HAO WANG

Herndon, VA | 215-266-4475 | hwang_pa@msn.com | LinkedIn

Summary

Staff Engineer with extensive experience building distributed systems at scale. Spent the last 4 years at Walmart building real-time data pipelines, event-driven architectures, and internal platforms that handle internet-scale traffic. My work has focused on the infrastructure side of things: low-latency APIs, Kafka streaming, feature aggregation systems, and developer platforms. Now looking to apply this experience to ML data infrastructure.

Strong in: Go, Python, Java, Kafka, distributed systems design, operational reliability

---

Skills

Languages: Go, Python, Java, C#, JavaScript/TypeScript, SQL
Data Systems: Kafka, Cassandra, Elasticsearch, SQL Server, MongoDB, Memcached
Infrastructure: Docker, Kubernetes, Azure, AWS, Microservices
Monitoring: Splunk, Grafana, ELK stack, distributed tracing

Experience

Staff Technical Expert

Walmart Global Tech | Herndon, VA | March 2020 – Present

Built several large-scale systems for Walmart's e-commerce security team. Most of my work involves real-time data processing, event-driven architectures, and making systems reliable enough to run 24/7 without intervention.

Real-time data systems I've built:

Counting System — Distributed feature aggregation for anomaly detection

- Processes millions of events/second with sub-second latency (p99 around 100ms)
- Stack: Golang, Memcache, SQL Server, Kafka-like messaging
- Horizontally scalable architecture - can add nodes to handle more load
- Built comprehensive monitoring: real-time dashboards, automated alerts, SLA tracking

- This is basically a feature store, though I didn't call it that at the time

Event Queue System — Multi-signal scoring for high-traffic events

- Handles Black Friday traffic (10x normal) without issues
- Stack: Golang, Kafka, Memcache, external APIs (MaxMind, Spur)
- Architecture: Kafka events → feature extraction → scoring → action
- Features include: geolocation, proxy/VPN detection, behavioral patterns, account history
- Built backfill capability for reprocessing historical data

IntelDb — Real-time threat intelligence system

- Streaming pipeline: Akamai SIEM → processing → scoring → REST API
- Stack: Golang, SQL Server, REST APIs
- Has a feedback loop: scores drive actions, outcomes feed back to improve scoring
- Added data lineage tracking so we can debug how scores are computed

Account Risk API — Low-latency service for account security

- Node.js API serving real-time risk scores in under 50ms (p99)
- Event-driven: Kafka backbone for async processing
- 99.99% uptime SLA - maybe 4 minutes downtime per month
- Full observability: distributed tracing, latency breakdowns, error tracking

Platform tools I've built:

Event Automation — Deployment orchestration system

- Think of it like a simple MLOps platform for rule deployment
- Stack: Golang, GraphQL, React.js, Memcache, REST APIs
- Handles multi-region rollouts across 4 markets
- Features: approval workflows, staged deployments, instant rollback, scheduled runs
- Built a "mission control" dashboard for monitoring deployments

Team Portal & Workflow Manager — Internal developer platform

- React.js frontend, Golang/Python backend

- Abstracts away infrastructure complexity - engineers write logic, platform handles the rest
- Centralized secrets, job scheduling, unified data access
- Cut time-to-production by about 60%

ELSA — Distributed search platform

- Elasticsearch cluster for massive log datasets
- Stack: Golang, Elasticsearch, SQL Server, React.js
- Similar scaling challenges to vector databases

Other stuff:

- Leading our Kubernetes migration (WCNP) - containerizing services
- Heavy Cursor AI user (top 2% globally) - been pushing AI-assisted development on the team
- Mentor 8 engineers, run design reviews, set technical direction
- On-call rotation - debug production issues, write post-mortems

---

Principal Software Engineer / Engineering Lead

Law School Admission Council (LSAC) | Newtown, PA | November 1996 – March 2020

Built systems ranging from ML applications to distributed test platforms over two decades at LSAC.

ML/AI systems:

Recommendation Engine — Personalized law school matching

- Python, Pandas, collaborative filtering
- Built microservices for serving recommendations

OCR Pipeline — Document processing system

- Python, Tesseract, Lucene
- Computer vision for transcript and bubble sheet scanning
- This was production ML before everyone was doing ML

Other systems:

Digital LSAT Platform — Got a patent for this one (US 10,078,968)

- Distributed test delivery system coordinating thousands of simultaneous users
- iOS, Android, Windows clients with real-time synchronization
- Similar challenges to distributed inference

SSO Platform — Identity system with Azure AD B2C

- Custom auth framework with microservices
- Risk-based authentication using behavioral signals

Data Sync Services — Real-time pipelines between CRM systems

- Dynamics 365, Oracle, SQL Server, Salesforce
- CDC (change data capture) patterns

IT Consultant

Fortune 500 Companies | Early Career

Enterprise data engineering and application development across multiple industries including pharmaceuticals, financial services, healthcare, and telecommunications.

Education

Master of Science, Computer Science
Saint Joseph's University — Philadelphia, PA

Patent

System and Method for Electronic Test Delivery
US Patent 10,078,968 | View on Google Patents

Why This Role?

I've been building infrastructure for systems that look a lot like ML pipelines - just not actual ML models yet.

Real-time feature computation, low-latency serving, Kafka streaming, operational reliability. The patterns are the same whether you're scoring fraud or serving model predictions.

What I bring:

- Extensive production systems experience with proven track record at scale
- 4 years building real-time data infrastructure at Walmart scale
- Strong Kafka background (Netflix uses this heavily)
- Know how to build reliable systems (99.99% uptime)
- Fast learner - top 2% Cursor AI user, always picking up new tech

What I need to learn:

- Spark/Flink (I've built equivalent systems, just not with these tools)
- Netflix's internal frameworks
- Working more directly with ML engineers and their specific needs

I'm looking for fully remote Staff/Principal roles. Compensation target around $300K-$500K total comp.