

Dilip B Sundarraj

Lead Engineer — AI Platform

Minneapolis, MN · [linkedin.com/in/dilip-bandla-sundarraj](https://www.linkedin.com/in/dilip-bandla-sundarraj)

US Patent App. #17558007 — Delivery Management Platform

SUMMARY

I lead the architecture of the GenAI platform every team at Target builds on.

- I own the architecture behind Target's GenAI platform — the multi-provider model gateway spanning Google, OpenAI, and Anthropic, agent frameworks and tracing, evaluation and model monitoring, moderation and AI-safety enforcement, batch throughput, and observability.
- Agent development and evaluation run through both halves of my work: I integrated Google's Agent Development Kit into the platform's agent builder so tenant teams stopped assembling their own frameworks, and advanced the platform's evaluation capability from V0 to release in partnership with Data Science.
- I moved into GenAI in April 2024 after 15 years in backend systems, with no prior experience in the domain — and within nine months had shipped three applications (RAG systems, LangChain and LangGraph agents, a text-to-SQL agent that moved merchandising off spreadsheets) plus my first platform contributions.
- I've spent 17 years in distributed backend systems — Java, Kotlin, Spring Boot, Micronaut, Kafka at Kubernetes scale — and much of that time setting technical direction across a portfolio of services.

CORE SKILLS

AI Platform

GenAI platform engineering · Multi-provider model gateway (Google, OpenAI, Anthropic) · Moderation & AI-safety guardrails · Agent observability & tracing (OpenTelemetry, Opik) · Evaluation & model monitoring · Model Context Protocol (MCP)

LLM Applications

LLM application development · Retrieval-Augmented Generation (RAG) · Vector databases · AI agents (LangChain, LangGraph, Google ADK) · Google Vertex AI · Prompt engineering · Text-to-SQL

Languages

Java · Kotlin · Python · TypeScript/JavaScript · Groovy · SQL

Frameworks

Spring Boot · Spring WebFlux · Spring Cloud · Spring Kafka · Spring Data (JPA/R2DBC/JDBC) · Micronaut · FastAPI · Flask · React · Next.js

Architecture

Microservices · Event-driven systems · REST APIs · Reactive/non-blocking · Resilience patterns (retry, circuit breaker, bulkhead)

Data & Messaging

Apache Kafka · PostgreSQL · Redis · Oracle · MongoDB

Platform & Ops

Kubernetes · Docker · GCP · OpenTelemetry · CI/CD (Vela, Drone) · Grafana · InfluxDB · Git

Testing

JUnit · Kotest · Spock · Cucumber (BDD) · Test-Driven Development · Performance testing

EXPERIENCE

Target · Minneapolis, MN — Oct 2015 – Present

Lead Engineer, AI Platform — Jan 2025 – Present

Technical lead on Target's enterprise GenAI platform — the multi-tenant platform every application team in the company builds and runs AI applications on.

- **Multi-provider model gateway:** led the design of the gateway layer that integrates Google, OpenAI, Anthropic, and future model providers, and wrote the core implementation. Each provider's integration is isolated, so a change to one carries no regression risk to the others; drove the rest of the team onto the same structure and testing patterns.

- **Agent observability and tracing:** stood up the OpenTelemetry collector from scratch as the bridge between agents and trace platforms such as Opik, diagnosing a mismatch between the collector's separate trace, metric, log, and health-check ports and the application platform's routing expectations and introducing an Nginx routing layer to resolve it. Integrated with the agent builder for end-to-end trace publishing — unblocking evaluation for agentic applications and supporting tenant teams writing their own evals.
- **Engineering standards for AI-assisted development:** authored in-repo coding skills that guide both engineers and AI coding agents toward consistent implementations — HTTP client patterns (Spring WebClient, Kotlin coroutines, Flow streaming, shared retry/circuit-breaker/timeout utilities) and controller-level integration testing (Kotest, WireMock, WebTestClient) — cutting pattern drift and review friction as the codebase grew across providers.
- **AI safety and moderation:** re-architected the Moderation API to isolate real-time from batch traffic on independent clusters, eliminating noisy-neighbor contention between tenants and cutting tenant-facing outages. Led the Model Armor integration spike across pre-processing, post-processing, inline, and async enforcement, and set policy and governance strategy with Responsible AI partners.
- **Batch platform — 15+ incidents per week to zero:** re-architected batch processing around asynchronous, event-driven execution with modular processors and config-driven concurrency in Kotlin coroutines. Added event-driven lifecycle tracking, image-to-text processing, and automated archival, and migrated persistence from JPA to JDBI — taking one of the platform's most heavily used systems from chronically unstable to incident-free — on the execution path that runs inference at roughly half the cost of real-time and streaming.
- **Resiliency and data integrity:** added dual persistence (Redis + database) and schema-validated Kafka ingestion to the usage-tracking service, with automatic fallback routing for quota and usage APIs during Redis outages — **600 TPS sustained through peak with zero data loss** and uninterrupted tenant access.
- **Agent platform and evaluation:** integrated Google's Agent Development Kit (ADK) into the platform's agent builder, giving tenant teams a supported path to build agents instead of assembling their own frameworks per team; advanced the platform's evaluation capability from V0 through release with Data Science, including a dedicated OpenTelemetry collector path around cluster stability constraints.
- **Observability foundations:** built a reusable JVM library standardizing HTTP request/response logging and Kafka publishing across platform services, with drop-in Micronaut integration; led the monitoring service POC (Kotlin, Micronaut, Kafka) that set the data model for platform-wide model monitoring.
- **Enterprise enablement and architecture:** exposed batch capabilities through the company's AI Studio — the front door every application team at Target uses to access generative AI — for self-serve multi-tenant use; led the MCP registry POC and groundwork for MCP servers; ran the GCP hosting analysis positioning the platform for multi-cloud; authored ADRs, internal blogs, and the 2025 GenAI peak support playbook that replaced tribal knowledge in incident response; mentored engineers onto the platform.
- **Evaluation Vanguard — 2025 Leadership Award**, recognized by the VP of Data Science, Generative AI Activation, for architectural leadership standing up the app monitoring and evaluation pipeline.

Stack: Kotlin, Java, Micronaut, Spring WebClient, Kafka, Redis, PostgreSQL, JDBI, Kotlin coroutines, Kubernetes, GCP, OpenTelemetry, Nginx, Docker, Google ADK, MCP, Kotest, WireMock

Lead Engineer, Generative AI — Apr 2024 – Jan 2025

Building generative-AI applications and agent workflows for teams across Target.

- **Text-to-SQL database agent — POC to production:** built a conversational agent that turns natural-language questions into SQL, runs it against the enterprise data warehouse, and returns the answer as a chart. Built the chatbot interface from scratch and the backend behind it — API contract, vector-store ingestion strategy, error handling — with retrieval-augmented generation over schema and query context keeping generated SQL grounded. Presented it at Target's Demo Day, which carried the POC into its MVP phase; now productionalized and used by buyers and sellers across Target, who moved off spreadsheets to natural-language data analysis for forecasting and merchandise planning. Advised the team that took it to production.
- **Chatbot-as-a-Service:** built the enterprise chatbot template from scratch — orchestrated the design, partnered with the internal quick-start template team to meet enterprise standards, migrated the front end from React to Next.js, integrated RAG, and automated the CI/CD pipeline for app creation, deployment, and validation. Product teams now launch a chatbot from a basic configuration instead of duplicating engineering effort — **cutting one partner team's time-to-launch from weeks to days**; drove adoption through live onboarding, instructional videos, and quick-start documentation.
- **Trend intelligence application:** built an application that ingests reports, videos, images, and spreadsheets, summarizes them, and answers follow-up questions anchored to the source — including page- and image-level referencing so conversations stay precise. Integrated Google Vertex AI as the model layer.

- **Platform contributions:** proposed the indexing approach the platform's RAG-as-a-Service adopted — ingesting synthetic question, SQL, and metadata examples so retrieval returns contextually similar precedents, improving generated SQL — and built the Kotlin HTTP client the company's AI Studio uses to call the platform.
- Designed the agent layers with Vertex AI, LangChain, and LangGraph — stateful, multi-step tool-using workflows rather than single-shot prompting, applying RAG and few-shot prompting — and built the full stack around them: React, Next.js, and TypeScript front ends over Python services on FastAPI and Flask, containerized with Docker and deployed through CI/CD.
- **Recognized in writing by a senior product manager in merchandising** for leadership on the POC — bringing external consultants up to speed and coaching data analysts and data scientists through the workflows.

Stack: Python, Google Vertex AI, LangChain, LangGraph, FastAPI, Flask, vector databases, React, Next.js, TypeScript, Kotlin, Docker, CI/CD

Lead Engineer, Parcel Platform — Last Mile Transportation — Apr 2022 – Apr 2024

Technical lead for the four-service portfolio powering Target's last-mile parcel network — a network averaging roughly \$200M in expense savings — working across a team of 14 engineers.

- Set and evolved the architecture for four production services — shipment labeling and manifesting, delivery mode optimization, rate engine, and carrier capacity management — while coding and reviewing PRs daily.
- Built the **Rate Engine**, which prices origin-to-destination shipments in real time for guest orders, **sustaining 400 TPS at 50 ms**, and the Delivery Mode Optimization service, which selects the cheapest viable carrier per shipment and feeds the network's expense-savings program.
- Delivered an in-house shipment label and manifest system that generates carrier-compliant labels and transmits manifest data directly to carriers, replacing external dependencies — **supporting package processing peaking at 100 TPS across every distribution center and store-as-DC**.
- Built carrier capacity management to track and enforce daily volume limits across UPS, USPS, and FedEx, preventing tender rejections during peak.
- Standardized resilience across the portfolio — retry, circuit breaker, and timeout policies on every service-to-service call — to contain carrier and downstream outages.
- **Rebuilt the portfolio's alerting from scratch**, introducing structured exception types that moved alert ownership out of external routing rules and into the application layer, where the context lives — separating actionable alerts from noise and cutting on-call fatigue. Paired it with Grafana/InfluxDB dashboards for on-call visibility.
- Established TDD as the team norm and built the end-to-end Cucumber suite that replaced manual endpoint-and-database validation on every release — eliminating hours of manual testing per cycle and serving as living documentation of the portfolio's business flows; owned performance testing for high-traffic services before production release.
- When corrupted rate-card data in a Kafka topic blocked multiple partner teams during peak seasonal testing, coordinated with business partners and drove a full data republish to restore the environment, unblocking every team without interruption to the test window.
- Built a library that automates Kafka consumer management during deployments, and documented the portfolio's APIs with OpenAPI specs so contracts stay current with the code instead of drifting in a wiki.
- Ran the portfolio architecture forum and mentored junior engineers on design, testing, and code quality.

Stack: Kotlin, Java, Spring Boot, Micronaut, Spring Data, PostgreSQL, React, Kubernetes (Target Application Platform), Docker, Vela CI/CD, Grafana, InfluxDB

Lead Engineer, Rate Engine & Driver App — Domestic Transportation — Apr 2019 – Apr 2022

Led backend development for transportation pricing and load-tracking services.

- Designed and built the Rate Engine that returns shipping rates for customer orders across origin-destination pairs — the pricing backbone for downstream transportation decisions.
- Built the Target Driver App backend: REST services and an event stream that capture truck driver load movements from pickup through delivery, giving Target real-time visibility into transportation spend. **This delivery management platform and driver application is the subject of a filed US patent** (see below) — it cut the friction in onboarding external carriers, letting Target scale its carrier network faster.
- Migrated services onto Spring Boot and Micronaut with PostgreSQL, containerized with Docker and deployed through Drone CI.
- Introduced resilience patterns and BDD coverage across the services, and mentored engineers on test-driven development.

Stack: Java, Kotlin, Spring Boot, Micronaut, Spring Data, PostgreSQL, REST, Docker, Drone CI

Lead Engineer, Local Inject — Domestic Transportation — Dec 2016 – Mar 2019

Owned the USPS integration for shipping target.com orders to guests.

- Delivered Local Inject, the USPS last-mile integration for target.com orders — a significant reduction in per-parcel shipping cost versus prior carrier arrangements. Owned the service end to end, including deployment across all environments and 24/7 production support.
- Among the first teams at Target to run production workloads on Kubernetes; built and deployed containerized Spring Boot microservices backed by PostgreSQL.

Stack: Java, Spring Boot, Spring Data, PostgreSQL, REST, Kubernetes, Docker, Drone CI

Senior Engineer, Domestic Transportation — Oct 2015 – Dec 2016

Supported the application portfolio moving goods from distribution centers to stores.

- Maintained and enhanced 11 applications underpinning domestic transportation, and led their migration off IBM WebSphere onto Tomcat running on RedHat OpenStack — cutting licensing cost and unblocking the move to open-source infrastructure. Provided 24/7 on-call support for the portfolio.
- **Built an iBATIS-to-MyBatis conversion tool** that automatically translates legacy XML mappings — a prerequisite for every WebSphere-to-Tomcat migration at Target. Adopted across teams and eliminated hundreds of hours of manual conversion work.

Stack: Java/J2EE, Spring MVC, Spring Batch, Spring JDBC, MyBatis/iBATIS, REST, SOAP, Oracle, DB2, WebSphere, Tomcat, UNIX

Earlier Experience — Infosys & Cognizant — Aug 2009 – Oct 2015

- **Infosys Limited, Technology Lead** (*Aug 2015 – Oct 2015*) — Ameriprise Financial Advisor Portal: built REST services (Jersey) and Spring MVC screens for the portal financial advisors use to guide clients on retirement, loans, and debt. *Java, Oracle, WebSphere.*
- **Cognizant — Delta Air Lines, Delta Connection Pipeline** (*Jun 2013 – Jul 2015*) — Built and supported a Tier 1 real-time flight operations platform processing **120,000 ACARS messages per hour**, serving 5,000 users with 3,000 concurrent at peak: JMS/MDB listeners, SOAP and REST rule-engine services, and Spring MVC components for validation, delay rationalization, and departure/arrival forecasting, with alerting to airports and control centers. *Java, Spring, iBATIS, Hibernate, Struts, JMS, Oracle, WebSphere MQ.*
- **Cognizant — JPMorgan Chase, Client Central** (*Jan 2011 – Jun 2013*) — Delivered enhancements to JPMC's authoritative wholesale party reference data system with **zero IST, UAT, and production defects**; led the decommission of legacy mainframe applications onto the Java platform and built a reusable reporting framework adopted across the program. Awards: Oscar of the Month · Catalyst of the Quarter · Phoenix of the Quarter. *Java/J2EE, Struts, JAX-WS, Oracle, WebSphere MQ.*
- **Cognizant — AstraZeneca, Sapphire** (*Aug 2009 – Dec 2010*) — Built the Java validation layer and owned the C# desktop front end for a clinical adverse-event monitoring system with a rules engine triggering regulatory reporting; authored its JUnit suite. *Java, C#, JavaScript, Oracle.*

PATENT & EDUCATION

US Patent Application #17558007 — Delivery Management Platform and Mobile Application (carrier onboarding + driver app; last-mile and line-haul).

B.E., Electronics & Communication Engineering — Anna University, Chennai, India · 2004–2008