ST0263 – TÓPICOS ESPECIALES EN TELEMÁTICA

2023-1
Introducción a BIG DATA

Universidad EAFIT

1.Datos Estructurados, No-Estructurados, Semi-Estructurados

	estructurado	semi-estructurado	no-estructurado
tecnologia y formatos	DBMS, DWH	XML, JSON, CSV, RDF	caracteres, binario
lenguajes	SQL	API's, algunos específicos	Lenguage Natural / QBE
modelos	modelos de datos pre-definidos schema-on-write	modelos definidos al momento de la lectura (schema-on-read) o aún sin esquemas.	no hay más allá de los formatos de archivos y texto, audio, video (coding & compresión)
datos	centrado en datos numéricos y categóricos, poco de texto como principal, audio o video.	centrado en texto categorizado, no numérico nativo, puede contener imágenes, video, etc.	centrado en texto crudo, imágenes, video, sonido
búsquedas	facil, lenguaje formal matemático y computacional (SQL)	algunos con índices	muy dificil en los datos crudos, debemos generar metadatos o modelos de aprendizaje.
Motores (almacenamiento, Consultas, Procesamiento)	SQL – DBMS	NoSQL	Sistemas de archivos (ej: hdfs), recuperación de información
Sintaxis en los datos	si (esquemas)	parcialmente (tags), lenguajes de marcación	No más allá del formato binario entendible por máquina
Semánt*ica en los datos*	si (tipos de datos)	no, scheme-on-read	no

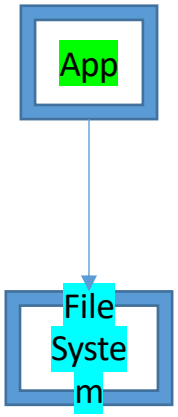
2. Almacenamiento y motores de bases de datos distribuidos

- 2. Almacenamiento y Motores
    1. Almacenamiento:
        1. Almacenamiento de datos no estructurados
        2. Sistemas de Archivos Distribuidos
        3. Bases de datos
        4. Data warehouse
        5. Datalakes
    2. SQL y bases de datos relaciones
    3. Bases de Datos NoSQL
    4. Big Data, Almacenamiento masivo, motores de procesamiento (Map – Reduce, Spark), ecosistemas
    5. Plataformas cloud para Big Data, SQL, NoSQL

¿cómo se almacenan los datos?

1. Sistemas de archivos

- Datos almacenados en el Disco, gestionados por el Sistema Operativo
- File Systems – sistemas de archivos
- Varios formatos:
  - FAT
  - NTFS
  - Ext3
  - XFS
  - AFS



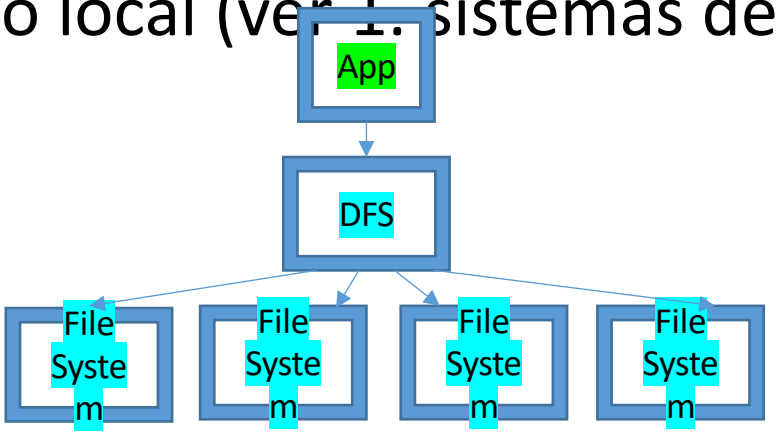
- Los programas (app) acceden a los datos mediante 'calls' (open,close, read, write, lock/unlock) al Sistema de Archivos (File System)
- Es usado???? Si, por los científicos de datos en sus desktops, workstations o laptops. Que problema hay acá? No escala.

2. Sistemas de archivos distribuidos

- Varios nodos se interconectan para formar un ‘solo sistema de archivo’ virtual.
- Es diferente a copiado de archivos (cp, scp, ftp, http, etc)
- Un programa accede a los datos locales o remotos, con la misma semántica de un acceso local (ver 1. sistemas de archivos).
- Sistemas predominantes:
  - NFS, AFS, iSCSI, WebDAV, SMB
  - GlusterFS
  - GFS
  - HDFS

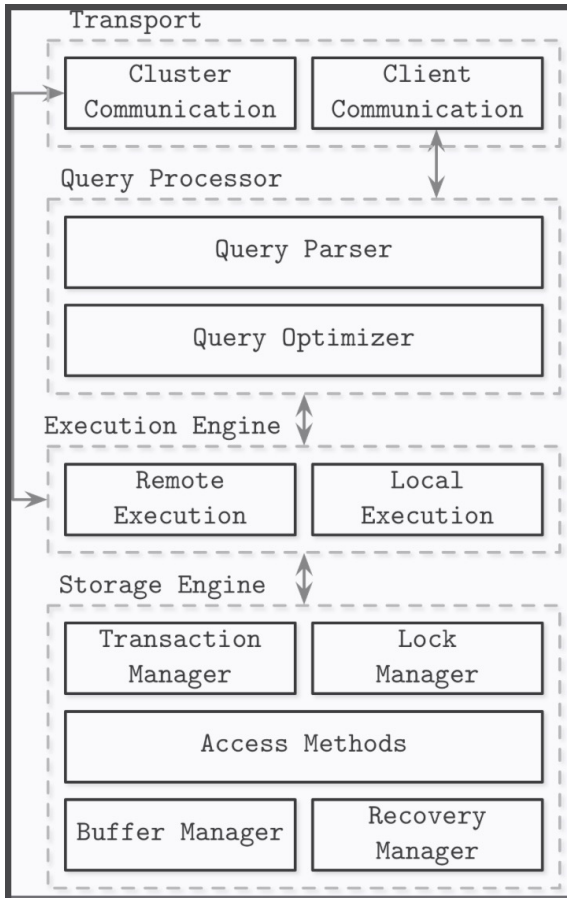
2. Sistemas de archivos distribuidos

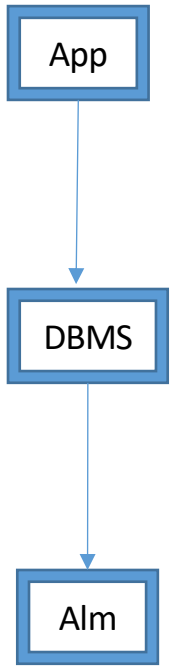
- Varios nodos se interconectan para formar un ‘solo sistema de archivo’ virtual.
- Es diferente a copiado de archivos (cp, scp, ftp, http, etc)
- Un programa accede a los datos locales o remotos, con la misma semántica de un acceso local (ver 1. sistemas de archivos).
- Sistemas predominantes:
  - NFS, AFS, iSCSI, WebDAV, SMB
  - GlusterFS
  - GFS
  - HDFS



3. Motores de almacenamiento

- Es una capa de software entre las aplicaciones y el almacenamiento, que mediante un modelo de datos y un mecanismo de modelado, permite a las aplicaciones GESTIONAR los datos a nivel de REGISTRO, documento, columna, gráfo, serie tiempo, etc.
  - Esa GESTIÓN incluye:
    - CRUD (create, read, update and delete) NO UTILIZADO EN ANALÍTICA ni Data Science, ni Ing de datos para analítica
      - Ojo: hay aplicaciones hibridas que pueden requerirlo, cuales?
    - Consulta y Transformación por Bloques, lotes, datasets, etc, MUY MUY utilizado en analítica.
- Generalmente se conoce como MOTOR DE BASE DE DATOS o DBMS (Data Base Management System), pero pueden haber otros sistemas como Repositorios de datasets (ej: CKAN)





motor

4. Motores de bases de datos (DBMS)

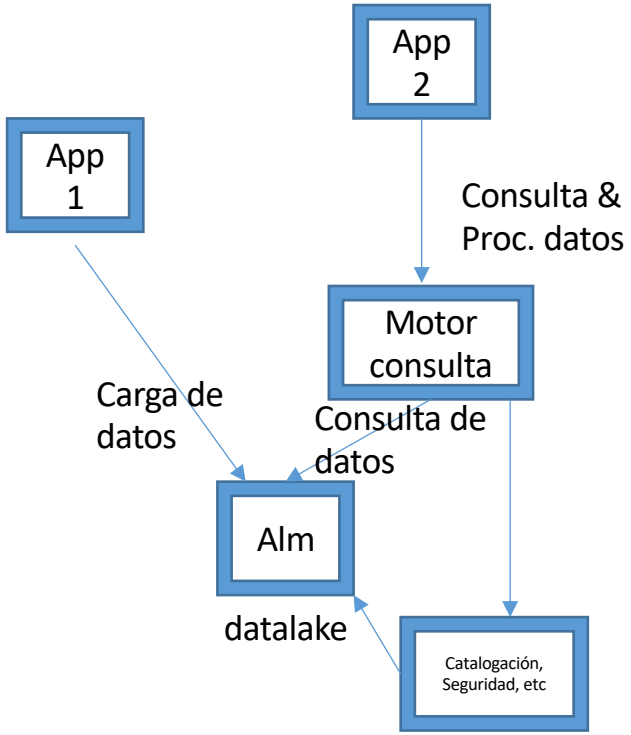
- Son sistemas de almacenamiento, indexación, gestión, búsqueda y recuperación de datos (generalmente a nivel de registros en el modelo relacional SQL)
- Motores – Data Base Management System
- Motores RELACIONALES basados en SQL – RDBMS (Relational DBMS)
  - MUY UTILIZADO EN CONSULTAS/Procesamiento en analítica / ciencia de datos
- Visión emergente: Motores NoSQL, de muchos tipos.
- Los programas acceden a los datos mediante Lenguajes de BD (Ej: SQL) o mediante APIs.
- Los datos son almacenados de forma optimizada de acuerdo al tipo de datos.
- Las tablas y registros se definen:
  - A Priori, Basados en Esquemas de Escritura (schema-on-write),
  - A posteriori, Esquemas de Lectura (schema-on-read)
  - O no se definen, y la aplicación los interpreta, o sin Esquemas (Schemaless)

El papel de los RDBMS y el SQL en Analítica & Big Data

- Los RDBMS para conformar Data warehouse tradicionales:
  - Desapareciendo
  - Evolucionando a los datalakes, DWH modernos, nubes y Lakehouse
- SQL como motor de consulta y lenguaje de procesamiento:
  - Muy muy vigente

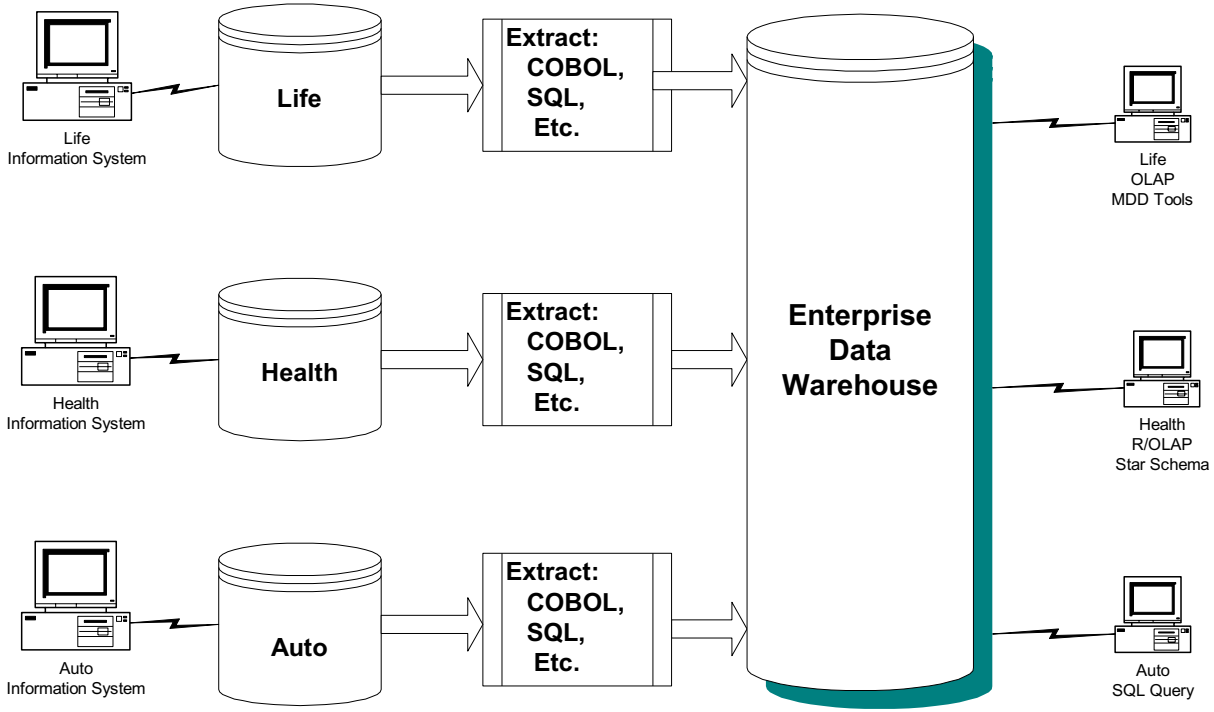
DBMS clásicos vs Motores de Consulta

- Los DBMS clásicos:
  - Es el software tradicional de bases de datos, como Oracle, SQLServer, Mysql, Postgress, por mencionar el mundo Relacional/SQL, pero también aplica a NoSQL como MongoDB en Documental o key-value como Redis.
  - El DBMS gestiona directamente el almacenamiento y está oculto a las aplicaciones.
  - Solo se pueden GESTIONAR datos por el DBMS
- Motores de Consulta:
  - Se tiene un software para realizar las consultas, procesamiento y algunas funciones de gestión (muy muy utilizado en analítica moderna)
  - PERO el almacenamiento es independiente del Motor de Consulta, y hay otras aplicaciones que llevan datos a dicho almacenamiento.
  - Es una combinación de Almacenamiento con Motor
    - Este Almacenamiento, será lo que llamaremos DATALAKE.
    - Ese Datalake requiere otras operaciones como Catalogación, Seguridad, Gobernanza, etc. QUE no son provistas por el Motor de consulta, y en un DBMS clásico lo incluye.

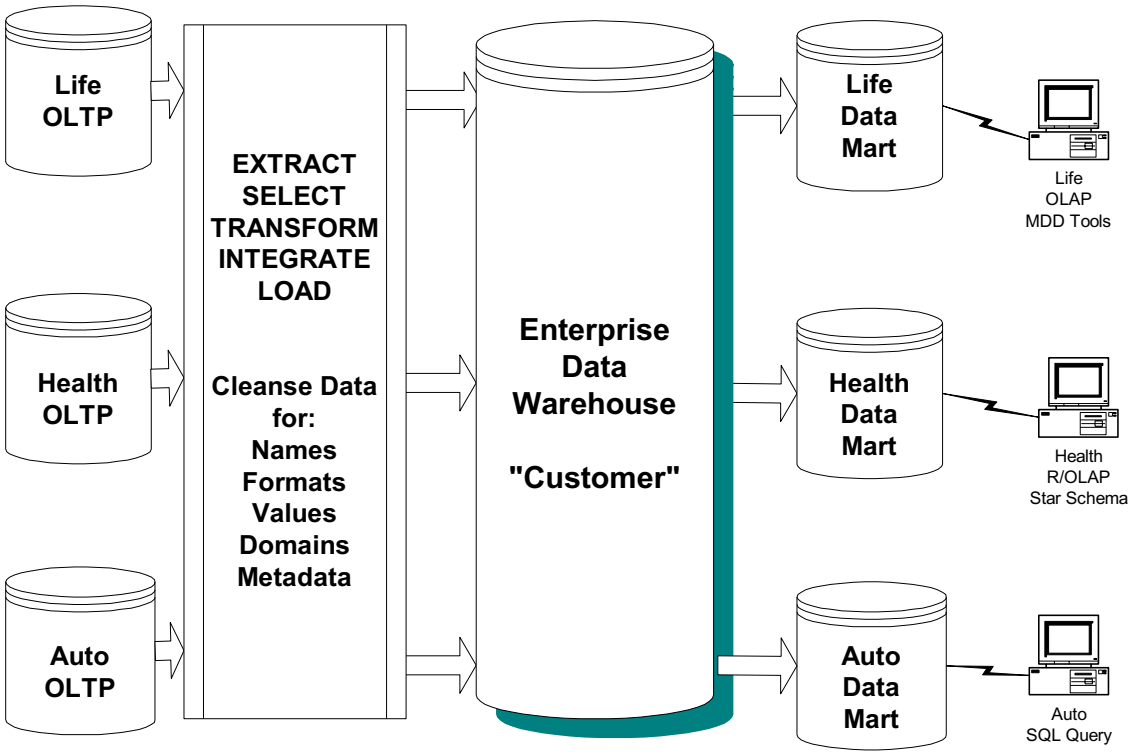


4. Data Warehouse tradicionales

- Sistemas que extraen datos operacionales, los cargan en un DWH en un modelo especial, y son procesados por un Servidor OLAP -> Apps (reporting, visualización)
- DWH es un motor RDBMS
- Transformación de los modelos E-R a (multi)dimensionales.

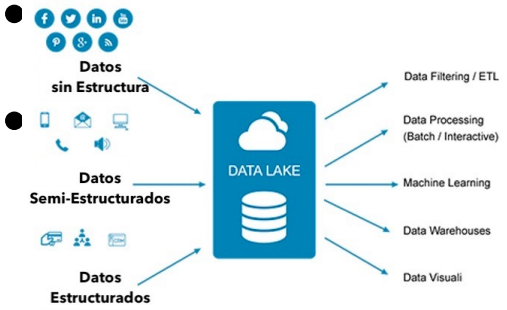


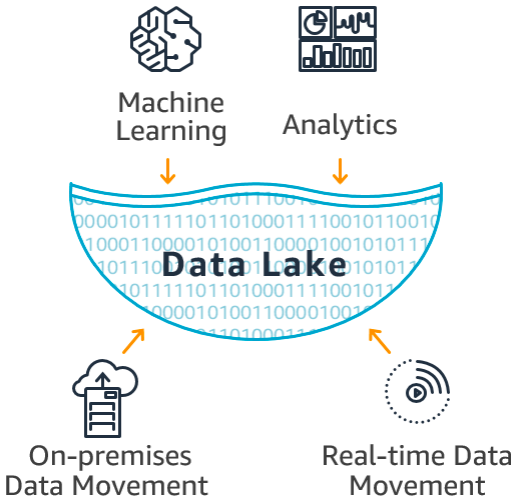
Data Marts Estructurados

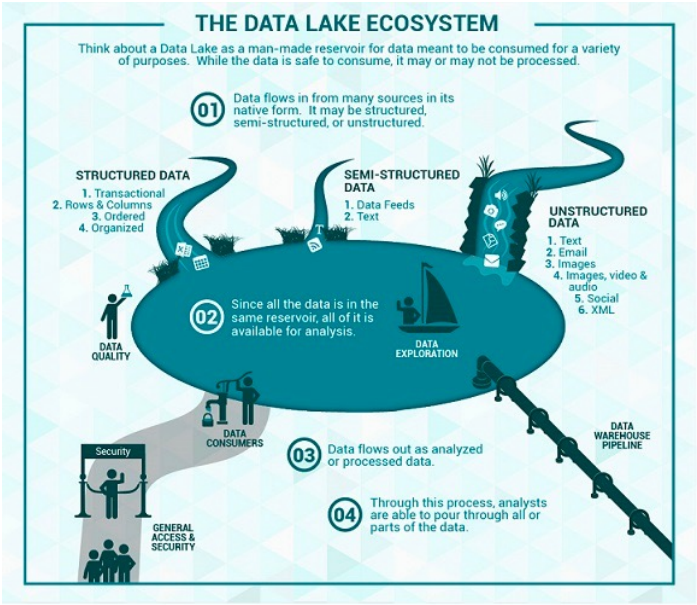


5. Datalakes

- E-L T1-use1, T2-use2, etc
- Schema-on-read
- Zonas
- Linajes
- Catálogos
-
-

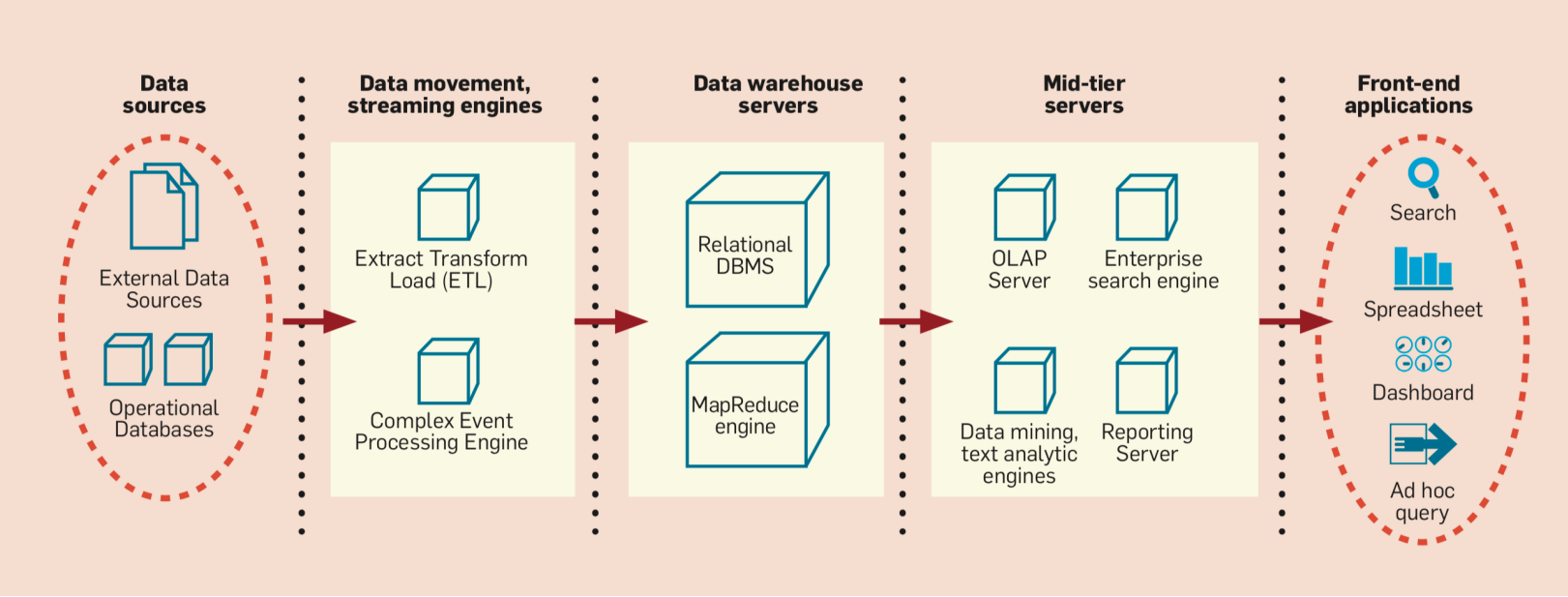




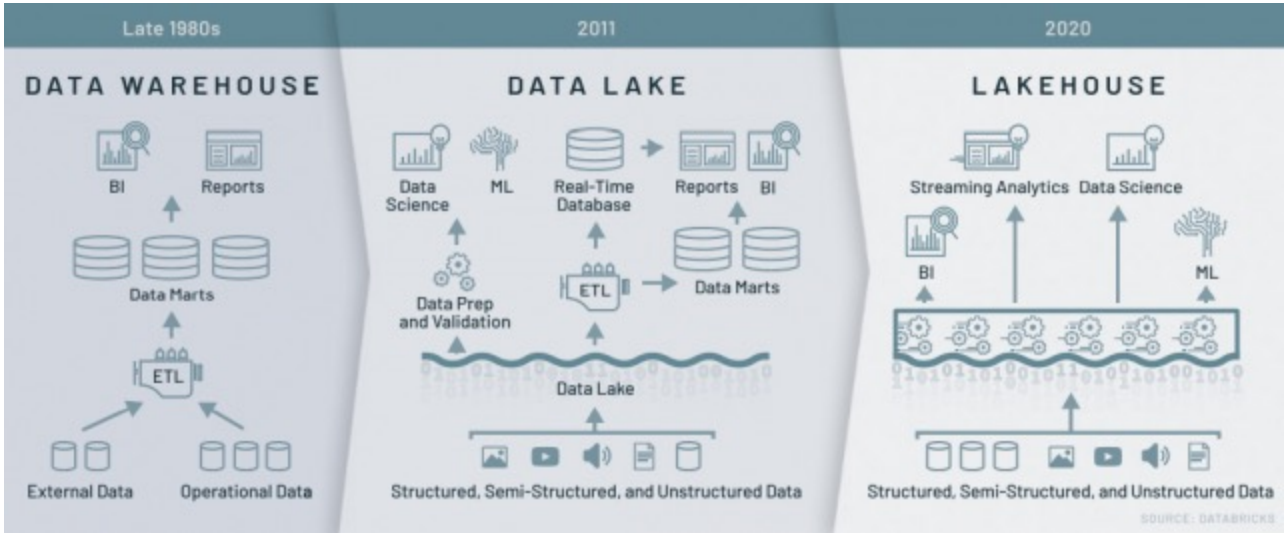


DATA WAREHOUSE	vs.	DATA LAKE
structured, processed	DATA	structured / semi-structured / unstructured, raw
schema-on-write	PROCESSING	schema-on-read
expensive for large data volumes	STORAGE	designed for low-cost storage
less agile, fixed configuration	AGILITY	highly agile, configure and reconfigure as needed
mature	SECURITY	maturing
business professionals	USERS	data scientists et. al.

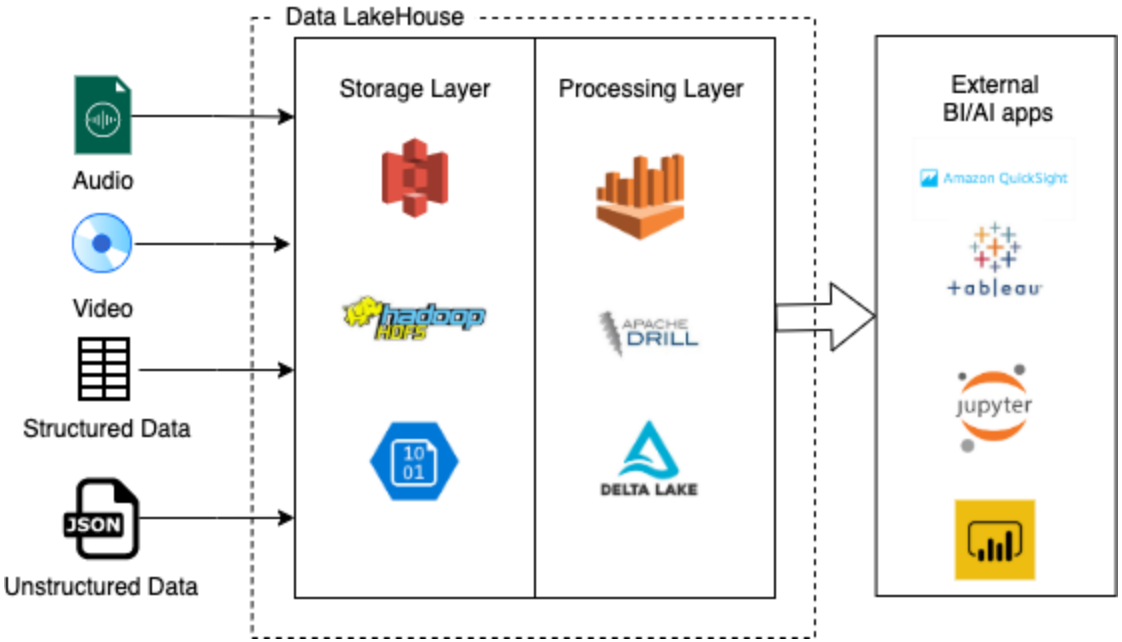
4. DWH moderno



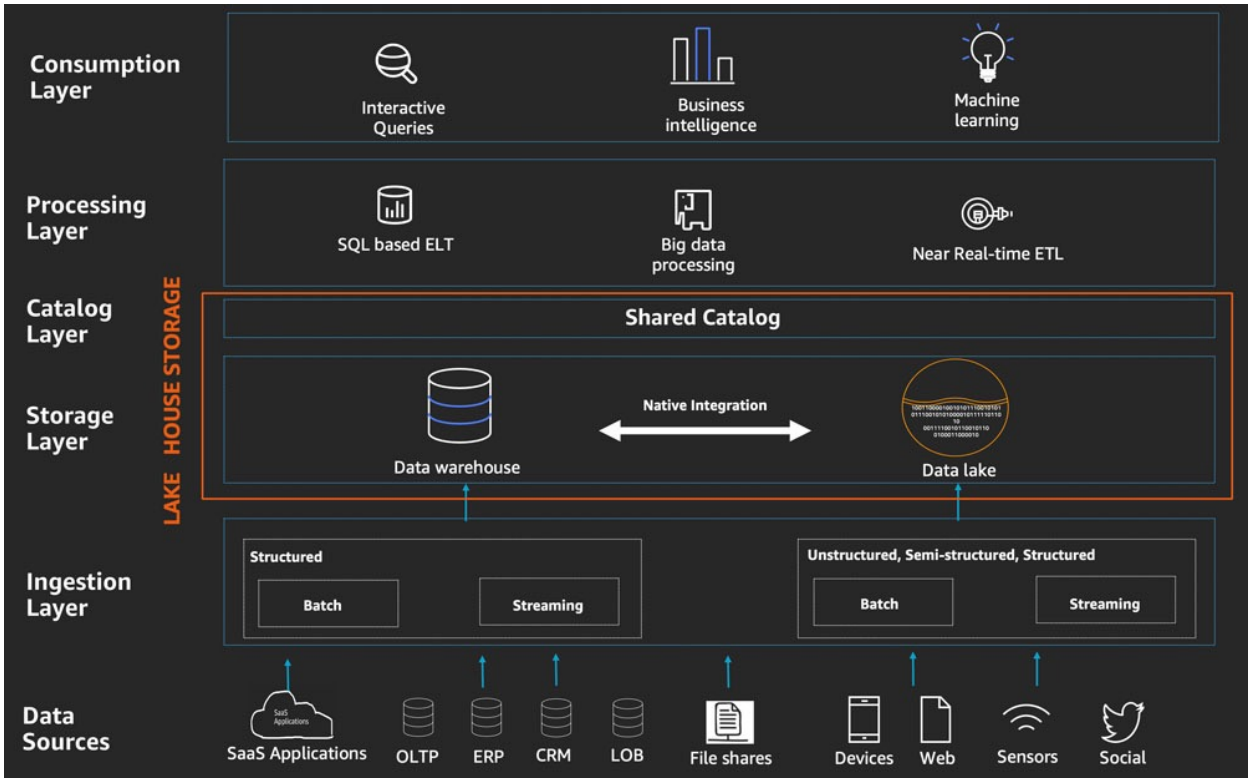
Lakehouse


Late 1980s
2011
2020
DATA WAREHOUSE
BI
Reports
Data Marts
ETL
External Data
Operational Data
DATA LAKE
Data Science
ML
Real-Time Database
Reports
BI
Data Prep and Validation
ETL
Data Marts
Data Lake
Structured, Semi-Structured, and Unstructured Data
LAKEHOUSE
Streaming Analytics
Data Science
BI
ML
Structured, Semi-Structured, and Unstructured Data

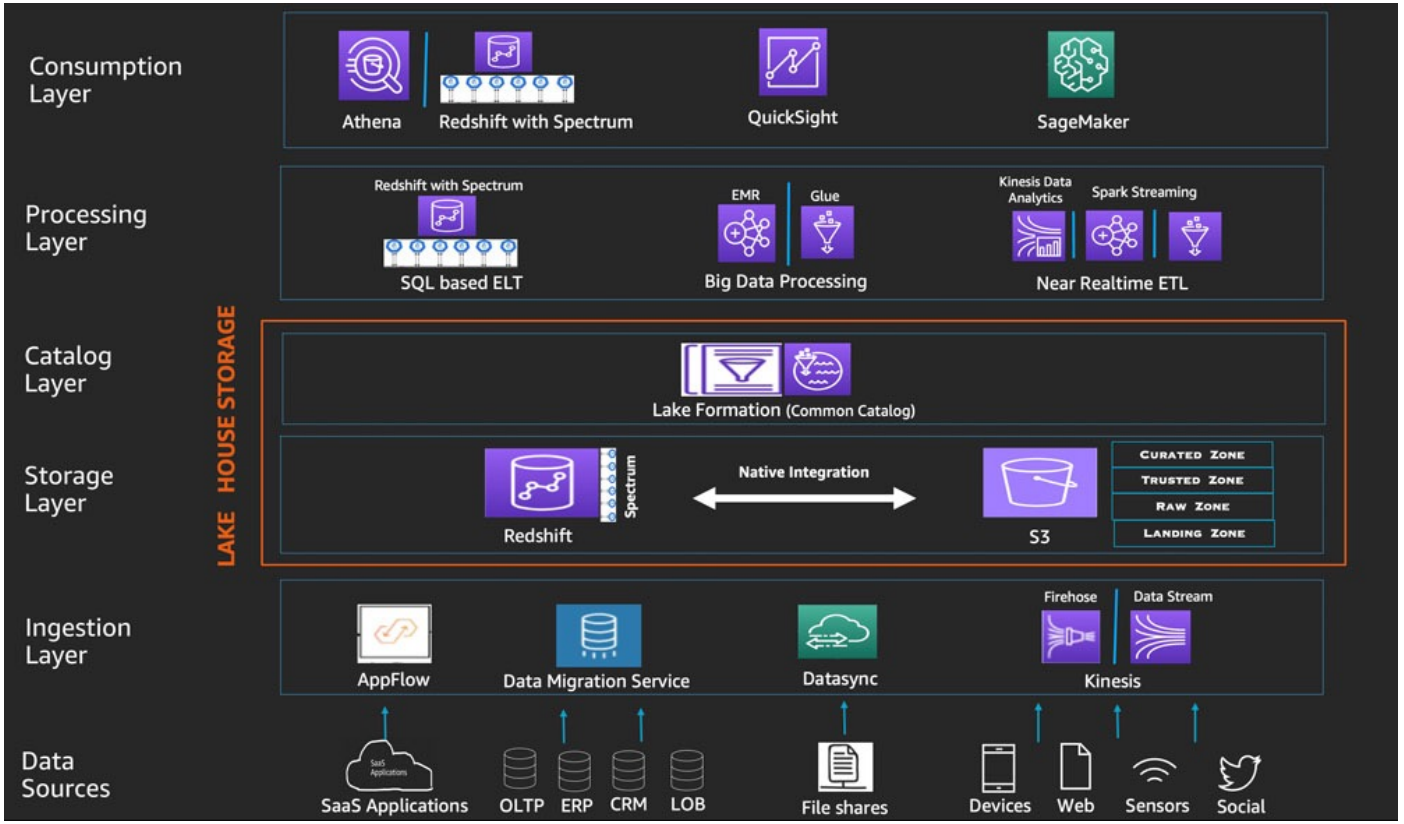
Lakehouse


Data LakeHouse
Storage Layer
Processing Layer
External
BI/AI apps
Audio
Video
Structured Data
Unstructured Data
JSON
hadoop
HDFS
APACHE
DRILL
DELTA LAKE
Amazon QuickSight
+ableau
jupyter

Lakehouse


Consumption Layer
Interactive Queries
Business intelligence
Machine learning
Processing Layer
SQL based ELT
Big data processing
Near Real-time ETL
Catalog Layer
Shared Catalog
LAKE HOUSE STORAGE
Storage Layer
Data warehouse
Native Integration
Data lake
Ingestion Layer
Structured
Batch
Streaming
Unstructured, Semi-structured, Structured
Batch
Streaming
Data Sources
SaaS Applications
OLTP
ERP
CRM
LOB
File shares
Devices
Web
Sensors
Social

Lakehouse: una implementación en AWS



ST0263 – TÓPICOS ESPECIALES EN TELEMÁTICA

2023-1
Introducción a BIG DATA
parte2

Universidad EAFIT

SQL vs NoSQL

SQL vs NoSQL

- SQL – Structured Query Language
- NoSQL – Not Only SQL
- NoSQL son bases de datos no fundamentadas en modelos relacionales con datos no estructurados o semi-estructurados.
- NO contienen modelos de almacenamiento ni procesamiento orientados a filas y columnas (Nota: el modelo sencillo de SQL esta super-vigente en Data Science relacionados con los DataFrames, aunque sin la rigurocidad de Esquemas SQL)
- Normalmente estos motores NoSQL son diseñados para alta disponibilidad, escalabilidad y rendimiento.
- NoSQL:
  - No preocupación por relación entre entidades
  - No muy preocupadas por la consistencia de datos ni transaccional
  - No problemas con duplicación de datos

Scheme on Write (SQL) vs Scheme on Read (NoSQL)

- SQL
  - Antes de almacenar cualquier dato, se TIENE que definir un esquema de tabla
  - Altamente estricto
  - Rígido
- NoSQL
  - Los datos se almacenan sin Esquema previo
  - Al momento de lectura, puede o no tener un esquema
  - El esquema puede ser dinámico y variable entre lecturas

- Diferencia FUNDAMENTAL entre Almacenamiento y Recuperación
  - OLTP-transaccionales
  - Data warehouse vs Big Data (datalakes)

Vistazo a bases de datos NoSQL

- Características:
  - El modelo de datos no se basa en tablas de columnas de longitud fija. Las bases de datos NoSQL pueden alojar datos semiestructurados o no estructurados con diferentes longitudes.
  - Los datos NoSQL no dependen de tablas predefinidas (no Schema-on-write). Se puede crear una tabla y añadir registros dinámicamente.
  - Hay más flexibilidad en el uso del almacenamiento porque no es necesario normalizar los datos para ahorrar espacio en disco.
  - El concepto de unir registros de múltiples tablas no es tan importante como lo es en las bases de datos relacionales.

Principales tipos NoSQL (1)

- bases de datos de documentos
  - se utilizan para almacenar documentos JSON (Java Script Object Notation).
  - Estos documentos JSON pueden tener diferentes esquemas y elementos dentro de ellos, pero se puede utilizar la misma tabla para alojar todos los datos.
  - Ej: MongoDB y Couchbase
- bases de datos key-value
  - almacenan datos en tuplas (clave,valor).
  - el valor puede ser una imagen, un documento JSON o incluso una cadena de texto o un número.
  - Ej: Amazon DynamoDB, Redis y Aerospike

Principales tipos NoSQL (2)

- Las bases de datos orientadas a columnas son lo contrario de las bases de datos relacionales basadas en filas.
  - En las bases de datos relacionales, los datos se guardan como filas en una tabla. Cada línea forma parte de un bloque de datos, lo que significa que todos los valores de columna para la misma línea se encuentran en un bloque de almacenamiento.
  - En una base de datos orientada a columnas, los valores de la misma columna se alojan en bloques de datos contiguos.
  - Ej:Apache Cassandra

Principales tipos NoSQL (3)

- Las bases de datos de GRÁFOS no representan datos en filas, columnas, tablas o familias de columnas.
  - utilizan componentes como ”edges", ”nodes" y ”properties" para almacenar y relacionar datos.
  - Los nodos son como entidades en una base de datos relacional y tendrán algunas propiedades.
  - La relación entre dos nodos también puede tener propiedades.
  - EJ: Neo4j o AWS Neptune

Casos de uso NoSQL

- NoSQL se utiliza en aplicaciones modernas, ligeras y habilitadas para Internet, como medios sociales, captura de secuencias de clics, carritos de la compra, chatbots, preferencias de los usuarios, ad-techs, IoT, mensajería, personalización, aplicaciones móviles o juegos en línea.
- Altamente utilizadas en sistemas de big data y analítica avanzada

3. Introducción a Big Data

Qué es Big Data

Es el área de conocimiento dentro de la informática que se ocupa de solucionar problemas de transporte, almacenamiento, procesamiento, análisis y aprovechamiento de datos caracterizados como Big Data, esto ocurre especialmente cuando las tecnologías tradicionales no son suficientes.

Características Big Data

Las 5 Vs pueden ser utilizadas para diferenciar conjuntos de datos categorizados como “Big Data” de los demás conjuntos de datos.

- Volumen
- Velocidad
- Variedad
- Valor
- Veracidad

volumen

- petabytes de datos emitidos por una variedad de fuentes, ya sean redes sociales, sensores, blockchain, video, audio o incluso transaccionales

variedad

- Estructurados
- Semi-estructurados
- No estructurados

velocidad

- Velocidad de llegada de los datos
- Velocidad de procesamiento de los datos
- Velocidad de salida de los datos

4. Ciclo de vida de datos

Ciclo de vida
Desde la Ingeniería de Datos

- DATA SOURCE: Generación (datasources) y Adquisición de datos.
- INGEST: Transmitir los datos de las fuentes al almacenamiento o procesamiento (ingest,collect)
- STORAGE: Sistemas de almacenamiento masivo y distribuido
- PROCESAMIENTO: Preparación, Análisis & Data Analytics
- APLICACIONES:
  - Visualización & Dashboards
  - Negocios, Científicos, Económicos, … para casi todos los dominios
  - Apps por: Datos estructurados, texto, web, multimedia,redes, móviles (bots)
  - Empresariales, IoT, Social Networks, Health, e-Science, Smart grid, etc.

Ciclo de vida

- DATA SOURCE: Generación y Adquisición de datos (ETL: Extract & Transform & Load)
  - Generación: Fuente
    - Datos empresariales operacionales - OLTP
    - Datos IoT, Redes Sociales
    - Datos científicos en muchos otros campos
- INGEST: Adquisición de datos (ETL: Extract & Transform & Load) / Streaming
  - Adquisición: (ingest,collect)
    - Datasets y/o real-time -> Colecciones de datos
    - Desde bases de datos
    - Desde eventos en tiempo real (streaming), IoT, Logs, Click Stream, etc.
    - Transformación de datos & Pre-procesamiento de datos
      - Integración
      - Limpieza
      - Eliminación de redundancia

Ciclo de vida

- STORAGE: Sistemas de almacenamiento masivos y distribuidos.
  - Datos transaccionales vs Datos para Analítica
  - CAP (Consistency & Availability & Partition Tolerance) en almacenamiento distribuido
  - Almacenamiento para datos masivos
  - Sistemas de almacenamiento distribuidos
  - Mecanismos de almacenamiento:
    - Tecnologías tradicionales: DAS, SAN, NAS
    - Distributed File System (NFS, GFS, HDFS)
    - Database system (SQL & NoSQL)
      - Tradicionales -> SQL
      - Key-Value, Column-oriented, Document, etc, -> NoSQL
    - Repositorios, ej: tipo CKAN

Ciclo de vida

- PROCESAMIENTO - INGENIERÍA:
  - Arquitectura para análisis de datos Big Data
    - Diversidad de fuentes
    - Diversidad de mecanismos de ingesta
      - Streaming - Real-Time
        - Ej: Kafka
      - Bases de datos
      - datasets
    - Arquitecturas:
      - Clúster de procesamiento paralelo usando RDBMS tradicionales
      - Plataformas de computación basadas en memoria (ej: greenplum de EMC, HANA de SAP, Spark)
    - Offline (batch)
      - El tiempo de respuesta no es critico
      - Hadoop, Scribe, Chukwa, etc.
    - Streaming: end-to-end, parcial, canalizadores, procesadores de flujo, almacenamiento, aplicación
    - Arquitecturas:
      - Batch, tradicional, viene de ETL.
      - Streaming, kappa
      - Hibridas -> lambda

Ciclo de vida

- PROCESAMIENTO – CIENCIA DE DATOS: Análisis & Data Analytics (big data analytics)
  - Análisis exploratorio de datos
  - Análisis de datos tradicional – OLAP-DWH
  - Métodos descriptivos
  - Métodos de analítica en Big Data
  - Arquitectura para análisis de datos Big Data
    - Análisis Real-Time vs Streaming vs Batch
  - Data Mining & ML & DL
  - Aplicaciones especificas: NLP, ComputerVision, Speech, etc.

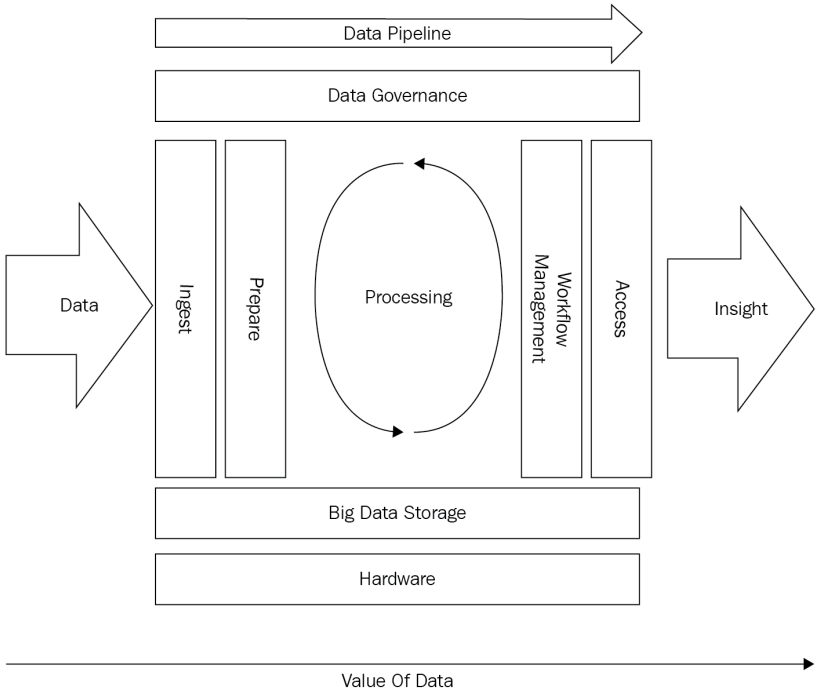
Ciclo de vida

- APLICACIONES:
  - Principalmente
    - Visualización
    - APIs
    - NLP
    - Chatbots
    - Sistemas de Recomendación

5. Arquitectura de Referencia

Arquitectura de referencia para un sistema que consume muchos datos

- Varias vistas y perspectivas
- Esta es la vista más simple de arquitectura
- Cuales componentes identifica?
- Ocho (8) componentes principales



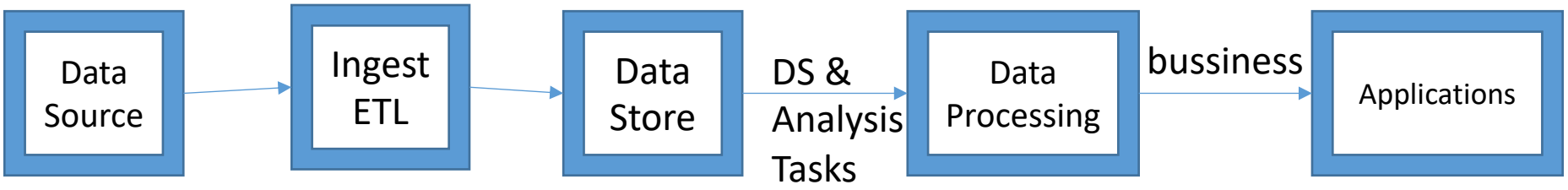
Arquitecturas básicas

BATCH

STREAMING (kappa)

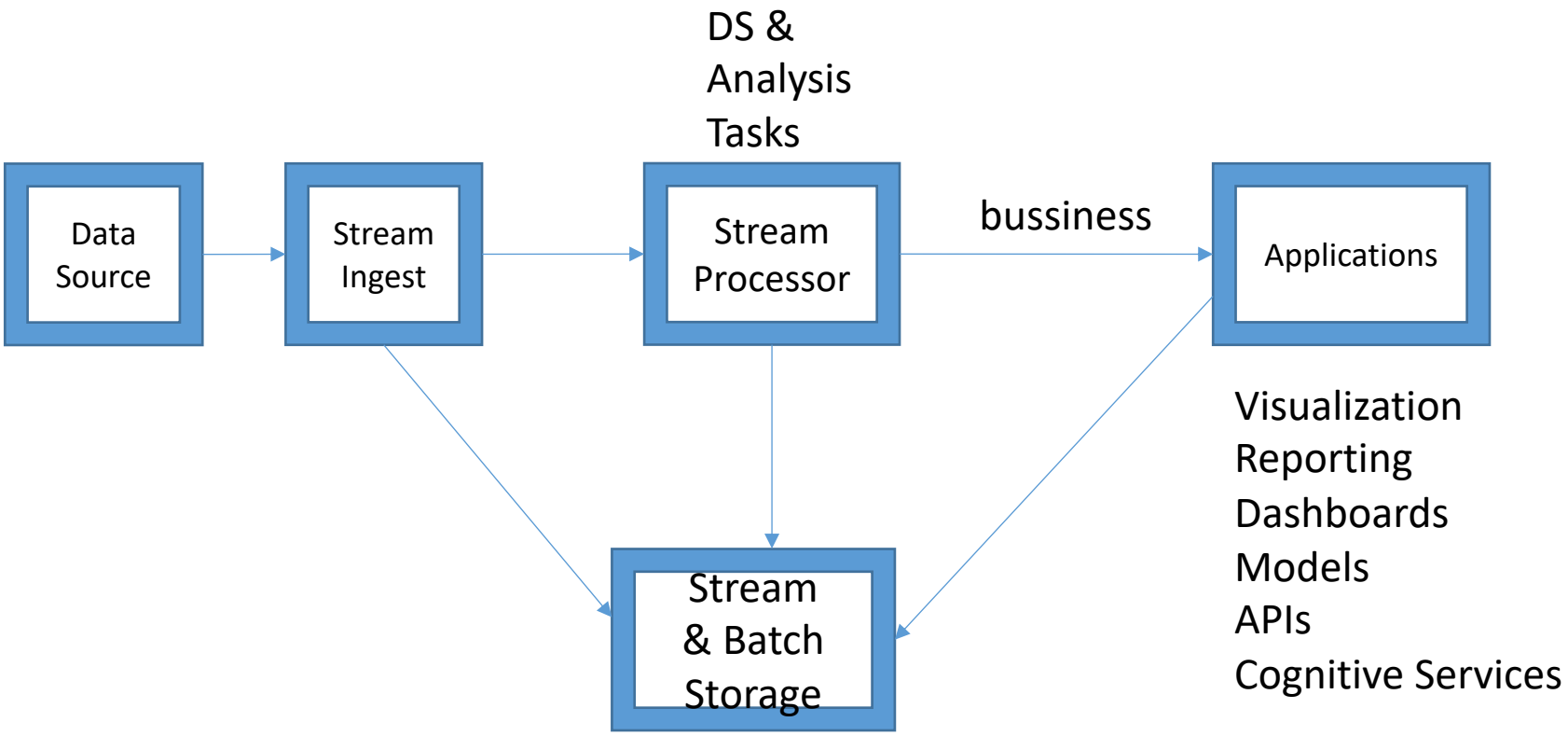
HIBRIDAS (lambda)

Arquitectura Batch

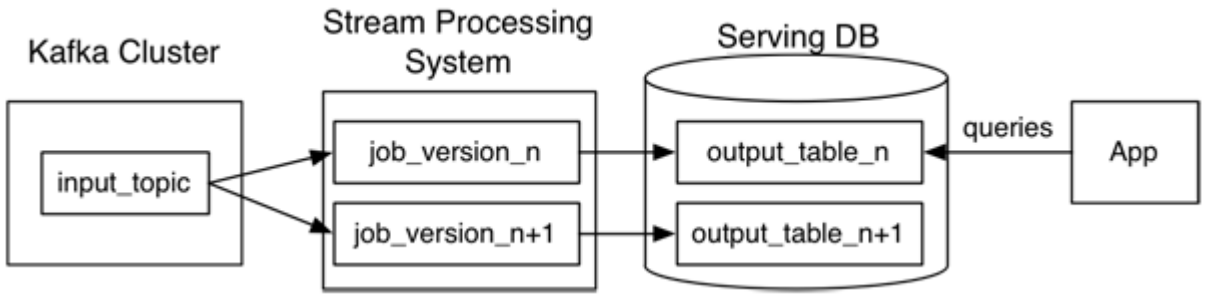


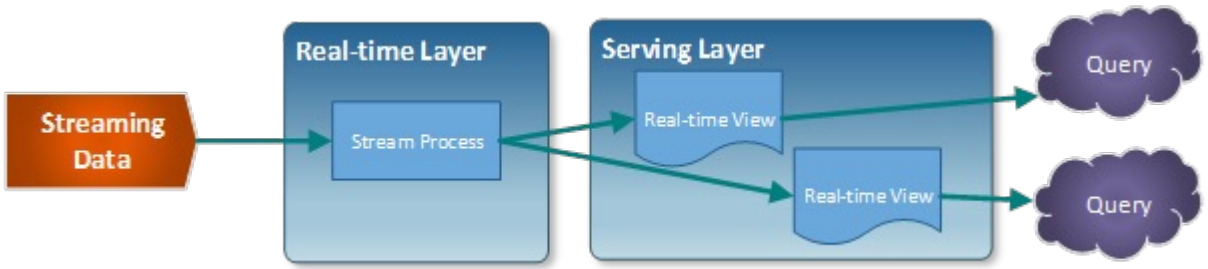
Visualization
Reporting
Dashboards
Models
APIs
Cognitive Services

Arquitectura Streaming



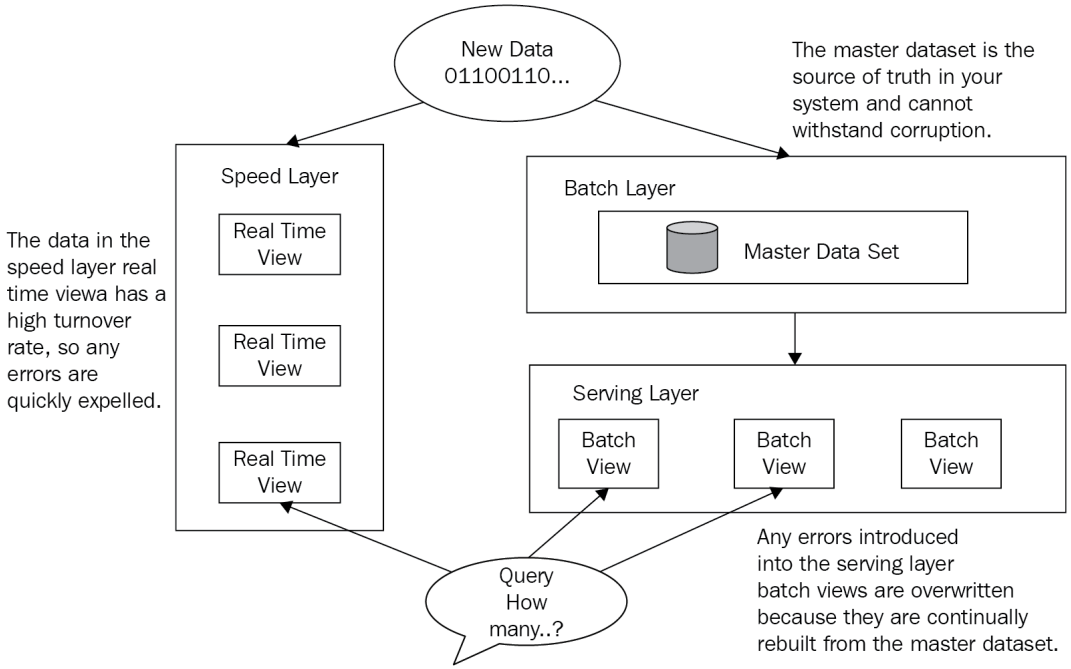
Kappa architecture (streaming)

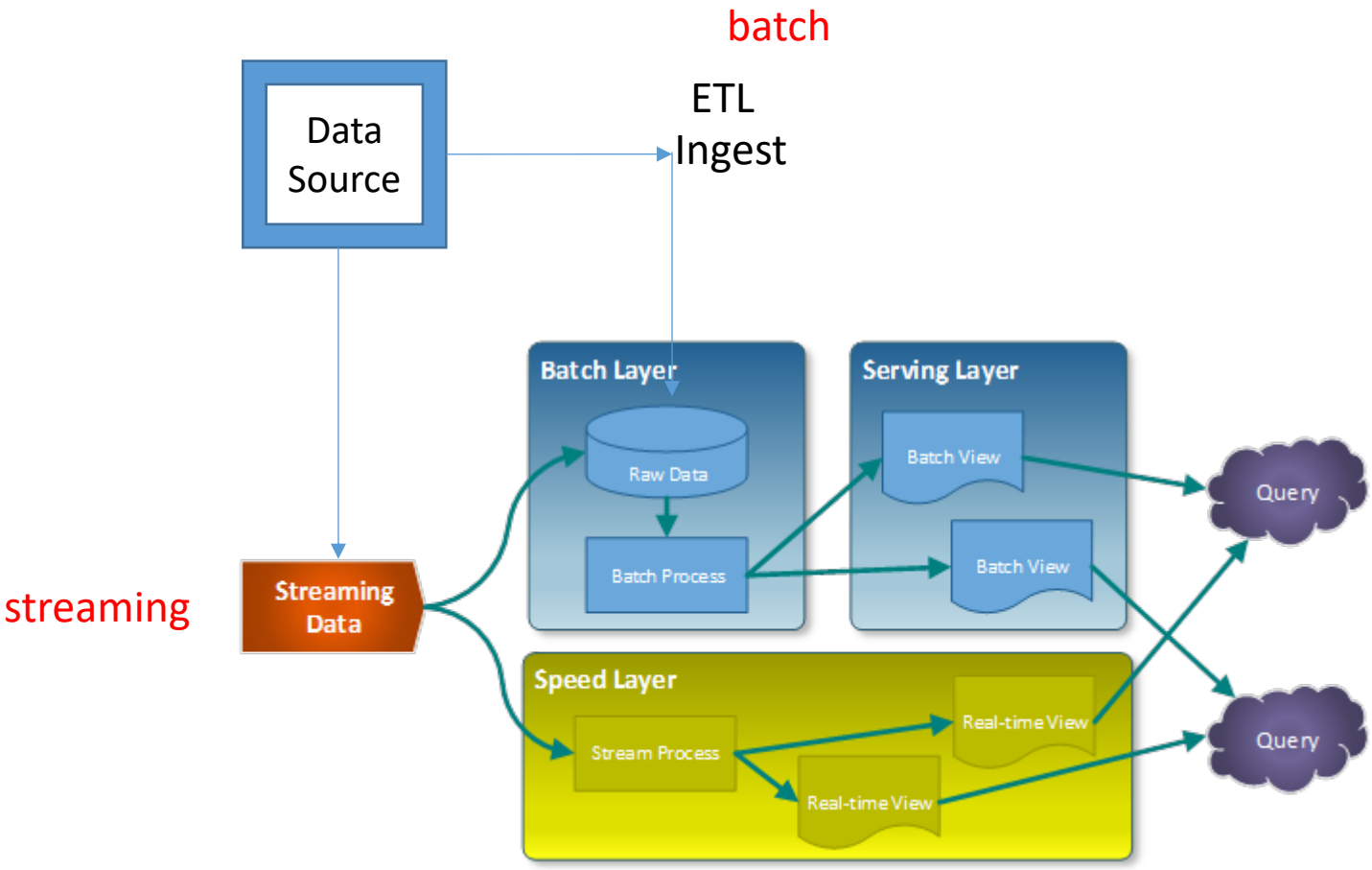




Arquitectura Hibrida

Originada desde Batch y requerimiento de actualizar los modelos en streaming




batch
Data Source
ETL
Ingest
Batch Layer
Raw Data
Batch Process
Serving Layer
Batch View
Batch View
Query
streaming
Streaming Data
Speed Layer
Stream Process
Real-time View
Real-time View
Query