Real-time Disaster RAG Platform

Technical Overview & Architectural Design

Wiam Lachqer, Ossama Oualy, Othmane BELHAJ

December 25, 2025

1. Introduction

The Real-time Disaster Retrieval-Augmented Generation (RAG) platform is a state-of-the-art system designed to monitor, ingest, and analyze global disaster events in real-time. By combining high-velocity data streaming via Apache Kafka with specialized small-parameter language models (SLMs), the platform provides immediate, context-aware insights into natural disasters such as earthquakes, floods, and wildfires.

2. Streaming Architecture & Ingestion

The platform's "streaming-first" philosophy is powered by a high-availability Apache Kafka cluster, ensuring zero data loss and sub-second processing latency.

2.1. Back-to-Back Kafka Pipelines

The system implements a two-stage message pipeline:

1. **Raw Ingestion**: Specialized producers poll official APIs (USGS, GDACS, NASA) every 180 seconds, pushing raw JSON payloads into the `raw-events` topic.
2. **Stream Processing**: A normalizer service consumes from `raw-events`, applies schema validation, and calculates severity levels. The sanitized output is then produced to the `processed-events` topic.

2.2. Consumer Group Management

To ensure persistent reliability, the **Embedding Builder** operates within a dedicated Kafka consumer group (`embedding-group`). This allows the system to:

- **Auto-Commit Offsets**: Track processing progress, ensuring that if a service restarts, it resumes exactly where it left off.
- **Rebalance Latency**: Distribute event loads across multiple instances in high-traffic scenarios.

3. The Vector Engine & Chronological Intelligence

The retrieval accuracy relies on a unified embedding strategy to bridge the gap between structured JSON and natural language queries.

3.1. Unified Document Synthesis

A crucial innovation of the platform is the **Synthesis Strategy**. Before embedding, every event is converted into a keyword-boosted natural language document. By prioritizing the *Event Type* and *Location* at the start of the string, we maximize the semantic weight of these key entities during the Transformer's attention phase.

3.2. Meta-Sorting & Feeds

Traditional vector databases often retrieve by relevance only, ignoring the temporal decay of disaster data. The platform implements a **Chronological Re-ranker** for the "Latest Arrivals" feed. The API fetches a candidate pool from ChromaDB and performs a stable metadata sort based on ISO-8601 timestamps, ensuring the user always sees the most recent anomalies across all heterogeneous sources (USGS, GDACS, NWS).

4. Hybrid RAG Implementation

The core intelligence of the platform is driven by a multi-stage, dual-path RAG pipeline.

4.1. Fetch Live: Bridging the "Retrieval Gap"

To handle the most immediate events (occurring seconds before the query), the platform implements **Fetch Live** functionality. When triggered, the system initiates a parallel on-demand fetch:

- **Path A (Stored)**: Standard KNN search in the vectorized 300+ event database.
- **Path B (Live)**: Direct REST polling of upstream feeds, followed by sub-second embedding injection.

These paths are merged and de-duplicated at the retrieval boundary, ensuring the LLM has a complete view of both historical and instantaneous context.

4.2. Context Enrichment & Generation

Before generation, the **NewsEnricher** queries the GDELT API. The final context is passed to the **Qwen-0.6B** model via Ollama. The model is constrained by strict prompt engineering to prioritize the official data sections.

5. Self-Healing Pipeline & Orchestration

To ensure 24/7 availability, the platform implements a **Supervisor Pattern** in the main orchestrator:

- **Process Monitoring**: A master watchdog monitors the health of ingestion, processing, and API services every 5 seconds.
- **Auto-Recovery**: Any crashed component is automatically re-instantiated with its previous state, maintaining the Kafka consumer group offset.

6. Conclusion

By integrating Kafka-driven streaming with a high-precision vector engine and specialized generation constraints, this platform transforms heterogeneous disaster data into reliable, actionable intelligence. The autonomous supervisor ensures that even in extreme network conditions, the pulse of global monitoring remains uninterrupted.