Работа с большими данными

Содержание курса

01 Введение. Распределенные вычисления. MapReduce.

02 HDFS. Apache Spark. RDD.

03 Базы данных. Spark SQL. Хранение больших данных.

04 Подробнее о модели вычислений Spark. Знакомство со Scala.

05 Алгоритмы машинного обучения на больших данных. spark.ml.

06 Рекомендательные системы. Виды. Их метрики. Spark.ML

07 Обработка потоковых данных. Structured streaming и интеграция с spark.ml.

08 Модели в продакшен. Управление кластером.

7. Обработка потоковых данных. Structured Streaming и интеграция с spark.ml.

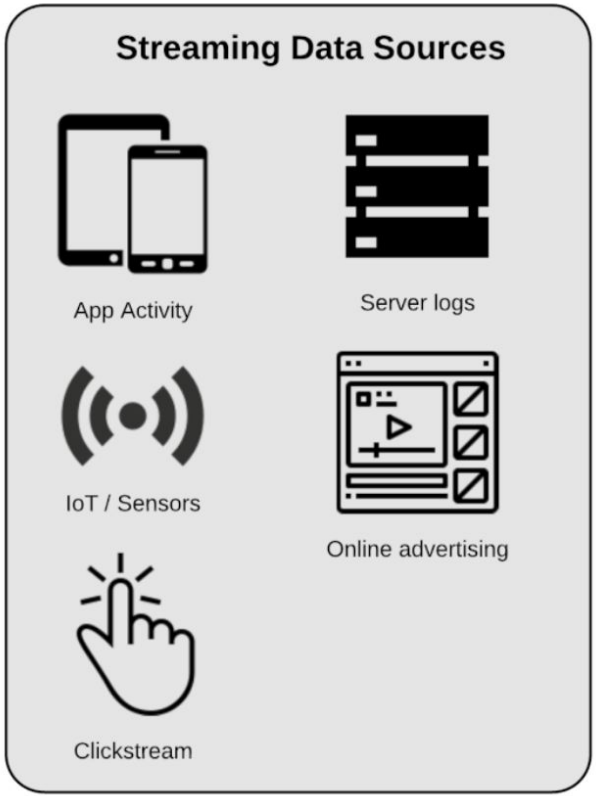
План:

1. Потоковые данные, где их искать.
2. Data Lake vs Data Warehouse.
3. Сценарии отказа. Обработка сценариев потери. Проблема византийских генералов.
4. Лямбда-архитектура.
5. Spark Streaming & DStream. Structured Streaming.
6. Интеграция с spark.ml.

Потоковые данные где их искать.

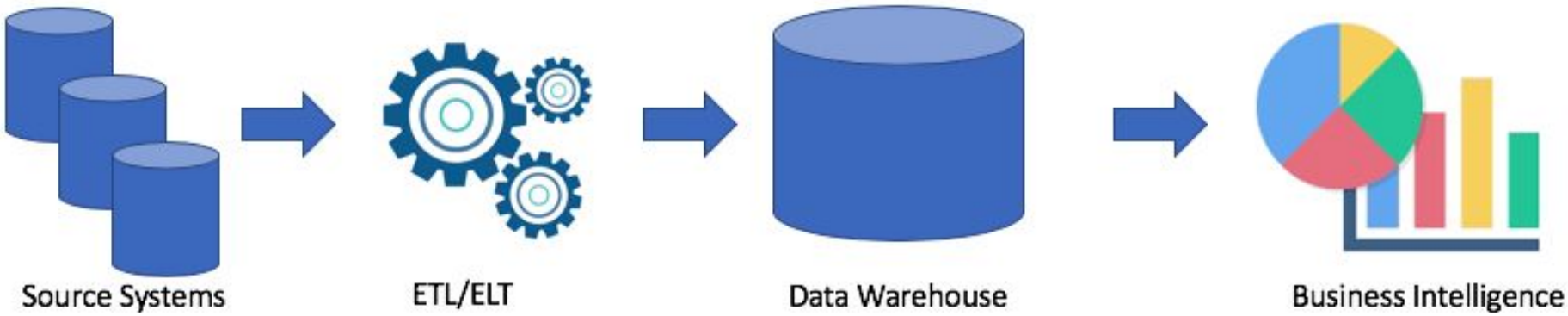
Потоковые данные где их искать.

- Датчики IoT
- Журналы серверов и логи безопасности
- Реклама в режиме реального времени
- Передача данных из приложений и веб-сайтов по кликам

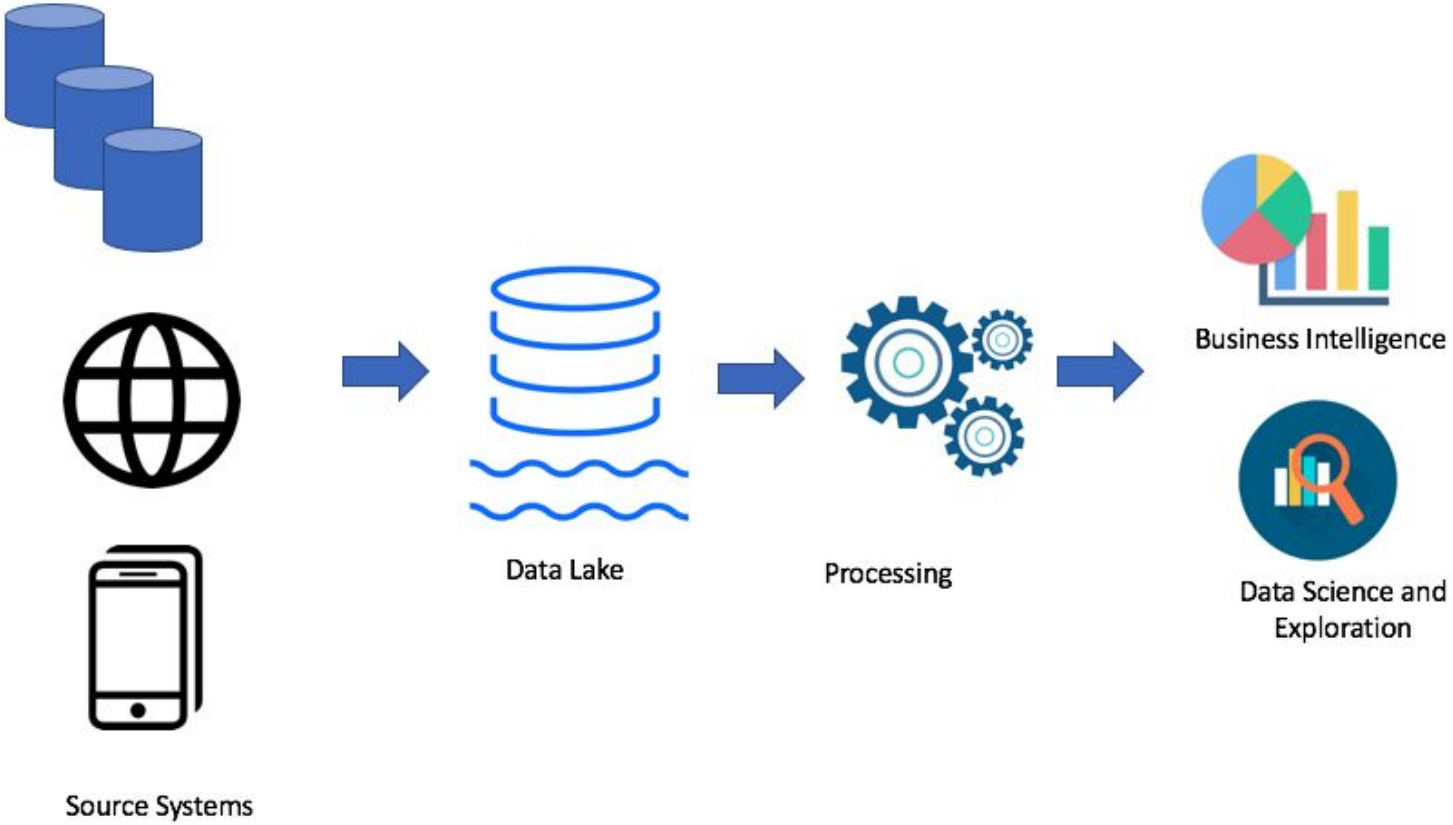


Data Lake vs Data Warehouse

Data Warehouse



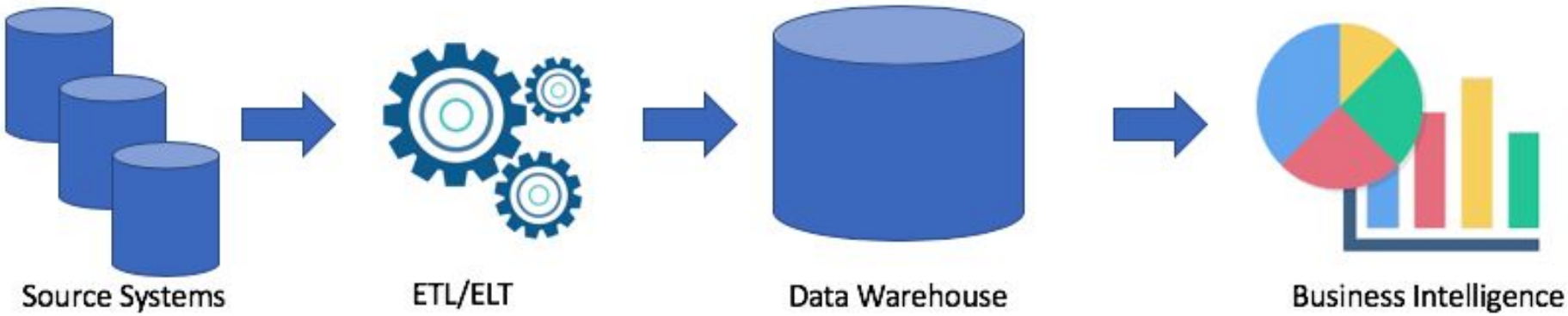
Data Lake



https://habr.com/ru/post/485180/

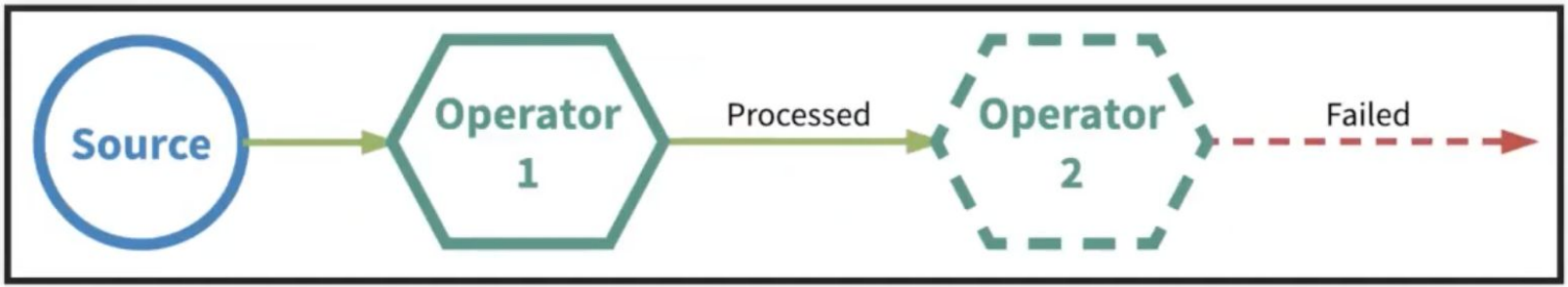
Data Lake	Data Warehouse
Complimentary to DW	Data Lake can be source for EDW
Schema on read	Schema on write
Structured/semi-structured/Unstructured	Structured data only
Fast ingestion of new data	Time consuming for new data
Advance Analytics + BI	BI use cases
Data at low level of detail/granularity	Data at summary/aggregated level
Loosely defined SLA	Tight SLAs
Flexibility in tools	Limited flexibility in tools (SQL)

Data Warehouse

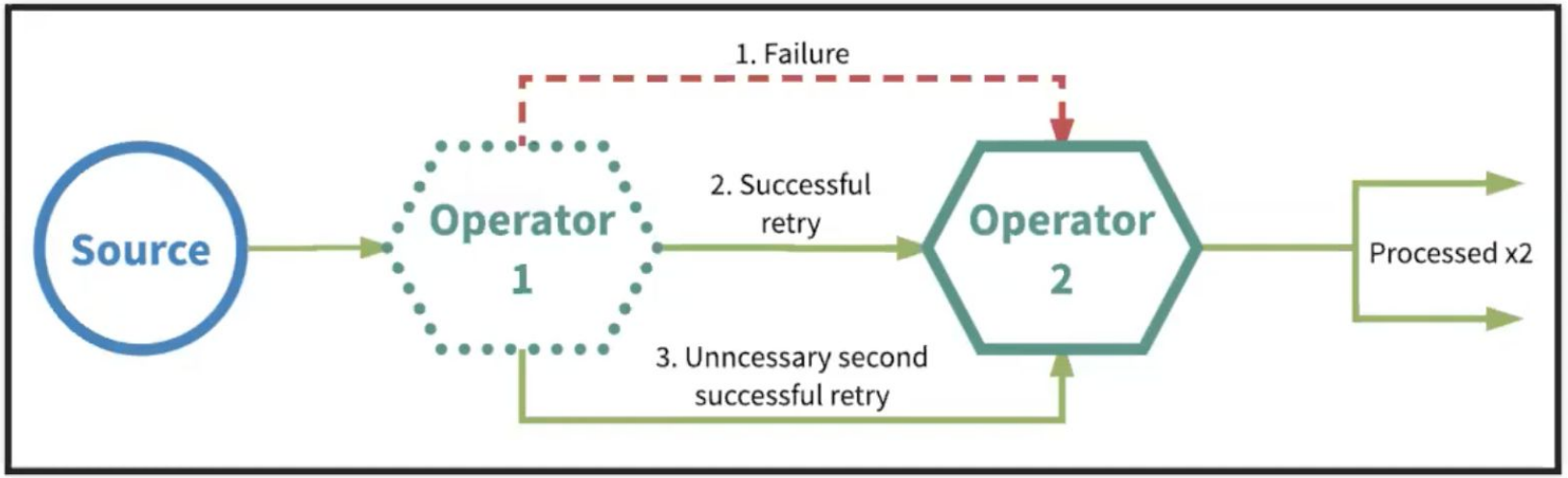


Сценарии отказа.

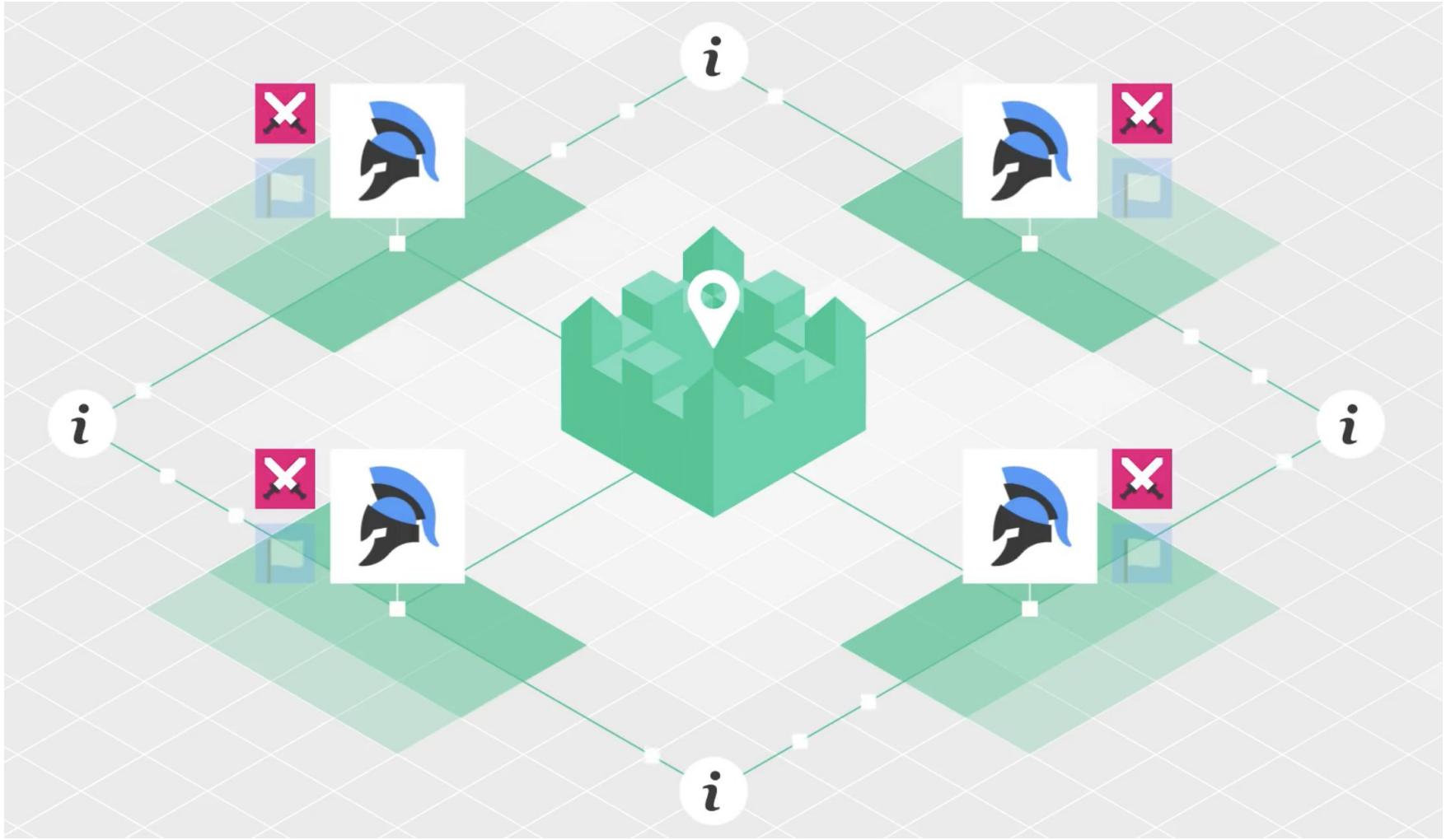
Отказ во время обработки.



Отказ во время отправки ответа о результатах обработки.



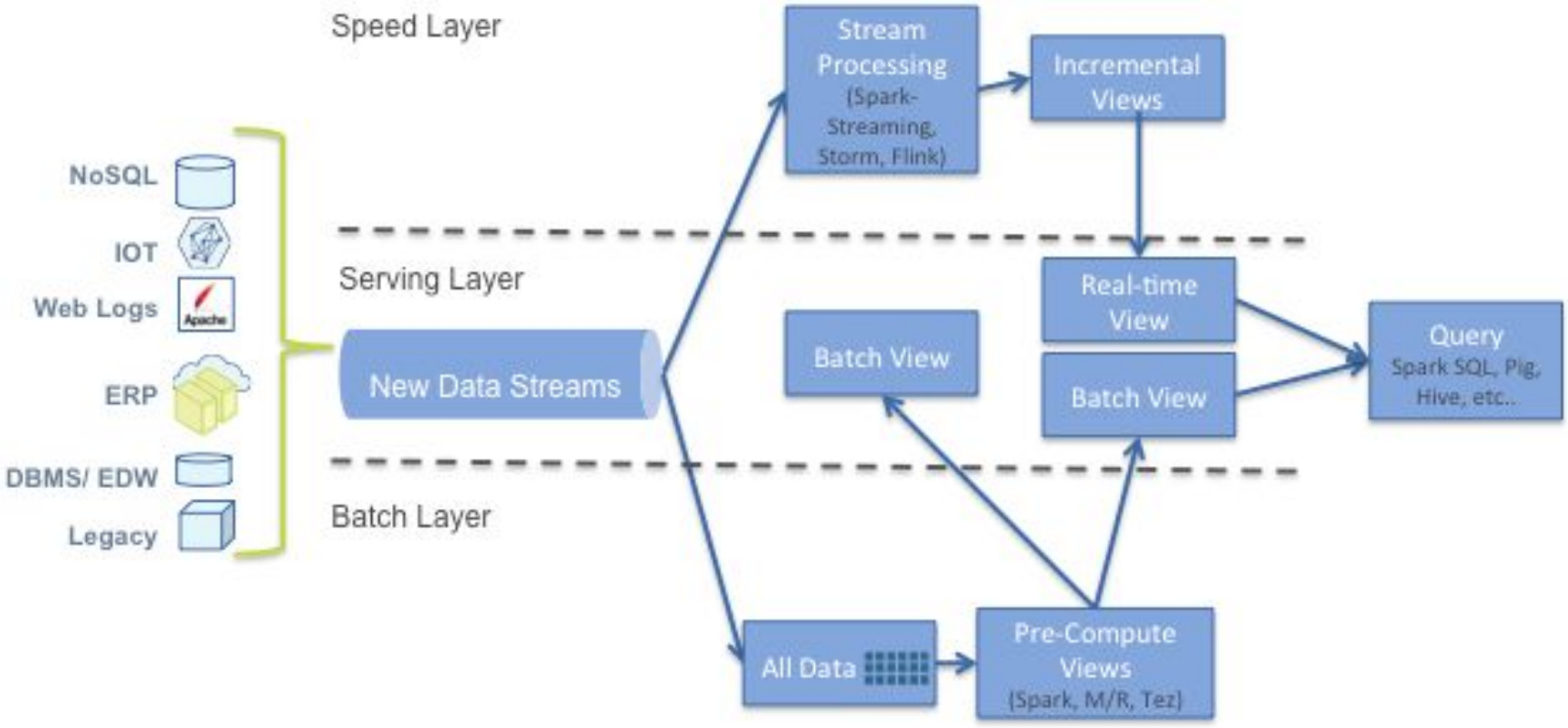
Проблема византийских генералов



Лямбда архитектура

Лямбда-архитектура

Архитектура обработки данных, предназначенная для обработки больших объемов данных с использованием как пакетных, так и потоковых методов обработки. Этот подход к архитектуре пытается сбалансировать задержку, пропускную способность и отказоустойчивость с помощью пакетной обработки для обеспечения всестороннего и точного представления пакетных данных, одновременно используя потоковую обработку в реальном времени для обеспечения представления онлайновых данных.


Speed Layer
Stream Processing (Spark-Streaming, Storm, Flink)
Incremental Views
NoSQL
IOT
Web Logs
Serving Layer
Real-time View
Query
Spark SQL, Pig, Hive, etc..
ERP
New Data Streams
Batch View
Batch View
DBMS/ EDW
Legacy
Batch Layer
All Data
Pre-Compute Views
(Spark, M/R, Tez)

Spark Streaming & DStream

Spark Streaming

Spark Streaming - это расширение основного API Spark, которое обеспечивает масштабируемую, высокопроизводительную и отказоустойчивую потоковую обработку потоков данных в реальном времени. Данные могут быть получены из многих источников, таких как Kafka, Kinesis или TCP-сокеты, и могут быть обработаны с помощью сложных алгоритмов, выраженных с помощью высокоуровневых функций, таких как map, reduce, join и window. Наконец, обработанные данные могут быть перенесены в файловые системы, базы данных и живые информационные панели. Фактически, вы можете применить алгоритмы машинного обучения Spark и обработки графов к потокам данных.

Spark Streaming

Spark Streaming

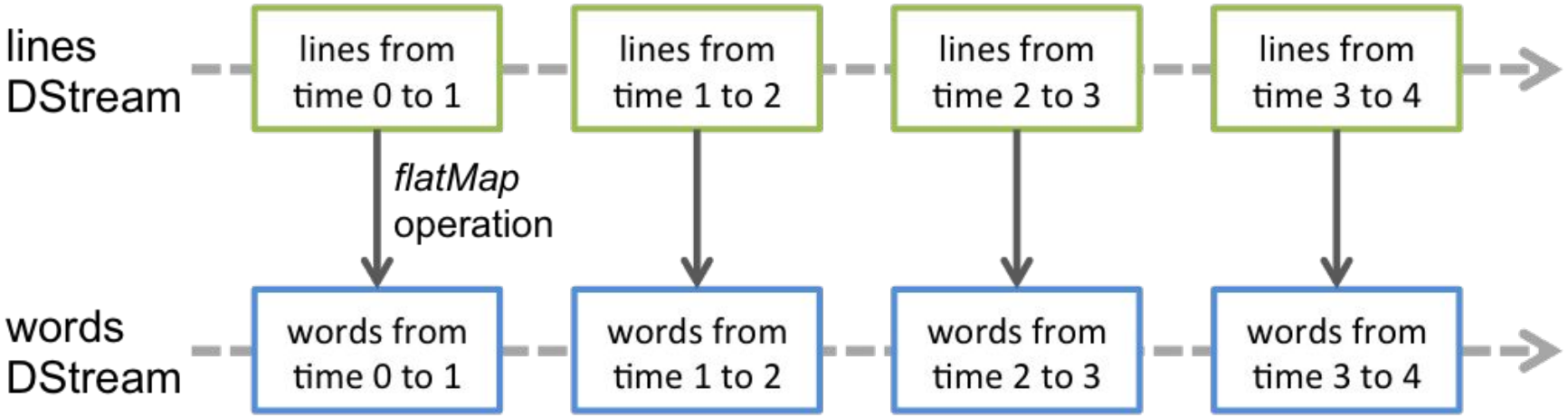
DStream

Дискретизированный поток или **DStream**-это базовая абстракция, предоставляемая Spark Streaming. Он представляет собой непрерывный поток данных, либо входной поток данных, полученный из источника, либо обработанный поток данных, созданный путем преобразования входного потока. Внутренне DStream представлен непрерывной серией RDDS, которая является абстракцией Spark неизменяемого распределенного набора данных (подробнее см. Руководство по программированию Spark). Каждый RDD в потоке содержит данные с определенного интервала, как показано на следующем рисунке.

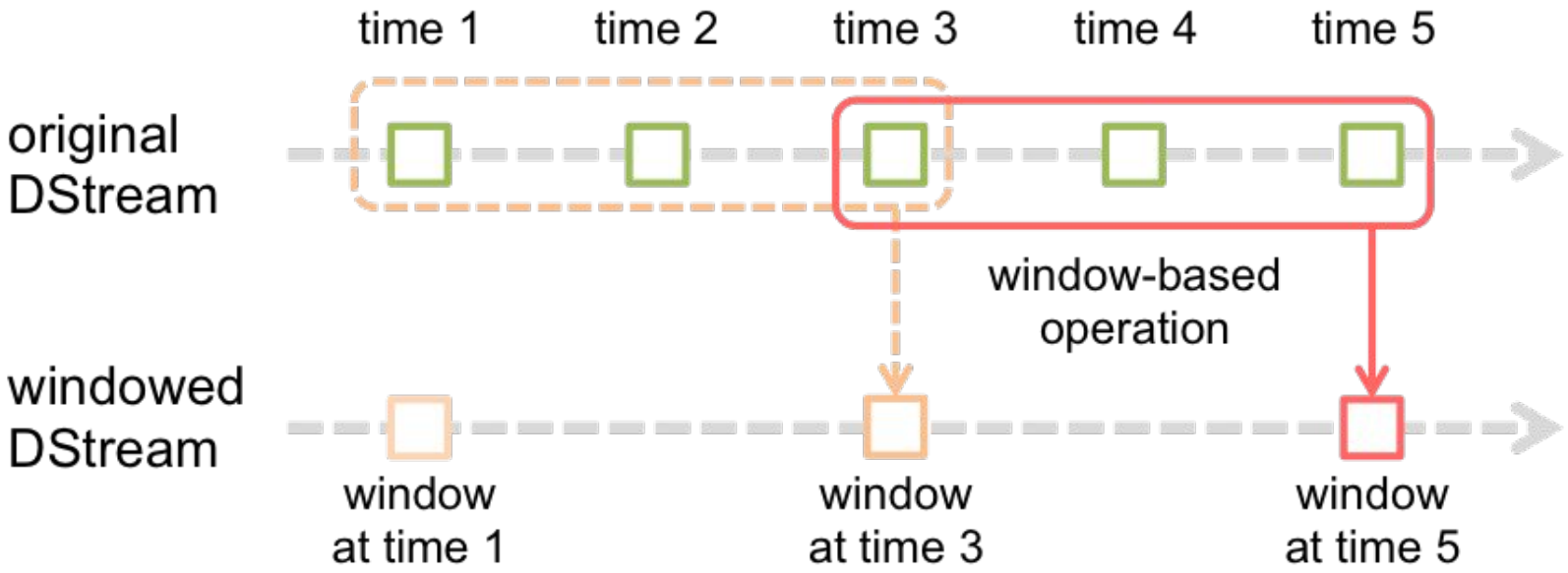
DStream


RDD @ time 1
RDD @ time 2
RDD @ time 3
RDD @ time 4
DStream
data from time 0 to 1
data from time 1 to 2
data from time 2 to 3
data from time 3 to 4

DStream


lines
DStream
lines from
time 0 to 1
lines from
time 1 to 2
lines from
time 2 to 3
lines from
time 3 to 4
flatMap
operation
words
DStream
words from
time 0 to 1
words from
time 1 to 2
words from
time 2 to 3
words from
time 3 to 4

DStream


time 1
time 2
time 3
time 4
time 5
original DStream
window-based operation
windowed DStream
window at time 1
window at time 3
window at time 5

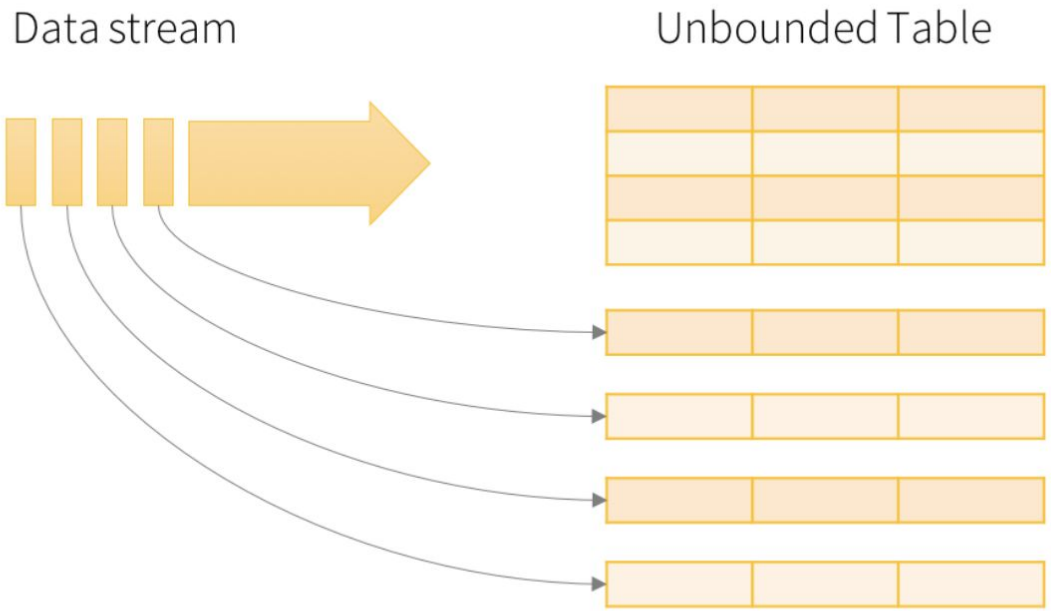
ML & DStream

Вы также можете легко использовать алгоритмы машинного обучения, предоставляемые **MLlib.** Прежде всего, существуют алгоритмы потокового машинного обучения (например, потоковая линейная регрессия, Потоковые **KMeans** и т. Д.), которые могут одновременно учиться на потоковых данных, а также применять модель к потоковым данным. Помимо этого, для гораздо более широкого класса алгоритмов машинного обучения вы можете изучить модель обучения в автономном режиме (то есть с использованием исторических данных), а затем применить модель онлайн к потоковым данным.

Structured streaming & Dataframe

Structured streaming

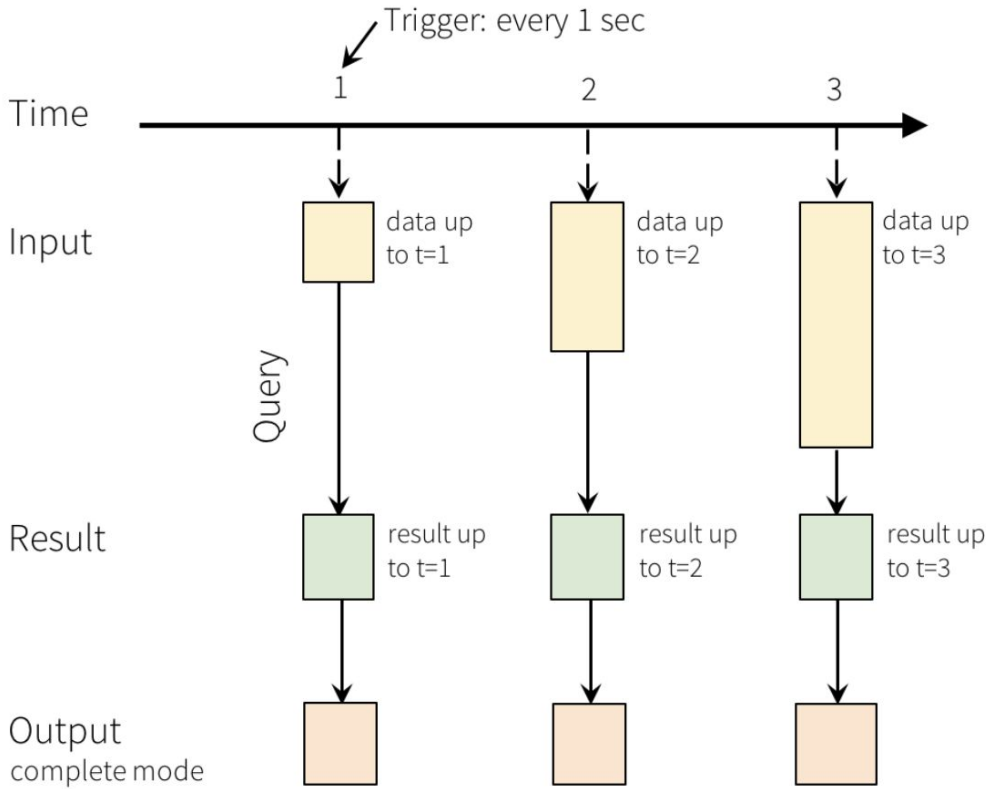
Ключевая идея структурированной потоковой передачи состоит в том, чтобы рассматривать поток живых данных как таблицу, которая постоянно добавляется. Это приводит к новой модели потоковой обработки, которая очень похожа на модель пакетной обработки. Вы будете выражать свои потоковые вычисления в виде стандартного пакетного запроса, как в статической таблице, а Spark запускает его как инкрементный запрос в неограниченной входной таблице.



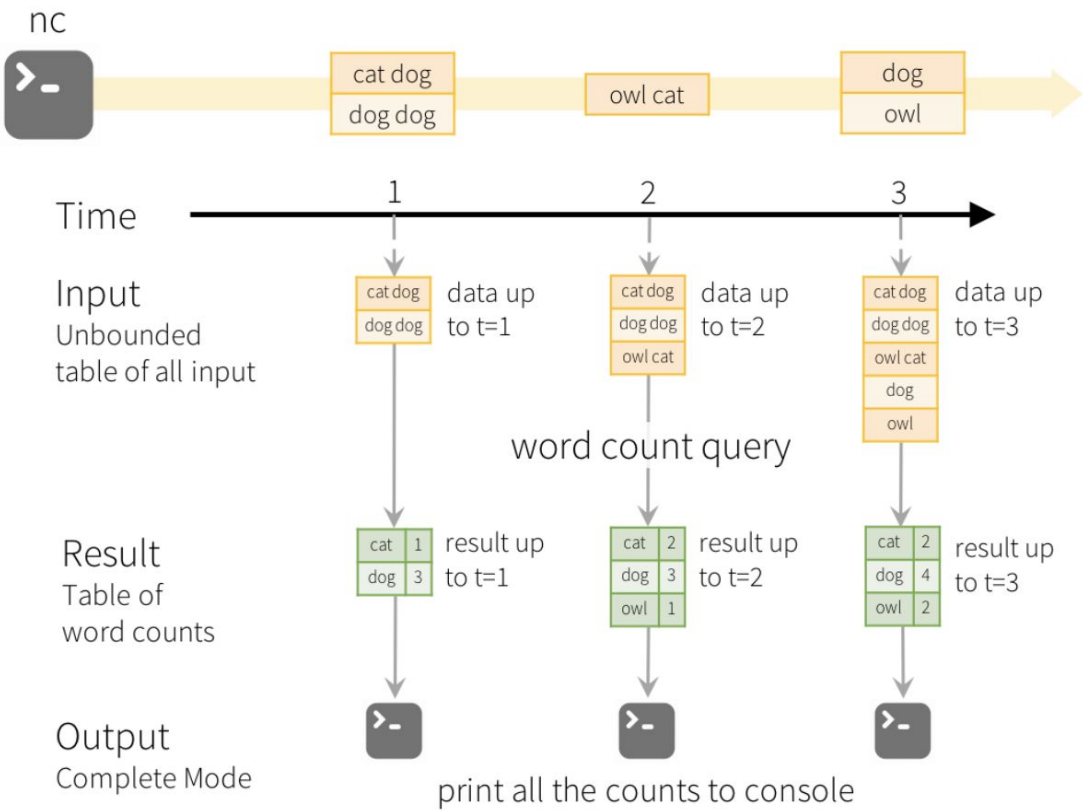
new data in the
data stream
=
new rows appended
to a unbounded table

Stream as a Table

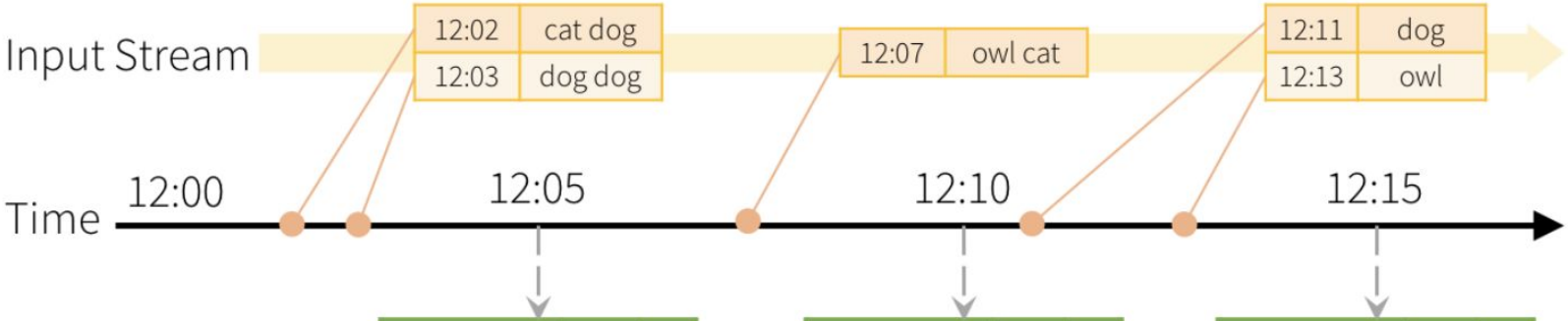
Data stream as an unbounded table



Programming Model for Structured Streaming



Model of the Quick Example



12:00 - 12:10	cat	1
12:00 - 12:10	dog	3

12:00 - 12:10	cat	2
12:00 - 12:10	dog	3
12:00 - 12:10	owl	1
12:05 - 12:15	cat	1
12:05 - 12:15	owl	1

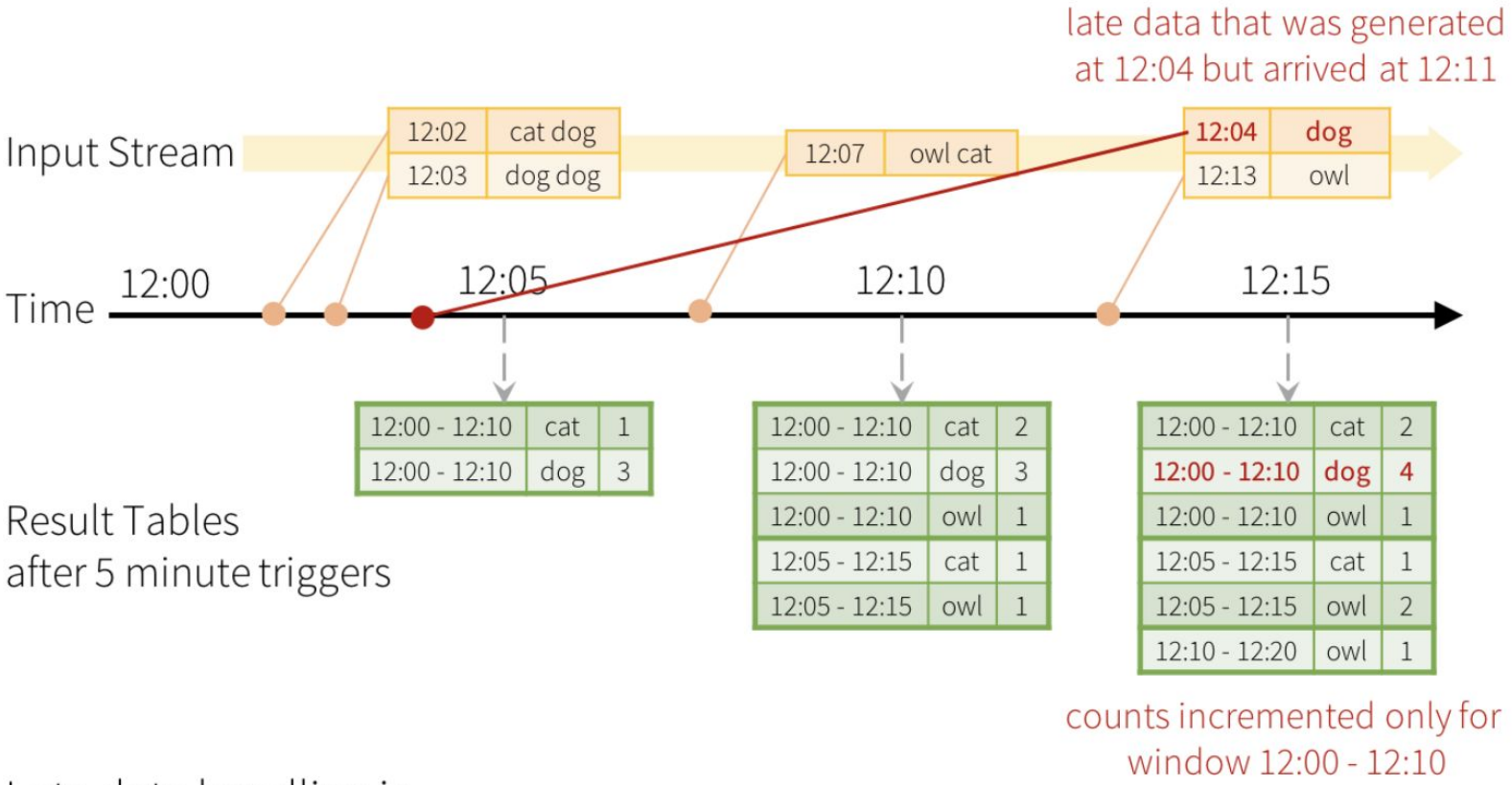
12:00 - 12:10	cat	2
12:00 - 12:10	dog	3
12:00 - 12:10	owl	1
12:05 - 12:15	cat	1
12:05 - 12:15	owl	2
12:05 - 12:15	dog	1
12:10 - 12:20	dog	1
12:10 - 12:20	owl	1

Result Tables
after 5 minute triggers

counts incremented for windows
12:00 - 12:10 and 12:05 - 12:15

counts incremented for windows
12:05 - 12:15 and 12:10 - 12:20

Windowed Grouped Aggregation
with 10 min windows, sliding every 5 mins



Late data handling in Windowed Grouped Aggregation