

Application for AI Systems Engineer – Agentic AI & Automation

Dear Hiring Manager,

Who I Am

I am an AI Systems Engineer specialising in agentic workflows, LLM-powered automation, and backend systems built for production. I take on AI automation problems end-to-end and ship them. My background spans the full AI stack: prompt engineering, RAG pipelines, vector databases, multi-agent orchestration, event-driven architectures, and backend API design across Python, Node.js, and TypeScript.

What I want to be clear about is how I approach problems. I treat every engineering challenge as a proper case study. Before writing a single line of code, I break the problem down, analyse it, compare approaches, and map out the full solution architecture including complexity trade-offs and system boundaries. I rely on visual thinking, Mermaid diagrams, flowcharts, and structured decision trees, to ensure the design is solid before execution begins. I also leverage multiple AI tools natively throughout my process, not to shortcut thinking, but to move faster once the thinking is done.

Previous Work

At Wellness at Work, I was the primary architect for a native macOS application integrating real-time ML models into live user workflows. I led the architectural design and drove the Mac app to launch, while also contributing to the Windows version of the product. My responsibility covered the full stack: system design, ML layer integration, UI/UX, and Firebase-based backend services. This was not a feature contribution. I owned the architecture.

At QClairvoyance Quantum Labs, I engineered automated data pipelines that structured a compound database of over 600,000 entries, and built a full-stack web application on React, Node.js, and PostgreSQL that made that data accessible and actionable across teams. The work here demanded precision in data engineering and reliability in system design at scale.

Current Work

I am currently building two production-grade AI systems in parallel, both of which I have designed and am taking to production independently.

The first is an **AI Agent Execution Platform** built on Node.js, TypeScript, BullMQ, and Redis. It features a decorator-based modular agent architecture for plug-and-play agent development, async queue-based execution with retry policies and structured lifecycle logging, multi-agent workflow orchestration with conditional routing and event-driven triggers, and REST APIs with role-based access control and ACL enforcement. Real-time operational control is handled through a Telegram integration with command routing, approval flows, and live execution monitoring.

The second is an **Intelligent Enquiry Pipeline with Semantic Search**, built for a domain-specific production use case. It uses PGVector and Llama 3 via Groq to power natural language search, for example resolving queries like “industrial warehouse with high ceilings in East London” against a structured database of listings. An AI agent orchestrator runs a sequential pipeline that extracts briefs, checks availability, scores leads, and drafts personalised email responses. The background processing layer is built on BullMQ and Redis with Bull Board for real-time job monitoring, Slack notifications, and Gmail integration. The stack is Next.js, Fastify, Supabase, and Docker Compose, with full test coverage and documentation across every layer.

These two systems represent the kind of work I do: retrieval, agent logic, API design, queue infrastructure, and deployment working together cleanly in production.

Why I Am a Strong Fit

AI Systems Engineering at its core requires three things: the ability to design systems that are architecturally sound, the ability to build them end-to-end without hand-holding, and the ability to reason about where AI fits in the pipeline versus where deterministic logic should take over. I work at that intersection every day.

I bring direct experience with LLM APIs from OpenAI, Anthropic, and Gemini; vector databases including FAISS, Qdrant, Pinecone, and PGVector; agentic frameworks like CrewAI and LlamaIndex; and workflow automation via n8n. On the backend side, I am equally comfortable with FastAPI, Node.js, PostgreSQL, MongoDB, and AWS infrastructure. I pick up existing codebases cleanly, understand design decisions quickly, and extend them without introducing chaos.

I understand the business problem before I open an editor. That instinct, combined with strong execution, is what separates an engineer who ships from one who stays in planning. I am looking for a team where that matters.