Apache Flink Tutorial

DataStream API

Agenda

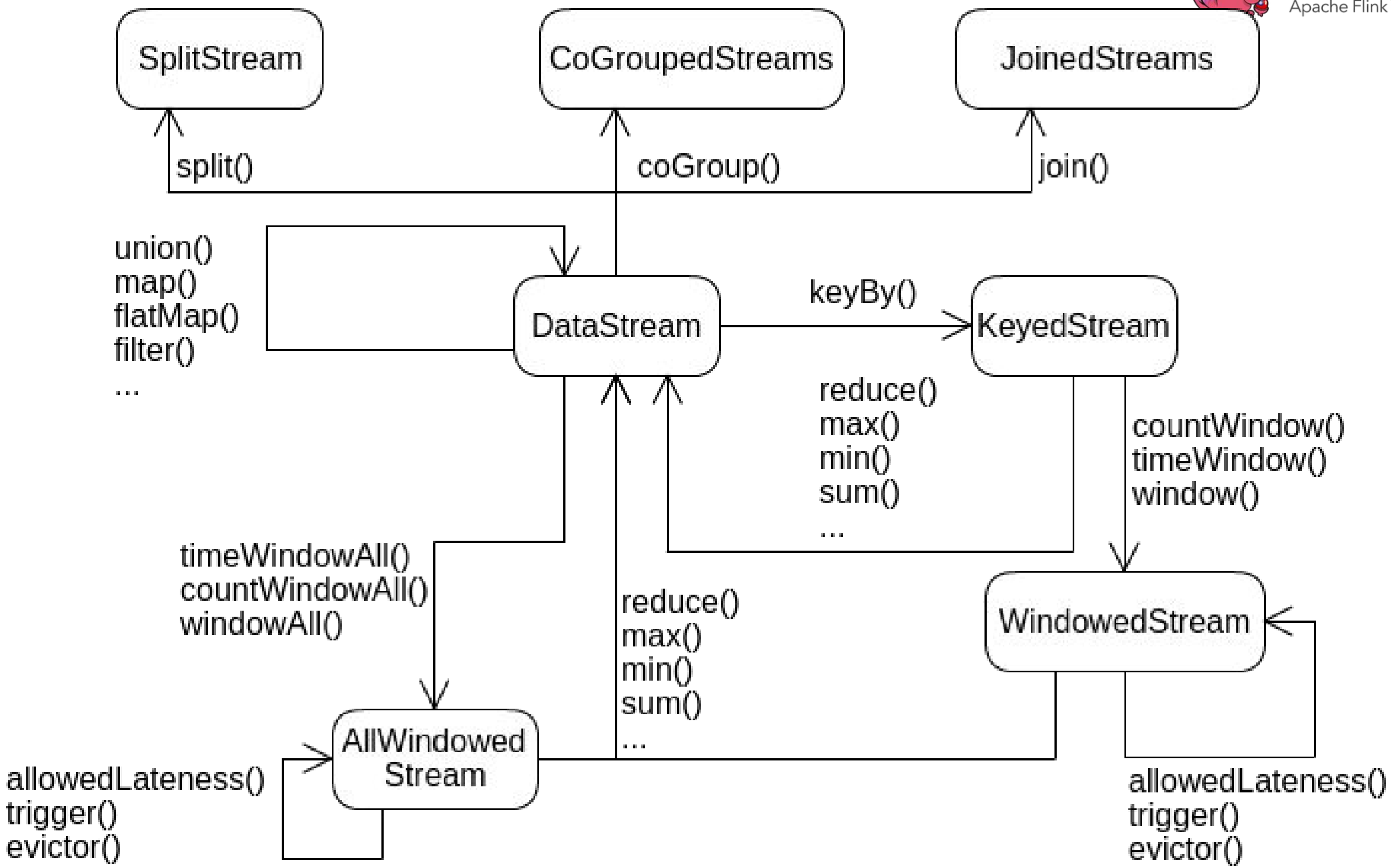
- Basic structure of a streaming program
- Overview of various data streams
- Time characteristics
- Windows
- Window Functions

Basic Structure

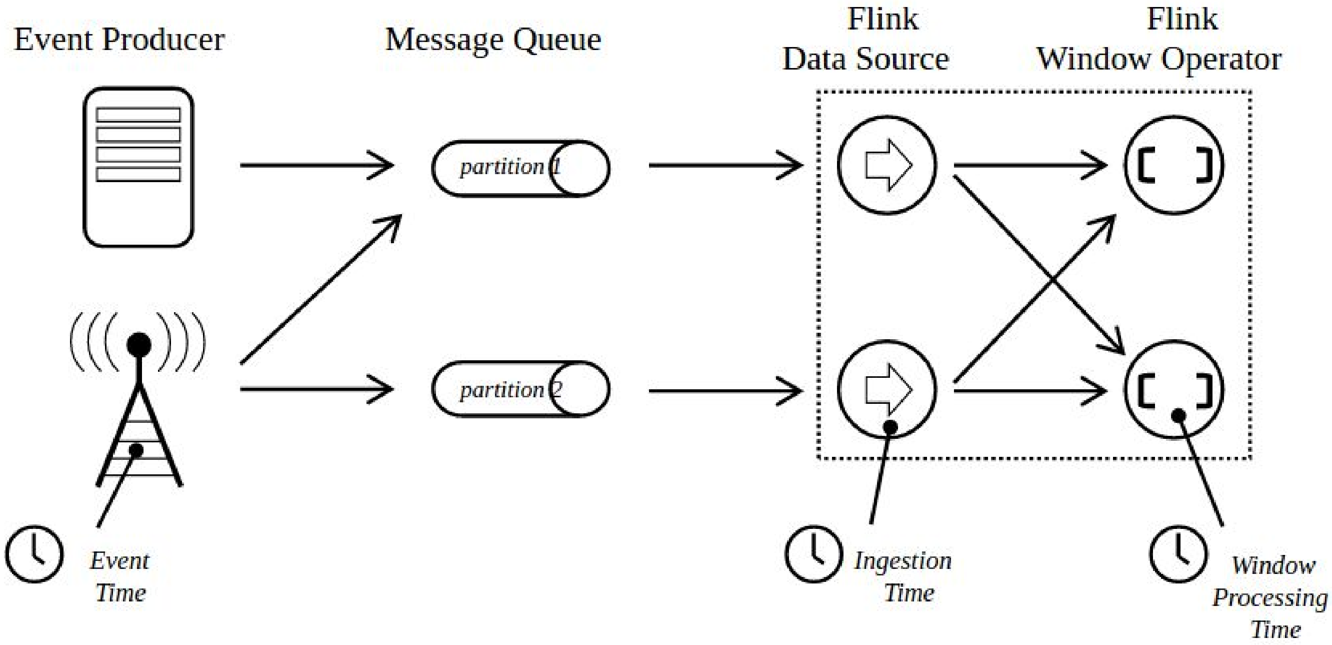
- For each Apache Flink DataStream Program
  - Obtain an execution environment.
    - StreamExecutionEnvironment.getExecutionEnvironment()
  - Load/create data sources.
    - read from file
    - read from socket
    - read from built-in sources (Kafka, RabbitMQ, etc.)
  - Execute transformations on them.
    - filter, map, reduce, etc. (**Task chaining**)
  - Specify where to save results of the computations.
    - stdout (print)
    - write to files
    - write to built-in sinks (elasticsearch, Kafka, etc.)
  - Trigger the program execution.

Hands-on BasicStructure

Various Data Streams in Apache Flink


SplitStream
CoGroupedStreams
JoinedStreams
split()
coGroup()
join()
union()
map()
flatMap()
filter()
...
DataStream
keyBy()
KeyedStream
reduce()
max()
min()
sum()
...
countWindow()
timeWindow()
window()
timeWindowAll()
countWindowAll()
windowAll()
reduce()
max()
min()
sum()
...
WindowedStream
AllWindowed Stream
allowedLateness()
trigger()
evictor()
allowedLateness()
trigger()
evictor()

Time Characteristics



E.g., ExecutionEnvironment.setStreamTimeCharacteristic(TimeCharacteristic.ProcessingTime)

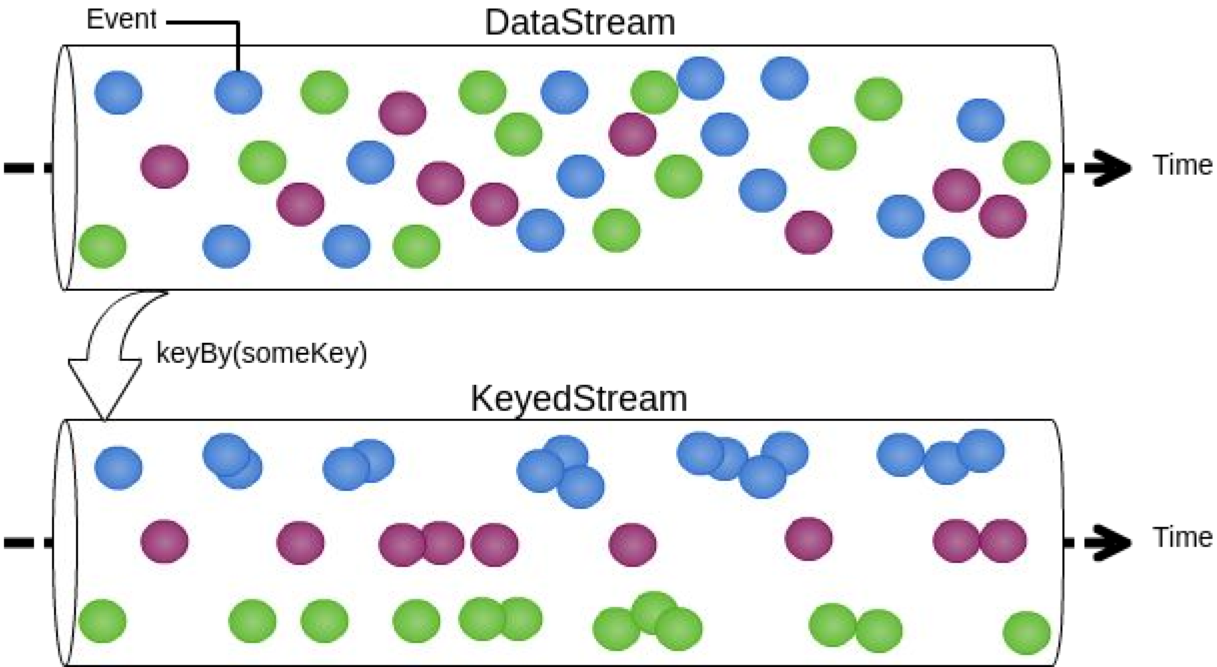
Windows

- The concept of Windows
  - cut an infinite stream into **slices** with **finite** elements.
  - based on timestamp or some criteria.
- Construction of Windows
  - Keyed Windows
    - an infinite DataStream is divided based on both window and key
    - elements with different keys can be processed concurrently
  - Non-keyed Windows
- We focus on the keyed windowing.

Windows

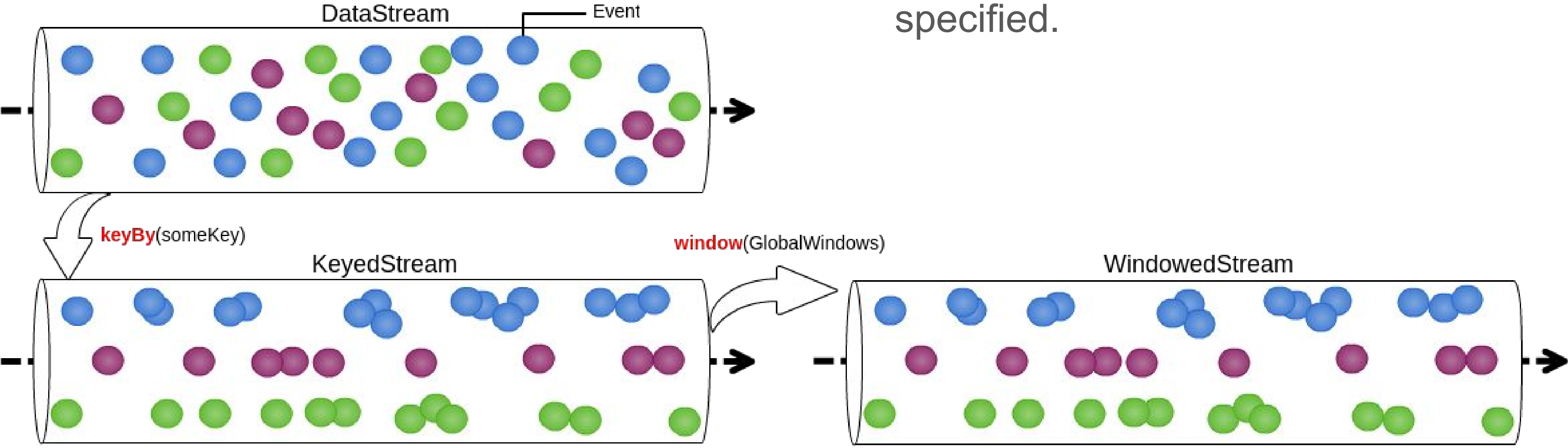
- Basic Structure
  - Key
  - Window assigner
  - Window function
    - reduce()
    - fold()
    - apply()

```
val input: DataStream[T] = ...

input

    .keyBy(<key selector>)

    .window(<window assigner>)

    .<windowed transformation>(<window function>)
```



Window Assigner - Global Windows

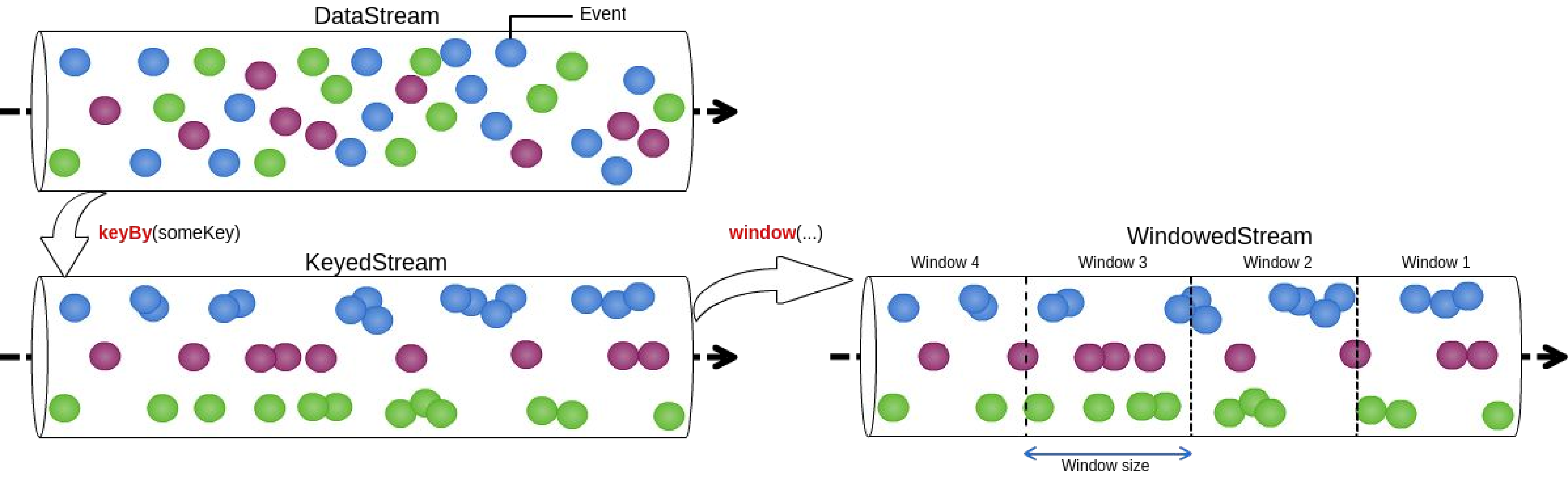
- Single per-key global window.
- Only useful if a custom trigger is specified.



Flink.tw
Apache Flink Taiwan User Group

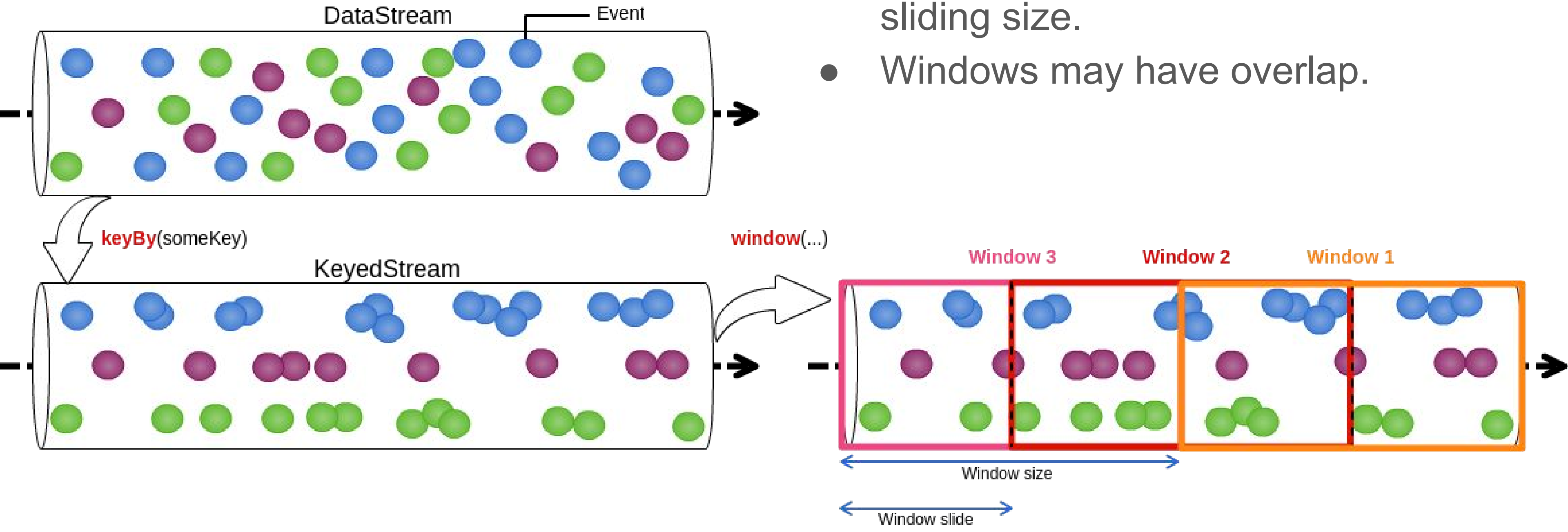
Window Assigner - Tumbling Windows

- Defined by window size.
- Windows are disjoint.



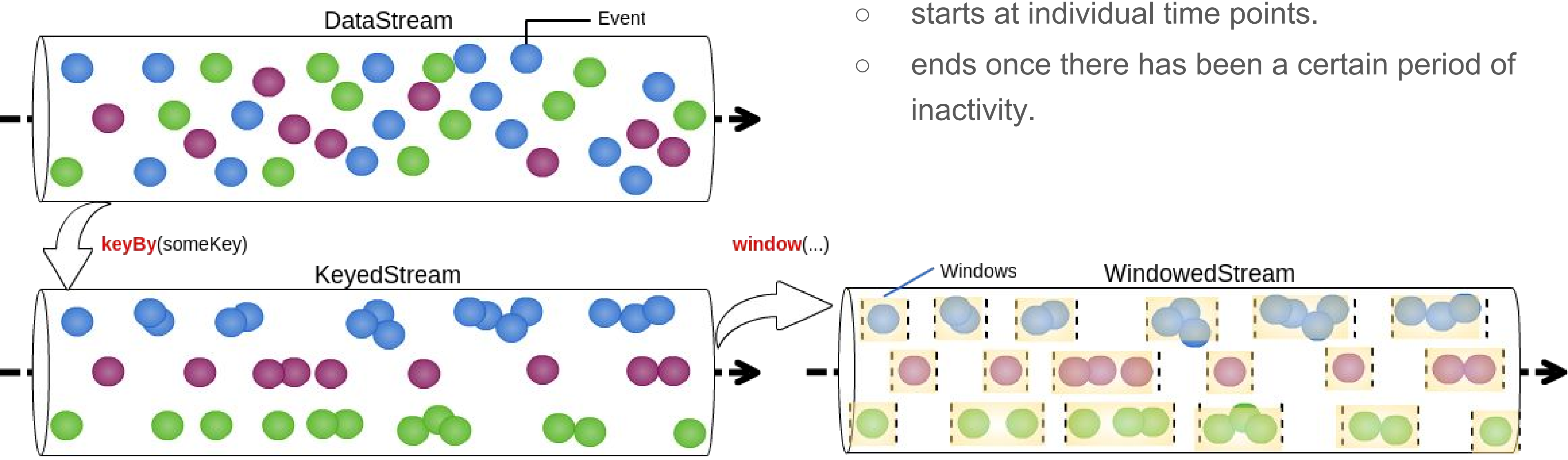
Window Assigner - Sliding Windows

- Defined by both window size and sliding size.
- Windows may have overlap.

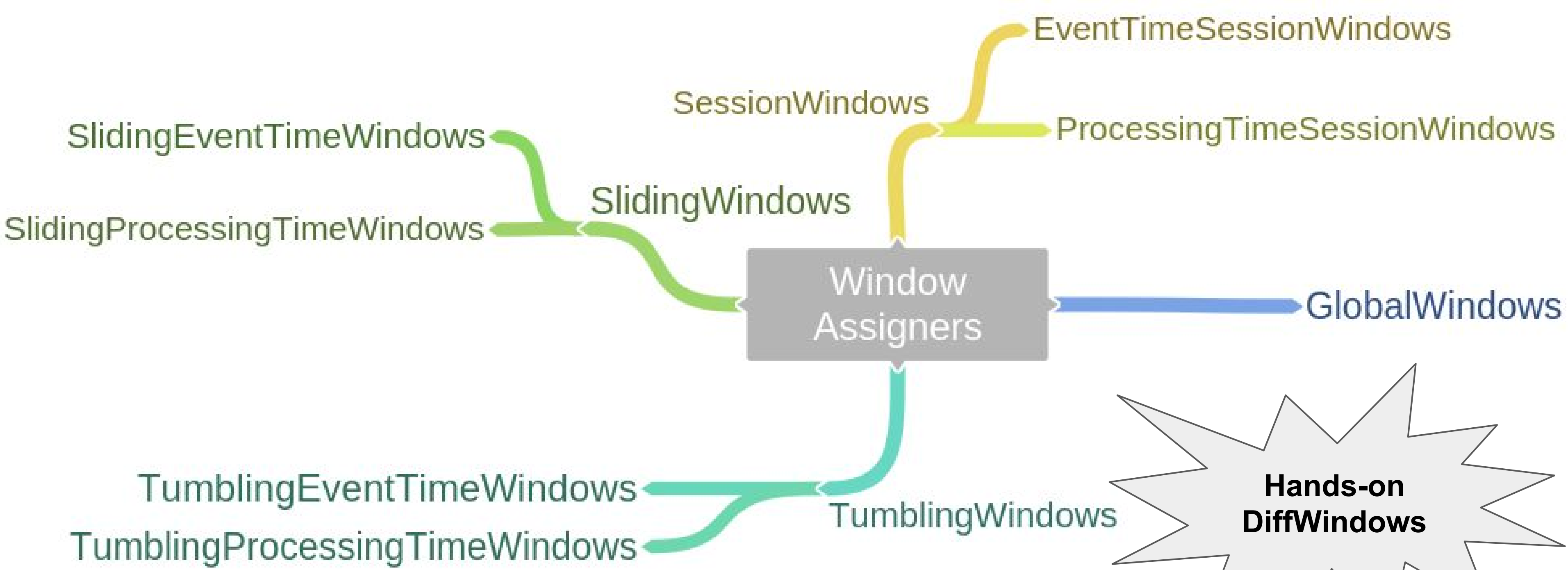


Window Assigner - Session Windows

- Defined by gap of time.
- Window time
  - starts at individual time points.
  - ends once there has been a certain period of inactivity.



Cheat Sheet of Window Assigners



Window Assigners

- SessionWindows
  - EventTimeSessionWindows
  - ProcessingTimeSessionWindows
- SlidingWindows
  - SlidingEventTimeWindows
  - SlidingProcessingTimeWindows
- GlobalWindows
- TumblingWindows
  - TumblingEventTimeWindows
  - TumblingProcessingTimeWindows

Hands-on
DiffWindows

Window Functions

- WindowFunction
  - Cache elements internally
  - Provides Window meta information (e.g., start time, end time, etc.)
- ReduceFunction
  - Incrementally aggregation
  - No access to Window meta information
- FoldFunction
  - Incrementally aggregation
  - No access to Window meta information
- WindowFunction with ReduceFunction / FoldFunction
  - Incrementally aggregation
  - Has access to Window meta information

Dealing with Data Lateness

- Set allowed lateness to Windows
  - new in 1.1.0
  - watermark passes end timestamp of window + allowedLateness.
  - defaults to 0, drop event once it is late.

Hands-on WindowFuncs

We’re all set. Thank you!!!

Just Flink It!