Spain Mobility Data Lakehouse: Proof of Concept

Julia Zwittlinger, Ainara Menendez Martin and Jorge Peris Belenguer

November 17, 2025

Abstract

This document describes the implementation of a 3-tier data lakehouse architecture for Spanish mobility analysis. The system ingests data from MITMA (Spanish Ministry of Transport) Open Data portal, processes it through Bronze (raw), Silver (cleaned), and Gold (aggregated) layers, and supports analytical queries for mobility pattern analysis and infrastructure gap identification. The implementation uses DuckDB for processing and follows modern data engineering practices.

Contents

1 Introduction

1.1 Project Overview

The Spain Mobility Data Lakehouse project implements a scalable data architecture to support transport domain experts in analyzing mobility patterns across Spain. The system processes origin-destination matrices, overnight stays, and trip demographic data from MITMA sources.

1.2 Architecture Principles

- **3-Tier Architecture**: Bronze (raw), Silver (cleaned), Gold (aggregated)
- **ACID Compliance**: Using DuckDB for transactional integrity
- **Scalability**: Designed to handle full 2022–2025 datasets
- **Reproducibility**: All transformations are versioned and reproducible

2 Data Ingestion Pipeline

2.1 Source Data Specification

Parameter	**Value**
Data Source	MITMA Open Data Portal
Time Range	2023-01-01 to 2023-01-02
Zoning Level	Municipalities
Data Version	Version 2
Data Types	OD matrices, overnight stays, trip counts

Table 1: Data ingestion specifications

2.2 Ingestion Process

1. Initialize `pyspainmobility.Mobility` client
2. Download origin-destination matrices with activity context
3. Download overnight stays data
4. Download trip counts with demographic breakdown
5. Ingest raw data into Bronze layer tables
6. Apply basic validation and add audit timestamps

2.3 Ingestion Code

Listing 1: Data Ingestion Implementation

```
from pyspainmobility import Mobility

# Initialize mobility client
mobility_data = Mobility(
    version=2,
    zones='municipalities',
    start_date='2023-01-01',
    end_date='2023-01-02',
    output_directory='./data/raw',
    use_dask=False
)

# Download all data types
od_df = mobility_data.get_od_data(keep_activity=True, return_df=
   True)
overnight_df = mobility_data.get_overnight_stays_data(return_df=
   True)
trips_df = mobility_data.get_number_of_trips_data(return_df=True)
```

3 Three-Tier Architecture Schemas

3.1 Bronze Layer: Raw Data Storage

3.1.1 Table: bronze.mitma_od_daily

Column	Type	Nullable	Description
date	DATE	NO	Trip date (YYYY-MM-DD)
hour	INTEGER	NO	Hour of day (0–23)
id_origin	VARCHAR	NO	Origin municipality ID
id_destination	VARCHAR	NO	Destination municipality ID
n_trips	DOUBLE	YES	Number of trips
trips_total_length_km	DOUBLE	YES	Total distance traveled
activity_origin	VARCHAR	YES	Activity at origin
activity_destination	VARCHAR	YES	Activity at destination
loaded_at	TIMESTAMP	NO	Ingestion timestamp

Bronze layer: OD daily schema

3.1.2 Table: bronze.mitma_overnight_stays

Column	Type	Nullable	Description
date	DATE	NO	Stay date
residence_area	VARCHAR	NO	Residence municipality ID
overnight_stay_area	VARCHAR	NO	Overnight stay municipality ID

Column	Type	Nullable	Description
people	DOUBLE	YES	Overnight stay count
loaded_at	TIMESTAMP	NO	Ingestion timestamp

Bronze layer: Overnight stays schema

3.1.3 Table: bronze.mitma_number_trips

Column	Type	Nullable	Description
date	DATE	NO	Trip date
overnight_stay_area	VARCHAR	NO	Municipality ID
age	VARCHAR	NO	Age group
gender	VARCHAR	NO	Gender
number_of_trips	VARCHAR	NO	Trip frequency category
people	DOUBLE	YES	People in category
loaded_at	TIMESTAMP	NO	Ingestion timestamp

Bronze layer: Number of trips schema

3.2 Silver Layer: Cleaned and Integrated Data

3.2.1 Table: silver.integrated_od

Column	Type	Nullable	Description
date	DATE	NO	Trip date
hour	INTEGER	NO	Hour
id_origin	VARCHAR	NO	Origin ID
id_destination	VARCHAR	NO	Destination ID
n_trips	DOUBLE	NO	Cleaned trip count
trips_total_length_km	DOUBLE	NO	Cleaned distance
activity_origin	VARCHAR	NO	Activity at origin
activity_destination	VARCHAR	NO	Activity at destination
time_period	VARCHAR	NO	Time-of-day segment
loaded_at	TIMESTAMP	NO	Processing timestamp

Silver layer: Integrated OD schema

3.2.2 Transformations Applied

Listing 2: Silver Layer Transformations

```
CREATE TABLE silver.integrated_od AS
SELECT
    date,
    hour,
    id_origin,
    id_destination,
    CASE WHEN n_trips < 0 THEN 0 ELSE COALESCE(n_trips,0) END AS
       n_trips,
    CASE WHEN trips_total_length_km < 0 THEN 0
         ELSE COALESCE(trips_total_length_km,0) END AS
            trips_total_length_km,
    COALESCE(activity_origin,’unknown’) AS activity_origin,
    COALESCE(activity_destination,’unknown’) AS
       activity_destination,
    CASE
        WHEN hour BETWEEN 6 AND 9 THEN ’morning_peak’
        WHEN hour BETWEEN 17 AND 20 THEN ’evening_peak’
        ELSE ’off_peak’
    END AS time_period,
    loaded_at
FROM bronze.mitma_od_daily
WHERE date IS NOT NULL;
```

3.3 Gold Layer: Business-Ready Aggregates

3.3.1 Table: gold.typical_day_patterns

Column	Type	Nullable	Description
id_origin	VARCHAR	NO	Origin ID
id_destination	VARCHAR	NO	Destination ID
hour	INTEGER	NO	Hour
time_period	VARCHAR	NO	Time category
avg_trips	DOUBLE	NO	Avg. trips
avg_distance_km	DOUBLE	NO	Avg. distance
observation_count	INTEGER	NO	Count of observations

Gold layer: Typical day patterns

4 Analytical Queries

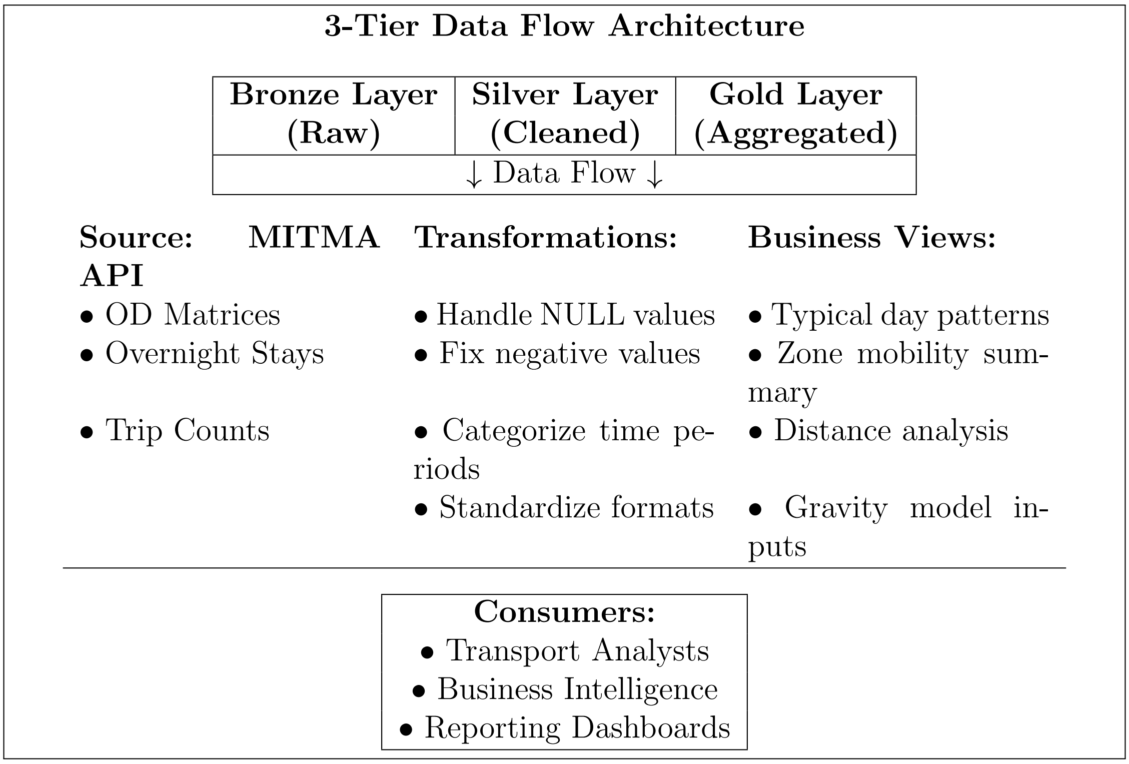
4.1 Hourly Mobility Patterns

Listing 3: Hourly mobility pattern analysis

```
SELECT
    hour,
    SUM(avg_trips) AS total_trips,
    AVG(avg_distance_km) AS avg_distance
FROM gold.typical_day_patterns
GROUP BY hour
ORDER BY hour;
```

5 Data Flow and Processing Architecture

5.1 ETL Pipeline Overview



3-Tier Data Flow Architecture

Bronze Layer (Raw)	**Silver Layer (Cleaned)**	**Gold Layer (Aggregated)**
↓ Data Flow ↓		

Source: MITMA API	**Transformations:**	**Business Views:**
• OD Matrices	• Handle NULL values	• Typical day patterns
• Overnight Stays	• Fix negative values	• Zone mobility summary
• Trip Counts	• Categorize time periods	• Distance analysis
	• Standardize formats	• Gravity model inputs

Consumers:
• Transport Analysts
• Business Intelligence
• Reporting Dashboards

Figure 1: 3-Tier Data Lakehouse Architecture Flow

6 Conclusion

The implemented 3-tier data lakehouse architecture provides a robust foundation for Spanish mobility analysis. It supports:

- Efficient MITMA data ingestion
- Bronze, Silver, and Gold tier processing
- Analytical workloads and statistical modeling
- High scalability and reproducibility