

Michael Welles

Email: mlwelles@gmail.com | Phone: 347-450-6518 | Location: Brooklyn, NY

Michael L. Welles

Address: 38 Covert St, Brooklyn NY 11207

Phone: 917-586-9218

Email: mlwelles@gmail.com

I'm writing about the Compute Engineering Lead role because the performance challenge you're describing is exactly the kind of problem I get excited about—where you need things to go fast without breaking what makes the product special in the first place.

At Raytheon, I led development of real-time telemetry pipelines for commercial jet engines. We were processing thousands of sensor readings per flight through fault detection models, and the whole thing needed to happen fast enough to trigger alerts that could range from "check this on the next scheduled maintenance" to "ground this plane immediately." The tricky part wasn't just making it fast—it was making it fast while maintaining complete audit trails showing exactly which model version and code revision produced each decision. We couldn't sacrifice the regulatory compliance that made the system trustworthy.

The way we solved it maps directly to what you're describing at Hex. We built abstraction layers that let us evaluate multiple models in parallel without adding latency, using asynchronous patterns and careful concurrency control. We established data-driven metrics for what "fast enough" meant at each stage of the pipeline, and we made incremental improvements we could measure and validate. Every optimization had to maintain the flexibility and auditability that regulators and airline operators needed.

I've been on both sides of the performance equation. At MediData, I inherited mobile teams where velocity was the problem—we increased sprint throughput by 2.5× through better CI/CD and disciplined practices. At Riverdrop, I architected a product search engine where millisecond-level latency was the differentiator. We built an ML-driven pipeline with event-driven processing on AWS (SQS/SNS, DynamoDB, Elasticsearch) where every stage needed to be independently optimizable. The microservices architecture meant teams could tackle bottlenecks in parallel—the image recognition team could optimize their pipeline without blocking the NLP entity extraction work.

What caught my attention about Hex is the emphasis on making performance a selling point, not just a technical checkbox. At Dayforward, we built an algorithmic underwriting platform that had to clear regulatory approval before launch. We had ten months, and the architecture decisions we made early determined whether we'd ship on time. We did—the platform launched the day we got approval—because we designed with clear guarantees at each layer and smooth rollout strategies baked in from the start.

I'm the kind of engineering leader who wants to understand where every millisecond is going. I'll pair with engineers to profile bottlenecks, establish the north star metrics that tell us if we're winning, and make sure everyone understands why this work matters to customers and the business. I've managed remote teams at Raytheon (14 developers across three projects) and at Istari Digital (distributed team building a zero-trust collaboration platform), and I've learned that the key to performance work is creating infectious excitement about making things go VROOM while maintaining the rigorous measurement and incremental progress that actually gets you there.

The compute architecture challenge you're facing—maintaining Hex's flexibility while delivering optimal performance across all data scales—sounds like exactly the kind of problem where you need someone who can think strategically about abstractions, communicate effectively across teams to multiply impact, and get deep into the technical details when needed. That's the work I want to be doing.

I'd love to talk about how we could make Hex's performance a competitive advantage.

Best,
Michael Welles