Structured Streaming in Apache Spark

Performing Streaming Operations in Spark

Janani Ravi

Co-founder, Loonycorn

www.loonycorn.com

System and Software Requirements

- Windows Subsystem for Linux, MacOS, or Linux machine

- Apache Spark version 3.5.x+

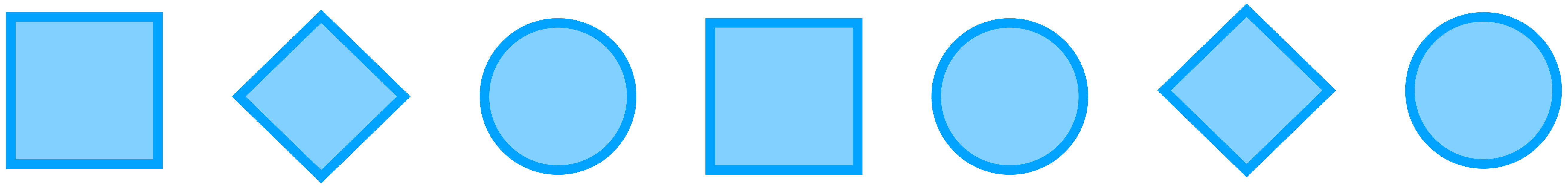
- Apache Kafka 3.7+

Batch and Stream Processing

Bounded datasets are processed in batches

Unbounded datasets are processed as streams

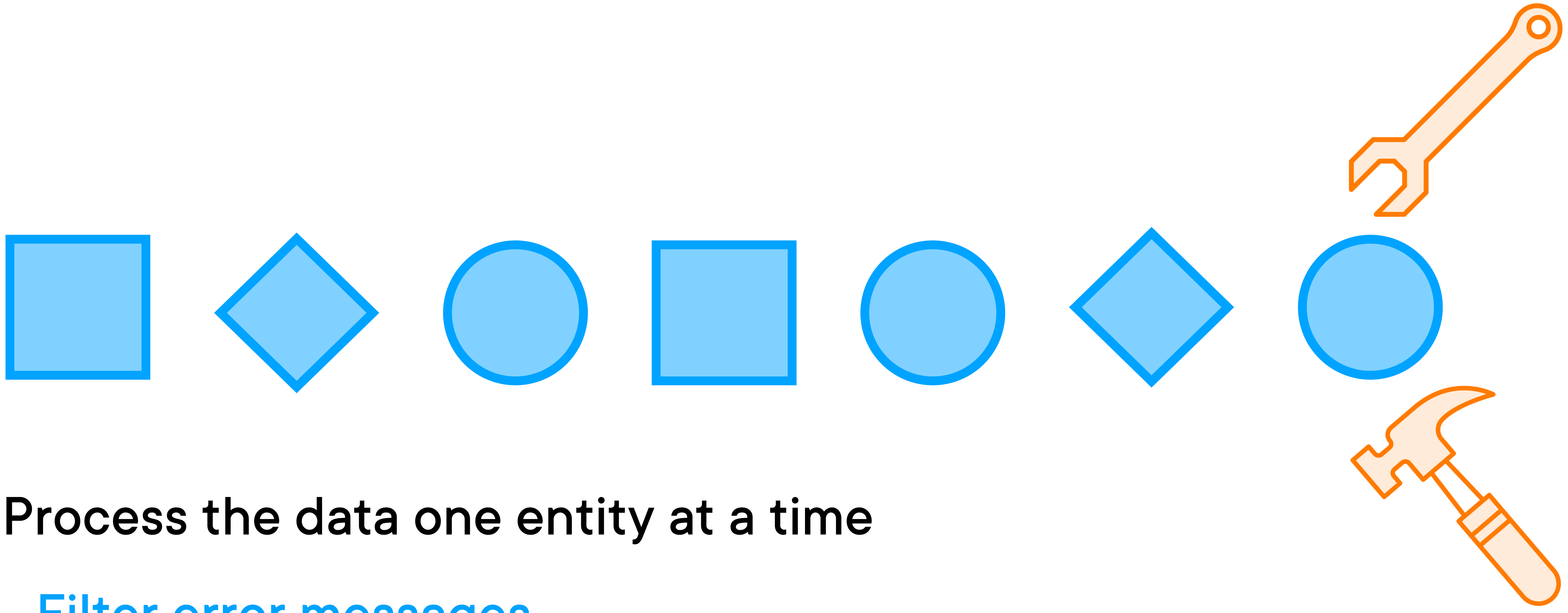
Stream Processing



Data is received as a stream

- Log messages
- Tweets (messages on X)
- Climate sensor data

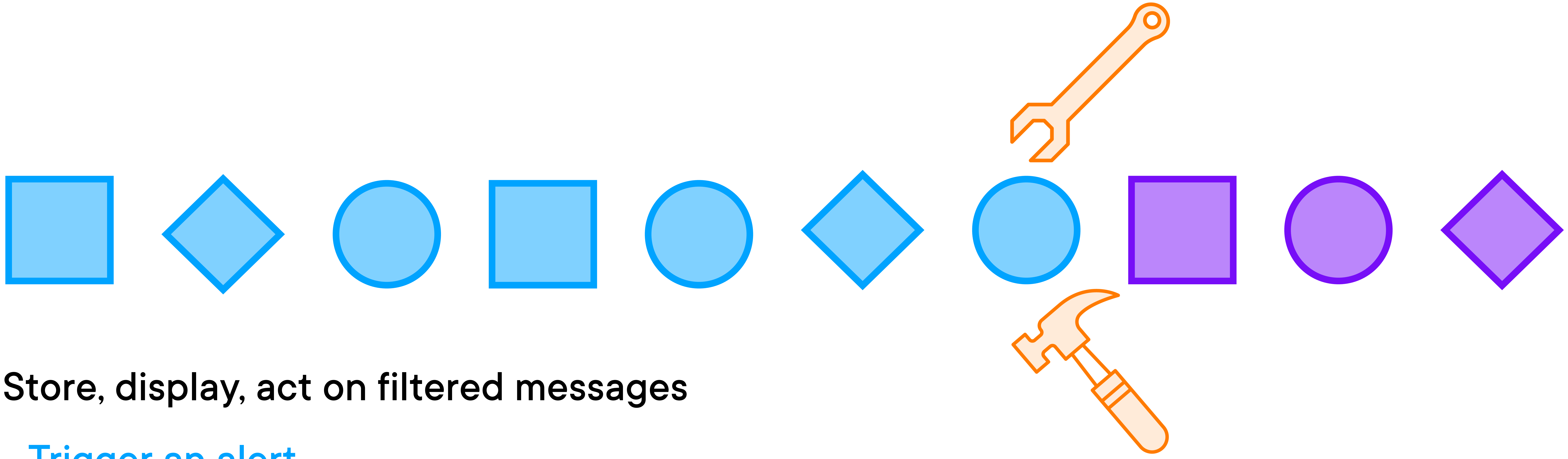
Stream Processing



Process the data one entity at a time

- Filter error messages
- Find references to the latest movies
- Track weather patterns

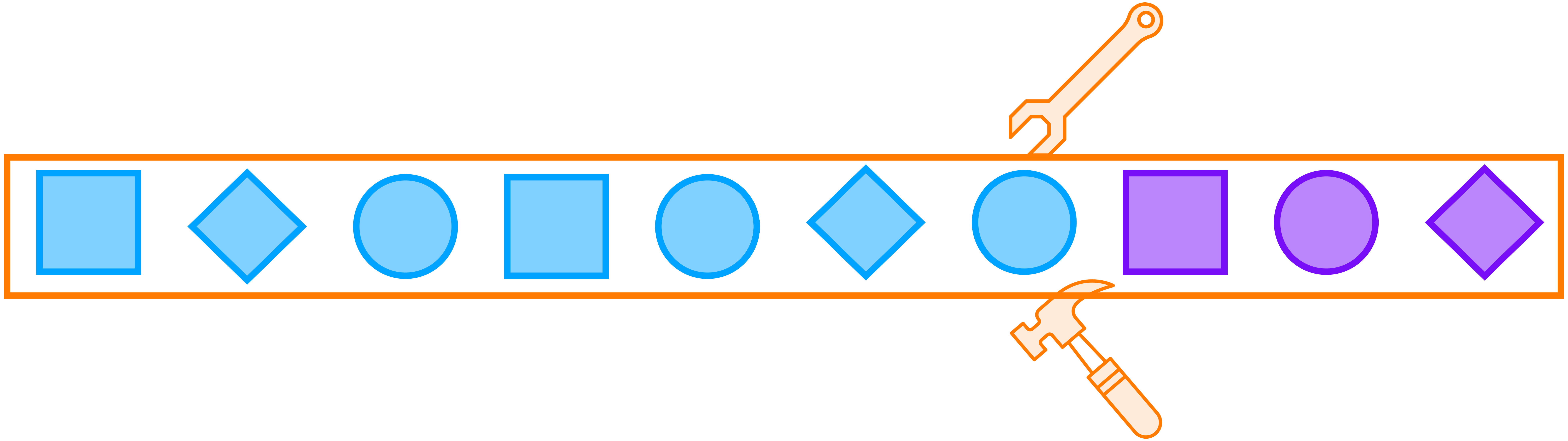
Stream Processing



Store, display, act on filtered messages

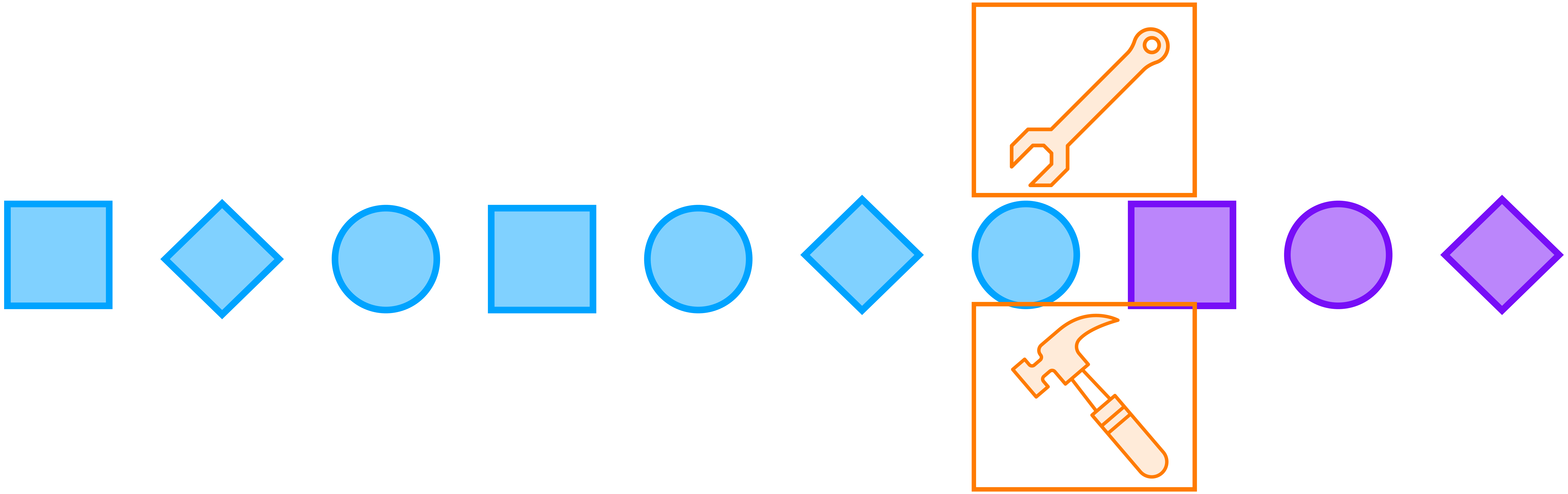
- Trigger an alert
- Show trending graphs
- Warn of sudden squalls

Stream Processing



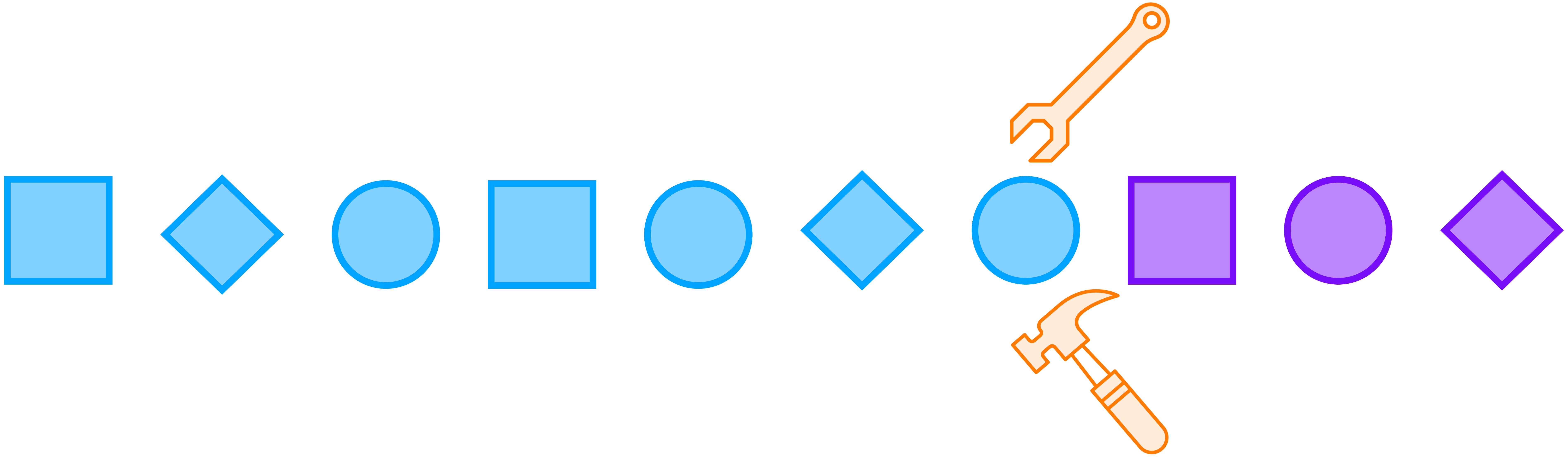
Streaming data

Stream Processing



Stream processing

Stream Processing



Batch vs. Stream Processing

Batch	vs.	Stream
Bounded, finite datasets		Unbounded, infinite datasets
Slow pipeline from data ingestion to analysis		Processing immediate, as data is received
Periodic updates as jobs complete		Continuous updates as jobs run constantly
Order of data received unimportant		Order important, out of order arrival tracked
Single global state of the world at any point in time		No global state, only history of events received

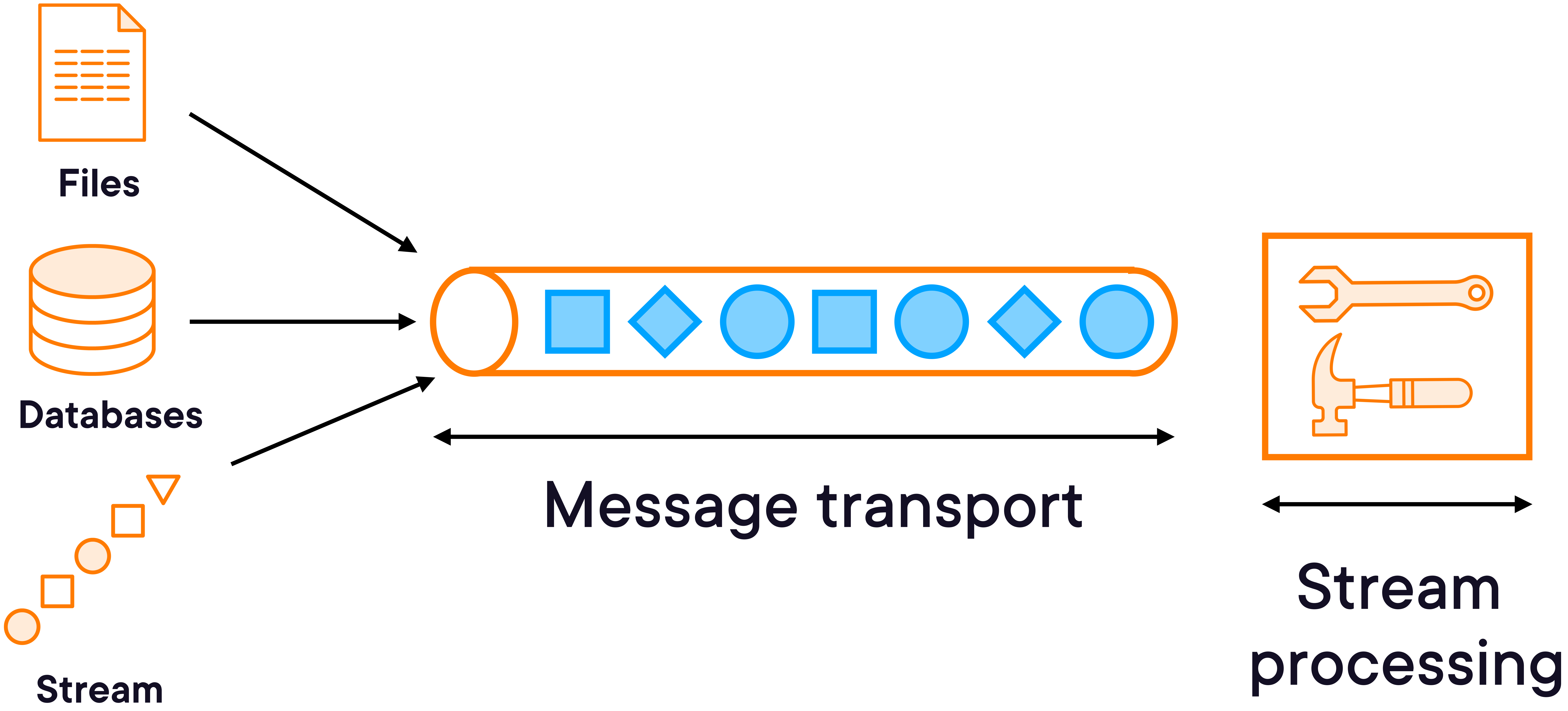
Storage Systems for Batch Data

Files

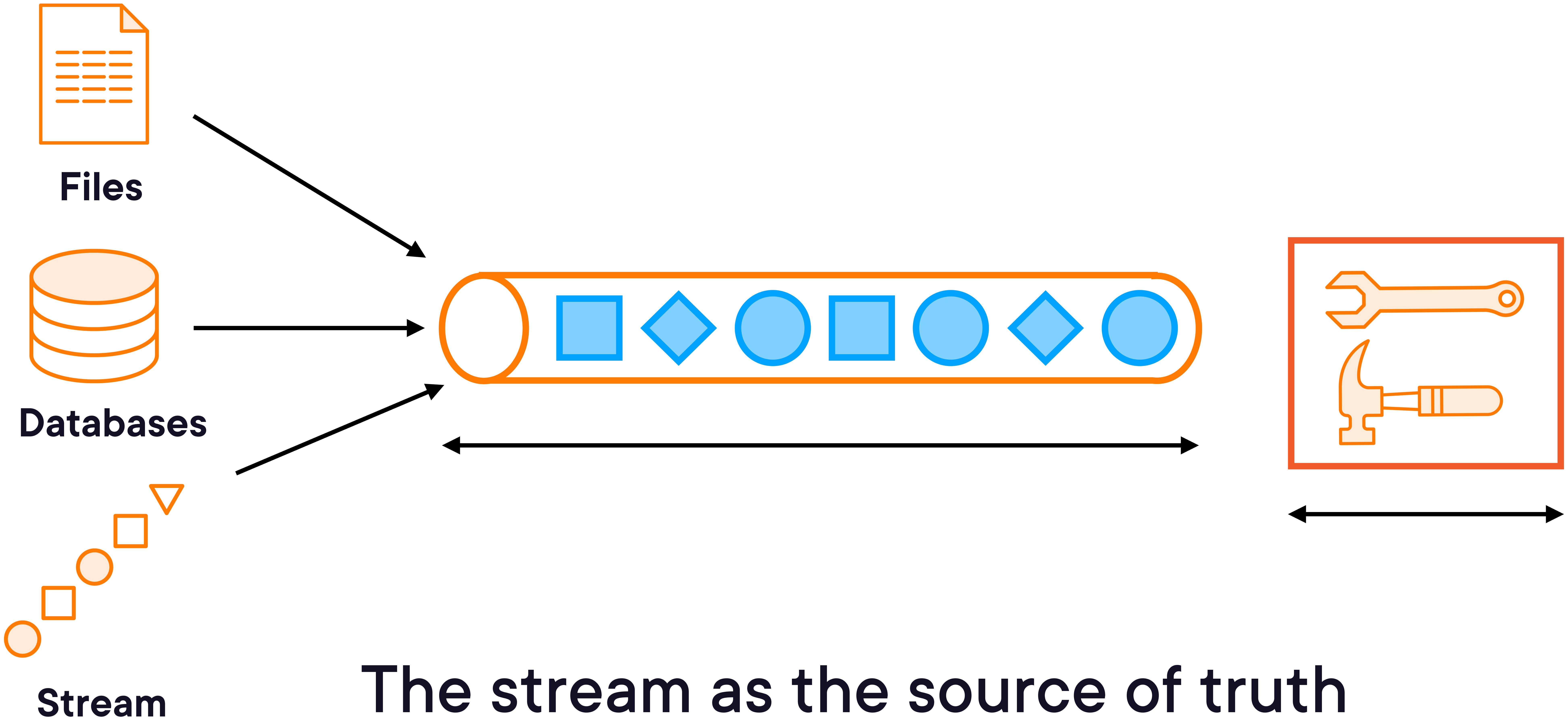
Databases

Reliable storage as the source of truth

Stream-first Architecture

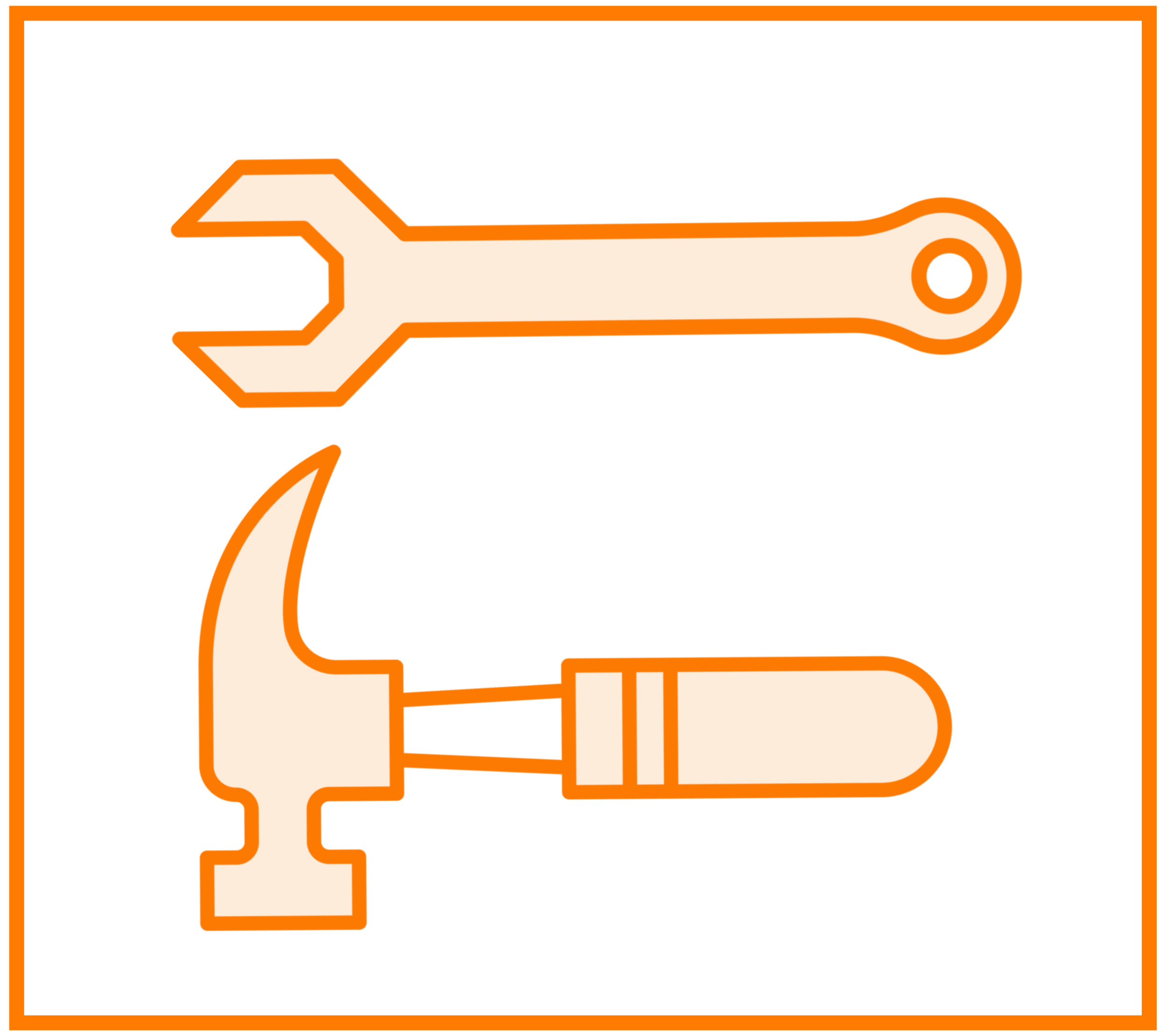


Stream-first Architecture



The stream as the source of truth

Stream Processing



High throughput, low latency

Fault tolerance with low overhead

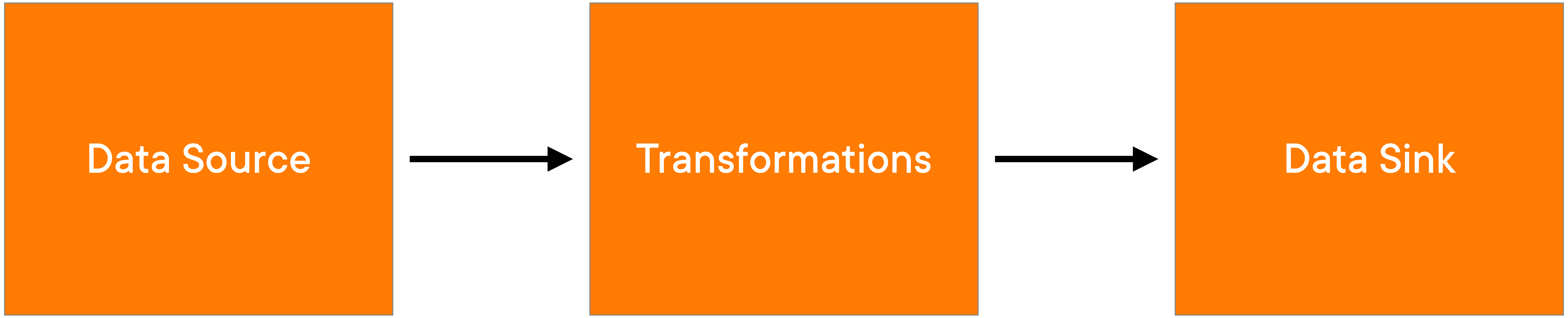
Manage out of order events

Easy to use, maintainable

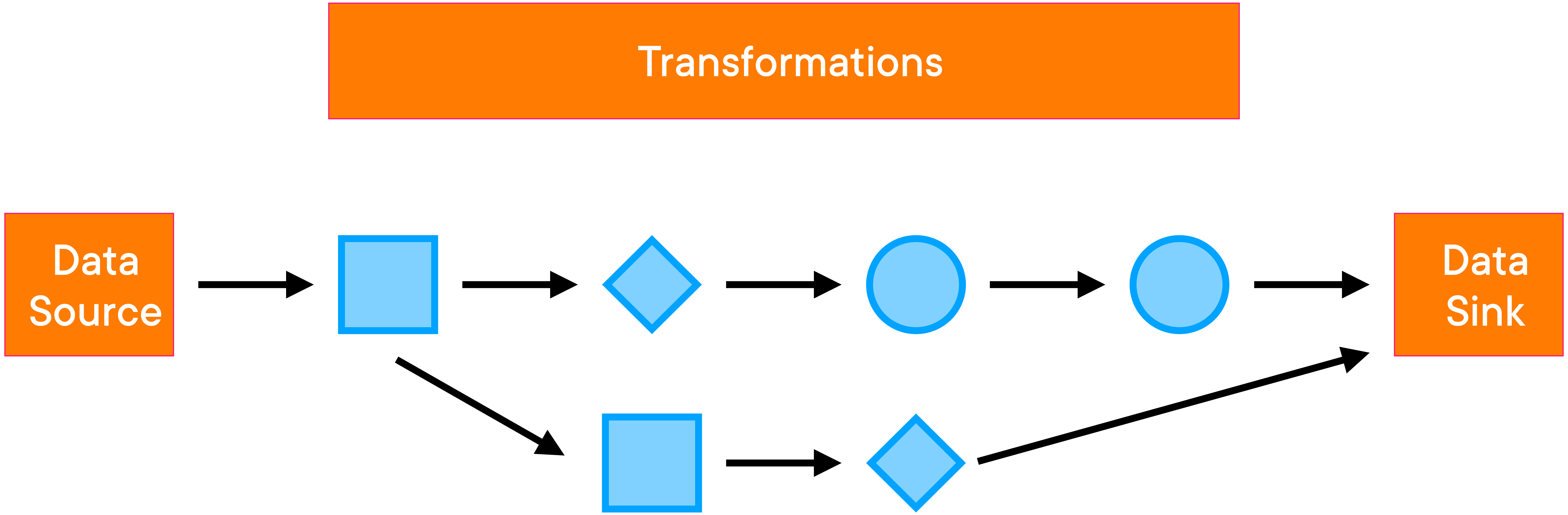
Replay streams

Spark Streaming, Storm, Flink

Stream Processing Model

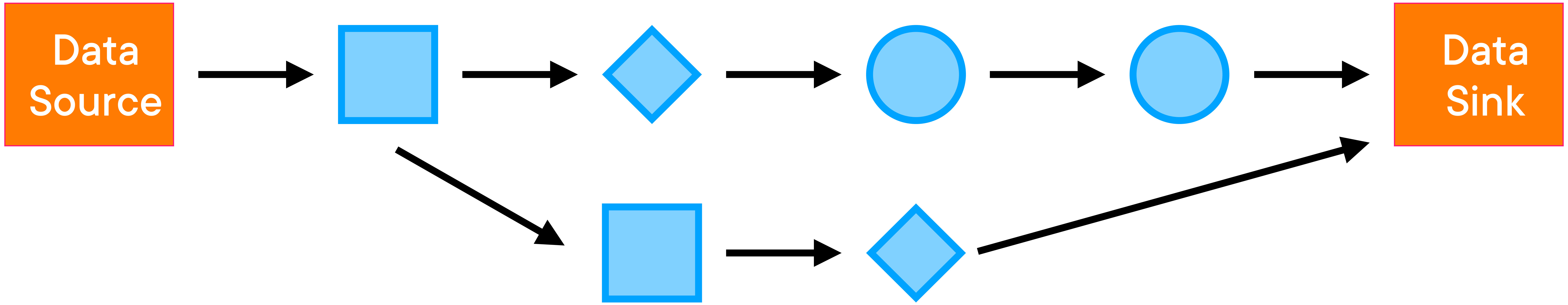


Stream Processing Model



Transformations

A directed-acyclic graph



Streaming in Apache Spark

Structured Streaming

Structured Streaming is a scalable and fault-tolerant stream processing engine built on the Spark SQL engine.

https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Structured Streaming

Batch and stream code virtually identical

Fault tolerance and exactly-once guarantees

Handles event-time and late data

Spark Streaming

What

A high-level API that takes burden off user

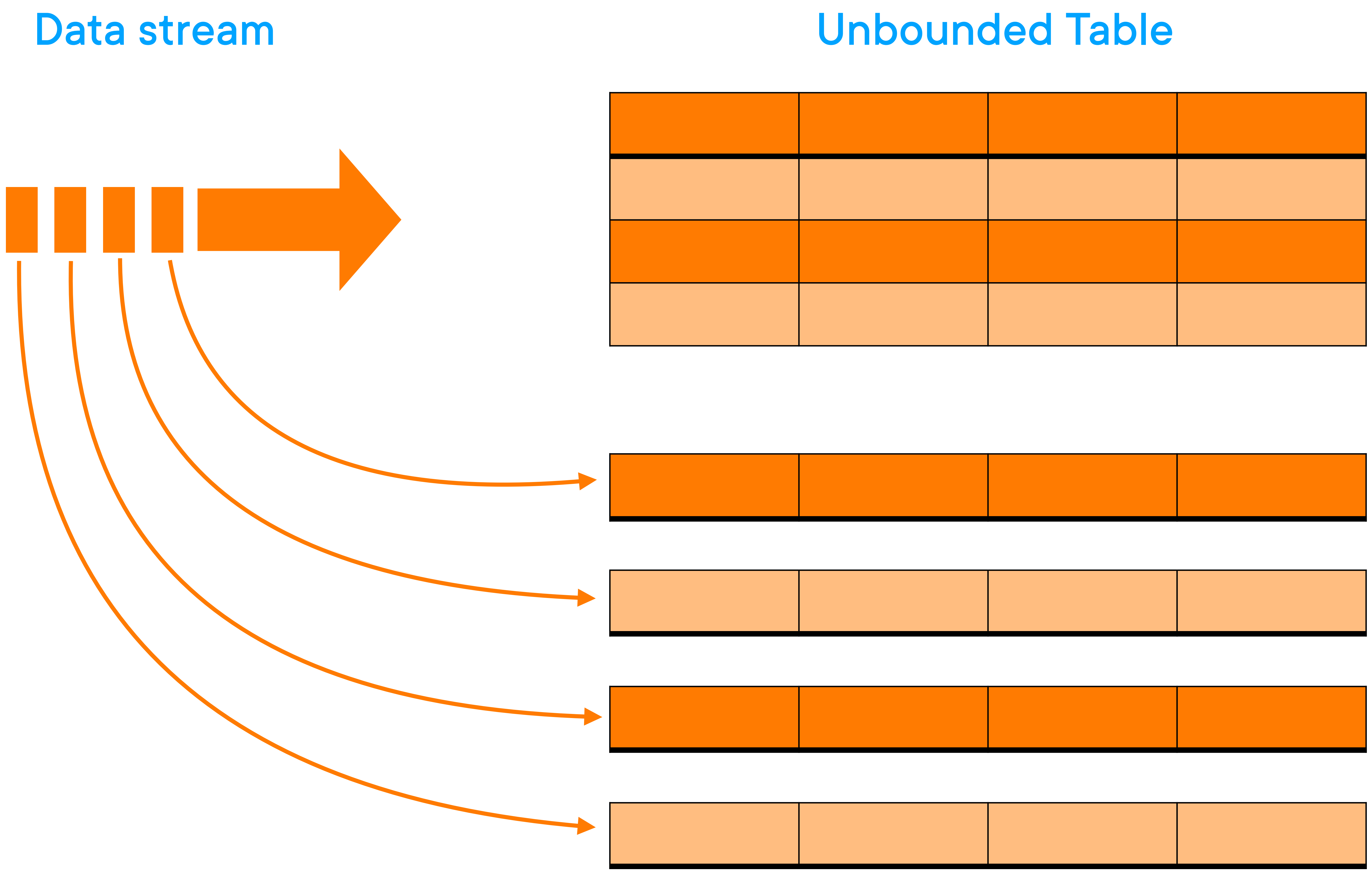
How

Micro-batch processing with exactly-once fault-tolerance

Why

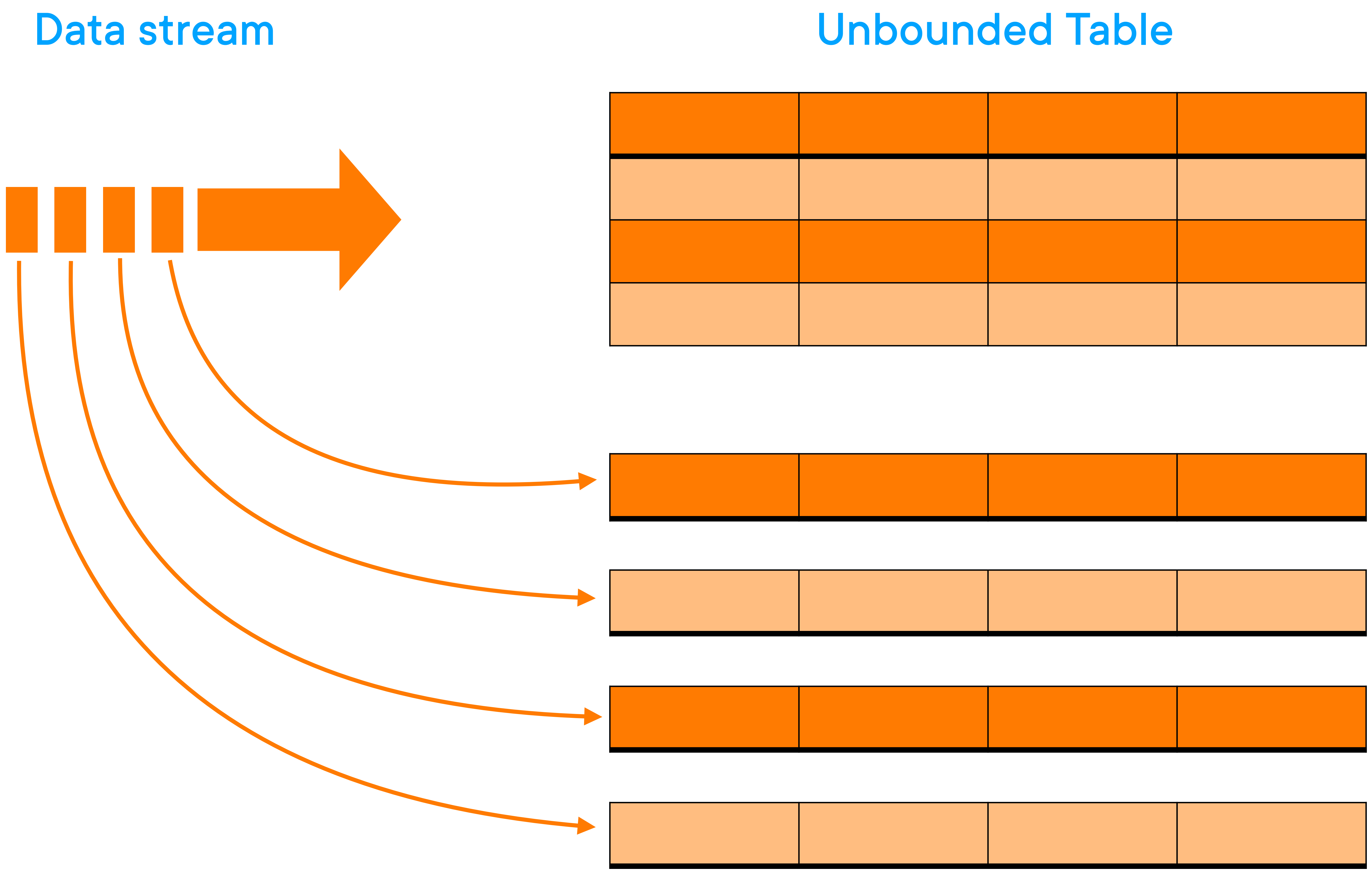
Code virtually identical for batch and streaming

Batch Is Simply Prefix of Stream



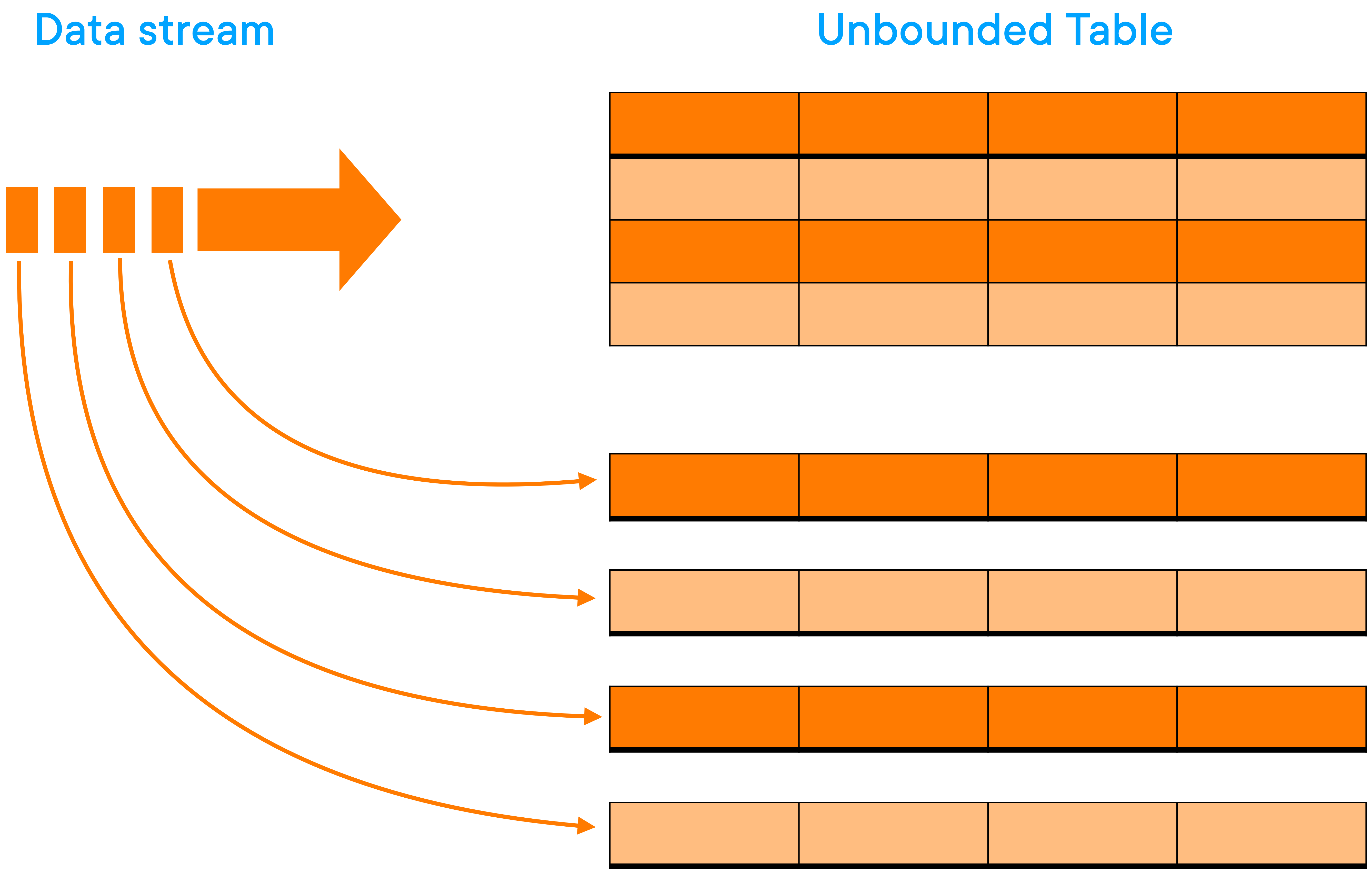
Every data item that is arriving on the stream is like a new row being appended to the input table

Batch Is Simply Prefix of Stream



In other words, the input table (batch) is simply a prefix of the stream

Batch Is Simply Prefix of Stream



All operations that can be performed on data frames can be performed on the stream

Structured Streaming treats a live data stream as a table that is being continuously appended

Burden of stream-processing shifts from user to system

Prefix Integrity

Running job on continuous data yields same result as running job on batch data (where the batch is a prefix or snapshot of continuous data)