ParisMobilityPulse

Real-time Paris Mobility Observatory (Streaming Data Engineering Portfolio Project)

Prepared by Selim Abouleila • GCP / Data Engineering

Executive summary

ParisMobilityPulse ingests live open mobility signals for Paris (e.g., Vélib stations, parking availability, traffic events), processes them in near real time on Google Cloud, and serves a live dashboard + alerting. It is designed to demonstrate end-to-end, production-style streaming engineering (ingestion, processing, warehousing, observability, and IaC).

Problem Statement

Urban mobility data is fragmented across multiple real-time sources (bikes, parking, EV charging, traffic events). While raw data is publicly available, there is no unified, real-time observability layer that allows continuous monitoring, anomaly detection, and operational insight.

Paris Mobility Pulse addresses this by building a production-style streaming data platform that ingests, processes, and serves live mobility signals using Google Cloud.

Objectives

- Certification-aligned: choices map to the Professional Cloud Data Engineer blueprint (streaming, Dataflow, BigQuery, reliability, security, cost control).
- Internship-ready: Terraform, CI/CD, monitoring, alerting, runbooks, and a recruiter-friendly live demo.
- Non-toy scope: real public data, continuous updates, windowed aggregations, anomaly detection, reproducible deployments.

Non-goals and constraints

- **Non-goals**
  - Not a predictive ML system (future demand forecasting out of scope)
  - Not a consumer-grade public app
  - Not guaranteed 24/7 availability
- **Constraints**
  - Poll-based ingestion (no push/WebSocket sources)
  - Public/open data only
  - Student budget (< $70/month)

Data model overview

- Raw events: append-only, source-specific JSON, event-time preserved
- Canonical events: standardized station/event schema
- Aggregates: windowed KPIs (1-min / 5-min / 15-min)
- Latest state: current availability per station (dashboard/map use)

Success Criteria (Definition of Done)

- Continuous ingestion of **at least** one real-time source for ≥24 hours without failure
- Streaming pipeline handles duplicates and late events correctly
- BigQuery tables are partitioned and dashboard-ready
- Looker Studio dashboard updates automatically from streaming data
- Infrastructure is reproducible via Terraform
- Budget alerts prevent uncontrolled cloud spend

Real-time data sources (Phase 1 and Phase 2)

Source	What we use it for
Vélib' Métropole (GBFS)	Station status + station metadata (updated ~every minute).
Paris traffic events (PC LUTECE)	City-reported events affecting traffic (updated every few minutes).
Saemes parking availability	Available spaces feed (refreshed ~1–2 minutes).
Bélib' EV charging availability	Real-time status of charging points.

Target architecture (GCP)

1. Collectors (Cloud Run) poll each feed on a schedule and publish standardized events into Pub/Sub.
2. Streaming processing (Dataflow / Apache Beam): validation, deduplication, windowed aggregations, anomaly detection; errors go to a dead-letter topic.
3. Storage: BigQuery raw (append-only) + curated marts (dashboard-ready); Cloud Storage for raw archive/replay.
4. Serving: Looker Studio dashboard (KPIs + maps) and optional Cloud Run UI for live map + incident replay.
5. Operations: Cloud Logging/Monitoring dashboards, alerting policies, and an incident runbook.

Skill Acquisition

Skill	Demonstrated by
Streaming systems	Pub/Sub + Dataflow (Apache Beam)
Data modeling	Raw vs curated BigQuery datasets
Cloud infra	Terraform + IAM
Cost control	Budgets, limits, region choice
Ops & reliability	Monitoring, DLQ, replay
Communication	Dashboard + documentation

Cost estimate (typical student / portfolio usage)

Dataflow is the main cost driver because a streaming job runs continuously. Estimates below assume deployment in europe-west9 (Paris) and modest data volumes.

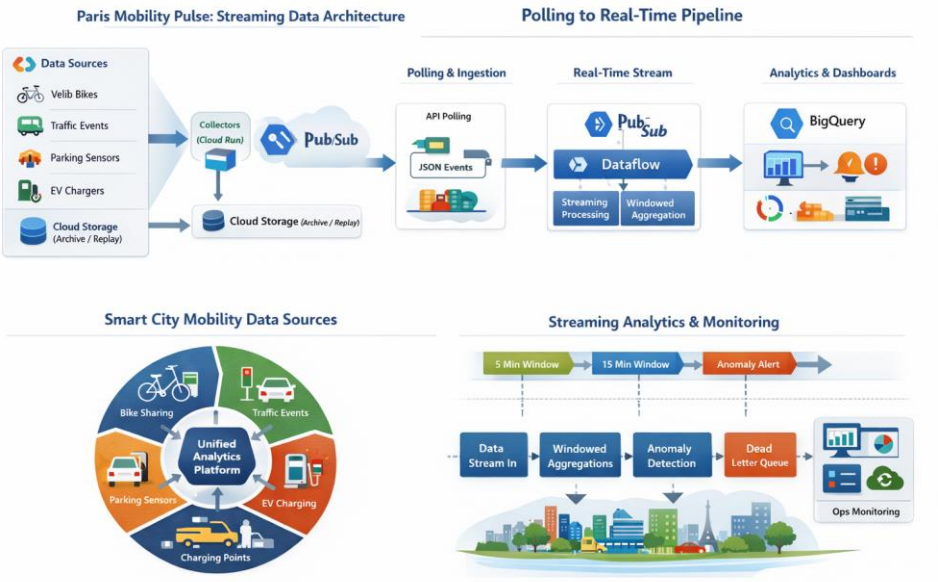
Scenario	Assumption	Est. monthly total	Dominant cost
Demo-only (cheap)	Dataflow ~3 hours/day	$9.07	Dataflow
Always-on MVP	Dataflow 24/7 (1 vCPU)	$65.79	Dataflow
Recruiter-grade	Dataflow 24/7 (2 vCPU)	$140.66	Dataflow

Notes:

- Dataflow is billed by vCPU-hour and GiB-hour; disk and network egress are not included here.
- BigQuery queries are often $0 for a small project because the first 1 TiB of query data processed per month is free.
- Cloud Run collectors typically fit within the Cloud Run free tier for this use case.

Cost control checklist

- Start in Demo-only mode: archive raw JSON to Cloud Storage so you can replay later.
- Keep everything in one region (europe-west9) to avoid cross-region transfer costs.
- Partition/cluster BigQuery tables; set 'maximum bytes billed' on dashboard queries.
- Reduce log volume (avoid DEBUG) and keep retention at default unless you really need longer.
- Set Billing Budgets + alerts from day 1.



References (pricing pages used)

Dataflow: https://cloud.google.com/dataflow/pricing	Pub/Sub: https://cloud.google.com/pubsub/pricing
BigQuery: https://cloud.google.com/bigquery/pricing	Cloud Run: https://cloud.google.com/run/pricing
Cloud Scheduler: https://cloud.google.com/scheduler/pricing	Observability: https://cloud.google.com/stackdriver/pricing
Cloud Storage: https://cloud.google.com/storage/pricing	Artifact Registry: https://cloud.google.com/artifact-registry/pricing