

Prabhav Nalhe

+1-631-710-5485 | nprabhav111@gmail.com | linkedin.com/in/nprabhav | prabhavnalhe.com | San Jose, CA

SUMMARY

Software engineer (6+ years) building high-throughput distributed systems and reliability infrastructure at hyperscale. At Meta, architected a **dependency-intelligence platform** consolidating 6 fragmented telemetry sources into the company's first canonical dependency graph, addressing **15% of top-severity incidents** across infrastructure serving **multi-trillion daily requests**.

TECHNICAL SKILLS

- Languages:** Python, Rust, C/C++, Java, JavaScript, TypeScript (React/Relay), SQL, Hack/PHP
- Systems:** Distributed systems, High-throughput streaming, Fault injection & chaos engineering, Microservices, Event-driven architecture, API design (Thrift/gRPC, GraphQL, REST, FastAPI), CI/CD, Git, Linux, SLO/SLI & reliability
- Cloud & Data:** AWS (Lambda, EC2, SQS, S3), Docker, Kubernetes, MySQL, PostgreSQL, Redis, Apache Solr, Kafka, Hive/Presto, NoSQL, distributed KV/graph stores, Data lake pipelines, Pandas/NumPy
- AI/ML:** LLMs, Agent harnesses, RAG systems, Multi-agent architectures, MCP, LLM Evals, Prompt caching, Embeddings, Agentic Workflows, Inference routing, PyTorch, scikit-learn

EXPERIENCE

Meta Platforms, Inc.

Menlo Park, CA

Software Engineer | Reliability Infra

Jan 2024 – Present

- Dependency Service:** Built the platform's canonical source-of-truth in **Rust** - stream-fusion pipeline normalizing 6 heterogeneous telemetry sources (RPC traces, service-router logs, firewall flow telemetry resolving IP/DNS to service identity) into one unified graph at **~20 GB/s**, exposed via realtime tailer subscriptions and bulk query APIs that power every downstream system on the platform.
- Real-time detection & new-dependency alerting:** Built live tailers detecting newly-appeared dependencies fleet-wide and the notification system routing them to owning teams - surfacing **475K dependencies** with no prior visibility, behind several major incident classes.
- Risky-dependency notification system:** Designed a full-stack alerting system on an LLM hard/soft classifier with per-oncall subscriptions. **Drove adoption across 1,200+ services** (full highest-criticality-tier coverage); **set the team's H1 2026 AI roadmap** and ran the Dependency Protection Program end-to-end, **onboarding 70+ teams** on an audit framework I designed for removing or softening unwanted dependencies; resolution rate doubled and time-to-resolve dropped 24 days → 8.
- Fault-injection feedback loop (Eris):** Led a 3-engineer, two-team program **protecting ~\$100M in annual revenue exposure** by eliminating risky Tier-0/1 dependencies via **Eris (700+ experiments/day, coverage 38% → 83%, contributed new dependency fault types)**; built the experiment harness that disables one dependency at a time and records SLO/SLI degradation - outputs feed back as **ground-truth labels improving the hard/soft classifier**.
- LLM explanation engine:** Built an agent harness generating failure-impact analysis across **2.2M dependency pairs** (16 purpose-built tools, tiered compute, inline quality gating, \$0.15/analysis) at **near-100% coverage**, cutting investigation time **30 min → 2**. Pattern adopted by 3 adjacent infra teams.

Software Engineering Intern | Core Data

May 2022 – Aug 2022

- Optimized MySQL promotion logic for multi-region clusters, reducing false-positive failover alarms **95%**.

Fidelity Investments

Durham, NC

Software Engineer | Advanced Strategies & Research

Feb 2023 – Dec 2023

- Designed 15M-document indexing and retrieval pipeline (AWS Lambda, SQS, Solr) for financial research.
- Led real-time reporting system on AWS using microservices and event-driven architecture, reducing analyst manual effort **70%**.

Quest Global

Pune, India

Software Engineer

Jun 2018 – Jul 2021

- Built automotive systems serving 10K+ users - real-time voice authentication and OTA update infrastructure for ECUs; optimized compute 40% and improved build speeds 70%.

TECHNICAL DESIGN HIGHLIGHTS

Tiered Processing Architecture | 2.2M dependency pairs - compute matched to criticality

- Three-tier pipeline matching compute to service criticality: deterministic pass (**~10s/pair**) for the well-typed long tail, full agent loop (**~7 min/pair**) for high-criticality services. Daily batch pre-compute + on-demand kept **full-fleet coverage affordable**.

Dependency Graph Data Model | modeling implicit dependencies beyond RPCs

- Most dependency tooling captures direct RPC edges. I extended the graph to model **implicit dependencies** through shared config and data stores; surfacing them required reconciling identity and timing semantics across six heterogeneous telemetry streams into one canonical edge.

Fault-Injection Feedback Loop | Eris-driven closed loop, self-improving classifier

- Closed-loop architecture: Eris chaos experiments produce ground-truth dependency-hardness labels (services that degrade when X fails, with measured SLO/SLI impact); labels feed back into the classifier's training set, eliminating manual labeling. **700+ experiments/day** across cogwheel and service-lab - every injection compounds classifier accuracy.

EDUCATION

Stony Brook University

Stony Brook, NY

Master of Science, Computer Science; GPA: 3.92

Aug 2021 – Dec 2022

Pune Institute of Computer Technology

Pune, India

Bachelor of Engineering, Computer Science

Aug 2014 – May 2018

PUBLICATION

Co-author - "Deep Learning Driven Natural Language Text to SQL Query Conversion" (IEEE TASLP / arXiv)