An introduction to Apache Beam

For streaming analytics

FOSDEM 2023
Brussels, Feb 4th

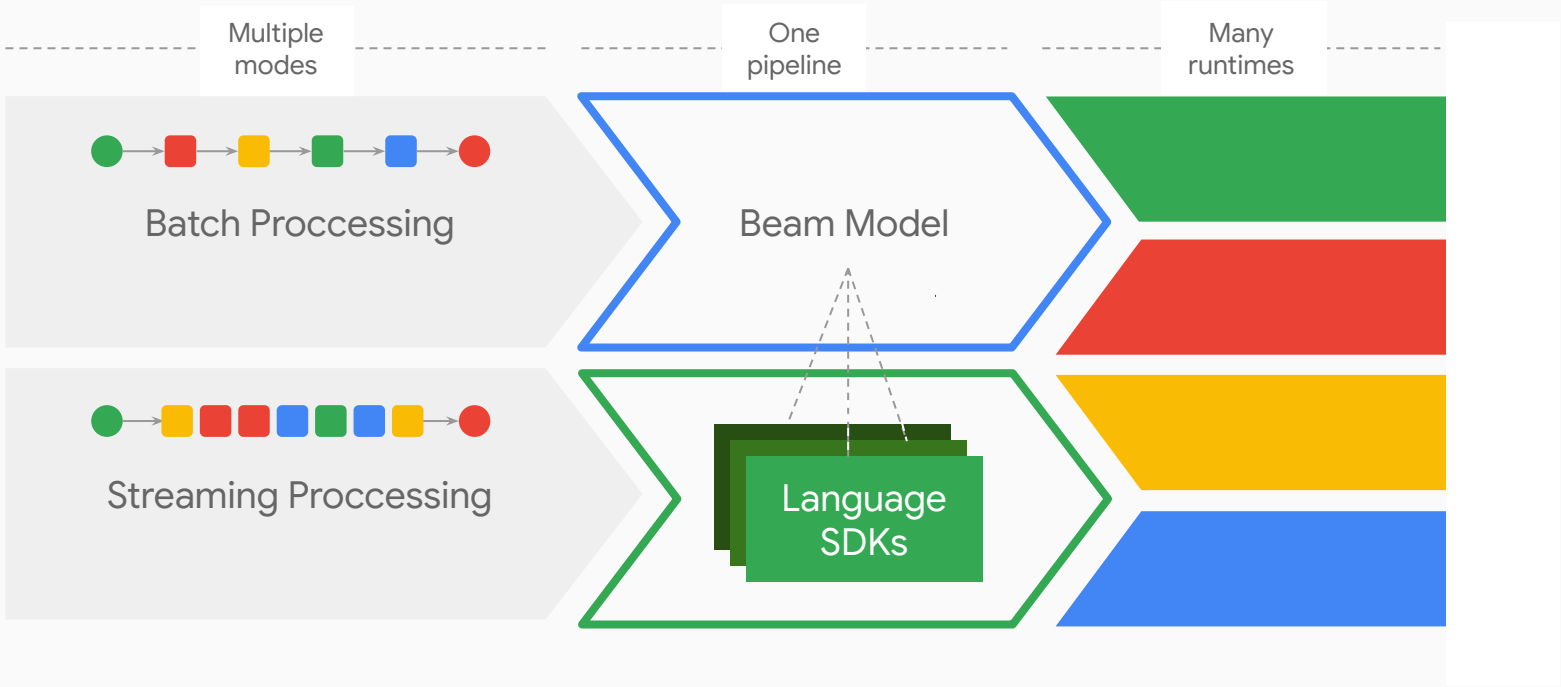
Israel Herraiz

@herraiz

Strategic Cloud Engineer
Google Cloud

Apache Beam

What is Apache Beam?



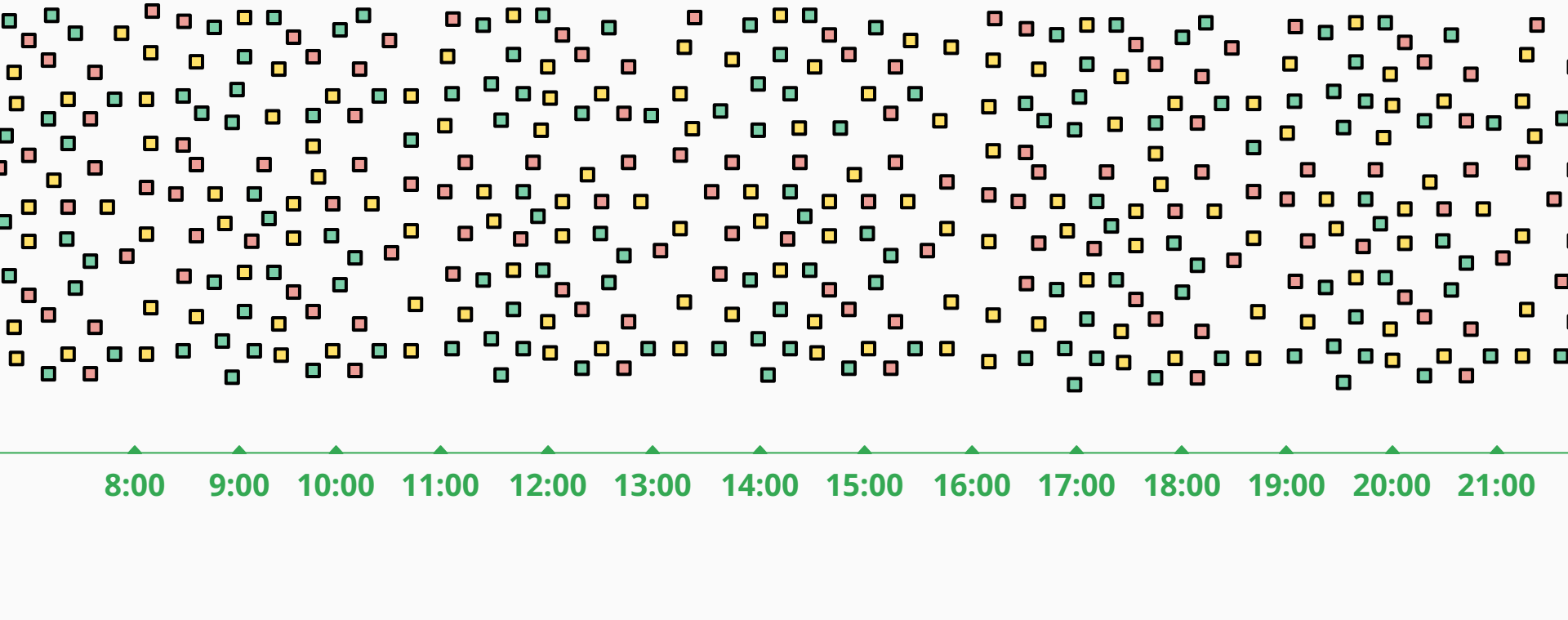
What is Apache Beam?

SDKs

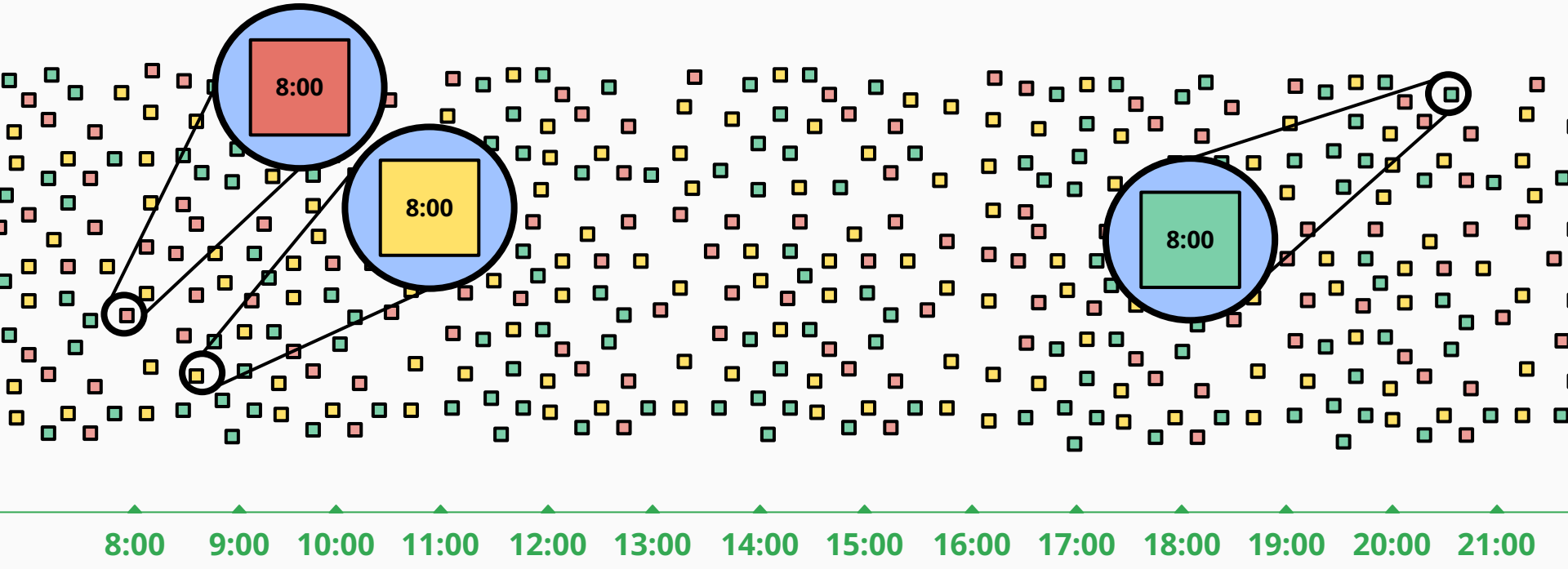
Runners

The problem with streaming

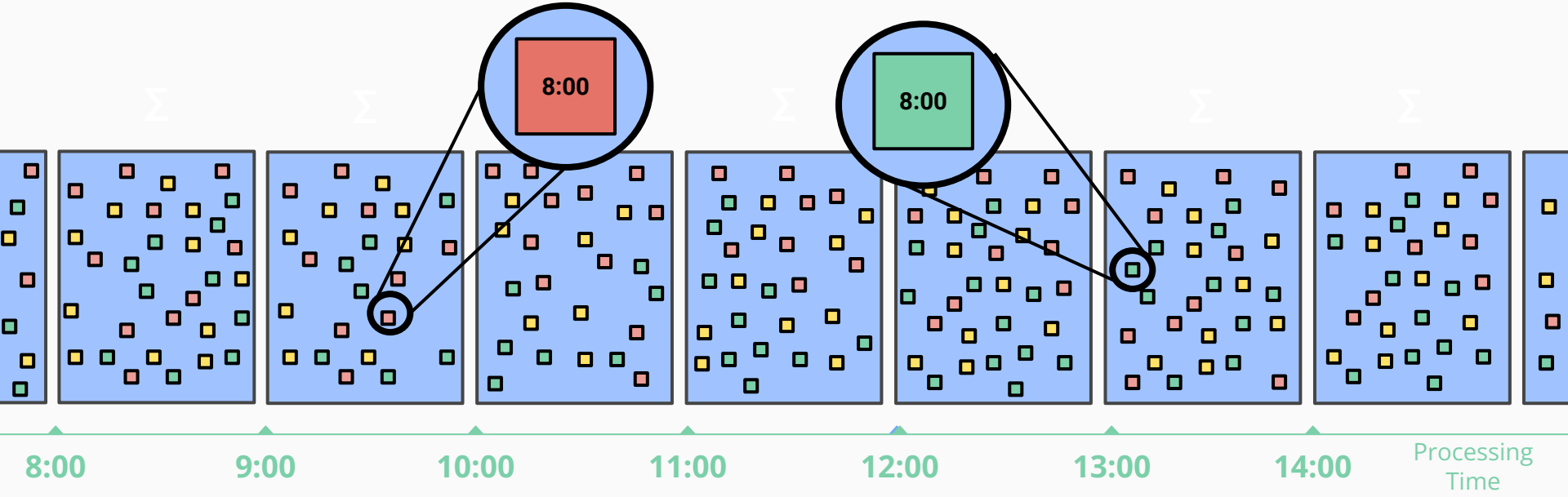
Data streams: unbounded data sources



Arrival out of order



Micro-batching does not solve the problem with out of order

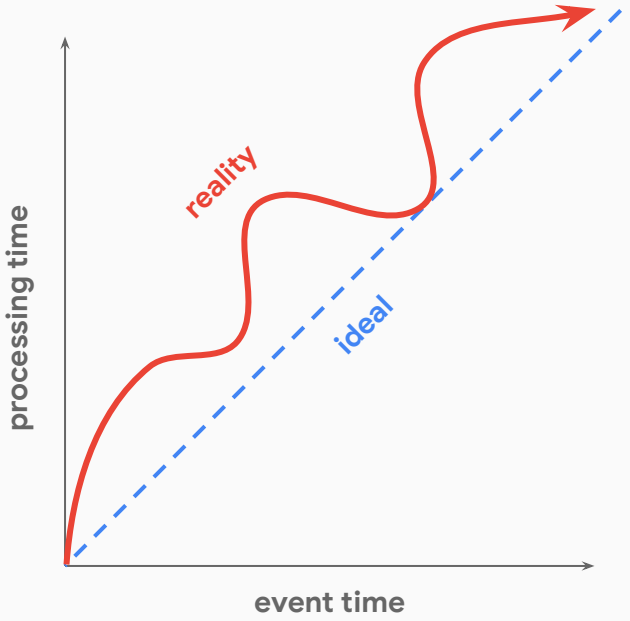


The watermark

Event time vs. processing time define the watermark

In any data processing system:

- There is a certain amount of lag between:
  - The **event time**, when a data event occurs (determined by the timestamp on the data element itself).
  - The **processing time**, when a data element gets processed at any stage in a pipeline (determined by the clock on the processing system).
- There are no guarantees that data events will appear in a pipeline in the same order that they were generated.



In other words: event time vs. processing time in Star Wars

Source: *Introduction to Apache Flink by Ellen Friedman, Kostas Tzoumas*

Dealing with out of order: Windows

Answering four questions

What results are calculated?

Where in event time are results calculated?

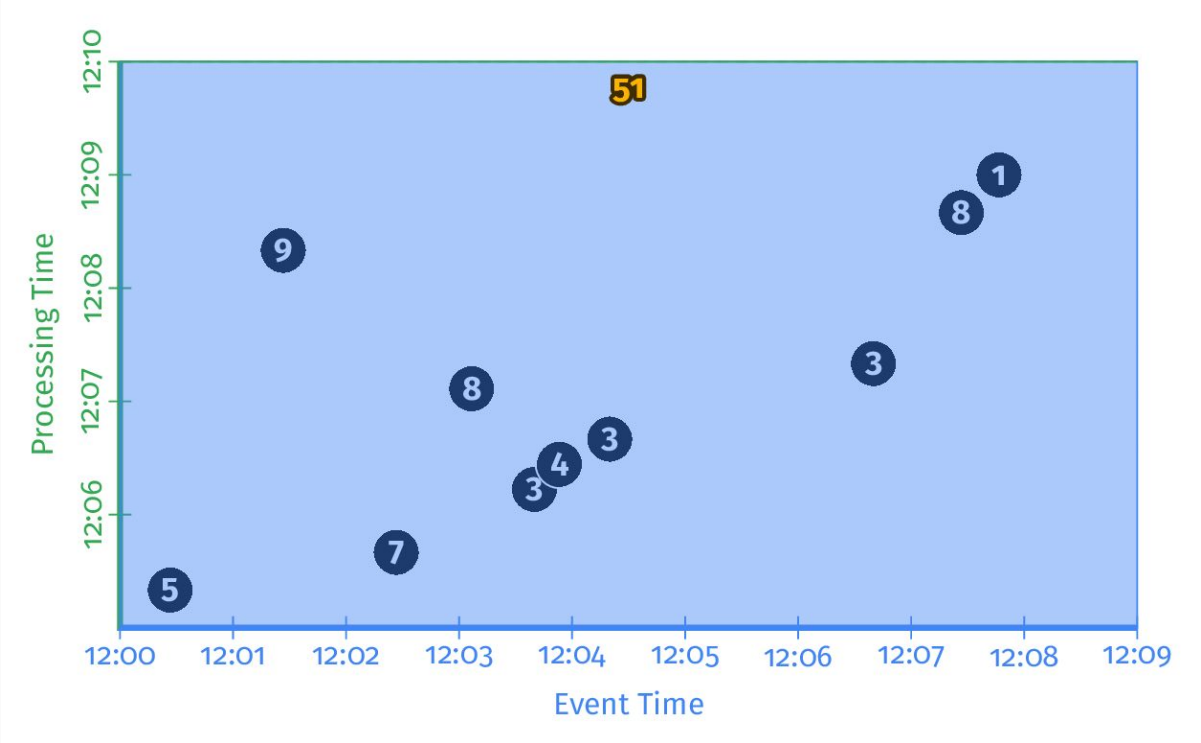
When in processing time are results materialized?

How do refinements of results relate?

The Beam Model: What is Being Computed?

```
PCollection<KV<String, Integer>> scores = input
    .apply(Sum.integersPerKey());
```

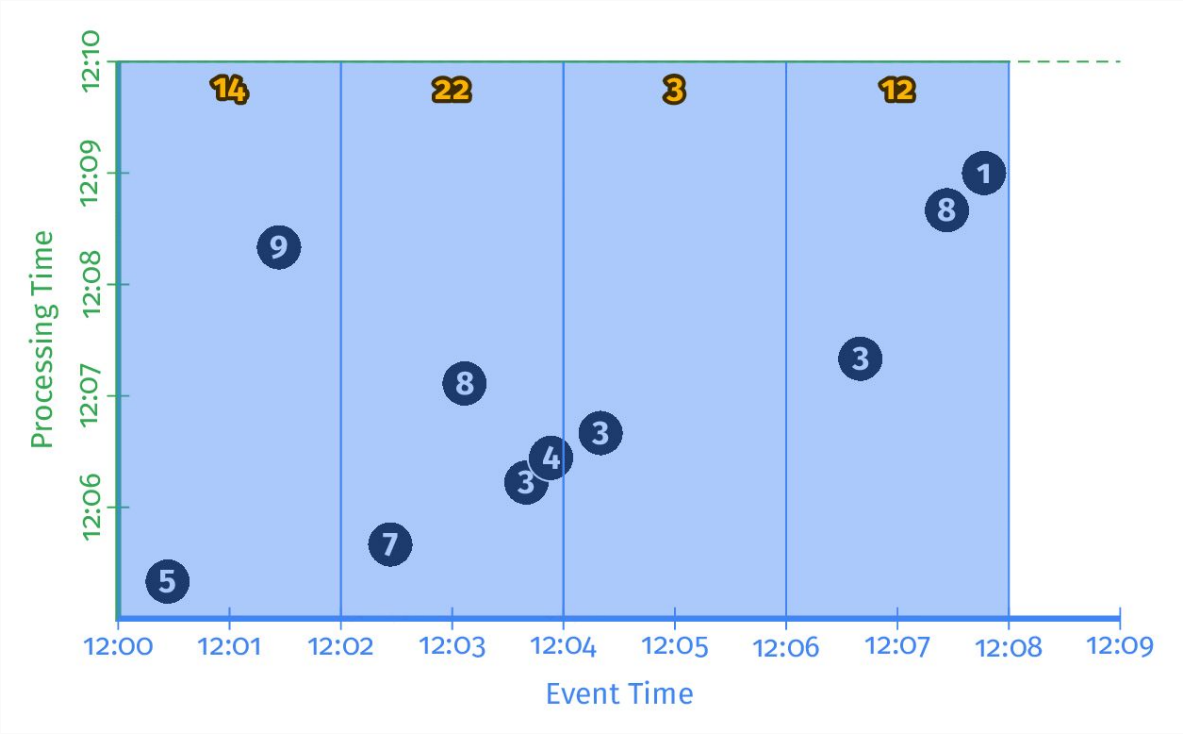
The Beam Model: What is Being Computed?



The Beam Model: Where in event time?

```
PCollection<KV<String, Integer>> scores = input
    .apply(Window.into(FixedWindows.of(Duration.standardMinutes(2))))
    .apply(Sum.integersPerKey());
```

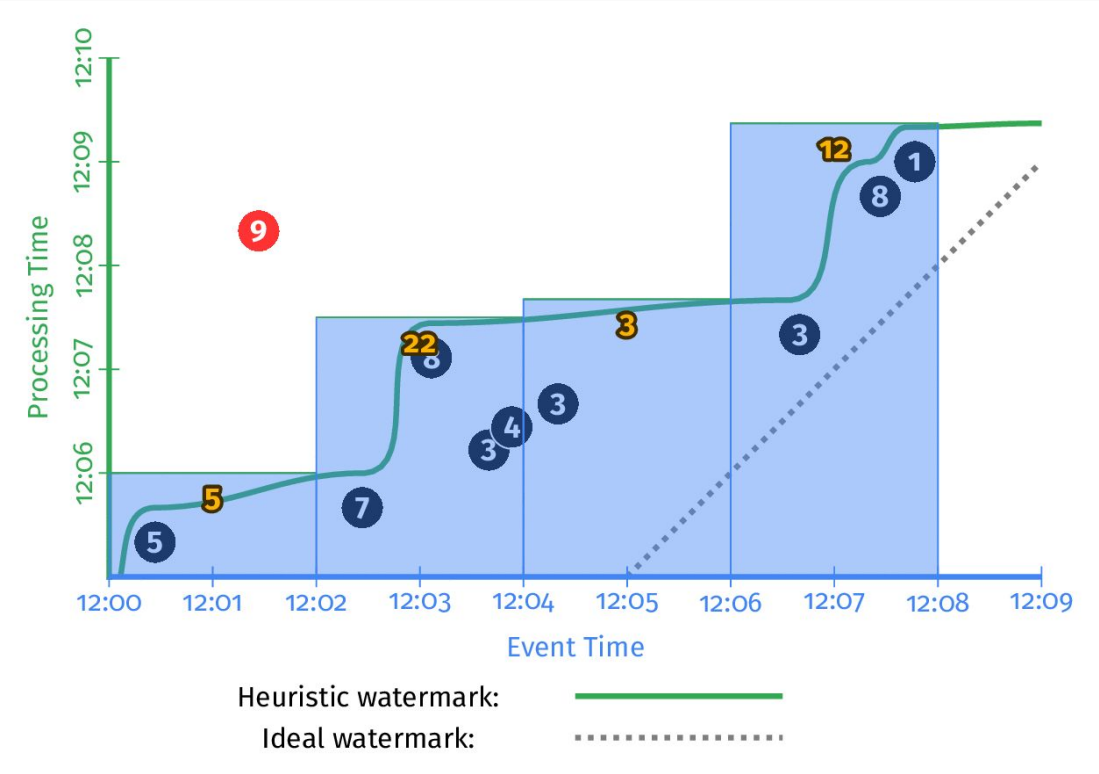
The Beam Model: Where in event time?



The Beam Model: **When** in processing time?

```
PCollection<KV<String, Integer>> scores = input
    .apply(Window.into(FixedWindows.of(Duration.standardMinutes(2))
           .triggering(AtWatermark()))
    .apply(Sum.integersPerKey());
```

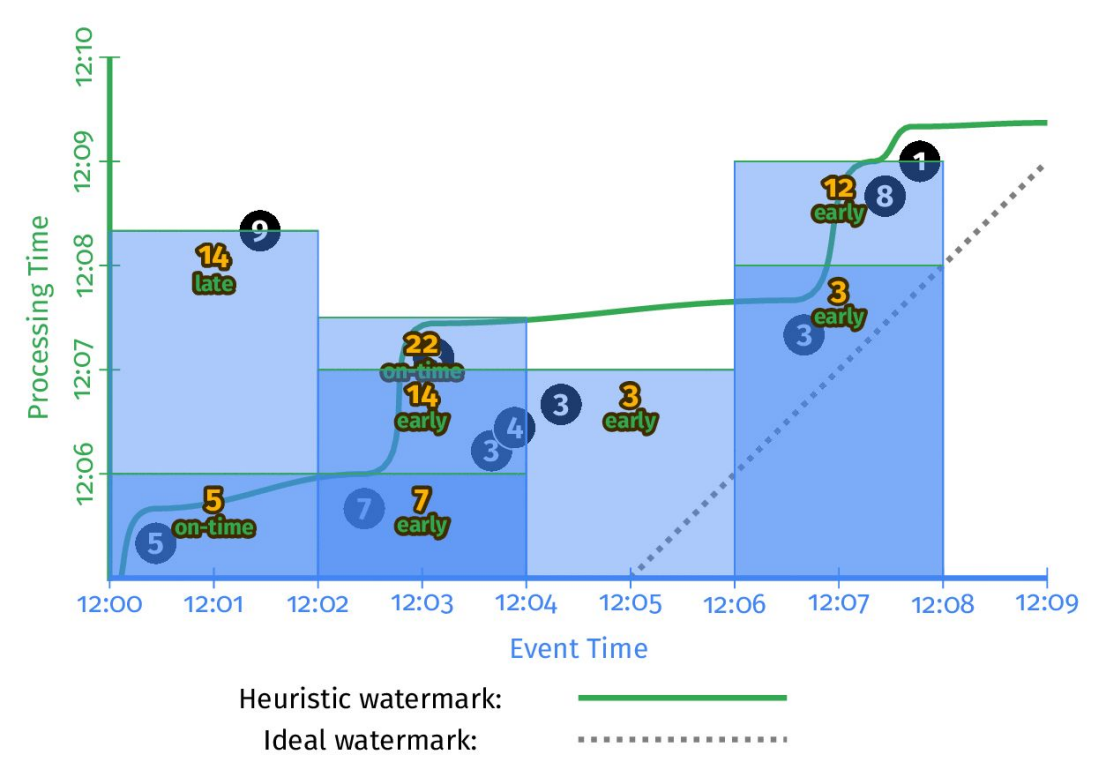
The Beam Model: When in processing time?



The Beam Model: How do we refine/recalculate?

```
PCollection<KV<String, Integer>> scores = input
    .apply(Window.into(FixedWindows.of(Duration.standardMinutes(2))
           .triggering(AtWatermark()
              .withEarlyFirings(AtPeriod(Duration.standardMinutes(1)))
              .withLateFirings(AtCount(1)))
           .accumulatingFiredPanes())
    .apply(Sum.integersPerKey());
```

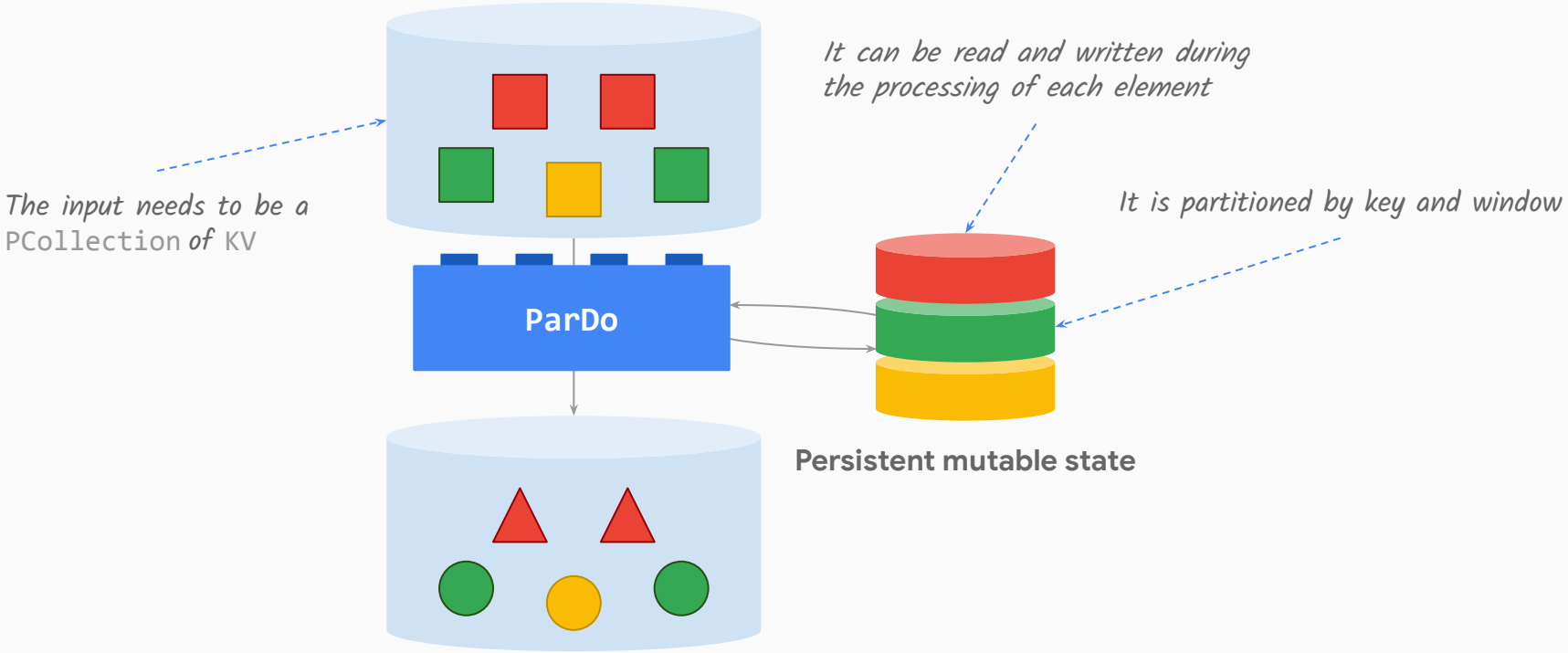
The Beam Model: How do we refine/recalculate?



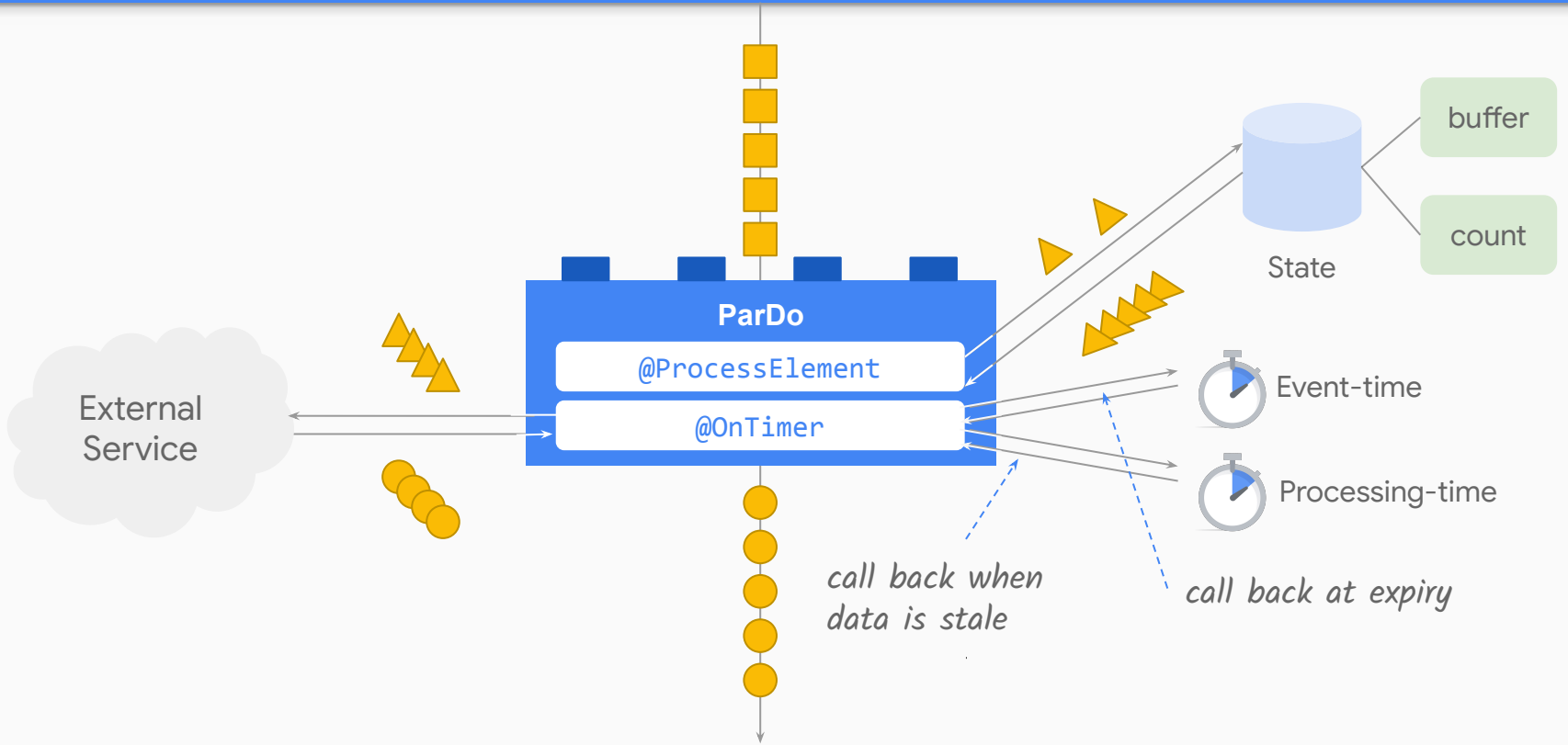
beam.apache.org/get-started/mobile-gaming-example/

Dealing with out of order: Stateful functions

Stateful processing



State variables and timers: example



beam.apache.org/blog/timely-processing/

State & timers patterns:
youtube.com/watch?v=RQjJ0BDKI_k

Using the Timer &State API to solve times series use cases
youtube.com/watch?v=Q_v5Zsjuuzg

Other goodies in streaming: ML inference at scale

ML inference

Local and remote inference

Resource hints: use GPU, specify memory requirements

```
with pipeline as p:
  data = p | 'Read' >> beam.ReadFromSource('a_source')
  model_a_predictions = data | RunInference(<model_handler_A>).with_resource_hints(min_ram="20GB")
  model_b_predictions = model_a_predictions | beam.Map(x) | RunInference(<model_handler_B>)
    .with_resource_hints( min_ram="4GB", accelerator="type:nvidia-tesla-k80;count:1;install-nvidia-driver")
```

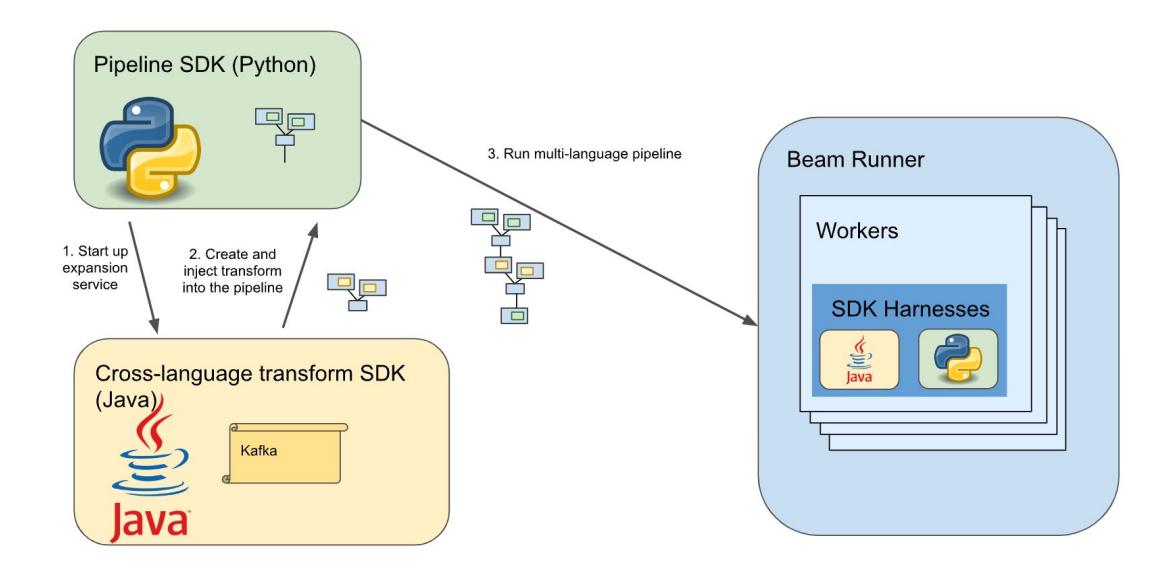
beam.apache.org/documentation/sdks/python-machine-learning/
beam.apache.org/documentation/ml/overview/

Other goodies in streaming:
In Java (or lang of choice) too!

Cross language transforms: use any transform from any SDK in any other SDK.

For instance, RunInference in Java

beam.apache.org/releases/javadoc/current/org/apache/beam/sdk/extensions/python/transforms/RunInference.html



beam.apache.org/documentation/programming-guide/#multi-language-pipelines

Recommended readings and links

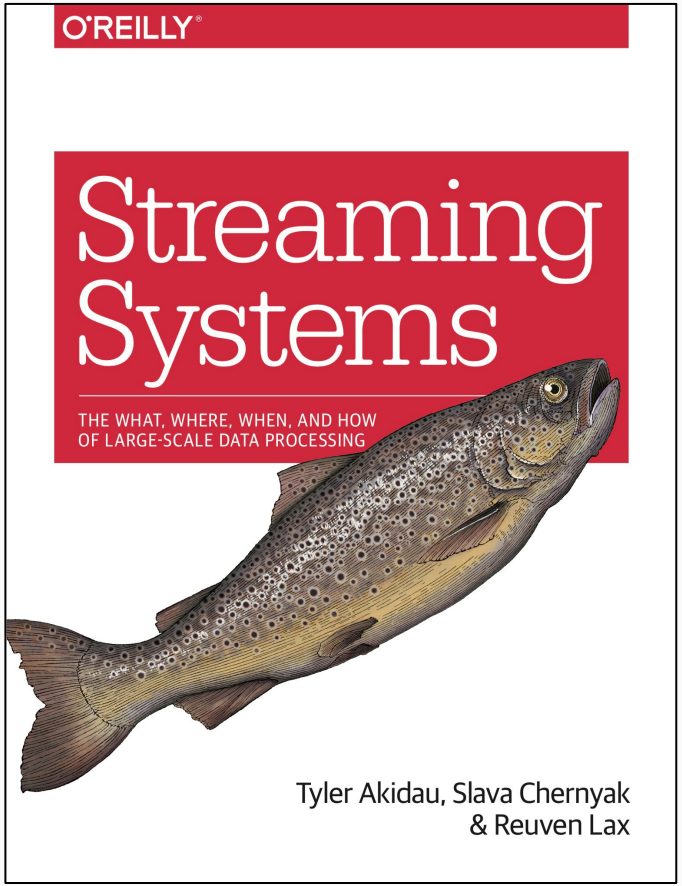
The book on Streaming Systems

Tyler Akidau, Slava Chernyak, Reuven Lax

http://streamingsystems.net/

"If you care about the correctness of your streaming and batch processing jobs, this book is a must-read. It provides the most clear-thinking and logical discussion of the topic that I have seen, and its ideas are brilliantly explained."

—Martin Kleppmann
University of Cambridge



beamcollege.dev/

BEAM SUMMIT

beamsummit.org/

beam.apache.org/

* this presentation was prepared with some of the Beam community materials available at **https://beam.apache.org/community/presentation-materials/**