

DATA STREAMING Y SERVICIOS EN LA NUBE

Procesamiento de Datos

Magister - Efraín Alberto Oviedo
alberto.oviedo@udea.edu.co

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA

ESPECIALIZACIÓN EN ANALÍTICA Y CIENCIA DE DATOS



**UNIVERSIDAD
DE ANTIOQUIA**

1 8 0 3

AGENDA

1. Serialización de Datos

- Protocol Buffers
- Apache Thrift

2. Procesamiento en Batch

3. Procesamiento en Streaming

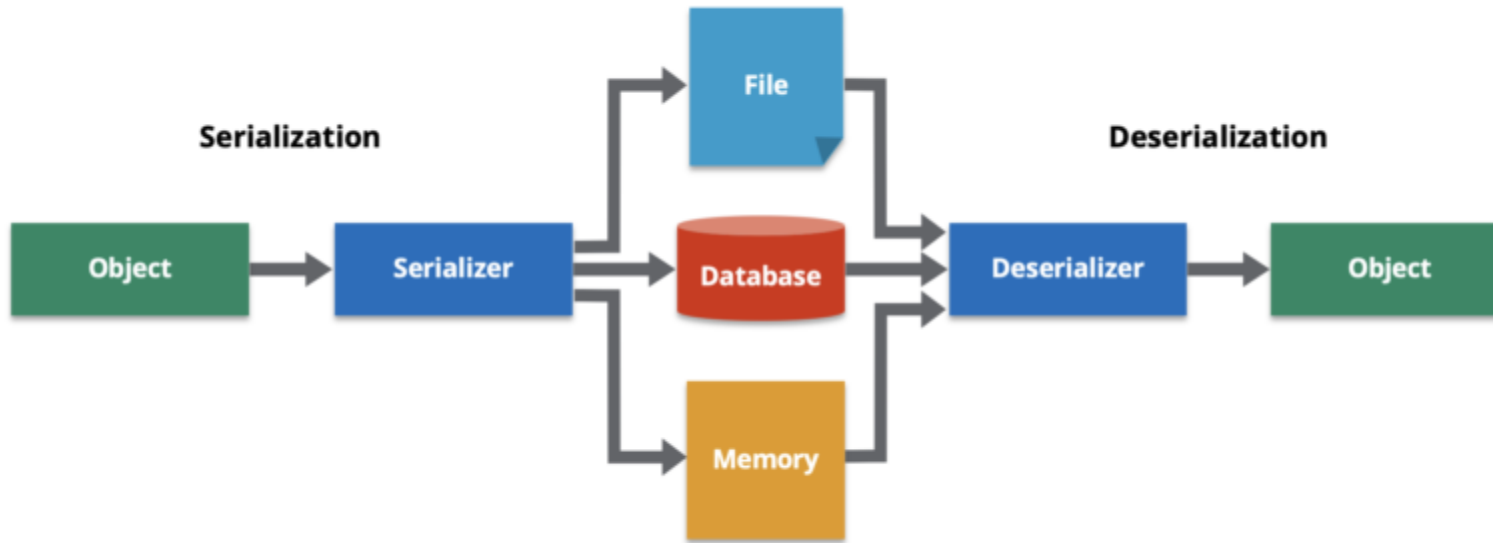
4. Procesamiento en micro-batch

Serialización de Datos

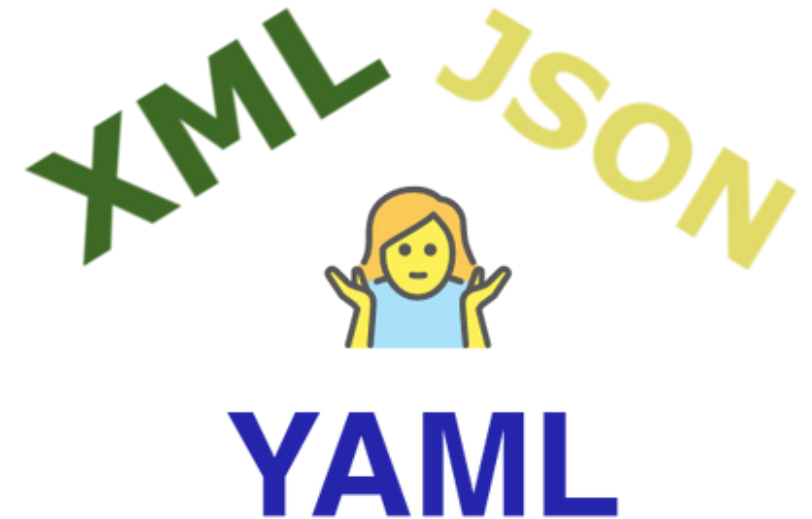
- Proceso de codificación de un objeto en un flujo de bytes con el fin de almacenarlo o transmitirlo. El objeto recibido/leído, es considerado como idéntico al original
- También suele ser utilizado como método de persistencia de objetos en forma de archivos, en memoria o en base de datos
- Otra de sus aplicaciones, permite detectar cambios en las variables en función del tiempo

Serialización de Datos

Proceso de Serialización



Lenguajes de Serialización



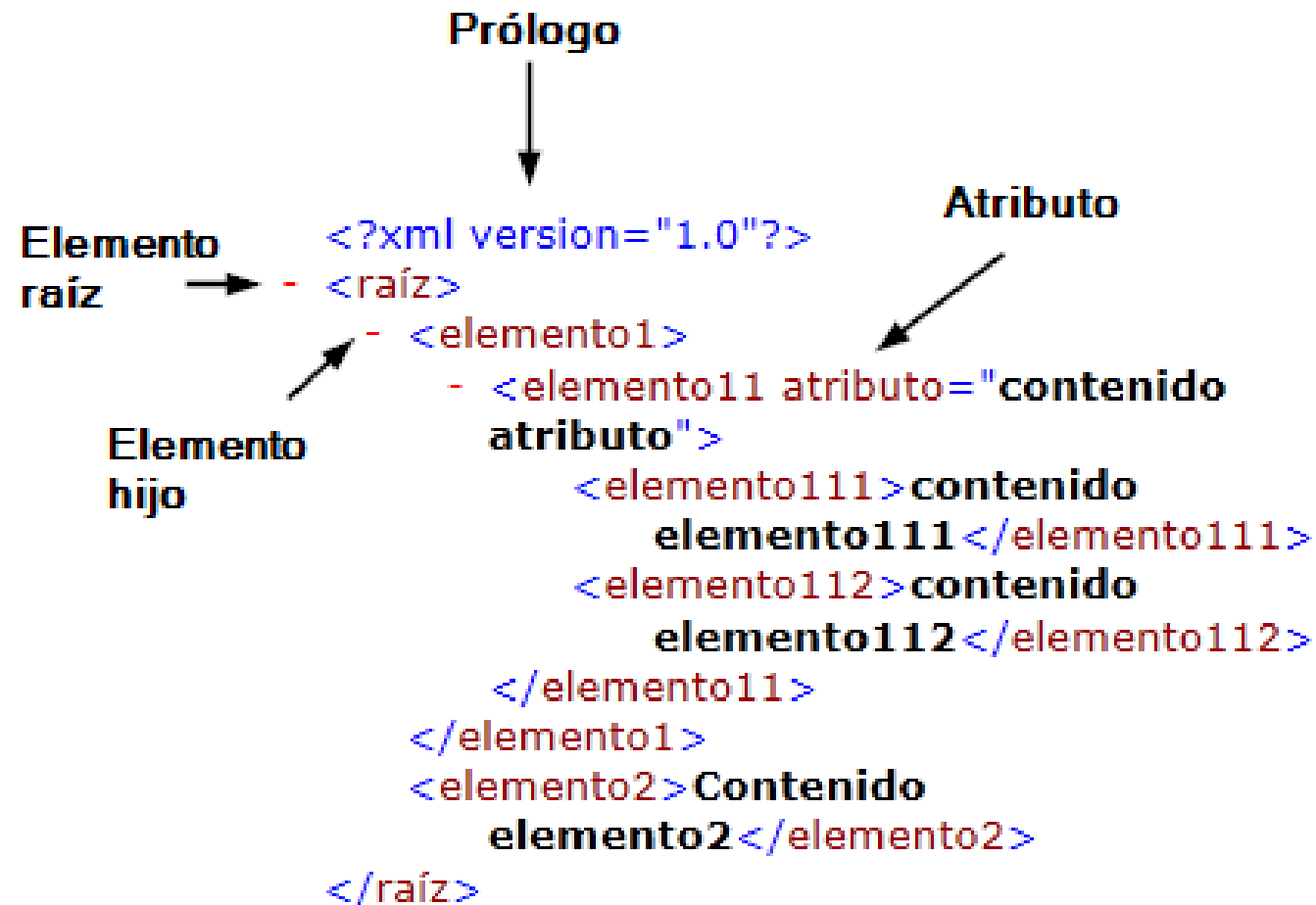
XML

XML (Lenguaje extensible de marcas)

- Estándar internacionalmente conocido
- Define etiquetas personalizadas para describir y organizar datos
- Se utiliza para transferir información de productos, transacciones, inventario, etc.



XML



```
<?xml version="1.0"
<quiz>
  <question>
    Who was the forty-second
    president of the U.S.A.?
  </question>
  <answer>
    William Jefferson Clinton
  </answer>
  <!-- Note: We need to add
    more questions later.-->
</quiz>
```

XML

XML

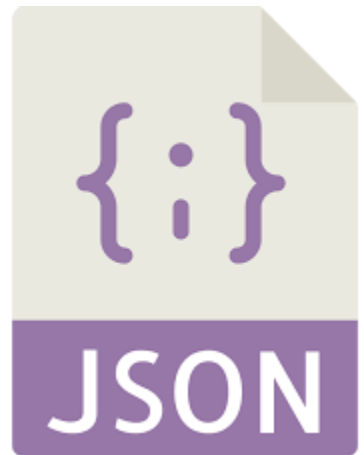
| ID | Nombre | Edad | Sexo | Profesión | Salario |
|----|--------|------|------|------------|-----------|
| 1 | Juan | 33 | M | Ingeniero | 4.500.000 |
| 2 | Ana | 38 | F | Arquitecta | 6.200.000 |

```
<?xml version="1.0" encoding="UTF-8" ?>
<empleados>
  <Id>1</Id>
  <Nombre>Juan</Nombre>
  <Edad>33</Edad>
  <Sexo>M</Sexo>
  <Profesión>Ingeniero</Profesión>
  <Salario>4500000</Salario>
</empleados>
<empleados>
  <Id>2</Id>
  <Nombre>Ana</Nombre>
  <Edad>38</Edad>
  <Sexo>F</Sexo>
  <Profesión>Arquitecta</Profesión>
  <Salario>6200000</Salario>
</empleados>
```

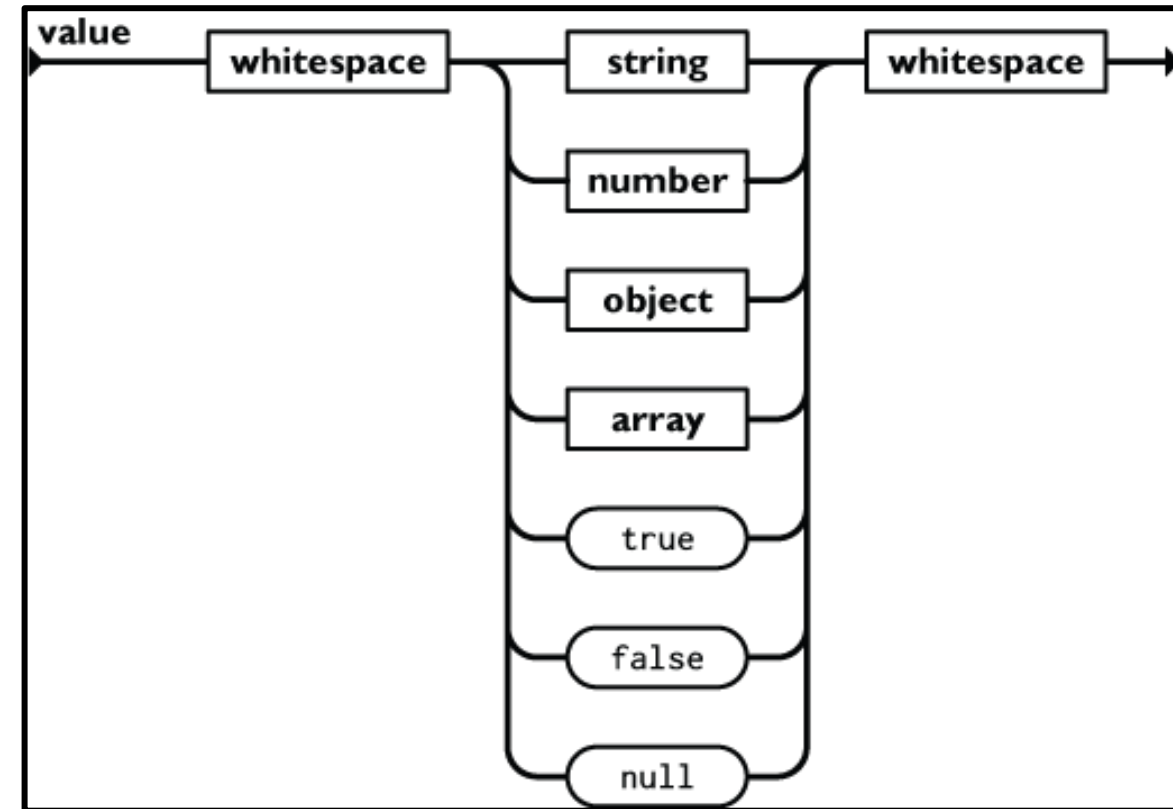
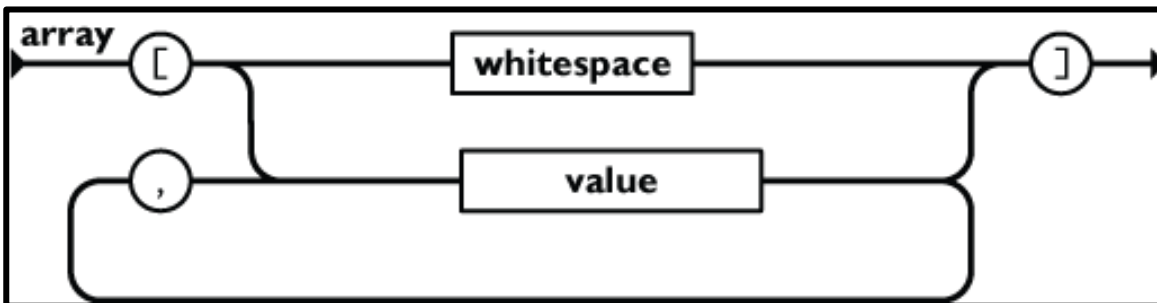
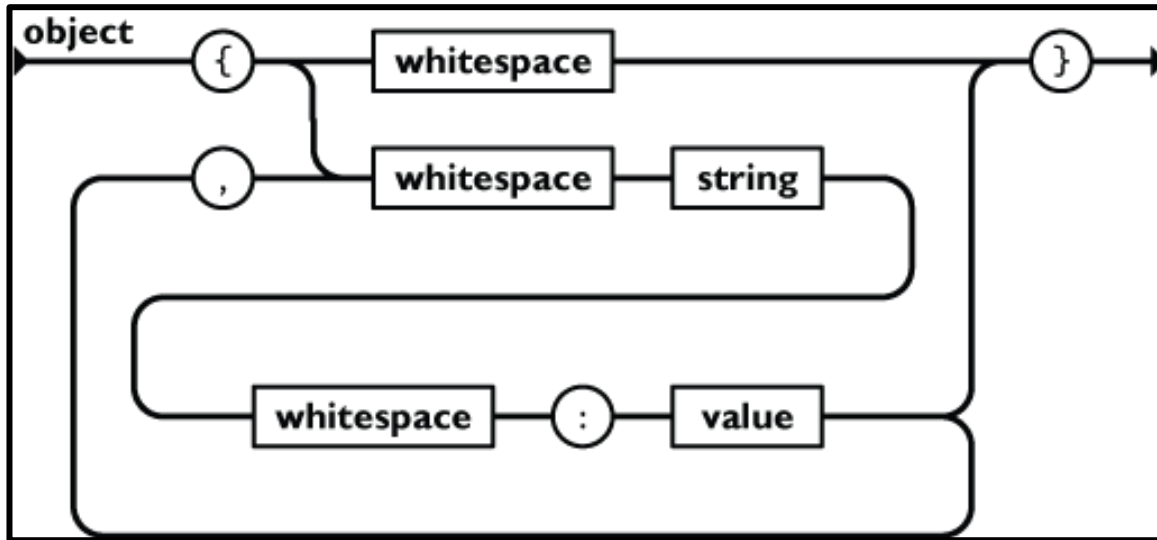
JSON

JSON (JavaScript Object Notation)

- Representa datos estructurados en la sintaxis de JavaScript
- Permite transmitir datos en aplicaciones Web
- Es un archivo de texto con extensión .json



JSON



JSON

| ID | Nombre | Edad | Sexo | Profesión | Salario |
|----|--------|------|------|------------|-----------|
| 1 | Juan | 33 | M | Ingeniero | 4.500.000 |
| 2 | Ana | 38 | F | Arquitecta | 6.200.000 |

```
{
  "empleados": [
    {
      "Id": 1,
      "Nombre": "Juan",
      "Edad": 33,
      "Sexo": "M",
      "Profesión": "Ingeniero",
      "Salario": 4500000
    },
    {
      "Id": 2,
      "Nombre": "Ana",
      "Edad": 38,
      "Sexo": "F",
      "Profesión": "Arquitecta",
      "Salario": 6200000
    }
  ]
}
```

YAML

YAML (YAML Ain't Markup Language)

- Lenguaje de serialización de datos utilizado frecuentemente en el diseño de archivos de configuración.
- Fue diseñado para ser útil y amigable para las personas que trabajan con datos
- La sangría se usa para la estructura, los dos puntos separan los pares clave valor y los guiones crean listas con viñetas
- Los archivos generados tienen extensión .yaml o .yml



YAML: YAML Ain't Markup Language™

What It Is:

YAML is a human-friendly data serialization language for all programming languages.

YAML Resources:

YAML Specifications:

- YAML 1.2:
 - Revision 1.2.2 # Oct 1, 2021 *New*
 - Revision 1.2.1 # Oct 1, 2009
 - Revision 1.2.0 # Jul 21, 2009
- YAML 1.1
- YAML 1.0

YAML Matrix Chat: '#chat:yaml.io' # Our New Group Chat Room!

YAML IRC Channel: libera.chat#yaml # The old chat

YAML News: twitter.com/yamlnews

YAML Mailing List: yaml-core # Obsolete, but historical

YAML on GitHub: # github.com/yaml/

YAML Specs: yaml-spec/

YAML 1.2 Grammar: yaml-grammar/

YAML Test Suite: yaml-test-suite/

YAML Issues: issues/

YAML Reference Parsers:

- Generated Reference Parsers
- YPaste Interactive Parser

YAML

<https://yaml.org/>

YAML

| ID | Nombre | Edad | Sexo | Profesión | Salario |
|----|--------|------|------|------------|-----------|
| 1 | Juan | 33 | M | Ingeniero | 4.500.000 |
| 2 | Ana | 38 | F | Arquitecta | 6.200.000 |

empleados:

- Id: 1

Nombre: Juan

Edad: 33

Sexo: M

Profesion: Ingeniero

Salario: 4500000

- Id: 2

Nombre: Ana

Edad: 38

Sexo: F

Profesion: Arquitecta

Salario: 6200000

Ejercicio

Represente en formato XML, JSON y YAML la información relacionada con las ciudades que ha visitado (País, departamento, fecha de visita, duración, motivo, etc)

Valide el ejercicio con un editor online

- <https://jsoneditoronline.org/>
- <https://jsonformatter.org/>
- <https://jsonformatter.org/yaml-formatter>



AGENDA

1. Serialización de Datos

- **Protocol Buffers**

- Apache Thrift

2. Procesamiento en Batch

3. Procesamiento en Streaming

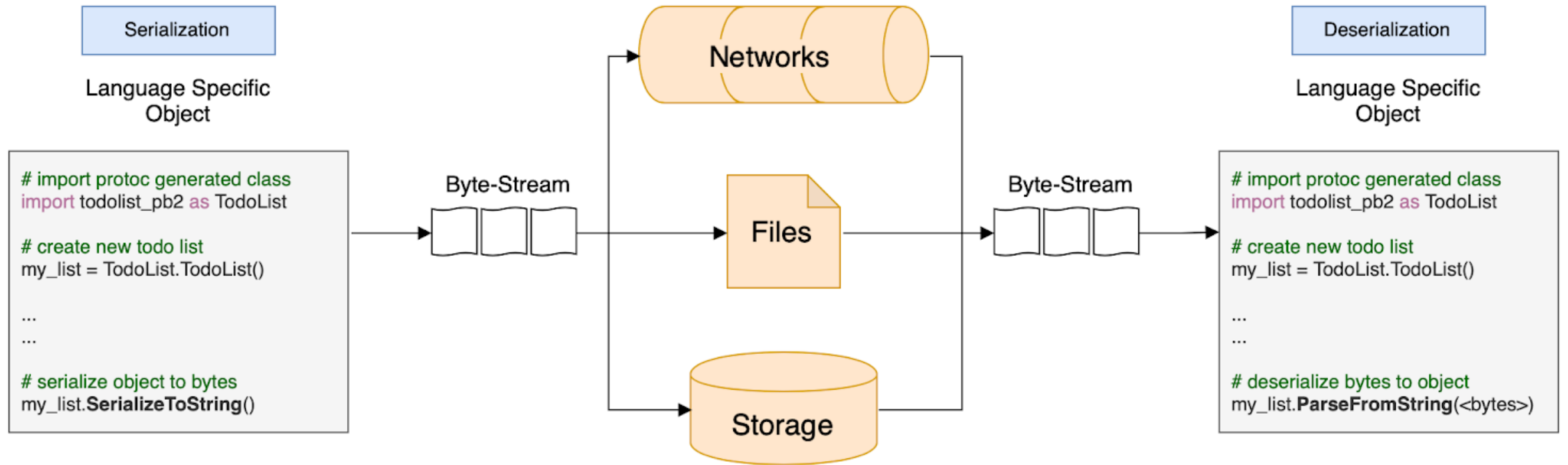
4. Procesamiento en micro-batch

Protocol Buffers

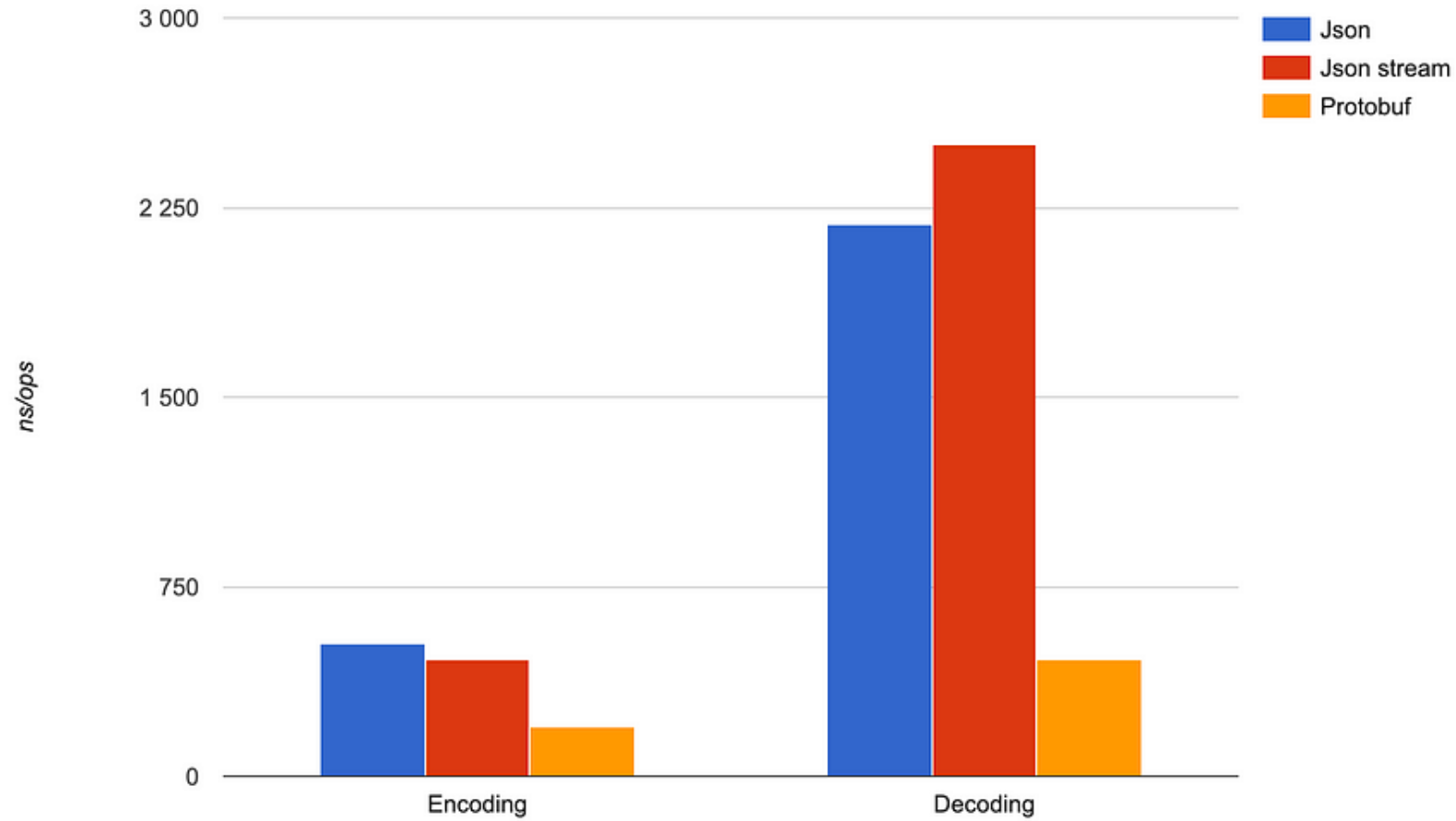


- Formato de serialización de datos que facilita el intercambio de datos entre aplicaciones, aún si están desarrolladas en distintos lenguajes.
- Desarrollado por Google y liberado en 2008 como proyecto de código abierto. Pensado para reemplazar el formato xml con el objetivo de ser más simple y rápido (entre 3 y 10 veces más pequeña, y entre 20 y 100 veces más rápida)
- La comunicación se realiza en formato cliente – servidor de forma local o remota
- Asigna etiquetas a los datos para utilizar un formato binario

Protobuf - Proceso



Protobuf - Rendimiento



<https://mnwa.medium.com/what-the-hell-is-protobuf-4aff084c5db4>

Protobuf – Empleados

empleado.proto

```
syntax = "proto3";  
message Empleado {  
  int32 id = 1;  
  string nombre = 2;  
  int32 edad = 3;  
  string sexo = 4;  
  string profesion = 5;  
  int32 salario = 6;  
}
```

```
emp1.id = 1  
emp1.nombre = 'Juan'  
emp1.edad = 33  
emp1.sexo = 'M'  
emp1.profesion = 'Ingeniero'  
emp1.salario = 4500000
```

```
b'\x08\x01\x12\x04Juan\x18!\x01M*\tIngeniero0\xa0\xd4\x92\x02'
```

```
syntax = "proto2";

package tutorial;

message Person {
    optional string name = 1;
    optional int32 id = 2;
    optional string email = 3;

    enum PhoneType {
        MOBILE = 0;
        HOME = 1;
        WORK = 2;
    }

    message PhoneNumber {
        optional string number = 1;
        optional PhoneType type = 2 [default = HOME];
    }

    repeated PhoneNumber phones = 4;
}

message AddressBook {
    repeated Person people = 1;
}
```

Protobuf

Tutorial

<https://protobuf.dev/getting-started/pythontutorial/>

Guía de programación

<https://protobuf.dev/programming-guides/proto3/>

AGENDA

1. Serialización de Datos

- Protocol Buffers
- **Apache Thrift**

2. Procesamiento en Batch

3. Procesamiento en Streaming

4. Procesamiento en micro-batch

Apache Thrift



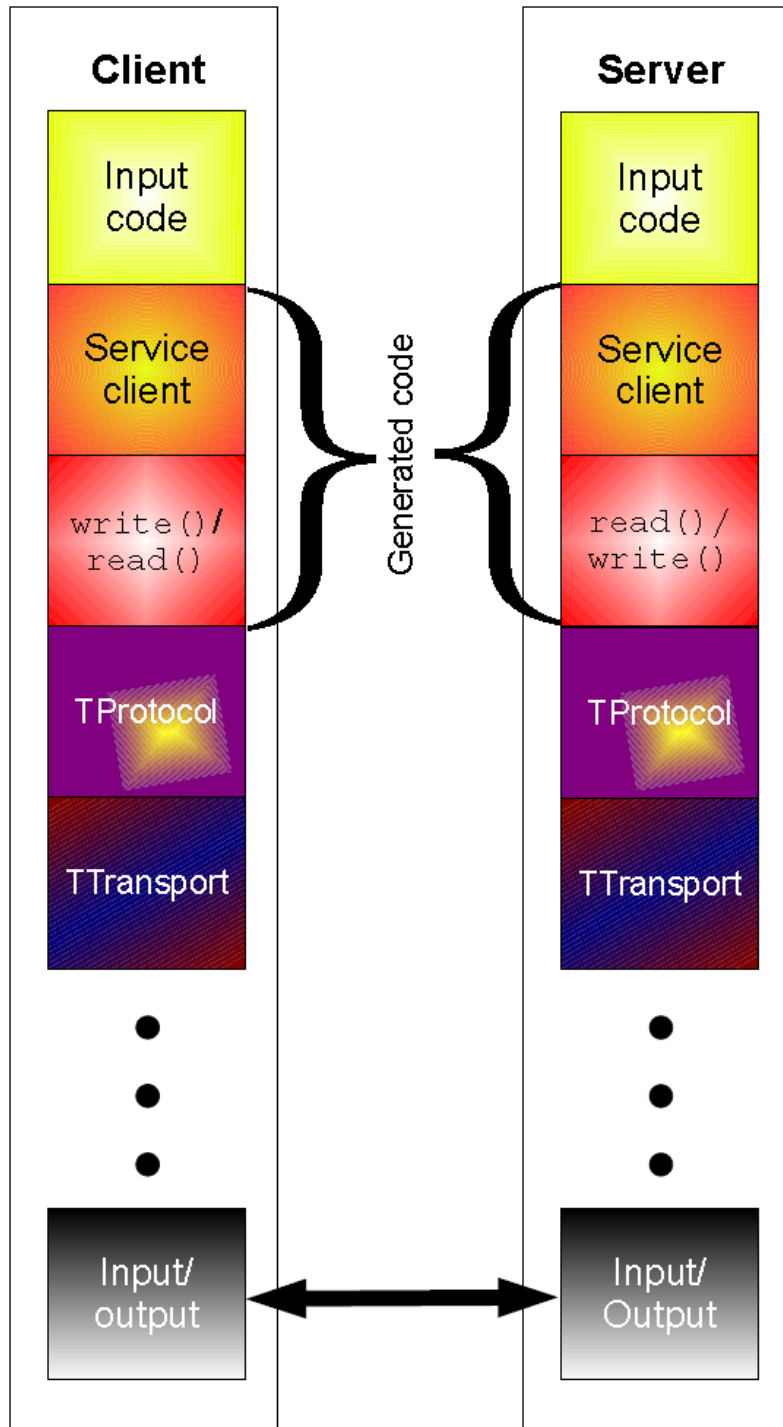
- Protocolo para la gestión de los macrodatos, que a demás de permitir **serialización** de objetos en formato binario, incluye el desarrollo de interfaces de **servicios** utilizando archivos de extensión .thrift
- Se enfoca principalmente en la capa de comunicación entre el cliente y el servidor RPC (Llamado a procedimientos remotos)
- Fue desarrollado por **Facebook** y posteriormente liberado, es de código abierto y se utiliza generalmente en C++ pero se puede adaptar a otros lenguajes

Apache Thrift

A partir del archivo .thrift se genera código tanto para el cliente como para el servidor que permite el intercambio de datos.

Protocolos (Binario, Json, etc)

Transporte (Socket, Memoria, etc)

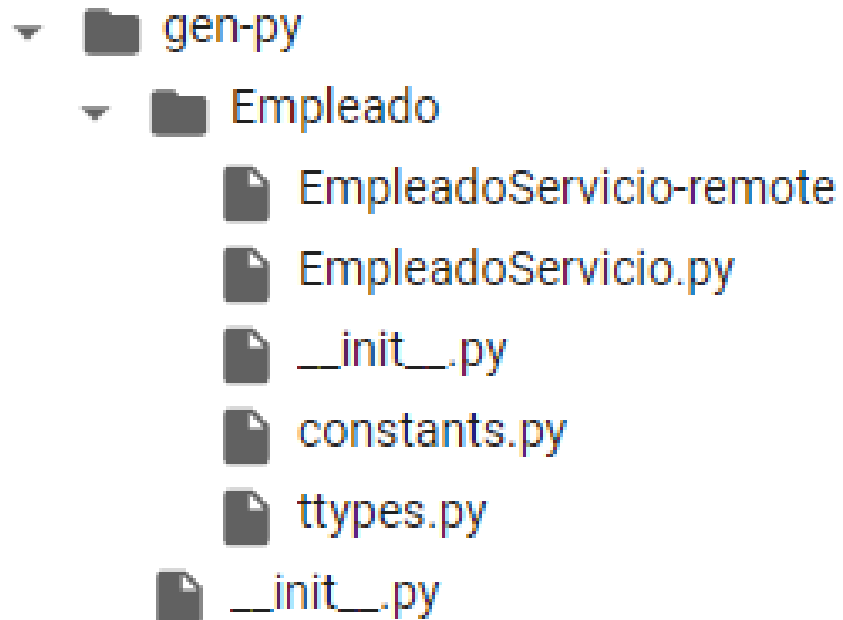


Apache Thrift

empleado.thrift

```
struct Empleado {  
  1: i32 id,  
  2: string nombre,  
  3: i32 edad,  
  4: string sexo,  
  5: string profesion,  
  6: i32 salario,  
}  
  
service EmpleadoServicio {  
  string msg(),  
  string send_Emp(1: Empleado  
new_emp),  
  bool mayor40(1: Empleado emp)  
}
```

!thrift -r --gen py local/data/empleado.thrift



AGENDA

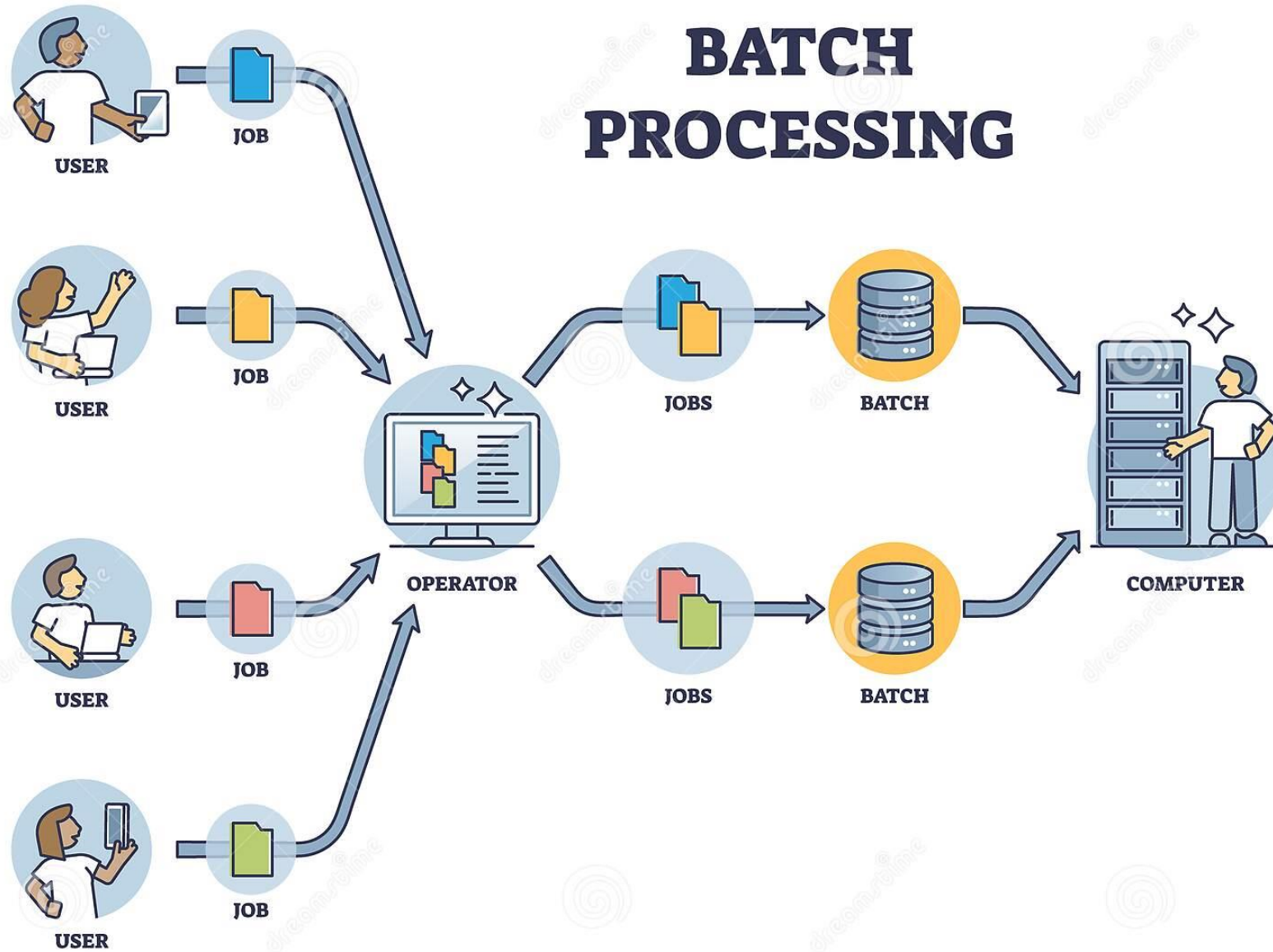
1. Serialización de Datos
 - Protocol Buffers
 - Apache Thrift
- 2. Procesamiento en Batch**
3. Procesamiento en Streaming
4. Procesamiento en micro-batch

Procesamiento en Batch

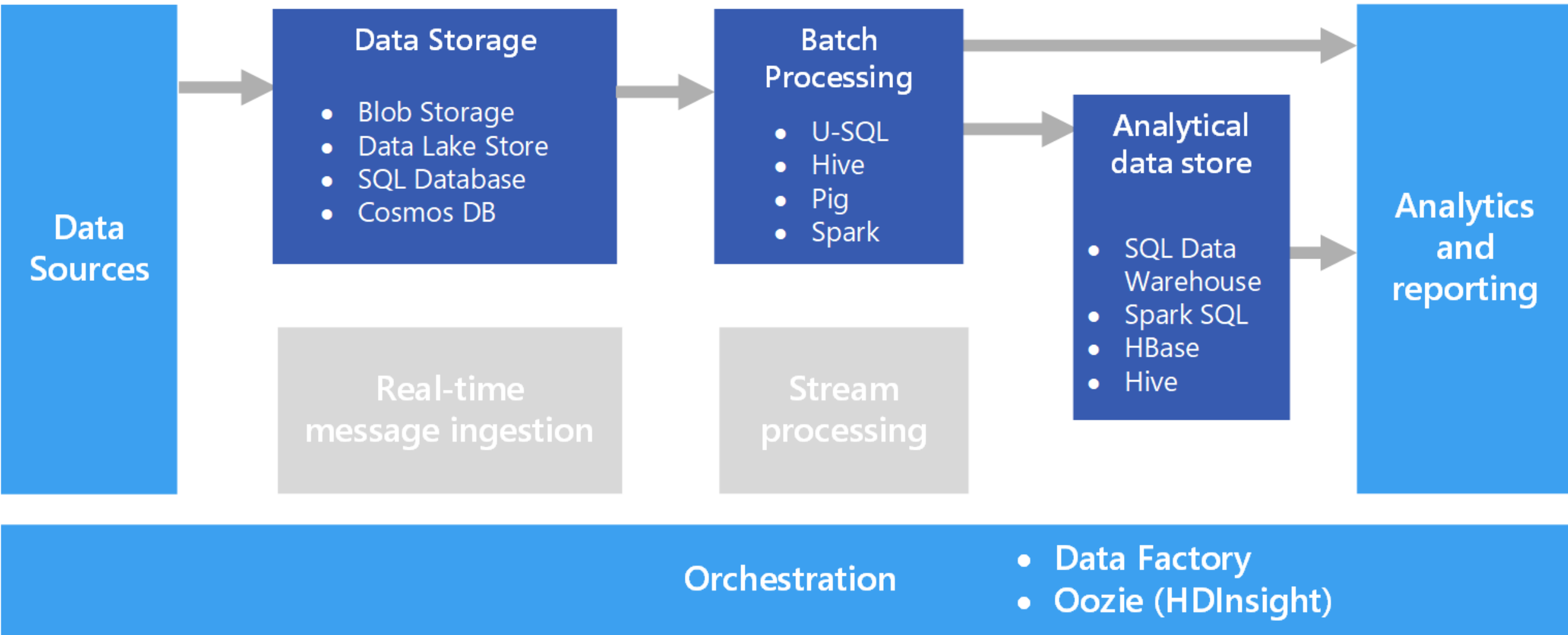
- El procesamiento en Batch o por lotes, contempla la **carga y proceso de grandes volúmenes de datos para realizar tareas repetitivas de manera automática.**
- Requieren una interacción humana mínima ejecutando de manera eficiente tareas repetitivas
- Se recomienda utilizarlos en proceso que tomen mucho tiempo en ser realizado sea por el gran volumen de los datos o por la complejidad misma del proceso



BATCH PROCESSING



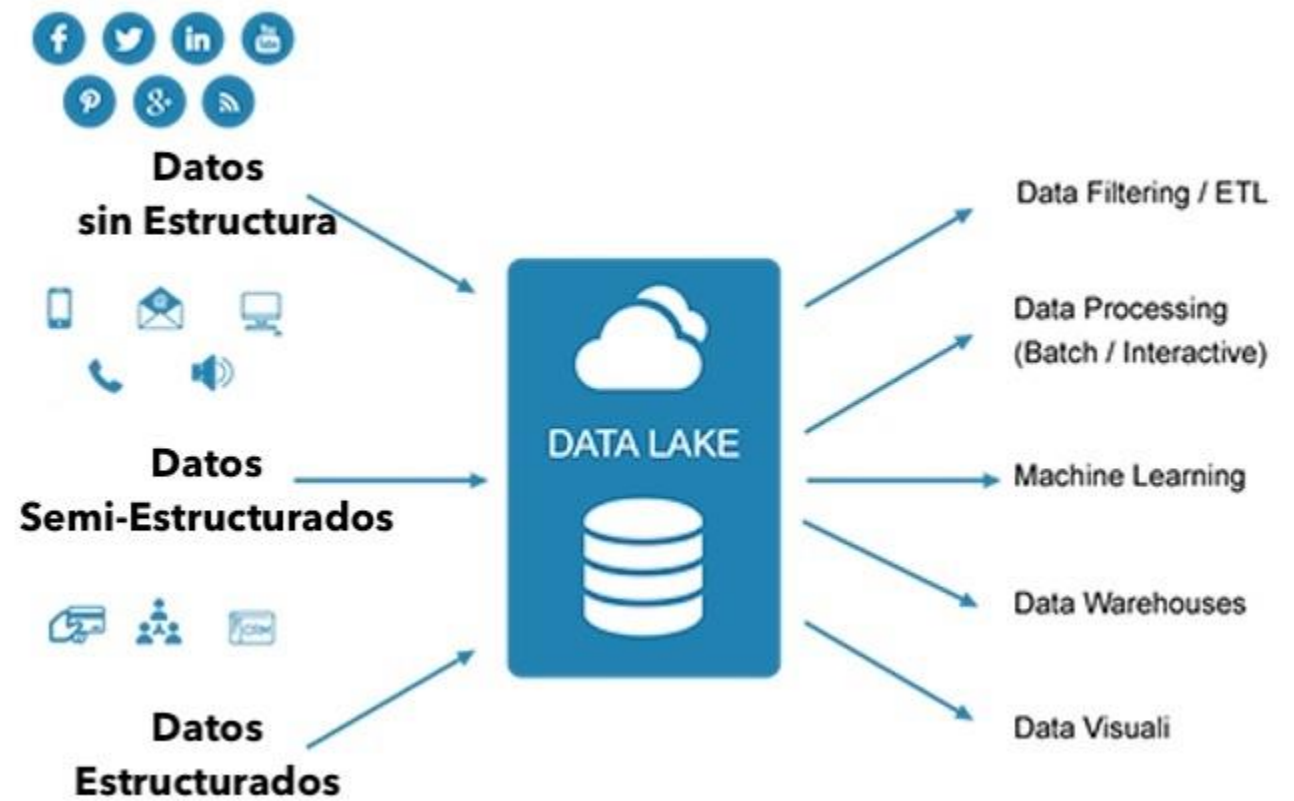
Procesamiento en Batch



Data Lake

Data Lake (Lago de datos) consiste en un repositorio centralizado donde es posible almacenar grandes cantidades de datos estructurados y no estructurados

Permite **almacenar los datos tal y como están**, y procesarlos **posteriormente bajo demanda**, es decir que no es necesario almacenarlos con una estructura de datos definida



Data Lake

| Características | Almacenamiento de datos | Lago de datos |
|----------------------|---|--|
| Datos | Relacionales provenientes de sistemas transaccionales, bases de datos operativas y aplicaciones de línea de negocio | No relacionales y relacionales provenientes de dispositivos de IoT, sitios web, aplicaciones móviles, redes sociales y aplicaciones corporativas |
| Esquema | Diseñado con anterioridad a la implementación del almacenamiento de datos (esquema en escritura) | Escrito al momento del análisis (esquema en lectura) |
| Precio/desempeño | Resultados de búsqueda más rápidos con almacenamiento de mayor costo | Resultados de consultas que se tornan más rápidos con almacenamiento de bajo costo |
| Calidad de los datos | Datos seleccionados cuidadosamente que funcionan como la versión central de la verdad | Cualquier dato seleccionado o no (es decir, datos sin procesar) |
| Usuarios | Analistas de negocios | Científicos de datos, desarrolladores de datos y analistas de negocios (con datos seleccionados) |
| Análisis | Generación de informes en lotes, inteligencia empresarial y visualizaciones | <i>Machine learning</i> , análisis predictivo, detección de datos y creación de perfiles |

<https://aws.amazon.com/es/big-data/datalakes-and-analytics/what-is-a-data-lake/>

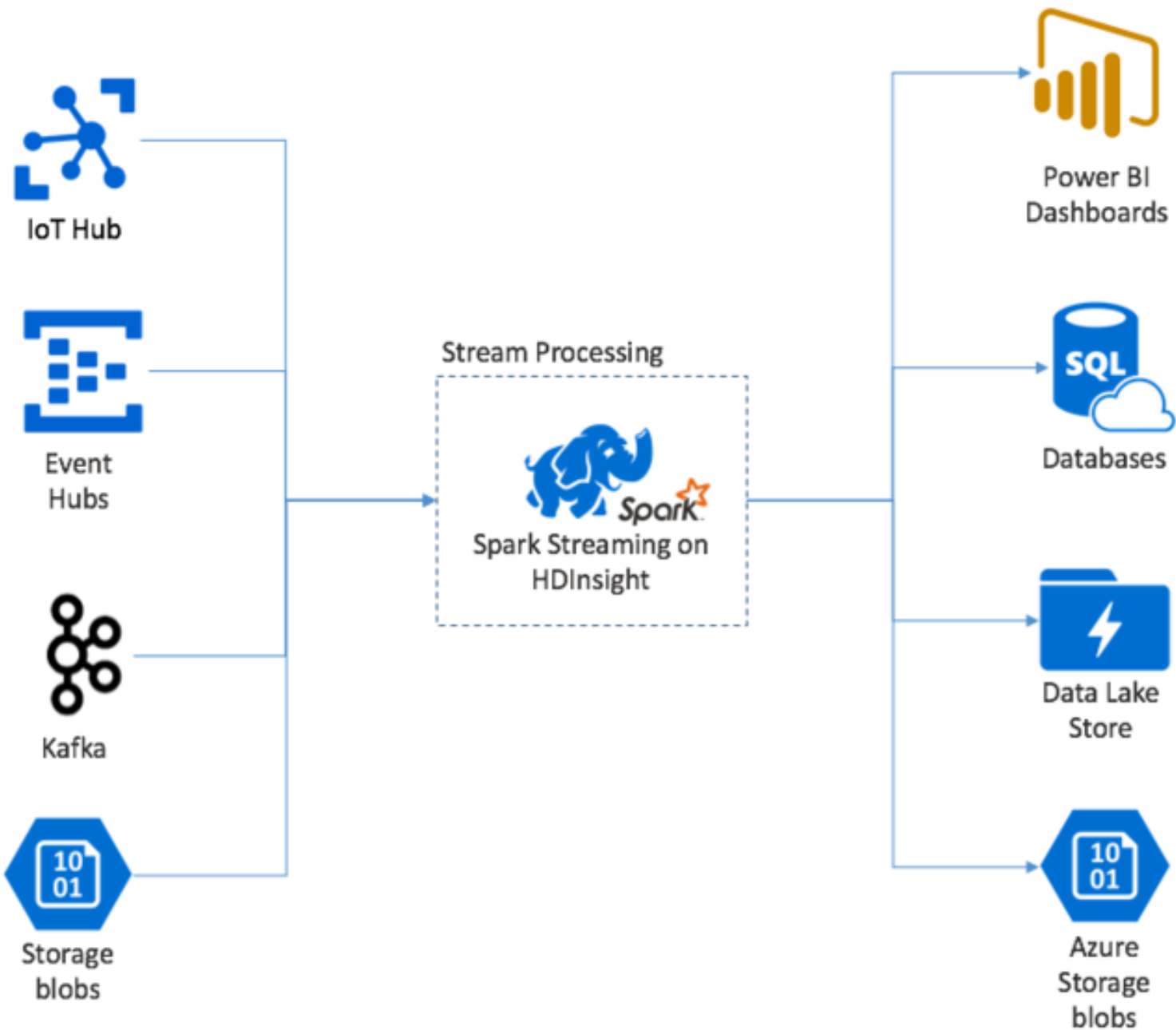
AGENDA

1. Serialización de Datos
 - Protocol Buffers
 - Apache Thrift
2. Procesamiento en Batch
- 3. Procesamiento en Streaming**
4. Procesamiento en micro-batch

Procesamiento en Streaming

- El procesamiento en streaming o flujo de datos permite **procesar los datos de forma continua en el mismo instante en que llegan** sin esperar ningún otro evento
- Cada lectura de datos se concibe como un flujo de datos que pueden representarse como un conjunto de lista de valores o como un conjunto de pares clave-valor





El procesamiento de datos en Streaming usualmente es realizado en **cluster de Spark** que garantizan que cada uno de los eventos sea procesado de manera correcta incluyendo el concepto de la tolerancia a fallas

Data Stream

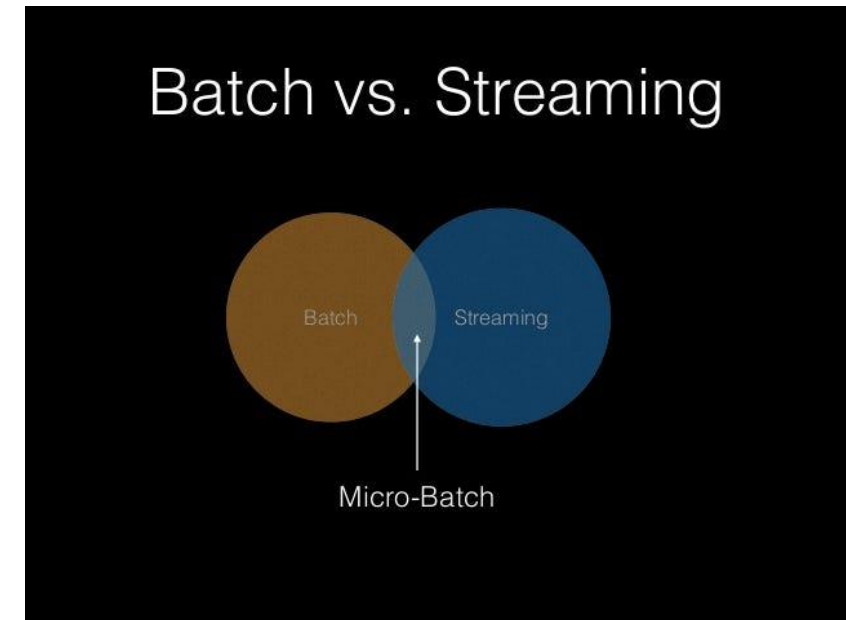
- Data Stream o flujo de datos se refiere a la transmisión de una secuencia de señales codificada digitalmente en paquetes para transmitir información
- Data Stream tiene la necesidad de **procesar y realizar análisis en tiempo real**
- En el Data Stream cuenta con una naturaleza continua de generación de datos a gran velocidad, que implica que los datos sean procesados sin necesidad de ser previamente almacenados

AGENDA

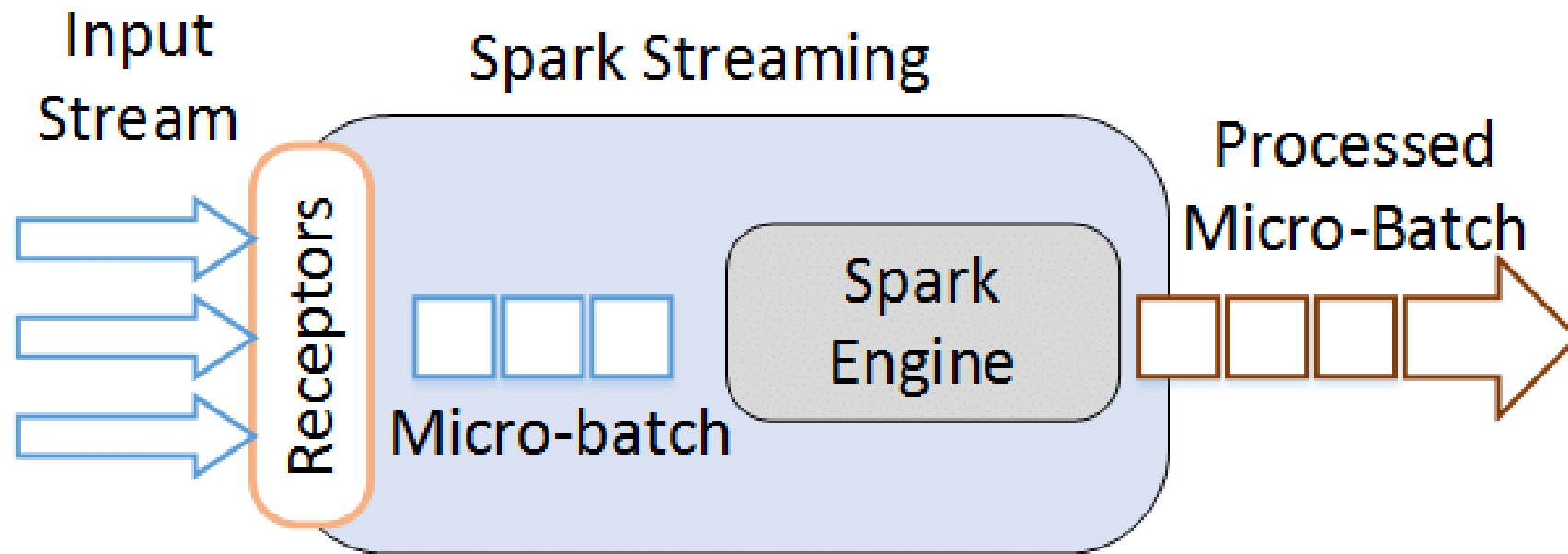
1. Serialización de Datos
 - Protocol Buffers
 - Apache Thrift
2. Procesamiento en Batch
3. Procesamiento en Streaming
- 4. Procesamiento en micro-batch**

Procesamiento en micro-batch

- El procesamiento en micro lotes consiste en agrupar los datos en pequeños grupos o lotes para luego ser procesados en conjunto.
- Es una variante del procesamiento por lotes, donde se procesan grupos pequeños de nuevos datos y se debe determinar una frecuencia de procesamiento de datos.
- A pesar de no ser un verdadero procesamiento en tiempo real, es suficiente para un gran número de aplicaciones

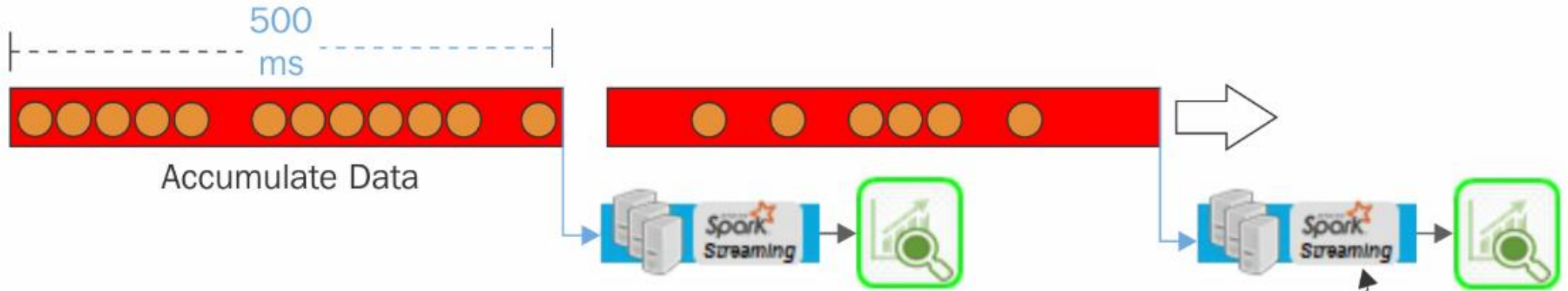


Procesamiento en micro-batch



micro-batch vs Native Stream

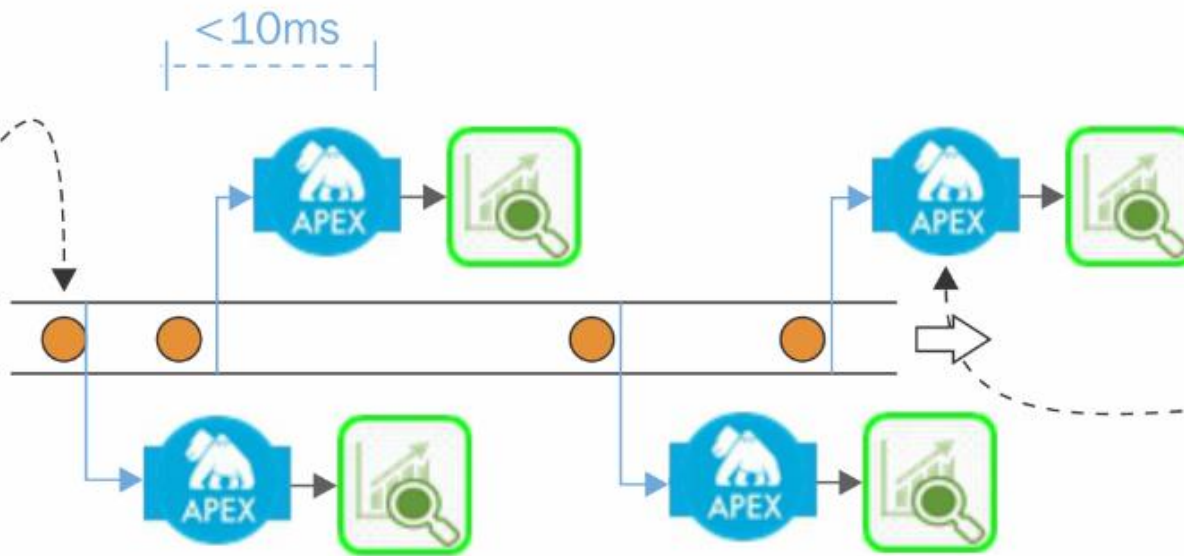
Micro-Batch Example



True Streaming Example



No accumulation
Processing begins
upon data arrival



At every Micro-batch
-Processing context is lost
-Content must be saved by application (difficult)
-Next task must be re-initialization

Context is automatically maintained by Apex platform,
No custom application code Required