Data Streaming Concepts and Tools

Kristo Raun

Introduction to Near Real-Time Data Analytics

August 2022

UNIVERSITY OF TARTU

1632

Agenda for the week

Kafka

Flink

Visuals

Project

Project

Agenda for the week

Kafka

Flink

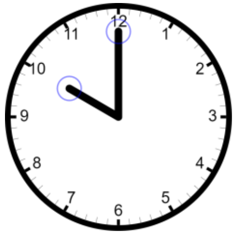
Visuals

Project

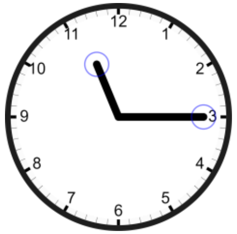
Project

Agenda for today

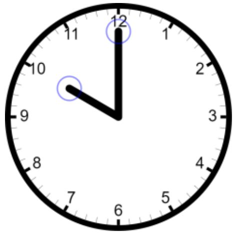
Agenda for today



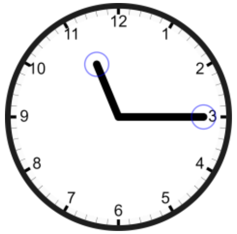
Data Streaming Concepts and Tools

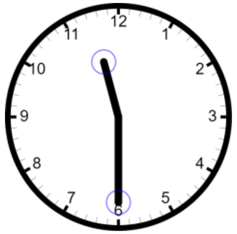


Agenda for today

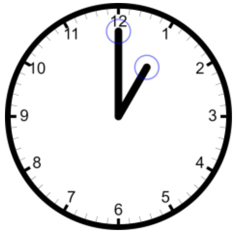


Data Streaming Concepts and Tools

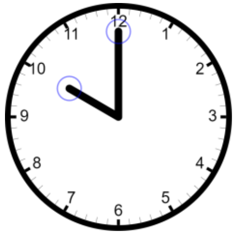




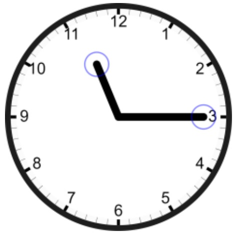
Apache Flink setup, practice

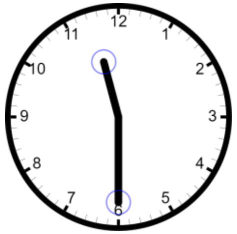


Agenda for today

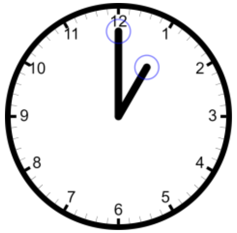


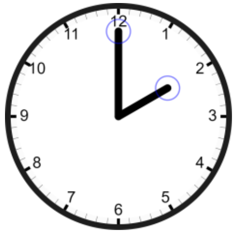
Data Streaming Concepts and Tools



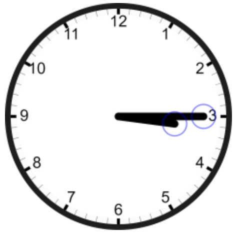


Apache Flink setup, practice



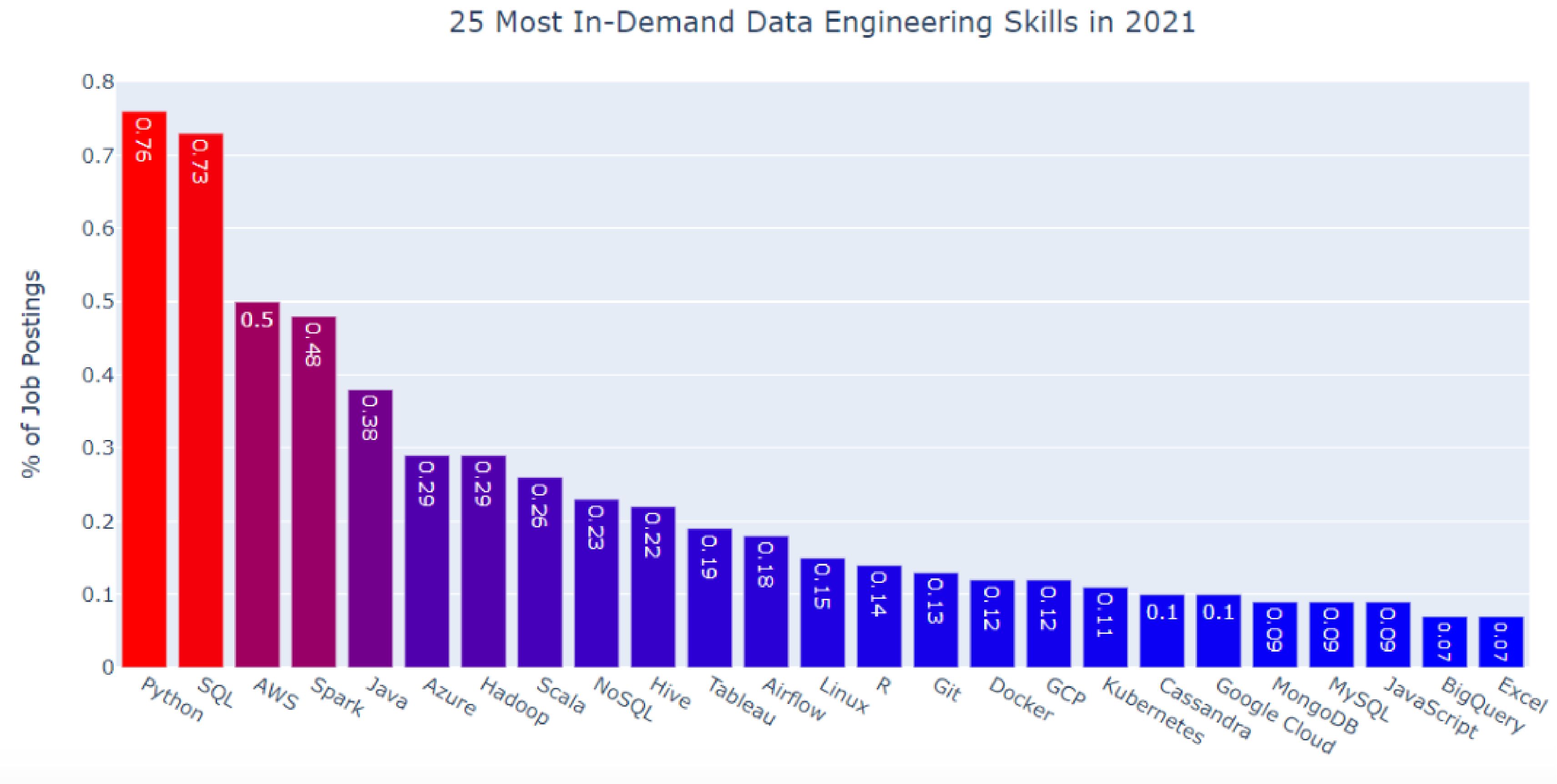


Connecting Apache Flink and Kafka



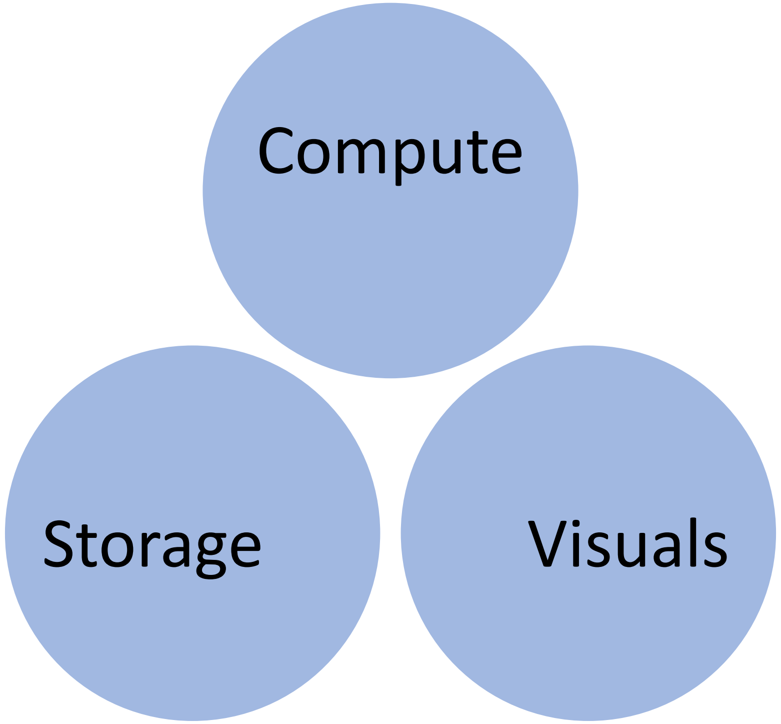
Use the right tool

Expectation



Source: https://www.kdnuggets.com/2021/05/most-demand-skills-data-engineers-2021.html

Classification

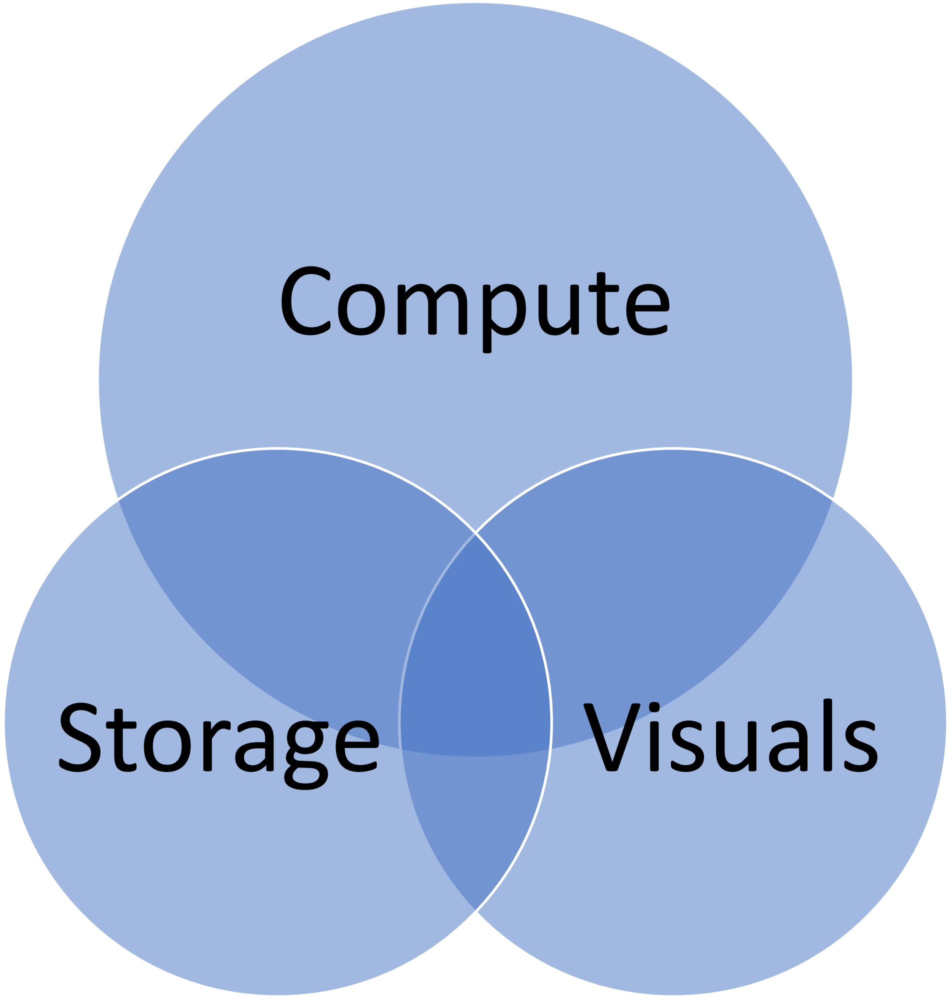


Compute

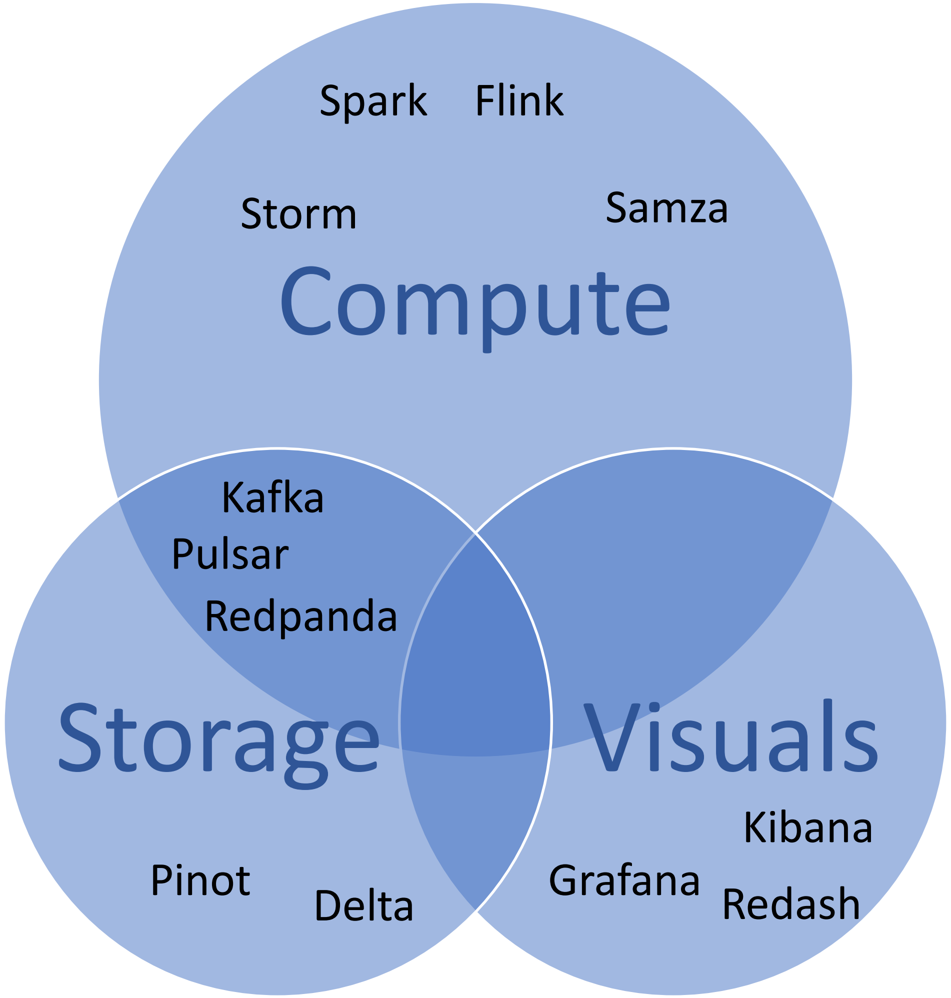
Storage

Visuals

Classification



Classification



Note: arbitrary classification of open-source frameworks

BECAUSE...
IT'S IN THE CLOUD
H HD
RY.COM
memegenerator.net

aws

Microsoft Azure

Google Cloud Platform

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink
Glue	Data Factory	Cloud Data Fusion	ETL

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink
Glue	Data Factory	Cloud Data Fusion	ETL
EMR	HDInsight	Dataproc	Managed open-source (Hadoop)

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink
Glue	Data Factory	Cloud Data Fusion	ETL
EMR	HDInsight	Dataproc	Managed open-source (Hadoop)
Redshift	Synapse	BigQuery	Data warehouse

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink
Glue	Data Factory	Cloud Data Fusion	ETL
EMR	HDInsight	Dataproc	Managed open-source (Hadoop)
Redshift	Synapse	BigQuery	Data warehouse
S3	Blob storage	Cloud storage	Storage / Data lake

aws	Microsoft Azure	Google Cloud Platform	
MSK	Event Hubs	Pub/Sub	Kafka
Kinesis	Stream Analytics	Dataflow	Flink
Glue	Data Factory	Cloud Data Fusion	ETL
EMR	HDInsight	Dataproc	Managed open-source (Hadoop)
Redshift	Synapse	BigQuery	Data warehouse
S3	Blob storage	Cloud storage	Storage / Data lake
Quicksight	Power BI	Looker, Data Studio	Visualization

Cloud agnostic / SaaS vendors

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda

Kafka

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink
Fivetran | Talend | Hevo | Matillion | …	ETL

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink
Fivetran | Talend | Hevo | Matillion | …	ETL
Databricks | Cloudera	Managed open-source (Hadoop)

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink
Fivetran | Talend | Hevo | Matillion | …	ETL
Databricks | Cloudera	Managed open-source (Hadoop)
Snowflake | Vertica | Databricks	Data warehouse

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink
Fivetran | Talend | Hevo | Matillion | …	ETL
Databricks | Cloudera	Managed open-source (Hadoop)
Snowflake | Vertica | Databricks	Data warehouse
Databricks | Snowflake	Storage / Data lake

Note: arbitrary incomplete list

Cloud agnostic / SaaS vendors

Confluent | Conduktor | Redpanda	Kafka
Ververica | Aiven	Flink
Fivetran | Talend | Hevo | Matillion | …	ETL
Databricks | Cloudera	Managed open-source (Hadoop)
Snowflake | Vertica | Databricks	Data warehouse
Databricks | Snowflake	Storage / Data lake
Qlik | Tableau	Visualization

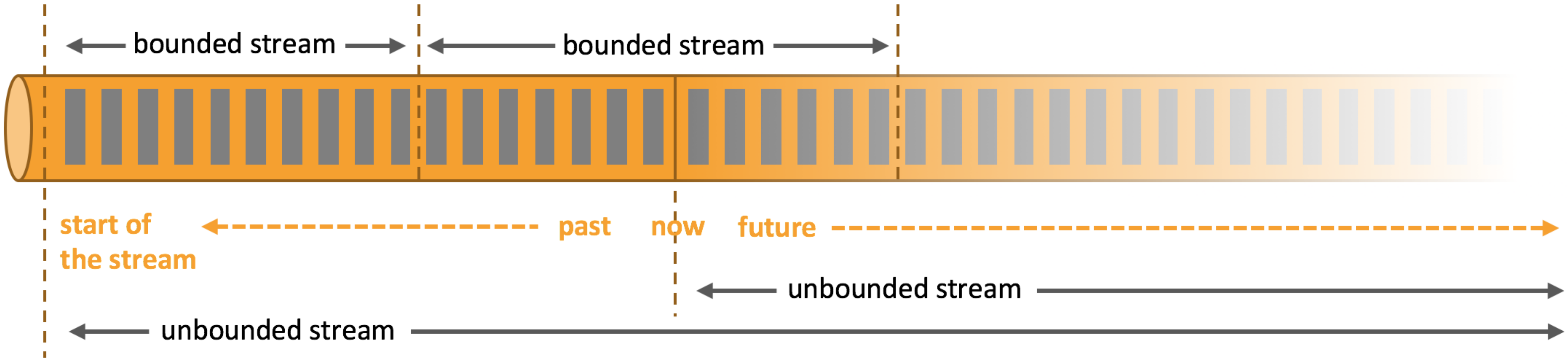
Note: arbitrary incomplete list

Community/trial versions to try out

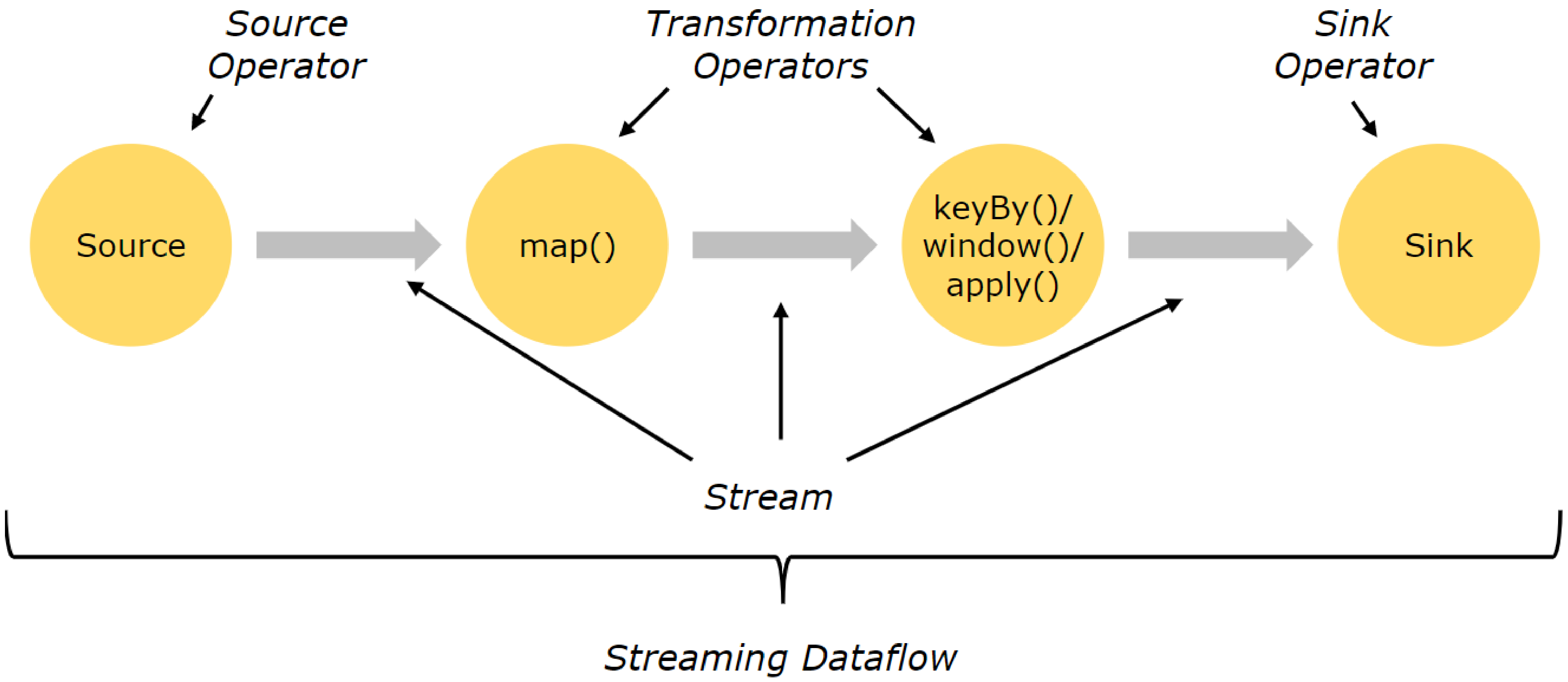
- https://confluent.cloud/signup
  - 60 days
- https://databricks.com/try-databricks
  - Free (forever?)
- https://www.ververica.com/getting-started
  - Free, need to install on K8s cluster

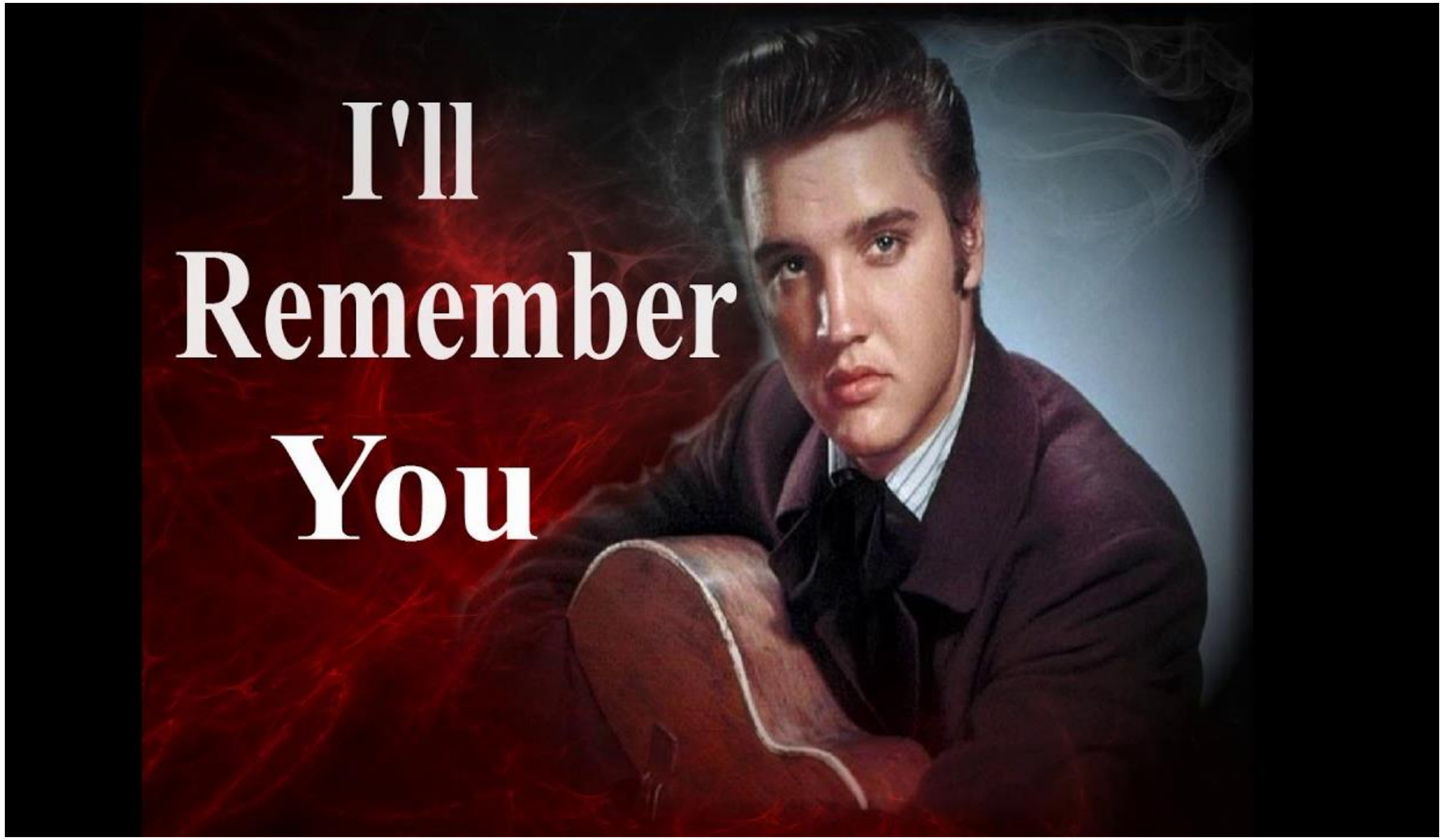
Flink

Flink


bounded stream
bounded stream
start of
the stream
past
now
future
unbounded stream
unbounded stream

Streaming Dataflow





Stateful streaming

- One individual *event at a time* (for example an event parser)

Vs

- Remember information across multiple events (for example window operators). = **stateful**

Examples of stateful operations:

- When an application searches for certain event patterns, the state will store the sequence of events encountered so far.
- When aggregating events per minute/hour/day, the state holds the pending aggregates.
- When training a machine learning model over a stream of data points, the state holds the current version of the model parameters.
- When historic data needs to be managed, the state allows efficient access to events that occurred in the past.

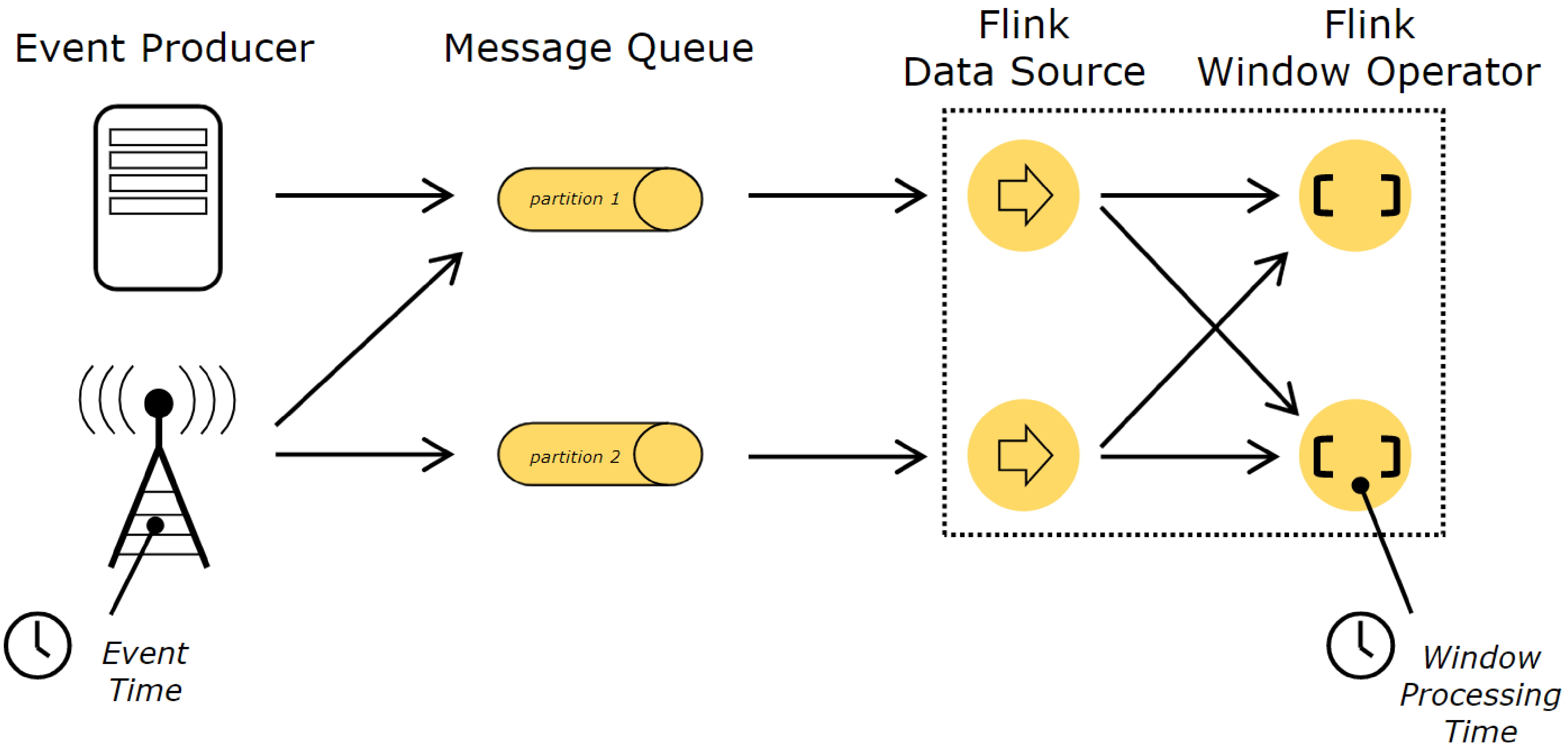
What time is it?

What time is it?

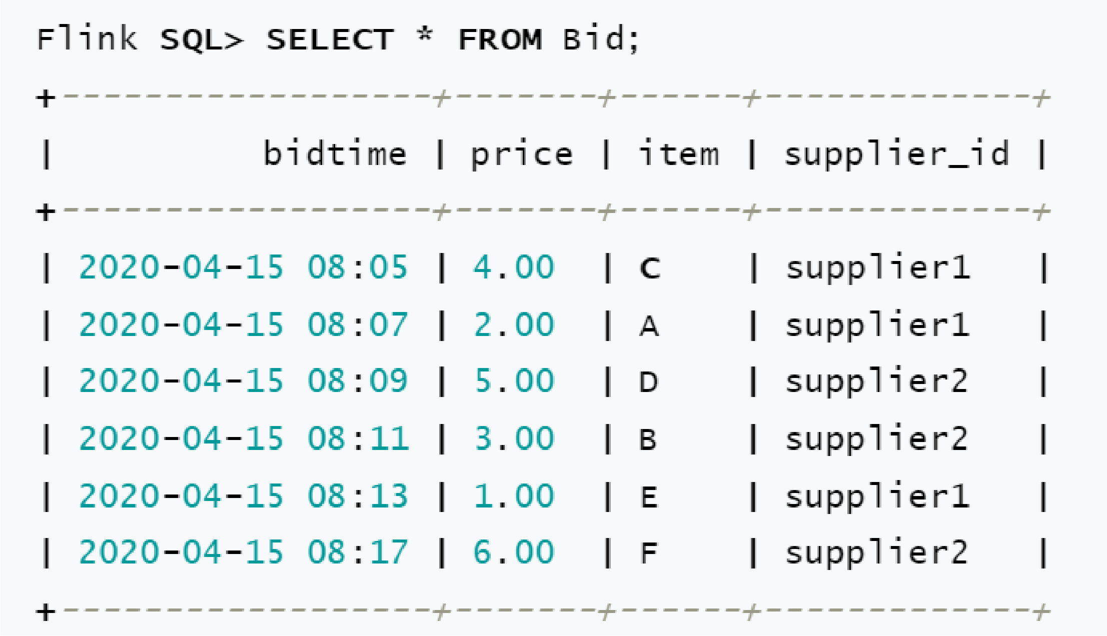
Processing time: system time of the machine that is executing the respective operation.

Event time: the time that each individual event occurred on its producing device.

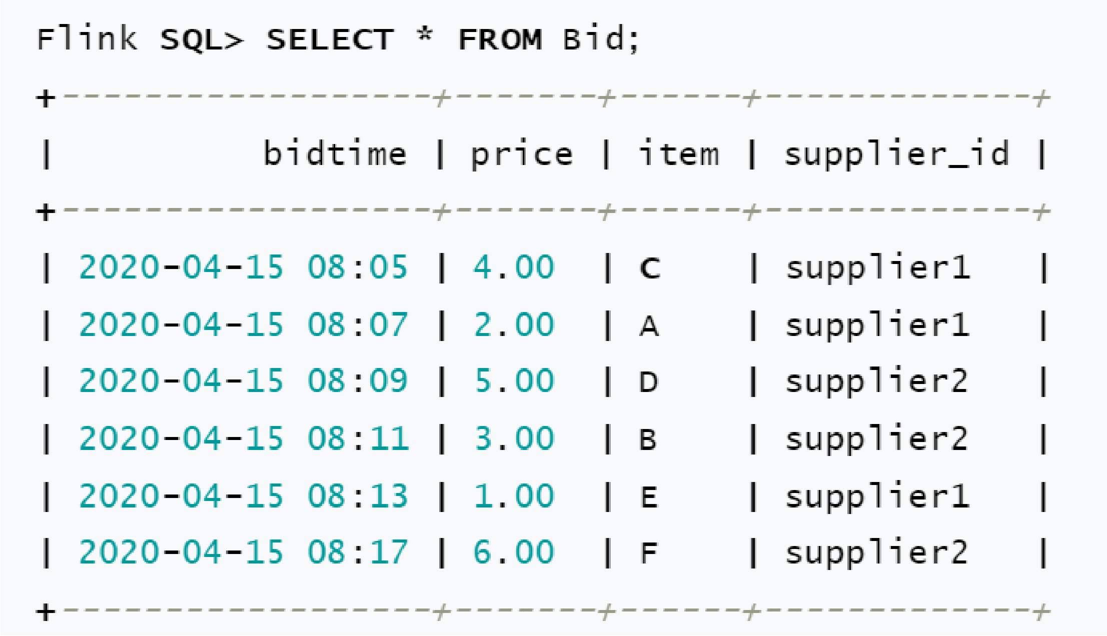
What time is it?



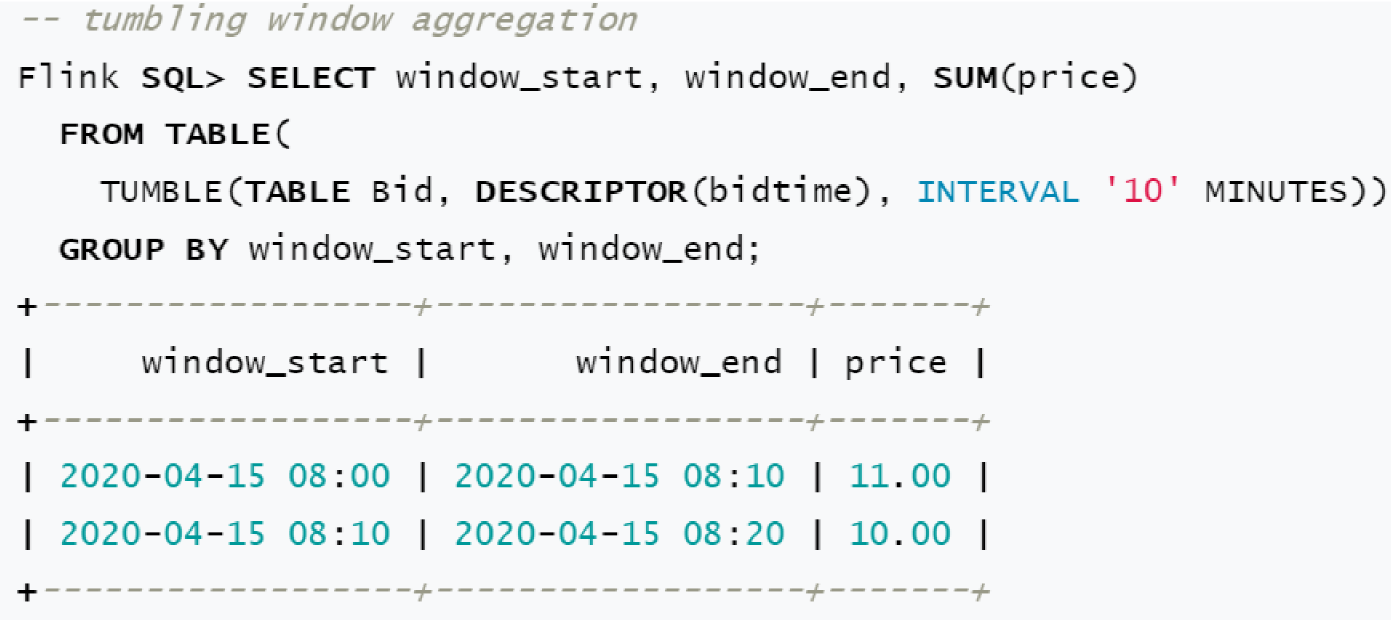
Windows



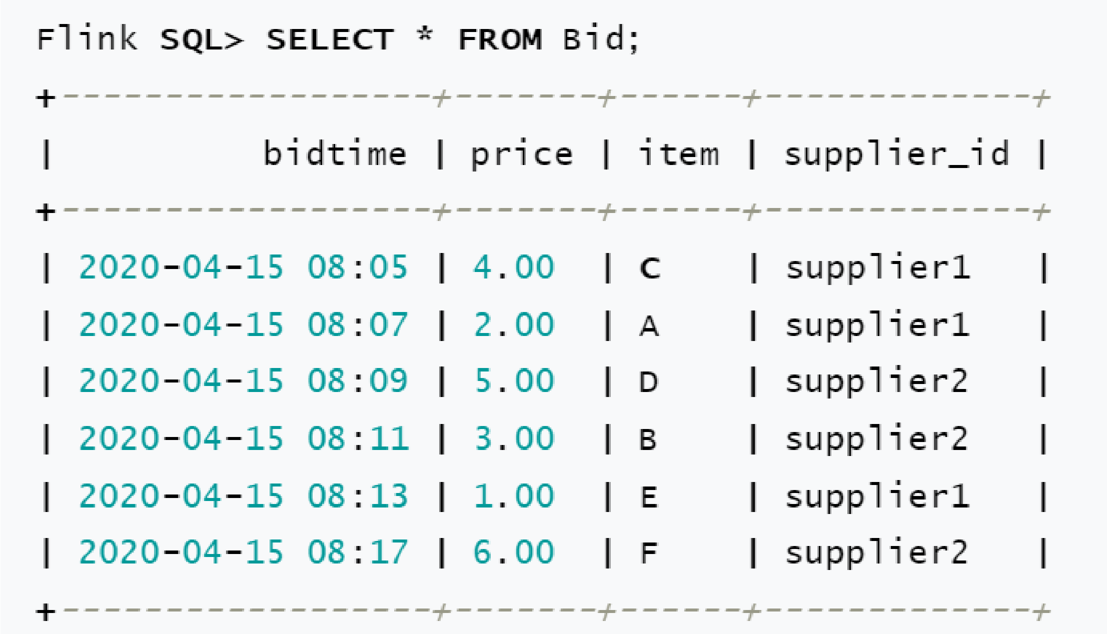
```
Flink SQL> SELECT * FROM Bid;
+------------------+-------+------+-------------+
|          bidtime | price | item | supplier_id |
+------------------+-------+------+-------------+
| 2020-04-15 08:05 | 4.00  | C    | supplier1   |
| 2020-04-15 08:07 | 2.00  | A    | supplier1   |
| 2020-04-15 08:09 | 5.00  | D    | supplier2   |
| 2020-04-15 08:11 | 3.00  | B    | supplier2   |
| 2020-04-15 08:13 | 1.00  | E    | supplier1   |
| 2020-04-15 08:17 | 6.00  | F    | supplier2   |
+------------------+-------+------+-------------+
```



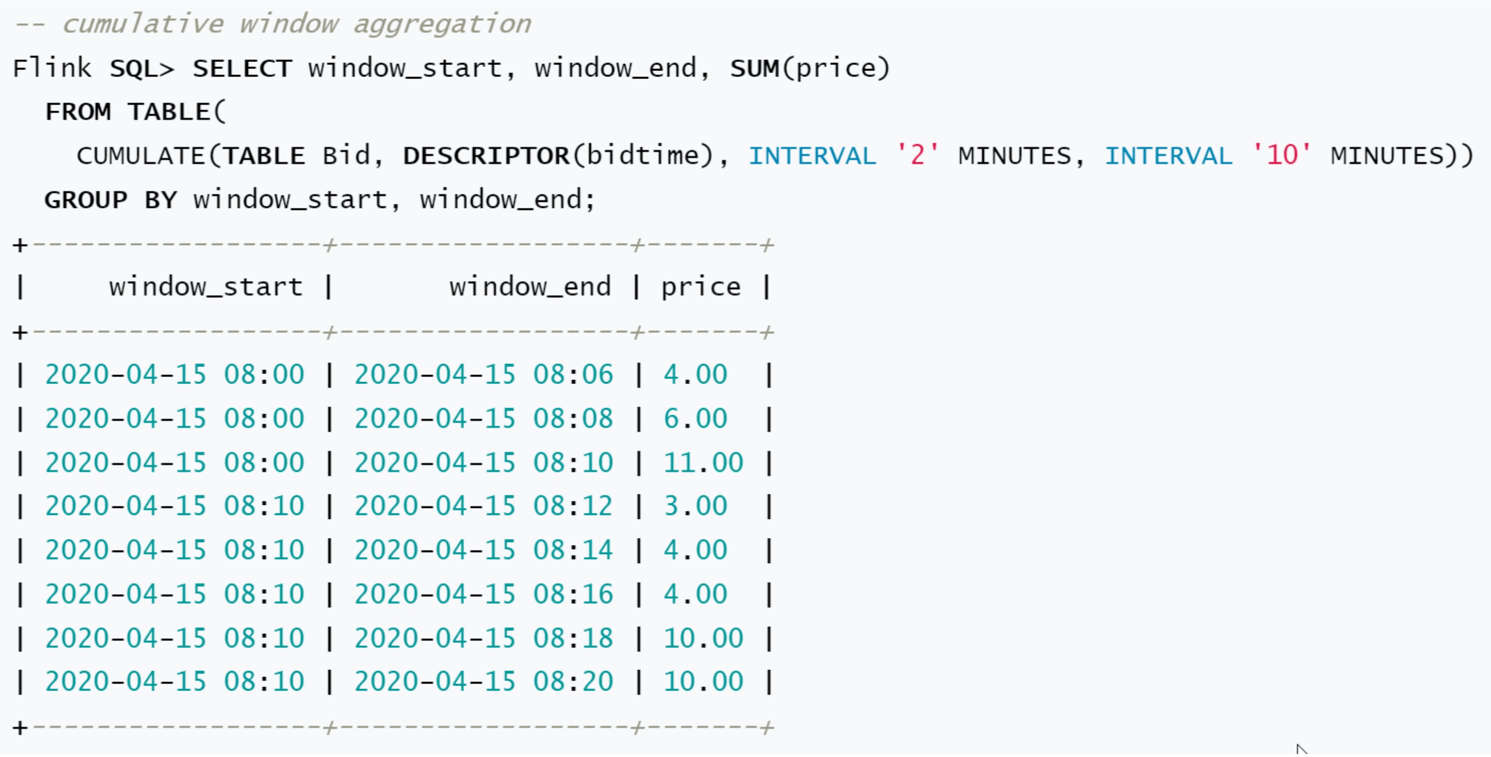
```
Flink SQL> SELECT * FROM Bid;
+------------------+-------+------+-------------+
|          bidtime | price | item | supplier_id |
+------------------+-------+------+-------------+
| 2020-04-15 08:05 | 4.00  | C    | supplier1   |
| 2020-04-15 08:07 | 2.00  | A    | supplier1   |
| 2020-04-15 08:09 | 5.00  | D    | supplier2   |
| 2020-04-15 08:11 | 3.00  | B    | supplier2   |
| 2020-04-15 08:13 | 1.00  | E    | supplier1   |
| 2020-04-15 08:17 | 6.00  | F    | supplier2   |
+------------------+-------+------+-------------+
```



```
-- tumbling window aggregation
Flink SQL> SELECT window_start, window_end, SUM(price)
  FROM TABLE(
    TUMBLE(TABLE Bid, DESCRIPTOR(bidtime), INTERVAL '10' MINUTES))
  GROUP BY window_start, window_end;
+------------------+------------------+-------+
|     window_start |       window_end | price |
+------------------+------------------+-------+
| 2020-04-15 08:00 | 2020-04-15 08:10 | 11.00 |
| 2020-04-15 08:10 | 2020-04-15 08:20 | 10.00 |
+------------------+------------------+-------+
```



```
Flink SQL> SELECT * FROM Bid;
+------------------+-------+------+-------------+
|          bidtime | price | item | supplier_id |
+------------------+-------+------+-------------+
| 2020-04-15 08:05 | 4.00  | C    | supplier1   |
| 2020-04-15 08:07 | 2.00  | A    | supplier1   |
| 2020-04-15 08:09 | 5.00  | D    | supplier2   |
| 2020-04-15 08:11 | 3.00  | B    | supplier2   |
| 2020-04-15 08:13 | 1.00  | E    | supplier1   |
| 2020-04-15 08:17 | 6.00  | F    | supplier2   |
+------------------+-------+------+-------------+
```



```
-- cumulative window aggregation
Flink SQL> SELECT window_start, window_end, SUM(price)
  FROM TABLE(
    CUMULATE(TABLE Bid, DESCRIPTOR(bidtime), INTERVAL '2' MINUTES, INTERVAL '10' MINUTES))
  GROUP BY window_start, window_end;
+------------------+------------------+-------+
|     window_start |       window_end | price |
+------------------+------------------+-------+
| 2020-04-15 08:00 | 2020-04-15 08:06 | 4.00  |
| 2020-04-15 08:00 | 2020-04-15 08:08 | 6.00  |
| 2020-04-15 08:00 | 2020-04-15 08:10 | 11.00 |
| 2020-04-15 08:10 | 2020-04-15 08:12 | 3.00  |
| 2020-04-15 08:10 | 2020-04-15 08:14 | 4.00  |
| 2020-04-15 08:10 | 2020-04-15 08:16 | 4.00  |
| 2020-04-15 08:10 | 2020-04-15 08:18 | 10.00 |
| 2020-04-15 08:10 | 2020-04-15 08:20 | 10.00 |
+------------------+------------------+-------+
```

THE PORTER
This isn't going to stop, is it?

Watermark

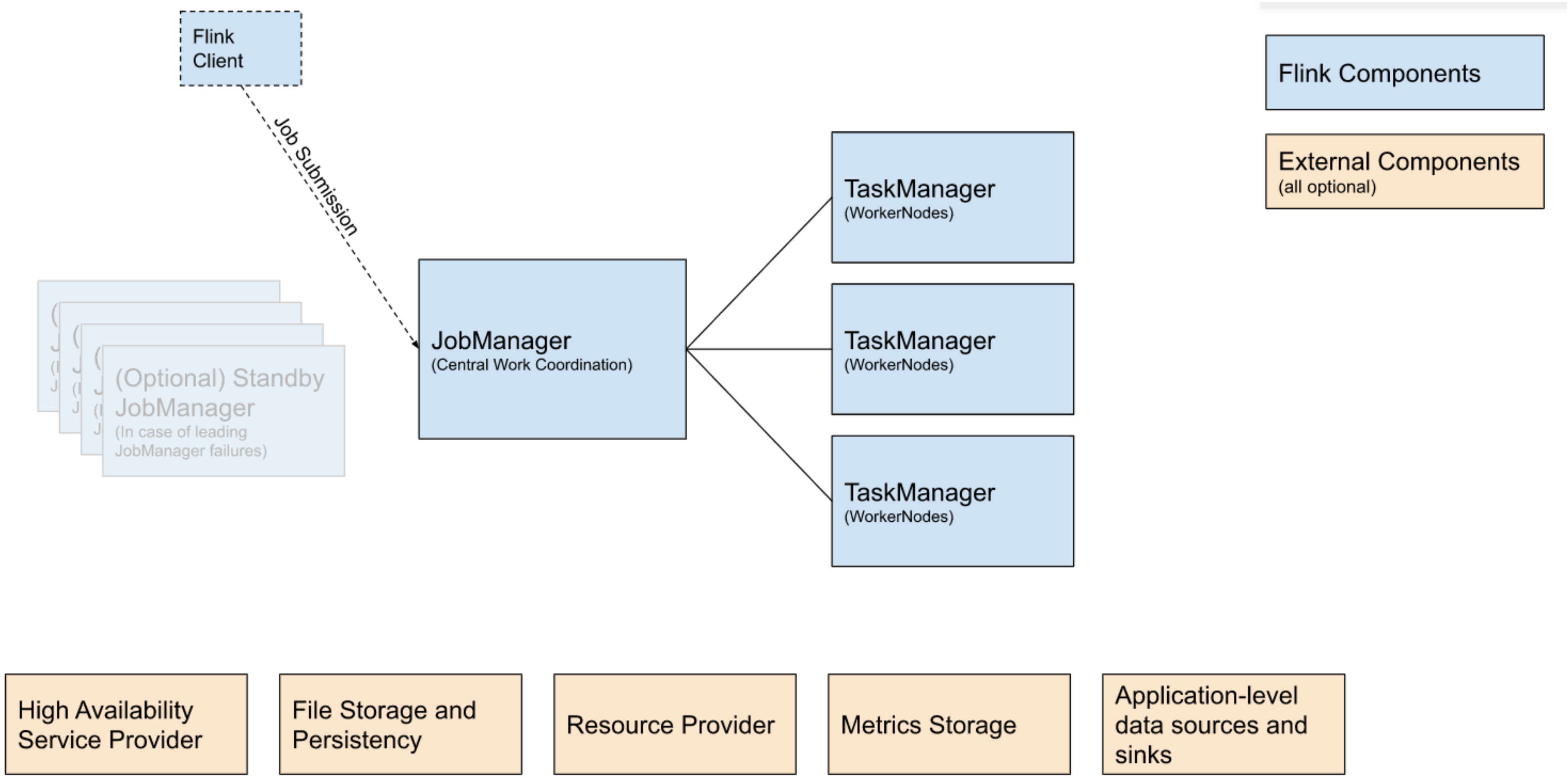
- A *Watermark(t)* declares that event time has reached time *t* in that stream, meaning that there should be no more elements from the stream with a timestamp *t’* <= *t* (i.e. events with timestamps older or equal to the watermark).

Watermark example

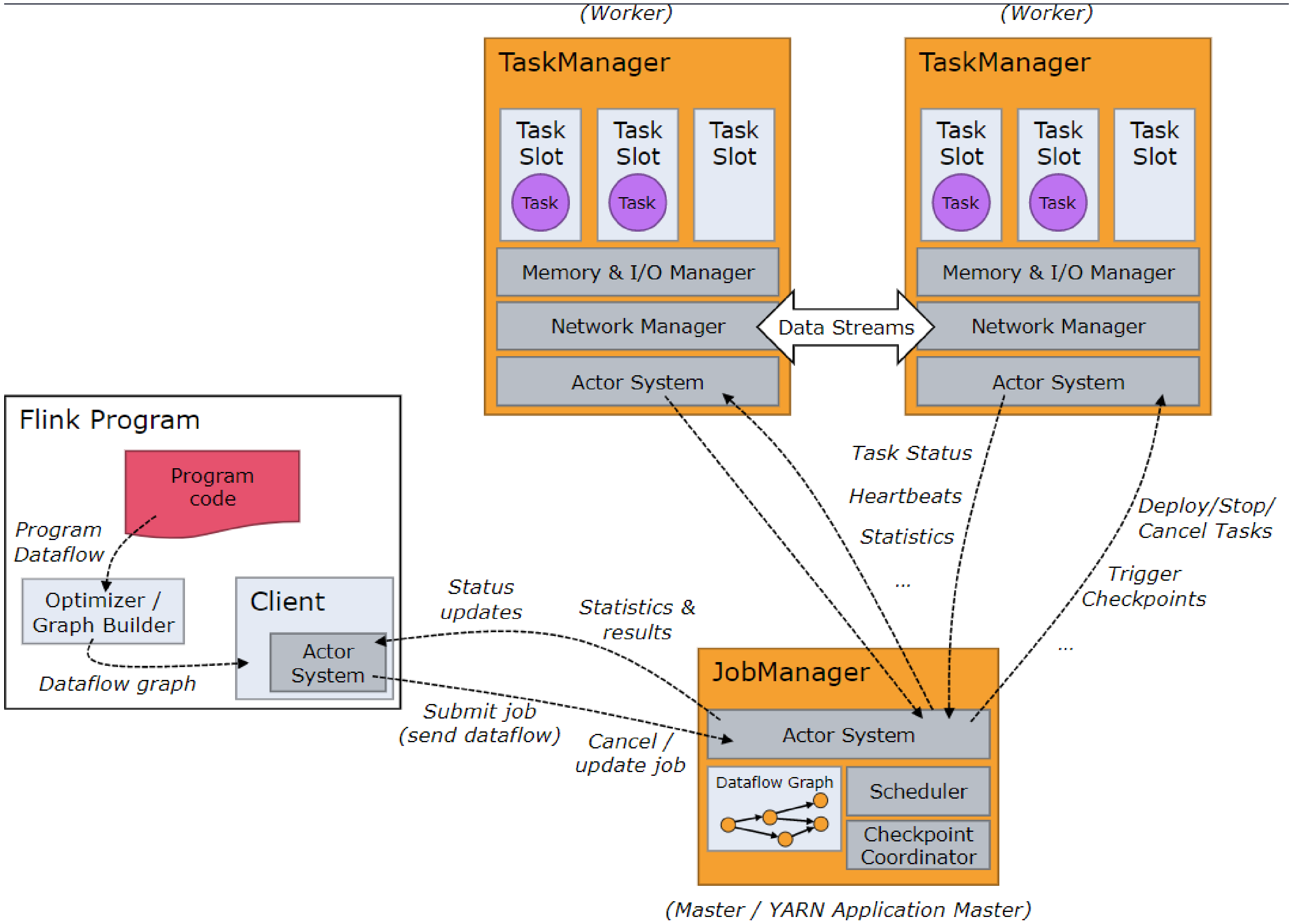
Watermark vs window

Watermark	Window
Progress of event time	Logical partition of events
Based on time	Based on time or count
Goal: Determine when is a portion of data processed, Handle out-of-order streams	Goal: Aggregations

Flink architecture

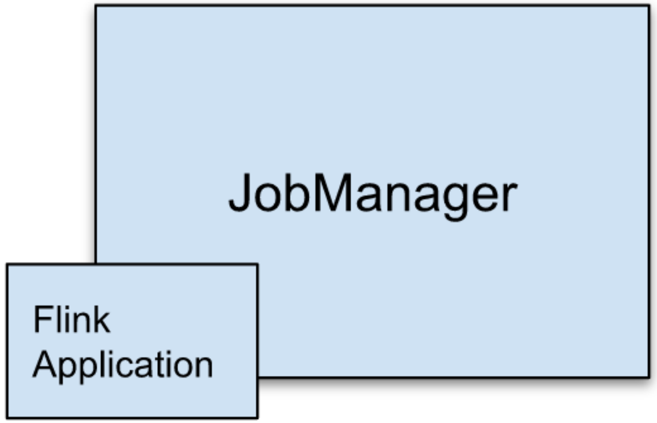

Flink Client
Job Submission
(Optional) Standby JobManager
(In case of leading JobManager failures)
JobManager
(Central Work Coordination)
TaskManager
(WorkerNodes)
TaskManager
(WorkerNodes)
TaskManager
(WorkerNodes)
Flink Components
External Components
(all optional)
High Availability Service Provider
File Storage and Persistency
Resource Provider
Metrics Storage
Application-level data sources and sinks

Flink architecture



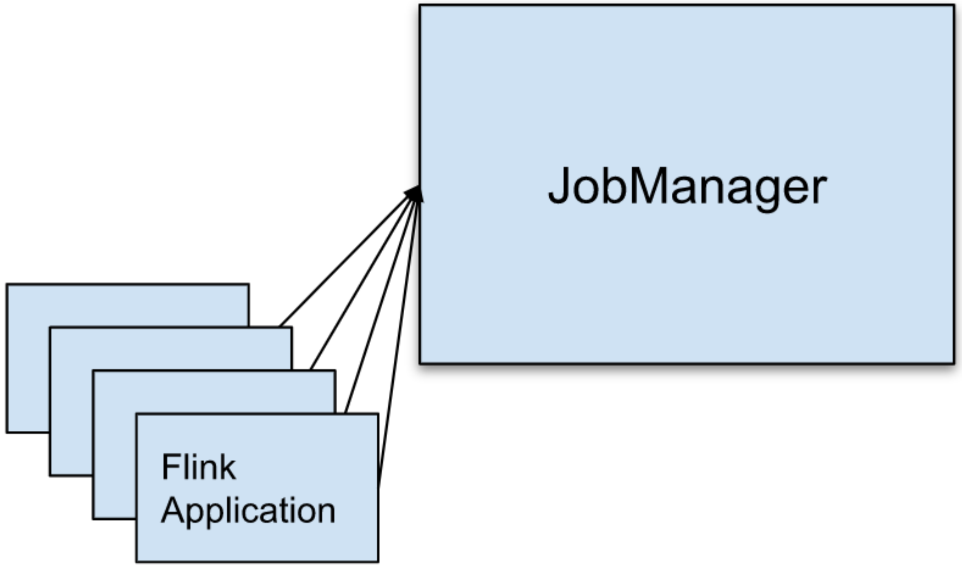
Flink Deployment

Application Mode



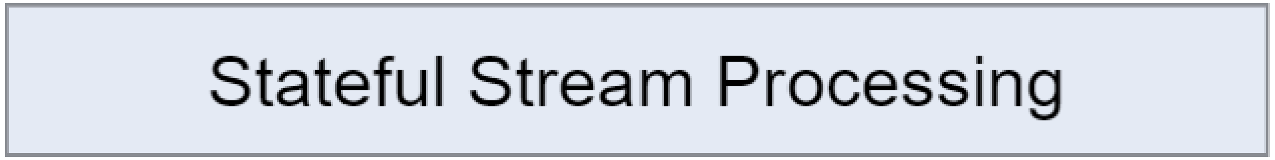
A dedicated JobManager is started for submitting the job. The JobManager will only execute this job, then exit.
The Flink Application runs on the JobManager.

Session Mode



Multiple jobs share one JobManager.

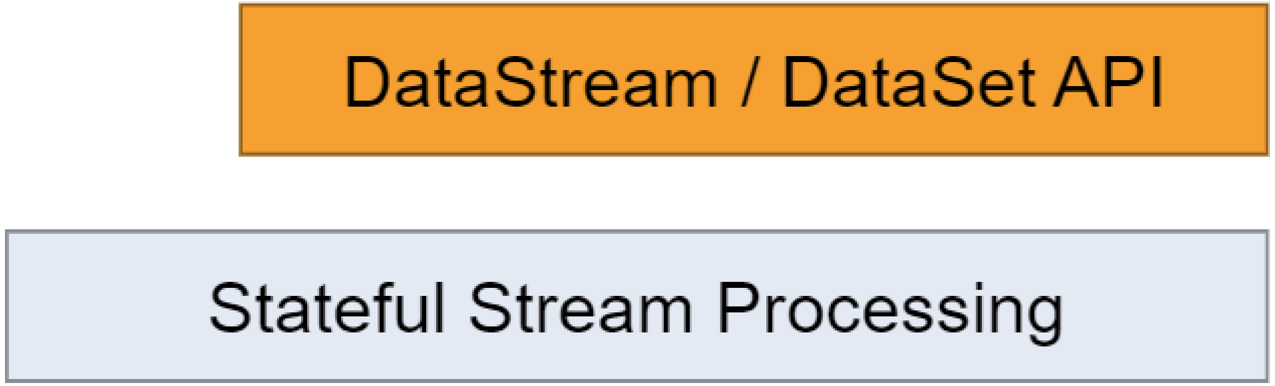
Flink APIs



Stateful Stream Processing

Low-level building block
(streams, state, [event] time)

Flink APIs



DataStream / DataSet API

Core APIs

Stateful Stream Processing

Low-level building block
(streams, state, [event] time)

Flink APIs

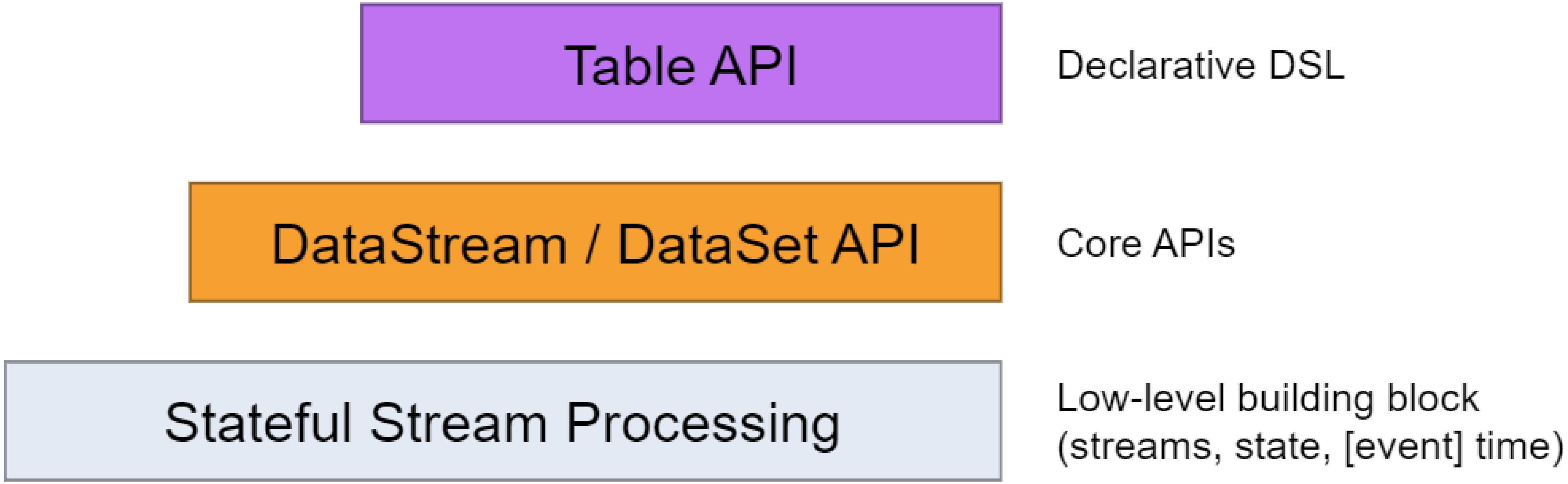


Table API

Declarative DSL

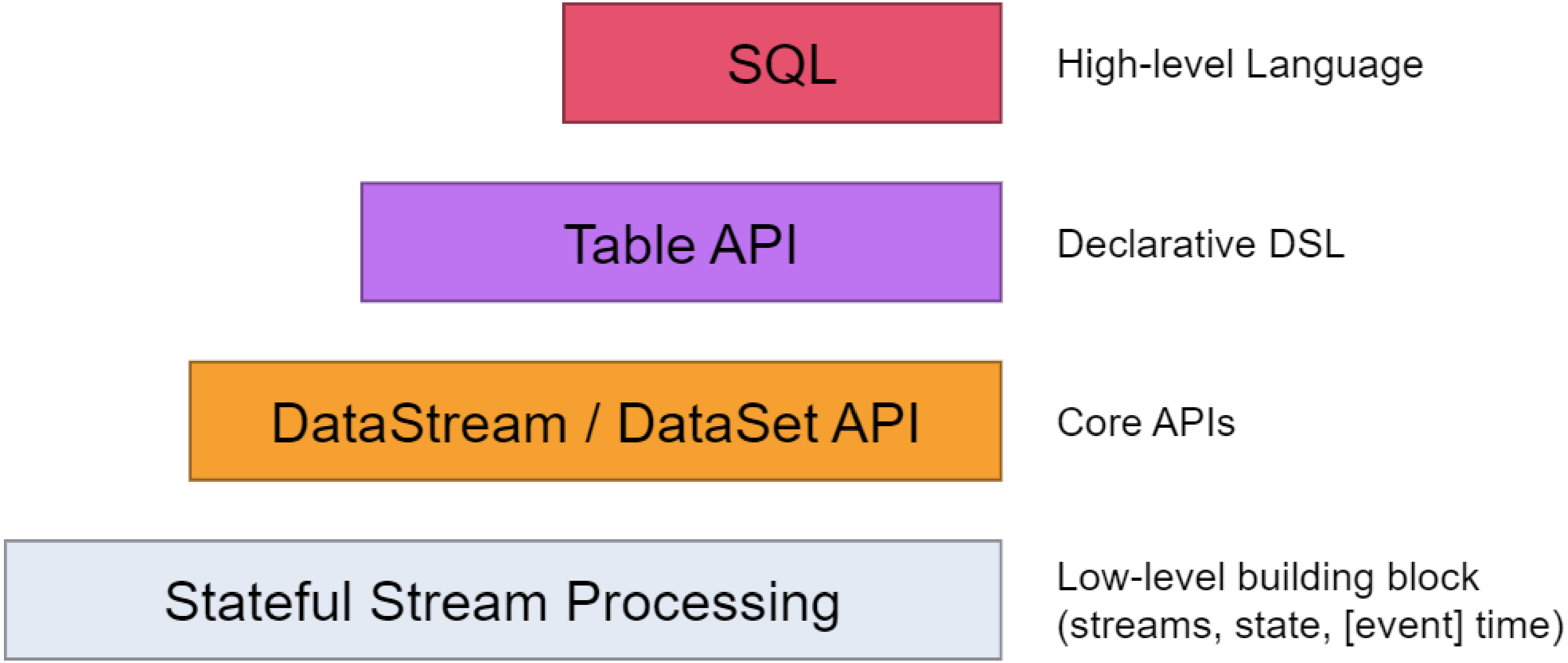
DataStream / DataSet API

Core APIs

Stateful Stream Processing

Low-level building block
(streams, state, [event] time)

Flink APIs



SQL

High-level Language

Table API

Declarative DSL

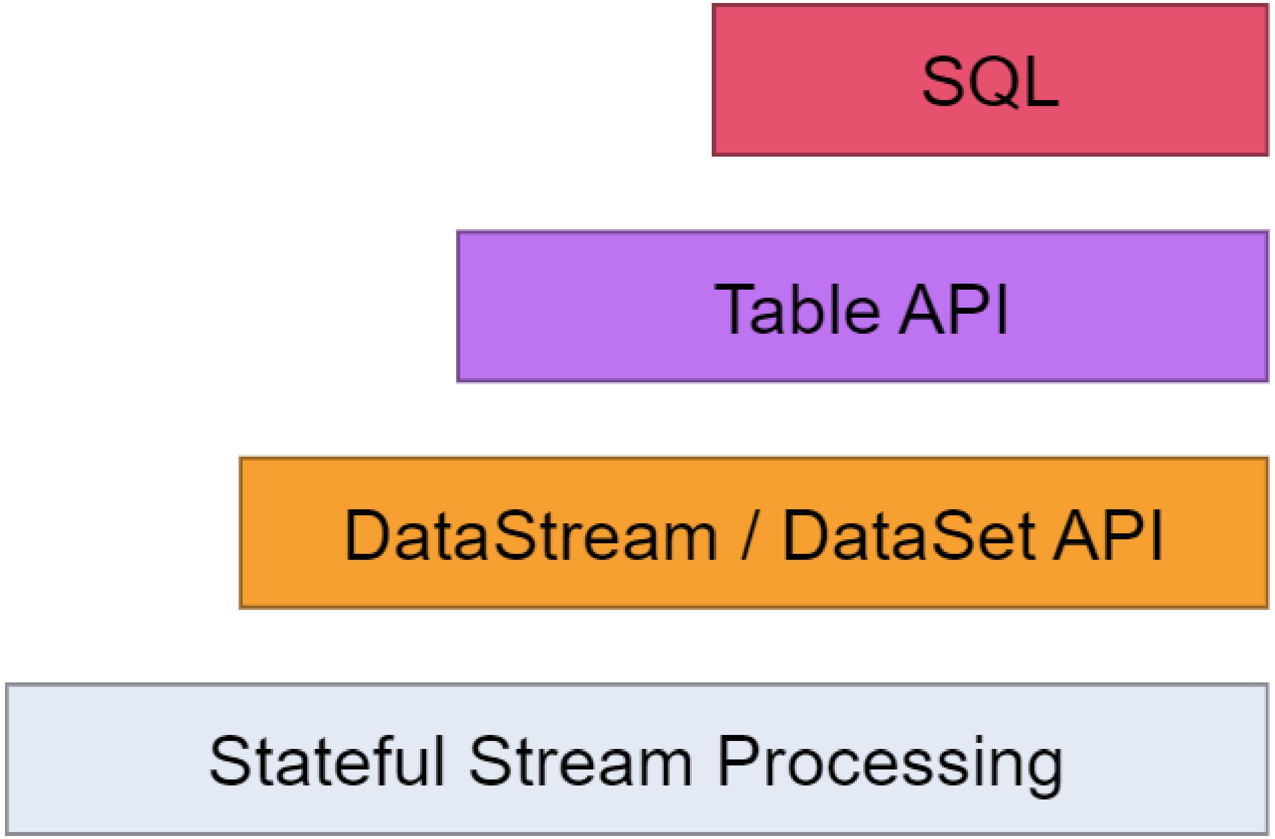
DataStream / DataSet API

Core APIs

Stateful Stream Processing

Low-level building block
(streams, state, [event] time)

Flink APIs



SQL	High-level Language	*SQL*
Table API	Declarative DSL	*Python + SQL*
DataStream / DataSet API	Core APIs	*Pandas / PySpark*
Stateful Stream Processing	Low-level building block (streams, state, [event] time)	

LETS ROLL
#silverdollarcity
SILVER DOLLAR CITY