

Kurri Govinda Reddy

AI / ML Engineer | AI Enablement & Platform (LLM Gateway / Routing)

govindreddy.kurri@gmail.com | +91 9494218866 | linkedin.com/in/govindareddykurri | github.com/codraven |
codraven.github.io/portfolio

Summary

AI / ML Engineer who builds the core building blocks of AI enablement platforms: provider-agnostic LLM routing with multi-provider failover, end-to-end observability and full request/response audit logging, multi-tenant access isolation, and API-based services. Shipped an agentic insurance claims fraud-detection platform, an autonomous multi-agent code-generation system, a multi-tenant RAG chatbot, an AI governance / compliance inventory scanner, and an open-source AI legal-operations agent. Hands-on with custom LLM routing across 10+ providers (Mistral 7B, LLaMA 3.2, Claude 3.5 Vision, Qwen 2.5 Coder, Gemini, Groq, Ollama), agent tooling, function calling, RAG over pgvector, document processing pipelines, and FastAPI / SQLAlchemy services. Drive the architecture and design decisions myself and execute in cross-functional settings.

Technical Skills

AI Enablement & Gateway Patterns: Provider-Agnostic LLM Routing, Multi-Provider Failover / Fallback, Cost-Tier Model Selection, Policy Enforcement (permission-gated actions), Multi-Tenant Access Isolation, Observability & Auditability (OpenTelemetry), AI Governance / Compliance Inventory, API-Based Platforms, BYOK, LiteLLM (hands-on)

Languages and Databases: Python, SQL, PostgreSQL, MySQL, MongoDB, Redis

AI / ML / LLM: Large Language Models, AI Agents, Multi-Agent Systems, LangChain, LangGraph, LangSmith, Retrieval-Augmented Generation (RAG), Vector Databases (pgvector), Embeddings, Document Chunking, Vision Language Models, Prompt Engineering, Function Calling, Agent Tool Development, Hugging Face, Model Context Protocol (MCP), Generative AI

Backend and APIs: FastAPI, Pydantic, SQLAlchemy, Alembic, REST API Design, Async Programming, Server-Sent Events (SSE), OpenTelemetry

Cloud and DevOps: Google Cloud Platform (GCP), Amazon Web Services (AWS), Azure, Docker, CI/CD Pipelines, Git, GitHub Actions

Frontend: TypeScript, Angular Material, JavaScript, HTML, CSS, SCSS

Tools and Other: Power BI, Tableau, SAP Analytics Cloud

Experience

MONOCEPT | Software Engineer

Jan 2026 – Present

Project: AgentFX – Agentic Claims & Fraud Automation (Insurance)

- **Tech Stack:** Python 3.12, FastAPI, Pydantic v2, LiteLLM, PostgreSQL + pgvector, SQLAlchemy, Alembic, scikit-learn, Angular 17.
- Architected a config-driven agentic framework where each customer is onboarded and tuned entirely from configuration – new tenants, routing, rule packs, and pipelines go live without redeploying.
- Built the claims fraud-detection pipeline as cooperating agents that hand off agent-to-agent (extraction, aggregation, and multi-dimension scoring) to reach a single final fraud verdict with rationale.
- Added document forensics – tampering detection (EXIF / ELA / PDF-structure) and AI-generated document detection – to catch manipulated or synthetic claim documents before straight-through processing.
- Built the RAG stack: watcher-driven ingestion, pluggable chunkers, hybrid dense (pgvector) + BM25 retrieval fused via RRF, cross-encoder rerank, and grounded citations.
- Built a call-intelligence pipeline (transcription, analysis, and quality / lead scoring) as a pluggable agent pack.
- Hardened LLM access behind a provider-agnostic AI gateway: only authorized consumers can reach the models, so any other machine that pings the LLMs is denied (per-consumer keyed access).

Project: NeuralShell – Autonomous Multi-Agent Code Generation

- **Tech Stack:** Python 3.11+, FastAPI, WebSocket, Ollama (qwen2.5-coder:14b/7b), MySQL, Docker, Textual (TUI), ruff, mypy.
- Built NeuralShell, an autonomous multi-agent system for automated code writing, review, testing, and deployment; validated against an internal .NET application (approx. 90 percent of test workload) and a Python application (approx. 10 percent), with code-generation quality rated at 90 percent through manual review.
- Ported and extended the reusable DAG workflow framework, introducing ArchitectureNode for intelligent codebase parsing and change-location detection prior to any code modification.

- Implemented parallel agent nodes (CodeWritingNode, ReviewNode, TestingNode, DeployNode) coordinated by a WorkflowRouter that dynamically delegates tasks across agents.
- Designed custom agent tools (file read/write, codebase navigation, context retrieval) exposed to LLM agents via function calling, including a self-updating context file refreshed at workflow start and completion to maintain project awareness across runs.
- Shipped a Textual-based TUI client supporting interactive mode, slash commands, dry-run previews, and target-directory scoping, deployable across any codebase regardless of language.

MONOCEPT | Software Intern

May 2025 – Jan 2026

Project: MozartBot – AI ChatBot

- **Tech Stack:** Python, FastAPI, LangChain, LangGraph, PostgreSQL, pgvector, TypeScript, Angular Material, SCSS, HTML5, Ollama providers.
- Architected a multi-node LangGraph workflow (retrieval to grading to generation) with semantic and late document chunking and checkpoint-based memory over pgvector; exposed FastAPI endpoints for multi-turn context recall, delivering end-to-end response latency of 5 to 7 seconds for document Q&A.
- Built a multi-format document processing pipeline for PDF, DOCX, Excel, CSV, and image (OCR) inputs, including PyMuPDF multi-page table extraction with token-aware chunking, feeding embedded content into a multi-tenant PostgreSQL+pgvector store with per-company database isolation (tenant-level access isolation).
- Integrated 8 to 10 LLM providers (Ollama, Google, etc.) – primarily Mistral 7B and LLaMA 3.2 – with hot-swappable model selection, achieving approximately 80 percent retrieval accuracy across 8 multi-turn conversation flows.
- Implemented SSE streaming for real-time chat responses and document upload progress across a multi-endpoint REST surface (chat, documents, products, sessions, companies, health) with async connection pooling and Alembic migrations.
- Built a responsive Angular Material frontend with real-time streaming for chat responses and document upload progress, delivering an interactive end-user UX.

Personal Projects

ScriptOCR – Multi-Script OCR Engine (Printed & Handwritten)

github.com/codraven/scriptocr

- **Tech Stack:** Python, ONNX Runtime, OpenCV, pyclicker, NumPy, Pillow.
- Built a multi-script OCR engine with automatic script detection that routes each text region to a dedicated recogniser – 6 languages (English, Hindi, Telugu, Kannada, Tamil, Malayalam), printed and handwritten.
- Extracts key/value pairs and tables from page geometry alone (no LLM), preserving coordinates for downstream validation.
- Trained the handwriting and Indic recognisers in-house and packaged the stack on ONNX Runtime for CPU-only, fully offline / air-gapped deployment with private model distribution.

Vouchsafe – AI/LLM Inventory & Compliance Scanner

github.com/codraven/vouchsafe

- **Tech Stack:** Python, Ed25519 signing, RFC 3161 timestamps, in-toto/SLSA attestations, Docker (multi-arch), GitHub Actions, cross-platform desktop installers.
- Built and launched Vouchsafe, a deterministic scanner that produces a signed inventory of every AI / LLM / agent / MCP touchpoint in a codebase, mapped to regulatory frameworks (EU AI Act, ISO/IEC 42001, NIST AI RMF, OWASP LLM Top 10) – supporting AI governance and approved-provider / model standards.
- Detects AI usage across 15 programming languages and 45+ provider SDKs (OpenAI, Anthropic, Azure, Google, AWS Bedrock, Mistral, Hugging Face), self-hosted runners (Ollama, vLLM, LM Studio), agent frameworks (LangChain, LangGraph, CrewAI), MCP servers, and vector / RAG platforms.
- Generates audit-grade evidence in multiple formats (PDF/A-3, SARIF, CycloneDX ML-BOM, SPDX AI Profile, in-toto / SLSA attestations) with Ed25519 signing and RFC 3161 timestamps for reproducible, tamper-evident auditability.
- Runs fully offline / air-gapped with deterministic, byte-identical output; 84 unit tests with cross-OS CI; shipped v0.3.0 with a native desktop GUI and multi-platform installers.

Legal Co-Pilot – Open-Source AI Legal Operations Agent

github.com/codraven/legal-copilot

- **Tech Stack:** Python 3.11+, FastAPI, async SQLAlchemy, Neon Postgres + pgvector, Pydantic v2, Alembic, PyMuPDF, httpx, Gemini 2.5 Flash, Groq, Ollama, MCP, ruff.
- Built an open-source, multi-provider AI legal-operations agent on a \$0 / month budget that ingests contract PDFs and runs the full lifecycle: extraction, risk analysis, redlining, drafted negotiation email, action layer, and long-term memory.

- Designed a custom provider-agnostic LLM routing layer (gateway-style, no LangChain) with hot-swappable backends across 4+ providers (Gemini, Groq, Ollama Cloud, Ollama Local) supporting cost-tier selection, BYOK, and automatic multi-provider failover on rate-limit and quota errors.
- Hardened the routing layer with 76 unit tests (212 total across 8 modules) covering routing logic, fallback chains, and rate-limit handling.
- Built an MCP-based action layer with a permission gate that enforces policy before any action is dispatched, keeping automated actions governed and auditable.
- Implemented a Pydantic-validated extraction pipeline producing structured clause-level output (parties, dates, governing law, obligations, signatures) and a risk analyzer emitting findings with severity, confidence, and visible reasoning per finding.
- Persisted long-term, counterparty-scoped agent memory on Neon Postgres with pgvector embeddings (gemini-embedding-001, 3072-dim) and Alembic-managed migrations.
- Built a reproducible cross-provider benchmark on the public CUAD legal dataset, scoring extraction precision/recall per clause type across providers; head-to-head runs showed Llama 3.3 70B (Groq) outperforming Gemini 2.5 Flash on F1 and latency by 17x, achieving 100 percent precision on governing_law, auto_renewal, and liability_cap clause matching.

Education

Vasireddy Venkatadri Institute of Technology

2021 – 2025

- B.Tech in Computer Science and Engineering (Artificial Intelligence and Machine Learning)

Certifications

- Associate Cloud Engineer – Google Cloud
- Generative AI Essentials Professional – Microsoft
- Data Analysis Essentials Professional – Microsoft
- AI for Everyone – DeepLearning.AI
- Docker Foundations Professional – Docker