AARON CHEN

(216) 801-3618 | aaronbychen@gmail.com | linkedin.com/in/aaronbychen | github.com/aaronbychen | aaronbychen.com

EDUCATION

Washington University in St. Louis **Aug 2023 - May 2027**
B.S., Computer Science and Mathematics (GPA: 3.95/4.00) *St. Louis, MO*
- **Achievements:** AIME Qualifier (5% International), Llama Stack Finalist (Meta/8VC), ARML National Bronze, ACSL State 1st

EXPERIENCE

Portal **May 2025 - Aug 2025**
Software Engineering Intern *Houston, TX*
- Scaled **distributed systems** (Node/Express) with **RESTful APIs** behind **AWS API Gateway** and a **load balancer** to a horizontally scalable, **fault-tolerant** multi-instance setup; supported **5.1k concurrent** sessions under **high concurrency**
- Built **Kafka**-backed async processing (**queue** + **workers**) for resume/LLM jobs; triggered **AWS Lambda** workers for event-driven tasks (bounded retries/backoff, **idempotency** keys), stabilizing bursts at **200-400** jobs/min with p95 queue lag <**3s**
- Implemented **Elasticsearch** search indexing for resumes/content; tuned mappings/analyzers and query patterns to reduce p95 search-latency ~**45%** (320ms to 175ms) and improve retrieval relevance for downstream search/QA
- Added **Redis** caching for hot reads (RBAC/auth, org config, 3rd-party lookups) with **TTL** + invalidation; achieved ~**80-90**% **cache hit rate**, reducing **40-60**% DB load and improving tail latency on key endpoints
- Optimized **Postgres**/Prisma access (JSONB + GIN indexes, composite uniques, hot-path secondary indexes); stored large resume artifacts/exports in **Amazon S3** (pre-signed URLs / lifecycle policies) to offload blobs and simplify access patterns
- Improved cross-service reliability (**RPC**-style internal calls): strict timeouts, fail-fast fallbacks, and observability (structured logs/metrics/alerts) with unit tests coverage to **90%**; shipped in an **Agile** workflow with **GitHub Actions CI/CD**
- Deployed with **Docker** and **Kubernetes** on **AWS** across development, staging, and production, enabling safe rollouts

CN Business Herald **May 2024 - Aug 2024**
AI Engineering Intern *Beijing, CN*
- Owned the end-to-end **RAG** platform: document ingestion/cleaning, chunking, **Azure Cognitive Search** indexing & retrieval, and **LangChain** orchestration (prompt templates, citation formatting, guardrails) for offline news analysis
- Built a self-serve **on-prem** admin dashboard (SSO, streaming, **MongoDB**-backed persistence) for retrieval tuning and output evaluation; cut combined platform run costs + ops/support time by **~27%** (normalized per 1k requests, rolling averages)
- Integrated **RouteLLM** routing to local models behind an **OpenAI-compatible API gateway**; reduced cloud API-served inference by **~38%** (gateway logs + token ratios + billing) while keeping client integrations unchanged
- Productized two editorial workflows (policy impact briefs; company/topic dossiers with timelines + Q&A banks), shrinking research from **0.5-2 days** to ~**10-20 mins** and standardizing citation-backed outputs for **200+ users**

PROJECTS

NL Video Scanner (Llama Stack Finalist) | ***Llama-Vision, LlamaIndex, NoSQL, OpenCV, FastAPI, Docker***
- Designed a fully offline retrieval layer with **LlamaIndex** + a local **NoSQL** KV/document store (caption metadata, frame/time indexes, tunable thresholds) to run fast **top-k** queries end-to-end on consumer **Apple Silicon** and **RTX-class GPUs** under on-device constraints
- Delivered an on-device, privacy-preserving **natural-language** video scanner at **1 FPS** using **Llama-Vision** + **OpenCV**; achieved **85%+ Hit@5** on 100-300 clips with **zero data egress**
- Enhanced robustness with OpenCV preprocessing (frame sampling, denoise, **CLAHE**, color normalization) for low-light/high-contrast scenes, improving retrieval quality by **40%+** and reducing false positives via tunable thresholds
- Served inference via **FastAPI** (async streaming + batching); improved concurrency and cut end-to-end latency by **~50%** (load-tested)

Distributed RPC Framework | ***Distributed Systems, HPC, Spring Boot, Etcd, Vert.x, Linux Cluster***
- Designed and implemented a low-latency **RPC** core in **Spring Boot + Vert.x** (async I/O, connection pooling, backpressure); reached **<5ms median latency** at **5k+ RPS** in steady-state concurrency tests
- Implemented **etcd-based service discovery** with fault-tolerant registration/failover; validated resilience via quorum/failure tests
- Developed a custom **Vert.x** async load generator to control concurrency and capture **p50**/**p95** latency distributions
- Deployed **benchmarked** on a small multi-node **Linux** cluster and ran end-to-end experiments (discovery, failover recovery, throughput) under realistic LAN conditions to validate stability and performance

SKILLS

- **Languages:** Python, TypeScript, JavaScript, SQL, Java, C/C++, Swift
- **Technologies:** AWS, Azure, React Native, Node.js, Express.js, FastAPI, Spring Boot, React, PostgreSQL, MySQL, Prisma-, MongoDB, NoSQL, Redis, Kafka, Elasticsearch, Docker, Kubernetes, GitHub Actions, Socket.IO, Etcd, Spark, Hadoop, Linux, Git
- **AI/ML:** LangChain, LlamaIndex, RAG, OpenAI APIs, RouteLLM, PyTorch, OpenCV, scikit-learn, NumPy, Pandas