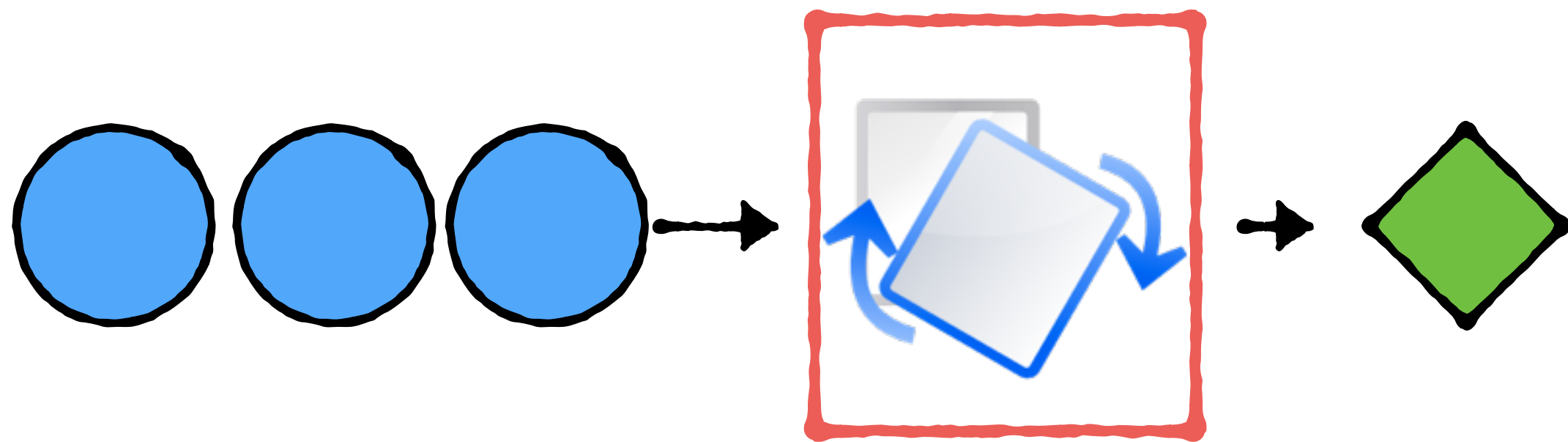


# Stateful Transformation

Transformation applied across  
multiple stream entities



**Accumulate data**

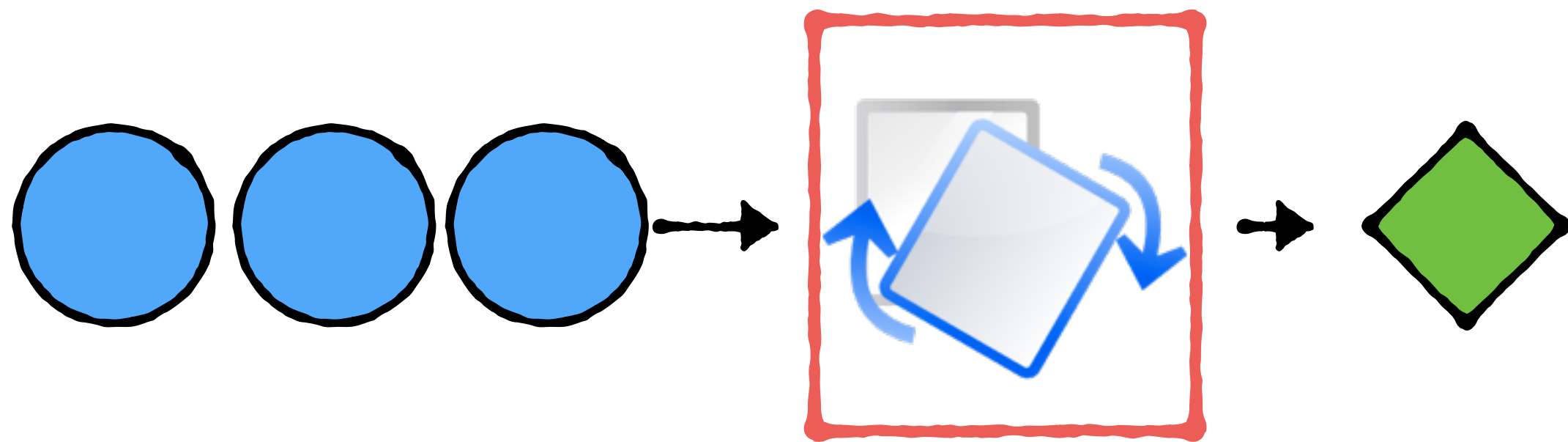
entire stream

window

per key, per operator

# Stateful Transformation

Transformation applied across  
multiple stream entities



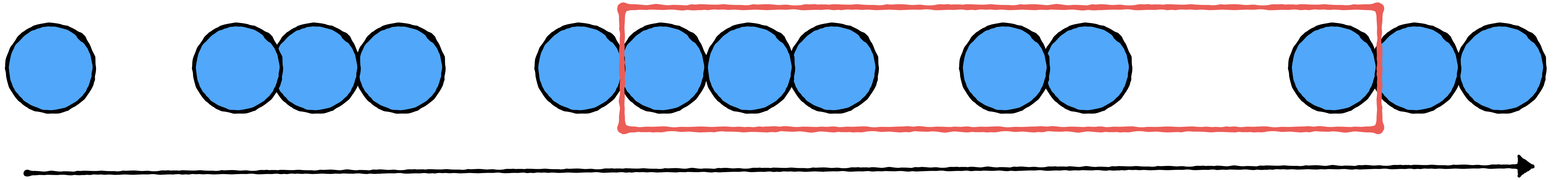
**Accumulate data**

entire stream

**window**

per key, per operator

Window



A subset of the stream

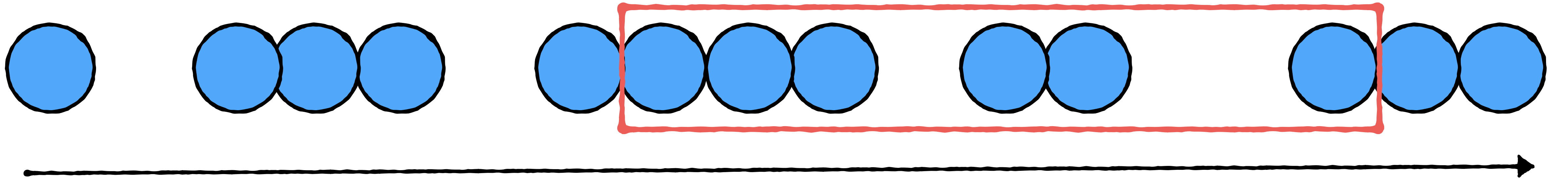
time

time

count

interval between entities

Window



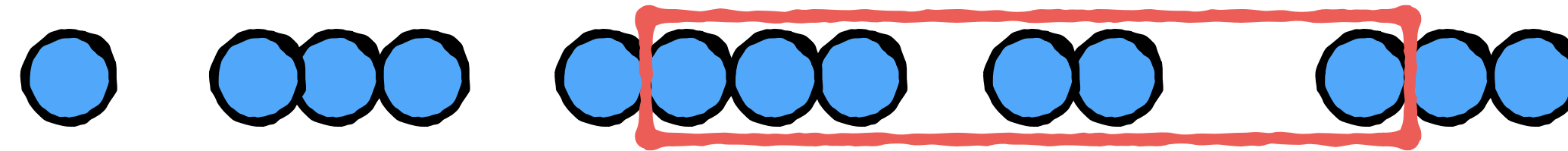
Aggregate data within the window

sum

min

reduce

# Types of Windows



## Built-in

Based on time,  
count etc

## Custom

Use the window  
API to define  
windows

# Built-in Window types

Tumbling

Sliding

Count

Session

Global

# Built-in Window types

Tumbling

Sliding

Count

Session

Global

Based on time

# Built-in Window types

Tumbling

Sliding

**Count**

Session

Global

Based on count  
of entities



# Built-in Window types

Tumbling

Sliding

Count

**Session**

Global

Based on time interval  
between entities

# Built-in Window types

Tumbling

Sliding

Count

Session

Global

All entities in one  
window

# Built-in Window types

Tumbling

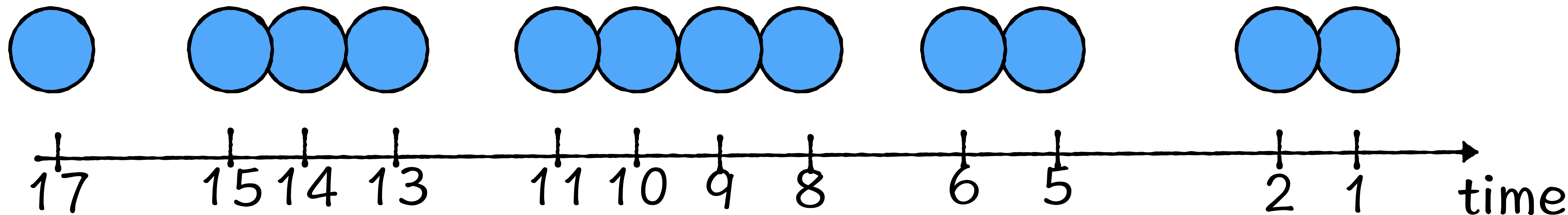
Sliding

Count

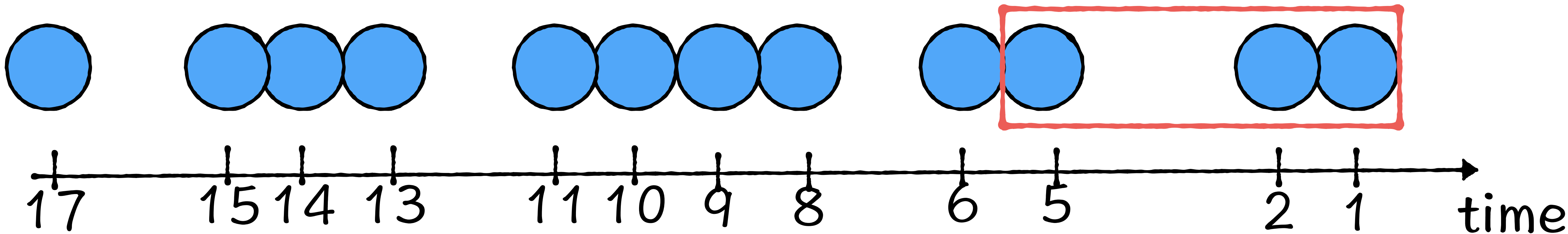
Session

Global

# Tumbling Window

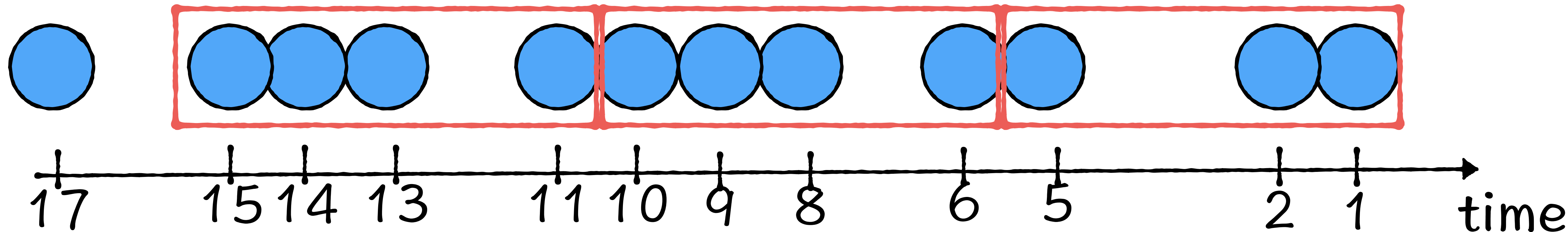


# Tumbling Window



Fixed window size based on time

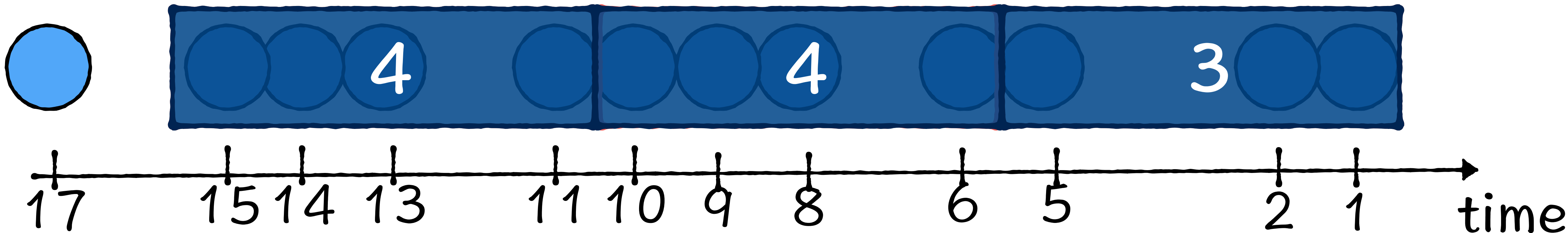
# Tumbling Window



Fixed window size based on time

Non-overlapping

# Tumbling Window

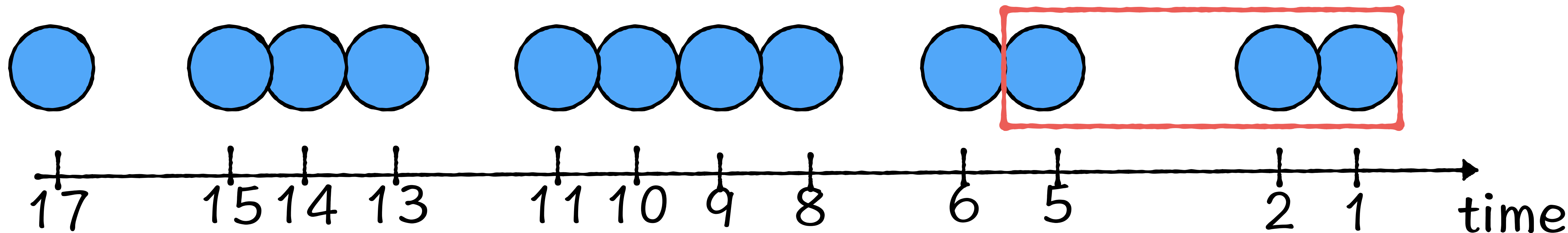


Fixed window size based on time

Non-overlapping

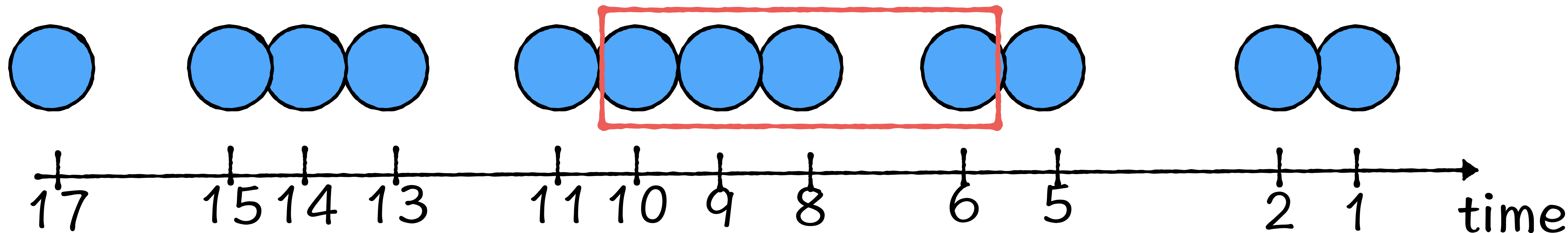
Different #entities per window

# Tumbling Window

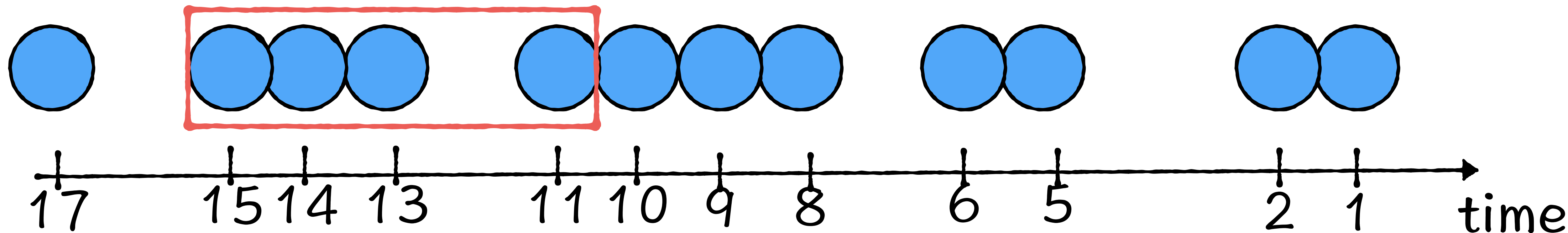




# Tumbling Window



# Tumbling Window



# Built-in Window types

Tumbling

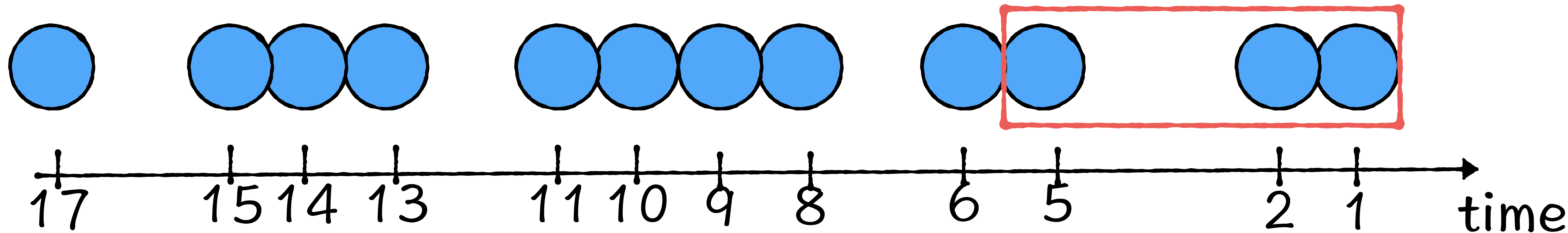
Sliding

Count

Session

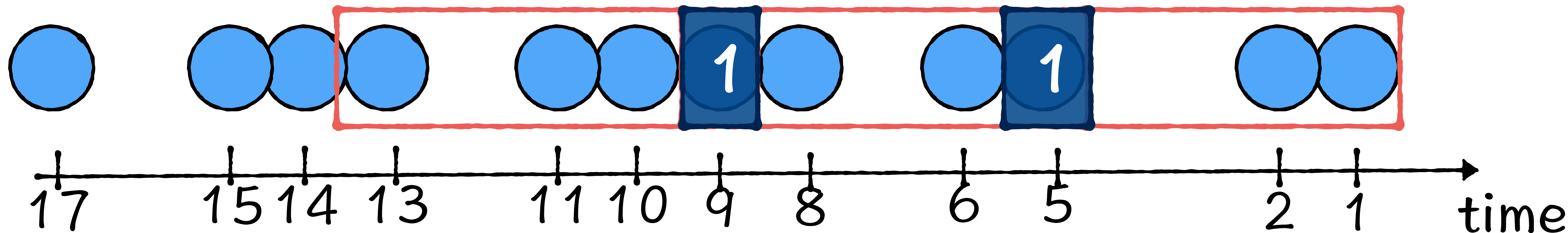
Global

# Sliding Window



Fixed window size based on time

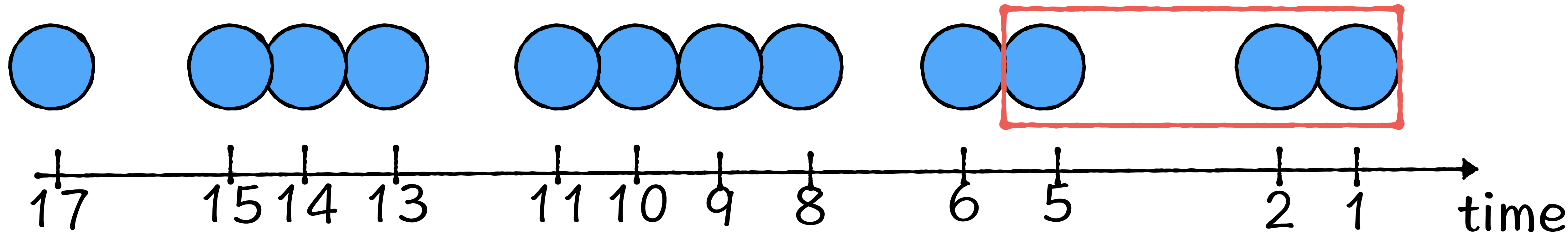
# Sliding Window



Fixed window size based on time

Overlapping time

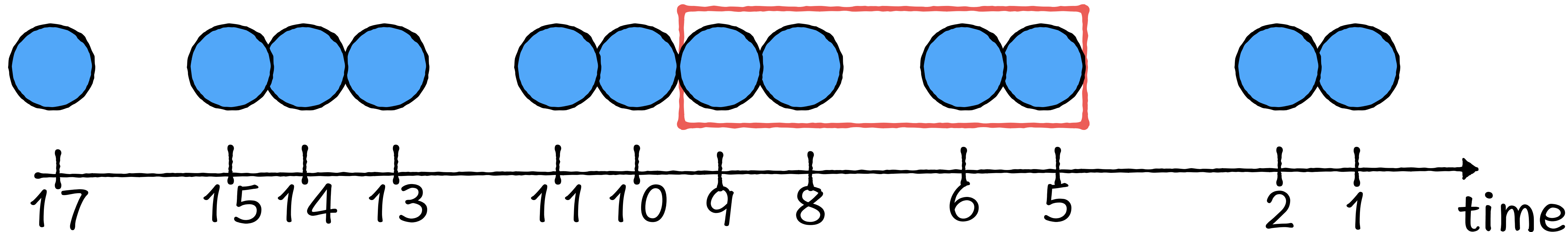
# Sliding Window



Window size 5

Sliding Interval 4

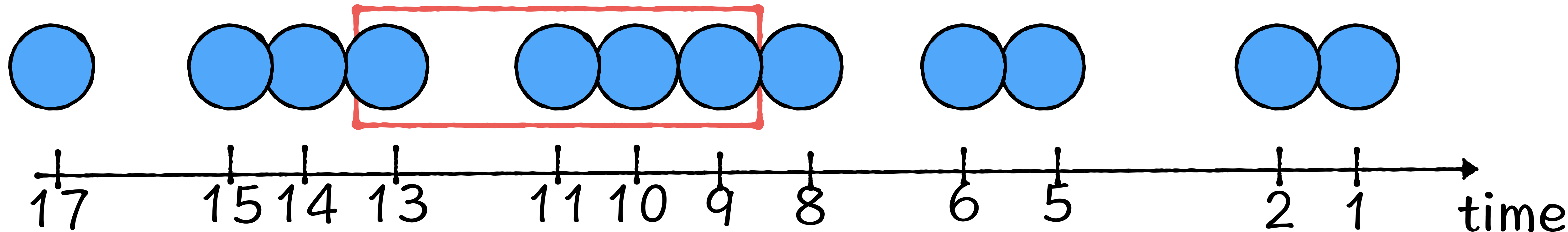
# Sliding Window



Window size 5

Sliding Interval 4

# Sliding Window

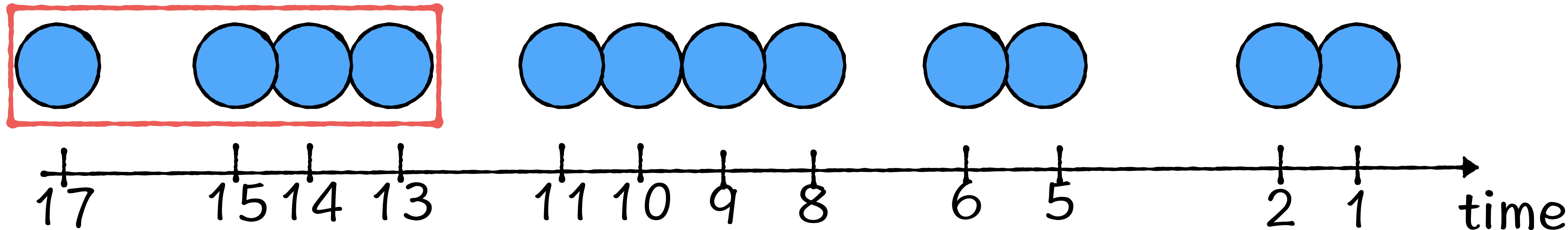


Window size 5

Sliding Interval 4



# Sliding Window



Window size 5

Sliding Interval 4

# Built-in Window types

Tumbling

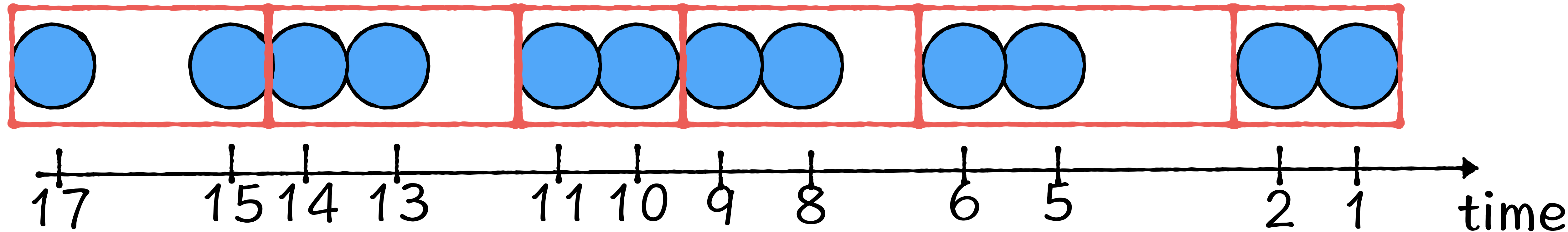
Sliding

Count

Session

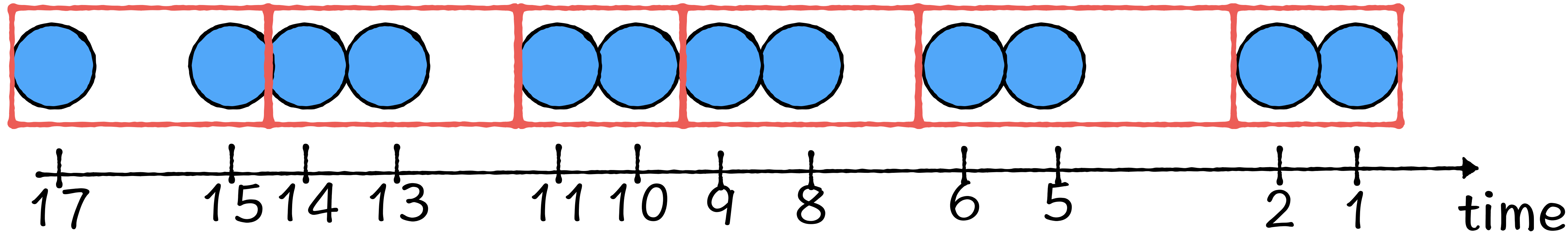
Global

# Count Window



Count of entities in the  
window is constant

# Count Window

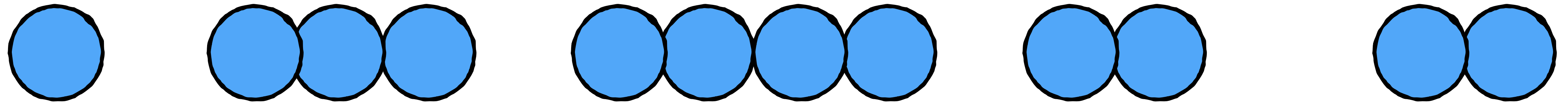


Count window is applied on  
keyed streams

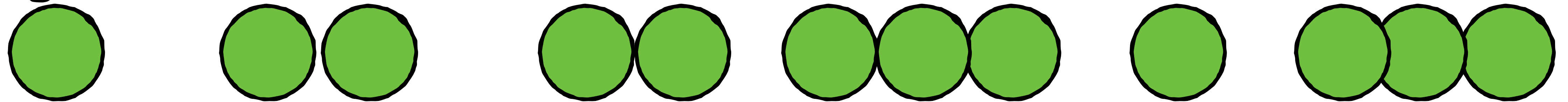
# Count Window

## Keyed streams

key = "a"



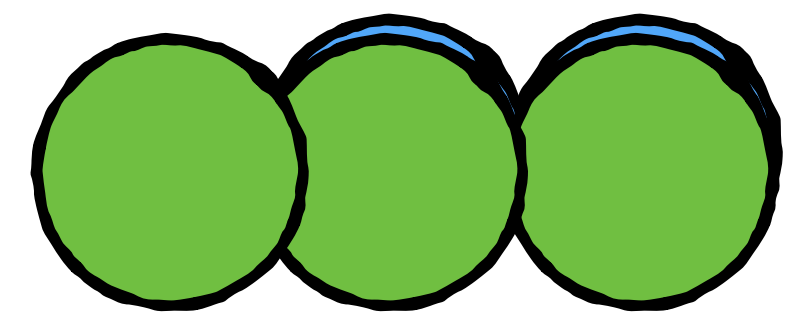
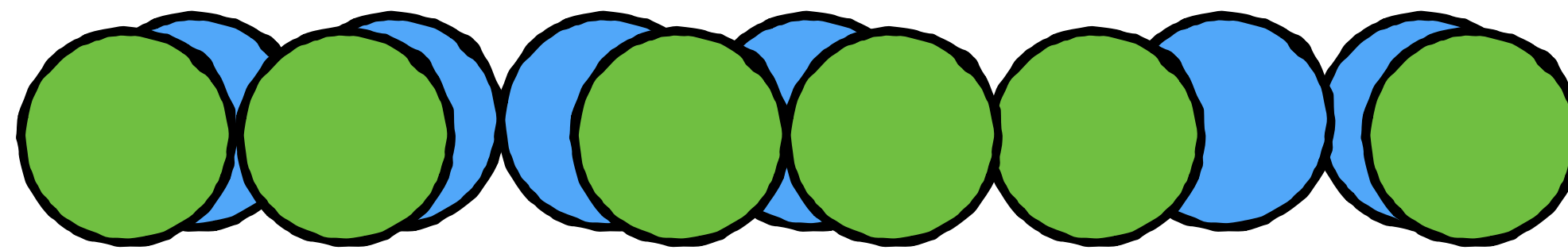
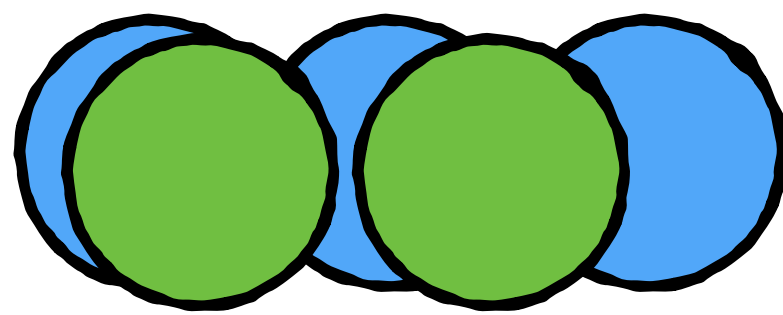
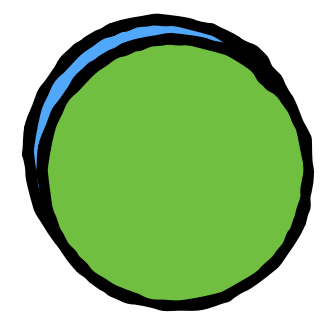
key = "b"



# Count Window

## Keyed streams

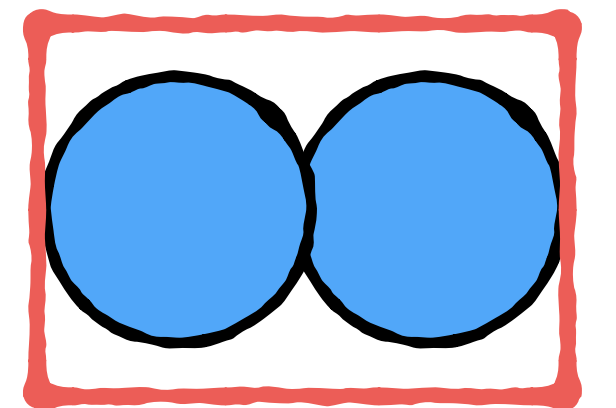
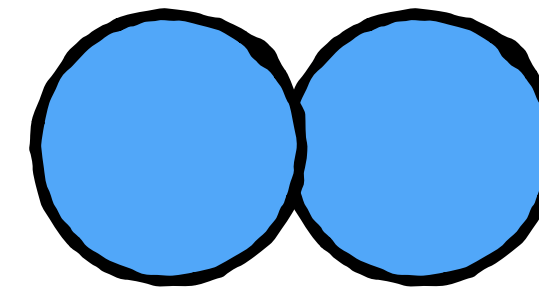
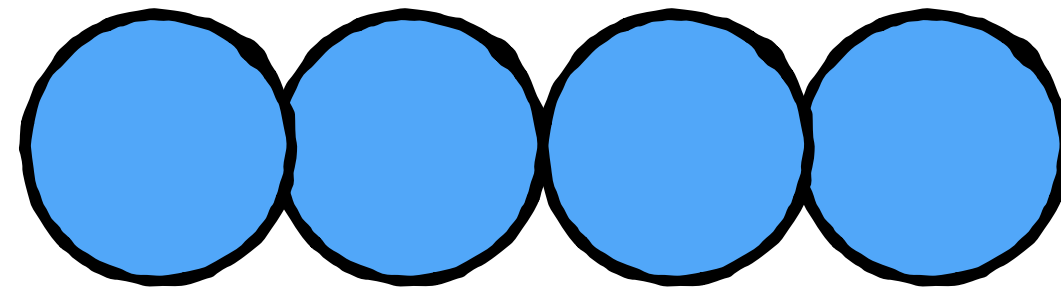
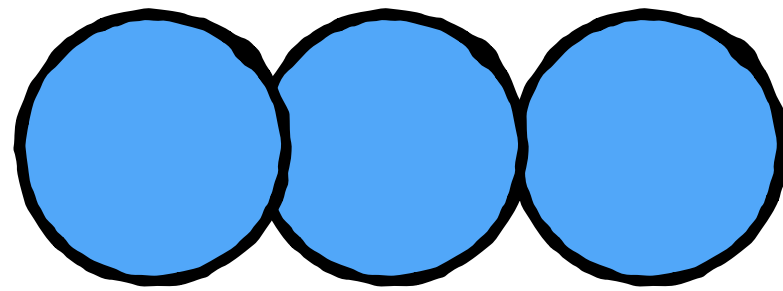
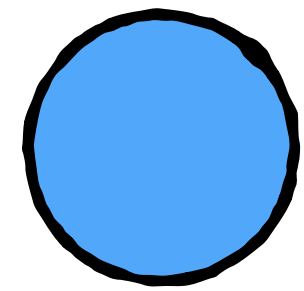
key = "a"    key = "b"



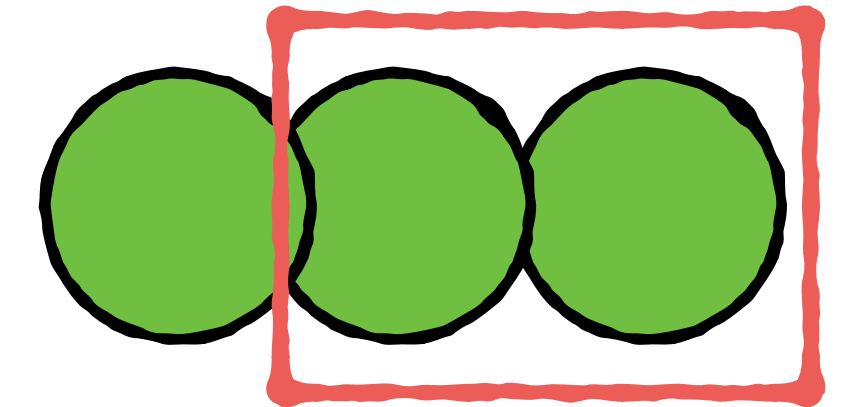
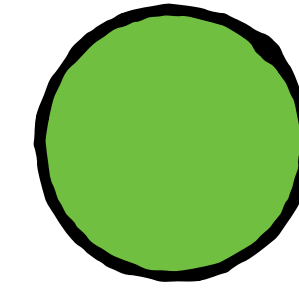
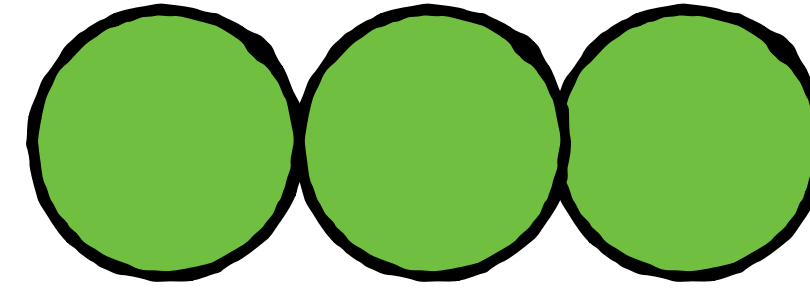
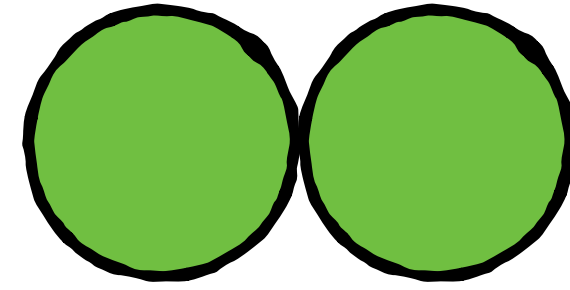
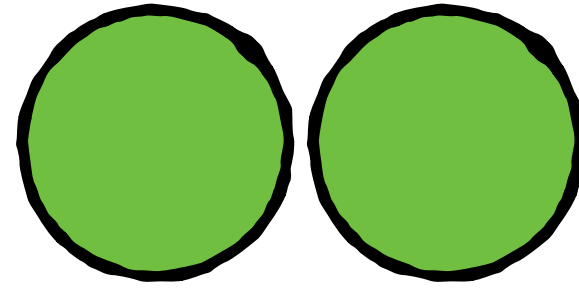
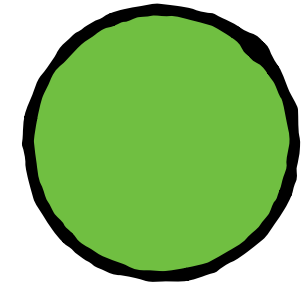
# Count Window

## Keyed streams

key = "a"



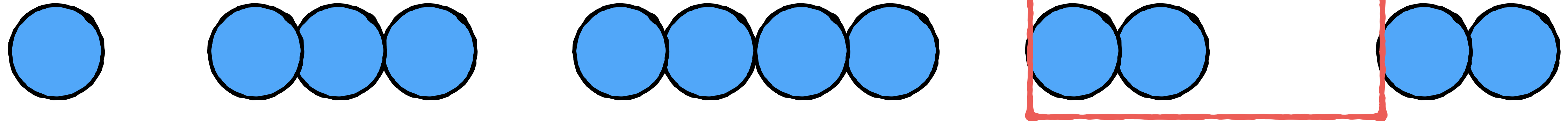
key = "b"



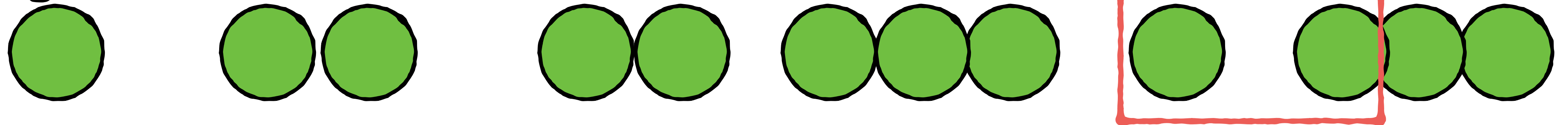
# Count Window

## Keyed streams

key = "a"



key = "b"

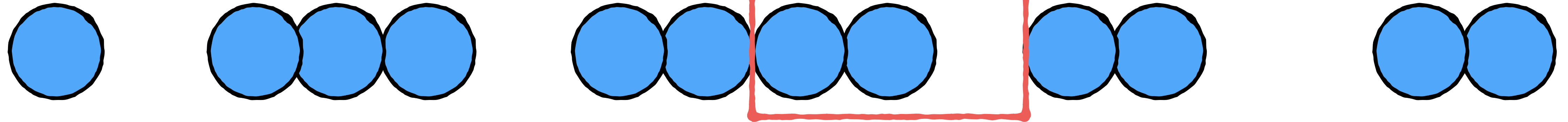




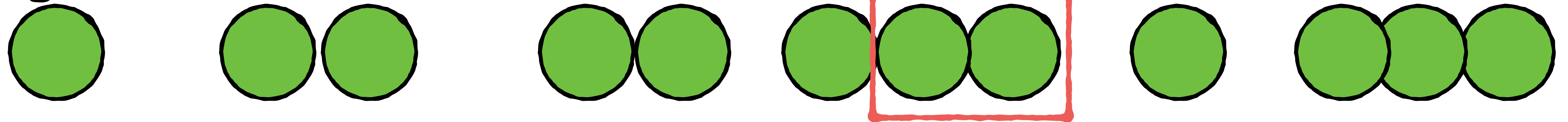
# Count Window

## Keyed streams

key = "a"



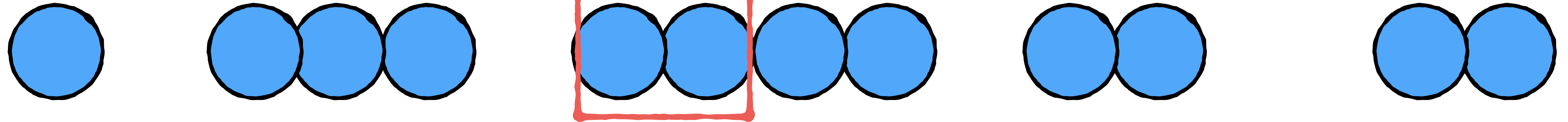
key = "b"



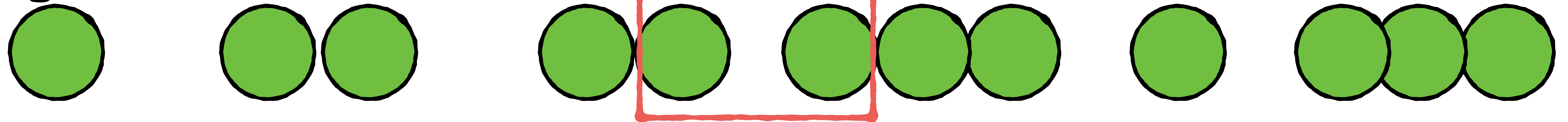
# Count Window

## Keyed streams

key = "a"



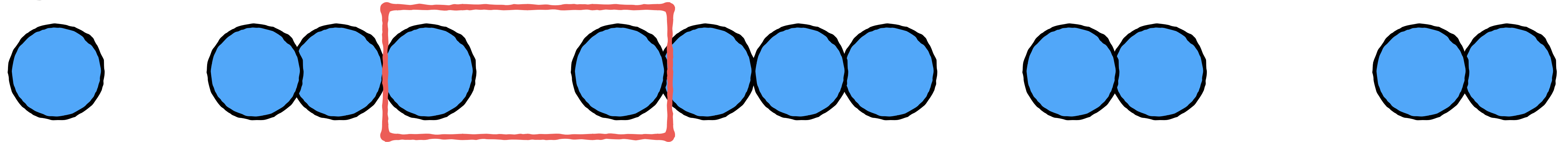
key = "b"



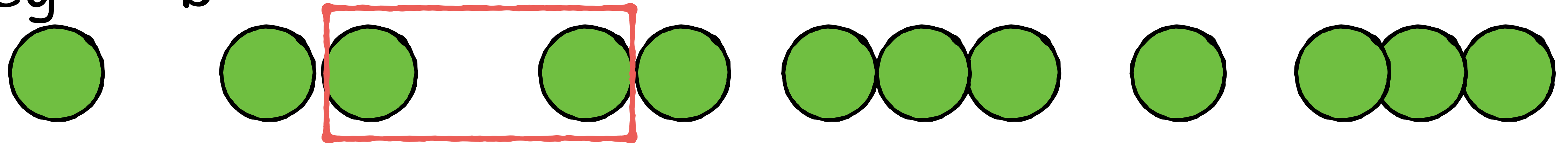
# Count Window

## Keyed streams

key = "a"



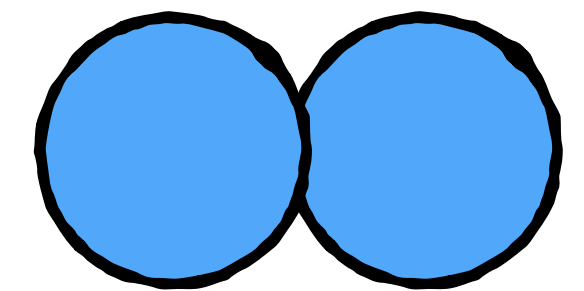
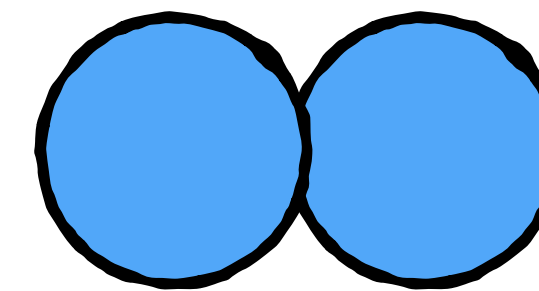
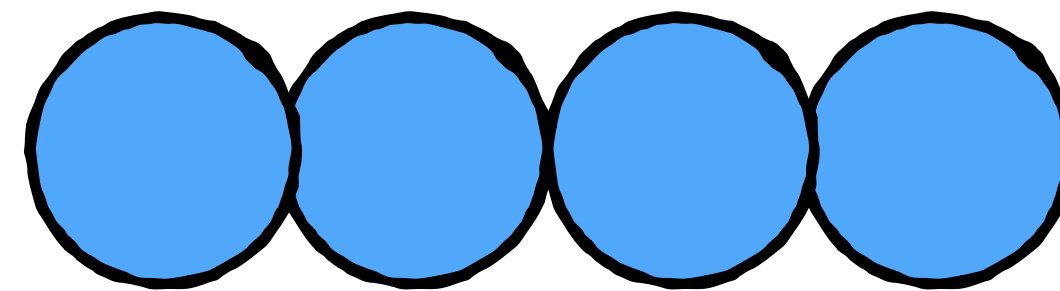
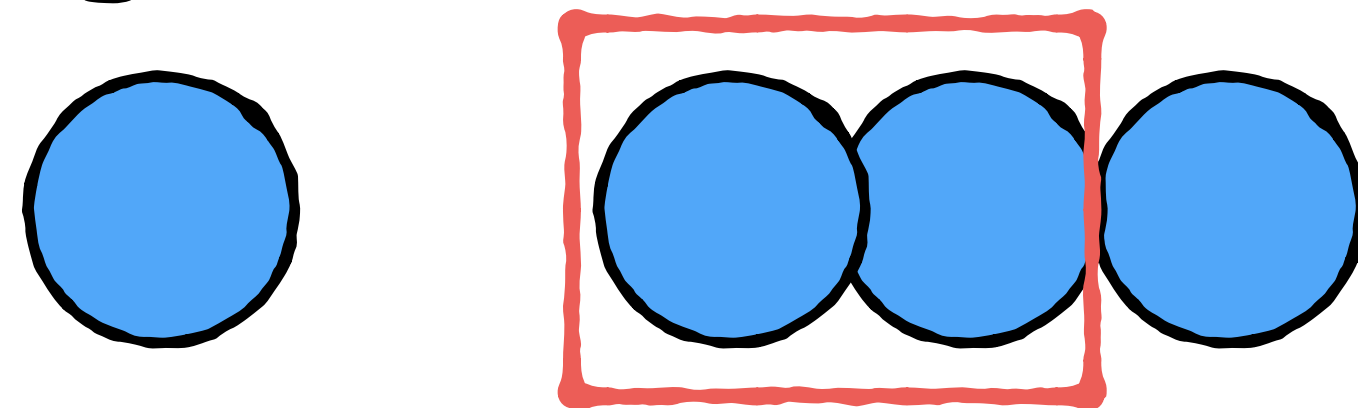
key = "b"



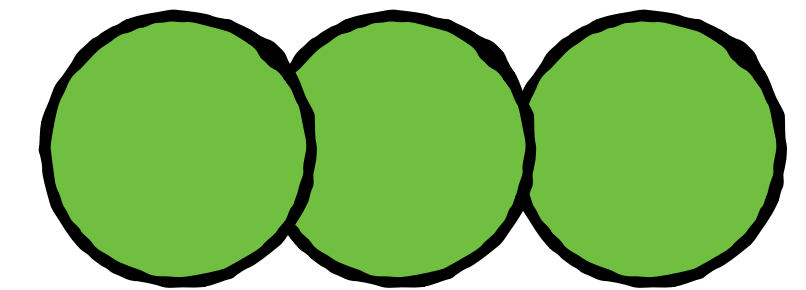
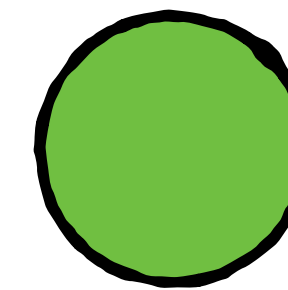
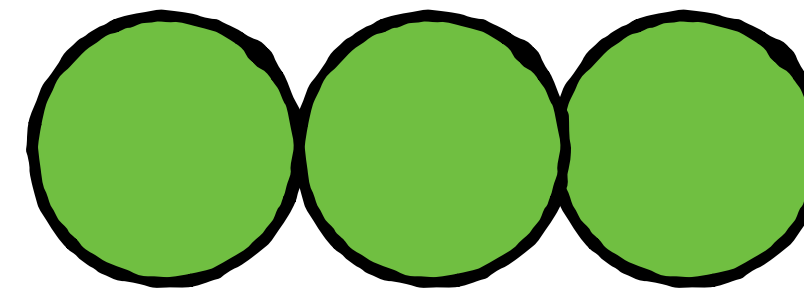
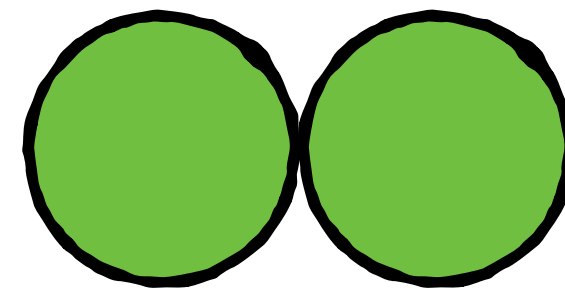
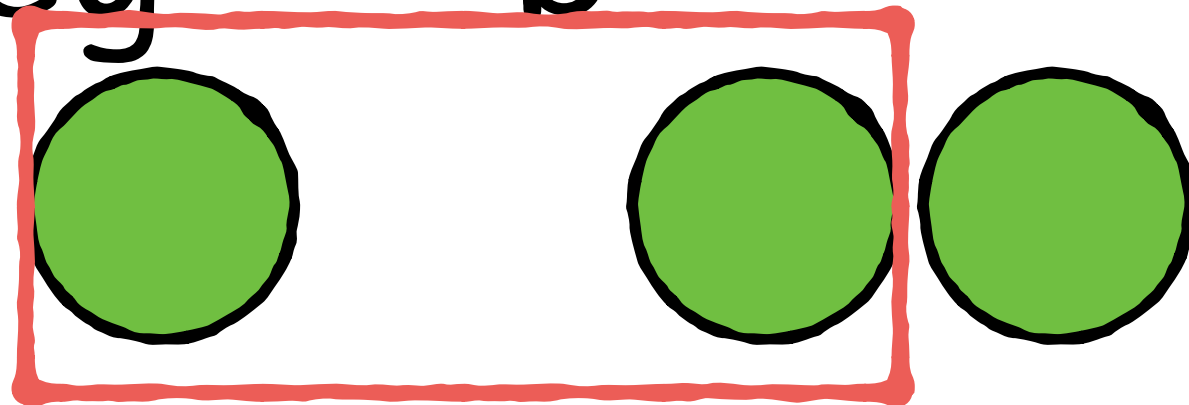
# Count Window

## Keyed streams

key = "a"



key = "b"



# Built-in Window types

Tumbling

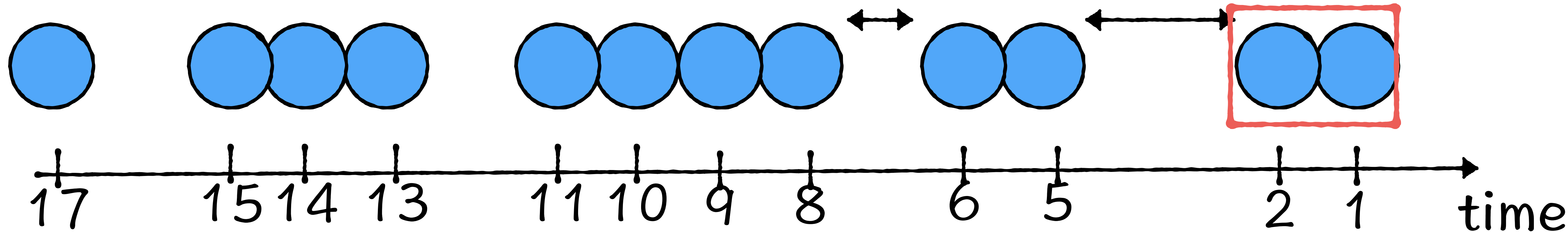
Sliding

Count

Session

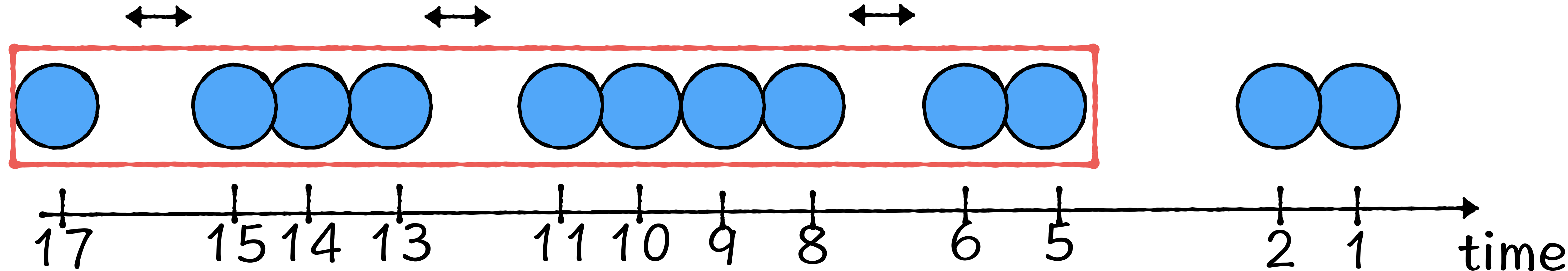
Global

# Session Window



Trigger new window based  
on gap between entities

# Session Window



Gaps not enough to start a  
new window

# Built-in Window types

Tumbling

Sliding

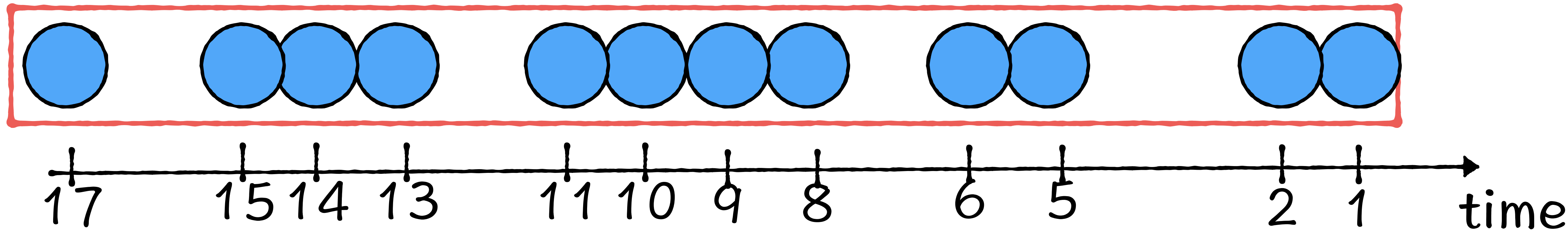
Count

Session

Global

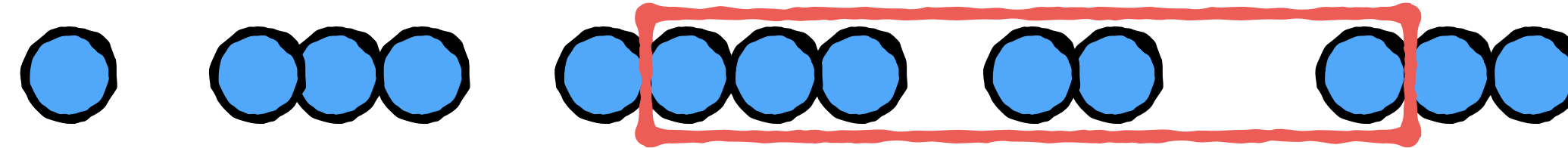


# Global Window



All entities in a single  
window

# Applying Windows on Streams



Non-Keyed

Window defined over  
the whole stream

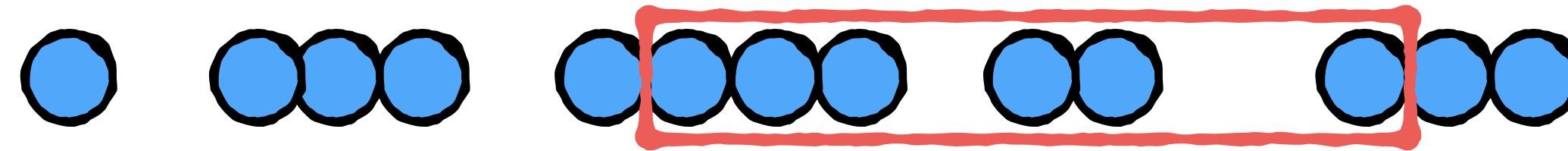
`windowAll()`

Keyed

Window applied  
on each group in  
a keyed stream

`window()`

# Types of Windows



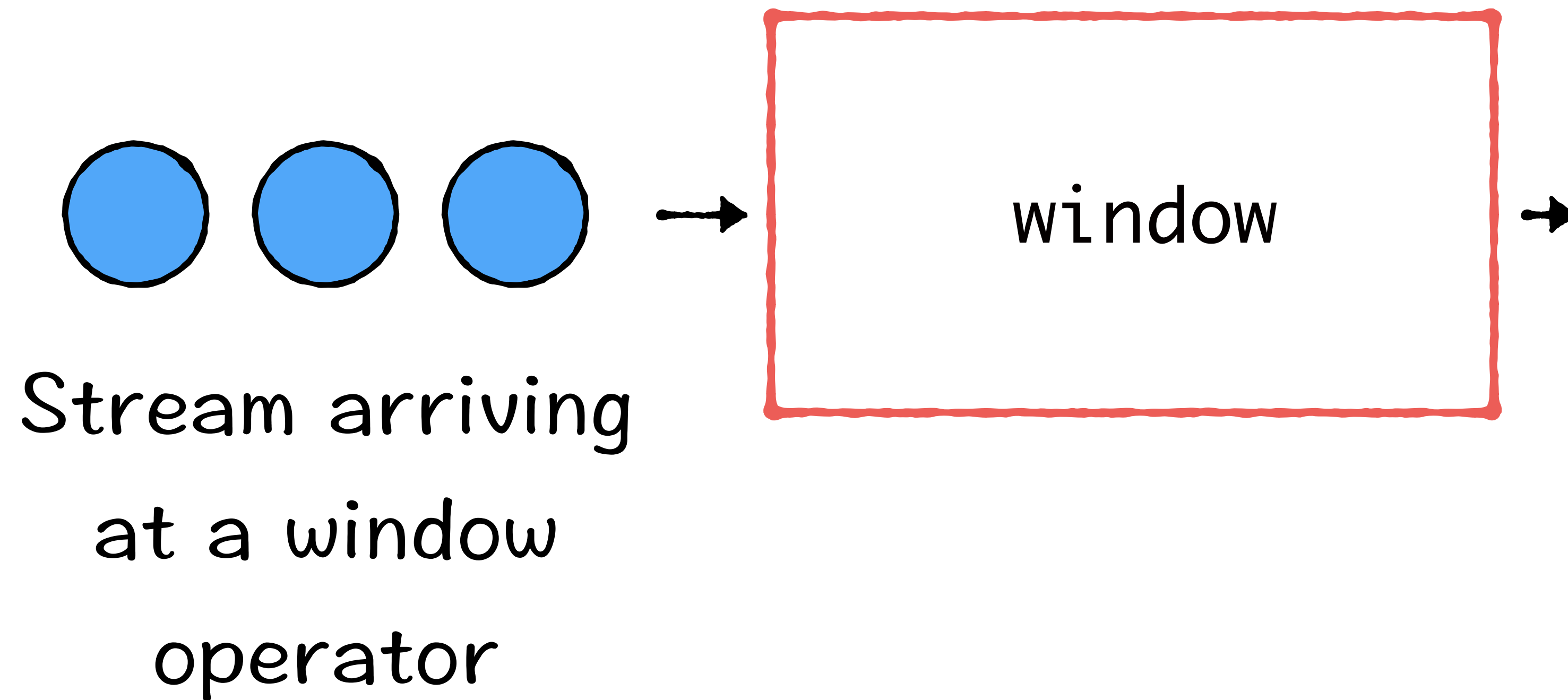
## Built-in

Based on time,  
count etc

## Custom

Use the window  
API to define  
windows

# Window API



# Window API

window

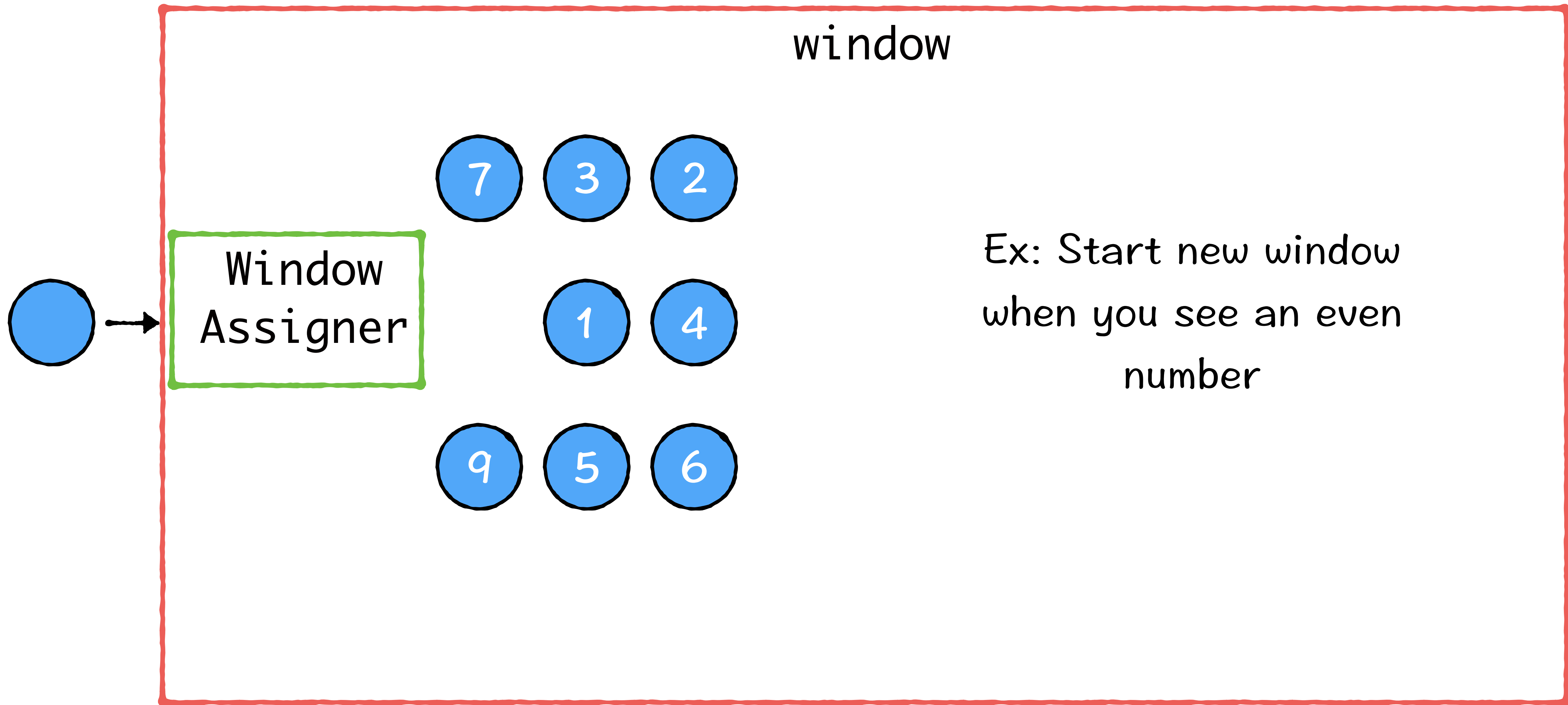
Decides when a new window starts

Ex: Start new window when  
you see an even number

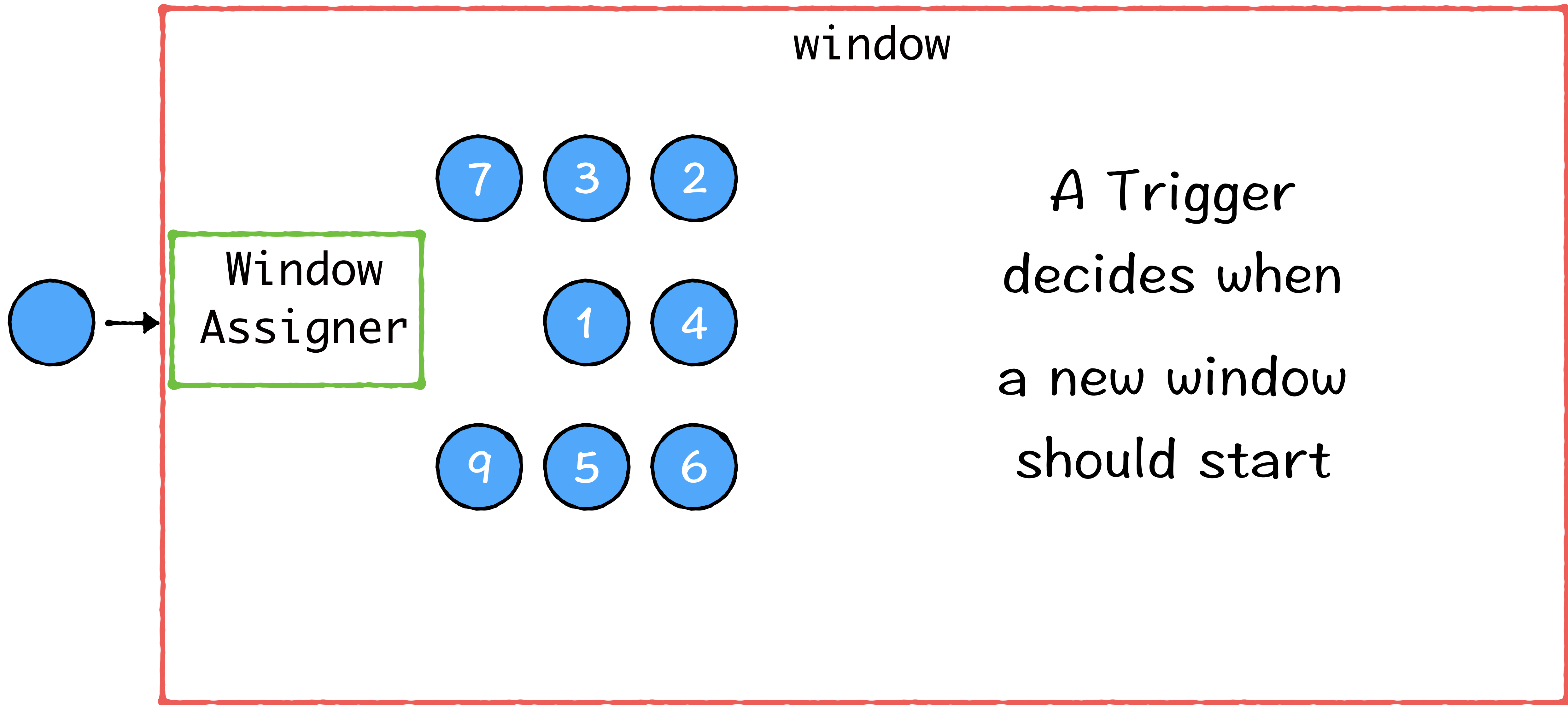


Window  
Assigner

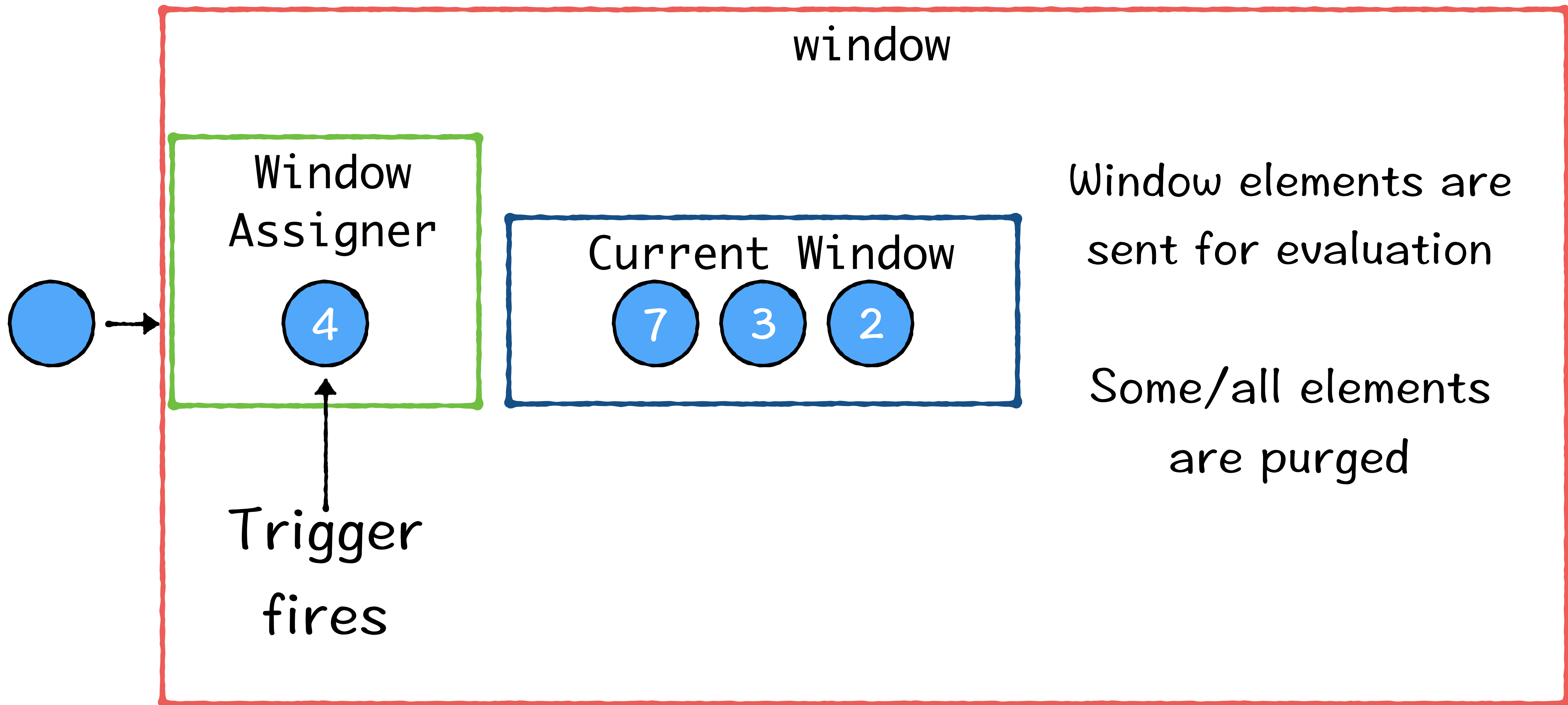
# Window API



# Window API

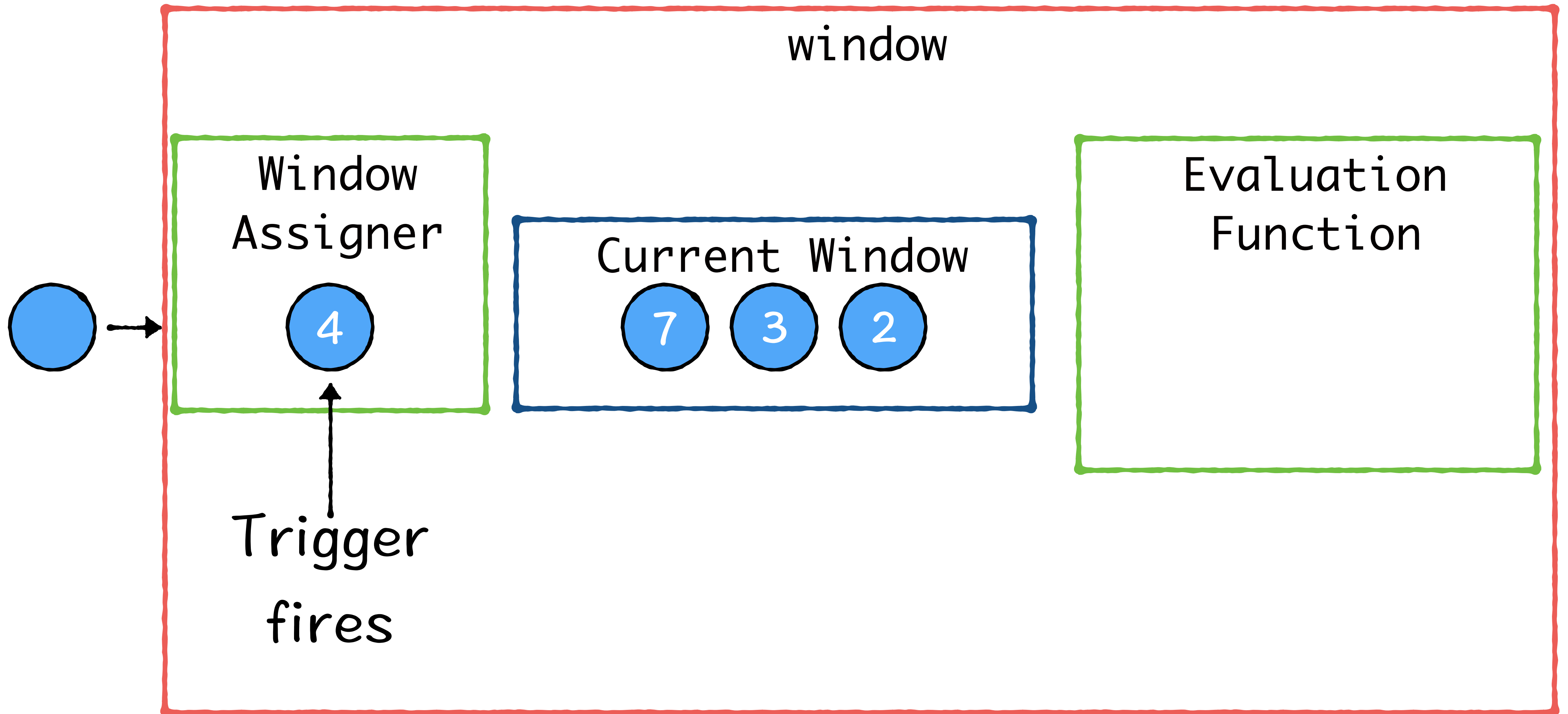


# Window API

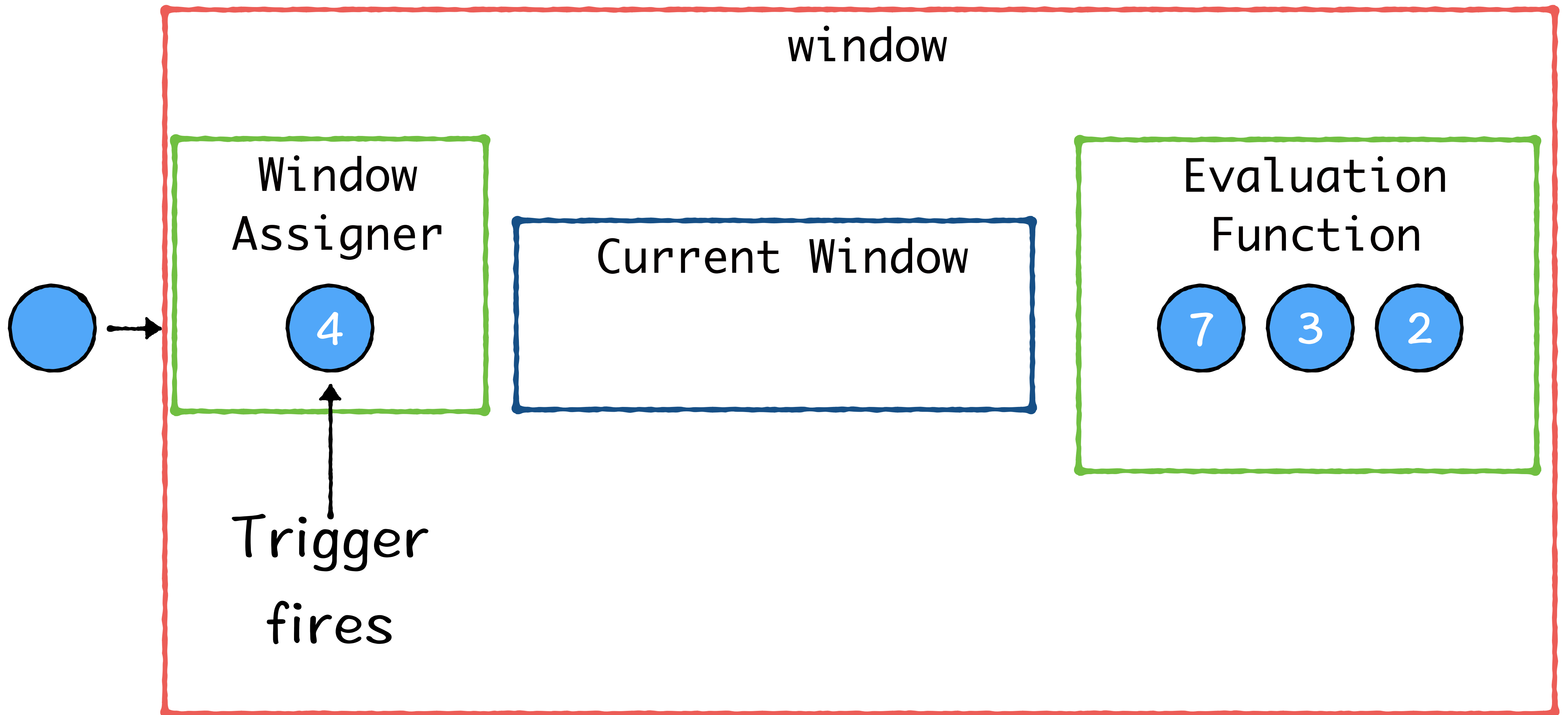




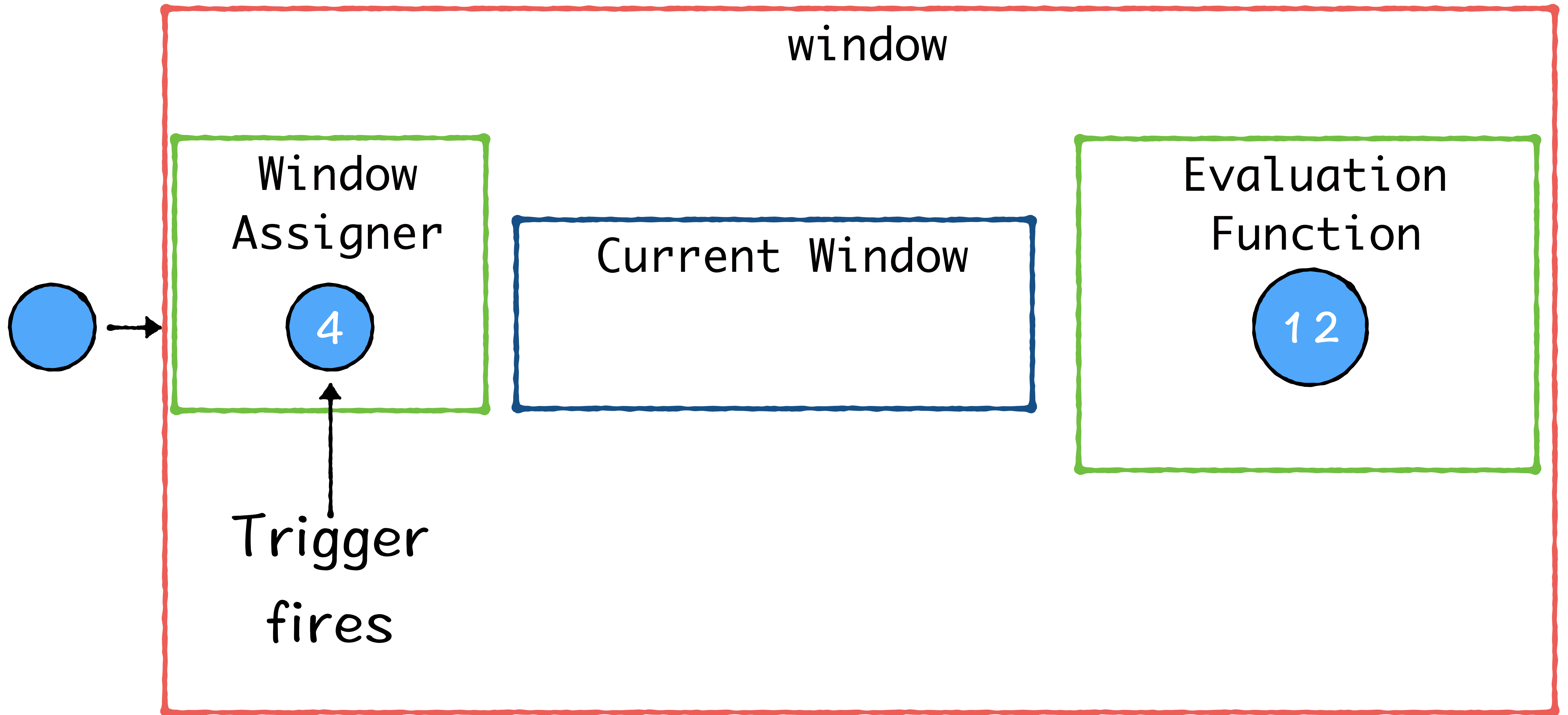
# Window API



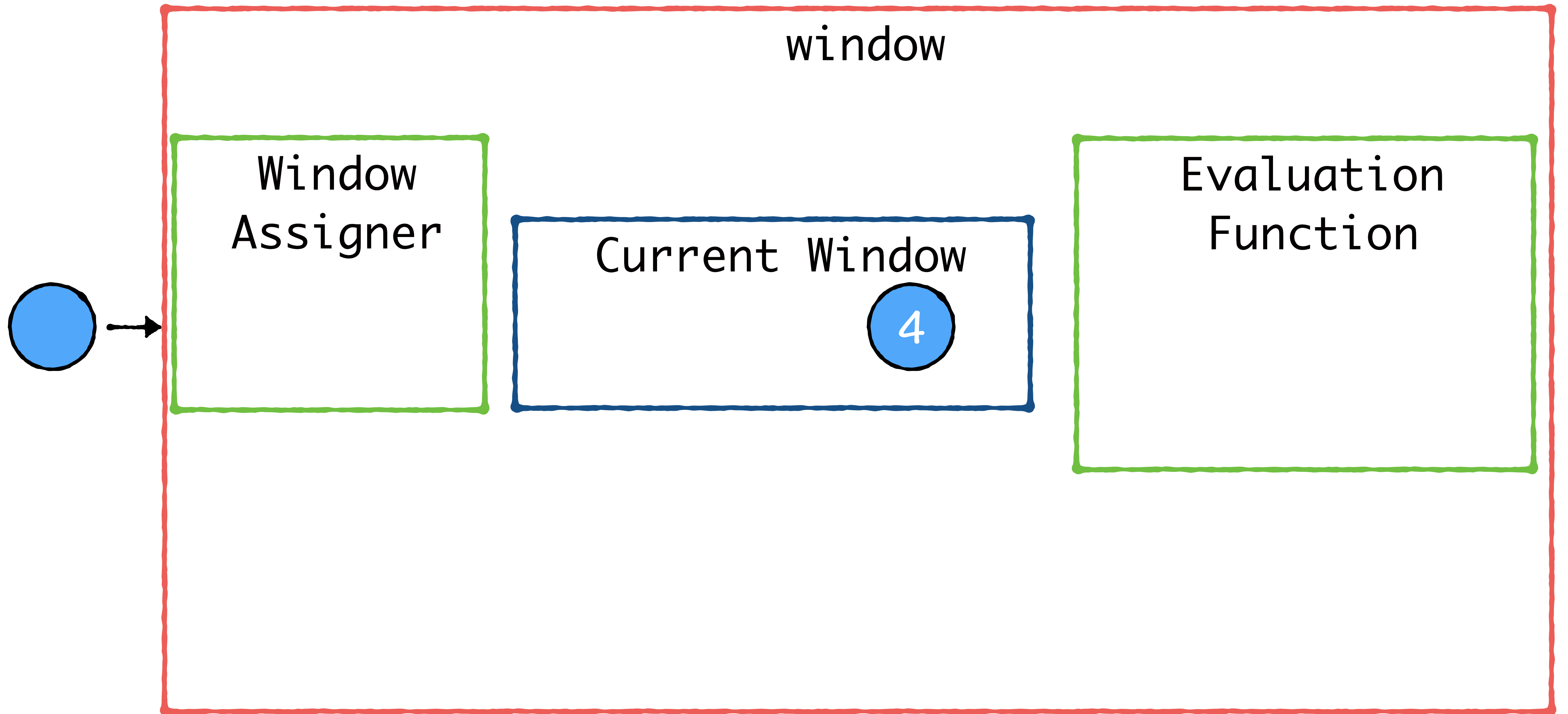
# Window API



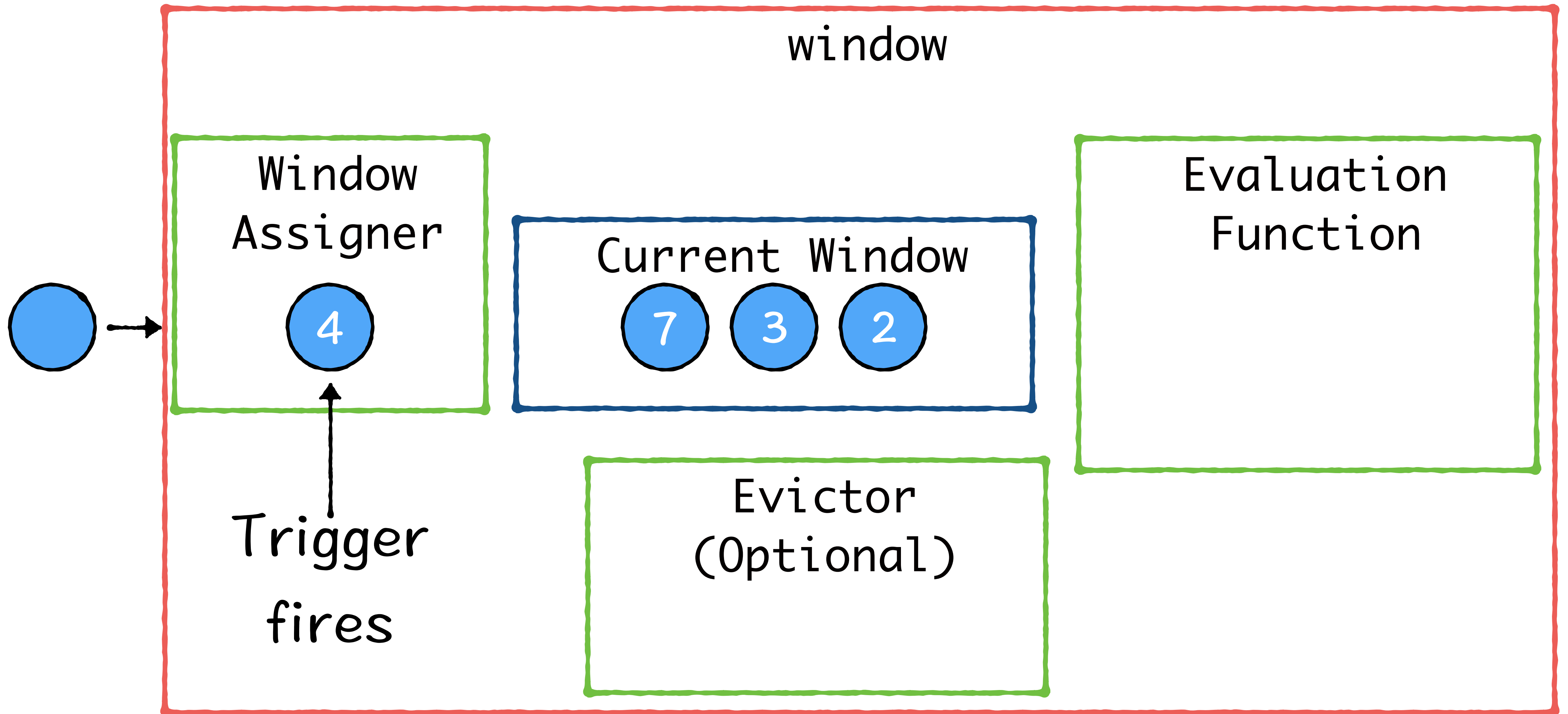
# Window API



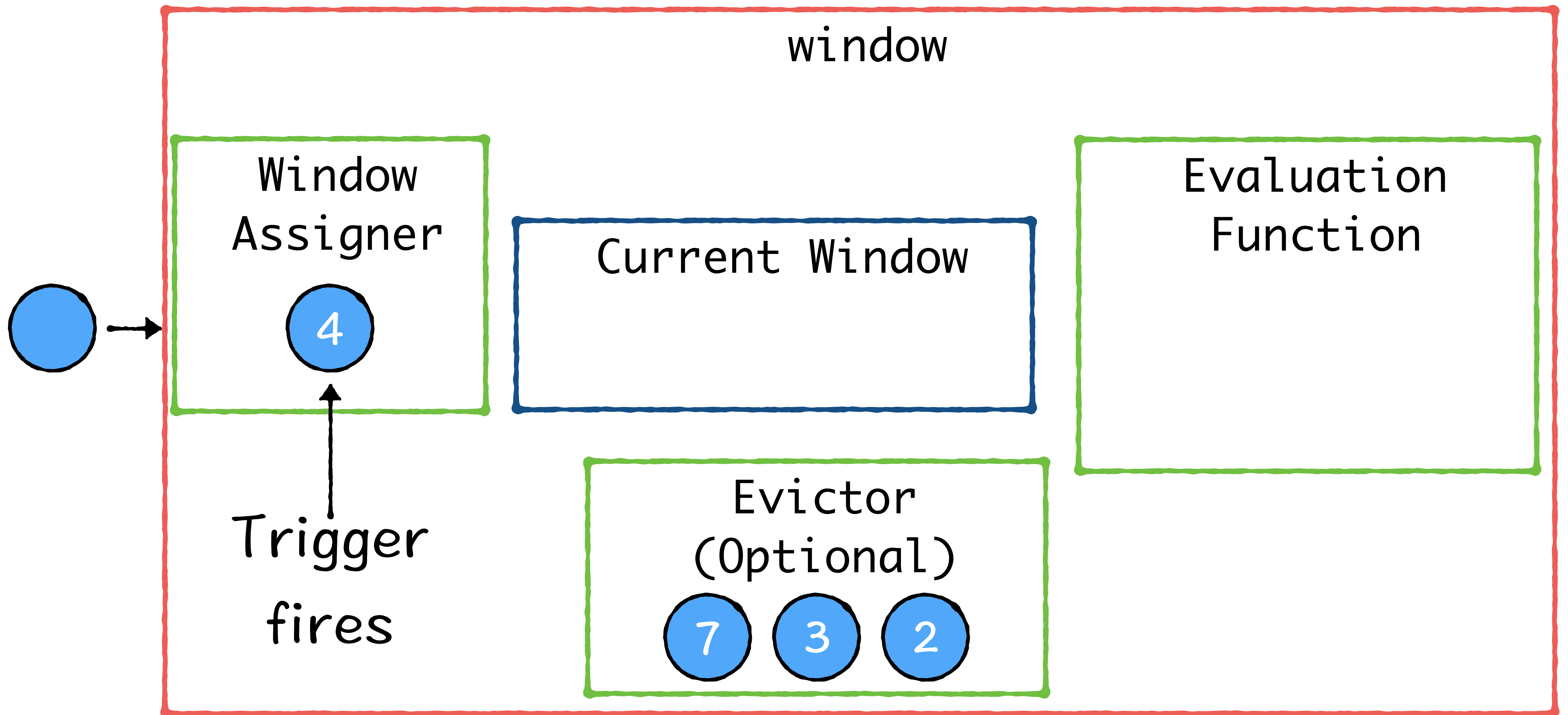
# Window API



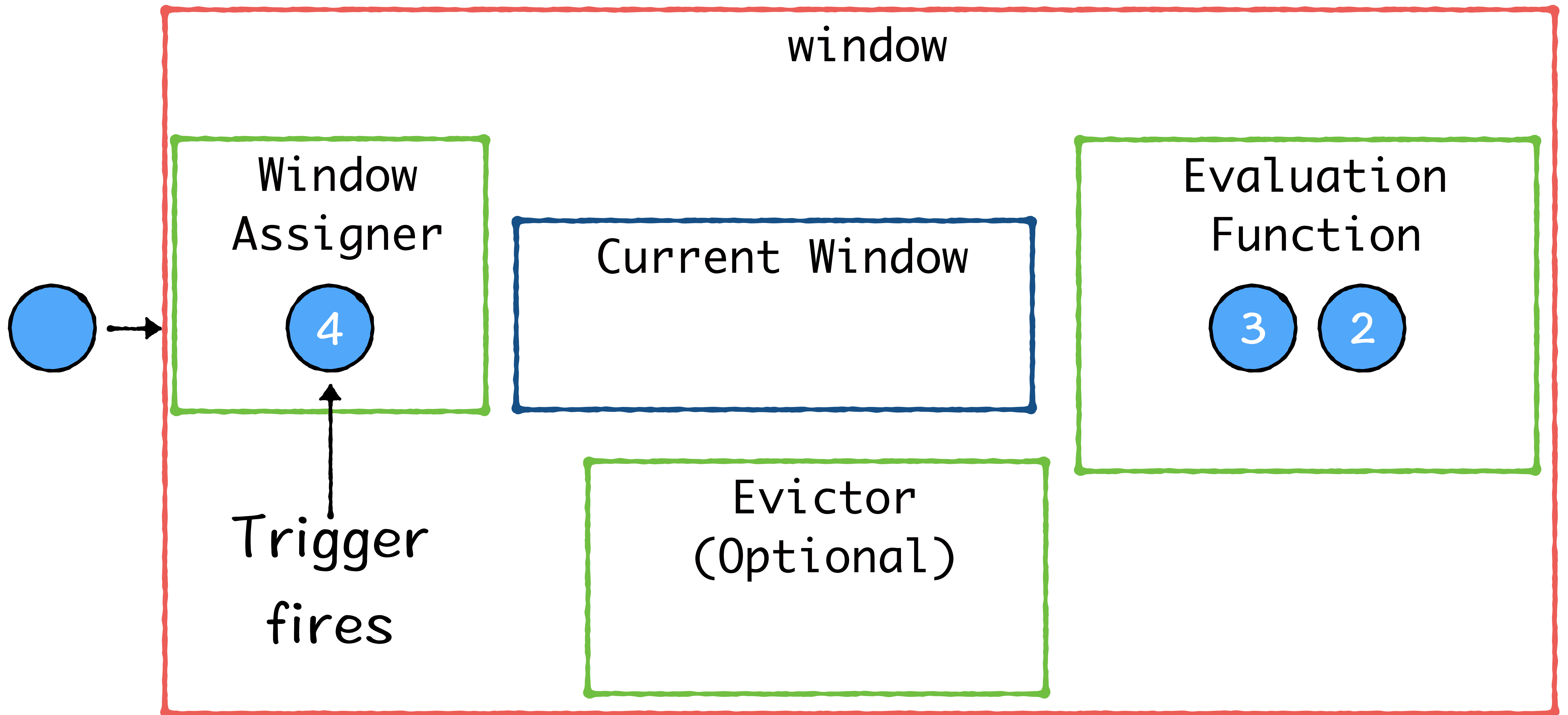
# Window API



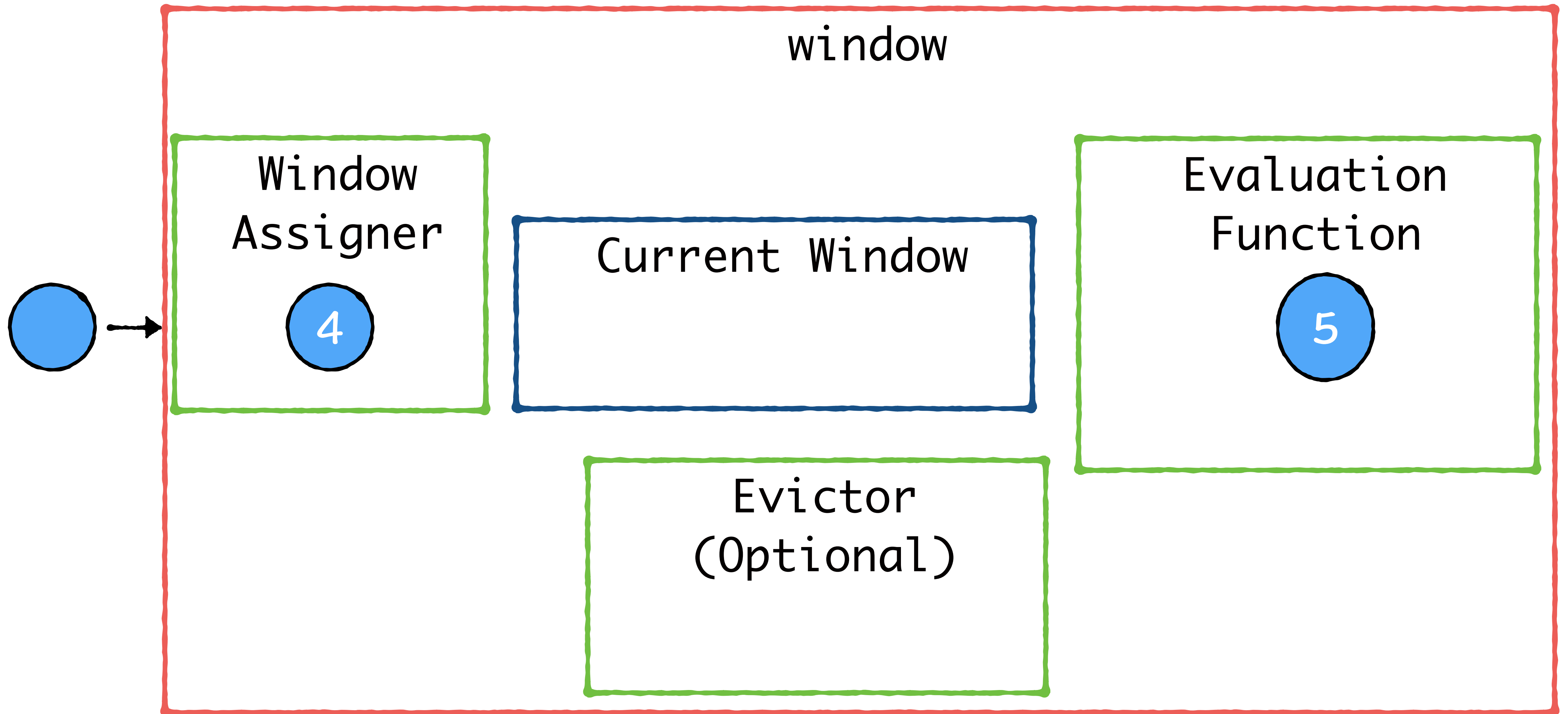
# Window API



# Window API

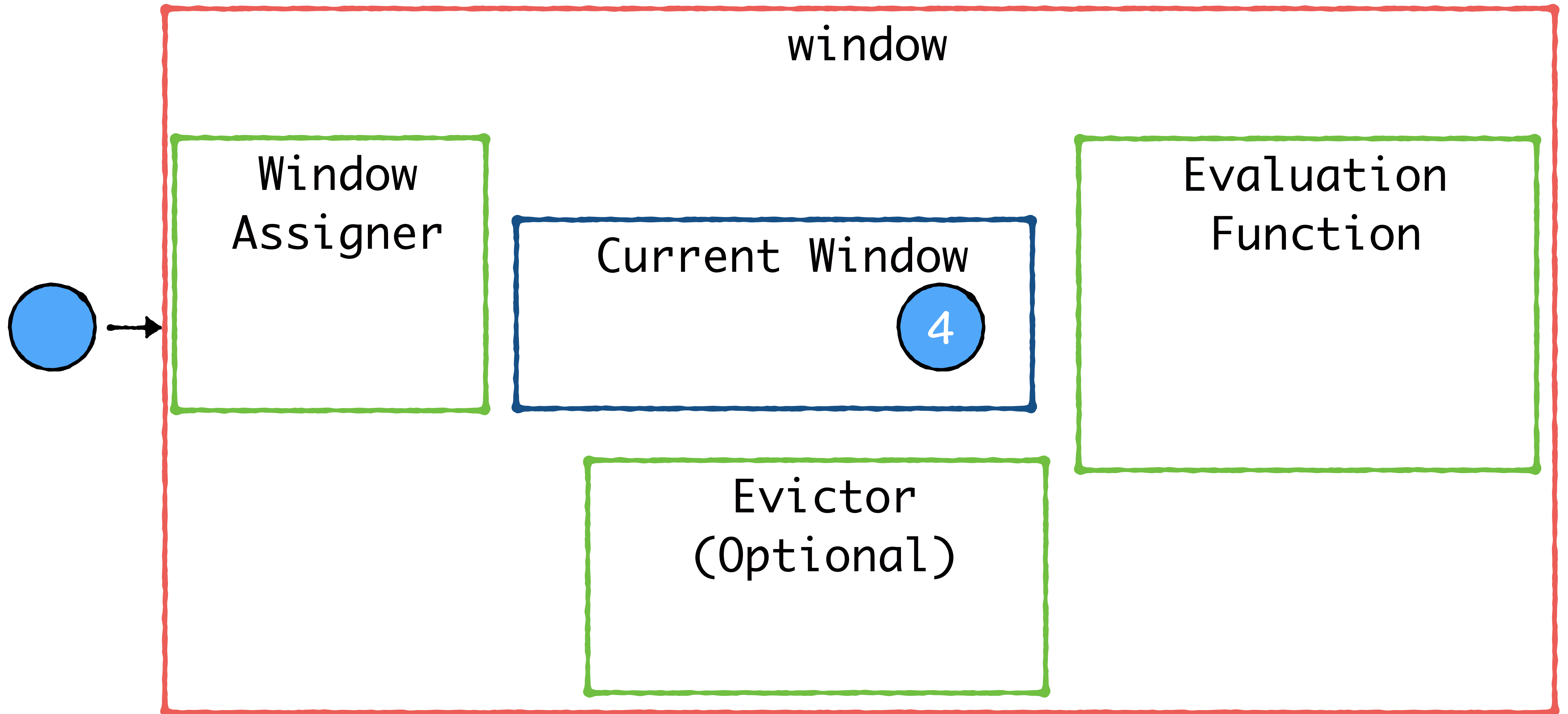


# Window API





# Window API



# Window API

Window Assigner

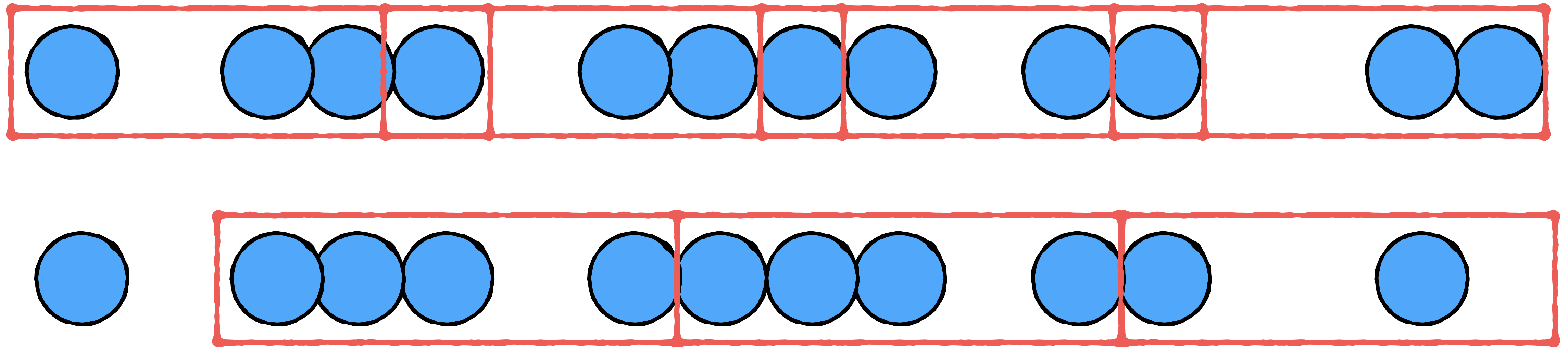
Trigger

Evictor (Optional)

Evaluation Function

Customize  
any of these  
components

# Tumbling and Sliding Windows



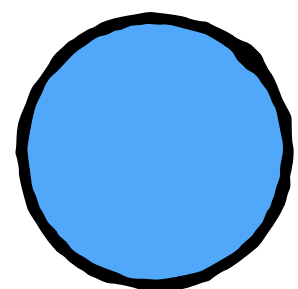
Windows based on time

# What is time?

Source

Flink Entry

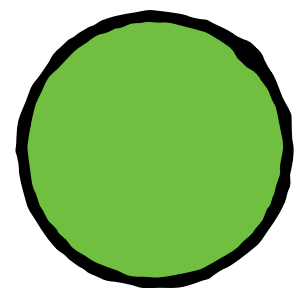
Flink Task



0

2

4



1

2

3

# What is time?

**Event time** When the event occurred

**Ingestion time** When the event is first seen

**Processing time** When the event is processed

# What is time?

Event time

Ingestion time

**Processing time**      The default choice

# What is time?

Event time

Ingestion time

Processing time

# Event time

When the event occurred at the source

Embedded within records

Useful when events are out of order/  
late



# Event time

Add a timestamp to events

TimeStampExtractor

# What is time?

Event time

Ingestion time

Processing time

# Ingestion time

When the event is first seen by Flink

Timestamp given by Flink

chronologically after the event time

Cannot handle out of order events

# What is time?

Event time

Ingestion time

Processing time

# Processing time

System time of the machine  
processing entities

Chronologically after the event time  
and ingestion time

No co-ordination required

# Iterate

