Exploring the Apache Spark Structured Streaming API for Processing Streaming Data

EXPLORING SOURCES AND SINKS

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Understanding Spark DataFrames

Data sources, data sinks, data transformations

Writing data to the console sink, file sink

Customizing write logic using the foreach sink and the foreachBatch sink

Prerequisites and Course Outline

Prerequisites

Comfortable programming in Python

Some exposure to Apache Spark and DataFrames

Some exposure to working with streaming data using Apache Spark

Prerequisite Courses

Conceptualizing the Processing Model for Apache Spark Structured Streaming

Course Outline

Exploring Sources and Sinks

Processing Streaming DataFrames

Performing Windowing Operations on Streams

Working with Streaming Joins

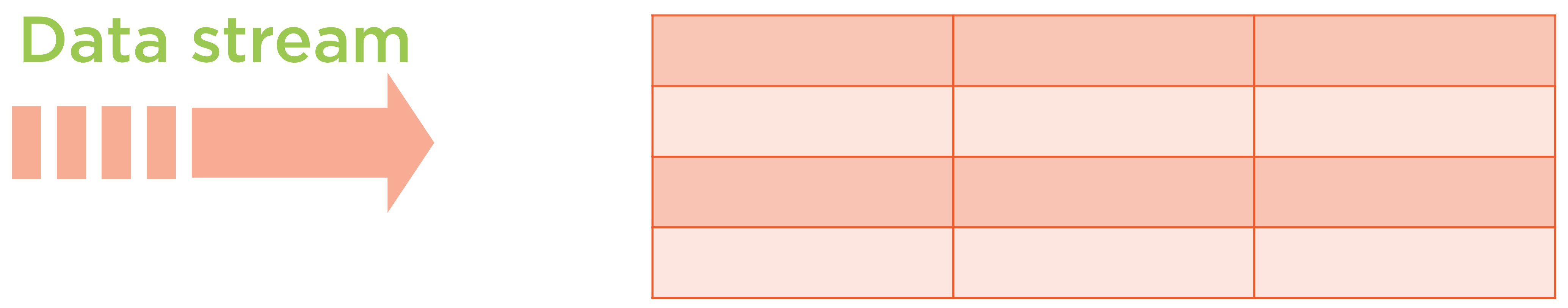
Managing and Monitoring Streaming Queries

Streaming DataFrames in Spark 2.x

DataFrame: Data in Rows and Columns

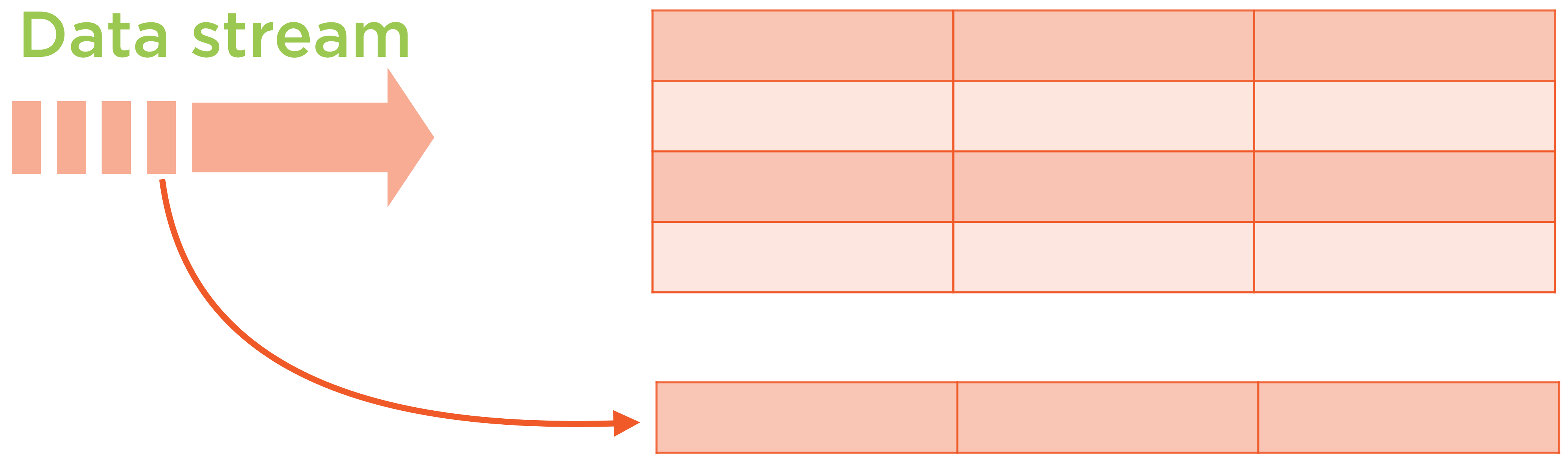
DATE	OPEN	...	PRICE
2016-12-01	772	...	779
2016-11-01	758	...	747
2006-01-01	302	...	309

Streaming Data Spark 2.x



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

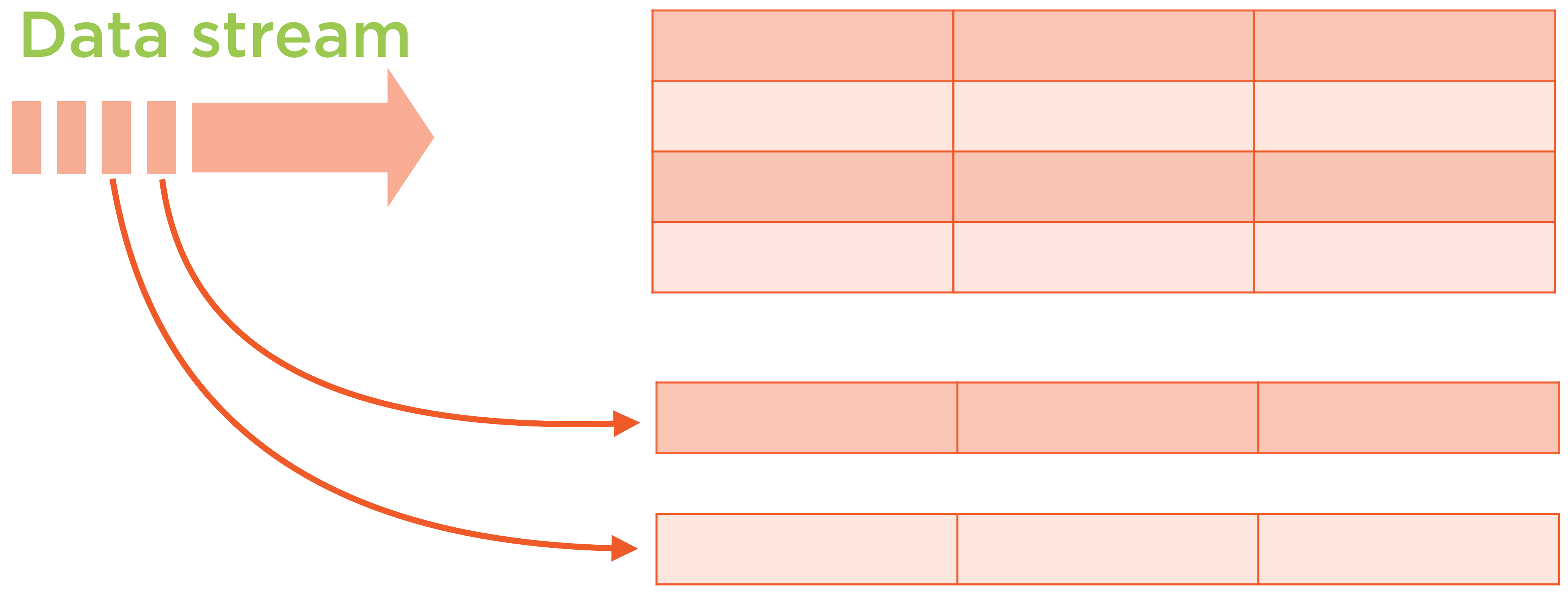
Streaming Data Spark 2.x



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

Data stream as an unbounded input table

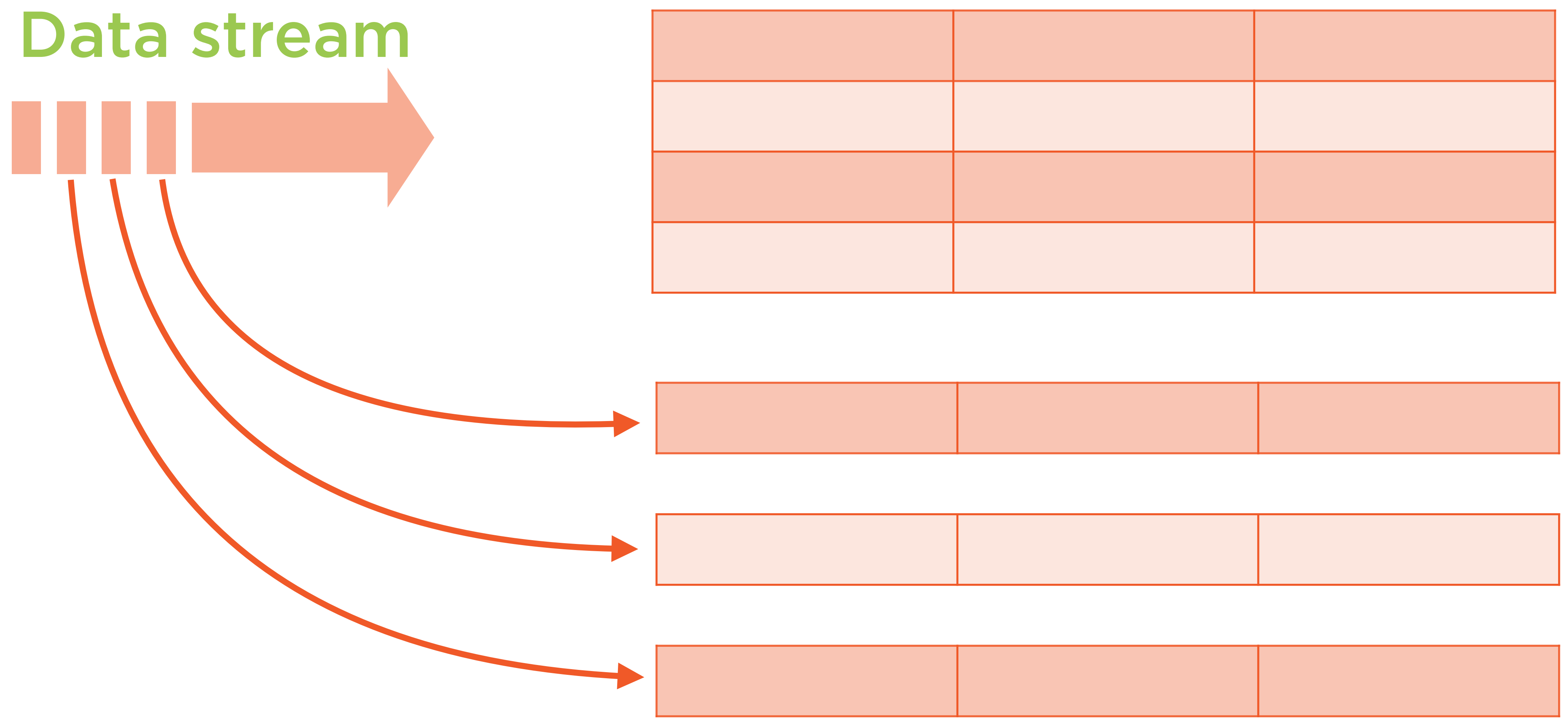
Streaming Data Spark 2.x



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

Data stream as an unbounded input table

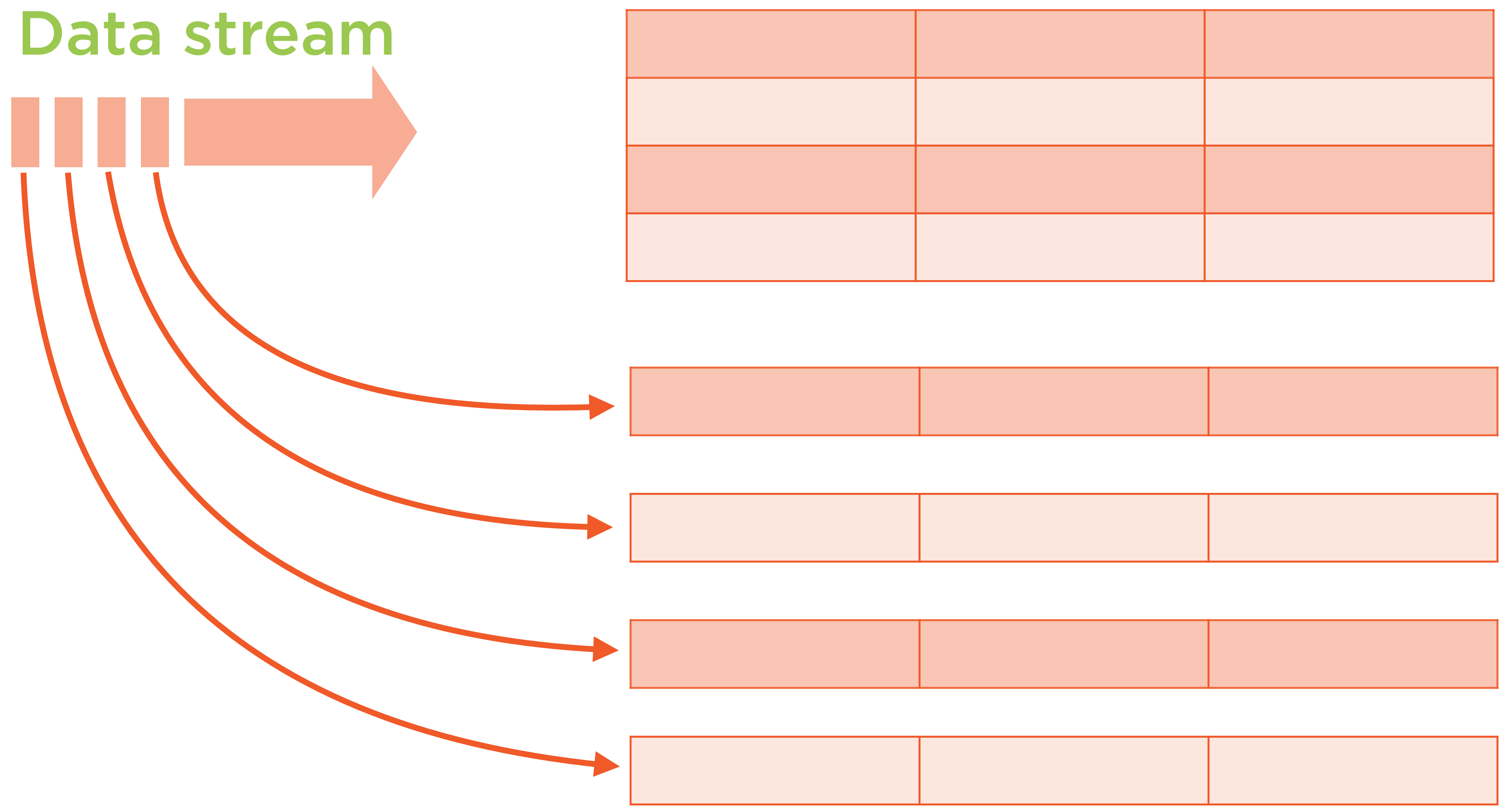
Streaming Data Spark 2.x



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

Data stream as an unbounded input table

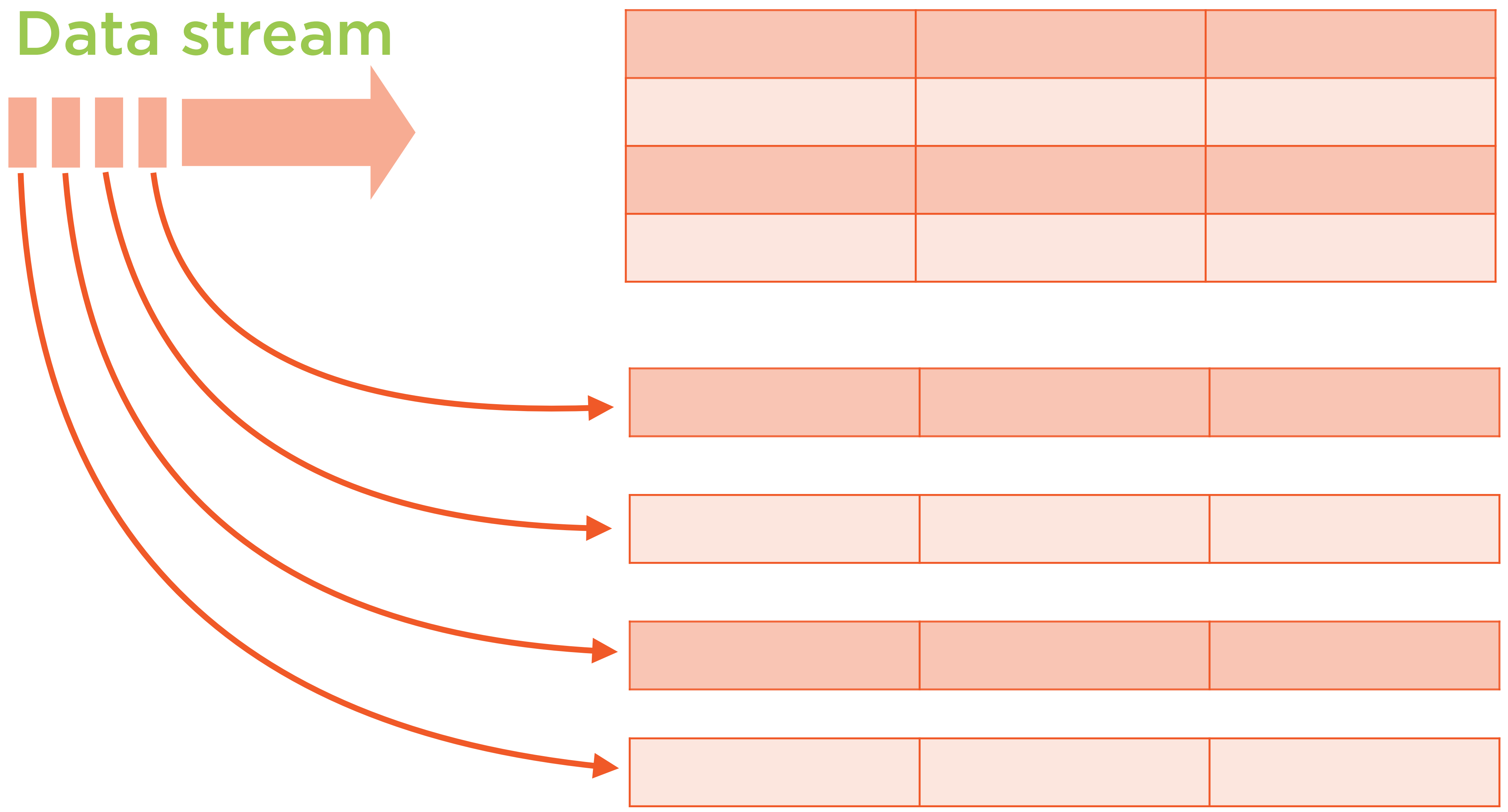
Streaming Data Spark 2.x



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

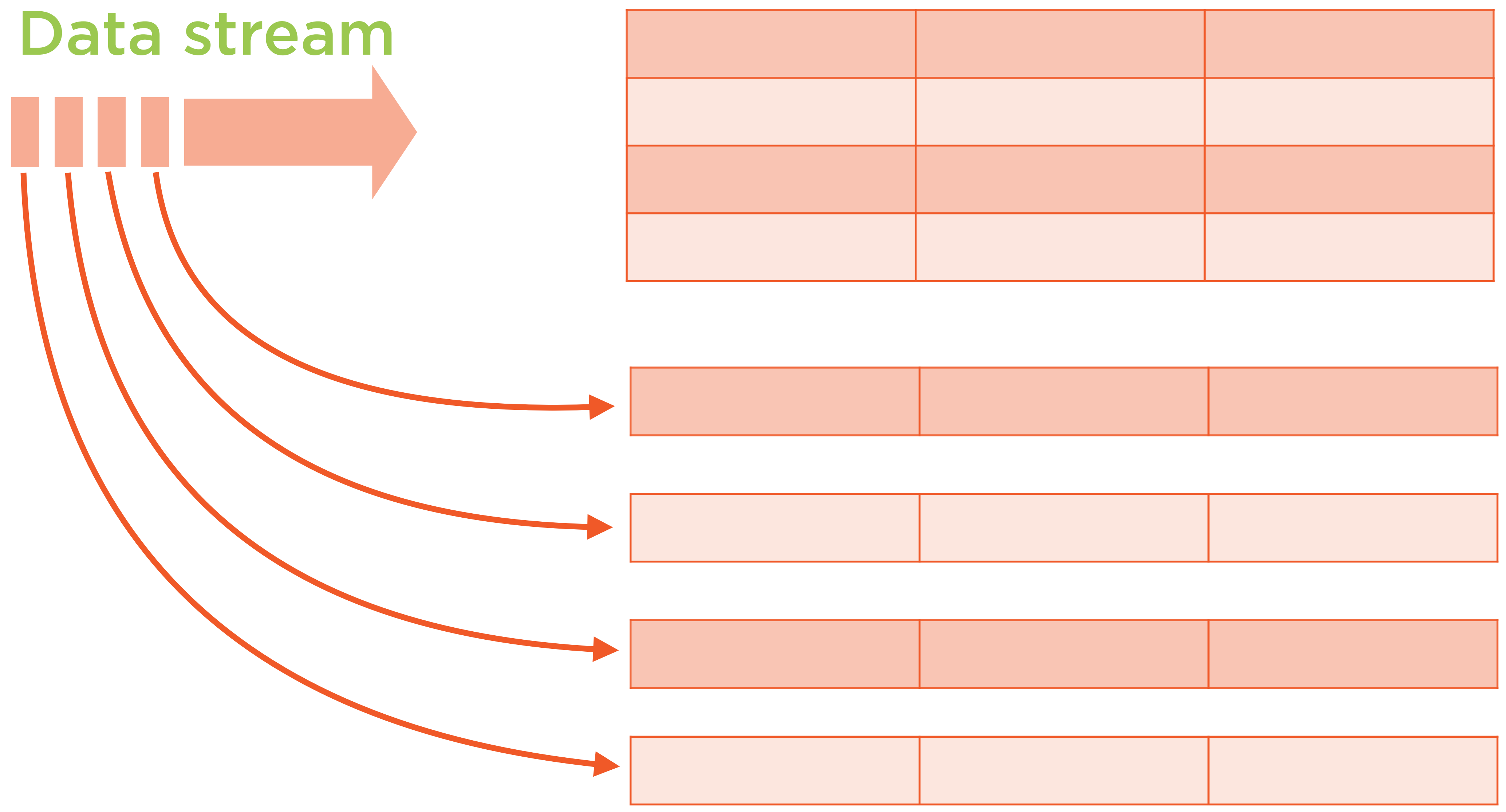
Data stream as an unbounded input table

Batch is Simply A Prefix of Stream



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

Batch is Simply A Prefix of Stream



All operations that can be performed on dataFrames can be performed on the stream

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

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)

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)

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)

Structured 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

Structured Streaming

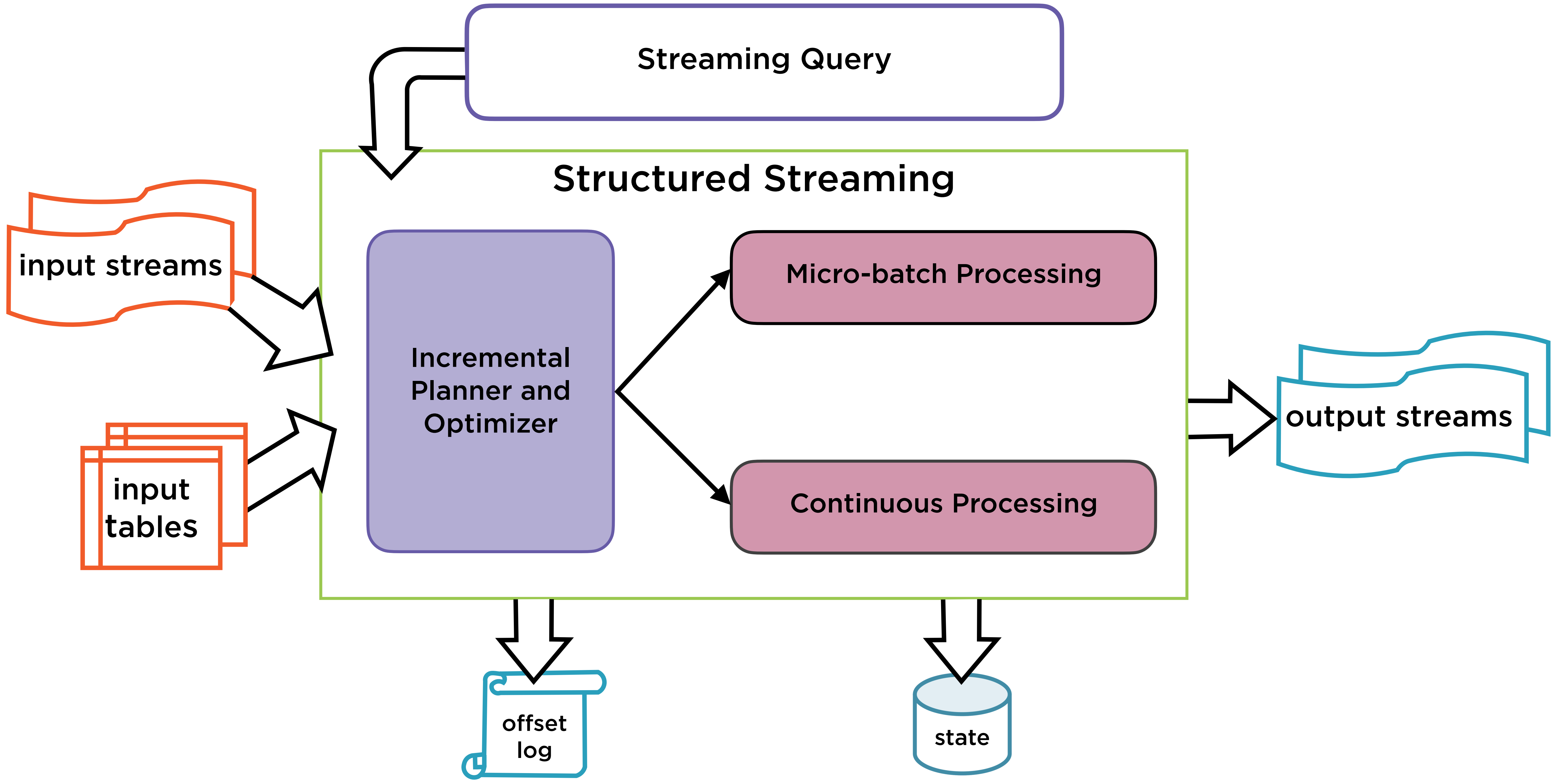
A stream is not a stream

- It is simply an unbounded dataset

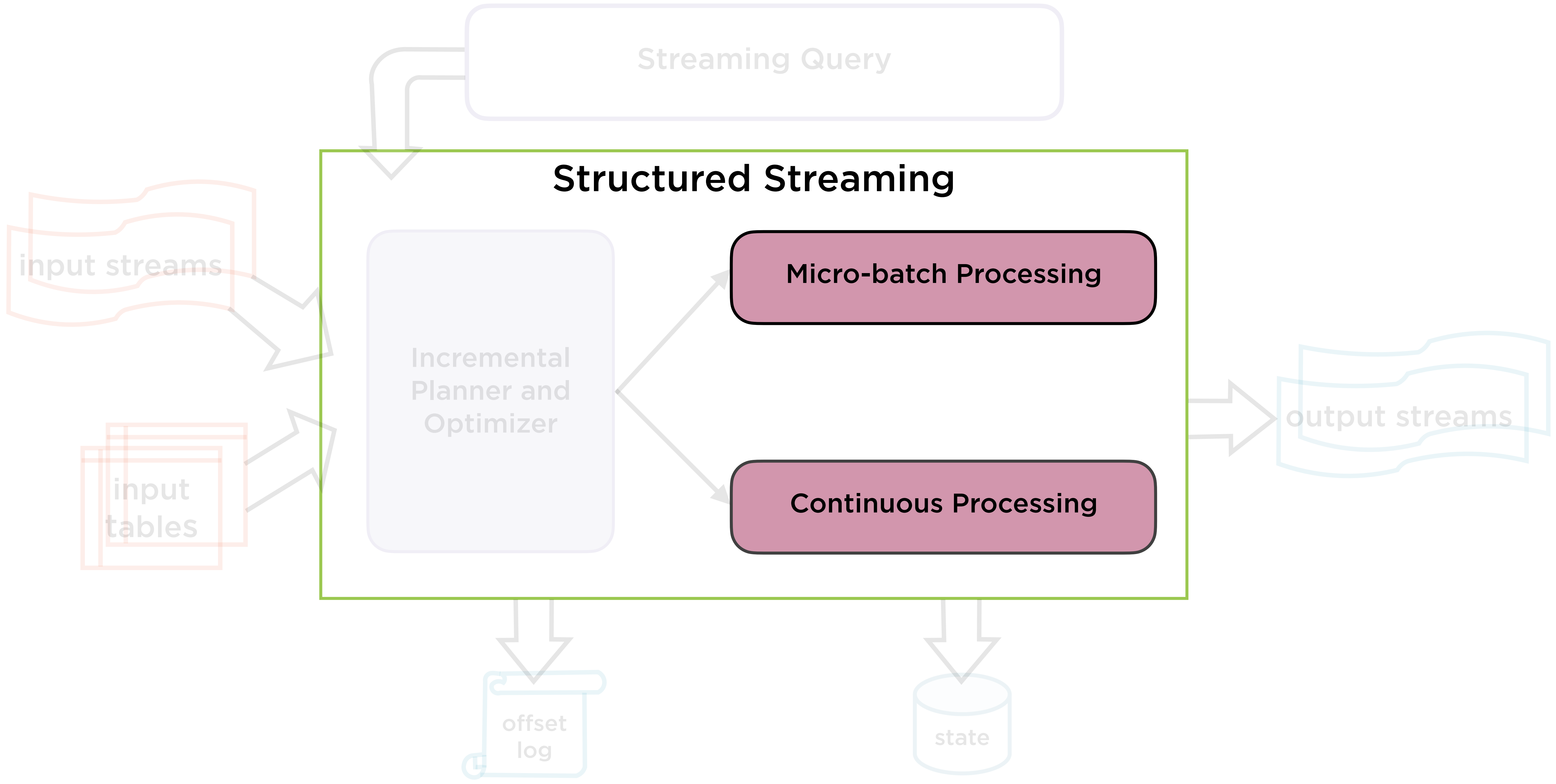
End-to-end application

- Unify batch and streaming pipelines
- With same queries for both

Micro-batch and Continuous Processing



Micro-batch and Continuous Processing



Micro-batch Processing in Spark

Default stream processing mode in Spark

Data streams processed as a series of batch jobs

End-to-end latencies as low as 100ms

Exactly-once fault-tolerance guarantees

Continuous Processing in Spark

Experimental stream processing mode in Spark

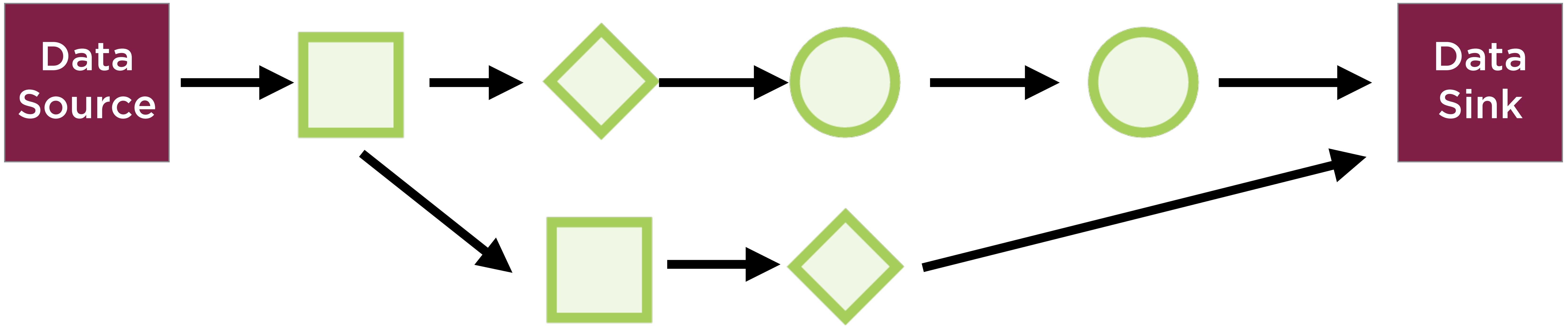
Data streams processed using long-running tasks

End-to-end latencies a few milliseconds

At least-once fault-tolerance guarantees

Built-in Sources and Sinks

Transformations on Streaming Data



Data Source

File source

Kafka source

Socket source

Rate source

File Source

Reads files written in a directory as a stream of data

Files processed in the order of modification time (or reverse order)

Files atomically placed in source directory

CSV, JSON, ORC, Parquet files

Kafka Source

Open source software to handle real-time data feeds

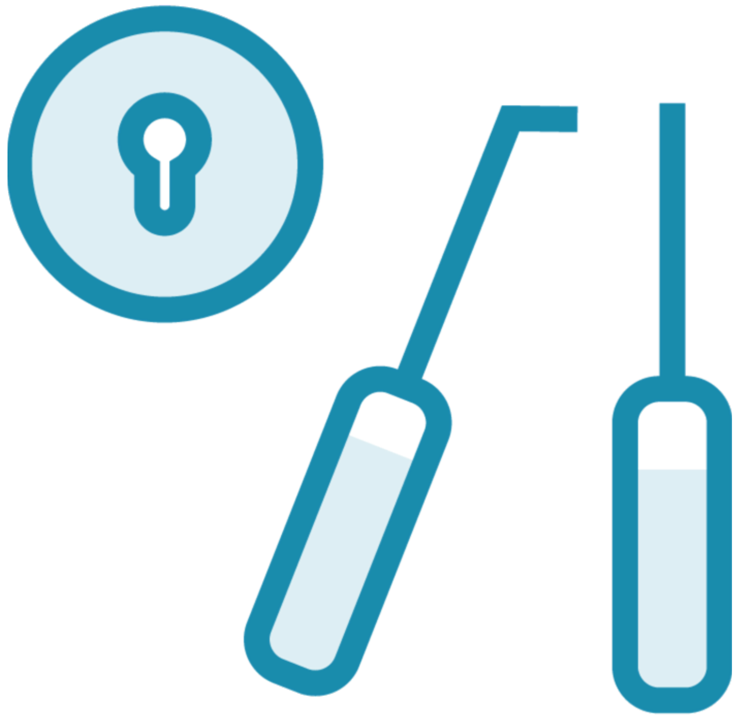
Compatible with Kafka broker version 0.10.0 or higher

Socket Source

Read UTF-8 text data from a socket connection

Test source, does not provide end-to-end fault tolerance guarantees

Rate Source



Generates data at the specified number of rows per second

Each row contains timestamp and value

Can be used for benchmarking streaming application code

Fault Tolerance of Sources

Different types of sources and sinks differ in their fault tolerance

For a source to be fault tolerant, it must be possible to replay data

- Then, can replay all data after checkpoint to restore system state

Fault Tolerance of Sources

Fault tolerant sources

- File sources
- Kafka sources
- Rate sources

Socket sources are not fault tolerant

Data Sink

File sink

Kafka sink

Foreach sink

Console sink

Memory sink

File Sink

Writes files to the specified output directory

Offers exactly-once fault tolerance semantics

Supports the append output mode

Kafka Sink

Writes output to one or more topics in Kafka

Supports at-least-once fault tolerance semantics

Supports the append, complete, and update output mode

Foreach Sink

Custom write logic for results

Perform write operations for each row

Supports at-least-once fault tolerance semantics

Supports the append, complete, and update output mode

ForeachBatch Sink

Custom write logic for results

Perform write operations for each micro-batch processed

Fault tolerance semantics depends on the implementation

Supports the append, complete, and update output mode

Console Sink

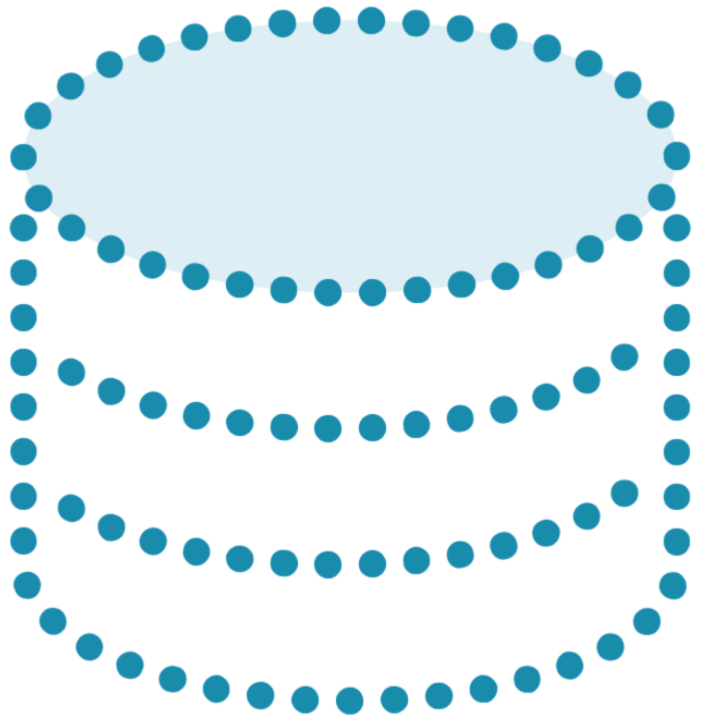
Prints results out to the terminal window

Used for testing and debugging purposes

No fault tolerance guarantees

Supports the append, complete, and update output mode

Memory Sink



Output stored in memory in a table format

Used for testing and debugging purposes

No fault tolerance guarantees

Supports the append and complete output mode

Demo

Exploring the environment set up

Demo

Writing results to the console sink

Demo

Writing results to the file sink using CSV and JSON files

Demo

Writing results using custom logic with the foreach sink

Demo

Writing results using custom logic with the foreachBatch sink

Summary

Understanding Spark DataFrames

Data sources, data sinks, data transformations

Writing data to the console sink, file sink

Customizing write logic using the foreach sink and the foreachBatch sink

Up Next:

Processing Streaming DataFrames