David Oyinbo

Senior Software Engineer | Backend, Distributed Systems & Real-Time Infrastructure

Mississauga, ON +1 437-217-6061
oyinbodavidbayode@gmail.com | linkedin.com/in/david-bayode-oyinbo | github.com/dev-davexoyinbo
davidoyinbo.com | davidoyinbo.com/engineering-logs

Hiring Manager

Re: Senior Software Engineering Opportunities

Dear Hiring Manager,

I am a senior software engineer based in Mississauga, Ontario, applying for backend, platform, distributed-systems, and real-time engineering roles. My recent work has involved taking systems from the original problem through architecture, implementation, deployment, observability, and documentation.

At Celeris Realtime, I am the sole architect, engineer, and operator of a public-beta, multi-region WebSocket platform. I built the Rust services, custom protocol, Kafka and Redis coordination, ClickHouse data layer, Nuxt/Vue dashboard, and AWS infrastructure. In recorded Criterion tests, the parser stayed near 0.3 microseconds from 128 B to 256 KiB. At 256 KiB, it took 305 nanoseconds versus about 38 microseconds for serde_json, roughly 125 times faster in that test. A single c8g.large (2 vCPUs, 4 GiB memory) node also sustained 96,114 messages per second for 25 minutes with 0.5 KiB payloads, 102 MiB memory, and 56% CPU. The load generator failed first, so this remains a tested operating point.

At 55 Tech, I owned a multitenant accounting service from database design through production delivery. It became the source of truth for balances, transactions, exposure, and settlement history, backed by PostgreSQL transactions, tenant isolation, and 794 automated tests across 35 suites. I also rebuilt a Python/FastAPI automatic bet-placement service, reducing p95 latency from 3,555 ms to 400 ms, removing an 8.7% failure rate on the comparison workload, and increasing throughput from 9 to 45 requests per second per node.

At HeySummit, I scaled realtime delivery for thousands of concurrent users by introducing Firestore as the primary datastore and routing selected data to Firebase Realtime Database instances according to load and participant count. I also built an account-linked RAG assistant across WhatsApp, Telegram, and Slack using LangChain, ChromaDB, PostgreSQL, custom APIs, and OpenAI and Anthropic models. In a blinded 30-question evaluation, retrieval reached 0.843 judged relevance at about 13 times lower cost per question than the direct GPT-5.5 comparison.

I would welcome a conversation about where this experience could help your engineering team. Thank you for your consideration.

Sincerely,

David Oyinbo