

Do you really need real-time?

01 Two delivery models

Batch moves data on a schedule. Streaming moves each event as it happens.

BATCH

- Runs nightly, hourly, on cron
- Cheap: compute only while running
- Easy to test and re-run
- Airflow, dbt, plain SQL

STREAMING

- Processes events in seconds
- Always-on infrastructure
- Hard: ordering, late data, replays
- Kafka, Flink, Kinesis

02 The freshness interview

Nobody needs "real-time". They need a freshness number. Get it.

ask "what decision changes if the data is an hour old?"

monthly report daily batch. Anything more is theater.

morning dashboard nightly batch, done before coffee

fraud, alerts, pricing genuine streaming. The real use case.

→ In most companies the honest answer to "how fresh?" is "by tomorrow morning". That is a batch job.

03 What real-time costs

You pay in money, complexity and sleep. Pay only if the decision needs it.

	BATCH	MICRO-BATCH	STREAMING
freshness	hours	minutes	seconds
infra cost	\$	\$\$	\$\$\$\$
failure mode	re-run the job	re-run the window	page someone at 3am
team skills	SQL	SQL + tuning	distributed systems

COMMON TRAPS

04 Trap: streaming a daily report

Kafka feeding a dashboard that one person opens on Monday.

the build 3 months of streaming infra for a weekly decision

the symptom the pipeline costs more than the insight

the fix micro-batch every 5 minutes covers almost everything

→ Start with batch. Move to micro-batch when someone actually waits on data. Stream when seconds change money.