

VIPUL LADE

Forward Deployed Engineer · Distributed Systems · AI Agent Reliability · Technical Translation

vipulslade@gmail.com | 857-763-6871 | [LinkedIn](#) | [GitHub](#)

SUMMARY

Delivered migration and reconciliation for **90M+ account records at 99.9%+ QA-validated accuracy** across a 700M+ domain-record financial core — and defended that correctness to compliance, audit, and tech-lead reviewers. Backend distributed-systems engineer (~3 YOE) who also builds open-source tooling that lets any team plug in their own LLM agent, build evals from a template, and benchmark it for production reliability. Comfortable scoping ambiguous, correctness-sensitive work end-to-end and translating system behavior for non-engineer stakeholders.

SKILLS

Languages: Java, Python, SQL, TypeScript

Backend & APIs: Spring Boot, FastAPI, REST APIs, PostgreSQL

Distributed Systems: Kafka, event-driven architecture, idempotency, event ordering, retry/DLQ handling, eventual consistency, reconciliation

Data Reliability: migration pipelines, source-of-truth sync, audit trails, change journals, downstream publishing

AI Platform: LLM/generative-model evals, agent tracing, guardrails, structured outputs, eval-driven iteration, benchmarking

Cloud & Observability: AWS, Docker Compose, Dynatrace, ELK, Git, CI/CD

EXPERIENCE

Credit One Bank Las Vegas, NV

Software Engineer, Enterprise Microservices Jul 2023 – Present

- Delivered migration and reconciliation logic for **90M+ account records at 99.9%+ QA-validated accuracy** against legacy source data, on a 3-engineer team modernizing a legacy core account system serving 100M+ accounts.
- Presented and defended migration correctness and reconciliation results to compliance, audit, and tech-lead reviewers — translating event-ordering and consistency guarantees into terms non-engineer stakeholders could sign off on.
- Owned Contact and Authorized User pipelines end-to-end (**700M+ domain records** across phone, address, email, and user-lifecycle data), from ingestion and parsing through business rules, PostgreSQL persistence, and publishing to 4+ downstream services.
- Improved pipeline throughput **60–70%** (Dynatrace-measured) under a 2–3M daily event workload by redesigning entity loading to fetch affected data segments rather than deserializing full account objects.
- Built reconciliation logic that staged unmatched records and linked legacy and modern IDs once prerequisites arrived, eliminating silent linkage failures from out-of-order events and race conditions.
- Resolved a duplicate-journal production incident by excluding audit-field-only JSON diffs from business-change detection, stopping repeated legacy-sync event loops and false-positive downstream updates.

Texas Christian University, Neeley School of Business Dallas-Fort Worth, TX

Python Developer Intern May 2022 – Aug 2022

- Built probabilistic modeling workflows (PyMC3, ArviZ) over large financial datasets and integrated FINRA API data into quarterly projection pipelines, improving dataset operations efficiency 15%.

SELECTED PROJECT

Agent Reliability Workbench — Open-Source (MIT) Agent Benchmarking Platform | [GitHub](#)

- Built an end-to-end full-stack platform where any team can plug in their own LLM agent, build evals from a template, and benchmark it for production reliability against a shared eval suite.
- Ported real distributed-systems failure modes (duplicate-key retries, DLQ poison messages, config drift, rate-limit storms, unsafe remediation) into a 12-case incident-response benchmark for tool-using agents.
- Designed a typed Agent-Under-Test contract and HTTP JSON adapter so any external agent can be registered, tested, and benchmarked against the same suite as the built-in reference agent.
- Built FastAPI/PostgreSQL services for agent runs, trace persistence, eval execution, comparison APIs, and external-agent registration (Pydantic, SQLAlchemy, Alembic).
- Persisted a full trace per run (inputs, outputs, schema/guardrail results, latency, token usage, estimated cost, model/prompt version) so teams can compare agents and make eval-driven model/config decisions — documented with methodology, architecture docs, custom-eval guidance, backend tests, and frontend type checks.

EDUCATION

Texas Christian University — B.S. Computer Science

Fort Worth, TX · May 2023