SRINIVASAN NAMBI

Senior Software Engineer

chat2srini@gmail.com
(980) 267-8227
Seattle, WA

SUMMARY

Senior software engineer with 10+ years building complex, scalable products from concept to production. Took a production AI agent from prototype to launch for 12,000+ users—integrating LLMs via Amazon Bedrock APIs, MCP Server, and AgentCore Runtime, with Bedrock Guardrails, prompt injection defenses, evals, and observability built in. Strong product instincts paired with hands-on security experience: led application security reviews and penetration testing, and independently explored protocol-level approaches to AI agent security and governance. Thrives on rapid prototyping, owning ambiguous problems end to end, and collaborating across research and go-to-market functions.

CORE SKILLS

- **AI-Powered Products:** LLM agents, multi-agent orchestration, Amazon Bedrock APIs, Bedrock Guardrails, agentic systems via MCP Server & AgentCore Runtime (Strands), Text2SQL, vector search, prompt engineering, agent evals & observability
- **Security:** Prompt injection defense, memory poisoning safeguards, agent threat modeling, application security reviews, penetration testing, secure architecture, agent identity & authorization design
- **Product Engineering:** Rapid prototyping, fast iteration, customer engagement, technical tradeoff analysis, zero-to-one delivery
- **Systems & Cloud:** Distributed systems, high-availability architecture, ETL pipelines, performance optimization; AWS (Bedrock, RDS, EC2, Lambda, Secrets Manager, Systems Manager)
- **Tools & Languages:** Python, Java, C#, SQL, JavaScript; Claude Code, Claude models (Sonnet 3.5 through Opus 4.7)

PROFESSIONAL EXPERIENCE

Amazon Web Services — Seattle, WA

Senior App Dev Engineer, Incentive Compensation Management | Dec 2016 – Present

- **Built and shipped BOB, a production AI agent for AWS sellers**, answering compensation and policy questions in natural language. Led technical design as lead developer; integrated LLMs through Amazon Bedrock APIs with dual integration paths via MCP Server and AgentCore Runtime (Strands), plus Text2SQL and vector search. Delivered the initial prototype in 6 weeks—40% faster than estimated—then advanced to a production pilot with the operations team (~50 users), with rollout to 12,000+ sellers underway over ~6 months. Sub-15-second responses; projected to cut support tickets and reduce resolution time from days to minutes.
- **Owned security, evaluation, and observability for the agent**. Implemented Bedrock Guardrails, prompt injection filters, and safeguards against memory poisoning. Performed agent threat modeling and implemented security enforcement using MCP-based tooling. Worked closely with the application security team to obtain formal security approval for production launch. Designed eval frameworks to measure response quality and accuracy, and built observability to monitor latency, tool calls, and failure modes in production.
- **Built a 10-agent SDLC pipeline for agentic application development** on Kiro, an AWS platform for building agentic applications, using AgentCore Runtime and Strands with composable skills, agent SOPs, and an agent handoff protocol. Distributed via internal package manager for company-wide installation and reuse across Amazon teams.
- **Drove a mission-critical cloud platform migration** (Varicent SaaS) for 12,000+ users requiring data parity to the nth decimal. Led application security reviews, penetration testing coordination, ETL pipeline redesign for API-based access, and conversion of 700+ SQL scripts. Built a custom GenAI-

powered conversion tool that saved 28+ engineering days and caught vendor errors before production. Delivered on schedule with fewer than 50 minor post-launch issues.

- **Built an incentive component workflow from inception** supporting 6,000+ sellers and 35% of annual compensation. Owned the full lifecycle and automated 400 hours/quarter of manual work. Handled complex business logic (leave-of-absence treatments, dynamic capping, break-in scenarios). Operating reliably in production for 3 years.
- **Led security onboarding for Unfabric compliance**, coordinating internal penetration testing and resolving critical findings (input validation, Content Security Policy, error handling), which accelerated downstream migration approvals. Engineered reliability safeguards including automated validation guard rails protecting payout integrity for 11,000+ payees.

Earlier Experience

- **Hughes Network Systems** — Software Engineer, MTS L2 (2016): Led Salesforce Service Cloud integration and customer notification module.
- **Tresata Analytics** — Software Developer Intern (2015): Built a retail analytics web app with Node.js REST services and MongoDB.
- **HCL Technologies** — Senior Software Engineer (2010–2014): Led banking application features; designed a document-generation framework and SQL automation tooling that cut development effort ~50%.

THOUGHT LEADERSHIP & WRITING

- **The Building Blocks That Outlive Your Agent Framework** — Published on AWS Builder Center (May 2026). Technical article on designing durable agentic systems. Covers five core primitives (tools, skills, memory, data access control, streaming), progressive growth model from Stage 1 to Stage 4, autonomy gates, evaluation discipline, and failure mode mitigation for production agent systems.

INDEPENDENT RESEARCH & DESIGN

Self-directed exploration of open problems in multi-agent AI systems, brainstormed and developed in collaboration with Claude. Conceptual; not implemented or published.

- **Agent Governance Protocol (AGP):** Proposed a security and governance layer for delegated AI agents—addressing portable agent identity, on-behalf-of delegation with scope attenuation, infrastructure-level scope enforcement, cross-domain revocation, and tamper-evident audit. Composes existing standards (OAuth 2.1, FAPI 2.0, MCP, SCIM, OIDC-A) with a quantum-aware cryptographic substrate (ML-DSA-65, ML-KEM-768).
- **Dabba-Agent Protocol (DAP):** Proposed a vendor-neutral state-portability layer for multi-agent systems in which task state travels inside a cryptographically sealed, append-only packet rather than a central orchestrator. Built on MCP and A2A; draws on distributed-systems primitives (CRDTs, SWIM membership, TEE remote attestation).

PATENTS & INTELLECTUAL PROPERTY

- **Patent Pending:** Adaptive Execution Graph with Self-Optimizing Dependencies for Multi-Agent Orchestration (Amazon Web Services)

EDUCATION

M.S. Computer Science — University of North Carolina at Charlotte | GPA 3.9/4.0 | 2015

B.E. (Engineering) — Anna University, Chennai, India | GPA 3.7/4.0 | 2010