

Apache Flink



Apache Flink



What is stream processing?

How is it different from other
kinds of data processing?

How does Flink do stream
processing?

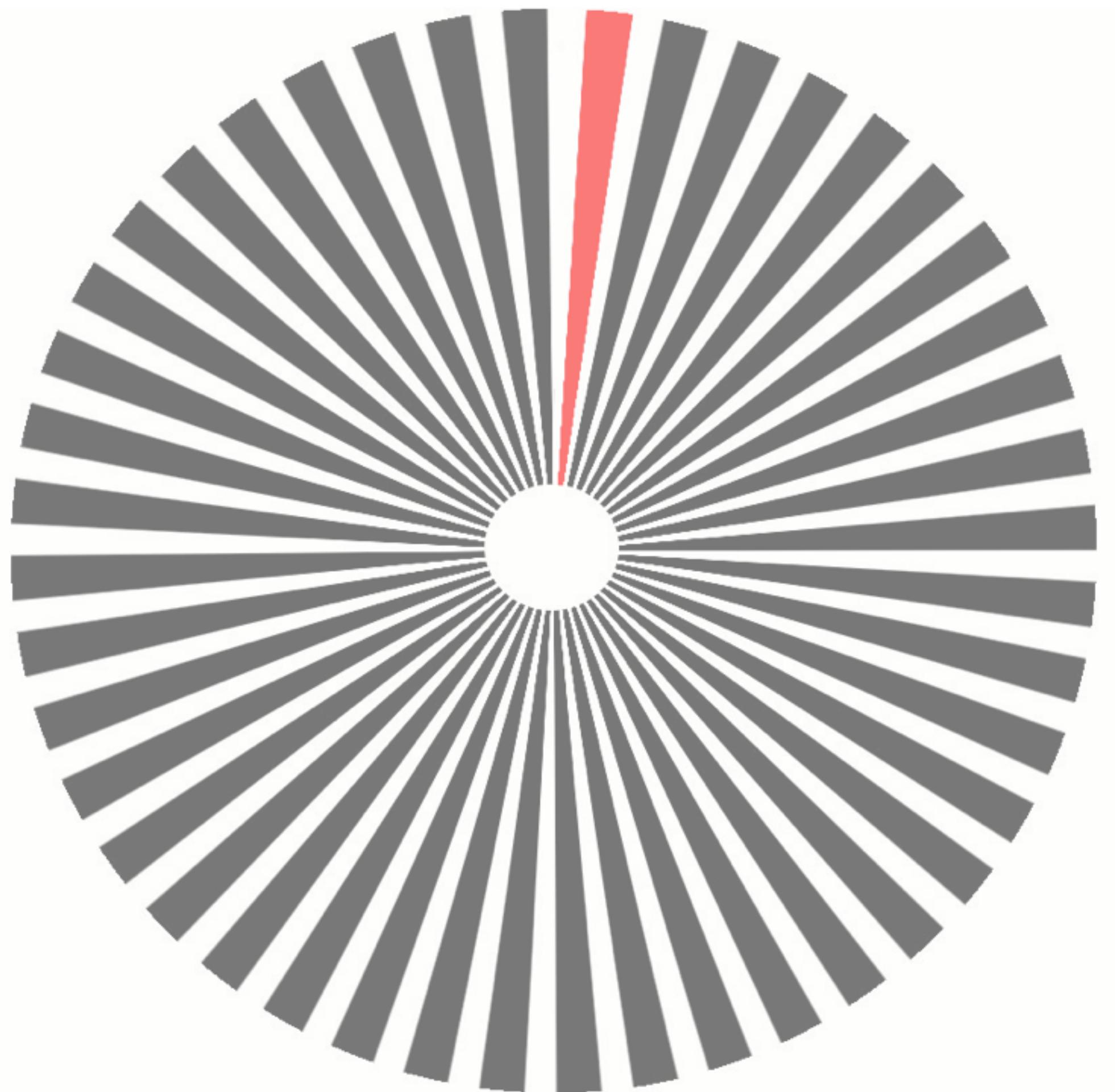
Life IS
10% OF WHAT
HAPPENS TO YOU,
AND 90% OF
HOW YOU REACT
TO IT.

In life and at work

Receive information

Process it

React to it



Receive information

Process it

React to it

FAST!!

FAST!!

Reaction time can
make all the difference



or





When you're driving
on the freeway



Reporters racing to
report the news



Traders making
trades



Companies
reacting to social
media meltdowns

Stream processing

Receive

Process

React

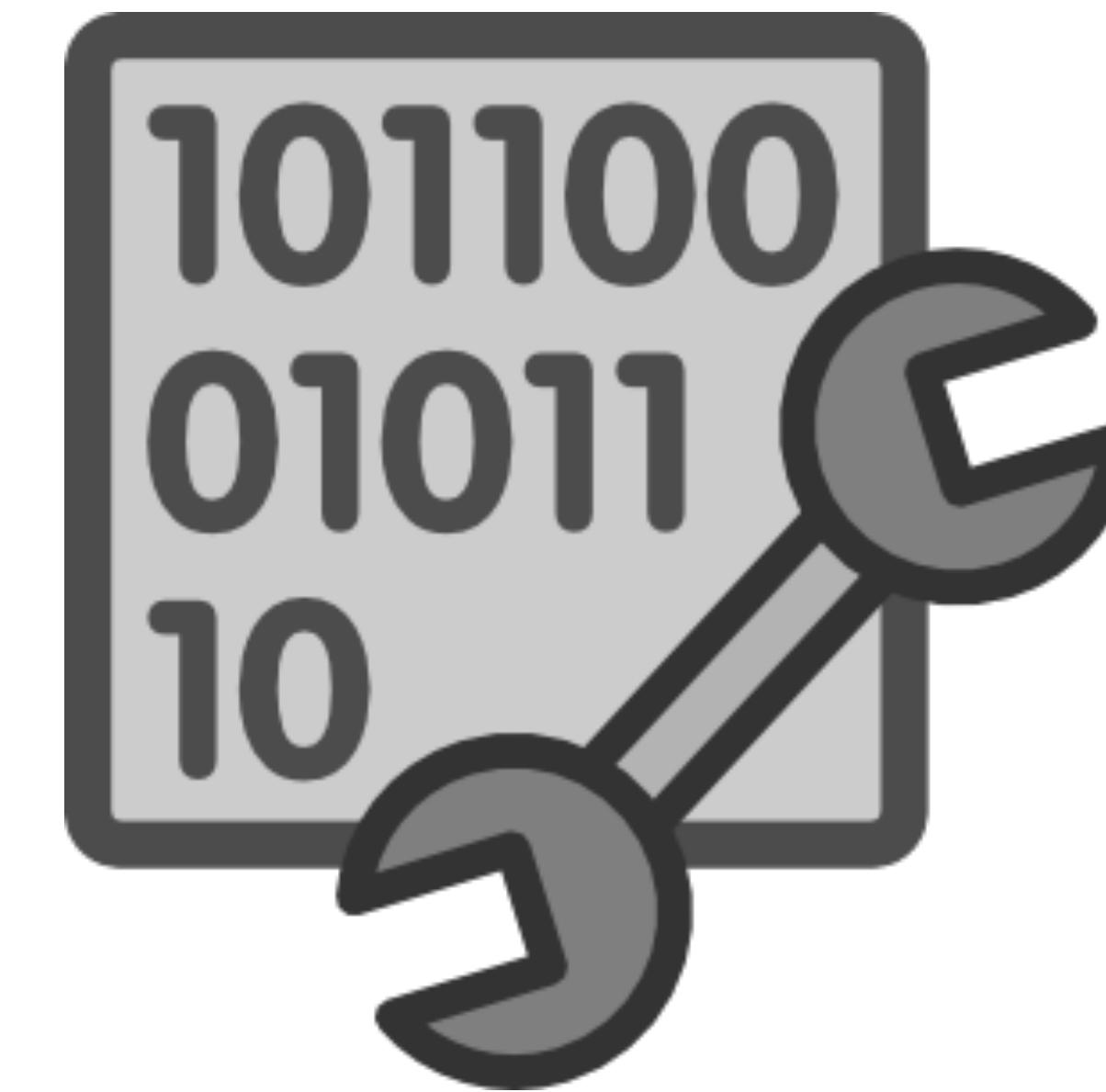
In real-time

Data processing

Store now

Process and
analyze later

Batch processing



Receive,
process and
react in real-
time

Stream processing

Why?

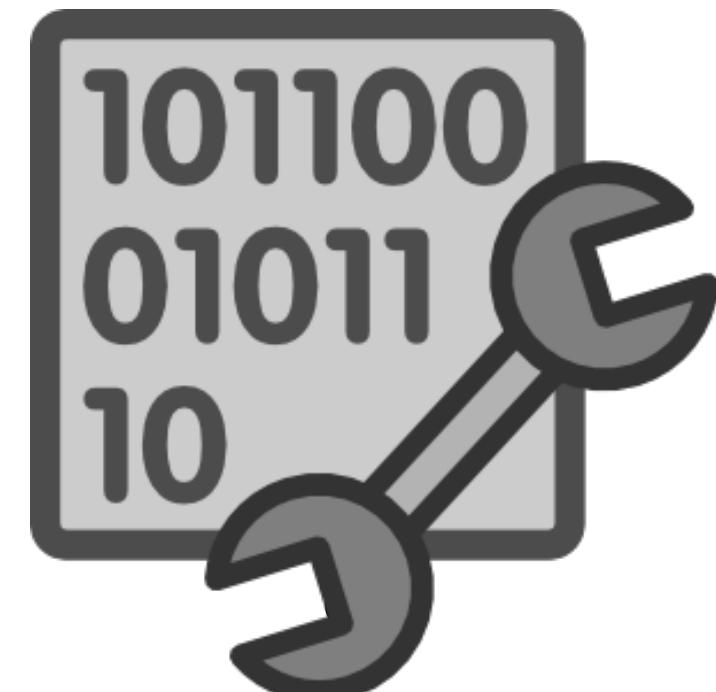
Batch processing

Learn from the past

- which promotions/product features worked?

Take big picture decisions

- when & where do we launch the new smartphone?



Stream processing

Why?

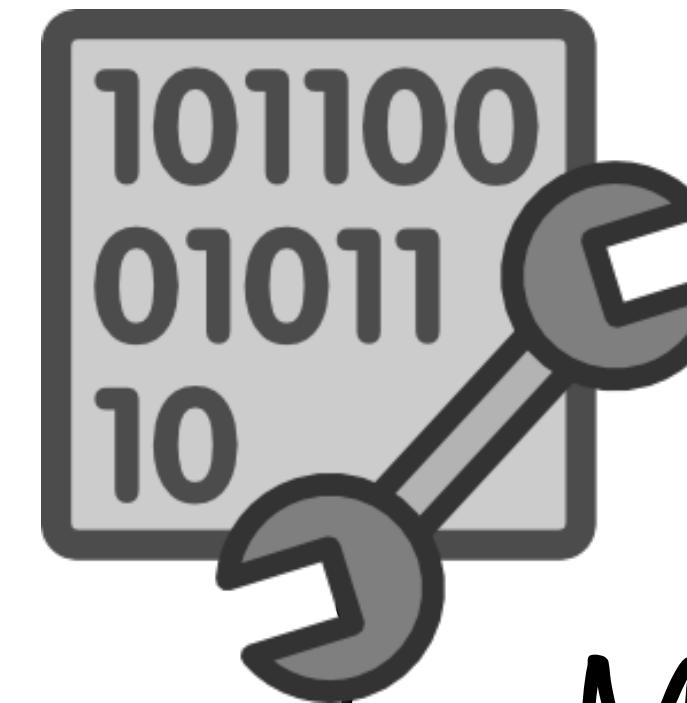
Batch processing

Learn from the past

- which promotions/product features worked?

Take big picture decisions

- when & where do we launch the new smartphone?



Stream processing

Monitor and track events

- stock price movements, payment failures

Take fast, tactical decisions

- place a trade, re-route to different payment gateway

What?

Batch processing

Periodic reports

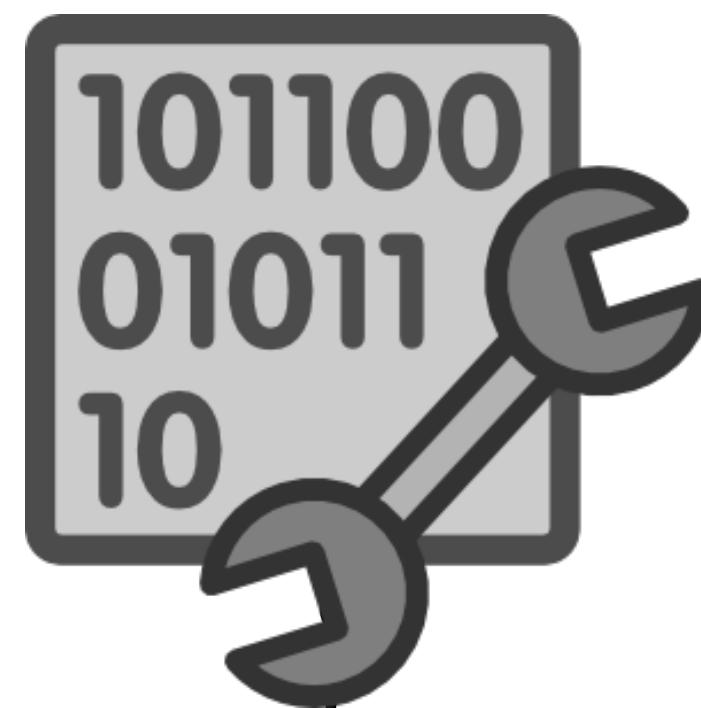
- day, week, month

Analyze trends

- WoW, MoM, QoQ

Compare slices

- cities, categories



Stream processing

What?

Batch processing

Periodic reports

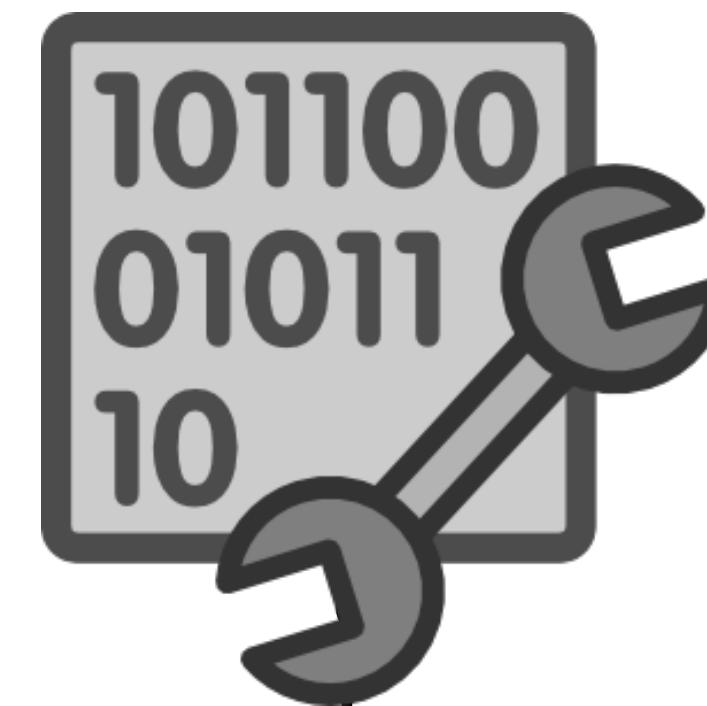
- day, week, month

Analyze trends

- WoW, MoM, QoQ

Compare slices

- cities, categories



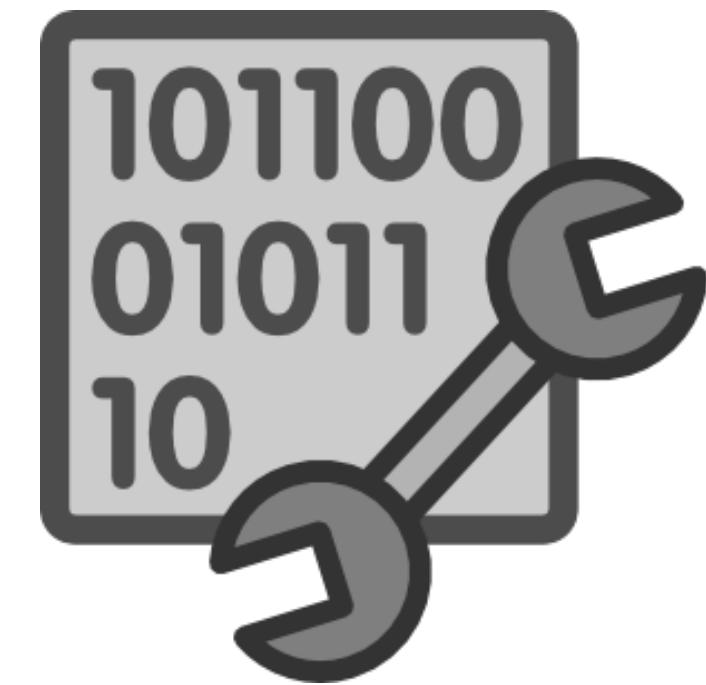
Stream processing

Process individual events

- tweets, stock quotes

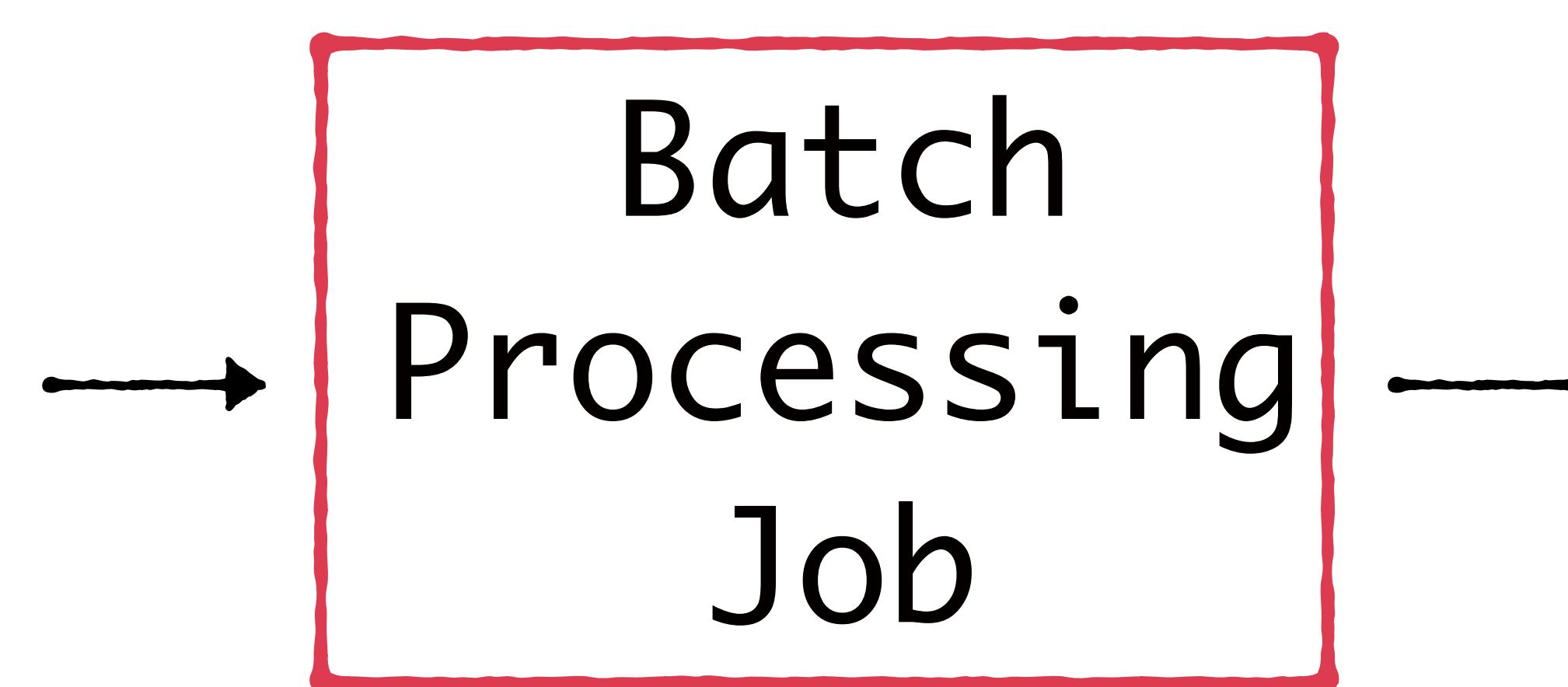
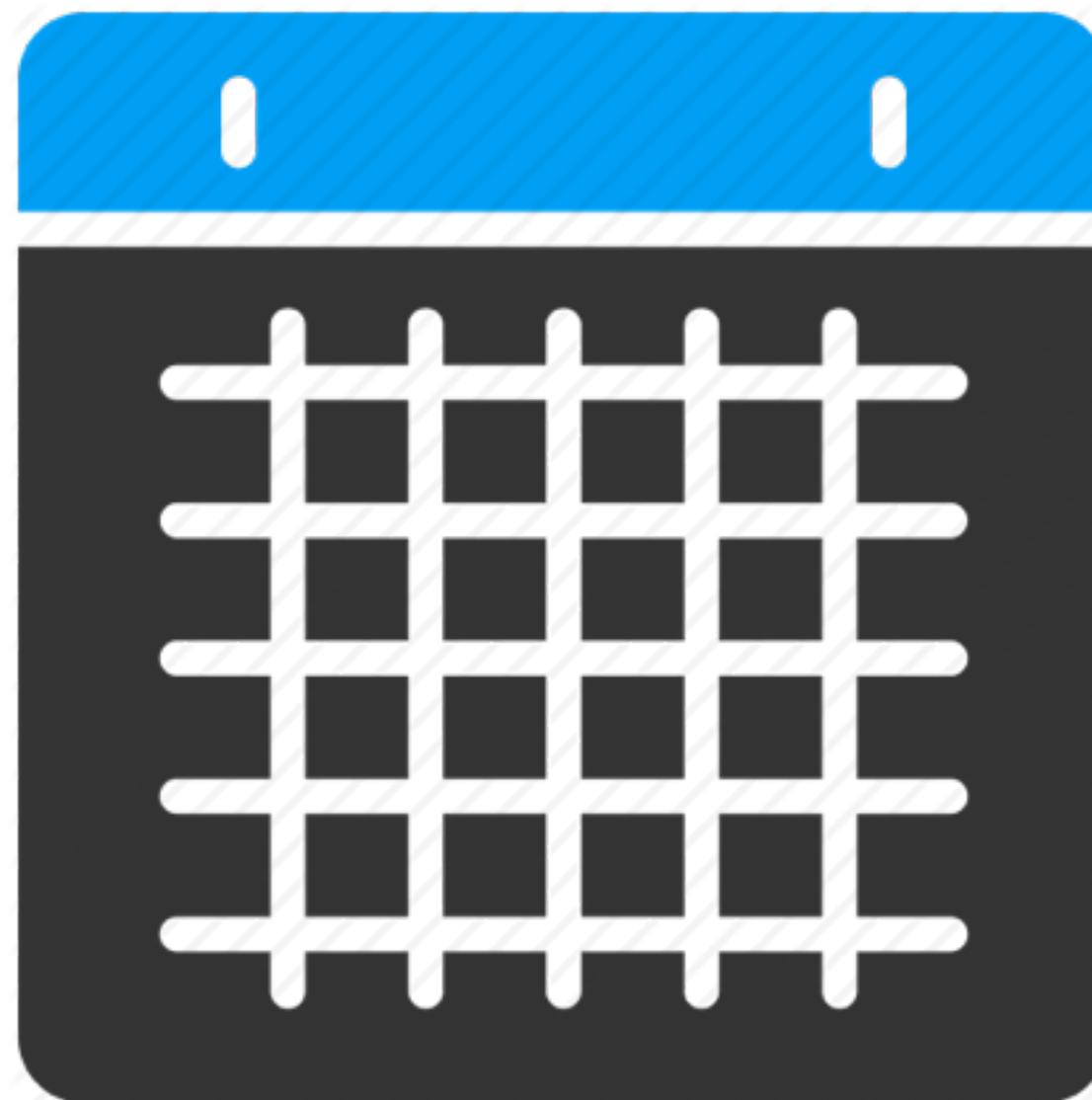
How?

Batch processing



Stream processing

Batch processing



Dataset

Reports, trends,
insights

Batch processing

Batch
Processing
Job

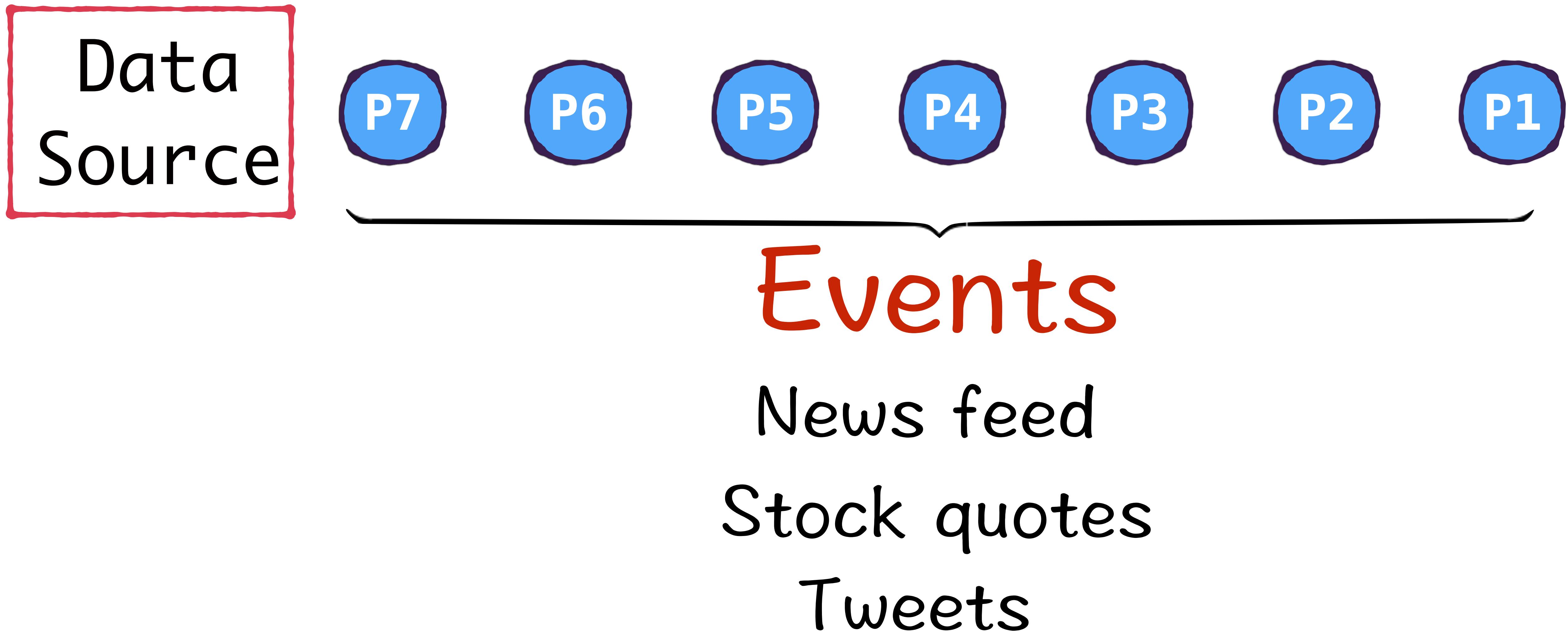
Finite unchanged datasets
(week, month, year)

Bounded Datasets

Runs for a specific
amount of time

Batch jobs

Stream processing



Stream processing



Events are processed as they arrive

Stream processing

Stream
Processing
App

Infinite datasets which are
added to continuously

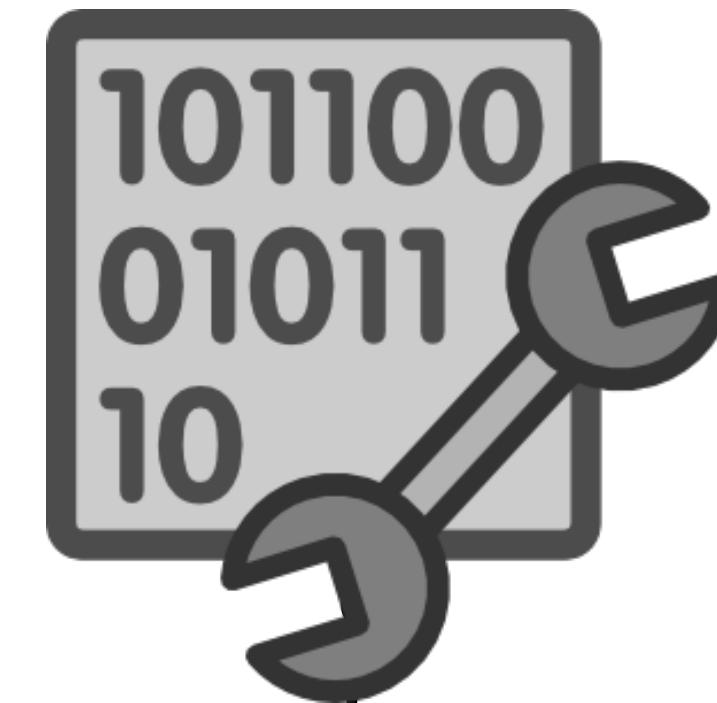
Unbounded Datasets

Runs constantly as long as
data is received

Continuous processing

System

Batch processing



Stream processing

Process bounded datasets

Periodic updates

Order of data received
unimportant

Single global state of the
world at any point in time

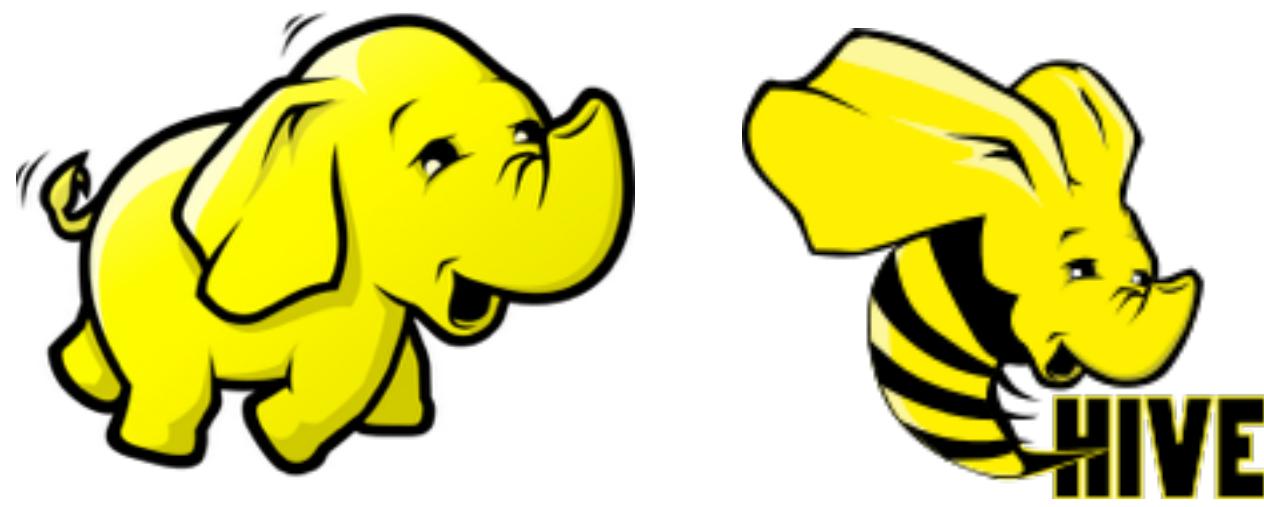
Process unbounded datasets

Continuous real-time updates

Order important, track out of
order arrival

No global state, only history
of events received

Hadoop,
MapReduce, Hive



Batch
processing

Spark, Flink



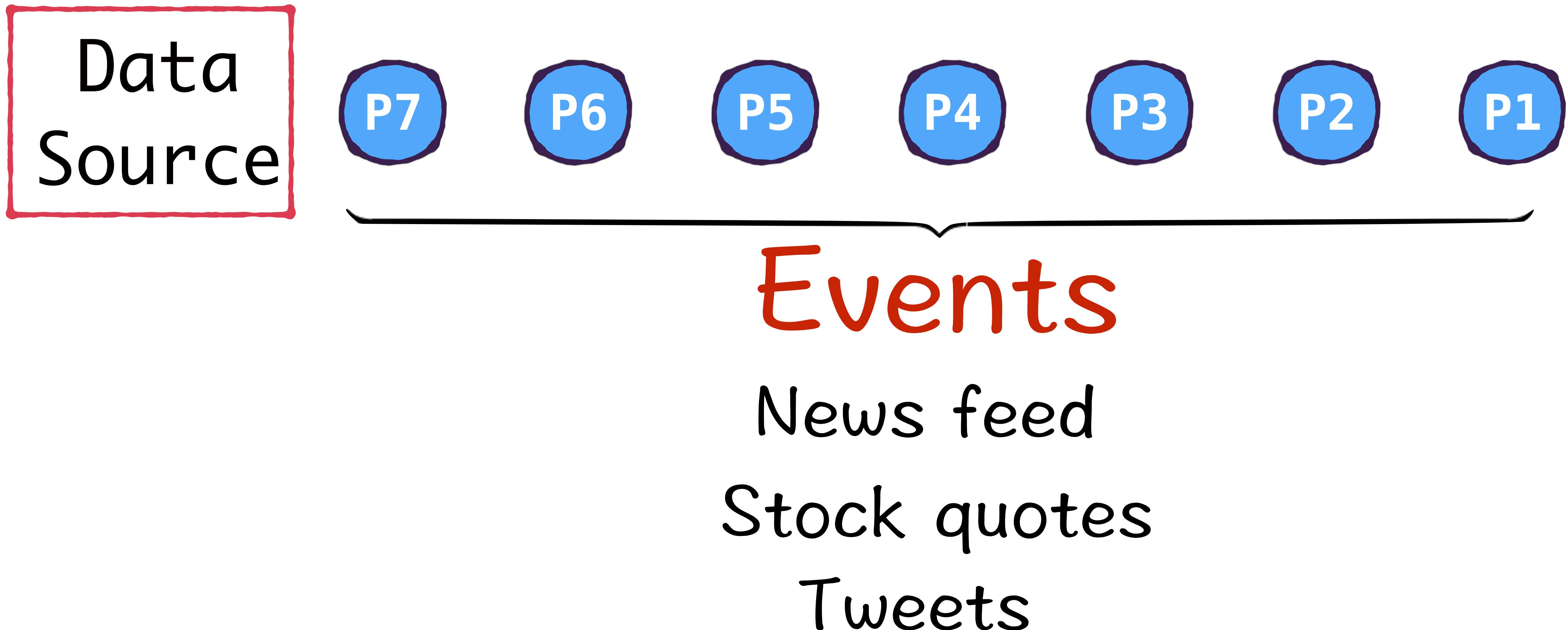
Batch +
Stream
processing

Apache
Storm



Stream
processing

Stream processing



Stream processing



Events are processed as they arrive

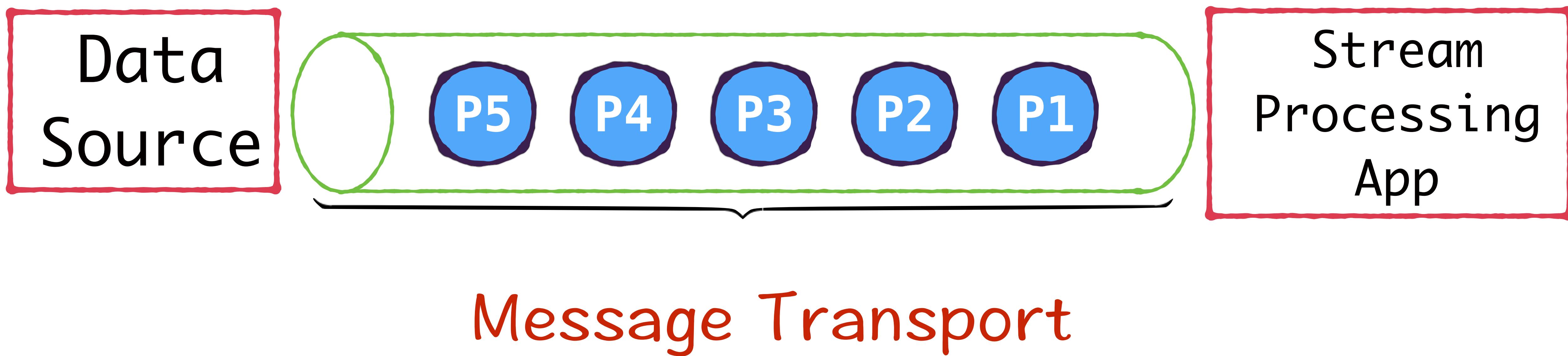
Filter, transform, combine

Stream processing



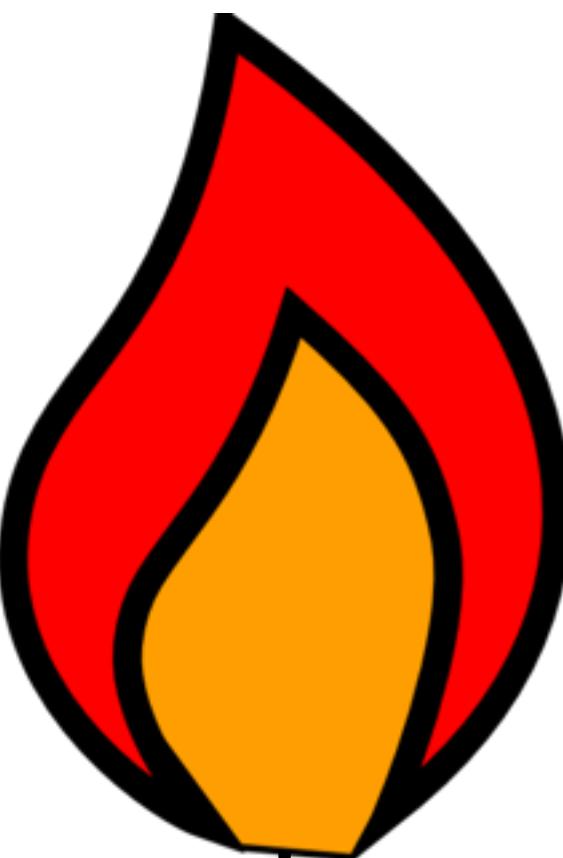
Process events fast with no
loss of information

Stream processing



Source of Truth

Traditional
Architecture



Persistent storage

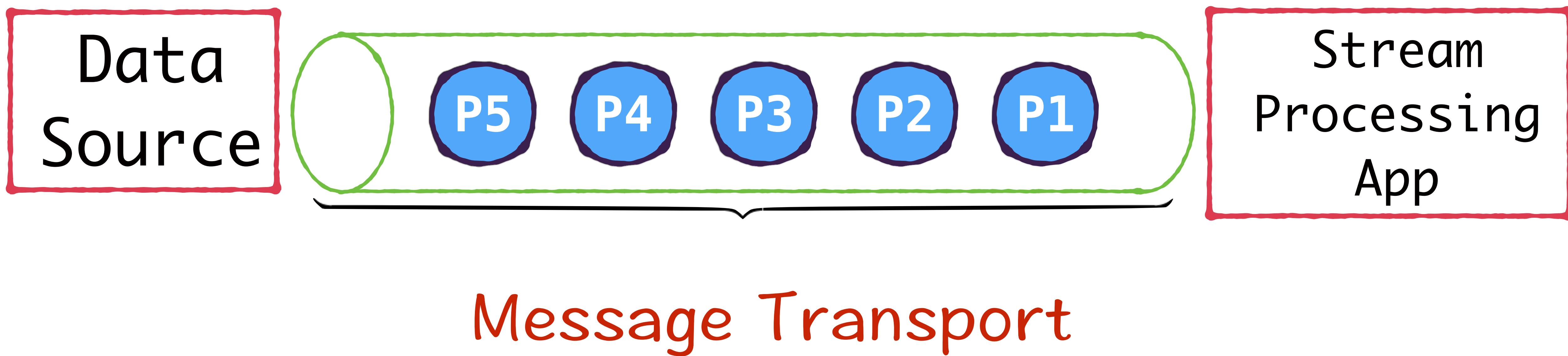
- files, databases

Streaming
Architecture

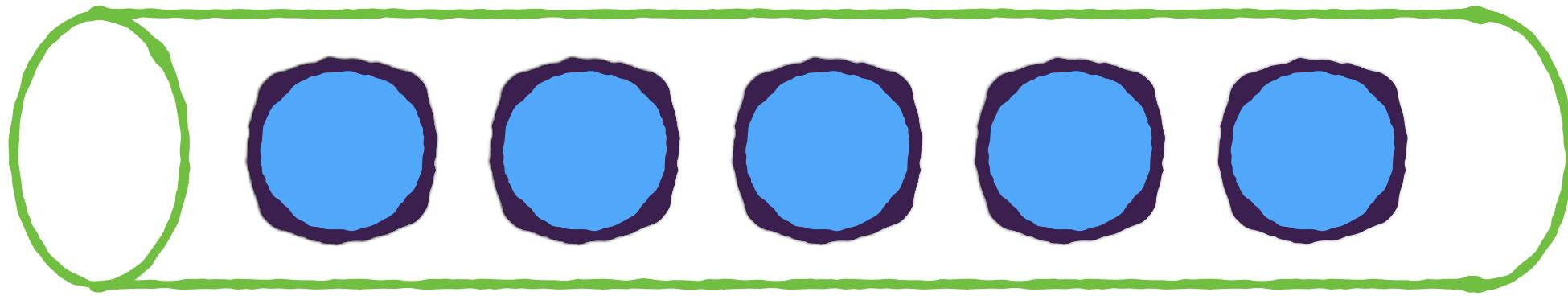
Stream

- message transport pipeline

Stream processing



Message Transport

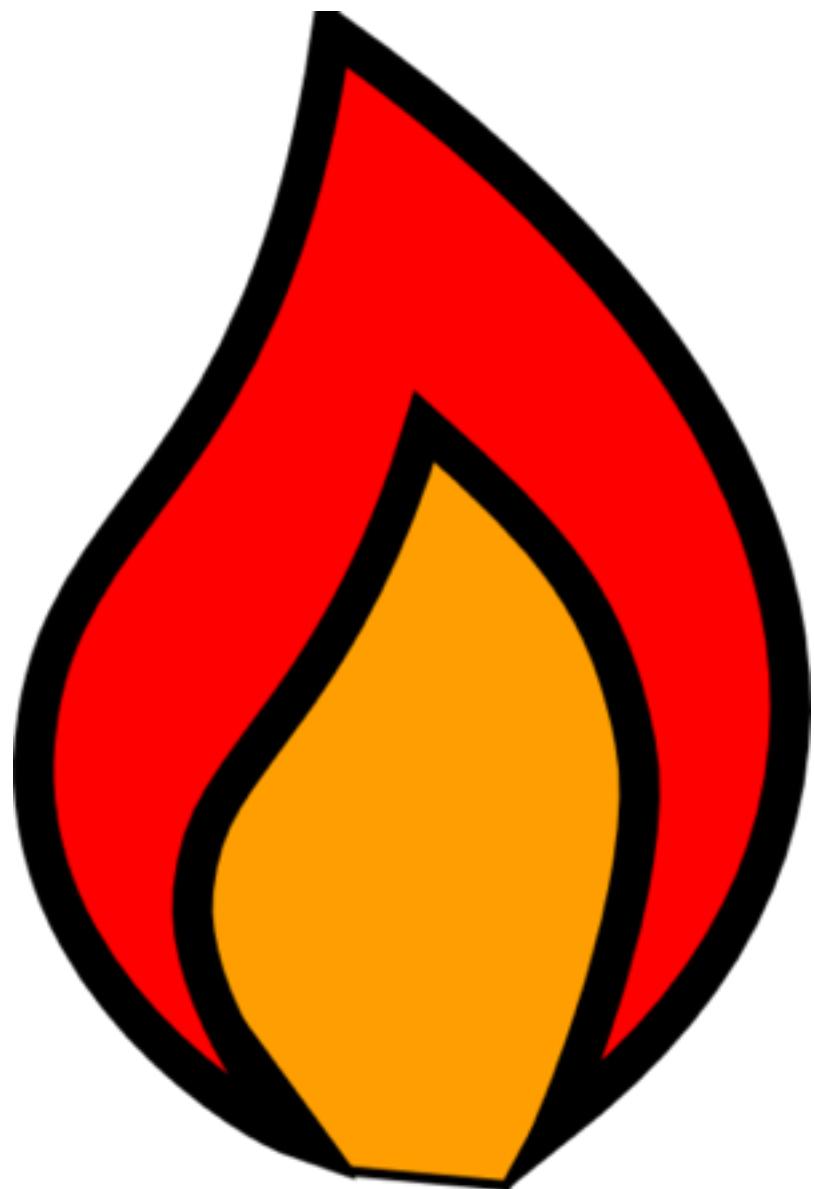


Buffer event data

Performant and persistent

Decoupling multiple sources
from processing

Maintaining Data consistency in a Streaming Architecture



Checkpointing/periodic backups to
manage failures

Deal with out of order events

Replay streams when required

Stream processing requirements



High throughput, low latency

Fault tolerance with low overhead

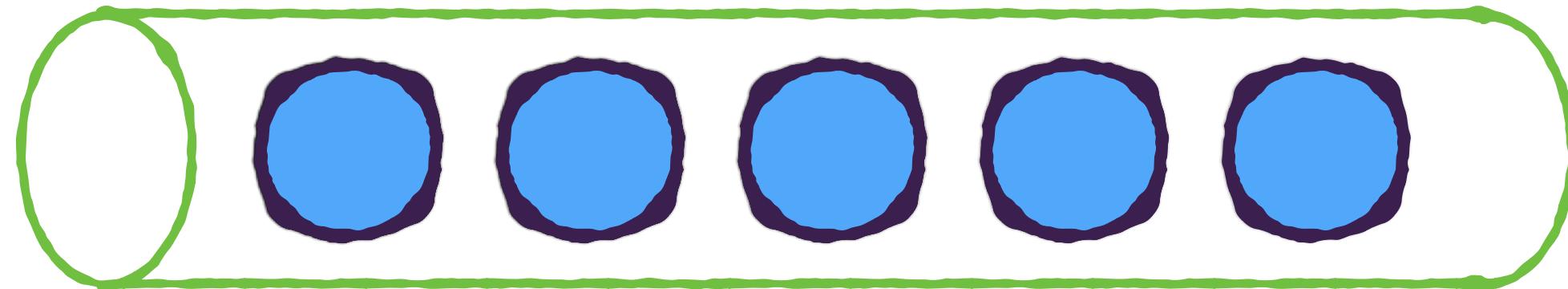
Manage out of order events

Easy to use, maintainable

Replay streams

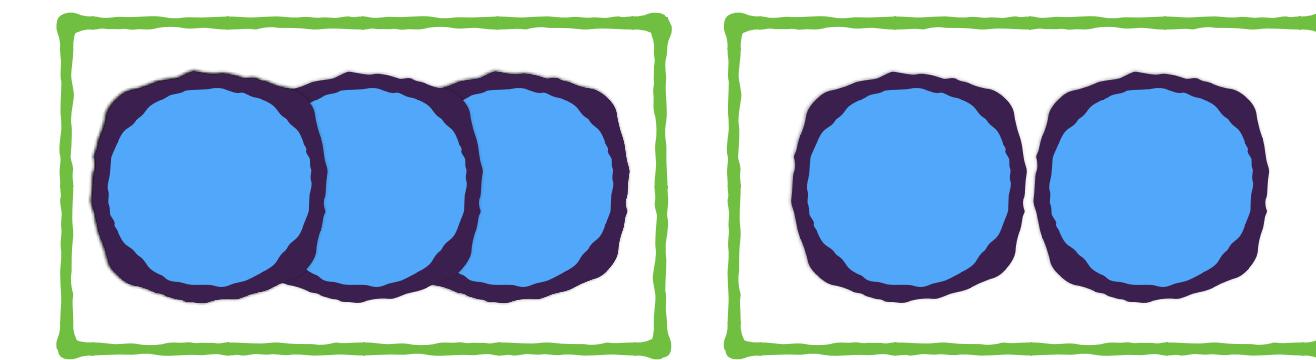
Streaming Architecture

Treat the stream
as a stream



Stream-first

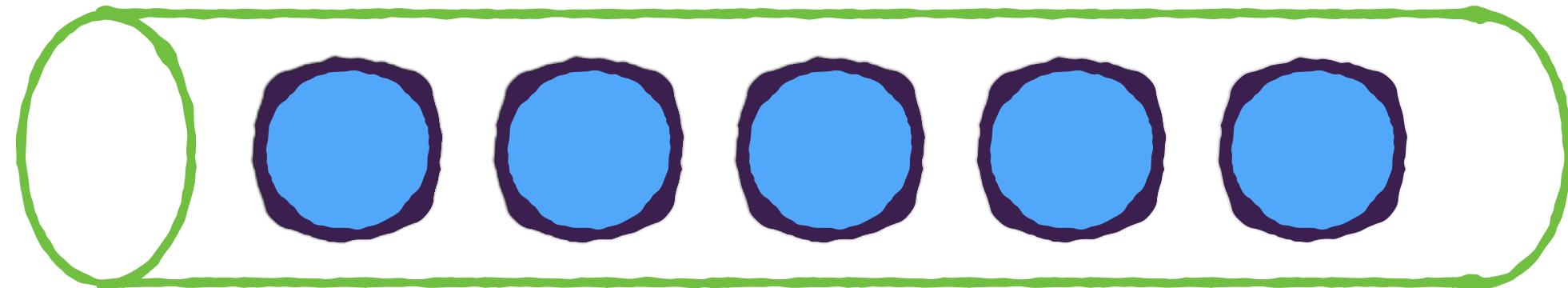
Treat the stream as a
collection of batches



Micro-batching

Streaming Architecture

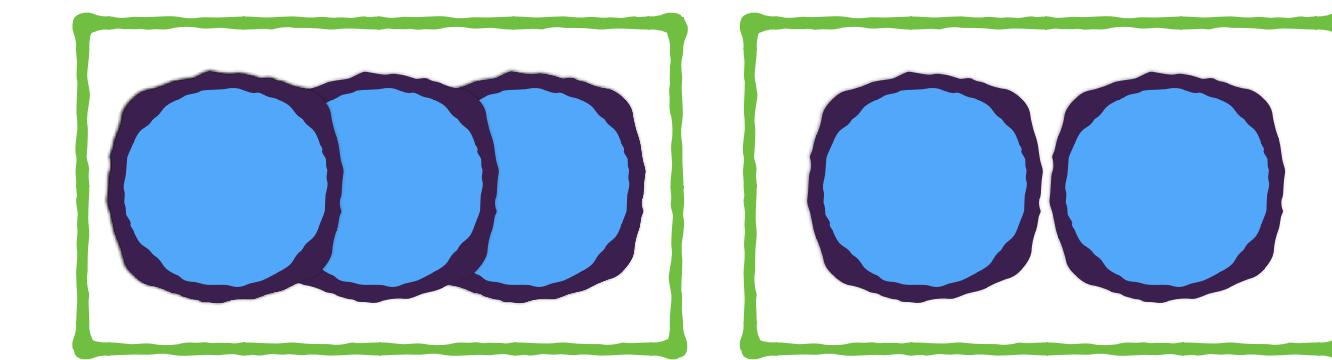
Stream-first



Flink



Micro-batching

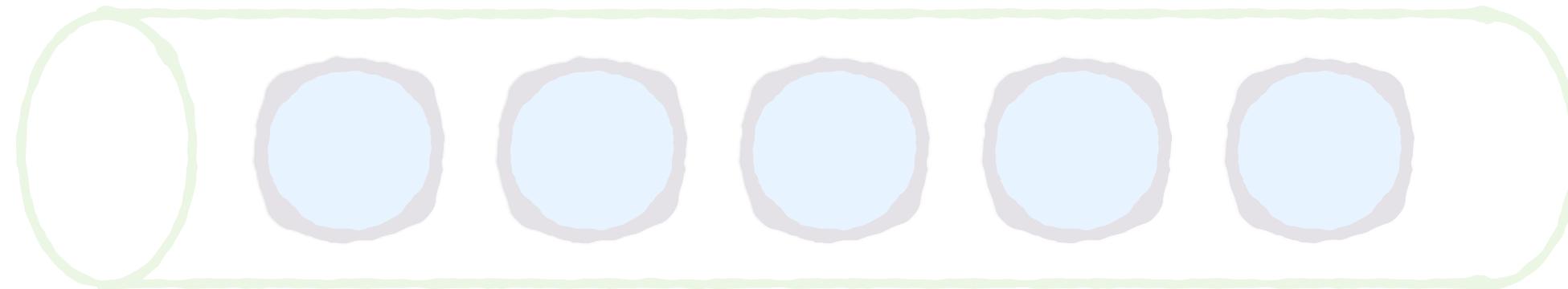


Spark
Streaming



Streaming Architecture

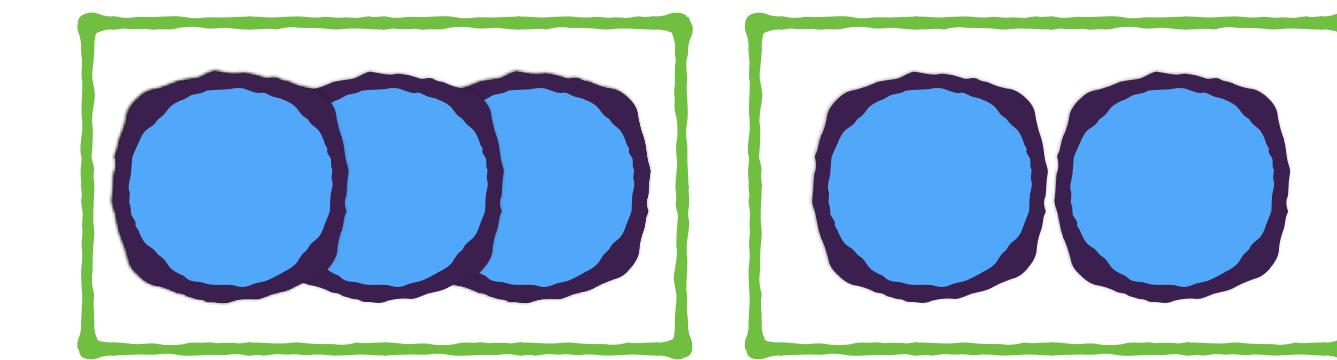
Stream-first



Flink



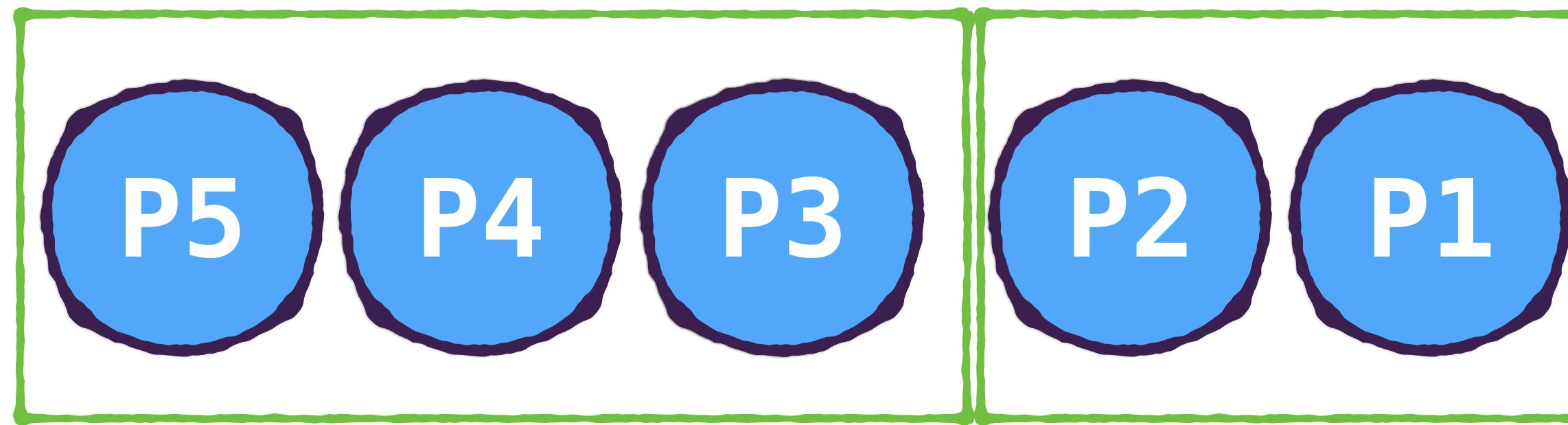
Micro-batching



Spark
Streaming



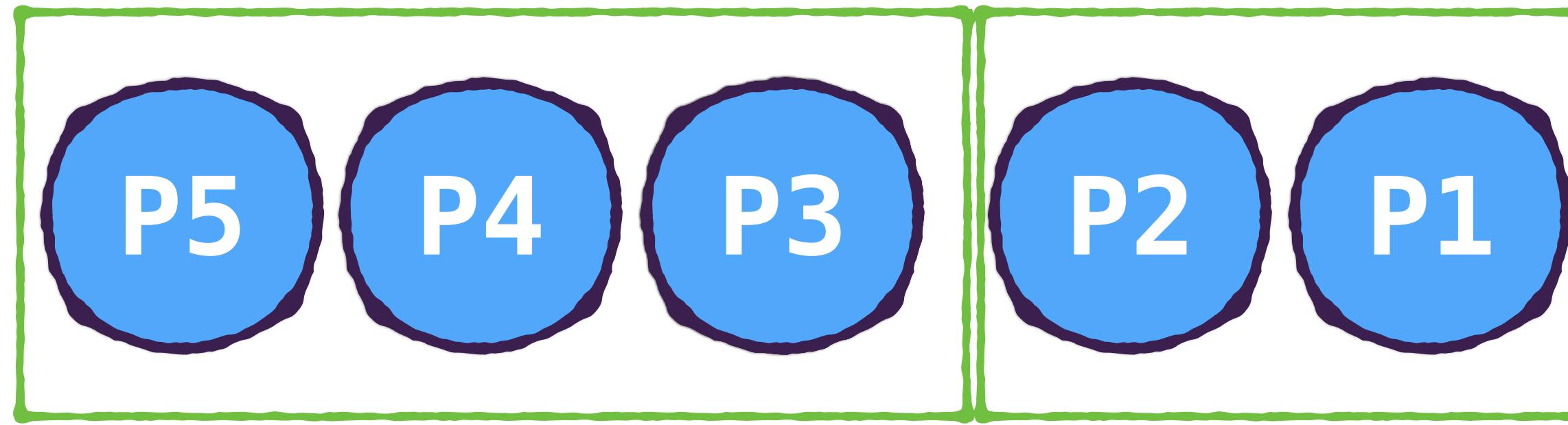
Micro-batching



Small batches

Approximate real-time
stream processing

Micro-batching

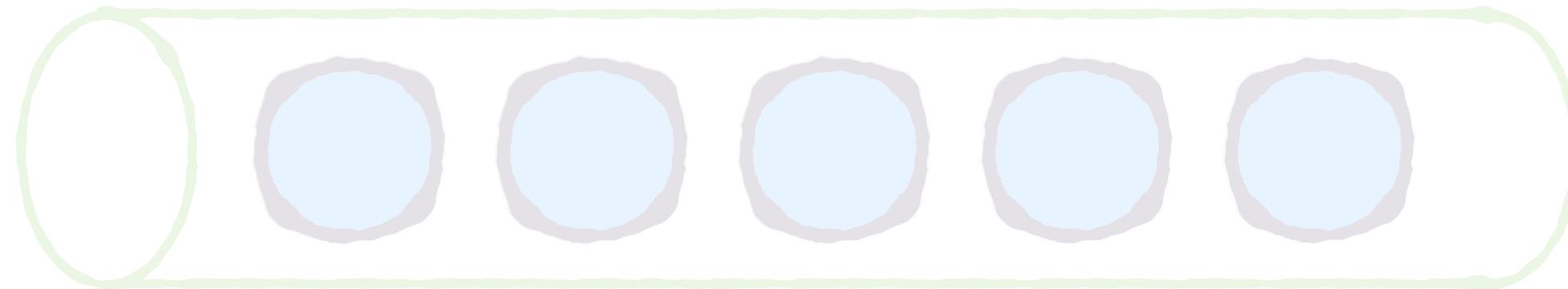


Exactly once semantics, replay micro-batches

Latency-throughput trade-off based on batch sizes

Streaming Architecture

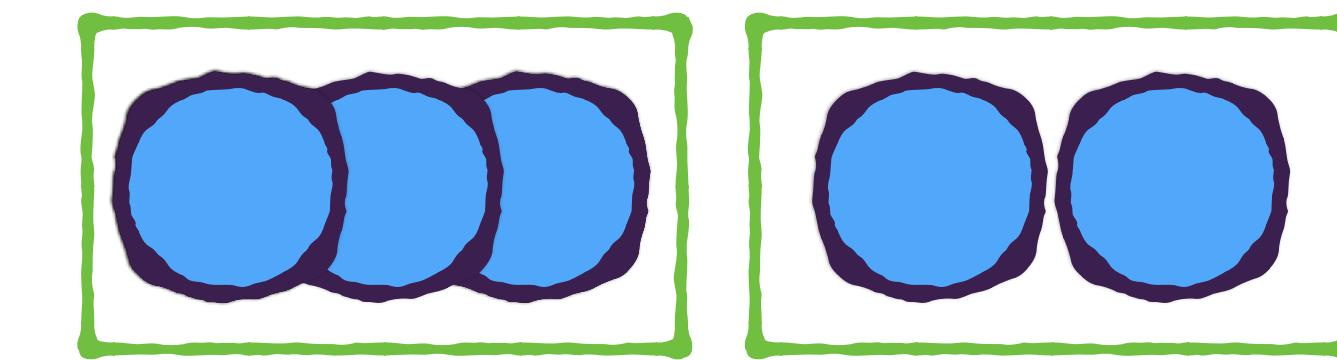
Stream-first



Flink



Micro-batching

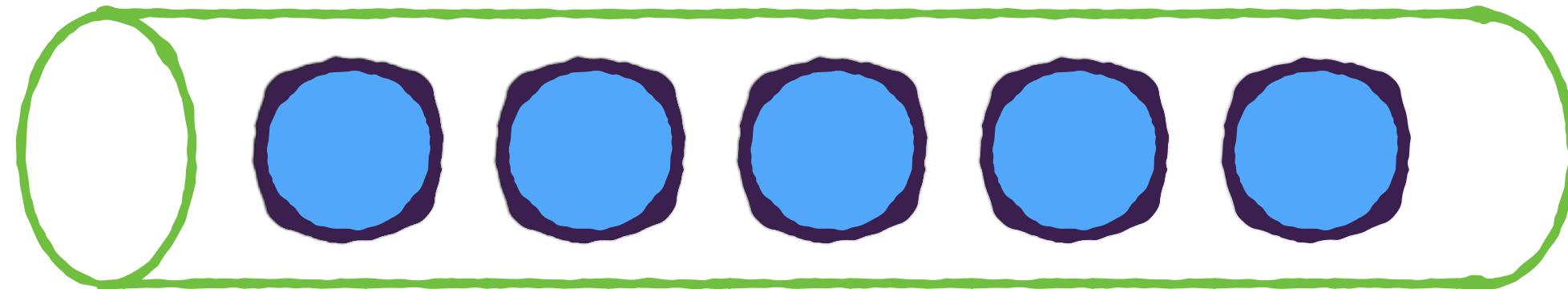


Spark
Streaming



Streaming Architecture

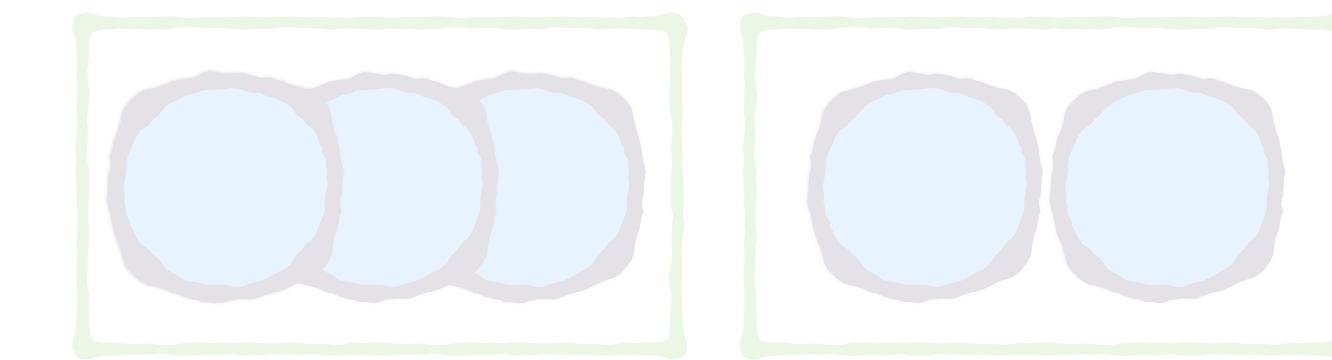
Stream-first



Flink



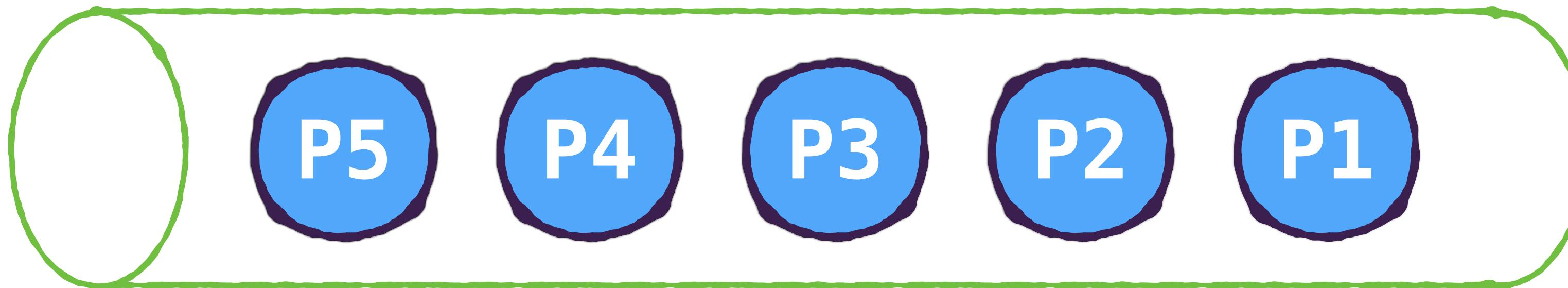
Micro-batching



Spark
Streaming



Stream-first

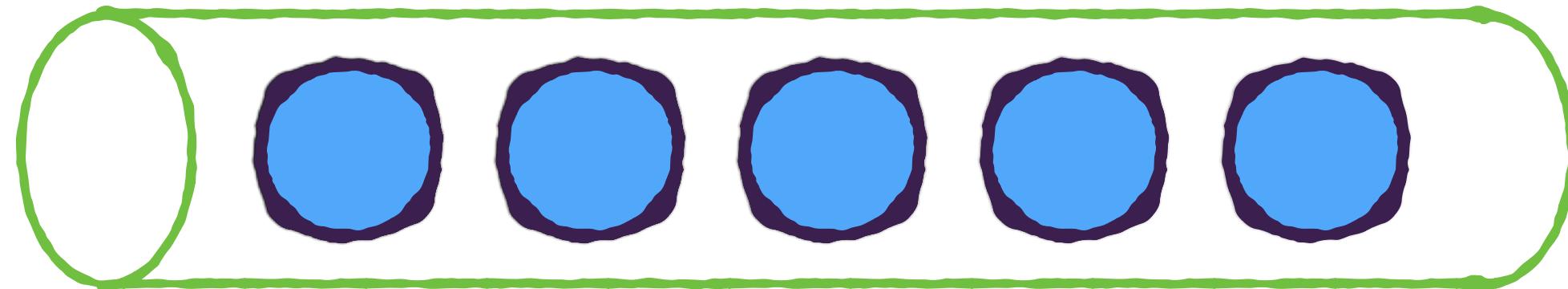


Process 1 event at a time

Batch processing as **a special case** of
stream processing

Streaming Architecture

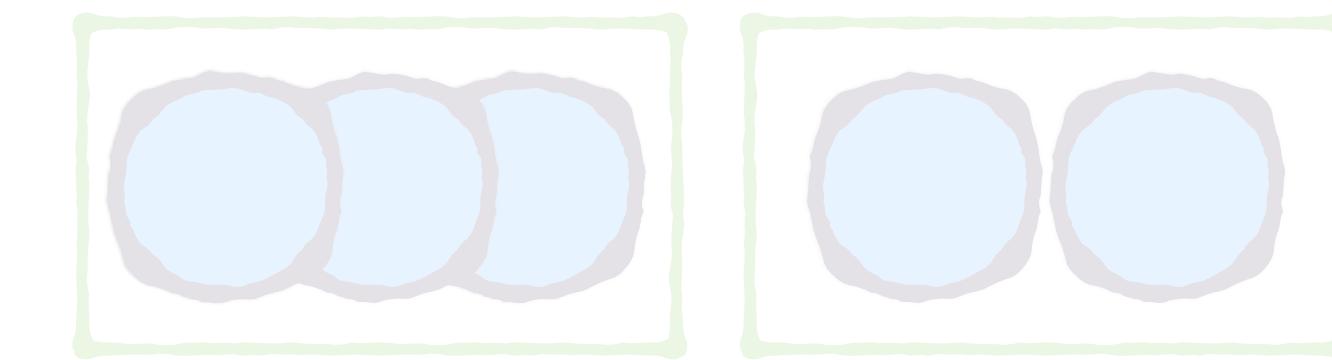
Stream-first



Flink



Micro-batching



Spark
Streaming



Flink



Deals with **out of order** data

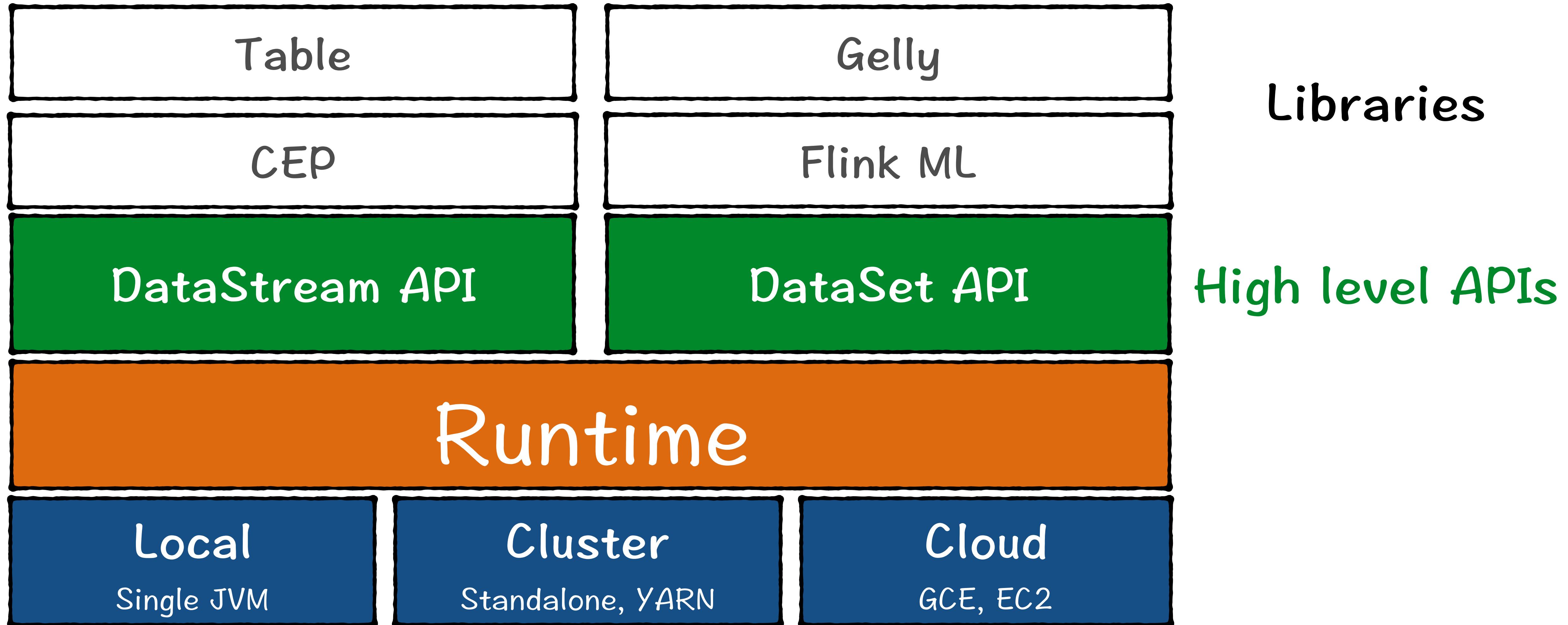
Exactly once processing

Great control over **Windowing**

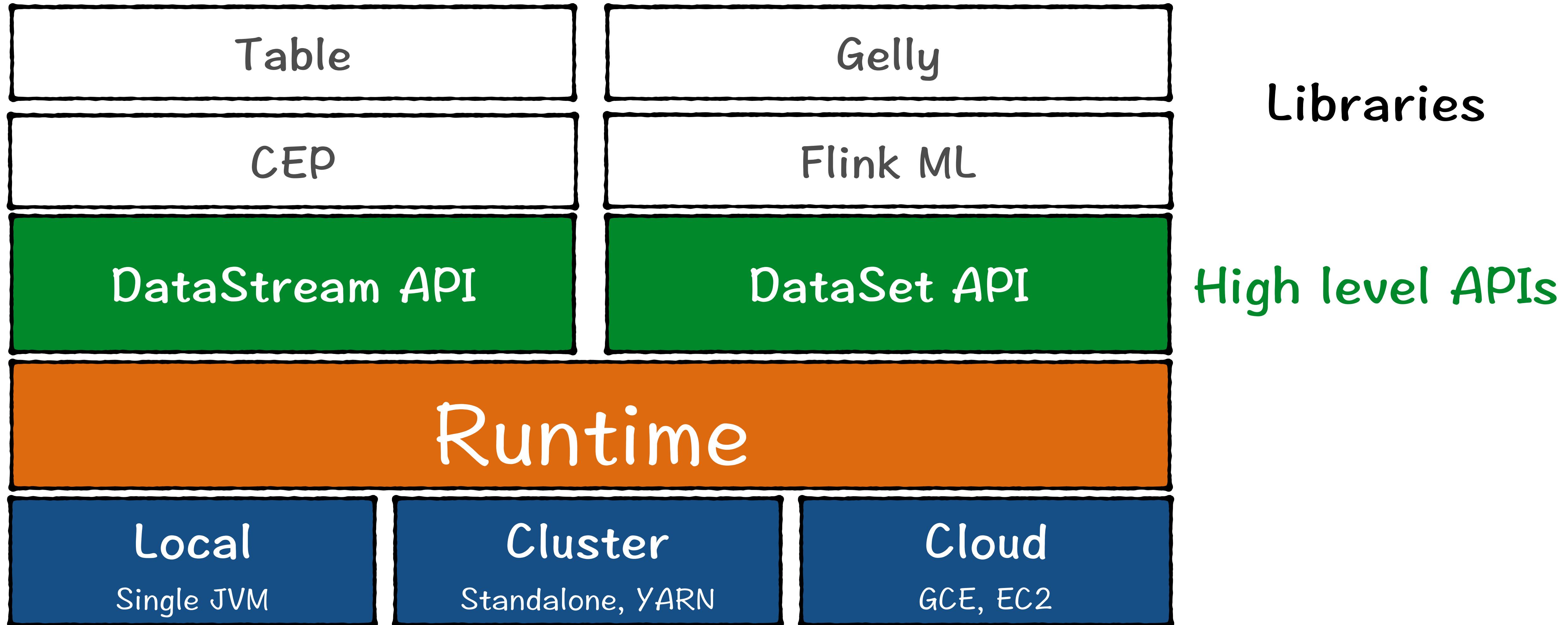
Lightweight fault tolerance and checkpointing

Distributed

Flink



Flink



Flink



DataStream API

DataSet API

High level APIs

Runtime

Local

Single JVM

Cluster

Standalone, YARN

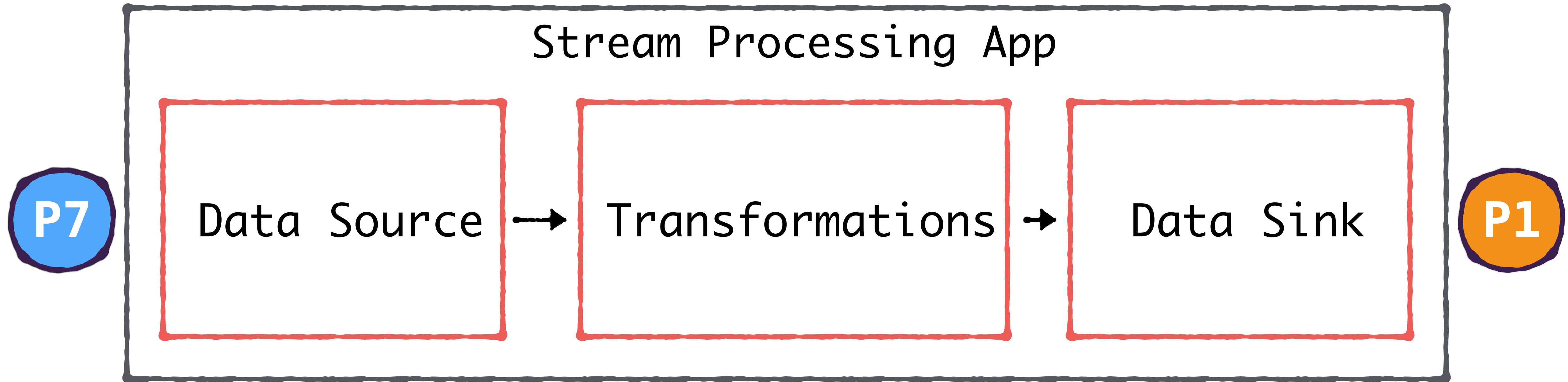
Cloud

GCE, EC2

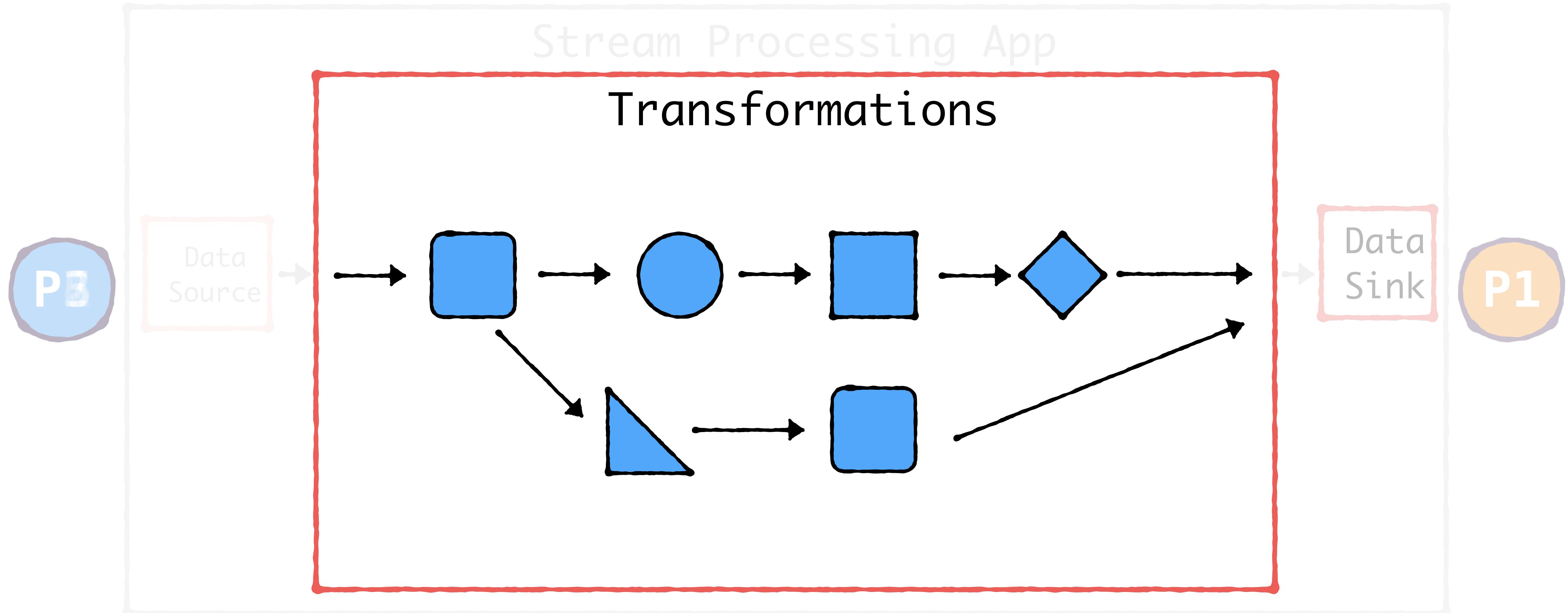
Flink Programming Model



Flink Programming Model

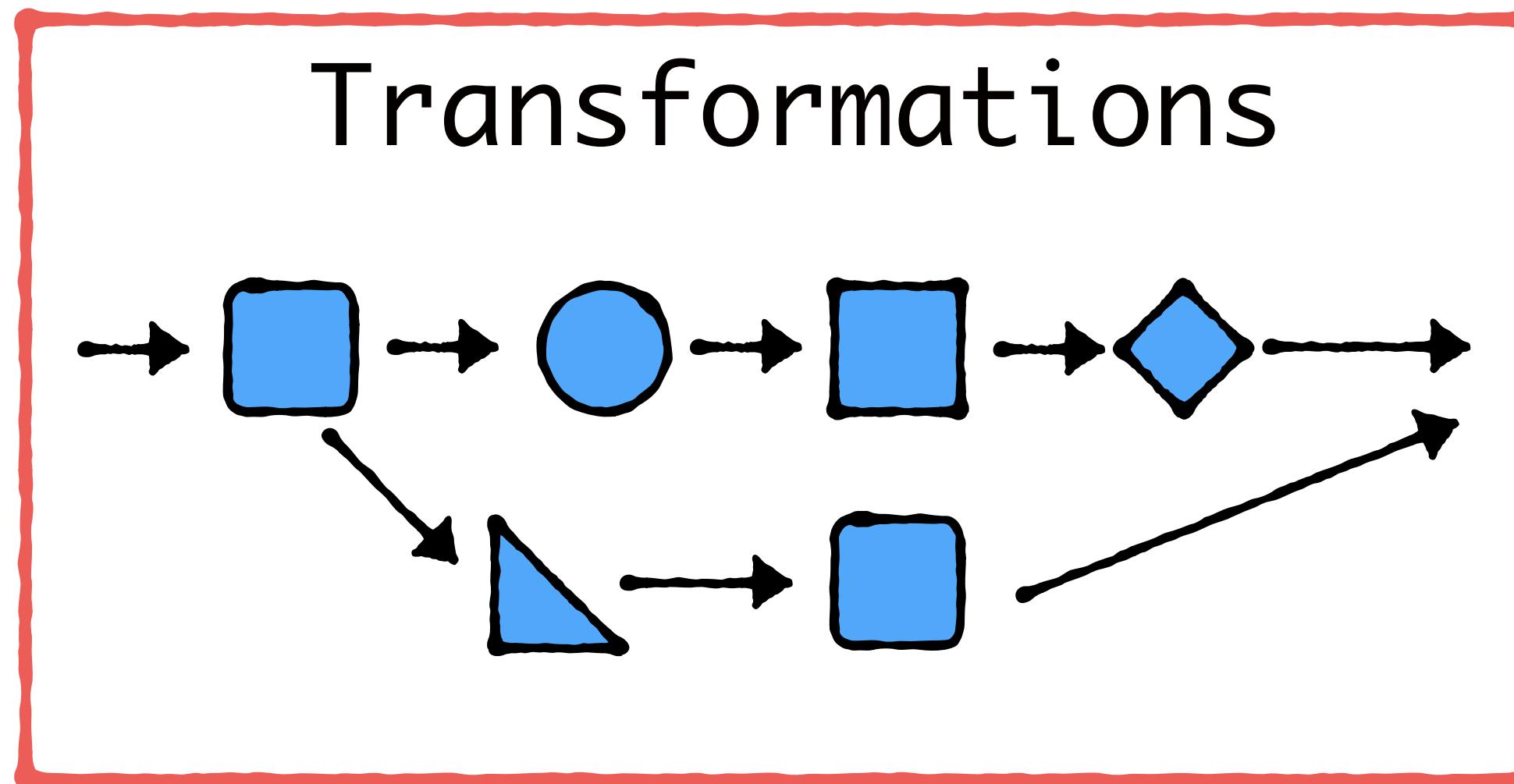


Flink Programming Model



Flink Programming Model

A directed acyclic graph



Users express
transformations

Flink translates
to a DAG of tasks