

# Omio Bookings Data Pipeline

Transforming complex booking exports into a scalable, trustworthy Star Schema for analytical reporting—optimized for BigQuery cost and performance.

# The Challenge & Solution

## Challenge

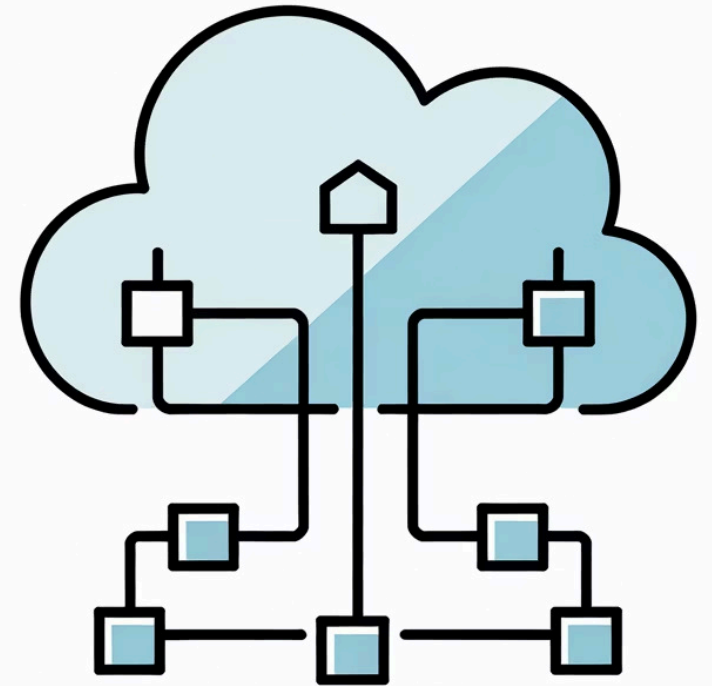
Transform nested, semi-structured daily booking exports (complex JSON) into a reliable Star Schema while optimising BigQuery costs.

## Architecture

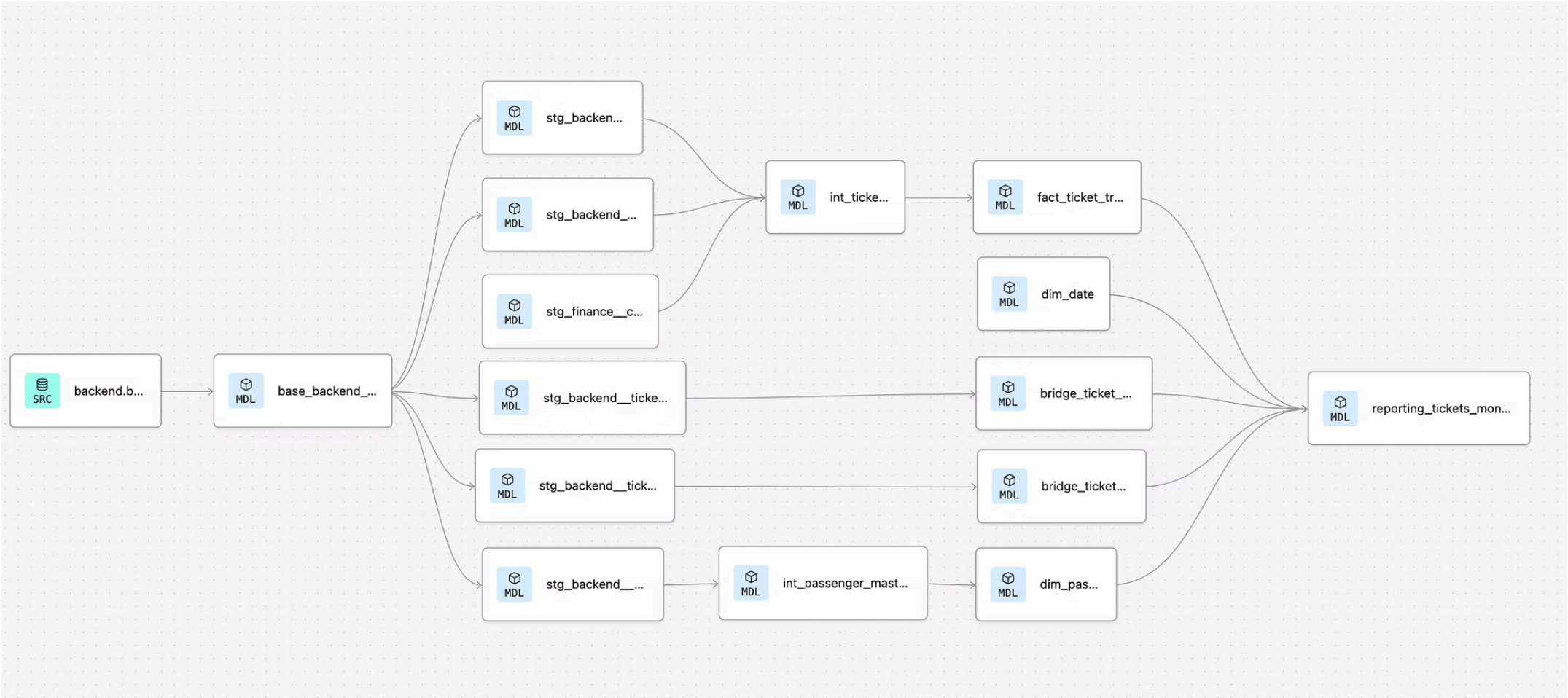
**Stack:** BigQuery + dbt Cloud + Airflow

**Layers:** Raw → Staging → Intermediate → Core Marts → Reporting Marts → BI

**Goal:** End-to-end lineage with reliable KPIs



# Lineage from Raw to Reporting



# Star Schema Design



## Fact Table

**fact\_ticket\_transaction** — one row per ticket. Revenue measured at ticket-level for scalability.

No double-counting of revenue



## Dimensions

**dim\_booking** (transaction header),  
**dim\_passenger**, **dim\_segment** — persistent master dimensions.

Clear separation between transaction and descriptive attributes

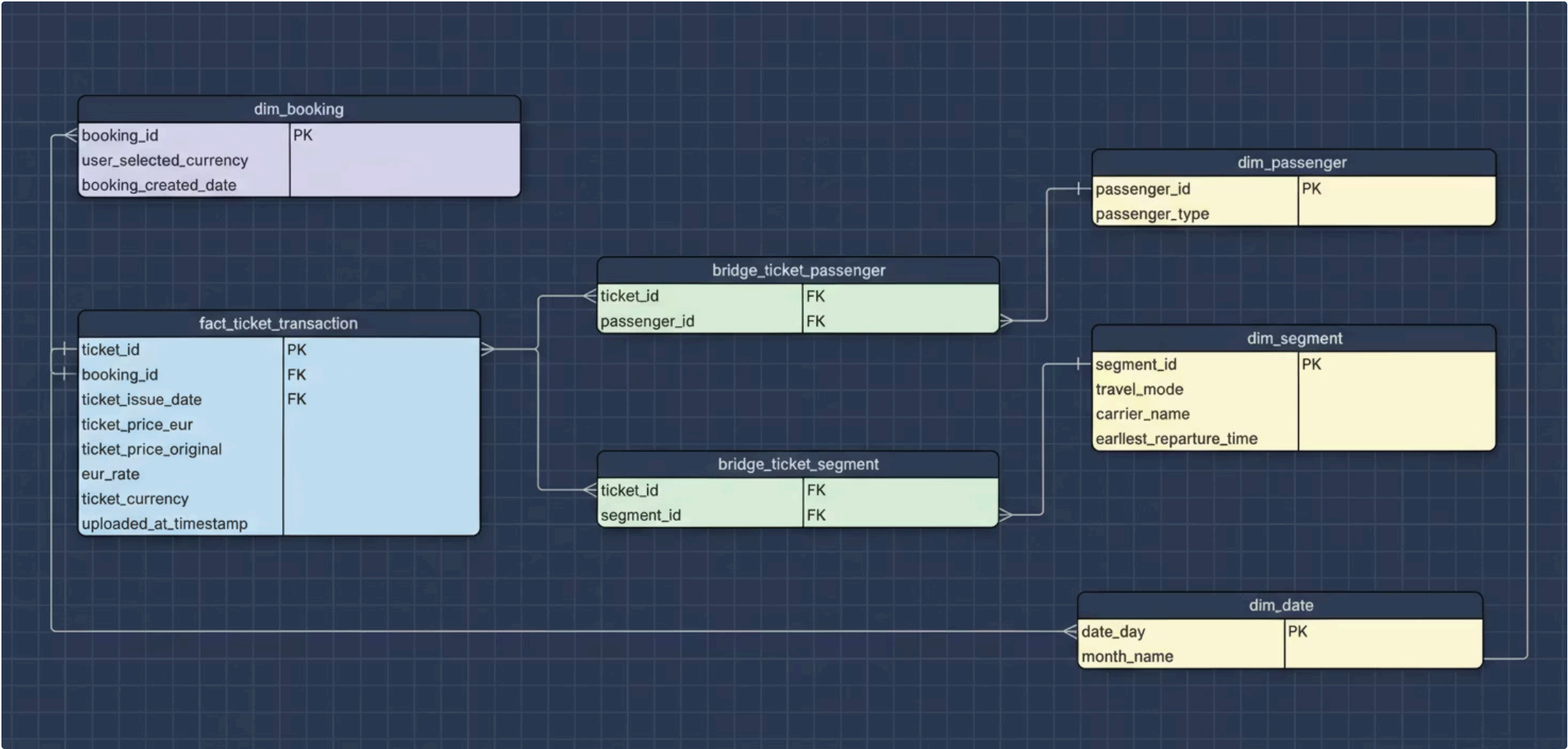


## Bridge Tables

**bridge\_ticket\_passenger**,  
**bridge\_ticket\_segment** — resolve many-to-many relationships cleanly.

Consistent join logic for multi-passenger or multi-segment bookings

# Star Schema of Core Marts



# Compute Strategy & Cost Optimization

Optimize BigQuery costs by limiting heavy transformations and ensuring efficient incremental updates.

Layer	Materialization	Rationale
Base (base_*)	View	Reads the raw semi-structured export directly from storage (e.g., GCS). We don't open or flatten the JSON
Staging (stg_*)	Incremental	Avoids re-parsing large JSON data daily; processes only new records
Intermediate (int_*)	Incremental	Prevents expensive joins from being recomputed for historical data
Marts (fact_, dim_, bridge_)	Incremental	Guarantees low-cost, fast BI reads with stable data
Reporting (reporting_*)	Table	Final, aggregated outputs for dashboard to minimise computation in BI tools

## Incremental Logic

- Uses `incremental_strategy='insert_overwrite'` based on `uploaded_at_timestamp`
- Applies 7-day lookback window for late-arriving records
- Balances cost efficiency with data completeness

# Data Quality & Governance

01

## Ingest Raw Export

Mock Bash task simulating daily ingestion

02

## Run dbt Transformations & Tests

Quality gate ensures all tests pass before proceeding

03

## Deploy to Production

Only if prior step succeeds (trigger\_rule='all\_success')

04

## Notifications

Email alerts sent to data team on failure

### Technical Tests

Structural integrity: PKs, FKs, nulls via dbt generic tests

### Business Logic Tests

Domain rules (e.g.,  $\geq 1$  passenger per booking) in custom SQL

### Anomaly Detection

Z-score + 7-day rolling window for price spikes or volume drops

# KPIs & Business Impact

Our robust data pipeline provides essential insights, driving strategic decisions and fostering stakeholder confidence across the organization.

KPI	Analytical Value
Total Net Revenue (€)	Core financial metric with standardized currency for YOY tracking.
Average Booking Value (ABV)	Indicates efficiency and upsell success per booking.
Revenue per Passenger	Tracks individual customer value and retention impact.
Booking Volume / Tickets	Measures market activity and operational load.
Cancellation Rate	Monitors post-purchase satisfaction and platform reliability.

## Building Stakeholder Trust

Consistent, business-aligned logic enforced in dbt models.	Automated DQ checks with clear test ownership.	Transparent lineage via dbt and Airflow DAG.
--	--	--