RAJ MANDALIYA

Los Angeles, CA (Ready to Relocate) | 714-684-6690 | raj.contact2206@gmail.com | Ready to Relocate | LinkedIn | GitHub | Portfolio

SUMMARY

Software Development Engineer II with senior-level expertise in building performance-critical backend systems using Rust, C++, and Java. Designed async Rust services at Cloudflare that handle 10M+ daily requests at sub-50 ms p95 latency and led a Go/Python-to-Rust migration that reduced CPU usage by 25 %. Secured high-throughput payment backends at Razorpay, delivering zero security breaches for 2,000+ users. Aiming to apply this performance-focused experience to drive scalable, secure services for the target organization.

WORK EXPERIENCE

Cloudflare **Nov 2024 - Jun 2026**

Software Development Engineer II

- Designed and implemented scalable Rust backend services processing 10M+ requests daily at p95 latency under 50ms, using Tokio and Actix Web for async microservices - 35% throughput improvement over the previous Node.js implementation.
- Led migration of legacy Go/Python services to Rust, cutting CPU usage 25% and eliminating a class of segfaults and race conditions that had driven 40% of production incidents.
- Built distributed caching and rate limiting with Redis and Rust workers, increasing request handling speed 30% and removing downtime caused by traffic spikes.
- Contributed to internal Rust libraries and tooling used by 3+ teams; deployed via Docker and Kubernetes at 99.99% uptime.

Razorpay **Jun 2022 - Dec 2023**

Software Engineer II

- Engineered OAuth2, TLS, and SQL injection defenses for high-throughput payment backends (Java, Spring Boot, AWS); zero reported security breaches across 2,000+ active users at 99.99% uptime.
- Designed and optimized 15+ production REST APIs (JAX-RS, Hibernate, PostgreSQL), reducing p99 query latency 30% under peak load.
- Architected event-driven payment pipelines on Apache Kafka for real-time transaction streaming with guaranteed delivery and fault-tolerant message handling; decomposed monolithic auth and payment services into fault-isolated microservices and raised test coverage above 85%.

OPEN SOURCE

infra-controller (NICo) **2026**

NVIDIA

- Fixed a production regression where DPU agent self-upgrade silently failed on every clean-tag release, caused by a version mismatch between carbide-api build_version and deb package stamping; fix spanned three Makefile.toml files including the nested PXE docker build path. A director opened a follow-up issue based on the approach. (Merged)
- Root-caused a cloud-init regression where tenant-provided network settings were silently ignored on every bare-metal provision; added a /network-config PXE route seeded into the NoCloud directory, verified end-to-end against real cloud-init. Also closed a host-mode test-coverage gap for NvidiaDgxVr BMC.

OPEN SOURCE

Hardware-Accurate Electronic Exchange (C++20) tickpipe **2026**

tickpipe

- Built a C++20 exchange that replays a full NASDAQ TotalView-ITCH 5.0 trading day - 8.7 GB, 282M messages - and reconstructs the order book message by message; complete decoder for all 21 message types with per-type length validation, zero malformed messages across the entire file.
- Modeled the matching engine as a 5-stage FPGA pipeline using a price-indexed ladder (slot = (price – base) / tick) and a priority-encoded occupancy bitmap for top-of-book: 2x faster than an STL control implementation with a tighter tail (40 ns vs 82 ns mean, 55 ns vs 162 ns at p99).
- Implemented an allocation-free order-reference resolution layer over a fixed-capacity open-addressing table, resolving 1.3M references with zero unresolved and zero orphaned operations; FNV-1a digests make every run bit-reproducible across machines and compilers.

Custom 16-bit CPU Emulator (Rust) cpu16 **Dec 2024 - Apr 2025**

cpu16

- Designed a complete 16-bit CPU from first principles: custom 35-instruction ISA with fixed-width encoding, two-pass assembler with forward-reference resolution, and a 5-stage in-order pipeline with RAW register and flag hazard stall detection plus 2-cycle branch flush.
- Built a direct-mapped write-through L1 cache classifying misses as cold or conflict; authored 6 assembly programs demonstrating real cache thrashing patterns. 49 integration tests across 5 releases; CI enforces fmt, clippy -D warnings, and the full suite on every push.

EDUCATION

Pace University **Jan 2024 - Dec 2025**

Master of Computer Science

- **Coursework:** Algorithms & Data Structures, Distributed Systems, Object-Oriented Programming, AI & ML

SKILLS

Languages: Rust, C++, Java, Python, Go, TypeScript, Solidity
Systems: CPU architecture, pipeline hazard detection, cache simulation, binary protocols, memory-mapped I/O, allocation-free hot paths, NUMA/core pinning, latency measurement
Backend & Infra: Tokio, Actix Web, Spring Boot, FastAPI, Hibernate, Kafka, Redis, Kubernetes, Docker, Terraform, AWS, GCP, GitHub Actions
Data: PostgreSQL, MySQL, Redshift, Snowflake, Amazon S3
Certifications: AWS Certified Developer – Associate, Google Professional Cloud Developer, Rust Foundation