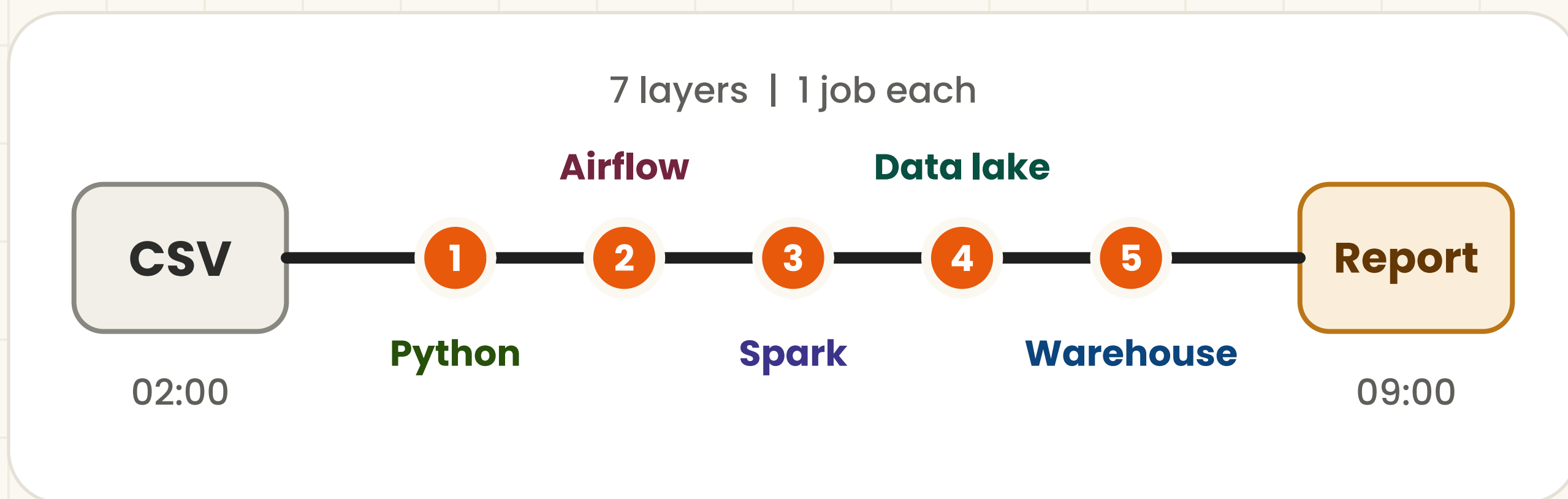


# A CSV file lands at **2 AM**.

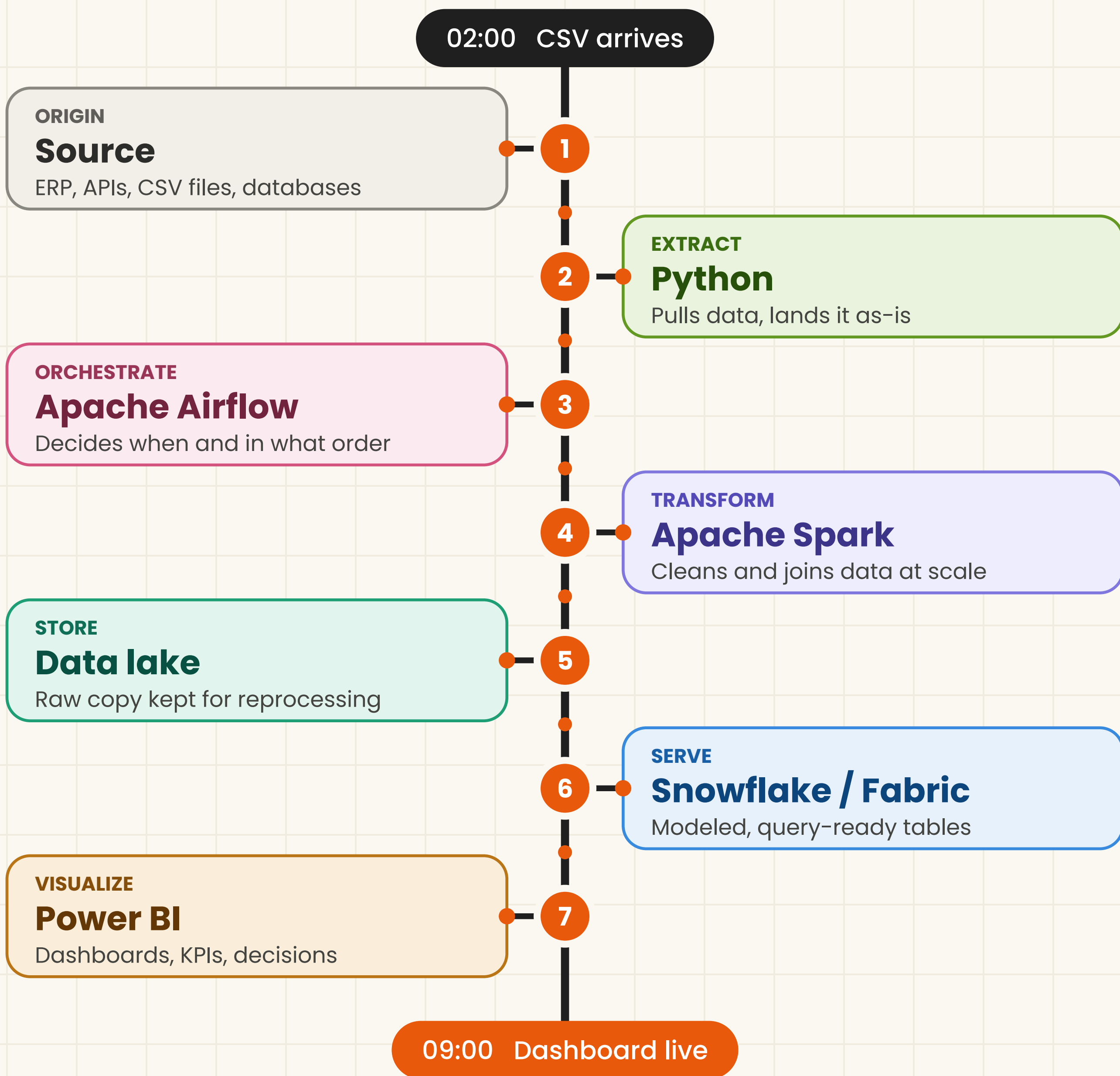
# By 9 AM, it is a Power BI **dashboard**.

Here's what happens in between.



# The complete flow

Seven layers. Each one does a single job.



# Ingest & orchestrate

Layers 1 to 3: getting data in, on time, in order.

1

## Source

ORIGIN

**Holds**

ERP tables, APIs, CSV drops, databases.

**Watch**

You do not control it. Schemas change without warning.

2

## Python

EXTRACT

**Does**

Connects, extracts and lands data raw.

**Rule**

No business logic here. Keep it replayable.

3

## Apache Airflow

ORCHESTRATE

**Does**

Schedules, orders, retries and alerts.

**Does not**

Process data. It tells other tools when to run.

# Transform, store, serve

Layers 4 to 7: from raw rows to business answers.

4

## Apache Spark

TRANSFORM

**Does** Dedups, casts types, joins millions of rows.  
**Why** Heavy transforms live here, not in BI.

5

## Data lake

STORE

**Does** Stores raw, cleaned and curated layers.  
**Why** The raw copy lets you replay any day.

6

## Snowflake / Fabric

SERVE

**Does** Serves modeled, query-ready SQL tables.  
**Why** Fast, governed access for analysts and BI.

7

## Power BI

VISUALIZE

**Does** Turns curated tables into dashboards.  
**Why** Light measures only. Logic stays upstream.

# The #1 mistake

Putting logic in the wrong layer.

## ✗ Common way

- Transform logic inside Airflow tasks
- Business rules buried in Power BI DAX
- No raw copy to reprocess from
- One change breaks three layers



## ✓ Right way

- Airflow only orchestrates
- Spark / SQL layer owns all transforms
- Data lake keeps the raw history
- Power BI only visualizes

**Each tool should do one job well.**

**Which layer causes the most issues in your pipelines?**

Comment below | Week 2: ETL vs ELT



**Ganesh Doifode**

Senior BI Engineer | Power BI | Fabric

**Follow for more**