BIG DATA

Procesado en Tiempo Real
Streaming / Batching

MSc. Felipe Meza

Contenidos

- Introducción
- Batching
- Streaming
- Proceso basico para procesado de datos.
- Hands-On

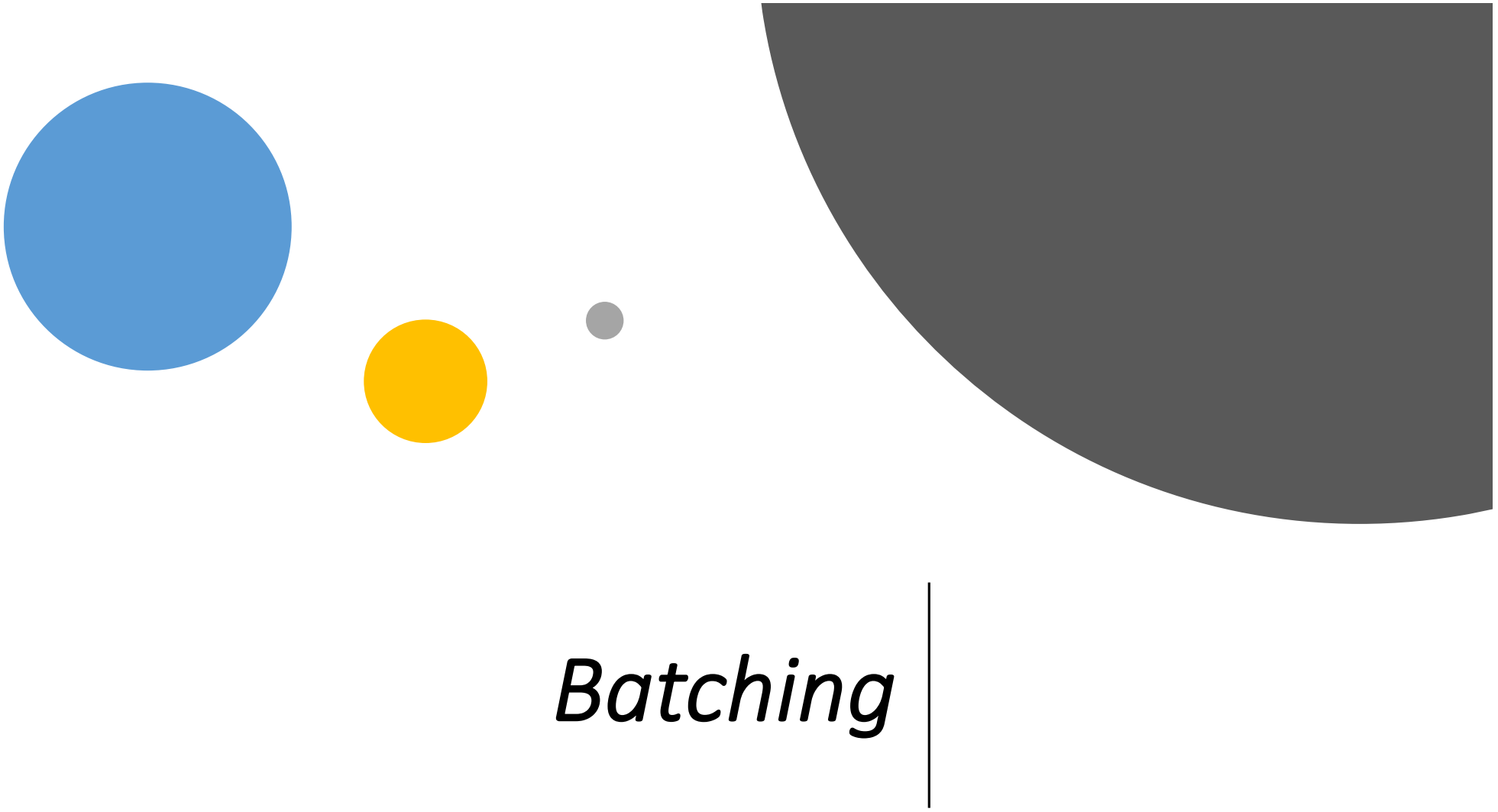
Introducción

Big Data en tiempo real…

- Hoy en día, la necesidad de tener procesado/análisis en tiempo real, es una de las ventajas competitivas más fuertes en los negocios.
- El objetivo: obtener información útil para la toma de decisiones.
- Las fuentes más comunes son: plataformas, dispositivos, aplicaciones y logs.

Big Data en tiempo real…

- Requieren de técnicas de acceso a los datos más eficientes que las convencionales.
- Las dos más populares son:
  - Batching.
  - Structured Streaming.
- Analogía para comprender el concepto:
  - Datos = Océano.
  - Batching = Cubetas de agua del Océano de diferentes tamaños.
  - Streaming = Tubería que extrae agua del Océano de manera continua.

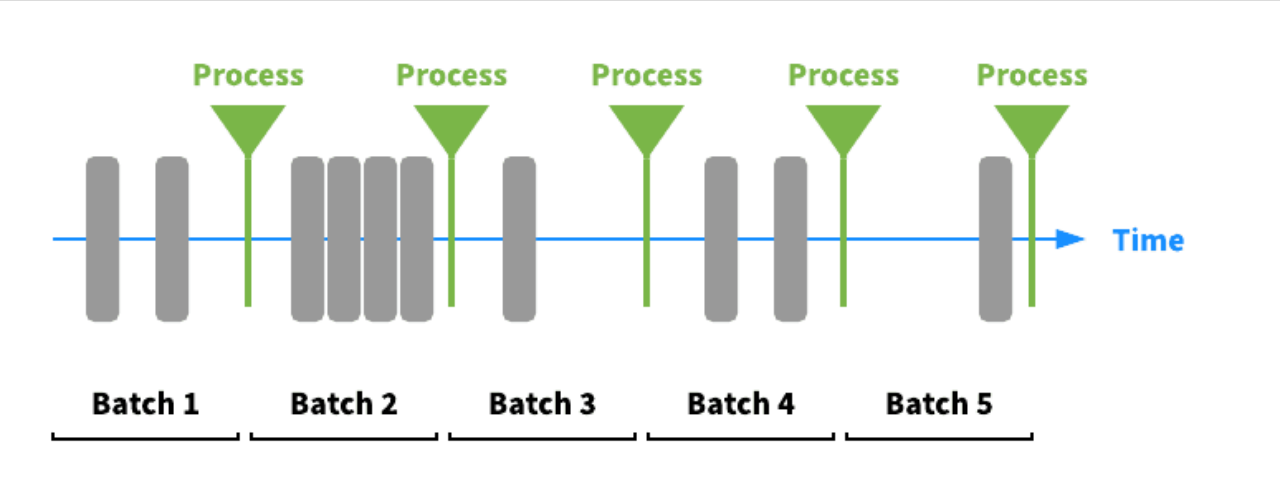


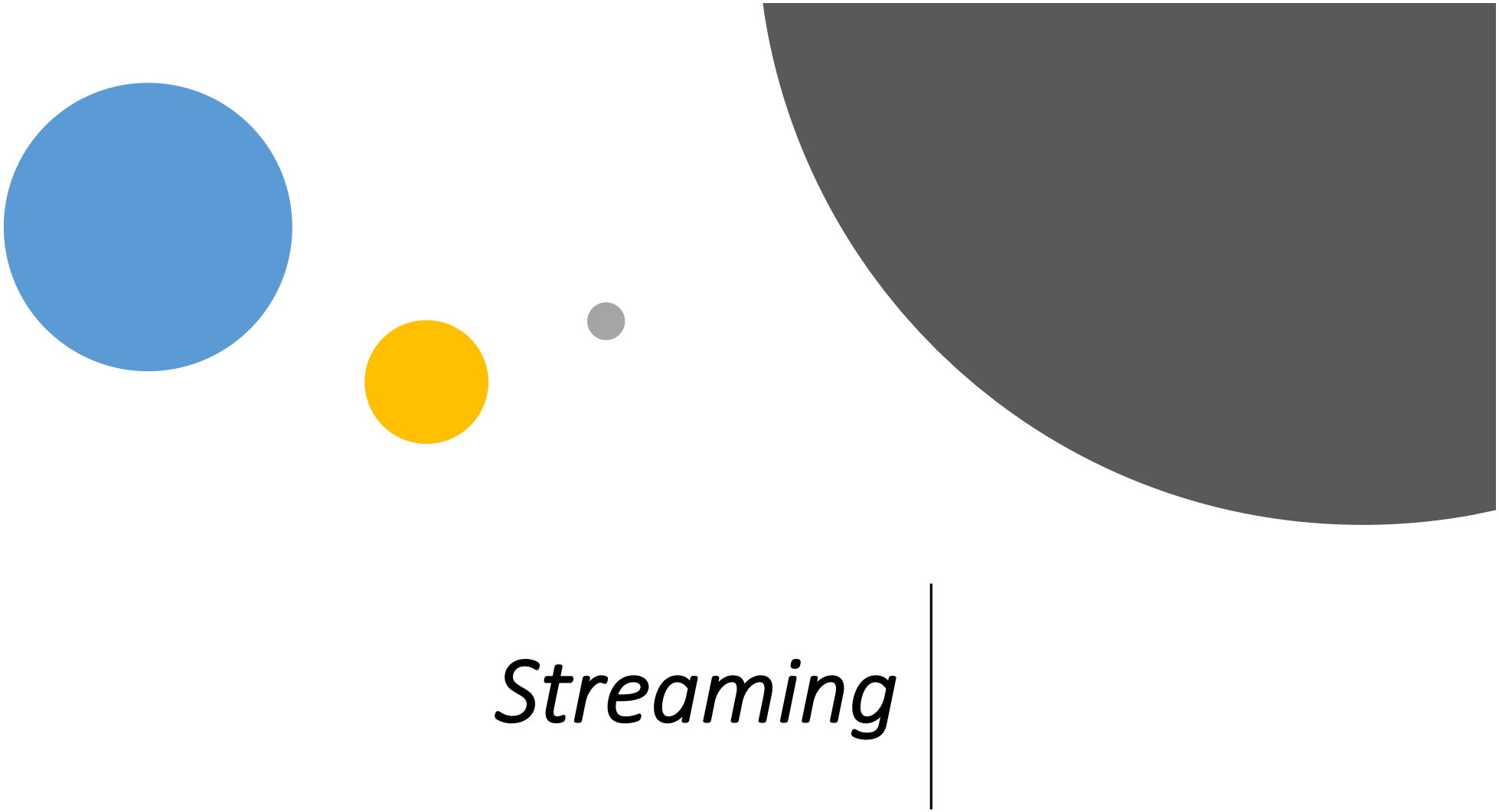
Batching

Batching

- Los datos se organizan en grupos de tamaño específico y se usan posteriormente para análisis y procesado.
- Algunos ejemplo de uso de batching:
  - En sistemas legados.
  - En BD-SQL.
  - Mainframes.
- La mayor diferencia cuando se compara con streaming, radica en que con batching los datos no son procesados conforme estos van llegando.

Batching

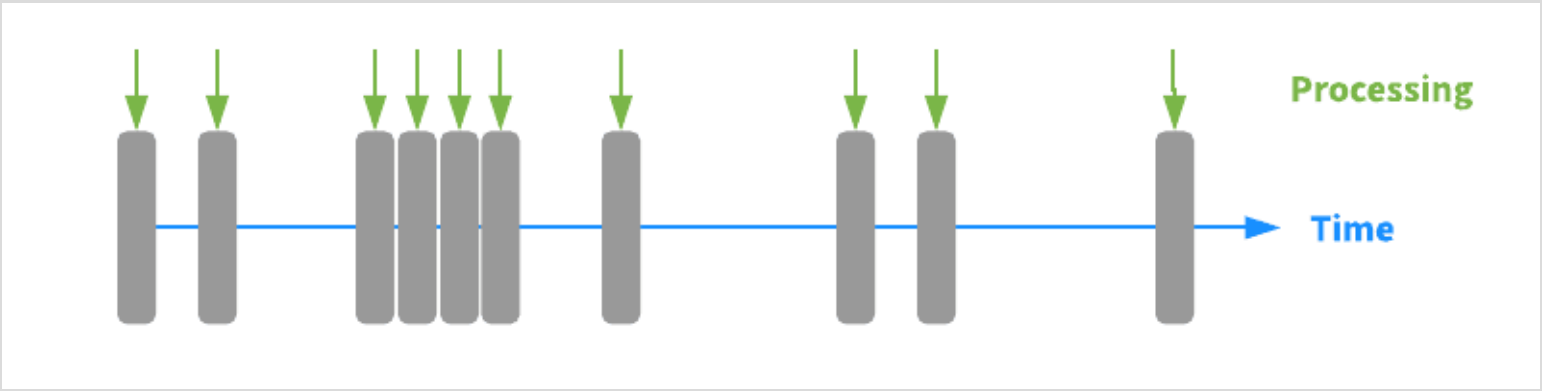

Process
Process
Process
Process
Process
Time
Batch 1
Batch 2
Batch 3
Batch 4
Batch 5


Streaming

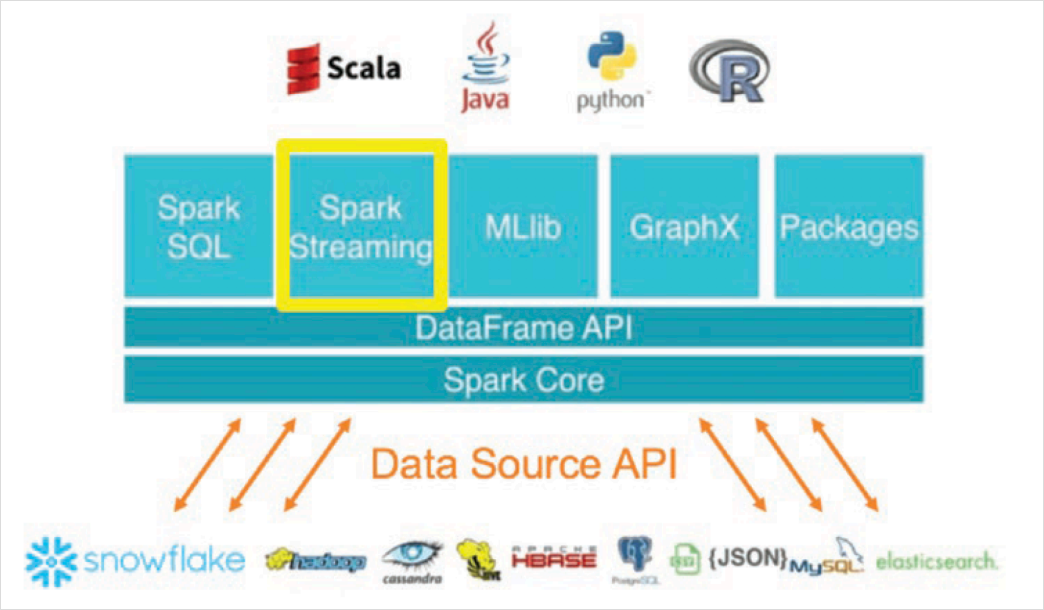
Streaming

- Procesado de datos en tiempo real (~near RT).
- No hay que esperar que los datos estén disponibles, se procesan/analizan conforme vayan estando disponibles (o de acuerdo a ventanas de tiempo).
- Como resultado, se obtiene una tabla de tamaño infinito, con datos agregados constantemente.
- Esta técnica ofrece las ventajas de procesado en tiempo real para aplicaciones críticas e.g detección de fraudes.

Streaming



Spark Streaming



Spark Streaming (versiones previas)

- API conocido como Dstream.
- Basado en RDDs.
- Tenia limitaciones.
- Recibía datos de varias fuentes y lo convertía en micro-batches.
- Procesado en Spark Engine.

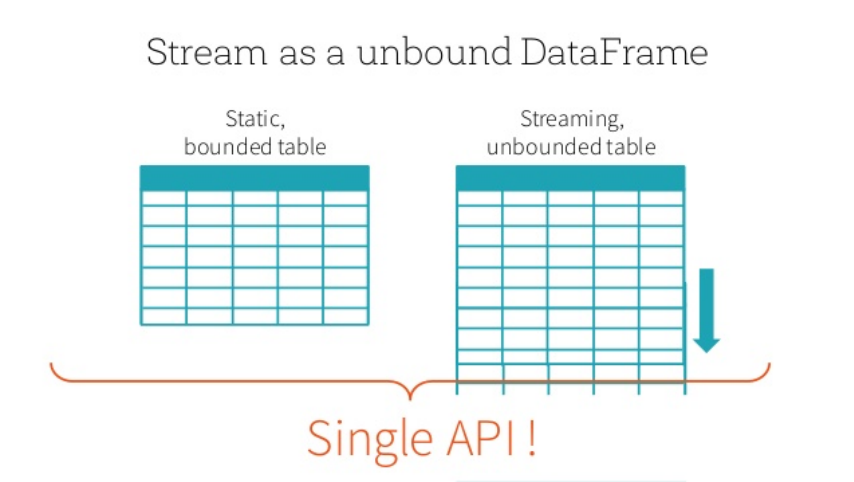
- Cada micro-batch es RDD, basado en ciertos intervalos de tiempo.

Spark Streaming (versiones previas)

- Muy poderoso en términos de manejo de datos para posterior procesado.
- Limitantes en términos de:
  - API para batch y streaming son distintos, por lo que se requieren muchos cambios para convertir de batch a Dstream.
  - Para le procesado NO maneja el concepto de “event time” solo “batch time”, genera problemas con los datos que llegan tardíos.
  - La tolerancia a fallas es muy limitada.

Structured Streaming (2016)

- Mejora sobre el modelo basado en RDD.
- La misma API para batch y streaming.

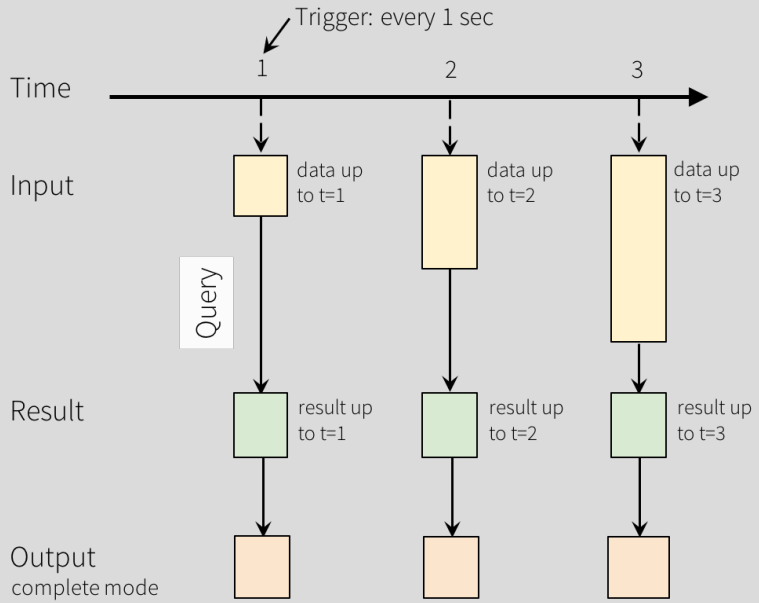


Structured Streaming (2016)

- Esta construido sobre el SparkSQL engine y usa DataFrame para múltiples operaciones (agregación, filtrado etc).

- Brinda garantía en la consistencia de los datos (tolerancia a fallas).

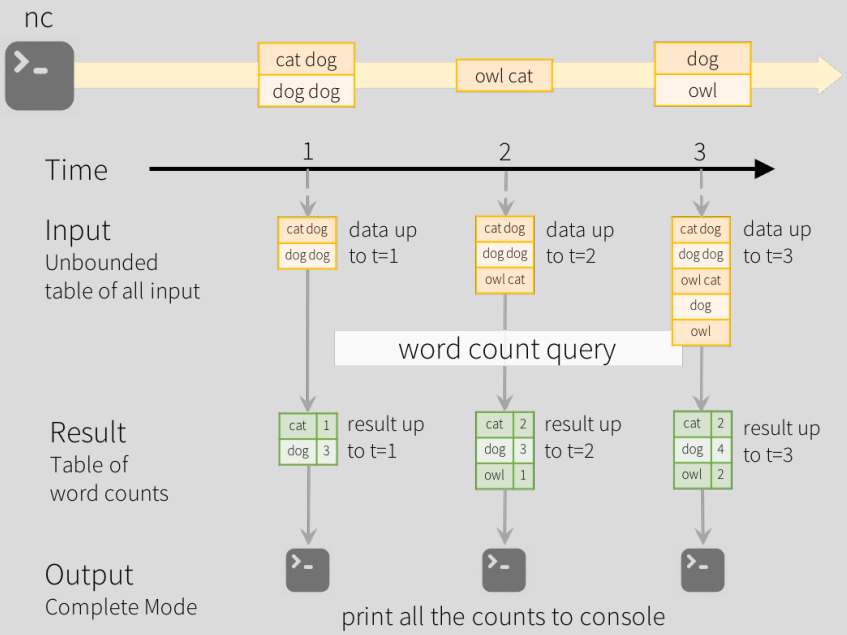
Structured Streaming (operación)



Programming Model for Structured Streaming

- DataFrame (t=1) → resulting in a larger DataFrame (t=2).
- Query execution → processing, transformation, join, aggregation.
- Output → console, memory, location.

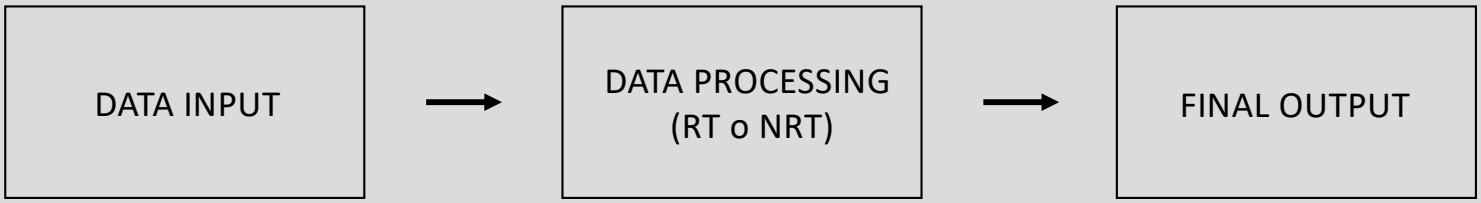
Structured Streaming (operación)



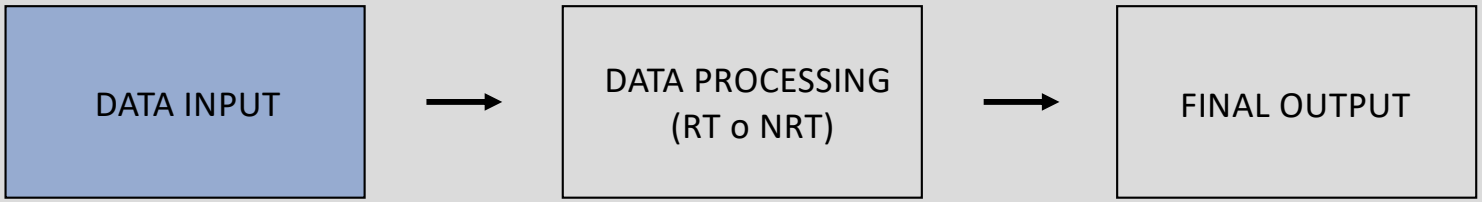
Model of the Quick Example

Structured Streaming (proceso básico)

- Desde a perspectiva del procesado de datos, se cuenta con un proceso definido que consta de 3 partes:

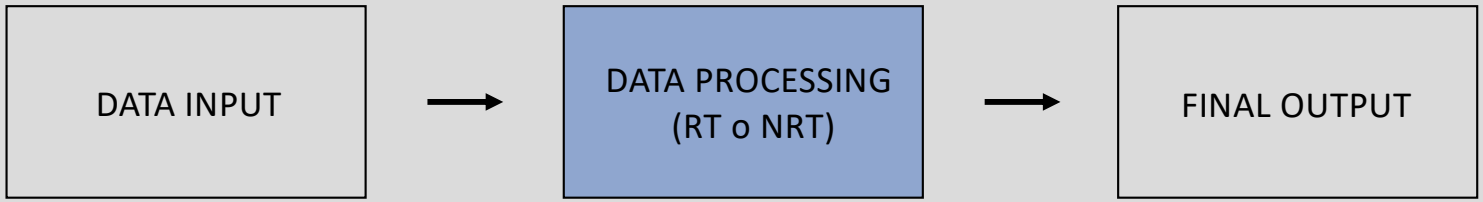


Structured Streaming (proceso básico)



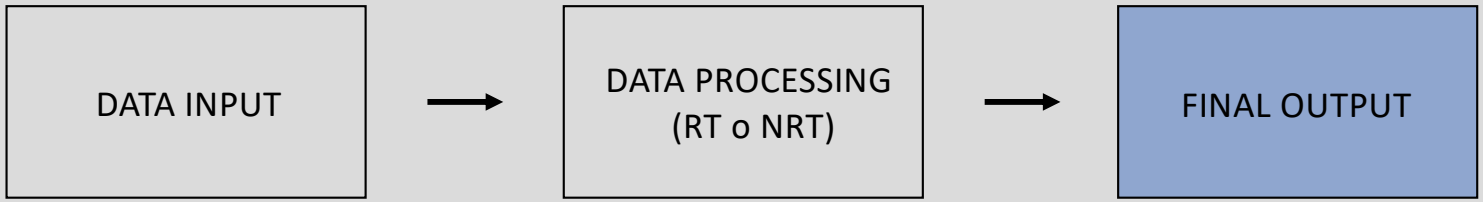
- Existen varias formas de introducir datos a un sistema de Big Data en tiempo real, entre los más usados destacan:
  - ***Messaging systems***: Tienen la tarea de capturar todos los datos generados por alguna fuente (aplicación web, aplicación móvil, dispositivo, IoT etc.), para luego colocar estos datos en la plataforma de *Structured Streaming*. Ejemplos: Apache Kafka, Flume, and Logstash.
  - ***File folders/directory***: Los archivos son leídos continuamente del directorio como un "stream" de datos, generalmente se usan calendarizadores para la colocación de nuevos archivos. El formato de los archivos puede ser texto, Parquet o JSON.

Structured Streaming (proceso básico)

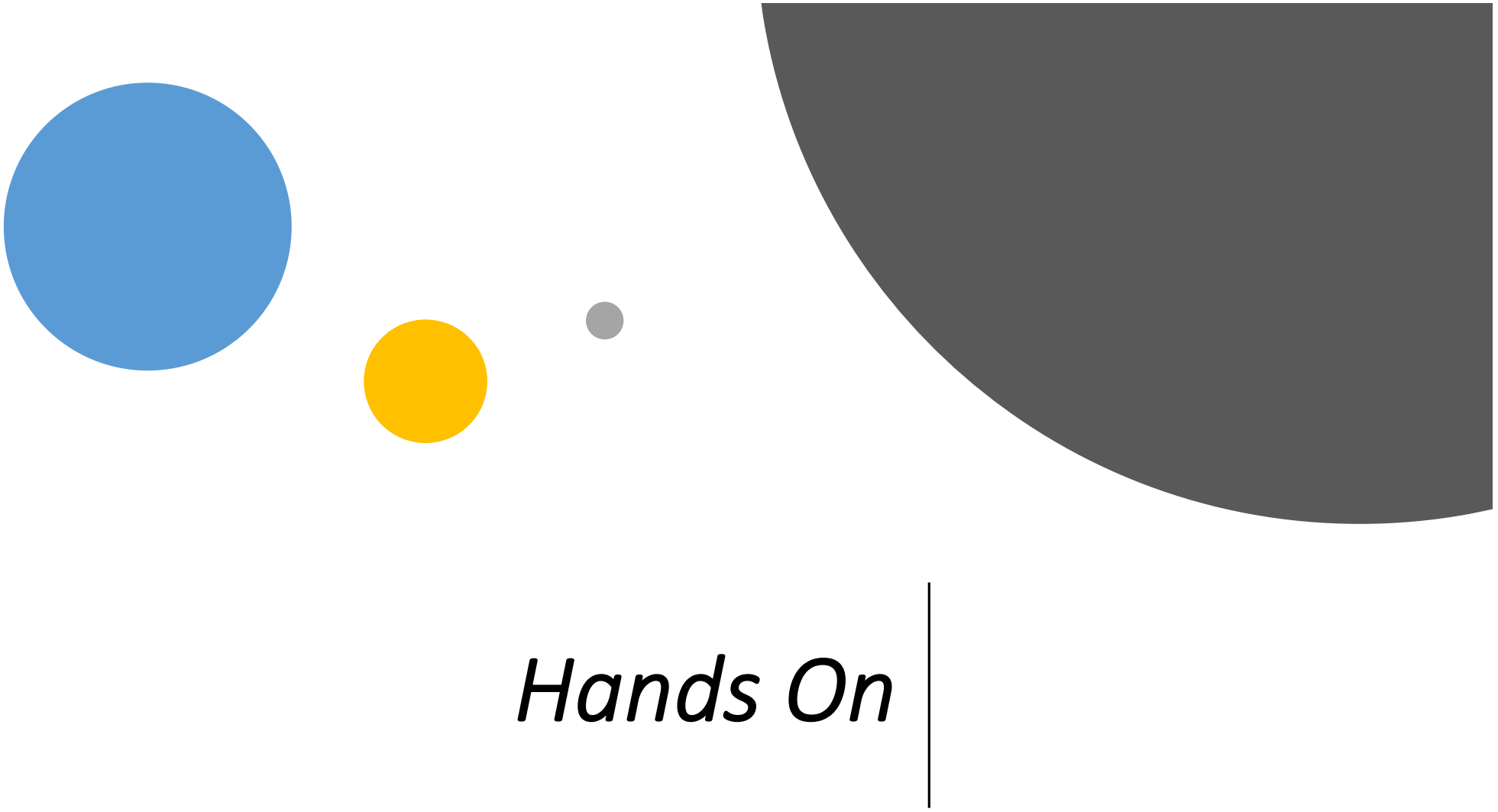


- En esta etapa se llevan a cabo los procesos en los datos, puede incluir agregaciones, filtrado, joins y ordenamiento entre otros.

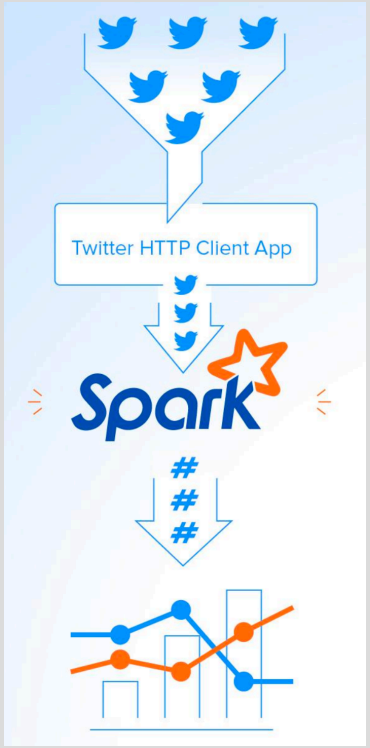
Structured Streaming (proceso básico)



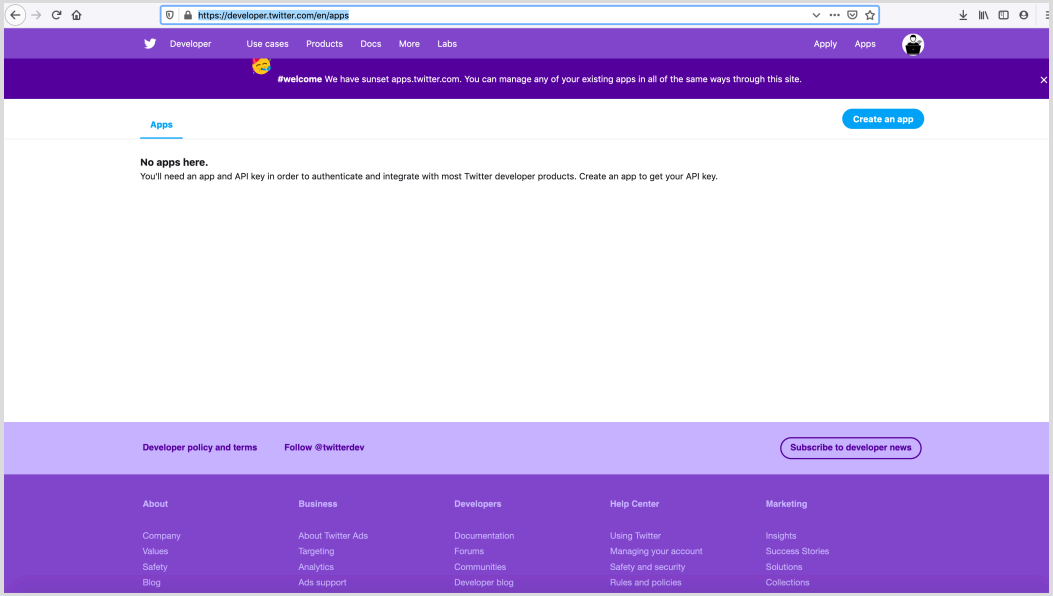
- En esta etapa se guardan los resultados, hay dos modos:
  - ***Append***: Se agregan solamente los nuevos resultados.
  - ***Complete***: Se actualiza la fuente de salida completa (e.g tabla de salida) con el resultado.


Hands On

CASO DE USO (analizando información en twitter)



https://developer.twitter.com/en/apps



TAREA # 4

Investigar las siguientes tecnologías:

- FlumeJava

- Flink Streaming
- Beam Streaming
- DataFlow