Explained by taking Flight Booking Domin Case Study

**1. Lambda Architecture (Batch + Speed Layers)**

Lambda has **two parallel paths**:

* **Batch layer** for historical data → cleans, enriches, aggregates large volumes.
* **Speed layer** for real-time/streaming → processes new events with low latency.
* Both feed into a **Serving layer** (reporting/analytics).

In your lab:

* **Batch Layer:** ADF → Raw container (CSV/JSON/API batch ingestion) → Databricks → Silver/Gold (aggregates, star schema).
* **Speed Layer:** Event Hubs → Databricks Structured Streaming → Delta (aircraft telemetry).
* **Serving Layer:** Power BI on Gold zone.

So the lab is essentially **Lambda Architecture**: batch ingestion (flight ops, bookings, crew, maintenance) + streaming ingestion (telemetry).

**2. Kappa Architecture (Streaming-Only with Replay)**

Kappa simplifies by **removing the batch layer**:

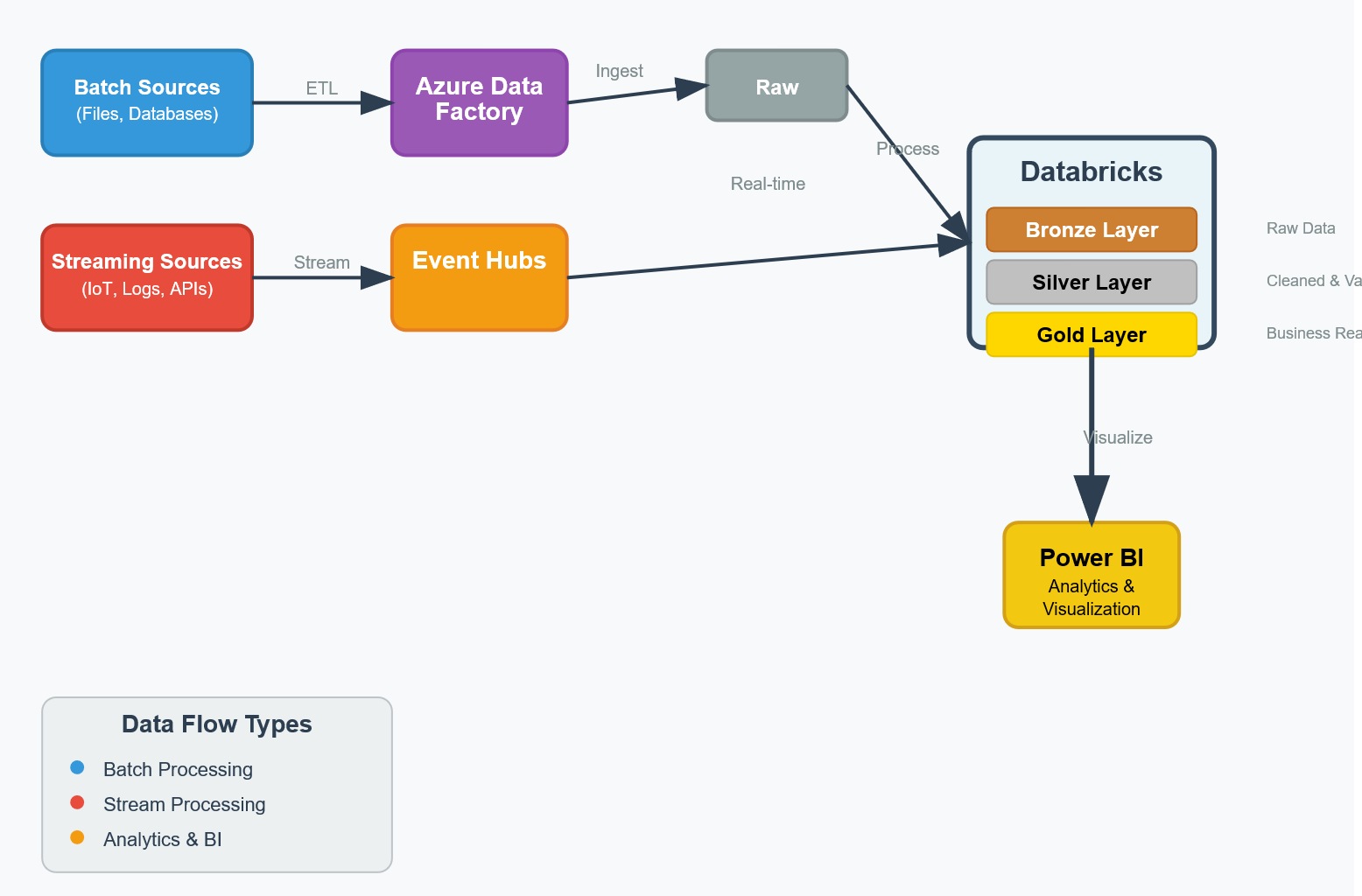
* Everything is ingested as a stream.
* Historical data is reprocessed by replaying the stream through the same pipeline.
* One code path = less complexity.

To run the lab as **Kappa**:

* Treat even batch CSVs as “mini-streams” (e.g., incremental ingestion using Databricks Autoloader instead of ADF Copy).
* Everything flows through **streaming ETL in Databricks** (Autoloader + Structured Streaming → Bronze → Silver → Gold).
* No separate ADF batch pipelines needed.

**Visual Mapping**

**Lambda**



**Kappa**

