

SRIKANTH BATTULA

BACKEND ENGINEER — PAYMENTS & TRANSACTION SYSTEMS | Python • AWS • Kafka • PostgreSQL

Rockville, MD | +1 (252) 284-5365 | [Email](#) | [Website](#) | [LinkedIn](#) | [GitHub](#)

SUMMARY

Payments backend engineer with 2 years building payment infrastructure at redBus, India's largest bus ticketing platform: reconciliation across 7 payment gateways, dynamic gateway routing, and a UPI collect-to-intent migration lifting success from ~55% to ~90%. Now building a card issuer-processor in public with a database-enforced double-entry ledger.

SKILLS

Languages: Python, SQL, C

Backend & Payments: FastAPI, SQLAlchemy, Pydantic, PostgreSQL, REST APIs, payment gateway integration (UPI, Google Pay, PhonePe), reconciliation, idempotency, BIN routing

Transaction Data & Event Streaming: Apache Kafka, Debezium (CDC), Apache Flink (deduplication, late-event handling), PostgreSQL, Delta Lake

Infra & Tools: AWS, Docker, Git, pytest, GitHub Actions (CI/CD), Grafana, Prometheus

Certifications: AWS Certified Data Engineer – Associate (2026)

EXPERIENCE

Software Engineer — redBus, Bengaluru, India

Feb 2022 – Jul 2023

- Built payment reconciliation in Python across 7 payment and refund gateways, matching internal transaction records against daily settlements — flagged hundreds of mismatches monthly and recovered payment shortfalls worth thousands of dollars every month.
- Migrated UPI checkout from collect to intent flows across all 4 client surfaces (Android, iOS, mobile web, desktop), lifting UPI success from ~55% to ~90% on the highest-volume payment method; integrated Google Pay and PhonePe.
- Built dynamic gateway switching that re-evaluated live success rates every 5 minutes and routed each transaction through the best channel (UPI apps, BHIM, direct bank rails, card networks), holding success steady through gateway outages and peak traffic; shipped a payment-failure redressal system, BIN-based card routing, and the redBus wallet (300K+ users).
- Load-tested payment and booking APIs ahead of Diwali and Sankranti peaks (up to 10x normal traffic), flagging slow endpoints and bottlenecks for remediation.

Software Engineering Intern — redBus, Bengaluru, India

Aug 2021 – Jan 2022

- Debugged 10–15 double-debit and double-booking escalations per week, tracing them to missing idempotency at the order layer — the fix eliminated that class of incident; supported production testing ahead of releases.
- Wrote technical documentation mapping platform architecture and data flows; converted to full-time.

PROJECTS

Card Issuer-Processor — Authorization, Transaction Processing & Ledger *Open source • 2025 – Present*

Python, FastAPI, PostgreSQL, Kafka, Docker, pytest, GitHub Actions • github.com/srikanthbattula9/card-issuer-processor

- Building a card issuer-processor in public: double-entry ledger with a DB-enforced balance invariant, tokenized card vault outside operational tables, and card lifecycle incl. one-time virtual cards. Schema, ledger tests, and CI live today.
- Next milestones, each a tested PR: authorization engine (balance, status, MCC, velocity, decline codes), idempotent auth → capture → settle → refund, simulated network, Kafka events, reconciliation.

Multi-Rail Settlement Reconciliation Engine *Open source • in progress • 2025 – Present*

Python, FastAPI, PostgreSQL, pytest, GitHub Actions • github.com/srikanthbattula9/settlement-recon

- Ingests daily settlement files from simulated gateways, matches them against a double-entry ledger, and classifies mismatches (missing, duplicate, amount, fee, timing) with a tested rule per class; exceptions API and Grafana next.

Real-Time Transaction Event Pipeline (CDC) *MS capstone • rebuilt in public • Dec 2024 – Present*

Python, PostgreSQL, Debezium, Kafka, Flink, Delta Lake, Docker

- Near-real-time sync of Postgres transaction changes to a downstream transaction store without polling — ~6s from commit to queryable in Delta Lake.
- Row-level changes captured from the WAL via Debezium + Kafka; Flink handled dedup, late events, and schema evolution.
- Rebuilt in public (github.com/srikanthbattula9/postgres-cdc-lakehouse): 41 pytest tests against real Delta tables (not mocks), ruff lint, and GitHub Actions CI on every push.

EDUCATION

Wilmington University — MS, Information Systems and Technology

Sep 2023 – May 2025

JNTU Hyderabad — B.Tech, Electronics and Communication Engineering

2018 – 2022