

# VEDANTA DHOBLEY

Solutions Engineer

vedanta.dhobley@gmail.com • 732-609-6676 • [LinkedIn](#) • [GitHub](#)

Engineering scalable data solutions for Global Finance and Risk at Citigroup. Building distributed pipelines that process mortgage, loan, and pricing data for risk, compliance, and wealth management. Personal projects leverage local LLM inference, agentic AI workflows, and modern orchestration tools.



## Work Experience

### Backend Data Engineer

*Citigroup*

**2021 – Present**

*Jersey City, NJ*

#### Orchestration Platform Reengineering

- Architected and built a modern orchestration platform using **Java Spring Boot** and **Swagger**, replacing a legacy monolithic ETL system with a RESTful microservices architecture
- Implemented real-time job queuing with **Apache Kafka**, enabling fine-grained control over pipeline execution order, retry policies, and parent-child job dependencies
- Integrated **Apache Livy** to dynamically submit **Spark** jobs to a cluster, pulling ETL JARs from **S3** and providing structured logging for multi-step orchestration auditing

#### Retail Pipeline Auditing Interface

- Designed a **Python** RESTful API connecting to **Oracle SQL**, **MongoDB**, and **S3**, exposing real-time pipeline metadata to an internal auditing dashboard used by operations teams
- Partnered directly with end users (operations and compliance teams) to gather requirements and iteratively design product features based on ongoing feedback
- Built a **Java** utility for secure **CyberArk** credential retrieval and authored **Bash** deployment scripts for Python virtual environments with Wheel dependencies in production
- Collaborated with front-end developers to ship an intuitive interface for tracking ingestion status, error rates, and data lineage across retail data processing pipelines
- Documented and packaged reusable deployment tooling, reducing onboarding time for new engineers and eliminating repeated manual configuration

#### Enterprise Database Encryption

- Developed a **Java**-based encryption service using **NIST-approved Format Preserving Encryption** (FF1/FF3-1), enabling in-place PII redaction without schema changes
- Encrypted international mortgage and risk datasets containing sensitive customer data, ensuring compliance while minimizing storage overhead and compute costs
- Achieved **25% performance improvement** through multithreaded file staging and parallel processing, with comprehensive test coverage for security validation

#### Automated ETL Workflow

- Built an automated ETL pipeline with **Spring Framework** that executes scheduled SQL queries and pushes results to **HDFS** for distributed processing across cluster nodes
- Implemented scheduling via **CRON** and **Spring Scheduler** with secure database access through **Spring JDBC** and **CyberArk** integration
- Extracted and certified an automated email notification system from legacy code, establishing it as a trusted shared dependency across multiple team projects

## Education & Technical Skills

### BS Applied Sciences in Engineering (2016–2020)

Rutgers University, New Brunswick

### AWS Solutions Architect – Associate (2024)

Amazon Web Services

**Languages:** Python, Java, GoLang, SQL, Bash

**Frameworks:** Flask, FastAPI, Spring Boot, React, Temporal.io

**Data:** MongoDB, Snowflake, Oracle, Kafka, Spark, Hadoop

**Infrastructure:** AWS, Docker, Jenkins, Git, S3, nginx

**ML/Orchestration:** llama.cpp, Dagster, Prefect

## Personal Projects

All projects are open source, fully containerized with [Docker Compose](#), and deployed to production with comprehensive documentation.

### “Found Footy” – Real-Time Football Video Pipeline

**GitHub:** [github.com/VedantaDhobley/found-footy](https://github.com/VedantaDhobley/found-footy)    **Live:** [vedanta.systems/workspace/found-footy](https://vedanta.systems/workspace/found-footy)

Architected a fully autonomous video discovery system that detects goals in real-time across 50 top European football clubs and curates high-quality highlight videos from social media. Handles VAR decisions, deduplicates content across resolutions, and ranks videos by quality.

- Designed a distributed microservices architecture using [Temporal.io](#) for durable workflow orchestration with automatic retries, exponential backoff, and workflow state persistence across restarts
- Built an [auto-scaling service](#) that monitors Temporal task queue depth and MongoDB active goals, dynamically scaling worker and Twitter instances (2–8 replicas) with 30-second check intervals and cooldown periods
- Implemented real-time goal detection via [API-Football](#) with a 3-poll debounce system (~90 seconds) that handles VAR reviews through O(1) set-based event diffing
- Created a [RAG pipeline](#) using [Wikidata SPARQL](#) and local [llama.cpp](#) inference to resolve team aliases (“Barcelona” → “FCB”, “Barca”, “Blaugrana”) for multi-keyword Twitter search
- Implemented [local vision model](#) validation using [llama.cpp](#) to verify discovered videos contain actual football/soccer content before storage, filtering false positives from keyword searches
- Developed [Selenium + Firefox](#) browser automation with persistent cookies for authenticated Twitter/X video discovery, with health-checked load balancing across instances
- Built perceptual hash deduplication with dense 0.25s frame sampling and histogram equalization, matching videos across resolutions using Hamming distance on 64-bit dHash frames
- Deployed 10-container [Docker Compose](#) stack: Temporal server, PostgreSQL, MongoDB, MinIO S3, 4 workers, 2 Twitter instances, scaler service, and noVNC for browser debugging

### “Vedanta Systems” – Personal Portfolio & Project Frontend

**GitHub:** [github.com/VedantaDhobley/vedanta-systems](https://github.com/VedantaDhobley/vedanta-systems)    **Live:** [vedanta.systems](https://vedanta.systems)

A modern frontend application serving as a live dashboard for my personal projects, featuring real-time data streaming from backend services.

- Built a [React + TypeScript](#) frontend with [Vite](#) and [Tailwind CSS](#), integrating with backend [RESTful APIs](#) via an [Express](#) server for real-time data
- Implemented [Server-Sent Events \(SSE\)](#) streaming to display live Found Footy goal updates, with the backend triggering refresh events on new data
- Configured [nginx](#) reverse proxy to route requests on a single port: static files to the React build, API calls to the Express server
- Deployed via [Cloudflare Tunnel](#) for secure, zero-exposed-port hosting with automatic SSL, connecting to backend services over Docker network

### “Legal Tender” – Political Campaign Finance Graph (In Development)

**GitHub:** [github.com/VedantaDhobley/legal-tender](https://github.com/VedantaDhobley/legal-tender)

Building a graph database of U.S. political campaign finance that traces money from source to candidate through any number of PAC transfers. Answers questions like: “How much did tech companies donate transitively to each congressperson, through which PACs, and from which subsidiaries?”

- Designing a [Dagster](#) asset-based pipeline that ingests FEC bulk data (7 file types, ~10GB per cycle) and builds a traversable political money flow graph with full data lineage
- Implementing [ArangoDB](#) graph database with 6 vertex collections and 6 edge collections, enabling variable-depth traversals through PAC transfer chains
- Building “Five Pies” aggregation that traces every dollar backwards through PAC transfers to classify funding by terminal source: Corporations, Trade Associations, Labor Unions, Ideological PACs, and Individuals
- Creating employer resolution pipeline using [Wikidata](#) entity linking and local [llama.cpp](#) inference to normalize employer names and build corporate family hierarchies (subsidiaries → parent companies)
- Processing ~5M individual contributions, ~2M committee transfers, and ~30K committee-candidate affiliations across 2020–2024 election cycles