Raj Kumar Nelluri

(201) 554-8009 | rajkumarn2002@gmail.com | LinkedIn | GitHub | Portfolio

Summary

AI engineer who builds LLM agent systems and then measures whether they actually work: deterministic checks around the model, eval harnesses with held-out test sets and mutation testing, and rules engines that keep the LLM out of the decisions that matter. Evaluated an AI triage workflow built on **Sim** against 536 real GitHub issues, and reported 2 bugs upstream that Sim's co-founder fixed and merged. Built a **multi-agent platform** that generates and deploys full-stack apps behind deterministic code checks, and **citecheck**, which found 0 fabricated quotes across 354 LLM citations and 9 bugs in its own checker. MS Computer Science, Pace University.

Technical Skills

Languages: Python, SQL, JavaScript, TypeScript
LLM & Agentic Systems: LangGraph, LangChain, Anthropic API, OpenAI API, MCP, RAG (hybrid search, reranking), Embeddings, ChromaDB, Structured Outputs (Pydantic v2), Prompt Engineering, Langfuse
Evaluation & Reliability: Eval Harnesses, LLM-as-Judge, Mutation Testing, Deterministic Guardrails, CI Regression Gating, Held-out Test Splits, Bootstrap Confidence Intervals
Voice & Multimodal: Retell Voice Agents, Whisper, Meta FAIR TRIBE v2
Backend & APIs: FastAPI, REST APIs, PostgreSQL, Redis, Docker, Nginx, React, Vite
Cloud & Tooling: AWS (EC2, S3), GitHub Actions, Docker Compose, pytest, PyPI, uv
Machine Learning: PyTorch, Scikit-learn, Hugging Face, Pandas, NumPy
Domain: ESI Triage (Healthcare), SEC Filings and SOX Internal Controls

Open Source

- **Sim** (simstudioai/sim): reported 2 bugs found while self-hosting and evaluating a workflow, both fixed and merged by Sim's co-founder (#8353, #8354); opened a docs PR correcting the Evaluator block's documented settings after checking them against the source (#8448, open)
- **Meta FAIR TRIBE v2** (facebookresearch/tribev2): submitted PR #20 (open, CLA signed, checks passing) adding an int8 `compute_type` fallback for WhisperX, so the full pipeline runs on Apple Silicon and CPU-only machines

Independent Projects

Autonomous AI Engineering Platform: Multi-Agent Code Generation with a Deterministic Code Integrity Engine
Python, FastAPI, Next.js, PostgreSQL, Docker, OpenAI/Anthropic/Gemini
Code: github.com/Rajkumar2002-Rk/ai-org

- **Architected a 10-stage multi-agent LLM pipeline** (requirements interview, architecture, code generation, security review, QA, deploy) that turns a plain-English brief into a deployed full-stack app, in ∼15.7k lines of async Python with per-stage model routing across OpenAI, Anthropic, and Gemini to control cost
- **Built a Code Integrity Engine of deterministic, LLM-free checks**: AST symbol resolution, dangling foreign keys, hallucinated packages verified with real pip installs, esbuild parse checks, and duplicate routes. Each check sends a targeted repair back to the agent and retries, and each one came from a real failed run and is locked in with a regression test
- **Diagnosed and closed 55+ documented defects across 20+ paid end-to-end runs**, including a security reviewer whose own auto-fixes rewrote files it had already certified and quietly put bugs back in. Fixed it with a gate that re-runs the original build checks on any change made after certification

sim-triage-eval: Evaluating an AI Triage Workflow Built on Sim
Sim (self-hosted), Claude Haiku 4.5, Python, pytest, GitHub Actions
Code: github.com/Rajkumar2002-Rk/sim-triage-eval

- **Built an issue-triage workflow on self-hosted Sim**, deployed it as an API, and scored it on **536 real issues** from Sim and Kubernetes against maintainers' own labels, with a dev/test split and an always-guess-the-majority baseline for every field
- **Mutation-tested code checks against Sim's LLM-judge Evaluator** across 12 injected error types: code checks caught 100% of invented numbers and broken structure but 0% of summaries with flipped meaning, while the judge caught 88 to 100% of injected errors yet only 41 to 54% of the model's real mistakes
- **Blind hand review showed most judge flags on correct outputs were false alarms** (0 of 8 flagged summaries were wrong); a calibrated prompt raised Kubernetes priority accuracy from 24% to 33% and within-one-level accuracy from 70% to 87%, stopping there to avoid fitting the test set

citecheck: Citation Verification for LLM-Extracted Audit Claims
Python, Pydantic v2, Anthropic API, pytest, GitHub Actions, uv
Code: github.com/Rajkumar2002-Rk/citecheck

- **Built a harness that checks every citation an LLM attaches** to claims extracted from 30 real SEC Item 9A filings, using deterministic checks that never call a model; hand-labeled all 30 filings so the answer key was not written by the system under test
- **Measured 98.9% citation integrity and 0 fabricated quotes across 354 cited fields**, and found 9 defects in the checker itself against 3 model errors, including one that produced a false 23% claim-error rate when the real figure was 10%
- **Mutation-tested the checker**: 92% of 539 injected defects caught (100% of fabricated quotes and flipped values), gated in CI at `--fail-under 0.85`

voice-agent-qa: Deterministic QA Harness for Voice Agents

Retell, Python, pytest, GitHub Actions

Code: github.com/Rajkumar2002-Rk/voice-agent-qa

- **Built a QA harness for Retell appointment-booking agents** with scripted adversarial callers and a rules-based scorer that names which field was wrong; ran **102 real calls** with every transcript committed
- **Found voice accuracy fell to 67 to 73% against 95% over text**, almost all on names misheard by speech-to-text, which a text-only test suite never sees; a hardened prompt cut unsafe wrong bookings from 10 to 2 by refusing to confirm unverified data

MedEval: Healthcare Triage Assistant with LLM Safety Architecture

LangGraph, GPT-4o-mini, FastAPI, Pydantic v2, Langfuse, React, TypeScript, Docker, AWS EC2, Nginx

Code: github.com/Rajkumar2002-Rk/medeval

- **Architected** an ESI triage system where a deterministic Python rules engine (**68 YAML-encoded clinical rules** with handbook citations) makes every urgency decision; the LLM is barred from choosing the level and only extracts typed facts (a 95-field Pydantic v2 contract) and explains the result
- **Reduced dangerous under-triage from 20% to 10%** by tuning extraction prompts against a 50-case suite built from ESI handbook practice cases, tracking exact and tolerant accuracy, under- and over-triage, and latency p50/p95
- **Published medeval-harness v0.1.0 on PyPI**, a CLI eval tool with `--fail-under` gating and standard exit codes, plus a GitHub Actions workflow that fails the build when accuracy drops below 60%

FinTel: AI Financial Research Agent

LangGraph, GPT-4o, FastAPI, yfinance, Tavily, Redis, Docker, AWS EC2, Nginx

Code: github.com/Rajkumar2002-Rk/financial-research-agent

- **Architected** a **9-node async LangGraph pipeline** that issues BUY/HOLD/SELL recommendations in **under 30 seconds** from live market data, company fundamentals, and news sentiment
- **Kept GPT-4o out of the decision**: a scoring engine (Technical 0–25, Fundamental 0–40, Sentiment 0–15) makes every call, and the model only classifies news sentiment as one scored input and writes the explanation; conflict-aware logic forces HOLD when signals disagree and blocks calls below 20% confidence

Enterprise RAG Chatbot: AI Document Intelligence

LangChain, ChromaDB, OpenAI API, FastAPI, Streamlit, Redis, Docker, AWS EC2, Nginx

Code: github.com/Rajkumar2002-Rk/rag-chatbot

- **Built and deployed** a RAG system end to end: PDF ingestion, chunking, OpenAI embeddings, a persistent ChromaDB store, scored cosine retrieval with a reranking pass (MMR and hybrid BM25 built in as alternatives), and streaming generation behind Nginx with HTTPS
- **Added guardrails** that gate the LLM call on retrieval confidence, so weak or empty retrieval returns a fallback instead of a guess, plus a `[SOURCE N: filename, Page: X]` format so every answer traces back to retrieved text

Education

Pace University — New York, NY
Master of Science in Computer Science — *Sep 2023 – Apr 2025*

Amrita Vishwa Vidyapeetham — Bengaluru, India
Bachelor of Technology in Artificial Intelligence — *Jul 2019 – Apr 2023*

Certifications

AWS Certified Cloud Practitioner, Amazon Web Services — 2025