John Guerra

Backend Software Engineer | Java & Python | Low-Latency Systems & Production Debugging

Laredo, TX (Remote) | gekkoquestdev@gmail.com | linkedin.com/in/john-guerra20 | github.com/gekkoquest | gekko.quest

PROFESSIONAL SUMMARY

Backend software engineer with 6+ years building and operating performance-sensitive systems in Java and Python across high-traffic multiplayer infrastructure, real-time abuse detection, legacy platforms, and real-time AI. Builds Netty and Spring Boot services for thousands of concurrent connections and debugs latency, memory, and GC failures under live production load.

PROFESSIONAL EXPERIENCE

Backend Software Engineer, GekkoQuest — Contract Backend Engineering for High-Traffic Real-Time Networks

Remote | Aug 2023 – Present | Clients: Mineplex, MunchyMC (BadBoyHalo), Bionic

- Hardened Mineplex's server-side abuse detection (anti-cheat) for a network averaging 7,000+ concurrent connections by extending packet inspection and movement validation, shipping changes through development, staging, and production.
- Built and maintained MunchyMC's (BadBoyHalo) server-side abuse detection (anti-cheat) end to end, owning detection logic, threshold tuning, and false-positive triage on a live network.
- Co-architected Bionic's Java game server from scratch and scaled it through a launch of 1,000+ concurrent connections.
- Diagnosed and resolved memory leaks and hot-path bottlenecks within 24 hours of going live, then held the 50 ms game-loop deadline at 99.9% uptime through JVM heap and GC tuning, Prometheus instrumentation, and live incident response.

Backend Developer (Volunteer), Dragonball Online Global

Remote | Feb 2020 – Sep 2025 | One of three developers on a live MMO platform

- Eliminated a two-year-old client freeze affecting the entire player base by tracing it through ProcMon to synchronous per-write flush logging compounded by antivirus scanning, then rewriting the logging path — a failure repeatedly misdiagnosed as a memory leak.
- Cancelled impossible player actions in flight with a Python real-time abuse-detection service that cross-validated client heartbeat cadence against movement and action rates, catching speed and automation exploits the client could not be trusted to self-report.
- Held a weekly release cadence to the live player base while reverse-engineering and incrementally modernizing legacy synchronous C++ subsystems, owning client stability, performance, and CI/CD across a three-person team.

SELECTED PROJECTS

Ring Logger — Lock-Free Java Logging Library — Java 24, JMH

- Cut steady-state hot-path allocation to zero via a 16K ring of pre-allocated direct ByteBuffers, with VarHandle release/acquire publishing and lap-tagged flags so the consumer never reads a half-written slot on wrap.
- Reached ~3x Log4j2 (async Disruptor) throughput on short messages under JMH using a 1 ms amortized clock thread (~25x cheaper than per-call System.nanoTime), cache-line-padded counters, and batched drain with spin/yield/park backoff.

Stella / Novi — Real-Time AI Streaming Platform — Python, asyncio, ONNX, WebSockets, SQLite, React, Tauri

- Built and operate a ~25,000-line solo real-time AI platform on a single-process asyncio runtime where every input module hands typed events to a single writer, so a failed dependency degrades to a status flag instead of stalling the voice loop.
- Merges microphone, Twitch EventSub, Discord E2EE voice, and vision into one ordered timeline, isolating VAD state per speaker and rate-limiting chat with token buckets and per-sender cooldowns so bursts cannot evict the live context window.
- Hardened the prompt path against untrusted input: speaker names sanitized against delimiter injection, identity keyed to immutable numeric IDs rather than renameable logins, inbound events PII-screened ahead of a two-level moderation filter.
- Holds a ~400 ms end-of-speech-to-first-audio voice loop by running ASR on CPU via ONNX, streaming the first sentence into TTS through a hand-written incremental parser before the model's JSON closes, and removing the extra round-trip from the live path.
- Drives a 120 Hz animation and lip-sync compositor chosen from measured frame-gap data after 60 Hz held ~15% of frames, with forced-alignment word timing sampled off the audio playback cursor.

TECHNICAL SKILLS

Programming Languages: Java, Python, C++, TypeScript, SQL

Backend & Performance: Spring Boot, FastAPI, Netty, REST APIs, WebSockets, asyncio, multithreading, JVM and GC tuning

Data & Infrastructure: PostgreSQL, MongoDB, SQLite, Redis, Docker, Linux, Git, CI/CD, Prometheus

Tooling: AI-assisted development (Claude Code, Codex)

Spoken: English (native/bilingual), Spanish (professional working)