Watermarks

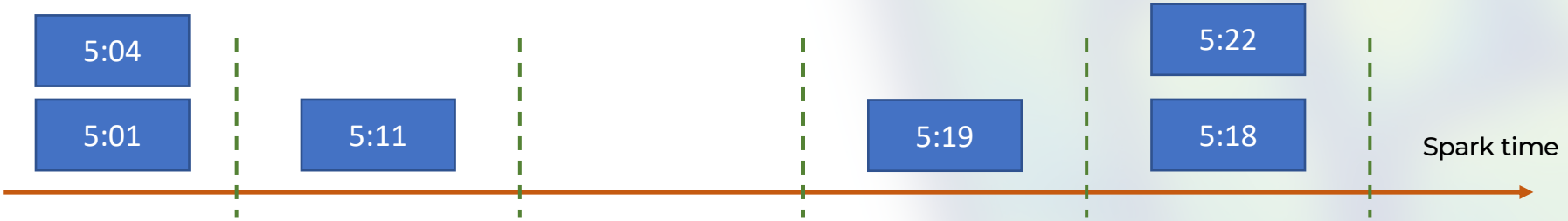
Objective

Handle late data in time-based aggregations

Window Functions

Example: count by window, in complete mode

- batch time = 10 minutes
- window duration = 20 minutes
- window sliding interval = 10 minutes



window	count
4:50 – 5:10	2
5:00 – 5:20	2

window	count
4:50 – 5:10	2
5:00 – 5:20	3
5:10 – 5:30	1

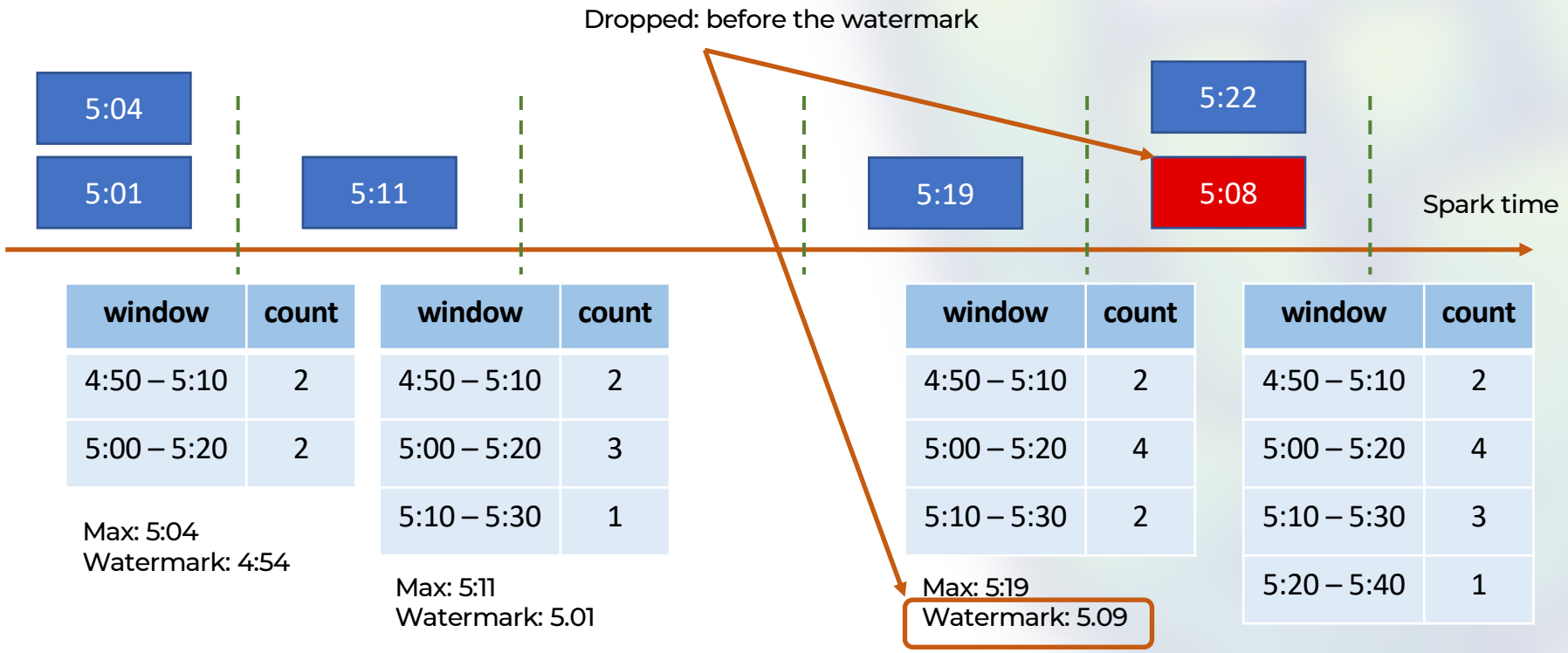
window	count
4:50 – 5:10	2
5:00 – 5:20	4
5:10 – 5:30	2

window	count
4:50 – 5:10	2
5:00 – 5:20	5
5:10 – 5:30	3
5:20 – 5:40	1

Watermark

= how far back we still consider records before dropping them

- assume watermark of 10 minutes



Takeaways

Add watermarks to a time column

```
val enhancedDF = purchasesDF.withWatermark("created", "2 seconds"))
```

With every batch, Spark will

- update the max time ever recorded
- update watermark as (max time – watermark duration)

Guarantees

- in every batch, all records with time > watermark will be considered
- if using window functions, a window will be updated until the watermark surpasses the window

No guarantees

- records whose time < watermark will not necessarily be dropped

Aggregations & joins in append mode need watermarks

- a watermark allows Spark to drop old records from state management

Spark rocks