Venkata Sai Surya Rao Beeraka

(312) 669-4463 | saisurya.beeraka@gmail.com | Linkedin | Portfolio | GitHub | Chicago,IL

Professional Summary

Software Engineer experienced in designing scalable backend systems and distributed architectures using Java, Spring Boot, and SQL. Focused on building microservices and multithreaded systems with high reliability and performance. Improved query efficiency by 35% and supported resilient pipelines handling 10K+ transactions daily in Agile and CI/CD environments.

Education

Master of Science in Computer Science August 2023 - May 2025
University of Illinois At Chicago, USA

Bachelor of Science in Computer Science August 2017 - June 2021
Jain University, India

Experience

Carelon Global Solutions (Elevance Health), India | Software Engineer October 2022 - July 2023

- Refactored monolithic workflows into resilient, event-driven Java/Spring Boot microservices using Kafka, enabling high-throughput data handling (15K+ XML feeds/month) with low-latency and fault-tolerant recovery during system loads.
- Constructed low-latency data pipelines with Microsoft SQL Server(T-SQL & SSIS), optimizing stored procedures and indexing strategies to enhance query performance by 35% during peak loads of 10k+ daily healthcare transactions.
- Orchestrated proactive log monitoring on Unix and Ipswitch servers, resolving file-transfer errors within minutes and preserving **99.9% system availability** for the agent-on-boarding process.
- Collaborated with QA automation to wire JUnit XML patterns into Jenkins CI/CD workflow, preventing malformed reports and boosting test result reliability by 20%.

Carelon Global Solutions (Elevance Health), India | Associate Software Engineer August 2021 - September 2022

- Executed and maintained Agent Commission Engine Tiering batch job for commission processing, automating bonus calculations for 3,000+ agents using Oracle PL/SQL reduced processing time by 50% and eliminated $200K in annual overpayments.
- Optimized PL/SQL procedures and database indexes on Oracle ACE-T bonus tables, cutting quarter-end processing time from 8 to 4 hours and increasing nightly batch capacity, resulting in a 30% system performance improvement.
- Led a Git feature-branch and protected-main workflow with mandatory peer reviews and CI build checks, standardizing merges and reducing ACE-T integration defects by 25%.
- Prevented database deadlocks in high-traffic Spring Boot services by refactoring transaction boundaries and tuning isolation levels, keeping the platform fully available during peak hours and avoiding revenue-impacting outages.

Projects

Distributed, Fault-Tolerant Banking System | Java, Python, HTTP, RM-GFD-LFD, Active & Passive Replication

- Pioneered a Java-based RM-GFD-LFD (Replication Manager, Global Fault Detector, Local Fault Detector) control plane that reduced **fail-over latency 60 %** and scaled to 50 + nodes by coding asynchronous heartbeats and rank-based leadership.
- Orchestrated deterministic total-order consensus that sustained 10K TPS with zero divergence by implementing log-ID assignment, batching, and No-Op bridging to commit pending transactions during proposer rotation.
- Integrated active and passive replication modes via Python checkpoint streaming, achieving lossless recovery and 4× faster RTO by snapshotting states and syncing replicas via HTTP.

BitTorrent Client | Distributed Architecture, Java 17, TCP/IP, AWS EC2, SHA-1, Threads, BitSet

- Architected a distributed BitTorrent client with modular system design, multithreading, and memory-efficient buffers, orchestrating scalable peer connections while conserving heap under heavy swarm loads.
- Deployed the Java peer engine on an Amazon EC2 instance with an EBS-backed volume for persistent piece storage, ensuring high availability and consistent environment replication.
- Optimized end-to-end performance with adaptive unchoke logic, periodic tracker announces, and rare-piece prioritization, sustaining 750 MB downloads 35% faster while upholding a 10% seed-share resilience threshold.

Cloud-Native API Rate Limiter and Monitoring System — Node.js, Redis, Docker, Kubernetes, Prometheus, Grafana

- Engineered a containerized Node.js API microservice with Redis-backed rate limiting, deployed via Kubernetes (Minikube), controlling abusive traffic with dynamic Retry-After headers and TTL-based counters for scalable request governance.
- Integrated real-time monitoring with Prometheus metrics and Grafana dashboards, enabling instant visualization of API traffic patterns, error rates, and system health, improving operational observability and resilience.

Technical Skills

Languages: Java, Python, C++, JavaScript (Node.js, React), HTML5, SCSS
Databases: Microsoft SQL Server, PostgreSQL, MongoDB, Redis
Cloud & Infrastructure: Amazon Web Services (EC2, S3, Lambda, RDS, API Gateway), Docker, CI/CD, Prometheus, Grafana
Frameworks & APIs: Spring Boot, Hibernate, RESTful APIs, Express.js, Microservices Architecture
Tools & Platforms: Apache Kafka, Postman, Git, Jira, Bitbucket, SoapUI, SDLC
Certifications: AWS Certified Cloud Practitioner
Coursework: Data Structures and Algorithms, Distributed Systems, Database Systems, Object-Oriented Design, Networking and Operating Systems, Security and Privacy in Distributed Systems