** applications with modern state management (**React Query/Redux Toolkit**), route-level code splitting, and performance budgets (LCP/CLS/TTI); configured **Vite/Webpack** optimizations and HTTP caching.
- Accelerated UI development by 30% by implementing a Storybook-driven design system and **WCAG** compliance (keyboard paths, focus management, contrast), reducing UI inconsistency and review churn.
- Improved reliability with resilient API clients (retry with jitter, timeouts, centralized error boundaries) and added client-side observability (**web-vitals** plus custom events, correlation IDs); integrated **Jest** unit tests and **Cypress/Playwright** E2E into **Jenkins** pipelines with artifact retention.

PROJECTS — Cloud-Native GoPaaS Platform Development

- Built a cloud-native GoPaaS on Go-micro v3 and Kubernetes with service mesh using hardware optimization techniques; standardized config, discovery, circuit breaking, and rate limiting; operated scalable clusters with Docker containers, NGINX Ingress, and custom controllers; achieved <50 ms inter-service latency and 99.95% uptime.
- Integrated Istio (mTLS, traffic shifting, retries) and Helm deploy/rollback workflows using market data protocols for financial markets applications; release cycles accelerated by 3×.
- Standardized microservices with gRPC/protobuf using binary protocols; added an API Gateway with Hystrix, load balancing, and rate limiting; Prometheus/Grafana observability cut peak-time failures by 65% and sustained +60% traffic growth.
- Automated Helm-based canaries and rollbacks, tripling deployment throughput while reducing change failure rate and aligning practices for multi-region release hygiene