SUBRATA MONDAL

FOUNDING AI ENGINEER | AI INFRASTRUCTURE & AGENTIC SYSTEMS ENGINEER
Bangalore, India | +91 6294505807 | subratasubha2@gmail.com | Portfolio | GitHub | LinkedIn

PROFESSIONAL SUMMARY

Founding AI Engineer who took an AI legal tech product from ideation to production as the Solo AI Engineer, serving **2,000+ users**. Built high-throughput agentic systems in async Python, saved **$12K/year** replacing Paid AI Native Rich Text Editor, cut LLM costs **5-10x**, and dropped P99 latency to **50ms**. Catches LLM regressions before code shipping to production via strict CI/CD evaluations.

TECHNICAL SKILLS

Backend: Python, FastAPI, Microservices, PostgreSQL, Vector DB, PG Vector, MongoDB Atlas, Redis, Message Queue Broker, REST APIs, WebSockets, SSE, AsyncIO, OpenTelemetry

Gen AI: LangChain, LangGraph, Multi Agent Orchestration, MCP, DSPy, LiteLLM, AI Agents, Agentic Workflows, RAG, CRAG, Hybrid RAG, Prompt Engineering, Evaluations (RAGAS, LLM-as-a-Judge), Hallucination Reduction, LangFuse, Guardrails

Infra: Docker, Kubernetes, AWS, Azure, CI/CD, KEDA, System Design, Latency Tuning, LLM FinOps & Cost Optimization

Tools: Claude Code, Cursor, Git, UV, Ruff, PyTest

Frontend: Next.js, React, TypeScript, Tailwind CSS, shadcn/ui, Zustand, React Query, Framer Motion

PROFESSIONAL EXPERIENCE

FOUNDING AI ENGINEER | LawWorld | Bangalore, India (On-site)

Aug 2024 - Jun 2026

Architected an AI Native Legal Tech Product end-to-end — production AI applications and scalable backend systems. Designed Agentic AI workflows and RAG pipelines serving 2,000+ users over a 170K-document corpus, prioritizing reliability, latency, and operational efficiency.

- **Replaced a $12K/year vendor subscription in-house** solo-building the agentic rich-text editor backend and its drafting orchestration layer — TipTap's paid AI toolkit only covered the editor surface, not the differentiating logic.
- **Saved ~$500/month in cloud overhead** building durable execution directly on MongoDB checkpoints and Azure Service Bus instead of adopting Temporal — sustains 3-hour long-running agents with automatic crash-resume and zero data loss.
- **Standardized on Model Context Protocol (MCP) for secure tool access** — deployed Dockerized servers in client VPCs with SSE tunnels for JSON-RPC schema discovery, executing agents without hardcoded tools or raw database credentials.
- **Gated architecture decisions on strict blind-evals** — defined accuracy/cost thresholds upfront against 550 production documents, forcing a data-driven choice between ReAct and structured-output rather than rationalizing decisions post-hoc.
- **Held P99 latency at 50ms** via an L2 Redis Stack semantic cache. Achieved a 20-30% hit rate on repetitive queries, offsetting ~$15,000/month in LLM inference spend using a $250/month cache at million-call scale.
- **Fan-out parallel search cut retrieval latency 1.3s to 443ms** then closed the multi-tenant gap structurally — session junctions are injected server-side into the vector pre-filter, so a query can narrow the search space but never widen it across tenants.
- **Cut per-unit inference cost 5-10x via Swarm model-tiering** — swapped GPT-5.2 for GPT-5.4-mini in extraction while keeping Qwen 3.5 for routing. Validated against versioned pricing across hundreds of legal Acts, scaling toward India's full statutory corpus.
- **Blocked hallucinated-package execution across federated agents** — implemented deterministic glass-box gates on an Agent-to-Agent (A2A) bus, running dependency-provenance checks against an allow-list before any untrusted payload executes.

Prior Experience (pre-LawWorld, short-term)

- **Python Developer, AlgoHype Analytics — Jul-Aug 2024 (contract):** Engineered synthetic-data generation pipelines using Azure OpenAI and shipped a QA microservice for automated data validation.
- **ML Engineer, Omdena — Nov-Dec 2023 (short-term):** Developed an LLM-powered interview-prep chatbot, architecting the conversational flow and retrieval logic.

OPEN-SOURCE & ENGINEERING PROJECTS

Argus: Multi-Agent Deep-Research Engine

GitHub

- **Hit flawless context recall on a 48-item research benchmark** with a planner, parallel-searcher, synthesizer orchestration engine, hand-built in async Python with no LangChain dependency.
- **Gates every deployment on 8 eval metrics** — RAGAS faithfulness, context precision, and an LLM-as-a-judge pipeline, including abstention scoring so the system declines unanswerable queries instead of guessing.

LLM-as-a-Judge Calibration Engine

GitHub

- **Moved judge agreement from Kappa 0.12 to 0.9261** across 5 architecture iterations — built a 10-class voice-agent intent taxonomy with blind field stripping and a stratified golden set to make the calibration honest.
- **Replaced manual prompt tuning with a closed loop** — an offline GEPA (Generate, Evaluate, Propose, Accept) cycle reads judge failure traces and rewrites system prompts itself, compiled through a deterministic DSPy graph with enforced output typing.

bare-agent: Multi-Agent Orchestration Runtime

PyPI | GitHub

- **Published a zero-dependency agent runtime to PyPI** — bare-agent compiles any complex multi-agent graph (utilizing LangGraph/MCP patterns) into a highly durable Python orchestration script (tool registry, 3-axis budget, HITL permission gates, LiteLLM gateway).
- **Made the agent loop hermetically testable** — a stateless reducer over an explicit message list gives free serialization for durability and eject-to-code compilation, verified by 29 tests with LLM and Redis fully faked, no daemon required.
- **Shipped a visual studio that compiles to real code** — a companion React Flow app (Next.js 16 / React 19) for graph-based agent authoring with live SSE token streaming, plus an Eject button producing a machine-verified standalone script.

EDUCATION

Bachelor of Technology in Computer Science Engineering (CSE) - Parul University — Graduated 2023