# RAGAVARSHINI R

SENIOR BACKEND ENGINEER · DISTRIBUTED SYSTEMS & AGENTIC AI

Chennai, India | +91 8526620740 | ragavs95@gmail.com

varshini39.github.io | linkedin.com/in/ragavarshini1812 | github.com/varshini39

## SUMMARY

Backend engineer with 9+ years building scalable, low-latency distributed systems for fintech and enterprise SaaS. Strong in Java and Spring Boot, REST and GraphQL API design, event-driven architecture on Apache Kafka, high-traffic query optimization, and zero-downtime database migrations on systems serving millions of users. Comfortable working up the stack into React and Next.js when a feature needs it. Currently building production agentic AI services, including agentic RAG retrieval routing, MCP servers, and multi-agent orchestration.

## EXPERIENCE

**Senior Software Engineer** | PayPal | Chennai 05/2025 – Present

- Built and shipped backend services and REST/GraphQL APIs for new checkout platform releases, adopted across 52+ EU countries.
- Engineered performance optimizations across the checkout module, reducing latency and producing measurable conversion gains.
- Re-engineered Amex delegation flows to minimize authentication overhead, raising transaction success rates.
- Architected an agentic AI service over BigQuery and Datadog that automatically surfaces root causes of user drop-offs.
- Automated real-time conversion monitoring and diagnostics, accelerating detection of production funnel leakage.
- Built custom Datadog reporting pipelines to speed up production debugging and performance analysis.
- Partnered with enterprise merchants to identify and resolve friction points in checkout integration flows.
- Served as Scrum Master, running Agile ceremonies and improving delivery predictability.

**Member of Technical Staff** | Zoho Corporation | Chennai 05/2017 – 05/2025

- Engineered scalability improvements to the Marketplace framework, reducing query execution time by 25%.
- Architected low-latency solutions for encounter functionality, improving system responsiveness and stability by 30%.
- Optimized high-traffic query patterns, cutting database load and sustaining 99.9% uptime SLAs.
- Executed zero-downtime database migrations for a platform serving millions of users.
- Integrated Apache Kafka for asynchronous, event-driven communication and real-time webhook processing.
- Streamlined logging across distributed services, improving traceability and reducing MTTR by 20%.
- Managed critical dependency and framework upgrades to hold security and performance baselines.
- Translated customer requirements into technical features, aligning the roadmap with business objectives.

## TECHNICAL SKILLS

**Languages:** Java, Python, TypeScript, SQL

**Backend & APIs:** Spring Boot, Spring Framework, Struts, Node.js, Express.js, REST and GraphQL API design, microservices, event-driven architecture

**Frontend:** React, Next.js, TypeScript, JavaScript, HTML5, CSS3

**Messaging & Streaming:** Apache Kafka, Zookeeper, event-driven architecture, webhook processing

**Databases & Data:** MySQL, MongoDB, BigQuery, query optimization, schema design, zero-downtime migrations

**Containers & Orchestration:** Docker, Kubernetes

**Build, CI/CD & Tooling:** Jenkins, Maven, Gradle, Bitbucket, Git

**Observability:** Datadog, Kibana, distributed logging & tracing, SLA/latency monitoring

**Agentic AI:** Agentic RAG and retrieval routing, CrewAI, AutoGen, LangChain, MCP servers, multi-agent orchestration, tool binding and guardrails, prompt engineering, Pinecone, LangSmith

## PROJECTS

**Multi-Agent Conversion Diagnostics** PayPal · Internal

- Architected a multi-agent pipeline that diagnoses checkout conversion dips end to end, scoped by country, merchant, or experiment.
- Structured it as an escalation ladder: a BigQuery conversion check gates the run, a diagnostic agent localizes the dip to a checkout stage from the drop-off funnels, and an analyzer agent runs a fixed query set per known problem class.

- Reserved open-ended search for a hypothesis agent invoked only when the deterministic tiers are inconclusive, querying event tables, Datadog, and call logs within a bounded surface and retrieving from a corpus of prior incident analyses, so recurring issues are recognized rather than re-investigated.
- Closed the loop with a report agent producing stakeholder-ready output: conversion movement, drop-off stage, root cause, and supporting statistics.
- Exposed BigQuery and Datadog to the agents as typed, read-only MCP servers with argument validation and scoped credentials, keeping production access out of the agent context.
- Stack: Python, multi-agent orchestration, MCP SDK, BigQuery, Datadog API.

**Payments Ledger with Read-Only MCP Server** GitHub

- Built a double-entry ledger whose five correctness invariants are enforced by PostgreSQL rather than the service: zero-sum via a deferred constraint trigger, append-only via a BEFORE UPDATE OR DELETE trigger, idempotency by unique constraint, and balances derived from entries rather than stored.
- Paired it with a read-only MCP server that holds no database credentials and reaches the ledger only through read HTTP endpoints, the same boundary any external consumer faces; every response cites the exact request it made.
- Deliberately omitted a raw-SQL tool, which would be the most flexible tool on the server and would move the trust boundary from the API to the model's judgment.
- Tested with Testcontainers against real PostgreSQL, never H2; an invariant contract test attacks each rule through raw SQL, bypassing the service, to prove the database is what refuses the write.
- Stack: Java 21, Spring Boot 3, PostgreSQL 16, Liquibase, Python MCP SDK, JUnit 5, Testcontainers.

**Agentic RAG Router (CrewAI)** GitHub

- Built an agentic RAG pipeline that routes each natural-language question to the retrieval path that fits it: semantic search over a PDF corpus, live web search, or a direct LLM answer with no retrieval at all.
- Split the system into two sequential crews, a router crew that classifies the question and a retriever plus answer crew constructed afterwards with exactly one tool bound to it, so an agent physically cannot call a tool outside its route. Conditional branching sits in plain Python between the crews, since CrewAI sequential process has no native branching.
- Separated retrieval from answer generation into two chained tasks so each agent output is inspectable, and constrained the answer agent to the retriever evidence. Out-of-scope questions return an explicit refusal rather than a hallucinated answer.
- Wrapped a LangChain Tavily search tool in a CrewAI BaseTool subclass to satisfy CrewAI tool validation, and reworked the pipeline onto kickoff_async so it runs inside the Jupyter event loop.
- Logged question, route, router reasoning, tool used, retrieved evidence, and final answer for every run to a CSV reasoning trace. Stack: Python, CrewAI, OpenAI GPT-4o-mini, Tavily, pandas, Jupyter.

**AutoGen Analysis Service** GitHub

- Built a FastAPI microservice exposing an AutoGen pipeline of four AssistantAgents, and wired it into an n8n workflow so the analysis runs as an automated job rather than a manual script.
- Added a mock mode flag so the service runs end to end without hitting live LLM APIs, making the pipeline testable and demonstrable offline.
- Wrote every run to a JSON Lines log for traceability, capturing inputs and agent outputs per invocation. Stack: Python, AutoGen, FastAPI, n8n.

**Flight Booking Application** GitHub

- Microservices-based booking platform covering authentication, flight scheduling, and ticketing, containerized and deployed to Kubernetes.
- Stack: Spring Boot, Docker, Kubernetes, Maven, REST microservices.

**Patient Health Information on Blockchain** GitHub

- Solidity smart contracts on Ethereum for tamper-evident storage and permissioned sharing of patient health records.
- Stack: Solidity, Ethereum, Web3.

## EDUCATION

**M.Tech, Software Engineering**, BITS Pilani

**B.Tech, Information Technology**, Karpagam College of Engineering, Coimbatore

## CERTIFICATIONS

- Professional Certificate Program in Blockchain, IIT Kanpur
- Applied Agentic AI: Systems, Design & Impact, Microsoft & Simplilearn