BDM Project 1 Report

Adrian Lim Patricio
Nishant Sushmakar
Oluwanifemi Favour Olajuyigbe
Universitat Politècnica de Catalunya

April 2025

Contents

1 Project Overview

Tripify is an AI-powered travel planning platform that instantly creates personalized and detailed itineraries tailored to each traveler's preferences. The platform integrates transportation options—bus, train, and flight—along with local events, helping users find the best routes, plan multi-city trips, calculate the total cost of their journey, and receive personalized recommendations based on their itinerary and expected location. The pipeline answers the question of how to simplify travel planning by reducing the time and effort travelers spend on researching and organizing trips while ensuring a seamless and personalized experience. It automatically generates optimized itineraries that combine various transport options with local activities while providing tailored recommendations.

The platform's values revolves around saving time, optimizing costs, and enhancing convenience. It efficiently compares transportation options across buses, trains, and flights to find the best routes and prices, while offering activity recommendations and real-time event updates based on user preferences.

To achieve this, Tripify integrates multiple data sources. Transport APIs and web scraping ensure up-to-date bus, train, and flight data, while structured event listings provide information on local activities. LLM-generated itinerary suggestions enhance personalization, and user feedback analysis continuously refines recommendations. By combining these elements, the platform ensures real-time recommendations, cost optimization, and personalized itineraries, allowing users to easily plan and book their entire trip in one place.

2 Architecture Design

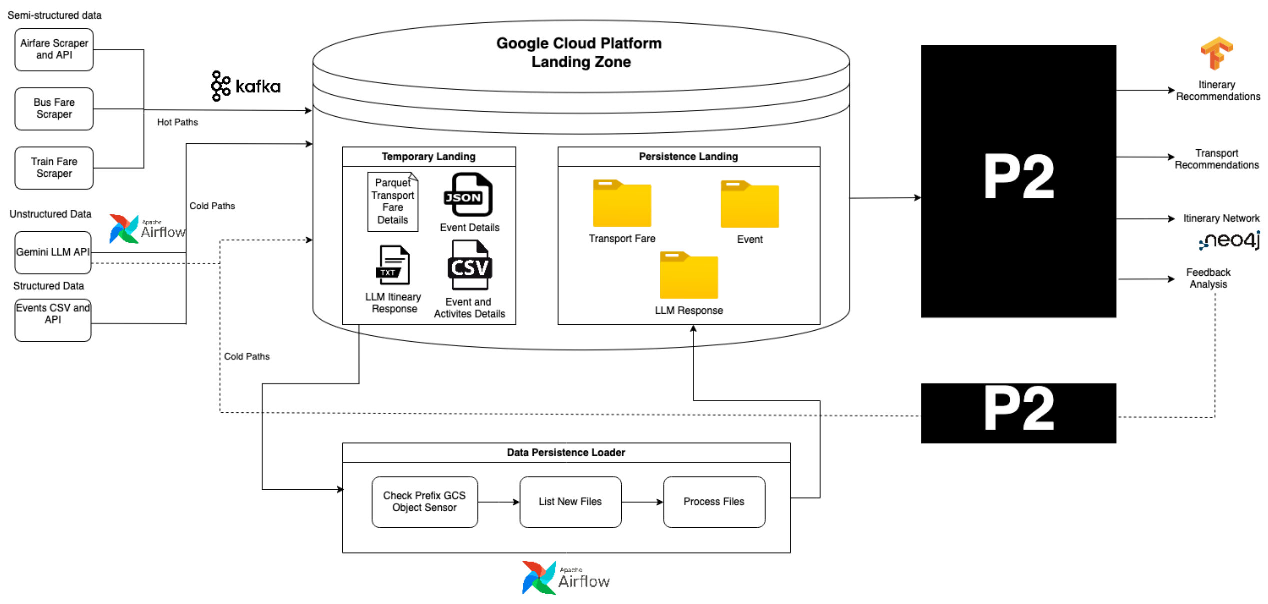


Figure 1: Landing Zone Architecture

3 Data Sources

For the project, we integrate multiple data sources in structured, semi-structured and unstructured formats, with each data source serving a distinct role in the pipeline:

- **Events Data (Structured data)**: Events data includes cultural and entertainment activities in Rome, Paris, Barcelona, and Madrid. It is a structured CSV format automatically downloaded from each city's open database. An API fallback ensures continuous updates if CSV downloads fail.

- **Transport Data (Semi-Structured data)**: Transport data includes flights, trains, and buses connecting the target cities. It is primarily extracted through web scraping google flights, alsa website, renfe website retrieving JSON data. An API acts as a fallback for missing or inaccessible data and ensures continuous updates when failure occurs.

- **LLM Itinerary Data (Unstructured data)**: LLM itinerary data provides personalized travel recommendations in unstructured textual data. It is obtained on demand through API calls, and the data is validated against structured transport and events data to ensure accuracy.

Sample of Events data (CSV)	ID-EVENTO,TITULO,PRECIO,GRATUITO,LARGA-DURACION,DIAS-SEMANA,DIAS-EXCLUIDOS,FECHA,FECHA-FIN,HORA 12732556,Abierto x Obras - Eva Fàbregas,Entrada gratuita,0,1,"M,X,J,V,S,D",,2025-04-10 00:00:00,0,2025-07-20 23:59:00,0
Sample of Transport data (JSON)	{"Departure":"BARCELONA", "Arrival":"MADRID", "Departure Time":"05:50 h", "Arrival Time":"09:15 h", "Duration":"3 horas 25 minutos", "Price":"103,40"}
Sample of LLM Itinerary data (txt)	"location": "Plaça de Catalunya", "time": "morning", "text": "Start your day at Plaça de Catalunya, the heart of Barcelona. Grab a coffee and pastry at a local cafe and take in the bustling atmosphere. This is a major transportation hub, so it's a good place to begin your City Sightseeing bus tour."

Table 1: Sample Data Extracted from Events, Transport, and LLM Itineraries

4 Data Collectors

4.1 Events Data Collector

Event listings are not updated frequently and do not require real-time processing, making them well-suited for batch processing. This approach was chosen because the relatively low volatility of events data makes batch processing both efficient and sufficient for our needs. As a result, a batch ingestion strategy has been implemented using Apache Airflow, which helps us schedule and manage workflows effectively. Airflow is a great choice because it offers a flexible scheduling mechanism, supports complex task dependencies, and ensures reliable execution of the data pipeline.

A scheduled cron job triggers the pipeline on the first day of each week, enabling periodic ingestion and ensuring that event data for each selected city is updated on a weekly basis. The ingested data follows a cold path, as it does not require immediate processing.

4.2 Transport Data Collector

Transport prices are subject to dynamic pricing models and can fluctuate minute by minute based on user demand and engagement. Therefore, maintaining near real-time accuracy of transport fare data is critical to ensure a seamless user experience. Any discrepancy between the displayed price during search and the actual price at the time of booking can result in user dissatisfaction and reduced trust.

To address this, a real-time streaming ingestion strategy has been implemented using Apache Kafka. Kafka was selected for its high throughput, scalability, and fault-tolerant architecture, making it well-suited for continuously ingesting large volumes of rapidly changing data.

Three distinct Kafka topics — Airfare, Trainfare, and Busfare — have been created to handle fare data for different modes of transportation. Dedicated Kafka producers fetch data from three separate web scrapers operating concurrently, enabling continuous collection of live fare updates.

On the consumption side, the Kafka consumer processes data in micro-batches, triggered either upon the accumulation of 1,000 messages or at the end of a rolling five-minute window, whichever occurs first. In the event of scraper failures, fallback APIs are used to ensure continuity of data ingestion, thus preventing system downtime and maintaining the availability of up-to-date transport prices essential for business operations.

4.3 LLM Itinerary Data Collector

Itineraries constitute a core component of our product offering, and Large Language Models (LLMs) play a central role in their generation. We leverage the Gemini LLM API for this purpose. Given that itinerary data is relatively static and does not require frequent updates, a batch ingestion strategy has been adopted using Apache Airflow. A monthly cron job is scheduled to run on the first day of each month, ensuring periodic refresh of itinerary content.

As the platform scales to support additional cities, user preferences and requirements will be gathered through feedback mechanisms. Based on these insights, a separate batch pipeline will be executed to generate tailored itineraries for newly added locations. This ingestion process is designed to be manually triggered as needed, allowing for controlled and scalable expansion.

5 Landing Zone

5.1 Temporal Landing Zone

Google Cloud Platform (GCP) is utilized for storing data in various formats due to its distributed storage architecture, which supports seamless scalability and high availability.

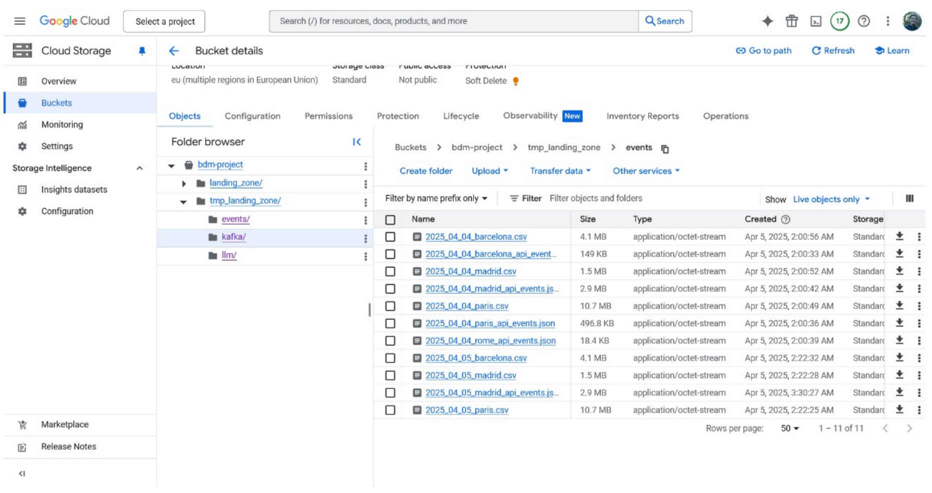
GCP employs an object storage model through Google Cloud Storage (GCS), where data is managed as objects within containers known as buckets. Each bucket is assigned a globally unique name and can be configured with custom storage classes (e.g., Standard, Nearline, Coldline) and geographic locations to optimize for both performance and cost-efficiency.

Additionally, GCP provides lifecycle management policies that automate data governance by transitioning objects between storage classes or deleting them based on predefined criteria such as age or access frequency. This automated policy enforcement enables effective cost optimization and resource management at scale.

The Temporal Landing Zone serves as the initial staging area in the cloud where raw data is ingested and stored by the configured data collectors. This zone supports multiple data formats, including CSV, Parquet, JSON, and plain text, enabling flexible ingestion from heterogeneous sources.

Within this zone, data is organized into three primary directories — events, kafka, and llm — corresponding to event-related data, transport fare data ingested via Kafka, and responses generated by the LLM respectively.

To manage storage efficiently, a GCP Object Lifecycle Management policy is applied to this zone, automatically deleting files after a retention period of seven days. The Temporal Landing Zone acts as the input source for the Persistent Loader, which subsequently transfers and restructures the data into the Persistent Landing Zone for long-term storage and further processing.



Location: eu (multiple regions in European Union) | Storage class: Standard | Public access: Not public | Protection: Soft Delete

bdm-project / landing_zone/ / tmp_landing_zone/ (events/, kafka/, llm/)

Buckets > bdm-project > tmp_landing_zone > events

Name	Size	Type	Created	Storage
2025_04_04_barcelona.csv	4.1 MB	application/octet-stream	Apr 5, 2025, 2:00:56 AM	Standarc
2025_04_04_barcelona_api_event...	149 KB	application/octet-stream	Apr 5, 2025, 2:00:33 AM	Standarc
2025_04_04_madrid.csv	1.5 MB	application/octet-stream	Apr 5, 2025, 2:00:52 AM	Standarc
2025_04_04_madrid_api_events.js...	2.9 MB	application/octet-stream	Apr 5, 2025, 2:00:42 AM	Standarc
2025_04_04_paris.csv	10.7 MB	application/octet-stream	Apr 5, 2025, 2:00:49 AM	Standarc
2025_04_04_paris_api_events.json	496.8 KB	application/octet-stream	Apr 5, 2025, 2:00:36 AM	Standarc
2025_04_04_rome_api_events.json	18.4 KB	application/octet-stream	Apr 5, 2025, 2:00:39 AM	Standarc
2025_04_05_barcelona.csv	4.1 MB	application/octet-stream	Apr 5, 2025, 2:22:32 AM	Standarc
2025_04_05_madrid.csv	1.5 MB	application/octet-stream	Apr 5, 2025, 2:22:28 AM	Standarc
2025_04_05_madrid_api_events.js...	2.9 MB	application/octet-stream	Apr 5, 2025, 3:30:27 AM	Standarc
2025_04_05_paris.csv	10.7 MB	application/octet-stream	Apr 5, 2025, 2:22:25 AM	Standarc

Figure 2: Temproary Landing Zone Directory

5.2 Data Persistent Loader

The Persistent Loader is orchestrated using Apache Airflow and leverages the GCSObjectsWithPrefixExistenceSensor to monitor the Temporal Landing Zone for newly updated files. This sensor specifically tracks incoming data related to events, transport and LLM-generated itineraries, ensuring timely ingestion and processing. Once the presence of new files is detected, the loader initiates the transfer of data to the Persistent Landing Zone for structured storage and further downstream processing.

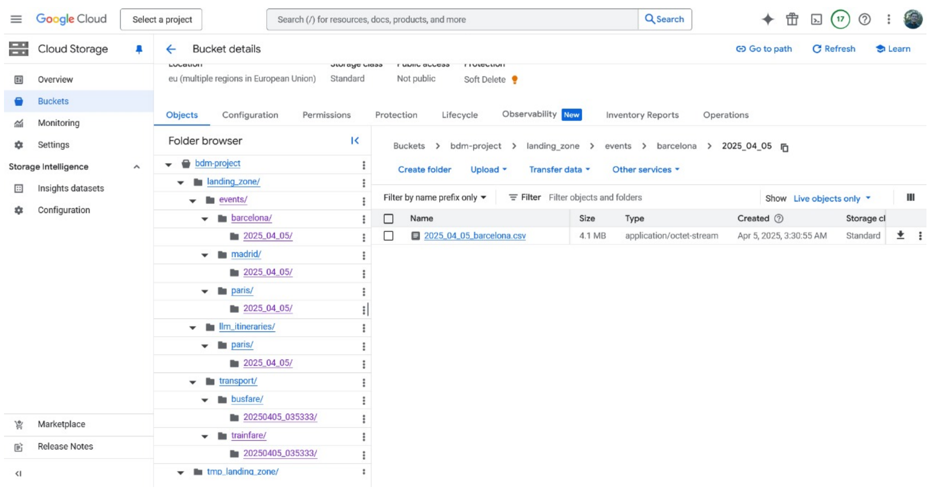
5.3 Persistent Landing Zone

The Persistent Landing Zone serves as the organized storage layer from which various microservices, to be implemented in Phase 2 (P2), will retrieve data. This zone is maintained in the same GCS bucket in a separate folder and follows a structured hierarchy for efficient data organization and retrieval.

The root directory contains three primary folders: events, transport fares, and LLM responses.

- **Events & LLM Responses**: Both follow a similar hierarchical structure. Within the events folder, subdirectories are created for each supported city. Each city folder contains datespecific subfolders, which further organize the data into CSV and JSON formats. The LLM responses folder follows the same structure, as these datasets do not require frequent updates.

- **Transport Fares**: This folder is divided into three subdirectories based on the mode of transport: train, bus, and airfare. Each of these subdirectories contains date-time stamped folders, ensuring that the most recent fare data can be efficiently retrieved. The use of time stamped folder naming is critical for maintaining data freshness and enabling real-time updates in Phase 2 (P2).



eu (multiple regions in European Union) | Standard | Not public | Soft Delete

```
bdm-project
  landing_zone/
    events/
      barcelona/
        2025_04_05/
      madrid/
        2025_04_05/
      paris/
        2025_04_05/
    llm_itineraries/
      paris/
        2025_04_05/
    transport/
      busfare/
        20250405_035333/
      trainfare/
        20250405_035333/
  tmp_landing_zone/
```

Buckets > bdm-project > landing_zone > events > barcelona > 2025_04_05

Name	Size	Type	Created	Storage cl
2025_04_05_barcelona.csv	4.1 MB	application/octet-stream	Apr 5, 2025, 3:30:55 AM	Standard

Figure 3: Landing Zone Directory