Param Shrikant Chaudhari/paramsc.dev

About me: +1 (984)-291-5335 | Mail: paramchaudhari.work@gmail.com | Github: certainly-param | LinkedIn: Param Chaudhari | Blog

*I am an **Engineer** who specializes in the full lifecycle of AI and hardware products. I have expertise in designing **AI-native applications** and **RAG pipelines**. I have specialized in systems programming and hardware security, demonstrating a track record of **optimizing low-level inference** and **verifying OS** architectures. I have had a significant impact on open-source across both **AI and hardware** ecosystems.*

Education

Masters in Computer Science, *University of North Carolina at Chapel Hill (UNC), NC, USA* **August 2024 – May 2026**

Bachelor's in Computer Engineering (Honours in AI/ML), *PES Modern College, Pune, India, **GPA: 9.42/10*** **August 2019 – May 2023**

Skills

Languages: *Python, C/C++, Rust, Go, TypeScript/JavaScript, Bash, C#, SystemVerilog*

AI/ML & Agentic Systems: *MCP, PyTorch, LLMs, vLLM, LangGraph, LangSmith, RAG, Hugging Face, Vector Databases, Hybrid Search, Model Quantization*

Backend & Systems Architecture: *FastAPI, Node.js, gRPC, SQLite, RESTful APIs, Microservices, Upstash, PostgreSQL, Redis, Kafka, Docker, K8s, Linux Kernel, RISC-V.*

Cloud, DevOps & Observability: *AWS, GCP, Git/GitHub/GitLab, CI/CD Pipelines, OpenTelemetry, LangFuse, Serverless (Next.js), ELK Stack*

Testing & Quality Assurance: *Pytest, Vitest, System Verification, Formal Methods*

Security & Authentication: *OAuth, IAM, OpenSSL, Biscuit, NextAuth, Zero-Knowledge Proofs (ZKPs), Consensus Algorithms, Cryptography*

Experience

Research Assistant, *Department of Computer Science, UNC, Chapel Hill* **August 2025 – Present**

- Using **Python** and **SystemVerilog**, I architected and developed end-to-end **formal verification** frameworks for **RISC-V (CVA6)** architectures, implementing **reachability analysis** and proactively identifying security vulnerabilities in RTL descriptions.
- I worked closely with **INTEL** to develop automated security properties, **unit testing** suites**,** and **CI/CD** pipelines. The framework developed showcased **a 90x speedup with 100% deterministic results** for large-scale open-source processor cores.
- I conducted an investigation into **RL-driven heuristics** to track information flow in **RISC-V** architectures to improve security and system reliability.

IT Technician, *Information Technology Services (ITS), UNC, Chapel Hill* **January 2025 – August 2025**

- Resolved high-priority hardware, software, and network infrastructure issues for a campus-wide network of **20,000+ users**.
- I utilized the **ELK Stack** for the purpose of real-time log aggregation and network monitoring.

Computer Programmer & Researcher, *Defense Research & Development Organization, India* **August 2023 – August 2024**

- Managed the end-to-end **SDLC** of **post-quantum cryptographic** libraries using **Agile/Scrum** methodologies, implementing high-performance **C++** and **Python** algorithms for quantum-resistant encryption.
- Optimized **error-correcting code** modules for secure communication, integrating custom test suites into **CI/CD** pipelines to ensure **99.9% reliability** and a **3% BER reduction** on a specialized **Linux-based OS**

Projects

VAC Protocol: Verifiable Agent Credentials (Rust, MCP, Biscuit, Docker)

- Architected a capability-based security framework in **Rust** that shifts from identity-based to task-scoped credentials, enabling fine-grained policies and instant revocation for **AI agents**, solving the "**over-privileged agent**" problem.
- Implemented receipt-based state transitions with **Biscuit** tokens, heartbeat-based liveness checks, and fail-closed **Datalog** policies, achieving **sub-100 ms latency** per request and **zero-trust delegation** to prevent privilege escalation.

TraceLens: Visual Debugger for LangGraph Workflows (Python, FastAPI, OpenTelemetry, SQLite)

- Created a detailed diagnostic center for **LangGraph** workflows using **OpenTelemetry** and **SQLite**, helping to solve the **"Silent Failure"** problem in AI systems by allowing real-time visualization and the ability to go back in time to debug issues.
- Engineered high-performance sidecar instrumentation that enables zero-code debugging for AI agents, maintaining high throughput for **50+ concurrent workflows** with minimal **latency (<10ms lookup / <20ms API)**.

Serverless RAG Chatbot: Scale-to-Zero RAG Architecture (Next.js, React, Upstash, gRPC)

- Architected a production-ready **RAG** chatbot with true scale-to-zero architecture using Next.js 16, Gemini, and Upstash, achieving **$0/month cost** when idle and **sub-50ms** cold starts on Vercel Edge runtime, **eliminating the $50/month minimum cost floor** of alternatives like **Pinecone**.
- Implemented semantic caching using **Upstash Vecto**r's built-in embedding (**BAAI/bge-base-en-v1.5**) and a **gRPC**-based gateway, achieving **30-60% cost reduction** through intelligent query caching and client-side parsing to avoid Edge runtime limits, reducing server costs to zero.

Garuda: RISC-V ML Accelerator (Python, SystemVerilog, CVA6, CVXIF)

- Designed a high-performance **RISC-V coprocessor** in **SystemVerilog** that extends RISC-V with custom **INT8** multiply-accumulate (MAC) instructions, achieving **7.5-9× latency reduction (p99: 307→34 cycles)** for batch-1 attention microkernels compared to standard CPU execution.
- Integrated the coprocessor with **CVA6 RISC-V CPU** using **CVXIF** interface, implementing **attention microkernel engine** with deterministic loop execution that eliminates CPU dispatch overhead, optimizing neural network inference latency through hardware-software co-design.

Open-source contributions

Pydantic-AI: Resolved high-concurrency race conditions by implementing asynchronous locking for **thread-safe** usage tracking and currently collaborating to architect a universal structured citation system supporting **OpenAI, Anthropic, Google/Gemini**, and **Perplexity** providers to improve AI response transparency.

vLLM: Developed special checks for **NVIDIA Blackwell GPUs (SM100+)** to fix key problems with **INT8** quantization by adding automatic backup options for **FP8** and enhancing how kernels are loaded.

MCP Python SDK: Developed a diagnostic suite to find and confirm problems with resource leaks in the **Streamable HTTP** transport, highlighting serious risks of running out of connection pools during failed **SSE streaming event**s.

Papers

Comparing Consensus in Vehicular Ad Hoc Networks: An Analytical Review: Published in: JETIR | Vol 10 | Issue 5 | May 2023.